Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Работа допущена к защите
Директор ВШТМиМФ,
Д.ф.-м.н., чл.-корр. РАН
_____ А.М. Кривцов
«___» _____2024 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Прогнозирование вида движения человека с использованием алгоритмов машинного обучения на основе данных с датчиков смартфона

по направлению подготовки
01.03.03 «Механика и математическое моделирование»
Направленность
01.03.03_02 «Биомеханика и медицинская инженерия»

Выполнил
Студент гр. 5030103/00201 А.А. Балашов

Руководитель
Старший преподаватель ВШТМиМФ,
к.ф.-м.н. С.А. Щербинин

Консультант
ассистент ВШТМиМФ А.Д. Ершов

Санкт-Петербург
2024

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Физико-механический институт

Высшая школа теоретической механики и математической физики

УТВЕРЖДАЮ

Директор ВШТМиМФ

А. М. Кривцов

«__»_____20__г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Балашову Андрею Алексеевичу, гр. 5030103/00201

1. Тема работы: Прогнозирование вида движения человека с использованием алгоритмов машинного обучения на основе данных с датчиков смартфона.

2. Срок сдачи студентом законченной работы: 30.05.2024

3. Исходные данные по работе: методы машинного обучения, методы обработки и анализа данных, алгоритмы визуализации данных.

4. Содержание работы (перечень подлежащих разработке вопросов): построение моделей машинного обучения, классифицирующих движение человека; анализ полученных моделей; сравнение между собой различных моделей; определение одной или нескольких оптимальных моделей на основе выбранных параметров.

5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено

6. Консультанты по работе: Ершов А. Д., ассистент ВШТМиМФ.

7. Дата выдачи задания 26.02.2024

Руководитель ВКР _____ Щербинин С.А., старший преподаватель ВШТМиМФ, к.ф.-м.н.

Задание принял к исполнению 26.02.2024

Студент _____ Балашов А.А.

РЕФЕРАТ

На 50 с., 28 рисунков, 7 таблиц.

ОПРЕДЕЛЕНИЕ ВИДА ДВИЖЕНИЯ ЧЕЛОВЕКА, МАШИННОЕ ОБУЧЕНИЕ, ЗАДАЧА КЛАССИФИКАЦИИ, МЕТОД К-БЛИЖАЙШИХ СОСЕДЕЙ, НАИВНЫЙ БАЙЕСОВСКИЙ КЛАССИФИКАТОР, МЕТОД СВЕРХСЛУЧАЙНЫХ ДЕРЕВЬЕВ

В рамках данной работы была решена задача классификации движения человека методами машинного обучения. Работа состоит из 3 частей: обработка и анализ данных, обзор, применение и анализ качества алгоритмов классификации и программная реализация визуализации анализа активности человека.

THE ABSTRACT

50 pages, 28 pictures, 7 tables.

HUMAN MOTION IDENTIFICATION, MACHINE LEARNING, CLASSIFICATION PROBLEM, K-NEAREST NEIGHBORS METHOD, NAIVE BAYES CLASSIFIER, EXTRA TREES CLASSIFIER

In this paper, the problem of classifying human motion was solved using machine learning method. The work consists of 3 parts: data processing and analysis, review, application and accuracy analysis of classification algorithms and software implementation of visualization of human activity analysis.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В реалиях сегодняшнего дня смартфоны стали незаменимыми помощниками человека, они становятся все более функциональными с каждым днем, помогая людям в их повседневной деятельности. Сравнительно недавно появилась опция, популярная среди спортсменов, которая позволяет пользователям отслеживать ежедневные шаги.

В более продвинутых версиях спортивных приложений появилась функция определения ходьбы и бега. Возможность различать виды движения достигается с помощью датчиков, считывающих данные о перемещении смартфона в пространстве, и последующей обработки данных, полученных таким образом.

Обоснованием использования в данной задаче методов машинного обучения являются его преимущества перед традиционными способами нахождения паттернов и аппроксимации данных. Список таких преимуществ следующий:

1. Машинное обучение выявляет глубокие, не наблюдаемые человеком, закономерности.
2. Машинное обучение позволяет обрабатывать большие объемы данных за короткий промежуток времени.
3. Машинное обучение более эффективно расходует ресурсы, такие как время и мощности вычислительной техники.
4. Машинное обучение более доступно для понимания, чем классические способы выявления глубоких закономерностей.

Актуальность работы заключается в востребованности пользователями получения точных данных о собственном передвижении, а также в современности используемого метода и наличии у него ряда преимуществ, описанных выше.

Целью данной работы является написание программы, которая считывает данные, полученные с датчиков смартфона, обрабатывает их и строит предиктивную модель, определяющую вид движения человека, предоставляет визуальный анализ активности человека.

Исходя из поставленной цели, решаются такие задачи как:

1. Обработка данных, подготовка их к использованию программой;
2. Реализация алгоритма машинного обучения, определяющего значение переменной "активность";
3. Исследование влияния различных параметров на качество и время выполнения программы, настройка параметров для оптимизации показателей;
4. Представление результатов в виде графического анализа активности человека.

В дальнейшем полученные наработки можно будет применять для создания программ, устанавливаемых на смартфоны, умные часы и другие подобные аксессуары, которые помогут спортсменам и любителям следить за количеством шагов и дистанцией бега.

ГЛАВА 1. ОБЩИЕ СВЕДЕНИЯ

1.1. Виды движения человека

Для того, чтобы понимать объект исследования, сначала необходимо обратиться к вопросу о том, что такое бег, а что – ходьба, изучить из чего состоят данные локомоции. На рисунке 1 представлены простейшие хронограммы ходьбы и бега. Из них следует, что, по мере увеличения скорости передвижения, происходит следующее:

- при ходьбе сокращается период двойной опоры (когда обе ноги находятся на земле) вплоть до почти полного его исчезновения при спортивной ходьбе;
- при беге увеличивается отношение длительности периода полета (когда обе ноги не касаются опоры) к длительности периода опоры [2].

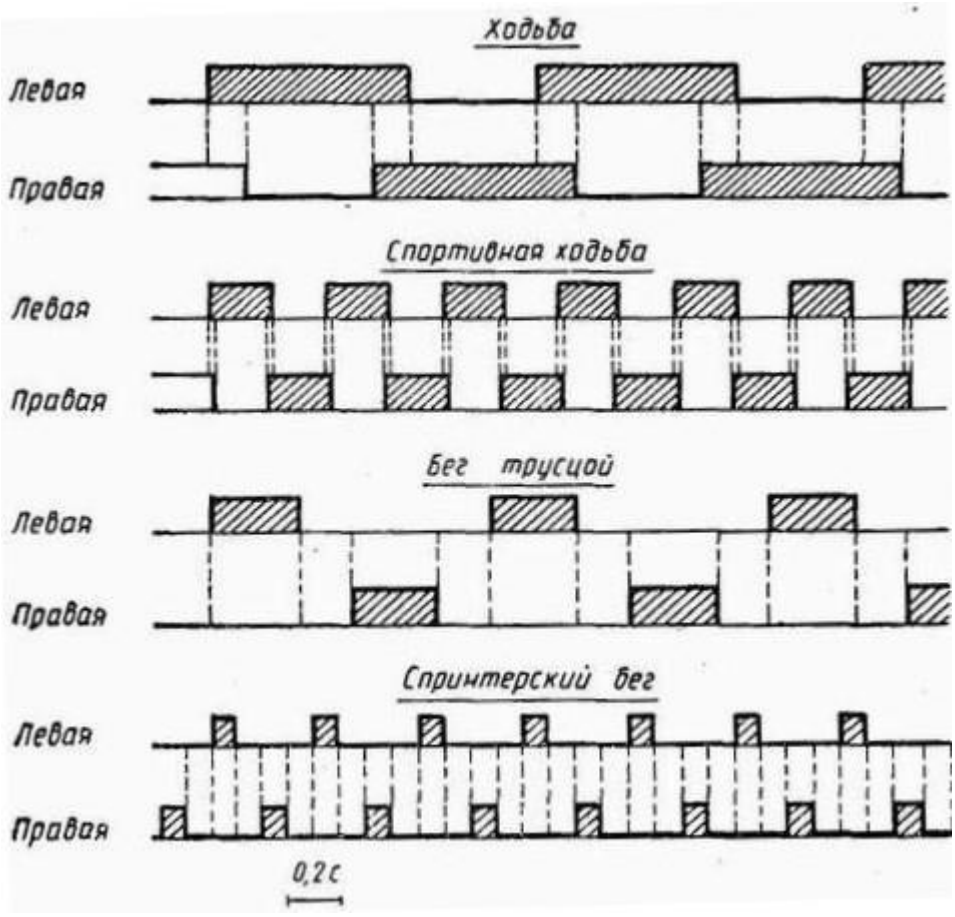

Рисунок 1 - Простейшие хронограммы обычной ходьбы, спортивной ходьбы, бега трусцой и спринтерского бега периоды опоры заштрихованы; вверху — левая нога, внизу — правая

Таким образом, ходьбой считается способ передвижения человека, при котором в каждом цикле есть периоды касания земли обеими ногами, бегом – когда в каждом цикле присутствуют фазы “полета”, то есть, когда обе ноги не касаются земли. Также из хронограмм, представленных на рисунке 1 следует, что переходом границы от ходьбы к бегу можно считать тот момент времени, когда человек перестает в процессе своего движения касаться земли двумя ногами, и начинает появляться период “полета”.

1.2. Машинное обучение

Машинное обучение – это средство анализа данных, позволяющее автоматически строить модель, считывающую и обрабатывающую данные и предоставляющую, на основе результатов анализа, необходимые выводы. Основой машинного обучения является получение машиной опыта, через который она улучшает свои навыки [14].

На рисунке 2 представлено дерево видов и подвидов машинного обучения, а также сферы их применения:

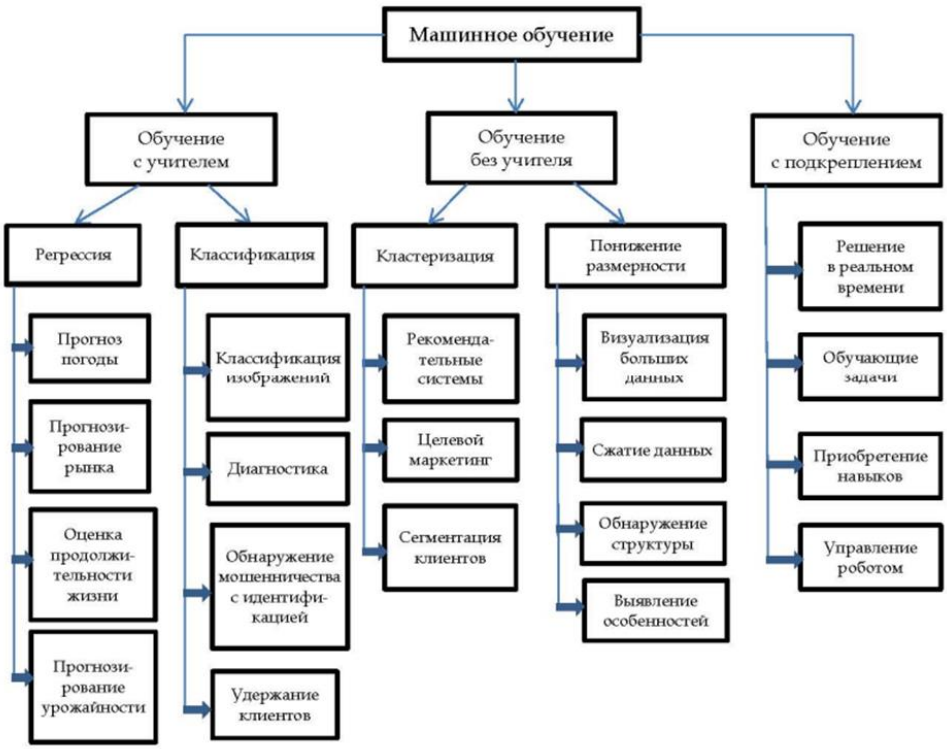

Рисунок 2 – Виды и подвиды машинного обучения с примерами их применения в различных сферах деятельности человека

Согласно схеме, машинное обучение изначально делится на 3 вида [10]:

1. Обучение без учителя — это один из разделов машинного обучения, который изучает широкий класс задач по обработке данных, в которых известны только описания множества объектов (обучающей выборки), и требуется обнаружить внутренние взаимосвязи, зависимости, закономерности, существующие между объектами [8].
2. Обучение с подкреплением – это метод заставить искусственный интеллект действовать так, чтобы максимизировать его вознаграждение, подобно дрессировке собаки, когда инструкции невозможно сообщить напрямую, но можно посредством системы поощрений и наказаний обучить собаку понимать, что от нее хочет владелец. Это старейший вид машинного обучения, использование которого берет свое начало в 1950-х годах прошлого века с разработки программ, предназначенных для игры в шашки. До сих пор обучение с подкреплением активно применяется в игровой сфере [16].
3. Обучение с учителем. Суть метода в том, что для обучения нейросеть получает специальный набор данных (датасет), в котором заранее отмечено, что эти данные означают. То есть нейросеть получает в том числе и информацию о том, какой ответ она должна давать. Она обрабатывает набор данных, находит корреляции между переменными и ответами и учится давать правильный ответ на основе построенных взаимосвязей [3].

