МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт автоматики и информационных технологий

Кафедра «Программная инженерия»

Тамабаева Кунасыл Муратбековна

Применение эффективных алгоритмов кластерного анализа при обработке цифровых данных

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

5В070400 – Вычислительная техника и программное обеспечение

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт автоматики и информационных технологий

Кафедра «Программная инженерия»

ДОПУЩЕН К ЗАЩИТЕ
Заведующая кафедрой ПИ
канд. физ-мат. наук, профессор
_____А.Н. Молдагулова
«_20_» _05_____ 2022 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Применение эффективных алгоритмов кластерного анализа при обработке цифровых данных»

По специальности 5В070400 – Вычислительная техника и программное обеспечение

Выполнила Тамабаева К.М.

Рецензент
старший преподаватель, доктор Ph.D.
_____ Даркенбаев Д. К.
«_20_» _мая_____ 2022 г.

Научный руководитель
сениор лектор, доктор Ph.D.
_____ Черикбаева Л.Ш.
«_20_» _мая_____ 2022 г.

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени К.И.Сатпаева

Институт автоматики и информационных технологий

Кафедра «Программная инженерия»

5В070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ
Заведующая кафедрой ПИ
канд. физ-мат. наук, профессор
_____А.Н. Молдагулова
«20» 05 2022 г.

ЗАДАНИЕ
на выполнение дипломного проекта

Обучающемуся *Тамабаевой Кунасыл Муратбековне*
Тема: *Применение эффективных алгоритмов кластерного анализа при обработке цифровых данных*
Утверждена *приказом проректора по академической работе* № 489-П/О
от "24" 12 *2021 г.*
Срок сдачи законченного проекта "25" 05 *2022 г.*
Исходные данные к дипломному проекту: *данные, полученные с электрокардиограмм, используемых в медицине.*
Перечень подлежащих разработке в дипломном проекте вопросов:
а) применение алгоритмов кластеризации
б) использование метрик для сравнения моделей машинного обучения
в) получение результатов эффективности моделей
г) сравнительный анализ эффективных моделей
Перечень графического материала (с точным указанием обязательных чертежей): *представлены 25 слайда презентации.*
Рекомендуемая основная литература: *из 20 наименований.*

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области, определение целей дипломного проекта	10.02.2022	Выполнено
2. Анализ данных, ознакомление с feature значениями	15.02.2022	Выполнено
3. Предобработка данных	25.02.2022	Выполнено
4. Применение алгоритмов кластеризации. Получение первоначальных результатов моделей	10.03.2022	Выполнено
5. Использование основных метрик задач кластеризации для определения эффективности алгоритмов	15.04.2022	Выполнено
6. Сравнительный анализ эффективных алгоритмов кластеризации	25.04.2022	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
Нормоконтролер	Жекамбаева М.Н. Доктор Ph.D., ассоциированный профессор	20.05.22	
Программное обеспечение	Марғұлан Қ. магистр техн.наук, лектор	19.05.22г	

Научный руководитель _____ Черикбаева Л.Ш.

Задание принял к исполнению обучающийся _____ Тамабаева К.М.

Дата «17» 11 2022 г.

АННОТАЦИЯ

Дипломный проект преследует цель нахождения эффективного алгоритма кластеризации сигналов электрокардиограммы, используемый в медицине. Были применены несколько алгоритмов машинного обучения, а также выведены самые эффективные из них.

Дипломная работа состоит из введения, основной части (исследовательский раздел, технологический раздел, проектный раздел), заключения, списка использованной литературы и приложений.

В исследовательском разделе рассматриваются цели и задачи дипломного проекта, основные определения, используемые термины, а также затрагивается вопрос актуальности проблемы сердечных заболеваний, описываются различия нормальных и аномальных сигналов ЭКГ.

Во втором разделе описана технологическая часть работы, в котором рассматриваются инструменты, использованные библиотеки, обоснование выбора языка программирования.

В проектном разделе описаны используемые алгоритмы кластеризации, способы реализации в дипломной работе.

В экспериментальном разделе приводится реализация кластеризации и результаты работы.

Дипломная работа состоит из 40 страниц, 13 рисунков и 2 приложений.

В работе использовались 20 источников.

Данную работу предлагается рассматривать как основание для будущих работ, преследующих цели использования методов ML в медицинских оборудованиях, в частности электрокардиограммы.

АНДАТПА

Дипломдық жоба медицинада қолданылатын электрокардиограмма сигналдарын кластерлеудің тиімді алгоритмін табуға бағытталған. Машиналарды оқытудың бірнеше алгоритмдері қолданылды, сонымен қатар олардың ең тиімділері алынды.

Дипломдық жұмыс кіріспеден, негізгі бөлімнен (зерттеу бөлімі, технологиялық бөлім, жобалау бөлімі), қорытындыдан, пайдаланылған әдебиеттер тізімі мен қосымшалардан тұрады.

Зерттеу бөлімінде дипломдық жобаның мақсаттары мен міндеттері, негізгі анықтамалар, қолданылатын терминдер қарастырылады, сонымен қатар жүрек аурулары мәселесінің өзектілігі, ЭКГ қалыпты және қалыптан тыс сигналдарының айырмашылықтары сипатталған.

Екінші бөлімде жұмыстың технологиялық бөлігі сипатталған, онда құралдар, пайдаланылған кітапханалар, бағдарламалау тілін таңдаудың негіздемесі қарастырылған.

Жоба бөлімінде қолданылатын кластерлеу алгоритмдері, дипломдық жұмыста іске асыру әдістері сипатталған.

Эксперименттік бөлімде кластерлеуді іске асыру және жұмыс нәтижелері келтіріледі.

Диссертация 40 беттен, 13 суреттен және 2 қосымшадан тұрады.

Жұмыста 20 дереккөз пайдаланылды.

Бұл жұмысты медициналық жабдықтарда, атап айтқанда электрокардиограммада ML әдістерін қолдану мақсаттарын көздейтін болашақ жұмыстардың негізі ретінде қарау ұсынылады.

ANNOTATION

The thesis project aims to find an effective algorithm for clustering electrocardiogram signals used in medicine. Several machine learning algorithms were applied, and the most effective of them were derived.

The thesis consists of an introduction, the main part (research section, technology section, project section), conclusion, list of references and appendices.

The research section discusses the goals and objectives of the diploma project, the main definitions, the terms used, and also touches on the relevance of the problem of heart disease, describes the differences between normal and abnormal ECG signals.

The second section describes the technological part of the work, which examines the tools, the libraries used, and the rationale for choosing a programming language.

The project section describes the clustering algorithms used, methods of implementation in the thesis.

The experimental section provides the implementation of clustering and the results of the work.

The thesis consists of 40 pages, 13 drawings and 2 appendices.

20 sources were used in the work.

This work is proposed to be considered as a basis for future work aimed at using ML methods in medical equipment, in particular electrocardiograms.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Ежегодно в Казахстане более 40 тысяч человек переносят инсульт. По данным ВОЗ число заболеваний растет с каждым годом и является одним из основных причин смертности в РК.

Использование алгоритмов машинного обучения может помочь точно определять диагноз заболеваемого, обнаруживать болезни сердца с помощью эффективных моделей кластеризации по цифровым данным, полученных с электрокардиограмм.

Электрокардиограммы (ЭКГ) - это неинвазивный и недорогой метод, обычно используемый кардиологами в их обычной клинической практике. Они часто используются для выявления нарушений сердечного ритма, измеряя электрическую активность сердца в течение определенного периода времени.

С помощью данных с ЭКГ можно определить первые нарушения в работе сердца, оценивается динамика сердечных патологий. Для анализа были взяты сигналы, записанные электрокардиограммой, которые зарегистрировали электрические импульсы, возникающие в сердце. В качестве датасета были использованы данные из архива MIT-BIH.

Сердечно-сосудистые заболевания являются одним из основных причин смертности в Казахстане и во всем мире. Цель использования методов ML в медицинских ПО и оборудованиях обусловлено тем, что точность постановки диагноза, в нашем случае обнаружения нормальных и ненормальных сигналов, повышается.

Использование эффективного алгоритма повышает точность постановки диагноза сердечных болезней. Использование методов машинного обучения исключает человеческий фактор и соответствует результатам.

В данной работе будут рассмотрены эффективные методы обработки и кластеризации сигналов ЭКГ.

1 Исследовательский раздел

1.1 Цель разработки

Цель данной дипломной работы заключается в сравнении алгоритмов кластерного анализа. Выяснение эффективного алгоритма для кластеризации цифровых медицинских данных.

Задачи дипломной работы:

- применение алгоритмов кластеризации;
- использование метрик для сравнения моделей машинного обучения;
- получение результатов эффективности моделей;
- сравнительный анализ эффективных моделей.

