МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИИТ)
Кафедра прикладной математики (ПМ)

СЕМЕСТРОВАЯ РАБОТА
по дисциплине
«МЕТОДЫ АНАЛИЗА ДАННЫХ»

Тема: «Распознавание движений человека с помощью акселерометров и методов машинного обучения»

Выполнил Студент группы ИМБО-02-18	«___» _____ 2021 г.	Г.М. Авхименко (подпись и расшифровка подписи)
Принял д.т.н., профессор	«___» _____ 2021 г.	В. И. Кузьмин (подпись и расшифровка подписи)
Семестровая работа представлена к защите	«___» _____ 2021 г.	Г.М. Авхименко (подпись и расшифровка подписи)
	«___» _____ 2021 г.	В. И. Кузьмин (подпись и расшифровка подписи)

Москва 2021

Оглавление

Введение

Распознавание человеческой деятельности - важная, но сложная область исследований с множеством приложений в здравоохранении, управлении техническими системами и системах безопасности. В наше время основные алгоритмы распознавания активности человека базируется на методах машинного обучения. Инженерные решения, полученные таким образом, по качеству не уступают и даже превосходят ранее разработанные методы. Существует несколько техник решения задачи с применением алгоритмов машинного обучения. Техники, основанные на компьютерном зрении, широко используются людьми для отслеживания активности, но они в основном требуют поддержки инфраструктуры, например, установки видеокамер в зоны мониторинга. Еще одним недостатком таких систем является высокая сложность их обучения и настройки. Существует также другой, менее затратный подход. Это распознавание движений человека по сигналам, поступающим с датчиков, которые закреплены на теле самого человека. Также данные могут поступать с носимых человеком устройств (смартфон, Apple Watch и т.д.) [1,2].

Целью данного учебного проекта является построение моделей машинного обучения для распознавания движений человека по данным, поступающим с четырех датчиков. Использовался набор данных (датасет), созданный группой исследователей из университета Рио-де-Жанейро, решавших аналогичную задачу [3]. Был выполнен основной цикл создания модели машинного обучения. Первичный анализ данных, их предобработка (препроцессинг), обучение и оптимизация гиперпараметров моделей, их итоговое тестирование и сравнение результатов. Подготовка данных производилась средствами библиотек Pandas, Matplotlib, Scikit-learn. Обучение и оценка моделей производилась средствами библиотек Scikit-learn, Keras (построение нейронных сетей) [4, 5].

Сбор данных

Набор данных был создан группой исследователей из университета Рио-де-Жанейро, решавших задачу распознавания движения по четырем акселерометрам, закрепленным на человеке [6]. Акселерометр - это прибор, измеряющий проекцию кажущегося ускорения (разности между истинным ускорением объекта и гравитационным ускорением). Его базовая конструкция представляет собой чувствительную массу, закреплённую в упругом подвесе. Отклонение массы от её первоначального положения при наличии кажущегося ускорения несёт информацию о величине этого ускорения. По конструктивному исполнению акселерометры подразделяются на однокомпонентные, двухкомпонентные, трёхкомпонентные. Соответственно, они позволяют измерять проекции кажущегося ускорения на одну, две и три оси. В данном случае использовались трехкомпонентные акселерометры модели ADXL335. Четыре датчика были размещены на каждом испытуемом следующим образом (рис 1).

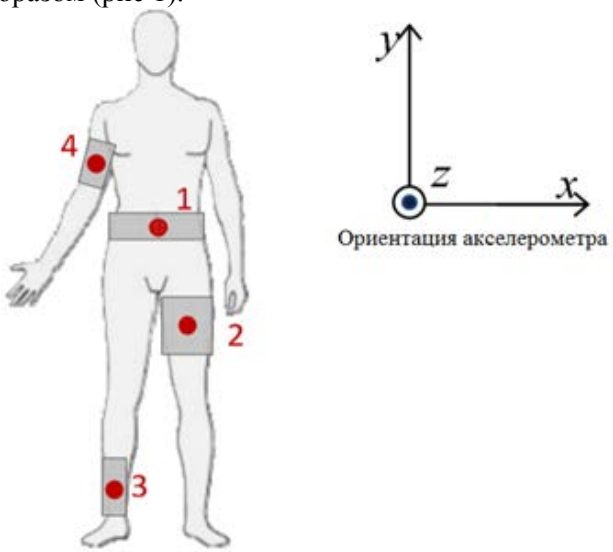

Рис 1 Схема установки и ориентации датчиков на человеке.

Акселерометры были размещены на :

1. Талии
2. Левом бедре
3. Правой щиколодке
4. Правой руке

Калибровка состояла в установке датчиков и выполнения считывания значений, принятых впоследствии за нулевые. После калибровки они вычитались из значений, полученных во время сбора данных. Целью калибровки было смягчить специфические проблемы неточности датчиков этого типа.