В рассматриваемой задаче будет использоваться именно обучение с учителем, поэтому необходимо подробнее изучить данный метод. Обучающие данные включают значения целевой переменной (target variable), называемые метками (label). К примеру, на рисунке 3 в первом случае меткой будет “доброкачественная” или “злокачественная” в последнем столбце, целевая переменная – “доброкачественная или злокачественная”, во втором случае данные неразмеченные.

Размеченные данные

ID	Кластер	Однородность размера клеток	Адгезия	Размер эпителиальной клетки	Голые ядра	Обычные ядрышки	Митозы	Доброкачественная или злокачественная
102039	1	7	5	8	7	10	3	Доброкачественная
102040	9	1	6	10	3	4	5	Доброкачественная
102041	4	3	6	11	2	1	9	Злокачественная
102042	2	3	7	6	3	4	3	Доброкачественная
102043	2	5	9	8	8	5	4	Доброкачественная

Неразмеченные данные

ID	Возраст	Образование	Доход, руб.	Задолженностей, руб.	Индекс места жительства	Отношение доход к долгу
323	85	Высшее	234134	101029	192281	2.32
324	47	Высшее	154300	45399	566345	3.40
325	41	Среднее	103000	56363	760334	1.83
326	90	Высшее	46000	11370	177756	4.05
327	76	Среднее	35400	2365	192281	14.97

Рисунок 3 – Размеченные и неразмеченные данные, образец

Для того, чтобы обучение с учителем работало, данные должны быть размечены, должна быть выделена целевая переменная, значение которой будет предсказываться, и метки – значения переменной. Пример размеченных и неразмеченных данных изображен на рисунке 3.

К важнейшим алгоритмам обучения с учителем относятся:

- линейная регрессия;
- логистическая регрессия;
- метод k-ближайших соседей;
- метод опорных векторов;
- деревья принятия решений и случайные леса;
- нейронные сети.

В данной работе решается задача классификации. Классификация – самая распространенная задача машинного обучения. Цель этого метода –

классифицировать объекты по заранее известному признаку [6]. Также обучение с учителем может быть использовано для решения задач регрессии, чтобы прогнозировать целевое числовое значение переменной, располагая набором характеристик или признаков. Основное отличие задачи регрессии и классификации – в первом случае целевая переменная выражается и рассчитывается как непрерывный спектр, во втором же – как дискретный.

ГЛАВА 2. ВХОДНЫЕ ДАННЫЕ

В настоящее время набор данных содержит 88588 выборок данных, собранных с акселерометра и гироскопа iPhone 5S с интервалом в 10 секунд и частотой ~ 5,4 $с^{-1}$. Эти данные представлены в виде следующих временных рядов (каждый столбец содержит данные датчика для одной из осей датчика):

- показания акселерометра по оси x;
- показания акселерометра по оси y;
- показания акселерометра по оси z;
- показания гироскопа по оси x;
- показания гироскопа по оси y;
- показания гироскопа по оси z.

Также определен тип активности, представленный столбцом "активность", который служит целевой переменной и отражает следующие виды деятельности:

- "0": ходьба;
- "1": бег.

Кроме того, набор данных содержит столбец "запястье", который представляет запястье, на котором было установлено устройство для взятия пробы:

- "0": левое запястье;
- "1": правое запястье.

Набор данных содержит столбцы "дата", "время" и "имя пользователя", которые предоставляют информацию о точной дате, времени и пользователе, который проводил эти измерения.

ГЛАВА 3. ПОСТАНОВКА ЗАДАЧИ

Решается задача бинарной классификации. Пусть задано множество объектов X и множество D классов этих объектов. В дальнейшем $X \subseteq R^n$, где R^n – множество всех действительных чисел, а D – конечное множество с небольшим числом элементов. Размерность n евклидового пространства R^n велика по сравнению с числом классов.

Далее элементы R^n будут называться векторами (точками) и обозначаться подчеркнутыми сверху буквами: $\bar{x}, \bar{y}, \ldots \in R^n$; в координатах – $\bar{x} = (x_1, \ldots, x_n)$. Будут рассматриваться операции сложения векторов

$$\bar{x} + \bar{y} = \begin{pmatrix} x_1+y_1 \\ x_2+y_2 \\ \cdots \\ x_n+y_n \end{pmatrix}, \text{ где } \bar{x} = (x_1, \ldots, x_n)', \bar{y} = (y_1, \ldots, y_n)' - \text{транспонированные}$$

и умножения на вещественное число

$$\alpha\bar{x} = \begin{pmatrix} \alpha x_1 \\ \alpha x_2 \\ \cdots \\ \alpha x_n \end{pmatrix}.$$

На векторах из R^n также определено их скалярное произведение $(\bar{x} \cdot \bar{y}) = x_1 \cdot y_1 + x_2 \cdot y_2 + \ldots + x_n \cdot y_n$. Норма (длина) вектора $\bar{x}$ определяется как

$$\|\bar{x}\| = \sqrt{(\bar{x} \cdot \bar{x})} = \sqrt{\sum_{i=1}^{n} x_i^2}$$

При решении задачи классификации исходят из обучающей выборки $S = \left((\bar{x_1}, y_1), \ldots, (\bar{x_n}, y_n)\right)$, где $\bar{x_i} \in X$ – вектор евклидового пространства R^n большой размерности n, y_i – это элемент конечного множества D с небольшим числом элементов (метка класса), например, $y_i \in \{0, 1\}$. Элементы $y_i \in D$ определяют классы объектов $\bar{x_i}$.

Предполагается, что выборка $S = \left((\bar{x}_1, y_1), \ldots, (\bar{x}_n, y_n) \right)$ генерируется (порождается) некоторым источником. Основное предположение об источнике, порождающем выборку S, заключается в том, что на парах $(\bar{x}, y)$, т.е. на пространстве $X \times D$ задано распределение вероятностей P, а пары $(\bar{x}_i, y_i)$, образующие выборку S, одинаково и независимо распределены.

Соответственно на множестве $(X \times D)^l$ задано распределение вероятностей $P^l = P \times P \times \cdots \times P$.

Правило или функция классификации – это функция типа $h: X \to D$, которая разбивает элементы $\bar{x}_i \in X$ на несколько классов. Также функция h называется классификатором.

В дальнейшем будет рассматриваться случай бинарной классификации $D = \{0, 1\}$, а функция $h: X \to D$ будет называться индикаторной. В этом случае вся выборка S разбивается на две подвыборки: $S^+ = ((\bar{x}_i, y_i),\ y_i = 1)$ – положительные примеры (или первый класс) и $S^- = ((\bar{x}_i, y_i),\ y_i = 0)$ – отрицательные примеры (или второй класс).

Качество произвольной функции классификации h будет оцениваться по ошибке классификации, которая определяется как вероятность неправильной классификации

$$\mathrm{err}_P(h) = P\{h(\bar{X}) \neq Y\} = P\{h(\bar{x}, y): h(\bar{x}) \neq y\}.$$

Здесь h(X) – функция от случайной величины X, также является случайной величиной, поэтому можно рассматривать вероятность события $\{h(\bar{X}) \neq Y\}$.

Основная цель при решении задачи классификации – для заданного класса функций классификации H построить оптимальный классификатор, т.е. такую функцию классификации $h \in H$, при которой ошибка классификации $\mathrm{err}_P(h)$ является наименьшей в классе H [1].

ГЛАВА 4. ХОД РАБОТЫ

4.1. Обработка и преобразование данных

Дальнейшая работа выполняется в интегрированной среде разработки на языке Python. Для обработки данных и работы с ними, их необходимо считать. Для этого используется библиотека Pandas, программная библиотека, предназначенная для обработки и анализа данных.

Первым делом таблица подготавливается к дальнейшей работе. Для этого удаляются столбцы даты, времени, запястья и имени пользователя.

Дата и время могут служить для дальнейшего представления данных в виде подготовленного анализа активности человека, но на этапе построения предиктивной модели в них нет необходимости, так как ошибочно будет искать взаимосвязь вида движения человека и даты и времени сбора данных.

В перспективе, для определения закономерностей в случае разных полов и возрастов испытуемых может понадобиться информация о пользователе, так как она поможет сепарировать разные модели и не допустить, к примеру, распространения паттернов, определенных для взрослого мужчины, на ребенка противоположного пола. Потому как, несмотря на общую схожесть состава локомоций у людей разных полов и возрастов, может быть полезно, для повышения точности модели, рассматривать модели людей разных полов и возрастов по отдельности, учитывая их различия [15]. На данный момент в работе имя пользователя не будет значимой переменной, поэтому эти данные тоже удаляются. Итоговый вид представлен таблицей 1.

Таблица 1 – Итоговый вид набора данных, первые 5 строк

	acceleration_x	acceleration_y	acceleration_z	gyro_x	gyro_y	gyro_z
1	0.265	-0.7814	-0.0076	-0.059	0.0325	-2.9296
2	0.6722	-1.1233	-0.2344	-0.1757	0.0208	0.1269
3	0.4399	-1.4817	0.0722	-0.9105	-0.9105	-2.4367
4	0.3031	-0.8125	0.0888	0.1199	0.1199	-2.9336
5	0.4814	-0.9312	0.0359	0.0527	0.0527	2.4922

Здесь столбец activity содержит целевую переменную, а данные, содержащиеся в остальных столбцах, могут в дальнейшем выступать в качестве признаков.

4.2. Анализ данных

Одним из основных показателей качества набора данных является его сбалансированность. Несбалансированность данных негативно сказывается на работе нейронных сетей — алгоритм игнорирует малочисленный класс, что приводит плохому результату классификации. В таком случае в обучающую выборку попадает такой небольшой процент данных, относящихся к малочисленному классу, что модель не может построить эффективные взаимосвязи для определения данного класса [9].

С помощью методов библиотеки Pandas можно представить набор данных в графическом виде и посмотреть распределение классов и значений переменных (рисунки 4–7).

На рисунке 4 представлено распределение по классам – “ходьба” и “бег”. По рисунку 4 видно, что данные хорошо сбалансированы, распределение близко к идеальному – половина данных относится к первому классу, и половина – ко второму.

Рисунок 4 – Распределение данных по классам (значениям целевой переменной)

Распределение данных по датам, в которые их собирали, с 30.06.2017 по 17.07.2017, изображено на рисунке 5:

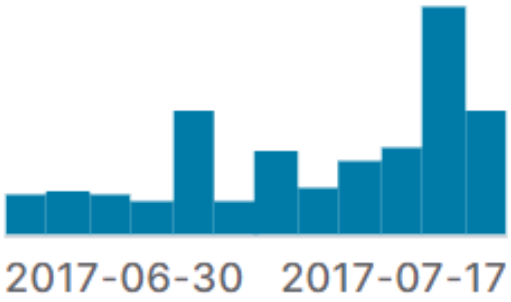

Рисунок 5 – Распределение данных по датам

Далее рассматривается распределение значений признаков. Распределения данных по ускорениям имеют вид нормального, с небольшой асимметрией (рисунок 6):

Рисунок 6 – Распределение данных по показаниям акселерометра, соответственно по каждой из трех осей, x, y и z

Распределения данных по показаниям гироскопа также имеют вид нормальных распределений, само распределение более симметрично, чем в случае акселерометра (рисунок 7):

Рисунок 7 – Распределение данных по показаниям гироскопа, соответственно по каждой из трех осей, x, y и z.

С точки зрения оценки качества данных кроме сбалансированности выделяют также следующие параметры:

- Измерения внутреннего качества:

1. Достоверность (точность) – показатель того, насколько значения данных согласуются с идентифицированным источником правильной информации. Данные, представленные в наборе собраны с помощью смартфона человеком, поставившим задачу собрать данные о своих передвижениях для их дальнейшего изучения, поэтому достоверность данных по большей части зависит от качества датчиков смартфона и программы, обрабатывавшей данные с датчиков смартфона, в точности которых сомневаться не приходиться, так как смартфон, а именно Apple iPhone 5s являлся на 2017-й год качественным устройством. В таком случае данные точны.