1.2 Определения, термины и сокращения

В таблице 1.1 сформулированы все термины и сокращения, которые используются в предметной области разрабатываемого проекта, а также специфические термины, связанные с программной реализацией проекта и используемыми технологиями при разработке.

Таблица 1.1 – Сокращения, термины и их определения

Сокращение или термин	Определение
ЭКГ	(сокр. от электрокардиограмма)
Kmeans	Метод k-средних
DBSCAN	(сокр. от Density-based spatial clustering of applications with noise)
SpectralClustering	Спектральная кластеризация
AglomerativeClustering	Агломеративный метод кластеризации
MinMaxScaler	Библиотека scikit-learn, позволяющая произвести нормализацию данных
PCA	(сокр. от Principal component analysis)
t-SNE	(сокр. от t-distributed stochastic neighbor embedding) Стохастическое вложение соседей с t-распределением
ML	(сокр. от Machine Learning)
KNN	(сокр. от англ. k-nearest neighbors) Метод k-ближайших соседей

Продолжение таблицы 1.1

Сокращение или термин	Определение
TP	(сокр. от True Positive) – истинно положительные
FP	(сокр. от False Positive) - ложноположительные
FN	(сокр. от False Negative) - ложноотрицательные
TN	(сокр. от True Negative) – истинно отрицательные
GaussianMixture	Модель смеси Гауссовых распределений
ВОЗ	(сокр. от Всемирная организация здравоохранения)
РК	(сокр. от Республика Казахстан)

1.3 Предметная область

Предметной областью данного дипломного проекта являются машинное обучение и медицина, в частности рассматривающая сердечные заболевания.

1.4 Актуальность проблемы сердечных заболеваний

Ежегодно в Казахстане более 40 тысяч человек переносят инсульт, из которых 5 тысяч погибает в течение первых 10 дней. В данном проекте были использованы несколько алгоритмов машинного обучения для кластеризации медицинских данных.

Распространенность инсульта по всему миру ежегодно растет, как в экономически развитых, так и развивающихся странах и составила по данным «Heart Disease and Stroke Statistics-2016 Update» от ассоциации American Heart 33 млн. человек, при чем 16,9 млн. это впервые установленные случаи инсульта, из которых 5,2 млн. лица в возрасте до 65 лет. К сожалению, Казахстан не является исключением по данным показателям. Из официальных источников Министерства здравоохранения РК сказано, что более сорока тысяц случаев инсульта, среди которых около пяти тысяч погибает в первые десять дней, а также пять тысяч погибает после выписки на дому в течении одного месяца, регистрируется ежегодно в нашей стране.

Ишемические болезни сердца, цереброваскулярные расстройства, болезни кровообращения сердца относятся к основным причинам смертности в Казахстане за 2013 год (Institute for Health Metrics and Evaluation, 2015). Основными факторами, которые наиболее выражены в этот период, измеряемыми в значениях утраченных лет здоровой жизни по показателям DALY, являются нездоровое питание, повышенное систолическое артериальное давление, высокий уровень индекса массы тела, повышенное, а также выскоий индекс массы тела. Диабет же вырос в рейтинге причин стойких нарушений здоровья с 8-го места в 1990 г. до 4-го места в 2013 г.

По показателям смертности населения от болезни сердца по основным классам причин смерти на 100 000 человек населения в 2019 году составлял 58,25%. В 2020 же году данный показатель вырос на 9,32 % и уже составляет 67,57% [1].

По оценкам ВОЗ можно спрогнозировать развитие болезни пациента по сердечно-сосудистым событиям по таким факторам здоровья, как возраст, давление, показатели уровня сахара и холестерина и т.д. По результатам исследования STEPS данных населения Актюбинской области выявили, что 18% популяции имеют около трех, пяти факторов риска развития болезни [2].

1.5 Основные определения ЭКГ

Каждый удар сердца содержит 1 серию отклонений от базовой линии на ЭКГ, или волн, которые отражают временную эволюцию электрической активности в сердце. P-волна представляет собой небольшое отклонение, вызванное деполяризацией предсердий, Q, R и S-волны обычно рассматриваются как одиночное событие, известное как QRS-комплекс, который является частью ЭКГ с наибольшей амплитудой, вызванной вентральной деполяризацией. T-образная волна вызвана вентральной реполяризацией. Наконец, в некоторых случаях за 5-й волной T может последовать дополнительная U-волна [3].

На рисунке 1.1 показана схема общей автоматической системы классификации аритмий.

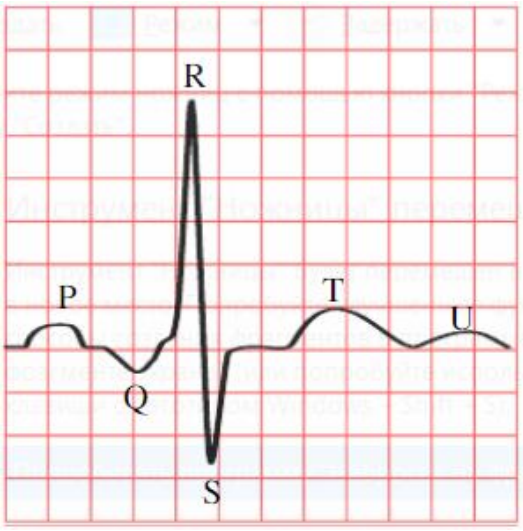

Рисунок-1.1 – Сигналы ЭКГ

Различные типы аритмий могут быть обнаружены с помощью анализа изменений, которые появились на этих 7 волнах.

Разработка полностью автоматической системы, способной классифицировать сердцебиения на ЭКГ, была предметом исследований, вызывавших большой интерес на протяжении последних десятилетий. Первоначально сигналы, которые были захвачены с помощью устройства, подвергаются предварительной обработке. Этот этап обычно включает удаление базовой линии и очистку от высокочастотного шума.

Нормальная ЭКГ содержит в себе волны, интервалы, сегменты и один комплекс. Волна может быть положительной, отрицательной в зависимости от базовой линии, указывающая на конкретное электрическое событие. Волны ЭКГ включают в себя P-волну, Q-волну, R-волну, S-волну, T-волну и U-волну.

Интервал в сигналах – это время между двумя конкретными событиями ЭКГ. Интервалы, обычно измеряемые на ЭКГ, включают интервал PR, интервал QRS (также называемый длительностью QRS), интервал QT и интервал RR.

Сегмент - расстояние между двумя определенными точками на ЭКГ, которые должны находиться на базовой амплитуде (не отрицательной или положительной). Сегменты на ЭКГ включают сегмент PR, сегмент ST и сегмент TP. Точка, называемая точкой J, же находится там, где заканчивается комплекс QRS и начинается сегмент ST.

Комплекс – это комбинация нескольких волн, сгруппированных вместе. Единственным основным комплексом на ЭКГ является комплекс QRS.

Волна P указывает на деполяризацию предсердий. Комплекс QRS состоит из Q-волны, R-волны и S-волны и представляет собой деполяризацию желудочков. Волна T появляется после комплекса QRS и указывает на реполяризацию желудочка [4].

1.6 Нормальные и аномальные сигналы ЭКГ

ЭКГ измеряет электрическую активность сердца. Данный тест может измерить многие аспекты. К примеру, сигналы могут описать то, как быстро бьется сердце, а также насколько хорошо его камеры проводят электрическую энергию.

Аномальные сигналы ЭКГ можно распознать по изменениям сердечного ритма. Такие сигналы в зависимости от периодичности появления могут считаться нормой и не влияют на здоровье человека. В других случаях аномальная ЭКГ может сигнализировать о надобности срочной медицинской помощи, такие как инфаркт миокарда (сердечный приступ), опасная аритмия.

Поскольку ЭКГ измеряет различные аспекты работы сердца, аномальные результаты могут указывать на несколько проблем. К ним относятся:

- дефекты или аномалии формы и размера сердца, при котором один или несколько аспектов стенок сердца больше, чем другой;
- электролитный дисбаланс, при котором из-за несбалансированного количества электролитов бывают ненормальные показатели ЭКГ;
- сердечный приступ или ишемия, при котором из-за нарушения кровотока плохо проводится электричество;
- нарушение сердечного ритма;
- побочные эффекты лекарств [5].

2 Технологический раздел

2.1 Python

Python – это язык программирования, включающий в себя высокоуровневые структуры данных, динамическую типизацию и связывание, открытые библиотеки и функции, используемые для задач машинного обучения, анализа данных, математических вычислений. Благодаря своей универсальности, Python актуален в своем современном виде для использования в задачах машинного обучения.