В эксперименте приняли участие четверо испытуемых разного пола и возраста:

- Уоллес Юглино (31 год)
- Дебора Кардадор (46 лет)
- Катя Вега (28 лет)
- Хосе Карлос (75 лет)

Сигналы были записаны в течение восьмичасовой активности, по два часа на каждого испытуемого. Каждый из четырёх акселерометров считывал данные восемь раз в секунду. Каждый испытуемый выполнил пять движений:

- Сидение
- Вставание со стула
- Посадка на стул
- Хождение
- Стояние

После этого окончательные результаты были собраны в таблицу, которая является окончательным набором данных и объектом исследований данной работы (табл 1).

	user	gender	age	how_tall_in_meters	weight	body_mass_index	x1	y1	z1	x2	y2	z2	x3	y3	z3	x4	y4	z4	class
0	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.2	-13.0	20.2	-15.1	-12.6	104.3	-89.4	-158.8	-102.7	-142.2	sitting
1	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.2	-13.0	20.2	-15.1	-12.6	104.3	-89.4	-158.9	-102.7	-142.1	sitting
2	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.2	-13.0	20.2	-15.1	-12.6	104.3	-89.4	-158.9	-102.6	-142.1	sitting
3	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.3	-13.0	20.2	-15.1	-12.6	104.3	-89.4	-159.0	-102.6	-142.1	sitting
4	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.3	-13.0	20.2	-15.1	-12.6	104.3	-89.4	-159.1	-102.6	-142.1	sitting
5	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.3	-12.9	20.2	-15.1	-12.6	104.3	-89.4	-159.1	-102.6	-142.1	sitting
6	debora	Woman	46	1,62	75	28,6	-0.7	97.3	-62.3	-12.9	20.2	-15.1	-12.6	104.3	-89.4	-159.2	-102.6	-142.1	sitting
7	debora	Woman	46	1,62	75	28,6	-0.7	97.4	-62.3	-12.9	20.2	-15.1	-12.6	104.3	-89.4	-159.2	-102.6	-142.0	sitting
8	debora	Woman	46	1,62	75	28,6	-0.8	97.4	-62.3	-12.9	20.2	-15.1	-12.6	104.3	-89.4	-159.3	-102.6	-142.0	sitting
9	debora	Woman	46	1,62	75	28,6	-0.8	97.4	-62.4	-12.9	20.2	-15.1	-12.7	104.3	-89.4	-159.4	-102.6	-142.0	sitting
10	debora	Woman	46	1,62	75	28,6	-0.8	97.4	-62.4	-12.9	20.2	-15.1	-12.7	104.3	-89.4	-159.4	-102.5	-142.0	sitting
11	debora	Woman	46	1,62	75	28,6	-0.8	97.4	-62.4	-12.9	20.2	-15.1	-12.7	104.3	-89.4	-159.5	-102.5	-142.0	sitting

Табл 1 Окончательный набор данных.

Описание набора данных

Таблица содержит 19 столбцов и 165633 строки. В таблице 2 содержится полное описание столбцов.

Таблица 2

Название	Единицы измерения	Описание
user	-	Имя испытуемого
gender	-	Пол испытуемого
age	Годы	Возраст испытуемого
how_tall_in_meters	М	Рост
weight	Кг	Вес
body_mass_index	кг/м²	Индекс массы тела
X1 , Y1 , Z1, X2 , Y2 , Z2, X3 , Y3 , Z3, X4 , Y4 , Z4	$м/с^2 * 10^{-3}$	Сигналы (значения трех компонент ускорений) с четырех акселерометров (12 столбцов)
class	-	Тип движения (целевой столбец)

Предбработка и визуализация данных

Перед построением моделей данные нужно подготовить. Нужно проверить наличие пропусков, обработать их, преобразовать признаки, некоторые признаки удалить, добавить новые.

Так же необходима визуализация для понимания природы данных.

Выясним количество объектов (1 измерение датчиков) каждого класса (рис 2).