2. Происхождение – отражает надежность данных, важным аспектом которой является способность идентификации источника любого элемента данных. Представленные данные содержат атрибуты, такие, как дата, время и имя пользователя, к которому относятся показания датчиков, которые позволяют нам проследить источник получения данных.

3. Структура данных – отражается структурной согласованностью в представлении аналогичных значений атрибутов как в пределах одного и того же набора данных. В случае данного набора данных аналогичные атрибуты строго типизированы, имеют одинаковые синтаксические форматы и структуры.

4. Семантическая согласованность (семантика) – относится к согласованности определений между атрибутами в модели данных, а также атрибутов с одинаковыми именами в различных наборах

данных и характеризует степень, в которой сходные объекты данных имеют общие имена и значения. Для данного набора данных все сходные объекты данных распределены по соответствующим графам, согласованность не нарушается [4].

- Измерения контекстного качества:

1. Полнота – относится к ожиданию того, что определенным атрибутам будут присвоены значения в наборе данных соответствующие установленным в их отношении правил. Всем разделам набора присвоены соответствующие значения, полнота соблюдена.

2. Согласованность данных друг с другом – отражает их целостность и их внутреннюю непротиворечивость. Данные согласованы друг с другом.

3. Своевременность данных – время, прошедшее с момента возникновения события до момента, когда данные, представляющие его оказываются доступны для использования. С момента создания набора данных прошло порядка 7 лет. За это время датчики движения, установленные в смартфонах, не претерпели существенных изменений, равно как и анатомия человека, поэтому набор данных удовлетворяет этому пункту.

4. Разумность данных – показатель, определяющий степень, в которой значения данных имеют разумный или понятный тип и размер. Показания датчиков и значения целевой переменной имеют понятное и разумное выражение [13].

Таким образом, представленный набор данных можно назвать качественным, дальнейшее изучение имеет смысл.

4.3. Валидация модели

Для дальнейшего решения задачи необходимо иметь возможность оценить качество получившихся моделей. Оценочные метрики будут рассмотрены далее, а в этом разделе будет рассмотрен еще один шаг в построении предиктивной модели, игнорировать который нельзя, чтобы не допустить ошибку. Дело в том, что делать прогнозы на основе обучающих данных и сравнивать эти прогнозы с целевыми значениями в обучающих данных неверно.

Например, точность любого, вычисленная для модели, которая обучалась и проверялась на одном и том же наборе данных, составит 1.0, но этот показатель нерепрезентативен [5].

Показатель, который только что был описан, можно назвать оценкой "по выборке". Была использована единая выборка как для построения модели, так и для ее оценки. Поскольку этот шаблон был получен на основе обучающих данных, модель будет выглядеть точной в тех же обучающих данных.

Однако, практическая ценность моделей заключается в прогнозировании на основе новых данных. Измеряется производительность на основе данных, которые не использовались при построении модели. Самый простой способ сделать это - исключить некоторые данные из процесса построения модели, а затем использовать их для проверки точности модели на данных, которые ранее не использовались. Эти данные называются данными проверки достоверности (валидации) [11].

Сделать такое разбиение позволяет инструмент train test split (разбиение на обучающие и проверяющие данные) библиотеки scikit learn, предназначенной для машинного обучения. Выбирать разбиение данных надо так, чтобы в обеих группах (обучающей и проверяющей) было достаточно данных, они были бы сбалансированы.

На рисунке 8 представлено плохое разбиение по выборкам:

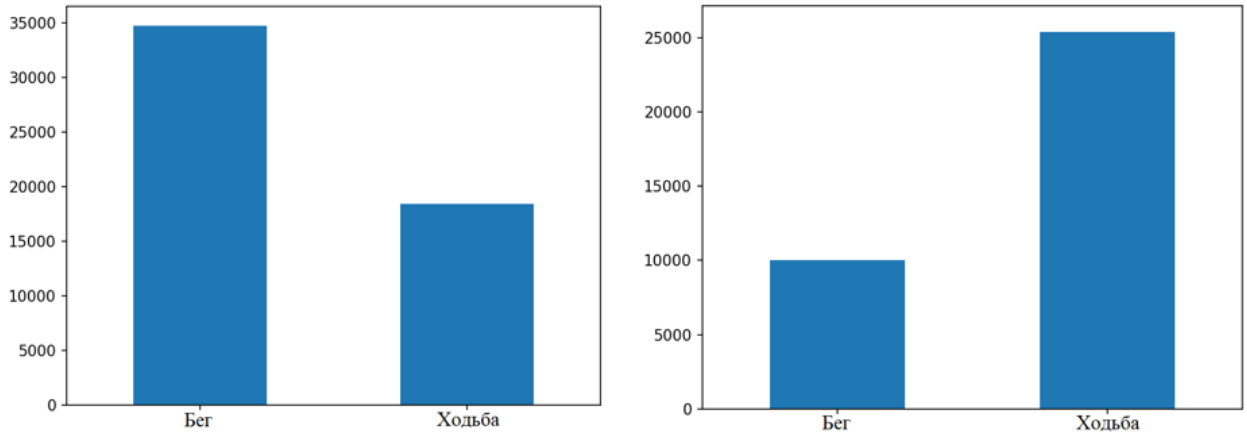

Рисунок 8 – случай некачественного разбиения данных, слева – распределение значений целевой переменной по обучающей выборке, справа - по проверяющей

Основная проблема такого разбиения заключается в том, что модель учится на основании данных, в которых преобладает один вид движения, а предсказывать она должна будет данные, большую часть которых составляет другой тип движения. Соответственно, ошибок в предсказанных данных будет достаточно много.

В дальнейшей работе используется сбалансированное разбиение в пропорции 1:1, что означает, что половина всех данных используется для обучения модели, вторая половина – для проверки. Распределение значений целевой переменной в обеих выборках изображено на рисунке 9:

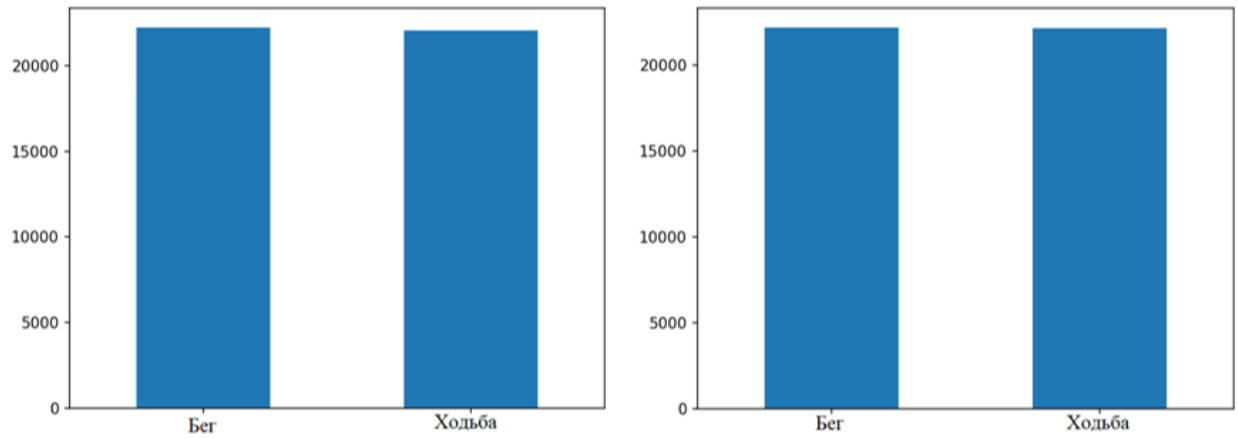

Рисунок 9 – случай качественного разбиения данных, слева – распределение значений целевой переменной по обучающей выборке, справа - по проверяющей

4.4. Метод K-ближайших соседей

Метод K-ближайших соседей крайне популярен, поскольку он прост для реализации, настройки, а главное – для понимания. Основная суть метода – оценка объекта на основании его соседей. Таким образом, объект классифицируется так, как классифицированы K его ближайших соседей, класс которых известен [12].

Метод можно проиллюстрировать рисунком 10:

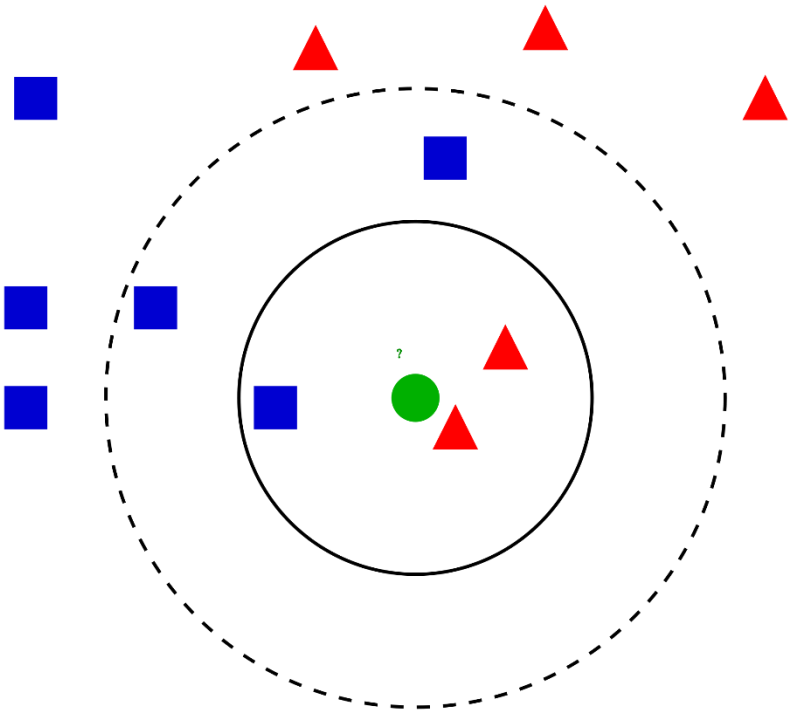

Рисунок 10 – Иллюстрация метода K-ближайших соседей

Пусть имеются два класса – синие квадраты (первый класс) и красные треугольники (второй класс). Задача – классифицировать неизвестный объект, представленный зеленым кругом. Если K = 3, предсказанный класс – второй, красный треугольник, так как их большинство среди трех ближайших соседей. K = 5 даст другой результат – первый класс.

Основные понятия метода K-ближайших соседей:

- Евклидова метрика – расстояние между двумя точками в евклидовом пространстве. Данная метрика хорошо работает в случае малых измерений, но плохо в случае больших, и вообще не работает, когда

признаки выражены категориальными переменными. Формула вычисления евклидова расстояния между точками p и q:

$$d(p,q)=\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

- Нормализация – значения различных признаков изменяются, соответственно, в различных диапазонах. В таком случае метод сильнее зависит от признаков, изменяющихся в большем диапазоне, так как такой признак существеннее изменяет дистанцию. Следовательно, данные зачастую проходят нормализацию. Применяется два вида нормализации:

  1. MinMax-нормализация:

$$x' = \frac{x - \min[X]}{\max[X] - \min[X]},$$

  в таком случае все значения лежат в диапазоне от 0 до 1.

  2. Z-нормализация:

$$x' = \frac{x - \mu}{\sigma},$$

  где μ – математическое ожидание x, σ – дисперсия.

4.5. Метод сверхслучайных деревьев

Extra trees (сокращение от extremely randomized trees, сверхслучайные деревья) — это метод машинного обучения с учителем, использующий деревья решений. Дерево решений, структура которого представлена на рисунке 11, — это древовидный метод, в котором любой путь, начинающийся от корня, описывается последовательностью, разделяющей данные, до тех пор, пока не будет достигнут логический результат в листовом узле.