Данный язык программирования был выбран за счет его гибкости. Python – универсальный язык, поддерживающие множество библиотек, включая модели машинного обучения, библиотеки для удобной работы с данными и т.д. В последние годы Python считается самым востребованным и актуальным языком, что делает его более желательным для использования в новейших технологиях и ПО. А также тот факт, что он бесплатный и имеет открытую документацию с легко читаемым кодом, повышает шанс выбора данного языка в производственном продукте.

2.2 Jupyter

Jupyter – интерактивный веб-инструмент с открытым source-кодом в виде вычислительной записной книжки, который исследователи могут использовать для объединения кода программы, результатов вычислений, дополнительного описательного 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 Sklearn

Sklearn – это библиотека для языка Python, широко используемая в машинном обучении, включающая в себя функции и алгоритмы кластеризации, классификации, регрессии и других моделей.

Scikit-learn предоставляет множество неконтролируемых и контролируемых алгоритмов обучения. Он построен на других библиотеках, таких как numpy, pandas и matplotlib!

Функциональные возможности, предоставляемые scikit-learn, включают:

- регрессия, включая линейную и логистическую регрессию;
- классификация, включая K-Nearest Neighbors;
- кластеризация, включая K-Means и K-Means++
- выбор модели (model selection);
- предварительная обработка.

Данная библиотека была использована для таких моделей, как KMeans, AgglomerativeClustering, DBSCAN, SpectralClustering, GaussianMixture, а также для предварительной обработки данных и метрик с целью вывода эффективности алгоритмов.

3 Проектная часть

3.1 SpectralClustering

Спектральная кластеризация помогает нам преодолеть две основные проблемы при кластеризации. Одна из них связана с формой кластера, а другая - с определением центра тяжести кластера. Алгоритм K-средних обычно предполагает, что кластеры являются сферическими или круглыми, т.е. в пределах k-радиуса от центра тяжести кластера. Для определения центра тяжести кластера требуется много итераций.

В SpectralClustering точки, которые находятся далеко, но связаны, принадлежат к одному кластеру, а точки, которые менее удалены друг от друга, могут принадлежать разным кластерам. Это означает, что алгоритм может быть эффективным для данных различной формы и размера.

По сравнению с другими алгоритмами SpectralClustering является быстрым решением в вычислительном отношении для разреженных наборов данных из нескольких тысяч точек данных. Вычисление для больших наборов данных может быть дорогостоящим, поскольку необходимо вычислить собственные значения и свои векторы, в последующем же выполнить кластеризацию. Количество кластеров (k) должно быть зафиксировано перед началом процедуры.

Интуитивная цель кластеризации состоит в том, чтобы разделить точки данных на несколько групп таким образом, чтобы точки в одной группе были похожи, а точки в разных группах отличались друг от друга. Если не имеются больше информации, чем показатели схожести между точками, то можно использовать график подобия для общего представления данных:

$$G = (V, E)$$

где, каждая вершина vi представляет точку данных xi. Вершины соединяются, если сходство между соответствующими точками данных является положительным или превышает определенный порог, отрицательным в других же случаях. Проблема кластеризации теперь может быть переформулирована с использованием графа подобия: мы хотим найти разбиение графа таким образом, чтобы ребра между разными группами имели очень низкие веса, а ребра внутри группы имеют высокие веса, что означает, что точки внутри одного и того же кластера похожи друг на друга [6].

Выбор числа k кластеров является общей проблемой для всех алгоритмов кластеризации, и для решения этой проблемы было разработано множество более или менее успешных методов. В настройках кластеризации на основе модели существуют хорошо обоснованные критерии для выбора количества кластеров из данных. Эти критерии обычно основаны на логарифмической вероятности данных, которые затем могут быть обработаны байесовским или

частотным способом. В условиях, когда нет или мало предположений о создана базовая модель, для выбора количества кластеров можно использовать большое разнообразие различных индексов.

В нашем случае был использован метод нахождения количества кластеров методом «локтя» в графике Kmeans. Метод локтя – один из самых актульных методов, используемый для выбора правильного значения k с целью повышения производительности точности модели. Он работает эмпирическим способом, т.е. вычисляется средняя сумма квадратов расстояний между точками данных.

3.2 AgglomerativeClustering

Агломеративная кластеризация - это наиболее распространенный тип иерархической кластеризации, используемый для группировки объектов в кластеры на основе их сходства. Алгоритм начинается с обработки каждого объекта как одноэлементного кластера. На каждом шаге алгоритма два наиболее похожих кластера объединяются в новый более крупный кластер. Затем пары кластеров последовательно объединяются до тех пор, пока все кластеры не будут объединены в один большой кластер, содержащий все объекты. Тем самым алгоритм работает по принципу «снизу-вверх». Результатом является древовидное представление объектов, получившее название dendrogram.

Другим методом агломеративной кластеризации является кластеризация с разделением, которая работает по принципу «сверху вниз». Он начинается с корня, в котором все объекты включаются в один кластер. На каждом шаге итерации наиболее разнородный кластер делится на два. Процесс повторяется до тех пор, пока все объекты не окажутся в своем собственном кластере.

Методы измерения сходства между кластерами используются для решения того, какие объекты стоит объединять или разделять.

Существует множество методов вычисления информации о подобии, включая евклидовы и манхэттенские расстояния. Между каждой парой объектов вычисляются расстояния в наборе данных. Результаты вычислений преобразуют в вид матрицы расстояний и различий.

Функция linkage принимает параметры расстояний, затем группирует пары объектов в кластеры на основе их сходства. Новые образованные кластеры соединяются, создавая более крупные кластеры. В дальнейшем этот процесс повторяется до тех пор, пока не выстроится иерархическое дерево связей объектов исходного набора данных.

После вычисления иерархического дерево следует рассчитать одинаково ли отражаются исходные расстояния и расстояние в дереве.

Одним из известных таких способов измерения точности расстояний является корреляция между ними. Данное значение – показатель того, насколько хорошо кластерной иерархическое дерево отражает выбранные данные. При

верной кластеризации связывание объектов в дереве кластеров описывается высокой корреляцией между объектами в финальной матрице.

Чем ближе значение коэффициента корреляции к 1, тем выше точность модели для кластеризации для наших данных. Значения выше 0,75 считаются хорошими. Широко используемый метод «усредненной» привязки дает высокие значения этой статистики. Это может быть одной из причин его такой популярности [7].

3.3 DBSCAN

DBSCAN – это плотностной алгоритм пространственной кластеризации с присутствием шума. Как следует из названия, DBSCAN рассчитывает схожесть между точками данных в зависимости от их плотности.

DBSCAN группирует плотно сгруппированные точки в единый кластер. Данный алгоритм распознает каждый кластер в пространственных наборах данных. DBSCAN устойчив к выбросам, так называемым шумам. Отличительной стороной от модели Kmeans является то, что он не требует предварительного определения количества кластеров.

DBSCAN принимает два базовых параметров: epsilon и minPoints. Эпсилон - это радиус окружности в наборе данных. Он рассчитывается вокруг каждой точки данных с целью проверки плотности. minPoints - это минимальное количество точек данных, допустимое внутри круга радиусом в epsilon. Если эта точка даных находится внутри, то автоматически считается основной.

При работе модели рассматривается каждая точка данных, затем с остальными значениями по радиусу, определенный в epsilon, проводятся сравнения, т.е если для определенная точка находится в радиусе epsilon и количество точек в этом диапазоне не ниже minPoints, то эту точку можно отнести в один отдельный кластер.

При значении количества минимальных значений, меньшей minPoints, то точка будет относиться к пограничному значению. А если вокруг какой-либо точки данных в радиусе эпсилона нет других точек данных, то оно рассматривается как шум.

DBSCAN очень зависит от значений epsilon и minPoints. От выбора значений этих параметров зависит точность модели. Небольшое изменение в подготовке параметров может значительно сказаться на результатах, полученных алгоритмом.

Значение minPoints должно быть по крайней мере на единицу больше, чем количество измерений в наборе данных.

$$\text{minPoints} >= \text{количество измерений} + 1$$

Не имеет смысла для minPoints принимать значение единицы. В таком случае каждая точка будет считаться отдельным кластером, в следствии чего такой выбор приведет к неточным результатом. Следовательно, он должен быть не менее 3. Как правило, он в два раза больше размером. Но знание предметной области также определяет ее ценность.

Значение эпсилона может быть определено из графика K-расстояния. Если выбранное значение epsilon слишком мало, то будет создано большее количество кластеров, и больше точек данных будет приниматься за шум [8].