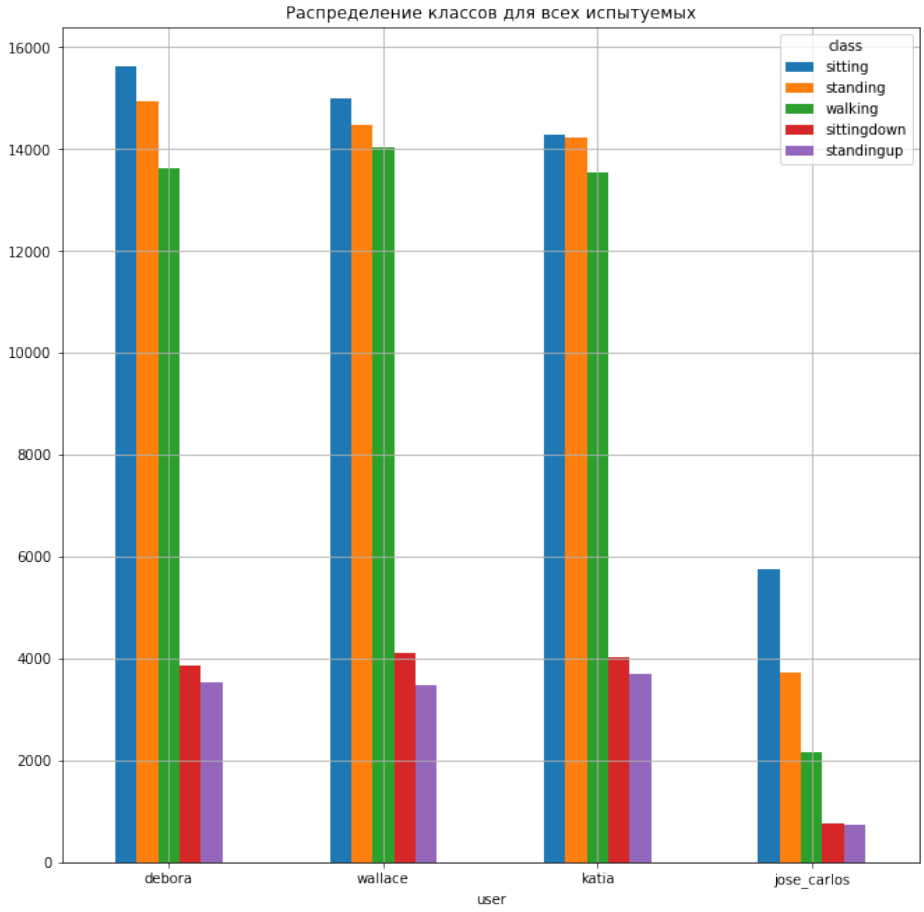

Рис 2 Распределение объектов по классам и испытуемым.

Распределение объектов по классам одинаково для каждого испытуемого. Выборка несбалансированная. Количество объектов каждого класса для каждого испытуемого записано в таблице 3.

Таблица 3

class	sitting	standing	walking	sittingdown	standingup
user					
debora	15615	14940	13622	3853	3547
wallace	14993	14467	14037	4115	3486
katia	14280	14234	13556	4017	3710
jose_carlos	5743	3729	2175	777	737

Нужные признаки для распознавания движения, это показания датчиков. В дальнейшем с ними мы будем работать.

Вот несколько графиков сигналов с акселерометров (рис 3, 4, 5).

Всего было построено 240 графиков (5 классов , 4 испытуемых и у каждого 12 координат). На графиках видно что данные сильно зашумленны. Это мешает решению задачи классификации. Следовательно сигналы нужно очистить от шумов. Для этого было использовано STL-разложение [7] с предварительной идентификацией почти-периодов[8].

Декомпозиция STL – это статистический метод разложения временного ряда на три компонента, содержащие сезонность, тренд и остаток. Сезонность – периодическая компонента. Тренд – задает основную тенденцию временного ряда. Остаток – это непрогнозируемая шумовая составляющая. Нам нужно выбрать тренд из сигнала.

Для выполнения декомпозиции нужно знать период колебаний (сезонность). Очевидно, что имеет место почти-период.

Определение почти-периода: пусть x: R → $\underline{\text{R}}$ — непрерывная функция, а $\varepsilon > 0$. Число τ называется ε-почти периодом функции x, если $||x(t+\tau) - x(t)|| \leq \varepsilon$ при

всех $t \in R$. Функция x называется почти периодической, если для любого $\varepsilon > 0$ найдется такое $L > 0$, что на каждом отрезке длины L функция x имеет ε-почти период.

Для вычисления почти периода нужно применить сдвиговую функцию Джонса.

$$a(\tau) = \frac{1}{n-\tau} * \sum_{t=1}^{n-\tau} |f(t+\tau) - f(t)| \qquad (1)$$

где
t – номер элемента временного ряда
τ- это переменная, которая характеризует почти-период временного ряда
$f(t)$- преобразованное значение временного ряда в точке t

Преобразование нужно для исключения трендовой составляющей, в целях получения достаточно точных резльтатов.
Одно из преобразований для исключения тренда:

$$\ln\left(\frac{y_{t+\Delta t} * y_{t-\Delta t}}{y_t^2}\right) \sim t_{t,\text{п}} \qquad (2)$$

Оптимальным значением Δt является то, при котором отношение среднего значения всех $t_{t,\text{п}}$ к максимальному значению$t_{t,\text{п}}$ минимально (близко к нулю).
Тогда выделение тренда из временного ряда происходит по следующему алгоритму.