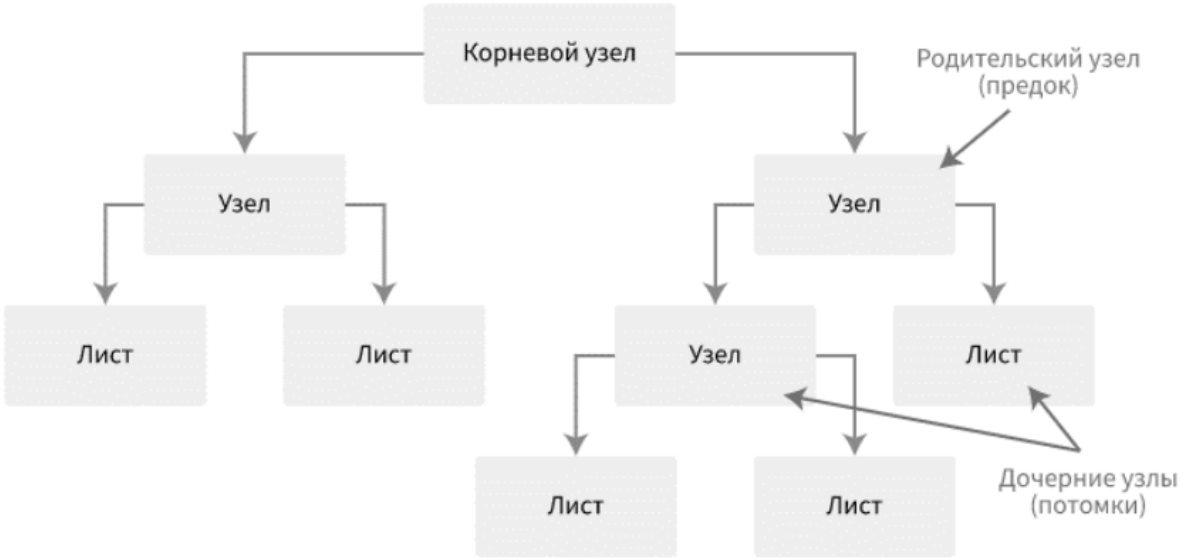

Рисунок 11 – Структура дерева решений

Дерево решений представляет собой последовательную модель, которая эффективно и связно объединяет серию основных тестов, где числовая характеристика сравнивается с пороговым значением в каждом тесте.

Алгоритмически этот метод реализуется следующим образом. Пусть задано обучающее множество S, содержащее n примеров, для каждого из которых задана метка класса C_i, $i = 1, \ldots, k$, и m атрибутов A_j, $j = 1, \ldots, m$, которые определяют принадлежность объекта к классу. Тогда возможны три случая:

1. Все примеры множества S имеют одинаковую метку класса C_i (т.е. все обучающие примеры относятся только к одному классу). Обучение не имеет смысла, будет создан только один лист.
2. Множество S вообще не содержит примеров. В этом случае для него тоже будет создан лист (применять правило, чтобы создать узел, к пустому множеству бессмысленно).
3. Множество S содержит обучающие примеры всех классов C_k. Тогда множество S разбивается на подмножества классов. Для этого выбирается один из атрибутов A_j множества S, который содержит два и более уникальных значения, где p — число уникальных значений признака. Множество S разбивается на p подмножеств $(S_1, S_2, ..., S_p)$,

каждое из которых включает примеры, содержащие соответствующее значение параметра. Затем выбирается следующий параметр и разбиение повторяется. Так продолжается, пока в листе не окажутся объекты одного класса, либо пока не сработает иной выбранный механизм остановки.

Алгоритм сверхслучайных деревьев, как и алгоритм случайных лесов, создает множество деревьев решений, но выборка для каждого дерева случайная, без замены. Это создает набор данных для каждого дерева с уникальными выборками. Определенное количество объектов из общего набора объектов также выбирается случайным образом для каждого дерева.

Атрибутом разбиения внутри дерева может служить вычисление локально оптимального значения с использованием индекса Джини:

$$\mathrm{Gini}(Q)=1-\sum_{i=1}^{n} p_i^2,$$

где Q — результирующее множество, n — количество классов, p_i — вероятность предсказания класса i; или энтропия:

$$H=-\sum_{i=1}^{n}\frac{N_i}{N}\log\left(\frac{N_i}{N}\right),$$

где n – количество классов, N – количество объектов во всем множестве, N_i – количество объектов класса i [17].

4.6. Наивный байесовский классификатор

Наивный байесовский классификатор основывается на формуле Байеса со строгим предположением о независимости признаков между собой.

Преимущество метода в том, что независимость признаков позволяет производить вычисления в одномерном пространстве, а не многомерном, как в случае взаимосвязанности всех признаков. Такое предположение может оказаться неверным и испортить модель, однако результаты, полученные при сравнении моделей, говорят о том, что в данной задаче наивный Байес показывает себя хорошо.

Формула Байеса выглядит следующим образом:

$$P(A|B)=\frac{P(B|A)\,P(A)}{P(B)},$$

где P(A), P(B) – вероятности событий A и B, P(A|B) и P(B|A) – вероятность события A, когда произошло B, и наоборот.

В терминах машинного обучения:

$$P(y_k|X)=\frac{P(X|y_k)\,P(y_k)}{P(X)},$$

где $P(y_k)$ – вероятность принадлежности случайного объекта к классу y_k, P(X) – апостериорная вероятность признаков X, $P(y_k|X)$ — апостериорная вероятность, что объект принадлежит к классу y_k при признаках X, $P(X|y_k)$ – вероятность признаков X при классе y_k.

Формула для n признаков:

$$P(y_k|X_1, \ldots, X_n)=\frac{P(y_k)\cdot \prod_{i=1}^{n} P(X_i|y_k)}{P(X_1, \ldots, X_n)}$$

В итоге:

$$y_k \propto \arg\max_{y_k}\left(P(y_k)\cdot \prod_{i=1}^{n} P(X_i|y_k)\right)$$

В датасете признаки распределены по Гауссу, поэтому логично использовать гауссовский наивный байесовский классификатор:

$$P(x_i|y)=\frac{1}{\sqrt{2\pi\sigma_y^2}}\cdot \exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right),$$

Где $P(x_i|y)$ – вероятность отношения признака x_i к классу y, μ_y и σ_y – математическое ожидание и дисперсия x_i [7].

4.7. Отбор признаков

Используя метод сверхслучайных деревьев, ввиду его точности, которая была наибольшей среди всех методов, проводится отбор признаков. График зависимости точности метода от набора используемых признаков представлен на рисунке 12, где значения наборов признаков лежат на оси x и обозначаются ax, ay, az – показания акселерометра по осям x, y и z, соответственно и gx, gy, gz – показания гироскопа по осям x, y и z.

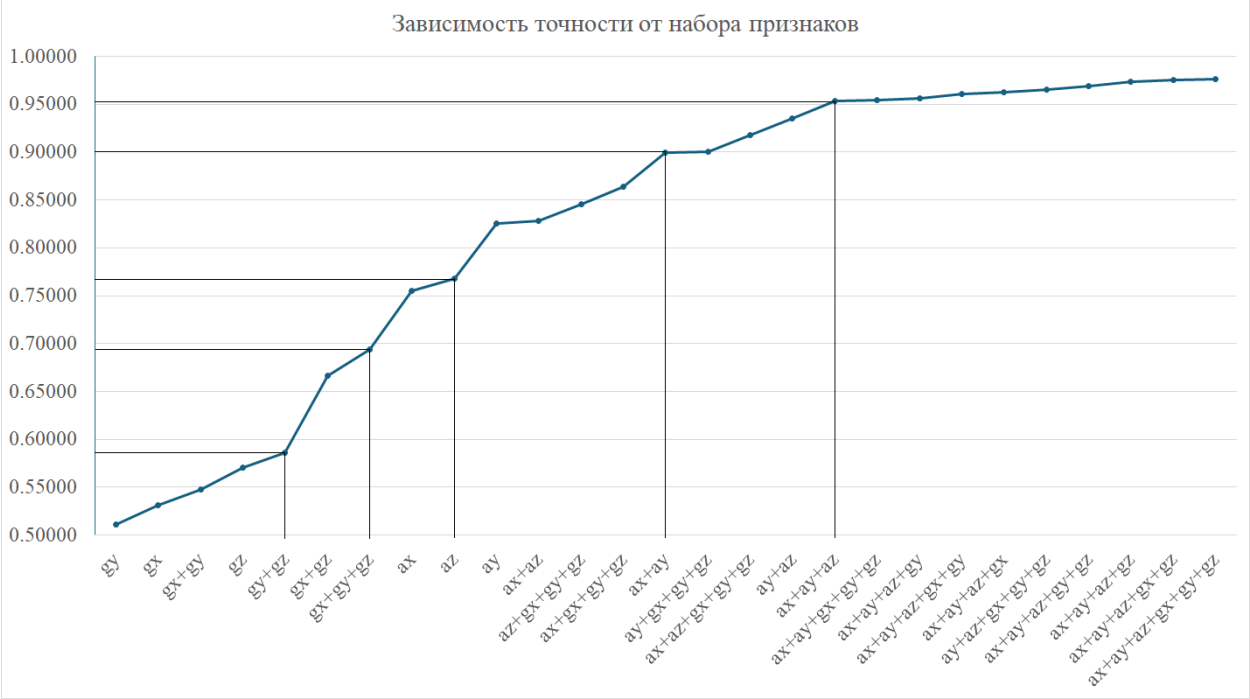

Рисунок 12 – График зависимости точности метода от набора используемых признаков

Оценка результатов будет производиться соответственно таблице 2.

Таблица 2 – Таблица оценивания точности метода

Точность	Оценка модели
0.9-1	Отлично
0.8-0.9	Очень хорошо
0.7-0.8	Хорошо
0.6-0.7	Удовлетворительно
0.5-0.6	Плохо

Пользуясь графиком на рисунке 14 и таблицей 2 оценки модели, можно сделать вывод о том, что модель, обученная на основании данных с гироскопа, не

сильно отличается в точности от случайного предсказателя, имеющего точность около 0.5, такую модель нельзя использовать. Модель, обученная только на показаниях акселерометра, не проходит порог точности в 0.85, но показывает себя лучше, чем предыдущая. Однако, такая модель все же не рекомендуется к использованию. Наилучшая точность, выше 0.95, свойственна моделям, обученным на показаниях обоих датчиков, либо для модели, обученной на показаниях трехосевого акселерометра.

В связи с тем, что на некоторых устройствах различные датчики могут быть рассинхронизированны по времени, и использование информации с датчиков, регистрирующих показания в разное время, может снижать точность прогнозирования, модель будет строиться на показаниях трехосевого акселерометра, без учета показаний гироскопа. Точность модели, обученной на таком наборе данных – 0.95.

4.8. Используемые метрики

Для понимания метрик необходимо так же знать, что в машинном обучении, в задаче классификации, на выходе получаются данные, которые можно разделить на 4 категории:

- Истинно положительные значения (TP — true positive): фактически истинные значения, которые были верно спрогнозированы как истинные.
- Ложноположительные значения (FP — false positive): фактически ложные значения, которые были неверно спрогнозированы как истинные.
- Ложноотрицательные значения (FN — false negative): фактически истинные значения, которые были неверно спрогнозированы как ложные.
- Истинно отрицательные значения (TN — true negative): фактически ложные значения, которые были верно спрогнозированы как ложные.

В данном случае можем рассматривать истинным значением activity = 1 (бег), ложным – activity = 0 (шаг). Тогда TP будет считаться случай, когда модель правильно предсказала бег (человек бежит, в графе активности записан бег, “1”),

FP – неправильно предсказала бег (на самом деле человек бежит, но в графе активности записан шаг, “0”), FN – неправильно предсказала шаг (на самом деле человек идет, но в графе активности записан бег, “1”), а TN - модель правильно предсказала шаг (человек идет, в графе активности записан шаг, “0”). На основании этих категория строится так же матрица ошибок (confusion matrix), изображенная на рисунке 13, в этом случае для метода сверхслучайных деревьев.

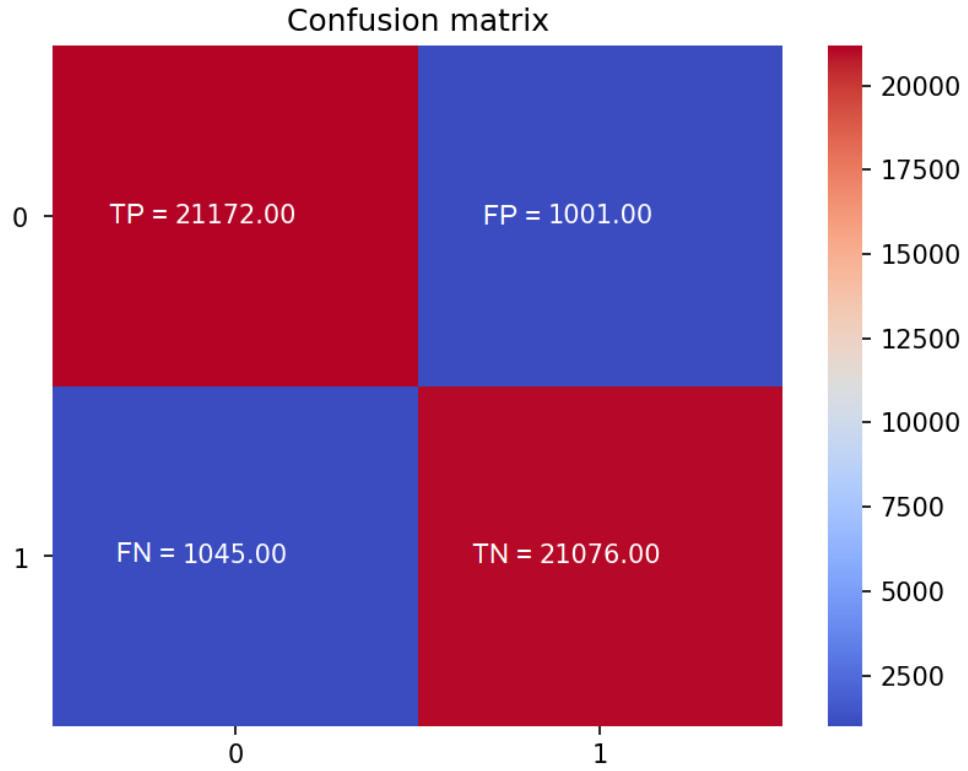

Рисунок 13 – матрица ошибок для метода сверхслучайных деревьев

Теперь можем рассмотреть используемые метрики.