3.4 Kmeans

Кластеризация Kmeans - это неконтролируемый алгоритм обучения, который используется для решения задач кластеризации в машинном обучении. Он группирует немаркированный набор данных в разные кластеры. В параметре k определяется количество предопределенных кластеров, которые необходимо создать в процессе.

Kmeans является итеративным алгоритмом, который делит немаркированный набор данных на k различных кластеров таким образом, что каждый набор данных принадлежит только одной группе, обладающей схожими свойствами. Это позволяет нам группировать данные в разные группы и является удобным способом самостоятельно находить категории групп в немаркированном наборе данных без необходимости какого-либо обучения [9].

Алгоритм основывается на центроиде, где каждый кластер связан с центроидом. Основная цель этого алгоритма - минимизировать сумму расстояний между точкой данных и соответствующими им кластерами.

В качестве входных данных алгоритм Kmeans принимает немаркированный набор данных и делит их на k-количество кластеров, повторяя процесс до поры времени, когда не найдет наилучшие кластеры. Значение k должно определяться заранее.

Алгоритм кластеризации k-средних в основном выполняет две задачи:

- определяет наилучшее значение для K центральных точек или центроидов с помощью итеративного процесса;
- к каждой точке данных задает ближайший по расстоянию к ней k-центр, т.е. точки, находящиеся рядом с определенным k-центром, создают свой кластер.

3.5 GaussianMixture

Модель смеси Гаусса или - это не столько модель, сколько распределение вероятностей. Это универсально используемая модель для генеративного неконтролируемого обучения или кластеризации. Оно также называется

кластеризацией с максимизацией ожиданий (Expectation-Maximization Clustering) и основано на стратегии оптимизации. Модели гауссовой смеси используются для представления нормально распределенных подгруппы в общей популяции. Преимущество такой модели заключается в том, что она не требует изначального обозначения к какой подгруппе принадлежит определенная точка данных. Это позволяет модели автоматически изучать подгруппы. Данная модель относится к моделям обучения без учителя.

Модель гауссовой смеси может быть использована для кластеризации, которая представляет собой задачу группировки набора точек данных в кластеры, а также можно использовать для поиска кластеров в наборах данных, где кластеры могут быть нечетко определены. Кроме того, GMMS можно использовать для оценки вероятности того, что новая точка данных принадлежит каждому кластеру. Модели гауссовой смеси также относительно устойчивы к выбросам, что означает, что они все равно могут давать точные результаты, даже если есть некоторые точки данных, которые не вписываются точно ни в один из кластеров. Это делает GaussianMixture гибким и мощным инструментом для кластеризации данных.

Данная модель является вероятностной моделью, в которой для каждой группы предполагаются гауссовы распределения, и у них есть средние значения и ковариации, которые определяют их параметры. GaussianMixture состоит из двух частей – средних векторов (μ) и ковариационных матриц (Σ). Распределение Гаусса определяется как непрерывное распределение вероятностей, которое принимает колоколообразную кривую [10].

3.6 t-SNE

T-SNE представляет собой метод машинного обучения для уменьшения размерности. Главным преимуществом t-SNE является способность сохранять локальную структуру. Это означает, что точки, которые находятся близко друг к другу в многомерном наборе данных, будут иметь тенденцию находиться близко друг к другу на графике.

Алгоритм t-SNE моделирует распределение вероятностей соседей вокруг каждой точки. Здесь термин «соседи» относится к набору точек, которые находятся ближе всего к каждой точке. В исходном многомерном пространстве это моделируется как гауссово распределение. В двумерном же оно моделируется в виде t-распределения [11].

Основной параметр, управляющий подгонкой, называется perplexity. Низкий уровень perplexity означает, что мы заботимся о масштабе и фокусируемся на ближайших других точках. Высокая степень perplexity требует больше подхода «общей картины».

Поскольку распределения основаны на расстоянии, все данные должны быть числовыми.

Стоит отметить, что t-SNE работает только с теми данными, которые ему предоставлены. Он не создает модель, которую затем можно применить к новым данным.

3.7 PCA

Анализ основных компонентов (PCA) - это хорошо известный неконтролируемый метод уменьшения размерности, который создает соответствующие объекты с помощью линейных (linear PCA) или нелинейных (kernel PCA) комбинаций исходных переменных. В нашем случае был использован линейный PCA.

Построение релевантных признаков достигается путем линейного преобразования коррелированных переменных в меньшее число некоррелированных переменных. Это делается путем проецирования (точечного произведения) исходных данных с использованием собственных векторов ковариационной и/или корреляционной матрицы, известной как главные компоненты [12].

Таким образом, PCA представляет собой ортогональное преобразование данных в ряд некоррелированных данных, находящихся в уменьшенном пространстве PCA, таким образом, первый компонент объясняет наибольшую дисперсию в данных, а каждый последующий компонент объясняет меньше.

Вкратце, анализ PCA состоит из следующих этапов:

Во-первых, исходные входные переменные, хранящиеся в X, оцениваются по z, так что каждая исходная переменная (столбец X) имеет нулевое среднее значение и единичное стандартное отклонение.

Следующий шаг включает в себя построение и собственное разложение ковариационной матрицы Cx = (1/n) X'X (в случае данных с z-баллами ковариация равна корреляционной матрице, поскольку стандартное отклонение всех признаков равно 1).

Затем собственные значения сортируются в порядке убывания, представляющем уменьшение дисперсии в данных.

Наконец, проекция исходных нормализованных данных на уменьшенное пространство PCA получается путем умножения первоначально нормализованных данных на ведущие собственные векторы ковариационной матрицы, т.е. PCs.

Новое уменьшенное пространство PCA максимизирует дисперсию исходных данных. Чтобы визуализировать прогнозируемые данные, а также вклад исходных переменных в совместный график, мы можем использовать биплот.

Существует верхняя граница значимых компонентов, которые могут быть извлечены с помощью PCA. Это связано с рангом ковариационной/корреляционной матрицы [13].

4 Экспериментальный раздел

4.1 Датасет

Эта база данных содержит 48 записей ЭКГ продолжительностью около 30 минут, отобранных с частотой 360 Гц с 11-битным разрешением от 47 различных пациентов. Каждая запись содержит два сигнала, первый из которых для всех записей является модифицированным выводом II (MLII), в то время как второй соответствует V1, V2, V4 или V5, в зависимости от записи. Следовательно, только MLII1 предоставляется всеми записями [14]. База данных содержит больше 20000 ударов, все они были независимо отмечены двумя или более экспертами-кардиологами, и разногласия были устранены. В соответствии с рекомендациями Ассоциации по развитию медицинского приборостроения AAMI, типы сердцебиения MIT-BIH сгруппированы в пять классов сердцебиения. Как было рекомендовано AAMI, записи с темпом ударов 5 не рассматривались, а именно 102, 104, 107 и 217. База данных сильно несбалансированна, так как около 90% битов относятся к классу N, тогда как остальные 3%, 6% и 1% битов принадлежат классам SVEB, VEB и F. Было принято решение игнорировать класс Q AAMI, как и другие авторы [15, 16], поскольку его практически не существует. Только 815 образцов относятся к классу Q [17].

4.2 Метрика

Для сравнения точности подмножества по соответсвию набора меток в y_true в качестве метрики был использован accuracy_score из библиотеки sklearn. Точность модели рассчитывается как отношение числа правильных прогнозов к общему числу прогнозов. Она расчитывается следующим образом:

$$\text{Accuracy Score} = (TP+TN)/ (TP+FN+TN+FP)$$

где, TP – истино положительные, FP – ложноположительные, FN – ложноотрицательные [18].

Также используются метрики из библиотеки sklearn:

- silhouette_score, рассчитывающий по среднему расстоянию;
- fowlkes_mallows_score, рассчитывающий по среднему геометрическому значению между точностью и отзывом;
- calinski_harabasz_score, рассчитывающий в зависимости от дисперсии между классами;
- davies_bouldin_score, рассчитывающий по средней мере сходства каждого кластера.

4.3 Предобработка

Данные были заранее предобработаны методом MinMaxScaler. Все объекты из датасета находятся в диапазоне от нуля до единицы. MinMaxScaler сохраняет форму исходного распределения. Это не приводит к значительному изменению информации, встроенной в исходные данные. Данное форматирование важно для последующего использования датасета, так как нормализация данных уменьшает риск ошибочных результатов.

Для предобработки и сжатия датасета с целью устранить шумы из сигналов были рассмотрены методы PCA и t-SNE. Для наилучшего показания эффективности использования данных методов будут сравниваться показатели точности для алгоритма K-means для каждого из предобработанных данных.