1) Выполнить исключение тренда по формуле (2) и выбрать оптимальное Δt. В данной работе $\Delta t = 3$.
2) Построить график сдвиговой функции Джонса (1) и исследовать её локальные минимумы.
3) Выполнить STL-разложение временного ряда для разных значений τ и оценить полученные графики трендовой составляющей.

Выделенный почти-период равен 37.
На рисунках 3, 4 приведены графики сигнала с координаты x1 для испытуемой Деборы, действие – сидеть. Рисунок 3 до исключения шума, рисунок 4 после исключения шума.

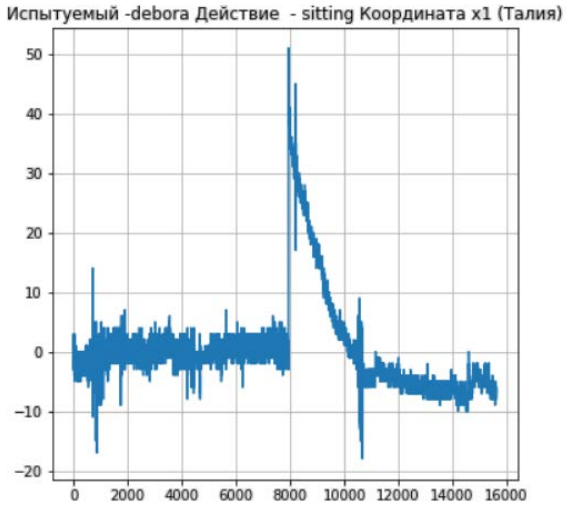

Рис 3 График сигнала до выделения тренда. Рис 4 График сигнала после выделения тренда.
Результат выделения тренда состоит с том, что теперь сигнал очищен от шума. Это значительно улучшит результаты обучения и применения алгоритмов распознавания [9].

Конструирование признаков

Теперь, сигналы очищены, нужно выделить из них определенные признаки, которыми можно описать объект – часть сигнала. В этом случае рассматривают часть временного ряда (несколько последовательных значений, называемые временным окном). Число рассматриваемых значений называют шириной окна. Сначала берут первые n значений, затем сдвигают окно на p значений и берут следующие n отсчетов после p-ого. Величина p – называется перекрытием временного окна. Затем по каждому окну считают различные статистические величины (среднее, стандартное отклонение, коэффицинт эксцесса и т.д). Применительно к данным с акселерометра (считываемые значения – это ускорение) можно также рассчитать и модуль ускорения по трем составляющим и статистические характеристики для него (модуль ускорения – тоже временной ряд). Такая методика конструирования признаков использовалась и подтвердила свою высокую эффективность в работах [1-3, 9].

В данной работе на основе 12 времянных рядов было сконструировано 64 признака. Использовалось временное окно шириной 8 отсчетов (1 секунда записи) с перекрытием в 2 отчета. Описание признаков приведены в таблице 3.

Таблица 3.

Имя колонки в таблице и количество колонок	Описание
x1_mean - z4_mean (12)	Среднее значение окна
x1_Minmax - z4_Minmax (12)	Разность максимального и минимального значений окна
x1_krt - z4_krt (12)	Коэффициент эксцесса окна
x1_Std - z4_Std (12)	Стандартное отклонение окна
a1_mean - a4_mean (4)	Среднее значение модуля ускорения окна
a1_Minmax - a4_Minmax (4)	Разность максимального и минимального значений модуля ускорения окна
a1_krt - a4_krt (4)	Коэффициент эксцесса модуля ускорения окна
a1_Std - a4_Std (4)	Стандартное отклонение модуля ускорения окна

В итоге был сформирован набор данных – таблица с 65 колонками и 57218 записей. Шестьдесят пятая колонка – целевая, обозначает класс движения. Таким образом будет решаться задача классификации. Посмотрим на распределение объектов по классам (рисунок 5). Выборка несбалансированная. Это влияет на оценку качества работы алгоритмов.