1. Accuracy – точность, простой расчет, широко используемый на практике. Построенная модель машинного обучения классифицирует целевую переменную на основании входных данных. Вычисляется общее количество прогнозов, сделанных моделью, и сколько из этих прогнозов верны. В математическом представлении это будет выглядеть следующим образом:

$$\text{Точность} = \frac{\text{Число верных предсказаний}}{\text{Общее число предсказаний}} = \frac{TP + TN}{TP + FP + FN + TN}$$

Данная метрика хорошо работает со сбалансированными наборами данных, однако в случае несбалансированного набора данных, данная метрика не является показателем качества модели.

Здесь и далее конкретные расчеты производятся для случая реализации метода сверхслучайных деревьев (ET):

$$\text{Точность} = \frac{TP + TN}{TP + TN + FP + FN} = \frac{21172 + 21076}{21172 + 21076 + 1001 + 1045} = 0.95381$$

2. Precision – прецизиозность, дословно - точность, отражает вероятность того, что модель правильно предскажет положительное значение:

$$\text{Прецизиозность} = \frac{\text{Число верных предсказаний положительного значения}}{\text{Суммарное число предсказаний положительного значения}} = \frac{TP}{TP + FP}$$

Другое название TNR (true negative rate, доля истинно отрицательных предсказаний).

Прецизиозность хорошо подходит для оценки качества модели в том случае, когда важно не допустить появление ложноположительных результатов.

ET:

$$\text{Прецизиозность} = TNR = \frac{TP}{TP + FP} = \frac{21172}{21076 + 1001} = 0.95901$$

3. Recall – полнота или TPR (true positive rate, доля истинно положительных предсказаний), показывает относительное количество истинно положительных результатов. Является показателем того, как часто модель точно спрогнозировала положительный результат, когда результат является положительным, т. е. точность нахождения положительных значений в наборе данных:

$$\text{Полнота} = \frac{\text{Число верных предсказаний положительного значения}}{\text{Число верных и неверных предсказаний положительного значения}} = \frac{TP}{TP + FN}$$

Цель состоит в том, чтобы “наказывать” модель за полученные в результате ложноотрицательные значения.

ЕТ:

$$\text{Полнота} = \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{21172}{21076 + 1045} = 0.95710$$

4. F1 – это метрика, делающая акцент на точности положительных прогнозов. Метрика F1 пытается сбалансировать значения полноты и точности, считая среднее гармоническое значение между ними:

$$\text{F1} = 2 \times \frac{1}{\frac{1}{\text{Полнота}} + \frac{1}{\text{Прецизиозность}}} =$$

$$= 2 \times \frac{\text{Прецизиозность} \times \text{Полнота}}{\text{Прецизиозность} + \text{Полнота}} = \frac{\text{TP}}{\text{TP} + \frac{\text{FP} + \text{FN}}{2}}$$

Чем хуже сбалансирован набор данных, тем ниже F1, не зависимо от остальных факторов.

ЕТ:

$$\text{F1} = \frac{21172}{21076 + \frac{1001 + 1045}{2}} = 0.95805$$

5. Kappa – Каппа Коэна, которая объясняется так: поскольку использование точности (Accuracy) может быть необоснованно в задачах с плохо сбалансированными данными, необходимо провести нормировку точности. Для этого используют статистику chance adjusted index (индекс с поправкой на вероятность): нормировка происходит с помощью точности, которую можно получить случайно $\left(\text{Accuracy}_{\text{chance}}\right)$. Под случайной здесь понимаем точность решения, которое получено из нашего случайной перестановкой ответов.

$$\kappa = \frac{\text{Точность} - \text{Точность}_{\text{случайная}}}{1 - \text{Точность}_{\text{случайная}}}$$

Чтобы лучше понимать отличие точности наблюдаемой от точности случайной, рассмотрим следующие формулы, пользуясь обозначениями предсказаний из таблицы 2:

Таблица 3 – Обозначение предсказаний

	a = 0 (Positive)	a = 1 (Negative)
y = 0 (True)	n_{00}	n_{01}
y = 1 (False)	n_{10}	n_{11}

В данной таблице столбцы “a” отвечают за реальное значение переменной, а строки “y” – за предсказанное, тогда

$$\text{Точность} = \frac{n_{00} + n_{11}}{n}, \text{ где n - сумма всех предсказаний}$$

$$\text{Точность}_{\text{случайная}} = \frac{n_{00} + n_{01}}{n} \times \frac{n_{00} + n_{10}}{n} + \frac{n_{11} + n_{01}}{n} \times \frac{n_{11} + n_{10}}{n}$$

В терминах матрицы ошибок формула для Каппы Коэна будет выглядеть следующим образом:

$$\kappa = \frac{2 \times (TP \times TN - FP \times FN)}{(TP + FP) \times (FP + TN) + (TP + FN) \times (FN + TN)}$$

ET:

$$\kappa = \frac{2 \times (21172 \times 21076 - 1001 \times 1045)}{(21172 + 1001) \times (1001 + 21076) + (21172 + 1045) \times (1045 + 21076)} =$$

$$= 0.90762$$

6. MCC – коэффициент корреляции Мэттьюса (Matthews correlation coefficient), еще один удобный инструмент для оценки точности метода, работающего с несбалансированной выборкой. Формула MCC выглядит так:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}},$$

$$-1 \leq MCC \leq 1$$

Он изменяется в диапазоне от -1 до 1. MCC = 1 означает безупречную классификацию, когда фактические и предсказанные классы совпадают для всех обучающих примеров (т. е. ложноположительные и ложноотрицательные классификации отсутствуют). Модель с MCC = 0, соответствует случайному предсказателю. MCC = −1 указывает на то, что модель не сделала ни единого верного предсказания (т. е. истинно положительные и истинно отрицательные классификации отсутствуют).

ET:

$$MCC = \frac{21172 \times 21076 - 1001 \times 1045}{\sqrt{(21172 + 1001) \times (21172 + 1045) \times (21076 + 1001) \times (21076 + 1045)}} = 0.90762$$

7. AUC (AUC-ROC) – Receiver Operating Characteristics curve (кривая рабочих характеристик приемника) и Area Under Curve (площадь под кривой).

ROC-кривая позволяет оценить качество работы классификатора, объединяя в себе характеристики TNR и TPR, давая оценку модели с точки зрения ее способности верно определять оба класса. Для этого

строится график в координатах FPR (false positive rate, доля ложноположительных результатов) и TPR. Вычисляется так: FPR = 1 – TNR = 1- $\frac{TN}{TN + FP} = \frac{FP}{TN + FP}$, значения этой величины могут изменяться от 0 до 1.

Таким образом ROC-кривая имеет следующее представление, показанное на рисунке 14:

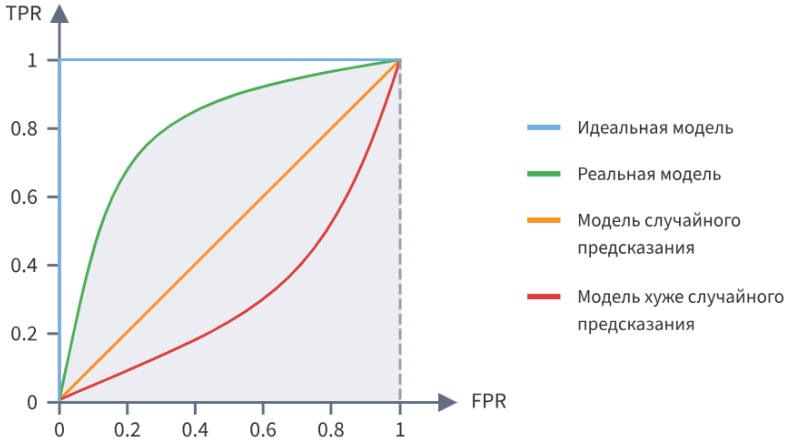

Рисунок 14 – ROC-кривая, общее представление

ROC-кривая для метода сверхслучайных деревьев изображена на рисунке 15:

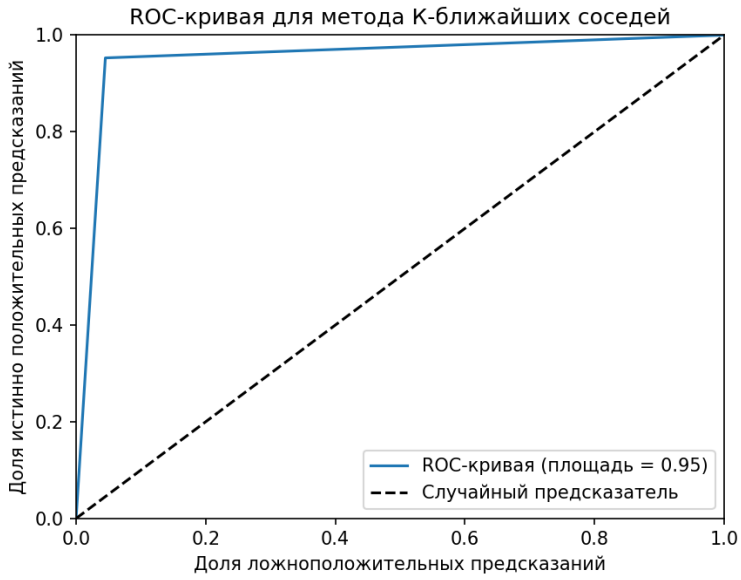

Рисунок 15 – ROC-кривая метода ET, площадь под кривой = 0.95

Для идеальной модели ROC-кривая будет ломанной, проходящей через точки (0,0), (0,1) и (1,1), площадь под кривой окажется равной 1. Однако, это будет в случае, если модель идеальна, что, скорее всего, недостижимо на практике. Поэтому отдельной метрикой является AUC, площадь под кривой, где 1 – идеальный предсказатель, не допускающий ошибок, 0.5 – соответствует случайному предсказателю, 0 – предсказатель не сделал ни одного правильного предположения. Для оценки будет принята следующая шкала (таблица 4):

Таблица 4 – шкала оценивания модели по величине AUC

Точность	Оценка модели
0.9-1	Отлично
0.8-0.9	Очень хорошо
0.7-0.8	Хорошо
0.6-0.7	Удовлетворительно
0.5-0.6	Плохо

ET: AUC = 0.95381, это отличный результат.

4.9. Сравнение различных моделей

Сравним описанные алгоритмы машинного обучения. Для этого рассмотрим сводную таблицу методов (таблица 5):

Таблица 5 – Сравнение методов машинного обучения, желтым выделен лучший результат

	Model	Accuracy	AUC	Recall	Prec.
knn	K Neighbors Classifier	0.9601	0.9601	0.9585	0.9615
nb	Naive Bayes	0.9570	0.9569	0.9207	0.9926
et	Extra Trees Classifier	0.9538	0.9538	0.9528	0.9547
	F1	Kappa	MCC	T (Sec)	
knn	0.9600	0.9202	0.9202	5.8255	
nb	0.9553	0.9139	0.9163	0.0271	
et	0.9537	0.9076	0.9076	0.0786	

Обзор таблицы на примере метода KNN, он же K-Neighbors Classifier, метод K-ближайших соседей. В первом столбце содержится сокращенное наименование модели, например KNN. В столбце Model записаны полные названия моделей машинного обучения, например K-Neighbors Classifier. Далее идут столбцы, содержащие метрики машинного обучения, обзор которым был дан в предыдущем разделе. В последнем столбце указано время в секундах, которое требуется методу для выполнения.

Лучшим методом по всем метрикам, кроме прециозности, является метод К-ближайших соседей. Лучшие прециозность и время выполнения программы у наивного Байеса.