PCA может использоваться, когда размеры входных объектов высоки. PCA также может использоваться для шумоподавления и сжатия данных. В нашем случае использование PCA обосновано с большим количеством переменных, а также с целью устранить шумы из сигналов ЭКГ [19]. Стоит отметить, что в сигналах, полученных с электрокардиограмм, имеются шумы (рисунок 4.1), избавиться от которых используется данный метод сжатия.

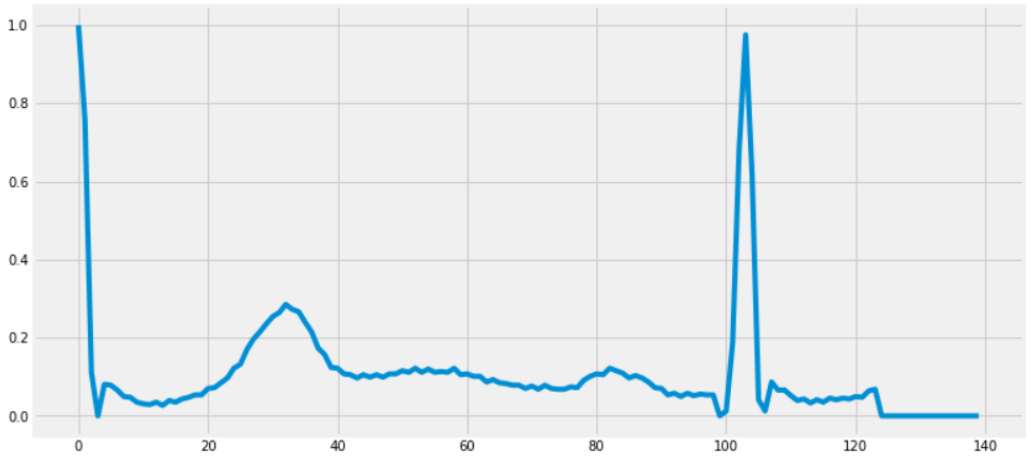

Рисунок-4.1 – Сигнал ЭКГ вместе с шумами

Некоторым алгоритмам, таким как KMeans, трудно точно построить кластеры, если набор данных содержит слишком много объектов. Точность может понизиться из-за высокой размерности данных.

Теория, лежащая в основе уменьшения объектов или размерности, заключается в преобразовании исходного набора объектов в меньшее количество искусственно созданных объектов, которые по-прежнему сохраняют большую часть информации, содержащейся в исходных объектах.

PCA сокращает исходный набор данных до определенного количества объектов, которые PCA называет основными компонентами. Мы должны выбрать количество основных компонентов, которые мы хотим видеть [20].

По итогам PCA были определены соответствующие компоненты (рисунок 4.2). В нашем случае на 29 основных компонента приходится 95% разницы.

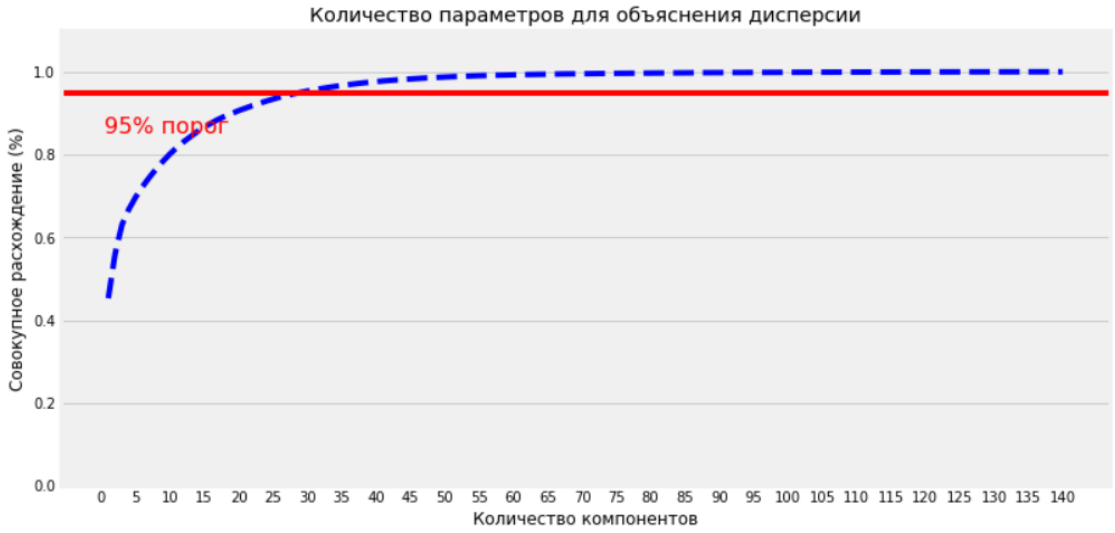

Рисунок-4.2 – Нахождение оптимального количества параметров для PCA

По итогам сравнений эффективности двух ранее упомянутых методов предобработки данных, точность которых определяется по точности алгоритма K-means, PCA показал лучший результат (рисунок 4.3).

	Метод	Результат
0	PCA	0.8299835556367623
1	t-SNE	0.8276082587246483

Рисунок-4.3 – Результаты сравнения методов PCA и t-SNE

t-SNE решает проблему, известную как проблема скученности, заключающаяся в том, что нескольком похожих точек в более высоком измерении коллапсируют друг на друга в более низких измерениях. К примеру, если нужен k-мерный вектор в качестве уменьшенного множества, а k не совсем мал, оптимальность полученного решения находится под вопросом. PCA, с другой стороны, всегда предлагает k наилучших линейных комбинаций с точки зрения объясненной дисперсии.

Данные были преобразованы по методу PCA (рисунок 4.4, рисунок 4.5).

	0	1	2	3	4	5	6	...	133	134	135	136	137	138	139
0	1.000000	0.758264	0.111570	0.000000	0.080579	0.078512	0.066116	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.908425	0.783883	0.531136	0.362637	0.366300	0.344322	0.333333	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.730088	0.212389	0.000000	0.119469	0.101770	0.101770	0.110619	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	1.000000	0.910417	0.681250	0.472917	0.229167	0.068750	0.000000	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.570470	0.399329	0.238255	0.147651	0.000000	0.003356	0.040268	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21887	0.928736	0.871264	0.804598	0.742529	0.650575	0.535632	0.394253	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21888	0.802691	0.692078	0.587444	0.446936	0.318386	0.189836	0.118087	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21889	1.000000	0.967359	0.620178	0.347181	0.139466	0.089021	0.103858	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21890	0.984127	0.567460	0.607143	0.583333	0.607143	0.575397	0.575397	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0
21891	0.973970	0.913232	0.865510	0.823210	0.746204	0.642082	0.547722	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0

21892 rows × 140 columns

Рисунок-4.4 – Данные до преобразования

	0	1	2	3	4	5	6	...	22	23	24	25	26	27	28
0	0.123450	0.492859	0.436323	0.346579	0.408850	0.426629	0.595796	...	0.153092	0.555325	0.345413	0.657696	0.531569	0.486930	0.403501
1	0.335506	0.273493	0.512871	0.648415	0.386118	0.305842	0.697339	...	0.442414	0.423977	0.442950	0.428495	0.598341	0.489307	0.291632
2	0.181623	0.492621	0.625088	0.483086	0.372613	0.495222	0.254881	...	0.475136	0.381946	0.482746	0.524975	0.536709	0.405331	0.381121
3	0.254310	0.429939	0.763196	0.636987	0.456076	0.800562	0.625392	...	0.455059	0.427006	0.479927	0.396870	0.540216	0.422871	0.472855
4	0.347872	0.502610	0.712452	0.417015	0.317140	0.454011	0.206740	...	0.265714	0.419534	0.531638	0.486344	0.491488	0.515127	0.354095
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
21887	0.570911	0.475195	0.680705	0.164575	0.517442	0.471293	0.786025	...	0.566801	0.452241	0.664417	0.485355	0.532276	0.593722	0.240772
21888	0.690485	0.441888	0.597850	0.372303	0.415619	0.416235	0.739964	...	0.473227	0.419032	0.357690	0.260703	0.415665	0.480711	0.424212
21889	0.145034	0.570377	0.583123	0.336979	0.529717	0.430824	0.464515	...	0.314576	0.545123	0.504248	0.491677	0.633386	0.364448	0.561457
21890	0.302732	0.542781	0.585831	0.191892	0.589261	0.519265	0.549950	...	0.600650	0.609891	0.590121	0.488559	0.363717	0.498313	0.377523
21891	0.478459	0.541781	0.651423	0.130550	0.613766	0.541557	0.690968	...	0.494544	0.395393	0.505292	0.581128	0.557215	0.520423	0.342099

21892 rows × 29 columns

Рисунок-4.5 – Данные после преобразования

4.4 Нахождение оптимальных параметров

Для дальнейшего разбора датасета было расчитано количество кластеров по «методу локтя» с помощью K-means.