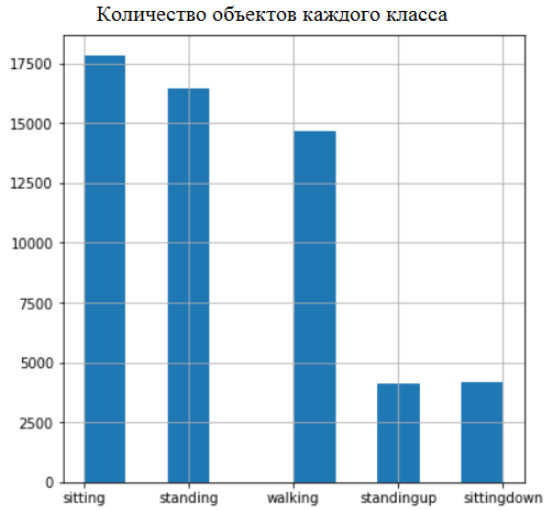

Рис 5 Распределение объектов по классам

Обучение моделей и отбор признаков

Нужно разделить выборку на 3 части в следующем процентном отношении:

- Обучающая (60%) – для обучения моделей и оптимизации их гиперпараметров (34330 объектов).
- Валидационная (20%) – для промежутого тестирования моделей и оптимизации гиперпараметров некоторых. Оптимизация на этой выборке не желательна, так как модель подстроится под неё и покажет худший результат на других данных (11444 объектов).
- Тестовая (20%)– требуется только для финальной оценки моделей и выбора наилучшей (11444 объектов).

Для оценки качества классификации было использована F-мера с микроусреднением.
Смысл F-меры.
Пусть имеется квадратная матрица ошибок размера N, где N равно количеству классов.
Числа на главной диагонали показывают количество правильно распознанных объектов.
Для определения F-меры дадим определения метрикам "точность" и "полнота" (Precision и Recall). Точность равняется отношению соответствующего диагонального элемента матрицы и суммы всей строки класса. Полнота – отношению диагонального элемента матрицы и суммы всего столбца класса.

$$Precision_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{c,i}} \quad (3)$$

$$Recall_c = \frac{A_{c,c}}{\sum_{i=1}^{n} A_{i,c}} \quad (4)$$

Результирующая точность классификатора рассчитывается как арифметическое среднее его точности по всем классам. То же самое с полнотой. Технически этот подход называется macro-averaging.

$$\mathrm{F} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (5)$$

F-мера представляет собой гармоническое среднее между точностью и полнотой. Она стремится к нулю, если точность или полнота стремится к нулю. Данная формула придает одинаковый вес точности и полноте, поэтому F-мера будет понижаться одинаково при уменьшении и точности и полноты. Таким образом точность распознавания объектов из малочисленных классов влияет на F-меру [10].

Подбор моделей начинается с простых – линейных классификаторов.
А именно softmax-регрессии.

Помимо обучения нужно провести оптимизацию гиперпараметров: коэффициента регуляризации. Был использован L_2 − регуляризатор. Он "штрафует" вектор весов за большую величину нормы.
Множество для поиска оптимального коэффициента регуляризации: от 0.5 до 9.5 с шагом 0.5. Поиск выполнялся при помощи перекрестной проверки (кросс-валидации) по 5 блокам. Найденное оптимальное значение равно - 9. Оценка на тестовой выборке - 0.9994757. Такой высокий результат означает хорошую линейную разделимость классов в данном пространстве признаков. Теперь нужно попробовать сократить количество признаков. Выберем определенные признаки, обучим на них линейную модель и сравним оценку качества на валидационной выборке. Рассмотренные варианты пространств признаков (имена взятых столбцов) приведены в таблице 4.

Таблица 4

Номер варианта	Названия столбцов	Описание
1	'x1_mean', 'y1_mean', 'z1_mean', 'x2_mean', 'y2_mean', 'z2_mean','x3_mean', 'y3_mean', 'z3_mean', 'x4_mean', 'y4_mean', 'z4_mean', 'x1_Std', 'y1_Std', 'z1_Std', 'x2_Std', 'y2_Std', 'z2_Std', 'x3_Std', 'y3_Std', 'z3_Std', 'x4_Std', 'y4_Std', 'z4_Std', 'a1_mean','a2_mean', 'a3_mean', 'a4_mean', 'a1_Std', 'a2_Std', 'a3_Std', 'a4_Std'	Среднее значение по 12 координатам (12) Стандартное отклонение по 12 координатам (12) Среднее значение модуля ускорения по 4 акселерометрам (4) Стандартное отклонение модуля ускорения по 4 акселерометрам (4)
2	'x1_mean', 'y1_mean', 'z1_mean', 'x2_mean', 'y2_mean', 'z2_mean','x3_mean', 'y3_mean', 'z3_mean', 'x4_mean', 'y4_mean', 'z4_mean','a1_mean','a2_mean', 'a3_mean', 'a4_mean'	Среднее значение по 12 координатам (12) Среднее значение модуля ускорения по 4 акселерометрам (4)
3	'x1_mean', 'y1_mean', 'z1_mean', 'x2_mean', 'y2_mean', 'z2_mean','x3_mean', 'y3_mean', 'z3_mean', 'x4_mean', 'y4_mean', 'z4_mean'	Среднее значение по 12 координатам (12)
4	'a1_mean','a2_mean', 'a3_mean', 'a4_mean'	Среднее значение модуля ускорения по 4 акселерометрам (4)

Оценки для вариантов на валидационной выборке приведены в таблице 5.