Наиболее важными показателями в данной работе являются точность и время. Точность важна, так как в данном случае данные сбалансированы по целевой переменной, и нет необходимости оценивать модель по ее способности верно угадывать один определенный класс. Таким образом, точность позволяет достаточно полно оценить качество предсказания данных. Время выполнения программы играет важную роль, потому что оно отражает то, насколько модель требовательна к производительности устройства. В случае задачи определения вида движения человека может возникнуть запрос на быстродействующую, не требовательную к производительности модель.

Стоит отметить, что в таблице приводятся значения для неоптимизированных методов, то есть по всем показателям есть потенциал для улучшения.

4.10. Выбор гиперпараметров для метода К-ближайших соседей

Для оптимизации метода, повышения его точности, его необходимо настроить. В случае метода К-ближайших соседей таким параметром будет являться К – количество соседей. Изменение точности представлено на рисунке 16:

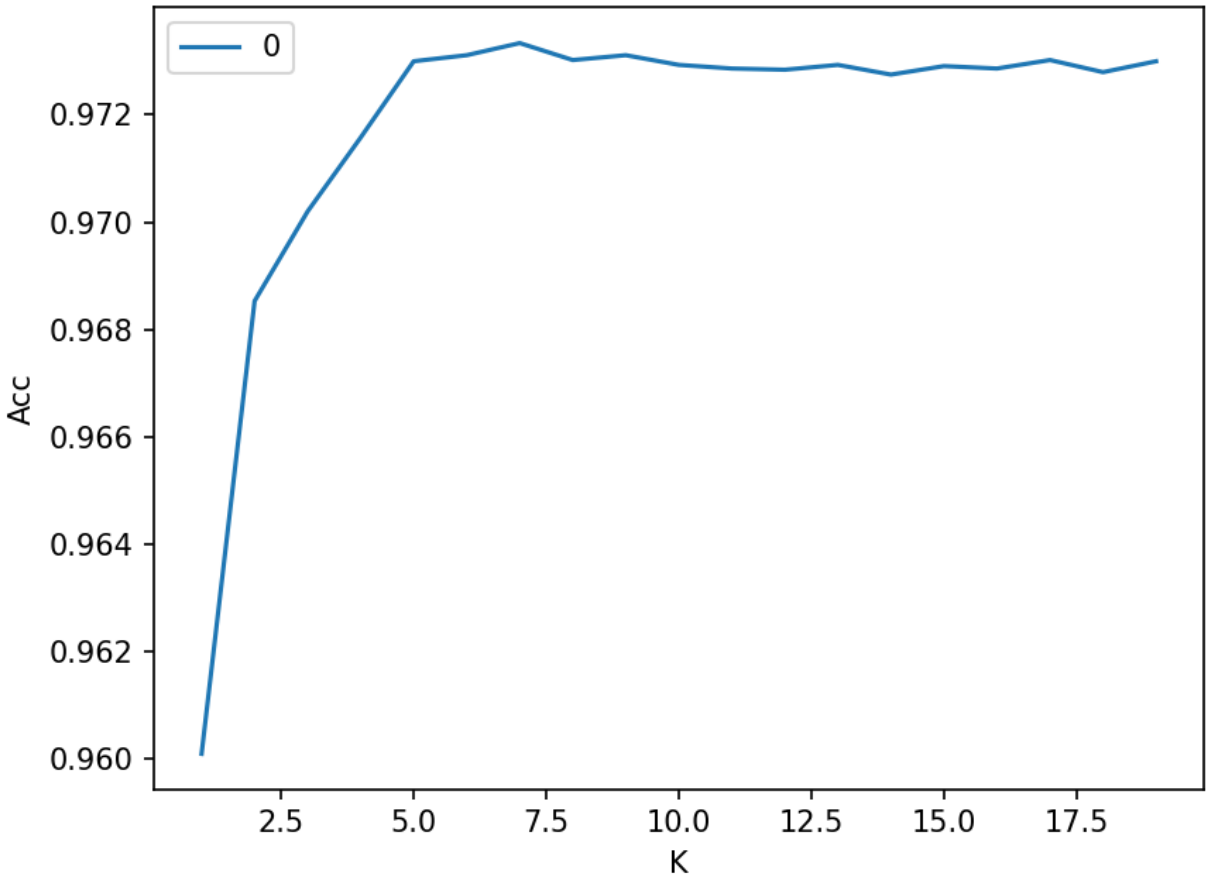

Рисунок 16 – График зависимости точности метода от числа соседей, K

Точность метода при K = 7 составила 0.97334 после настройки против 0.96008 до настройки. Время выполнения программы - 4.521 с.

4.11. Выбор гиперпараметров для метода сверхслучайных деревьев

Для метода сверхслучайных деревьев подбирается оптимальное число деревьев, создаваемых в процессе выполнения. Для этого строятся графики, отражающие зависимость точности (рисунок 17) и времени выполнения программы (рисунок 18) от количества деревьев:

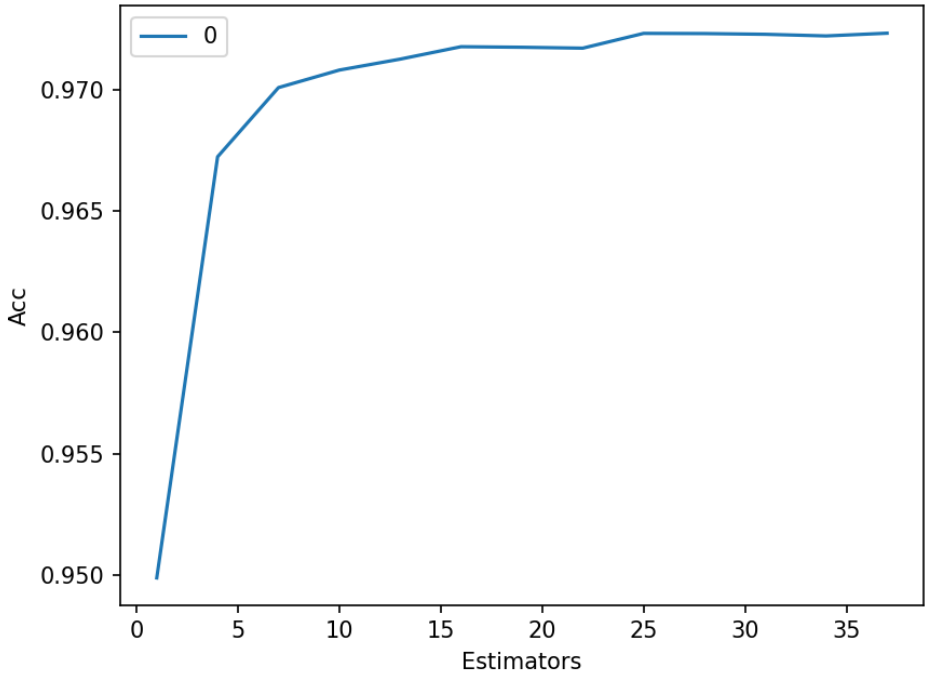

Рисунок 17 – График зависимости точности метода от числа деревьев

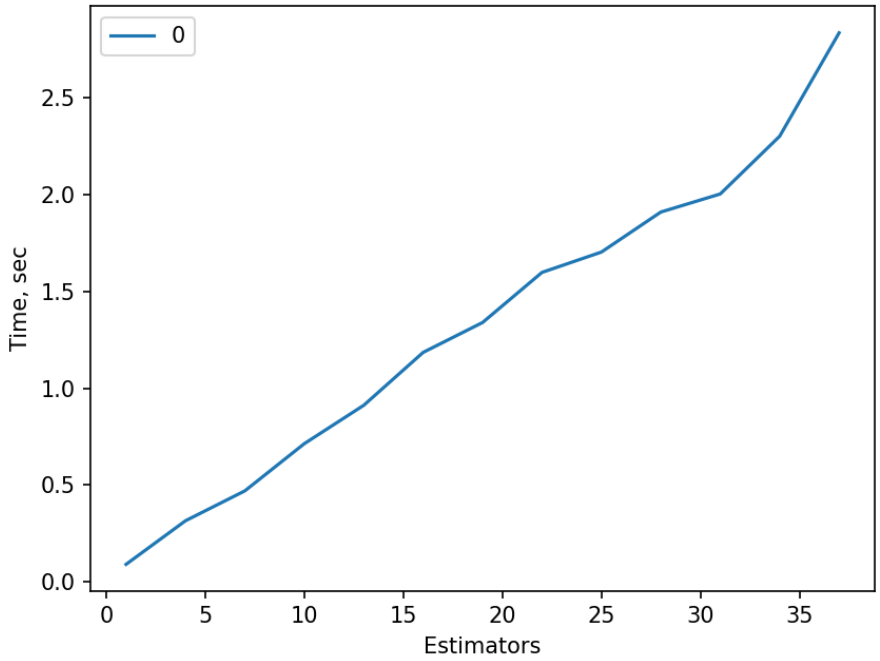

Рисунок 18 – График зависимости времени выполнения программы от числа деревьев

При построении 25 деревьев точность достигает локального максимума около 0.972 при времени выполнения 1.8 с. Выбирается данное число деревьев для дальнейшего использования в работе.

Затем рассматривается зависимость производительности метода от максимальной глубины дерева, которая показывает, сколько в каждом дереве будет разбиений. Результаты представлены на рисунке 19 и рисунке 20:

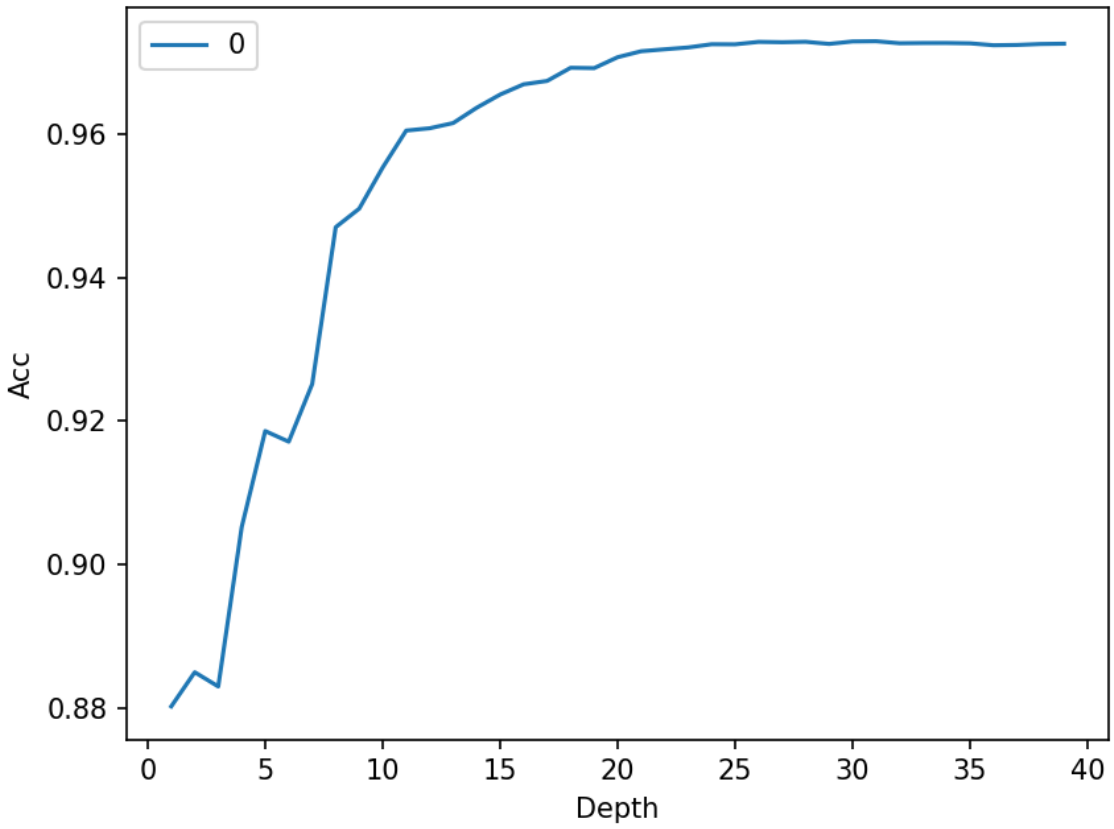

Рисунок 19 – График зависимости точности метода от глубины дерева

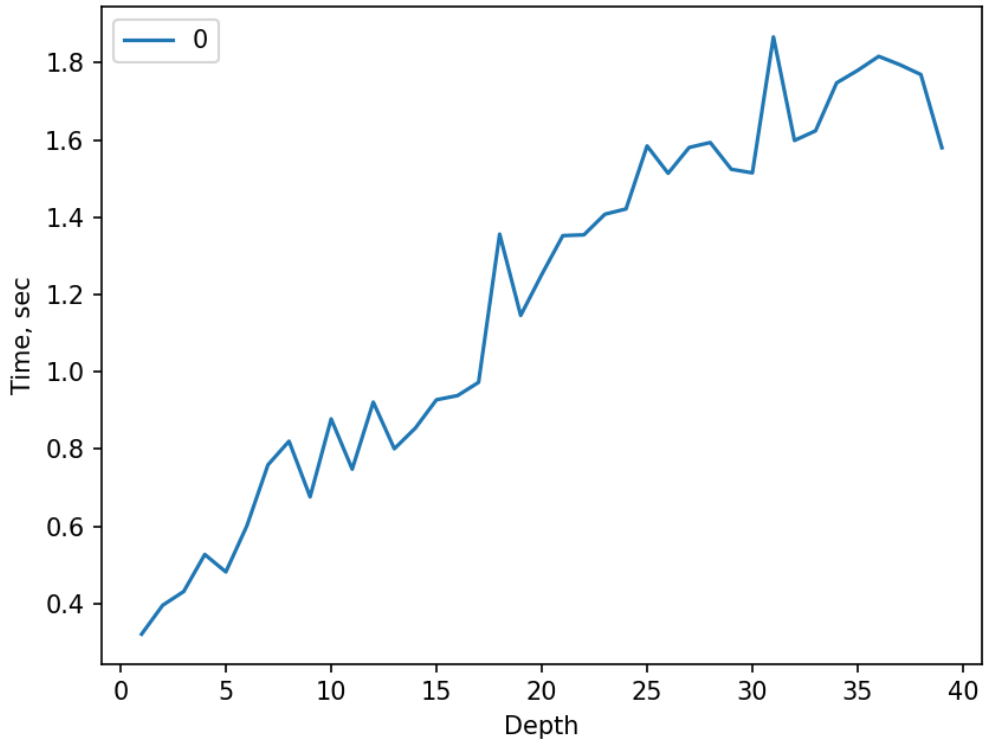

Рисунок 20 – График зависимости времени выполнения программы от глубины дерева

Дальше рассматривается модель с глубиной дерева равной 24. Такой параметр позволяет достичь точности порядка 0.973 при времени выполнения программы 1.04 с.

4.12. Сравнение настроенных моделей

После настройки методов (за исключением метода наивного Байеса, у которого нет параметров, позволяющих произвести значимую настройку) были получены результаты, которые можно сравнить и выбрать наиболее подходящую модель. Результаты сравнения методов представлены в таблице 6 и на рисунках 21–26.

Таблица 6 – Сравнение настроенных методов, желтым выделен лучший результат

Model	Accuracy	AUC	Recall	Precision
K Neighbors Classifier	0.9733	0.9733	0.9585	0.9878
Extra Trees Classifier	0.9725	0.9725	0.9544	0.9901
Naive Bayes	0.9570	0.9570	0.9207	0.9926
	F1	Kappa	MCC	TT (Sec)
K Neighbors Classifier	0.9729	0.9467	0.9471	4.521
Extra Trees Classifier	0.9719	0.9450	0.9456	1.044
Naive Bayes	0.9553	0.9139	0.9163	0.027

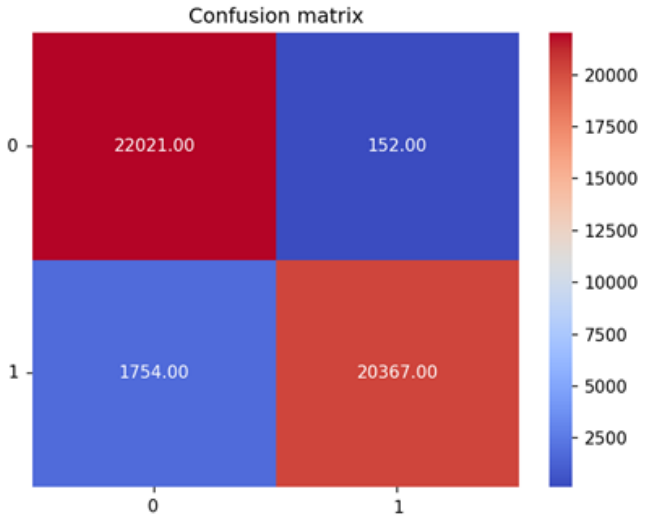

Рисунок 21 – Матрица ошибок для метода наивного Байеса

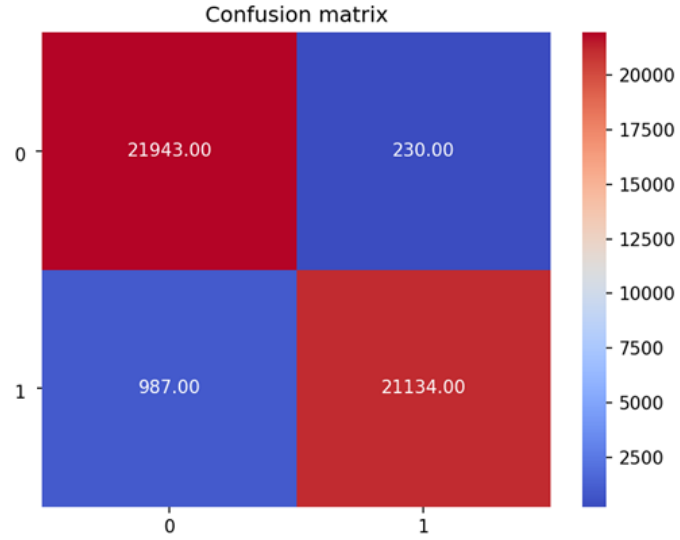

Рисунок 22 – Матрица ошибок для метода сверхслучайных деревьев

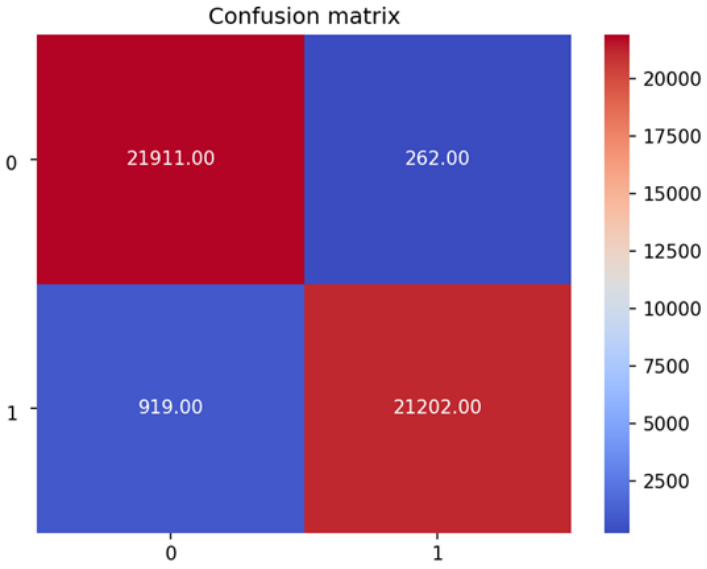

Рисунок 23 – Матрица ошибок для метода К-ближайших соседей

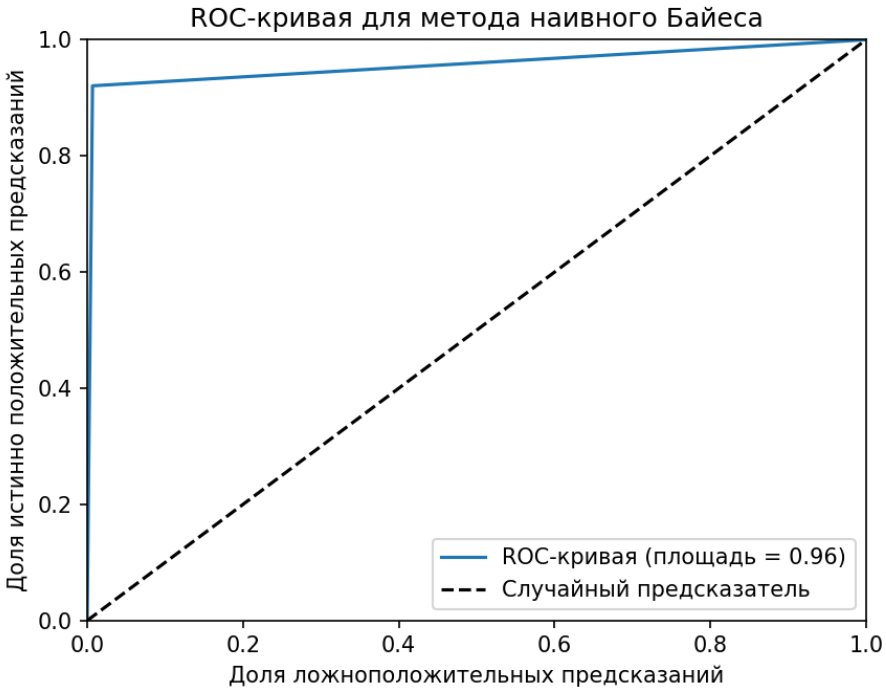

Рисунок 24 – ROC-кривая для метода наивного Байеса

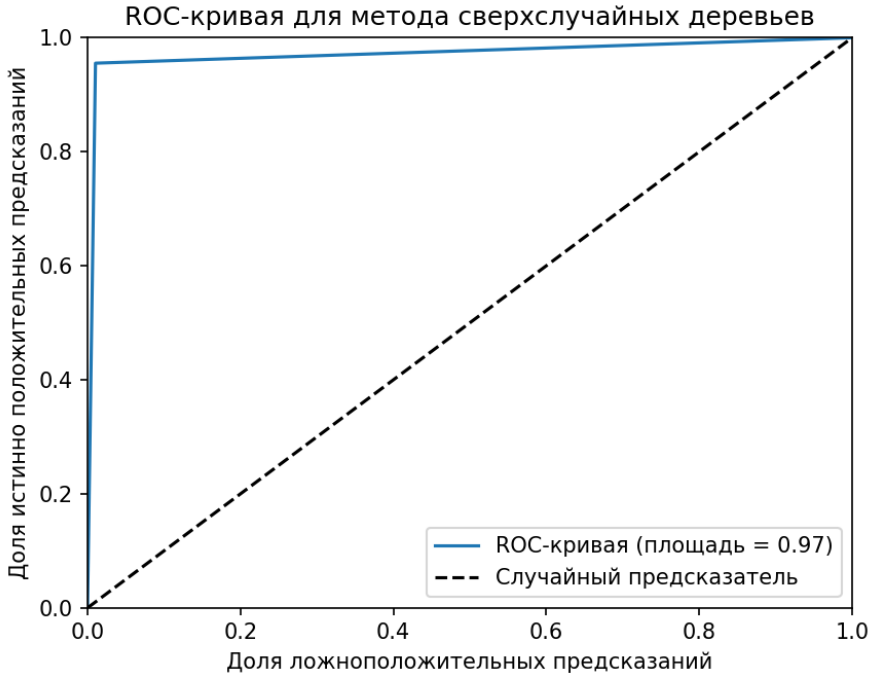

Рисунок 25 – ROC-кривая для метода сверхслучайных деревьев

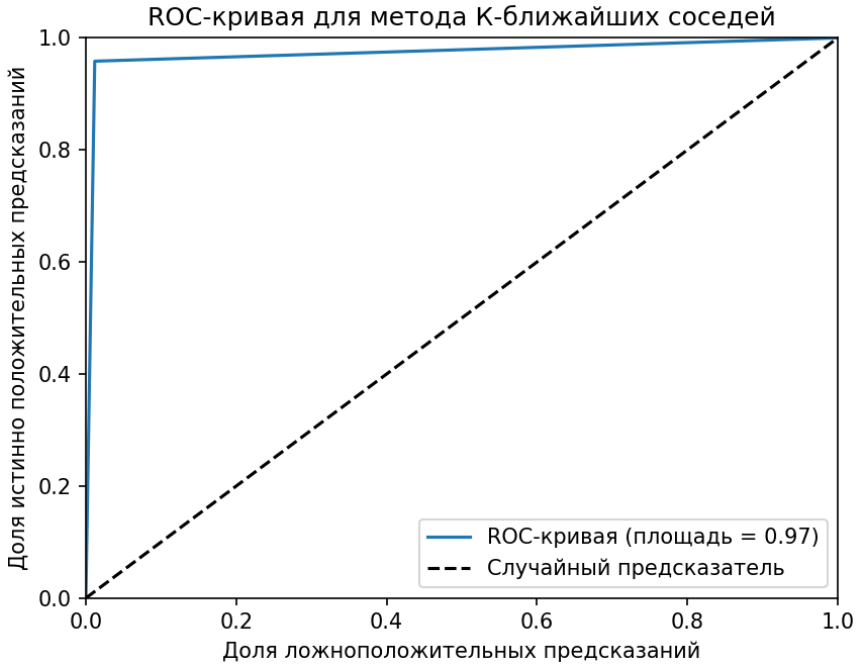

Рисунок 26 – ROC-кривая для метода К-ближайших соседей

Из таблицы видно, что лучшим как по точности, так и по большинству других метрик стал метод К-ближайших соседей.