Данный метод выполняет кластеризацию k-средних в наборе данных для диапазона значений k. Был взят диапазон от 0 до 6. Выполняется кластеризация K-средних с различными значениями k. Для каждого из значений k мы вычисляем средние расстояния до центра тяжести по всем точкам данных. На график наносятся эти точки и находят точку, где среднее расстояние от центра тяжести внезапно падает.

Как можно заметить на графике (рисунок 4.6) метод локтя определил 2 кластера в нашем датасете. Но так как данные распределены неравномерно, т.е.

количество нормальных сигналов превышает количество ненормальных, мы не можем точно полагаться на данное количество кластеров.

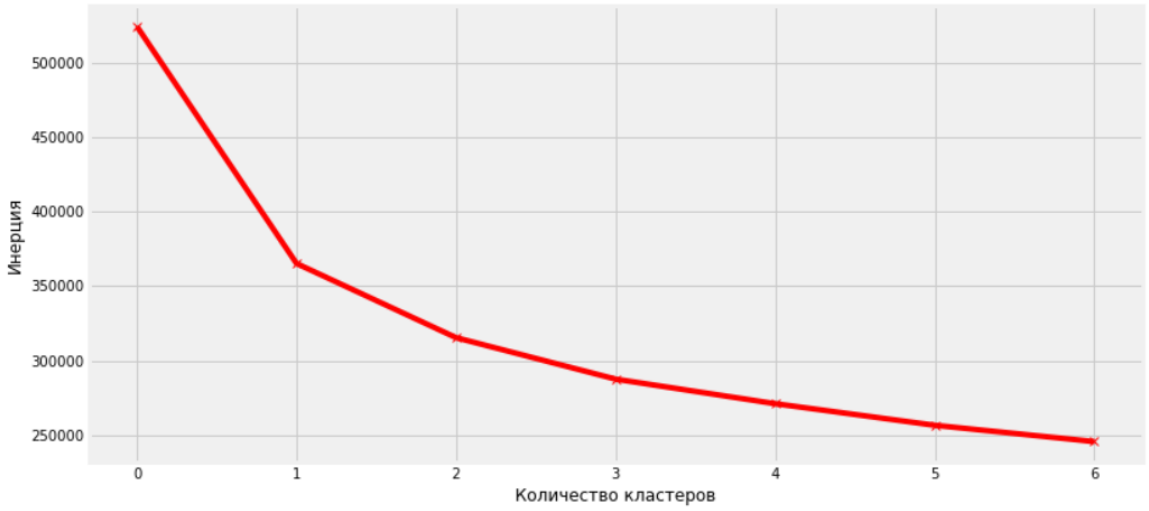

Рисунок-4.6 – График для определения количества кластеров

4.5 Использование алгоритма DBSCAN

Основным параметром, который принимает алгоритм DBSCAN является значение epsilon. Это максимальное расстояние между двумя образцами. Это наиболее важный параметр DBSCAN, который был выбран соответствующим образом для набора данных и функции расстояния.

DBSCAN несколько сложнее настроить по сравнению с алгоритмами параметрической кластеризации, такими как K-Means. Такие параметры, как epsilon для DBSCAN или для дерева набора уровней, менее интуитивно понятны для рассуждений по сравнению с параметром количества кластеров для K-средних, поэтому сложнее выбрать хорошие начальные значения параметров для этих алгоритмов.

Для определения epsilon был использован «метод локтя» с помощью метрического алгоритма NearestNeighbors, показанный на рисунке 4.7. Оптимальным значением найдено epsilon = 0.45.

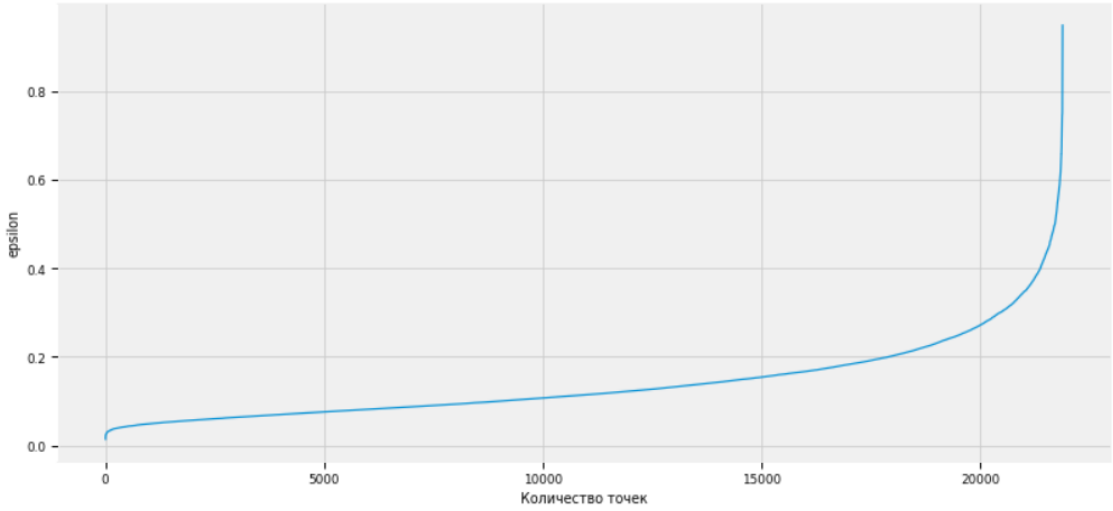

Рисунок-4.7 – График определения параметра epsilon для DBSCAN

А также данный алгоритм принимает параметр min_samples, который рассматривает количество образцов в соответствующем расстоянии, определенный для epsilon. Для этого параметра было взято значение пяти образцов.

Алгоритм DBSCAN показал следующие результаты (рисунок 4.8):

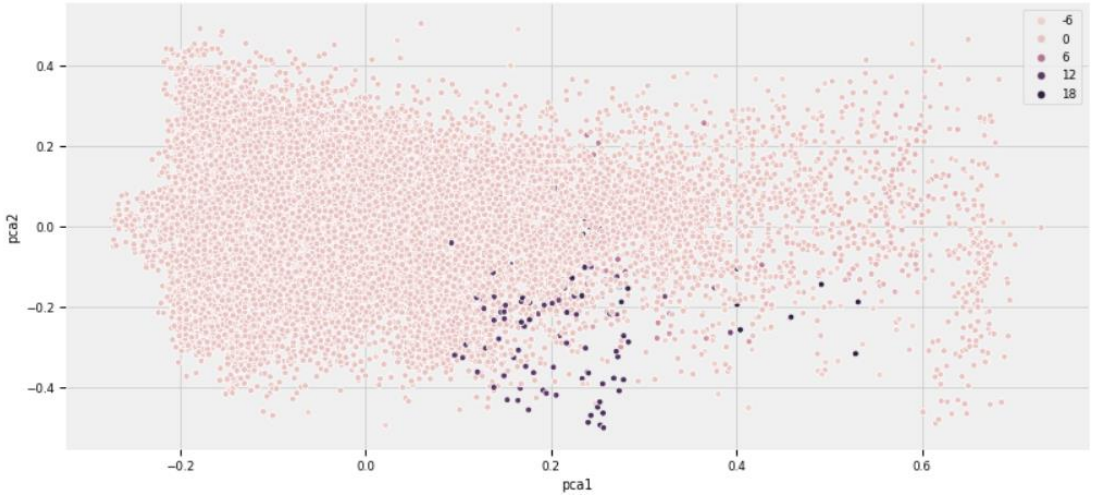

Рисунок-4.8 – Распределение кластеров DBSCAN

4.6 Использование алгоритма Kmeans

На рисунке 4.9 можно увидеть распределение кластеров Kmeans.