Таблица 5

Номер варианта	Оценка на валидационной выборке
1	0.9993883
2	0.9974659
3	0.9965047
4	0.7991960

Из таблицы 5 видно, что при снижении количества признаков объекты можно линейно разделить. И только при 4 признаках достигается худшая линейная разделимость.
В таблице 6 приведена оценка качества классификации на тестовой выборке для пространств 3 и 4.

Таблица 6

Номер варианта	Оценка на тестовой выборке
3	0.9969416
4	0.8036525

Далее модели машинного обучения будут строиться и оцениваться только в этих пространствах признаков.

Для корректной работы алгоритмов машинного обучения данные нужно масштабировать двумя способами (для каждого алгоритма свой способ):

- Нормализация – линейное преобразование данных в диапазоне [0..1], где минимальное и максимальное масштабируемые значения соответствуют 0 и 1.
- Стандартизация - данных на основе среднего значения и стандартного отклонения: деление разницы между переменной и средним значением на стандартное отклонение.

В таблице 7 приведено, какое преобразование данных было применено для каждого алгоритма

Таблица 7

Вид преобразования	Алгоритм
Стандартизация	Softmax – регрессия
Стандартизация	Решающее дерево
Стандартизация	Случайный лес
Стандартизация	Градиентный бустинг над решающими деревьями
Нормализация	Метод K ближайших соседей
Нормализация	Метод K ближайших соседей со взвешенной метрикой
Нормализация	Нейронная сеть

Построение и выбор модели в пространстве из 12 признаков

Пространство признаков определено (вариант 3 в таблице 5). Выборка разделена на три части: тренировочную, тестовую и валидационную. Нужно подобрать модель которая покажет максимальный результат на тестовой выборке. Для обучения и оптимизации гиперпараметров использовалось тренировочная выборка. Валидационная выборка использовалось для оценки промежуточных результатов классификации. Оптимизация гиперпараметров методом решетчатого поиска оценкой на перекрестной проверке по 5 блокам. Наименования моделей, пространство поиска оптимальных гиперпараметров и лучшие их комбинации приведены в таблице 7.

Таблица 8

Наименование модели	Пространство поиска гиперпараметров	Лучшая комбинация параметров	Описание параметров
Softmax – регрессия	C: с 1 по 15 с шагом 0.25	14.75	Коэффициент регуляризации.
Метод K ближайших соседей	p: 2, 3, 4 n_neighbors: с 2 по 20	p = 2 n_neighbors = 2	n_neighbors - количество ближайших соседей p - параметр метрики Минковского
Метод K ближайших соседей со взвешенной метрикой	Двадцать пять векторов на 12 элементов. Их элементы - случайные числа от 0.1 до 3 с одним знаком после запятой.	Вектор с координатами: 1.82, 2.89, 2.47, 2.07, 0.96, 2.01, 1.63, 2.53, 2.94, 1.62, 1.29, 1.75. p = 2 n_neighbors = 2	Координаты вектора - это веса признаков во взвешенной метрике Минковского. Остальные параметры остались те же что и в предыдущем методе ближайших соседей.
Нейронная сеть	Количество скрытых слоев и нейронов в них. Методы оптимизации и их параметры. Конкретные значения указанные в таблице 8. Подбор параметров производился на валидационной выборке.	Число нейронов в слое, начиная с входного: 12, 10, 8, 5. Метод оптимизации – Adam с базовыми настройками. Функции активации: ReLU, sigmoid, softmax.	
Решающее дерево	'criterion': 'gini', 'entropy' 'max_depth' : 6,8,10,12, 14, 16, 18,20 'min_samples_split': 30,40,50,60,70,80,90 'min_samples_leaf' : 2,10,20,30,40,50 'random_state': 48,56,78,91,112,134,145,167	criterion: gini, max_depth: 20, min_samples_leaf: 2, min_samples_split: 30 random_state: 167	Criterion – Функция измерения качества разделения. max_depth – максимальная глубина дерева min_samples_leaf – минимальное количество

			выборок, необходимое для разделения во внутреннем узле min_samples_split – минимальное количество выборок, которое требуется для конечного узла random_state - параметр генератора случайных чисел. Генератор отвечает за случайное разделение выборки в процессе обучения.
Беггинг над решающими деревьями	n_estimators: 2,3,4,5 max_samples : 0.6 , 0.65 , 0.7 , 0.75 , 0.8 , 0.85 max_features: 0.65 , 0.6 , 0.75 , 0.8 oob_score : True ,False random_state' : 25,34,41,56,78	max_features: 0.75 max_samples: 0.8 n_estimators: 5 oob_score: True random_state: 34	max_features - число признаков, используемое для обучения модели max_samples - число образцов, используемое для обучения модели n_estimators – число моделей в ансамбле oob_score - random_state: - параметр генератора случайных чисел. Генератор отвечает за случайное разделение выборки в процессе обучения.