Наивный Байес стал лучшим по прециозности, что означает, что он точнее остальных методов определяет бег, а также наивный Байес показал лучший результат по времени – в 39 раз быстрее, чем метод сверхслучайных деревьев, и в 167 раз быстрее метода К-ближайших соседей.

Метод сверхслучайных деревьев занял промежуточную позицию, показав результаты по большинству метрик хуже, чем метод К-ближайших соседей, и лучше, чем наивный Байес. Этот метод также второй по времени выполнения.

Для анализа и графического представления результатов будет использован метод К-ближайших соседей.

4.13. Анализ результатов

Анализируя полученную предиктивную модель можно выявить самый распространенный паттерн ошибок. Он представляет собой единичную аномалию – в ряду объектов одного класса возникает объект другого. К примеру, среди объектов, классифицированных как бег, есть одно значение ходьбы. Учитывая информацию из главы “Входные данные”, а именно то, что показания датчиков снимаются чаще, чем раз в 0.2 секунды, и принимая во внимание информацию из главы “Общие сведения”, подглава “1. Кинематика движения человека”, где говорится о том, что фаза шага длится около секунды, спортивной ходьбы – 0.5 секунды, бега трусцой – 1 секунды, спринтерского бега – 0.3 секунды, можно сделать вывод, что человек не способен перейти с шага на бег и обратно, либо наоборот – с бега на шаг и обратно, быстрее, чем за 0.4 секунды. Таким образом, можно отфильтровать значения, стоящие в рядах противоположных значений по одному или по двое. Это предположение можно проверить и оценить изменение точности, что отображено в таблице 7.

Таблица 7 – Сравнение метода К-ближайших соседей без фильтра и с фильтрами, желтым выделен лучший результат

Model	Accuracy	AUC	Recall	Precision
K Neighbors Classifier, сдвоенные аномалии удалены	0.99801	0.99830	1.00000	0.99526
K Neighbors Classifier, единичные аномалии удалены	0.99589	0.99643	0.99968	0.99056
K Neighbors Classifier	0.97334	0.97332	0.95846	0.98779
	F1	Kappa	MCC	
K Neighbors Classifier, сдвоенные аномалии удалены	0.99762	0.99592	0.99593	
K Neighbors Classifier, единичные аномалии удалены	0.99510	0.99156	0.99159	
K Neighbors Classifier	0.97290	0.94667	0.94709	

По таблице видно, что если убрать одиночные и сдвоенные аномалии, то метод становится ощутимо точнее. Для дальнейшего анализа будет использован метод К-ближайших соседей с исключенными одиночными и сдвоенными аномалиями.

ГЛАВА 5. ПРЕДСТАВЛЕНИЕ РЕЗУЛЬТАТОВ

На основании метода К-ближайших соседей с фильтром на аномалии можно рассмотреть графическое представление анализа суточного движения человека на рисунках 27–28.

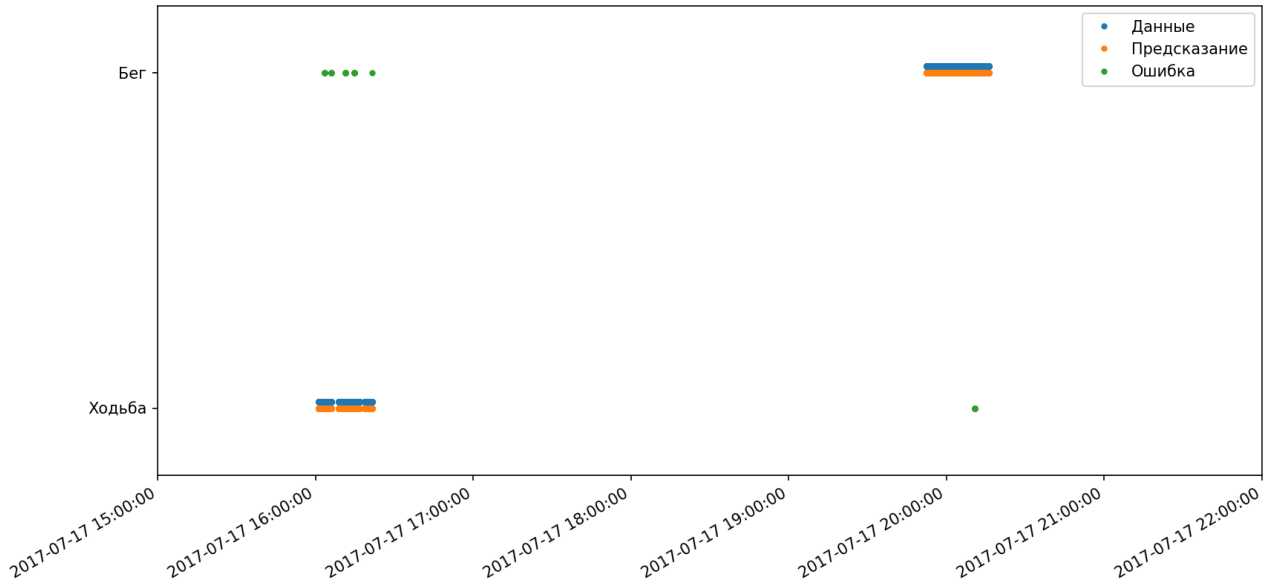

Рисунок 27 – Графическое представление суточной активности человека, масштаб – 24 часа

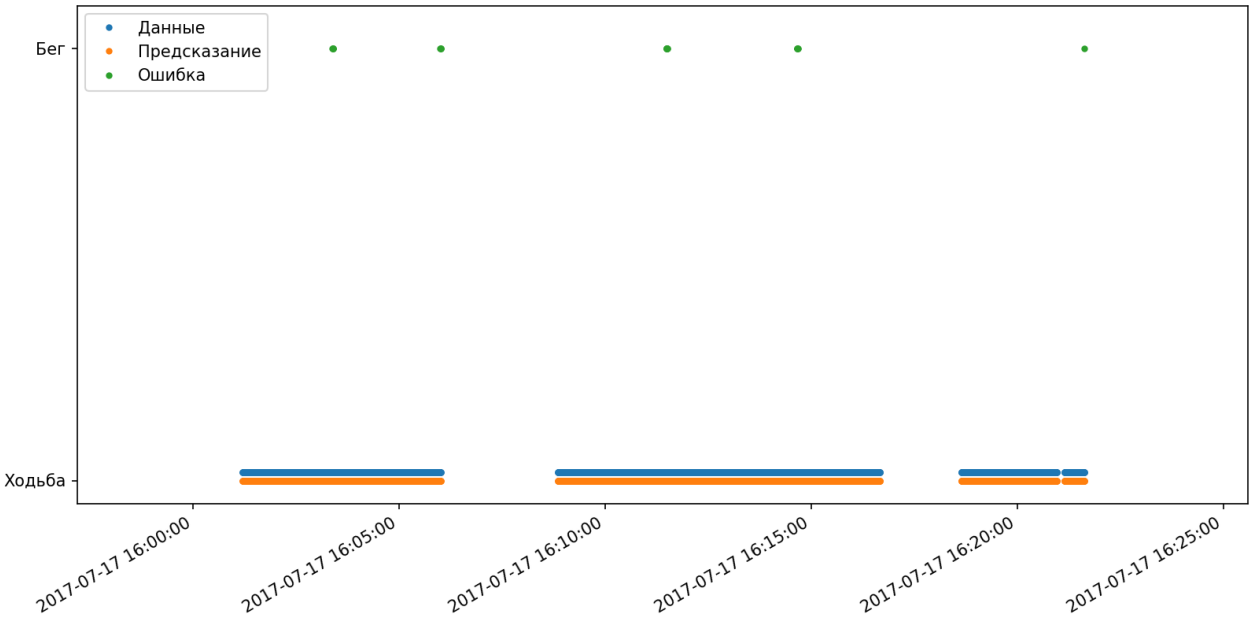

Рисунок 28 – Графическое представление суточной активности человека, масштаб – 25 минут

На рисунках синим отображаются известные значения активности, оранжевым – верно предсказанные, зеленым – предсказанные с ошибкой. Можно видеть, что подавляющее большинство значений предсказано верно.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была реализована модель для определения вида движения человека. Она позволяет отслеживать суточную двигательную активность, что позволит людям, занимающимся спортом или следящим за своим жизни получать доступ к анализу своей двигательной активности, полученном на основании данных с датчиков смартфона или другого устройства с акселерометром.

Были проведены обработка и анализ данных, полученных с телефона, сравнительный анализ 3 методов, проверенные на отдельной тестовой выборке, где их качество подтвердилось. Методы были описаны, исследованы и настроены для получения наилучших результатов. Выбран оптимальный для данной задачи алгоритм К-ближайших соседей, который позволяет добиться максимальной точности.

Для данного алгоритма был проведен анализ ошибок и выработана теория о природе ошибок, а также предложение по устранению этих ошибок. На практике этот метод устранения ошибок дал отличные результаты.

В итоге была получена программа, которая позволяет определять вид движения человека с точностью выше 0.998, а также предоставлять графический анализ суточной активности человека.

Таким образом, можно считать, что цель выпускной квалификационной работы выполнена и задачи выполнены полностью. В дальнейшем возможны такие направления развития работы, как увеличение набора данных для повышения точности, добавление в выборку данных от людей различного пола и возраста, повышение точности модели, оптимизация программы для повышения производительности на слабых устройствах и реализация полноценного приложения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Вьюгин В.В. «Элементы математической теории машинного обучения». М.: 2010. - 341 с.
2. Уткин В.Л. Биомеханика физических упражнений. - М.: Просвещение, 1989. - 210 с.
3. Акжолов Р.К. Машинное обучение // Вестник науки. - 2019. - №75. - С. 348-351.
4. Байшев А.В. Характеристики качества данных // Вестник Тувинского государственного университета. Технические и физико-математические науки. - 2023. - №2. - С. 7-13.
5. Бенинг В.Е. О риске оценок, основанных на выборках случайного объема // Вестник Московского университета. Серия 15. Вычислительная математика и кибернетика. - 2020. - №1. - С. 19-29.
6. Донцова Ю.С. Анализ методов бинарной классификации // Известия Самарского научного центра Российской академии наук. - 2014. - №116. - С. 434-438.
7. Золина Е.В. Гамова Н.А. Наивный классификатор Байеса для решения задачи сентимент-анализа тестов // Шаг в науку. - 2019. - №1. - С. 140-142.
8. Магжанова А.Т. Интеграция информационных источников с использованием кластер- анализа по схеме машинного обучения без учителя // Теория и практика современной науки. - 2017. - №101. - С. 1037-1040.
9. Никулин В.Н., Канищев В.С., Багаев И.В. Методы балансировки и нормализации данных для улучшения качества классификации // Компьютерные инструменты в образовании. - 2016. - №3. - С. 16-24.
10. Полетаева Н.Г. Классификация систем машинного обучения // Вестник Балтийского федерального университета им. И. Канта. Серия: Физико-математические и технические науки. - 2020. - №1. - С. 5-22.

11. Сальников А.В., Французов М.С., Виноградов К.А., Пятунин К.Р., Никулин А.С. Верификация и валидация компьютерных моделей // Известия высших учебных заведений. Машиностроение. - 2022. - №6. - С. 100-115.
12. Стрюков Р.К., Шашкин А.И. О модификации метода ближайших соседей // Вестник ВГУ. Серия: Системный анализ и информационные технологии. - 2015. - №1. - С. 114-120.
13. Уланов К.А. Метрики качества данных // Молодой ученый. - 2024. - №20. - С. 17-19.
14. Черкасов Д.Ю. , Иванов В.В Машинное обучение // Наука, техника и образование. - 2018. - №92. - С. 85-87.
15. Шендеров В.А., Китаев Н.Н., Негреева М.Б. Биомеханическая экспертиза: выявление индивидуальных особенностей походки и осанки при идентификации личности // Российский журнал биомеханики. - 2007. - №4. - С. 531-534.
16. Шибанова А.Д. Обучение с подкреплением: введение // Теория и практика современной науки. - 2020. - №101. - С. 477-482.
17. Geurts P., Ernst D., Wehenkel L. Extremely randomized trees. - Liege, Belgium: Springer Science + Business Media, 2006. - 40 p.