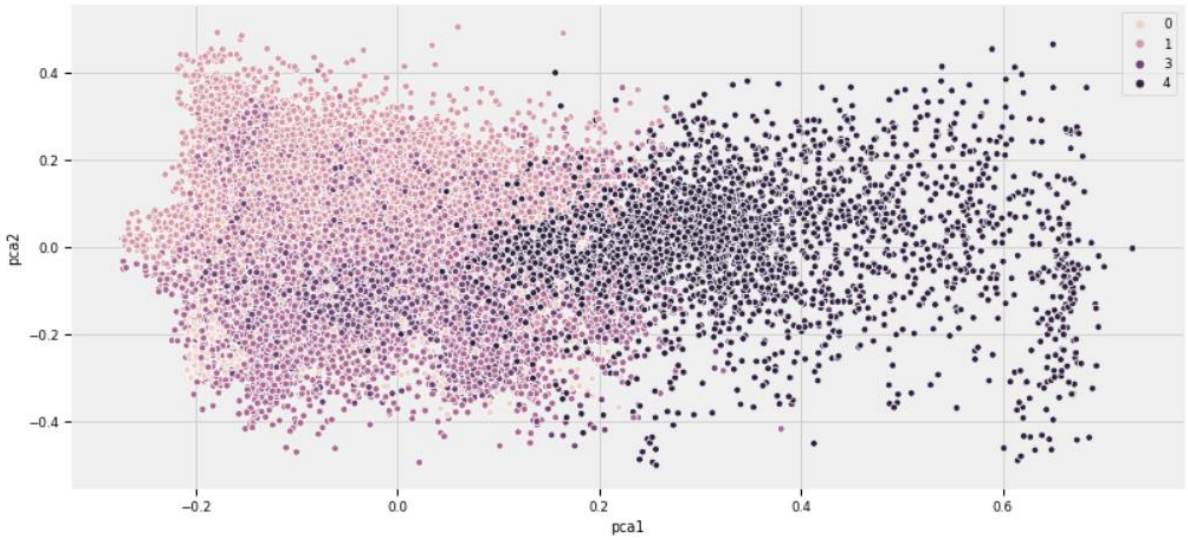

Рисунок-4.9 – Результаты алгоритма Kmeans

4.7 Использование иерархических алгоритмов

Иерархические алгоритмы по сравнению с вышеупомянутыми алгоритмами работают дольше за счет того, что он рассматривает каждый кластер отдельно, партиями деля на общие кластеры. Результат алгоритма показан на рисунке 4.10.

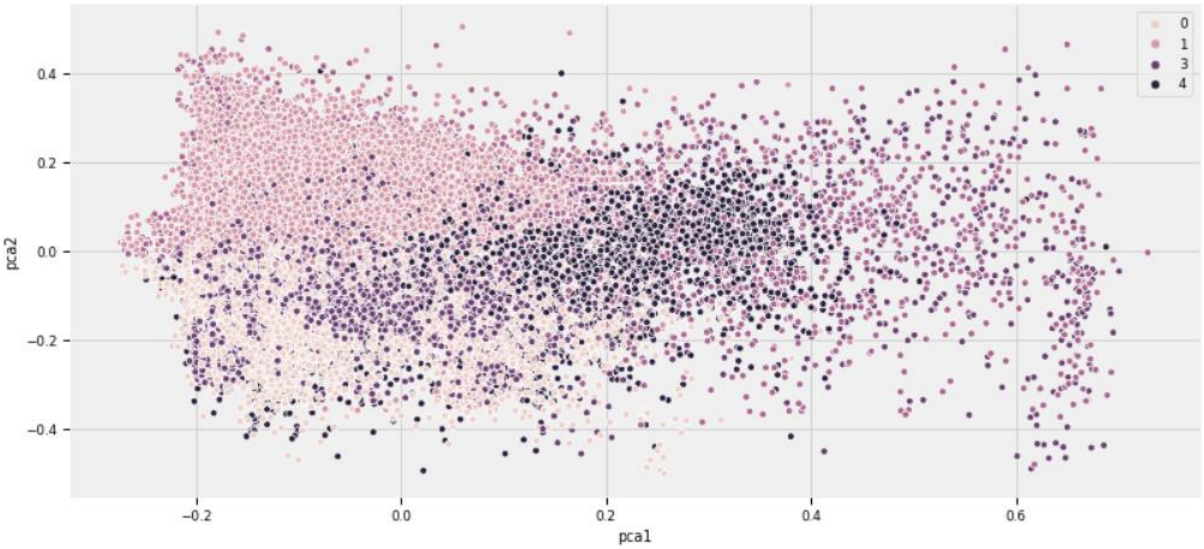

Рисунок-4.10 – Результат алгоритма AgglomerativeClustering

На рисунке 4.11 показан результат деления кластеров по методу SpectralClustering.

Рисунок-4.11 – График распределения кластеров по методу SpectralClustering

4.8 Результаты алгоритмов

По итогам сравнительного анализа (рисунок 4.12) применение иерархических алгоритмов показало наилучший показатель для медицинских данных, взятых с электрокардиограмм.

Иерархические модели кластеризации сработали лучше других алгоритмов. На практике же использование иерархических методов кластеризации может быть не обусловленным, так как данный тип алгоритмов работает намного дольше других рассмотренных алгоритмов.

В реальных условиях выбор метода стоит между Kmeans и GaussianMixture, результаты которых ничуть не хуже ранее упомянутых. При сравнении этих алгоритмов, если учитывать их производительность, то метод Kmeans уступает GaussianMixture. Kmeans - это простой и быстрый метод кластеризации, но он может не отражать гетерогенность, присущую облачным рабочим нагрузкам. Модели гауссовой смеси могут обнаруживать сложные закономерности и группировать их в связные, однородные компоненты, которые являются близкими представителями реальных закономерностей в наборе данных.

По итогам сравнительного анализа по метрикам кластеризации результаты показаны на рисунке 4.12.

	Алгоритм	silhouette_coefficient	fowlkes_mallows_score	calinski_harabasz_score	davies_bouldin_score
0	DBSCAN	0.322	0.834	4.659	2.244
1	Kmeans	0.052	0.464	828.397	3.446
2	GaussianMixture	0.037	0.598	654.873	5.658
3	SpectralClustering	0.002	0.384	305.066	6.208
4	AgglomerativeClustering	-0.014	0.597	364.256	4.998

Рисунок-4.12 – Сравнительный анализ алгоритмов

ЗАКЛЮЧЕНИЕ

В данной работе были рассмотрены методы обработки и кластеризации медицинских данных на примере сигналов ЭКГ. Были учтены шумы, встречающиеся в сигналах ЭКГ, а также датасет был нормализован с использованием библиотеки scikit-learn. Были рассмотрены основные понятия рассматриваемых методов кластеризации с учетом используемого датасета.

Сигналы ЭКГ были взяты с архива MIT-BIH. Данные сигналов пациентов были разделены на тренировочные и тестовые значения с целью последующего использования метрики точности для сравнения моделей ML. Были рассмотрены два метода предобработки данных, в последствии чего был определен наилучший метод PCA.

Были использованы следующие модели:

- Kmeans;
- GaussianMixture;
- AgglomerativeClustering;
- SpectralClustering;
- DBSCAN.

Результаты анализа показали, что иерархические модели показывает намного лучше результаты при кластеризации сигналов ЭКГ. Стоит учитывать, что данные алгоритмы работают дольше других рассмотренных алгоритмов. На практике выбор эффективного алгоритма встанет между Kmeans и GaussianMixture, так как они показали хорошие показатели и не менее уступают иерархическим моделям.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1 Статистические сборники «Здоровье населения Республики Казахстан и деятельность организаций здравоохранения» [Электронный ресурс] – Режим доступа: http://www.rcrz.kz/index.php/ru/statistika-zdravookhraneniya-2, свободный.

2 Профилактика неинфекционных заболеваний и борьба с ними в Казахстане [Электронный ресурс] – Режим доступа: https://www.euro.who.int/__data/assets/pdf_file/0004/409927/BizzCase-KAZ-Rus-web.pdf, свободный.

3 Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers [Электронный ресурс] – Режим доступа: https://www.researchgate.net/publication/327263145_Heartbeat_classification_fusing_temporal_and_morphological_information_of_ECGs_via_ensemble_of_classifiers, свободный.

4 Introduction to ECG [Электронный ресурс] – Режим доступа: https://www.healio.com/cardiology/learn-the-heart/ecg-review/ecg-interpretation-tutorial/introduction-to-the-ecg, свободный.

5 Rachel Nall. Abnormal EKG [Электронный ресурс] – Режим доступа: https://www.healthline.com/health/abnormal-ekg#results, свободный.

6 Keerthana V. What, why and how of Spectral Clustering [Электронный ресурс] – Режим доступа: https://www.analyticsvidhya.com/blog/2021/05/what-why-and-how-of-spectral-clustering, свободный.

7 Agglomerative Hierarchical Clustering [Электронный ресурс] – Режим доступа: https://www.datanovia.com/en/lessons/agglomerative-hierarchical-clustering, свободный.

8 Abhishek Sharma. How to master the popular DBSCAN clustering algorithm for machine learning [Электронный ресурс] – Режим доступа: https://www.analyticsvidhya.com/blog/2020/09/how-dbscan-clustering-works, свободный.

9 In Depth: k-Means Clustering [Электронный ресурс] – Режим доступа: https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html, свободный.