Об обучении нейронных сетей. Обучение производилась в течение 1700 эпох. Размер пакета - 300 образцов. Поскольку тренировочная выборка была несбалансированная, то для достижения хорошего качества работы количество классов нужно уравнять. То есть произвести оверсемплинг выборки. Количество объектов всех классов сделать одинаковым, просто сделав несколько копий. Класс, имеющий максимальное число объектов: sitting. Количество образцов в этом классе - 10685. После оверсемплинга число объектов в обучающей выборке стало равно 53425. После этого нейронные сети были обучены на этой выборке, затем протестированы на валидационной выборке. Описание вариантов нейронных сетей приведено в таблице 8. Входной слой во всех вариантах содержал 12 нейронов, а выходной – 5 нейронов. Функция активации выходного слоя – softmax-функция.

Таблица 9

Число нейронов в скрытых слоях	Функции активации скрытых слоев	Метод оптимизации
9, 7	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
10, 7	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
9, 8	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
10, 8	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
10 , 8, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
9 , 8, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
10, 7, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.01, момент – 0.9
9, 7	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
10, 7	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
9, 8	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
10, 8	'relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
10 , 8, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
9 , 8, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
10, 7, 6	'relu','relu','sigmoid'	Стохастический градиентный спуск (модификация Нестерова), скорость обучения – 0.125, момент – 0.9
9, 7	'relu','sigmoid'	Adam
10, 7	'relu','sigmoid'	Adam
9, 8	'relu','sigmoid'	Adam
10, 8	'relu','sigmoid'	Adam
10 , 8, 6	'relu','relu','sigmoid'	Adam
9 , 8, 6	'relu','relu','sigmoid'	Adam

10, 7, 6	'relu','relu','sigmoid'	Adam

Параметры лучшей модели:

- Число нейронов в скрытых слоях: 10, 8.
- Функции активации скрытых слоёв: 'relu','sigmoid'.
- Метод оптимизации: Adam.

Результаты работы моделей на тестовой и валидационной выборке приведены в таблице 9.

Таблица 10

Наименование модели	Результат на валидационной выборке	Результат на тестовой выборке
Softmax – регрессия	0.9970290	0.9971163
Метод K ближайших соседей	0.9999126	0.9998252
Метод K ближайших соседей со взвешенной метрикой	1.0	0.9999126
Нейронная сеть	0.9866305	0.9866305
Решающее дерево	0.9965047	0.9961551
Беггинг над решающими деревьями	0.9989514	0.9989514

Вывод: лучшая модель для пространства из двенадцати признаков - Метод K ближайших соседей со взвешенной метрикой.

Построение и выбор модели в пространстве из 4 признаков

Пространство признаков определено (вариант 3 в таблице 5). Выборка разделена на три части: тренировочную, тестовую и валидационную. Для обучения и оптимизации гиперпараметров использовалось тренировочная выборка. Валидационная выборка использовалось для оценки промежуточных результатов классификации. Оптимизация гиперпараметров методом решетчатого поиска оценкой на перекрестной проверке по 5 блокам. Наименования моделей, пространство поиска оптимальных гиперпараметров и лучшие их комбинации приведены в таблице 10.

Таблица 11

Наименование модели	Пространство поиска гиперпараметров	Лучшая комбинация параметров	Описание параметров
Softmax – регрессия	C: с 1 по 15 с шагом 0.25	11.25	Коэффициент регуляризации.
Метод K ближайших соседей	p: 2, 3, 4 n_neighbors: с 2 по 20	p = 2 n_neighbors = 2	n_neighbors - количество ближайших соседей p - параметр метрики Минковского
Метод K ближайших соседей со взвешенной метрикой	Двадцать пять векторов на 4 элементов. Их элементы - случайные числа от 0.1 до 3 с одним знаком после запятой.	Вектор с координатами: 2.38, 2.86, 2.36, 0.73. p = 2 n_neighbors = 2	Координаты вектора - это веса признаков во взвешенной метрике Минковского. Остальные параметры остались те же что и в предыдущем методе ближайших соседей.
Решающее дерево	criterion: 'gini', 'entropy' max_depth: 6,8,10,12, 14, 16, 18,20 min_samples_split: 30, 40, 50, 60, 70 min_samples_leaf: 2, 10, 20, 30, 40 random_state: 48,56,78,91,112, 134 max_features: 2,3,4	criterion: 'entropy', max_depth: 20, max_features: 4, min_samples_leaf: 2, min_samples_split: 30, random_state: 91	Criterion – Функция измерения качества разделения. max_depth – максимальная глубина дерева min_samples_leaf – минимальное количество выборок, которое требуется для конечного узла min_samples_split – минимальное количество выборок, которое требуется для конечного узла