10 Gaussian Mixture Models: What are they & when to use [Электронный ресурс] – Режим доступа: https://vitalflux.com/gaussian-mixture-models-what-are-they-when-to-use, свободный.

11 TSNE scikit-learn 1.0.2 documentation [Электронный ресурс] – Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html, свободный.

12 PCA scikit-learn 1.0.2 documentation [Электронный ресурс] – Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html, свободный.

13 A Step-by-Step Explanation of Principal Component Analysis (PCA) [Электронный ресурс] – Режим доступа: https://builtin.com/data-science/step-step-explanation-principal-component-analysis, свободный.

14 Comparison of seven approaches for holter ECG clustering and classification? [Электронный ресурс] – Режим доступа: https://www.researchgate.net/publication/5843829_Comparison_of_seven_approaches_for_holter_ECG_clustering_and_classification, свободный.

15 Mar T., Zaunseder S., Matrnez J.P., Llamedo M. Optimization of ECG [Электронный ресурс] – Режим доступа: https://doi.org/10.1109/51.932724, свободный.

16 Zhang Z., Dong J., Choi K.S. Heartbeat classification using disease-specific [Электронный ресурс] – Режим доступа: https://doi.org/10.1016/j.compbiomed.2013.11.019, свободный.

17 Mondejar-Guerra V., Novo J., Penedo M.G., Ortega M. Heartbeat classification fusing temporal and morphological information of ECG via ensemble classifiers [Электронный ресурс] – Режим доступа: https://www.researchgate.net/publication/327263145_Heartbeat_classification_fusing_temporal_and_morphological_information_of_ECGs_via_ensemble_of_classifiers, свободный.

18 Kumar B. Scikit learn accuracy_score [Электронный ресурс] – Режим доступа: https://pythonguides.com/scikit-learn-accuracy-score, свободный.

19 Tan A. Principal components of electrocardiograms [Электронный ресурс] – Режим доступа: https://medium.com/@andrewtan_36013/principal-components-of-electrocardiograms-14874b3a96b1, свободный.

20 Mikulski B. PCA—how to choose the number of components [Электронный ресурс] – Режим доступа: https://www.mikulskibartosz.name/pca-how-to-choose-the-number-of-components, свободный.

Приложение А
(обязательное)

Техническое задание

А.1.1 Техническое задание на исследовательский анализ применения алгоритмов кластеризации медицинских данных

Настоящее техническое задание распространяется на исследовательский анализ применения алгоритмов кластеризации медицинских данных, взятых с электрокардиограмм.

А.1.2 Основание для анализа

Исследовательский анализ проводится на основании устного распоряжения дипломного руководителя.

А.1.3 Назначение

Исследовательский анализ проводится с целью определения эффективного алгоритма класстеризации сигналов ЭКГ пациентов.

А.1.4 Требования к функциональным характеристикам

Приложение должно обеспечить возможность выполнения следующих функций:

- определение нормированного или ненормированного сигнала по показаниям ЭКГ
- сравнение и анализ алгоритмов класстеризации для нахождения самого эффективного из них

А.1.5 Требования к надежности

Продолжение к приложению А

Обеспечить высокую точность методов класстеризации, а также эффективности используемых метрик на анализируемом датасете.

Приложение Б
(обязательное)

Текст программы

```
import pandas as pd
import numpy as np
from sklearn import metrics
from sklearn.decomposition import PCA
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from scipy.stats import mode
import seaborn as sns
from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN,
SpectralClustering
from sklearn.mixture import GaussianMixture
import scipy.cluster.hierarchy as shc

plt.style.use('fivethirtyeight')  # стиль для графиков
%matplotlib inline

X = pd.read_csv('mitbih.csv', header=None)
y = X.iloc[:,187]
y = y.astype(int)
np.unique(y)
X = X.iloc[:,:140]

pca = PCA().fit(data_rescaled)
plt.rcParams["figure.figsize"] = (12,6)
fig, ax = plt.subplots()
xi = np.arange(1, 141, step=1)
y = np.cumsum(pca.explained_variance_ratio_)
plt.ylim(0.0,1.1)
plt.plot(xi, y, marker='o', linestyle='--', color='b')
plt.xlabel('Количество компонентов')
plt.xticks(np.arange(0, 141, step=5))
plt.ylabel('Совокупное расхождение (%)')
plt.title('Количество параметров для объяснения дисперсии')
plt.axhline(y=0.95, color='r', linestyle='-')
plt.text(0.5, 0.85, '95% порог', color = 'red', fontsize=18)
ax.grid(axis='x')
plt.show()
```

Продолжение к приложению Б

```
# 95% of variance
from sklearn.decomposition import PCA
pca = PCA(n_components = 0.95)
pca.fit(data_rescaled)
reduced = pca.transform(data_rescaled)

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
reduced_X = scaler.fit_transform(reduced)

from sklearn.manifold import TSNE

tsne = TSNE(n_components=2)
tsne_proj = tsne.fit_transform(reduced)
# Compute the clusters
kmeans = KMeans(n_clusters=5, random_state=0)
clusters = kmeans.fit_predict(tsne_proj)

# Permute the labels
labels = np.zeros_like(clusters)
for i in range(10):
    mask = (clusters == i)
    labels[mask] = mode(y[mask])[0]

# Compute the accuracy
metrics.accuracy_score(y, labels)

from sklearn.cluster import KMeans
n_clusters=8
cost=[]
for i in range(1,n_clusters):
        kmean= KMeans(n_clusters = i)
        kmean.fit_predict(X_train)
        cost.append(kmean.inertia_)
plt.plot(cost, 'bx-', color='red')
plt.xlabel('Количество кластеров')
plt.ylabel('Инерция')
plt.show()

from sklearn.neighbors import NearestNeighbors
from sklearn.cluster import DBSCAN
from matplotlib import pyplot as plt
```

Продолжение к приложению Б

```
neigh = NearestNeighbors(n_neighbors=2)
nbrs = neigh.fit(reduced)
distances, indices = nbrs.kneighbors(reduced)
distances = np.sort(distances, axis=0)
distances = distances[:,1]
plt.plot(distances)
plt.xlabel('Количество точек')
plt.ylabel('epsilon')
plt.show()

db = DBSCAN(eps=0.45, min_samples=5)
score = metrics.accuracy_score(y,db.fit(reduced_X).labels_)

dbm = db.fit(reduced_X)
labels = dbm.labels_
sns.scatterplot(results["pca1"], results["pca2"], hue=labels, data=results);
plt.show()

db = DBSCAN(eps=0.8, min_samples=5).fit(reduced_X)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_

# Number of clusters in labels, ignoring noise if present.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)

print("Silhouette Coefficient: %0.3f"
      % metrics.silhouette_score(reduced_X, labels))
print("fowlkes_mallows_score: %0.3f"
      % metrics.fowlkes_mallows_score(y, labels))
print("calinski_harabasz_score: %0.3f"
      % metrics.calinski_harabasz_score(reduced_X, labels))
print("davies_bouldin_score: %0.3f"
      % metrics.davies_bouldin_score(reduced_X, labels))

kmeans = KMeans(n_clusters=5, random_state=0)

clusters = kmeans.fit_predict(reduced_X)

# Permute the labels
labels = np.zeros_like(clusters)
```

```
for i in range(10):
   mask = (clusters == i)
   labels[mask] = mode(y_test[mask])[0]

# Compute the accuracy
metrics.accuracy_score(y_test, labels)

gm = GaussianMixture(n_components=2, random_state=0)
clusters = test_m.fit_predict(reduced_X)
# Permute the labels
labels = np.zeros_like(clusters)
for i in range(10):
   mask = (clusters == i)
   labels[mask] = mode(y[mask])[0]

from sklearn.cluster import SpectralClustering

clusters1 = SpectralClustering(n_clusters=5,
      random_state=0).fit_predict(np.split(reduced_X,2)[0])
clusters2 = SpectralClustering(n_clusters=5,
      random_state=0).fit_predict(np.split(reduced_X,2)[1])
clusters = np.concatenate((clusters1, clusters2))
# Permute the labels
labels = np.zeros_like(clusters)
for i in range(10):
   mask = (clusters == i)
   labels[mask] = mode(y[mask])[0]

clusters1 =
AgglomerativeClustering(n_clusters=5).fit_predict(np.split(reduced_X,2)[0])
clusters2 =
AgglomerativeClustering(n_clusters=5).fit_predict(np.split(reduced_X,2)[1])
clusters = np.concatenate((clusters1, clusters2))
# Permute the labels
labels = np.zeros_like(clusters)
for i in range(10):
   mask = (clusters == i)
   labels[mask] = mode(y[mask])[0]
```