			random_state - параметр генератора случайных чисел. Генератор отвечает за случайное разделение выборки в процессе обучения.
Случайный лес	n_estimators: 5, 10, 15, 20, 25, 30, 35, 40 criterion: gini, entropy max_depth: 4, 6, 8, 10, 12 min_samples_leaf: 2, 4, 6, 8, 10, 12, 16, 18, 20 min_samples_split: 2, 4, 6, 8, 10, 12, 16, 18, 20 max_samples: 0.7, 0.85	n_estimators: 35 criterion: gini max_depth: 12 min_samples_leaf: 2 min_samples_split: 8 max_samples: 0.85	criterion - функция измерения качества разделения max_depth – максимальная глубина дерева min_samples_leaf - минимальное количество выборок, которое требуется для конечного узла min_samples_split - минимальное количество выборок, необходимое для разделения во внутреннем узле max_features - число признаков, используемое для обучения модели max_samples - число образцов, используемое для обучения модели n_estimators – число моделей в ансамбле
Градиентный бустинг над решающими деревьями	max_depth: 2, 3, 5 min_samples_split: 20, 30, 40, 50, 60 min_samples_leaf: 3, 8, 10, 20, 25	max_depth: 5 min_samples_leaf: 20 min_samples_split: 50	max_depth - максимальная глубина дерева min_samples_leaf - минимальное количество выборок, которое требуется для конечного узла min_samples_split - минимальное количество

			выборок, необходимое для разделения во внутреннем узле
Беггинг над методом К ближайших соседей (взвешенная метрика)	n_estimators: 2,3,4 max_samples: 0.75 , 0.8 , 0.85	n_estimators: 4 max_samples: 0.85	max_samples - число образцов, используемое для обучения модели n_estimators – число моделей в ансамбле
Голосующий классификатор (случайный лес, градиентный бустинг над решающими деревьями)	Случайный лес, Градиентный бустинг над решающими деревьями – их параметры были подобраны ранее (соответствующие модели в таблице)		
Голосующий классификатор (Метод К ближайших соседей со взвешенной метрикой , Беггинг над методом К ближайших соседей (взвешенная метрика))	Метод К ближайших соседей со взвешенной метрикой , Беггинг над методом К ближайших соседей (взвешенная метрика) – их параметры были подобраны ранее (соответствующие модели в таблице)		

Результаты работы моделей на тестовой и валидационной выборке приведены в таблице 9.

Таблица 12

Наименование модели	Результат на валидационной выборке	Результат на тестовой выборке
Softmax – регрессия	0.7992834	0.8037399
Метод К ближайших соседей	0.9950192	0.9947570
Метод К ближайших соседей со взвешенной метрикой	0.9968542	0.9960678
Решающее дерево	0.9815623	0.9800768
Случайный лес	0.9940580	0.9930968
Градиентный бустинг над решающими деревьями	0.9955435	0.9945823
Беггинг над методом К ближайших соседей (взвешенная метрика)	0.9958930	0.9945823
Голосующий классификатор (случайный лес, градиентный бустинг над решающими	0.9952813	0.9940580

деревьями)		
Голосующий классификатор (Метод К ближайших соседей со взвешенной метрикой , Беггинг над методом К ближайших соседей (взвешенная метрика))	0.9962425	0.9955435

Вывод: лучшая модель для пространства из четырёх признаков - метод К ближайших соседей со взвешенной метрикой

Выводы

Задача распознавания движений человека по показаниям с акселерометра разрешима методами машинного обучения. Инженерные решения такого типа показывают очень высокое качество работы на реальных данных. Эти методы легко реализовать и использовать на практике. Они показывают хорошую эффективность по времени и по памяти. В данном проекте лучше всего показала себя модель: Метод K ближайших со взвешенной метрикой. Оптимальное количество признаков для классификации равно 12 (см. таблицу 4, номер варианта 3). В данном пространстве объекты хорошо линейно разделимы. Также в этом пространстве отлично работает модель: softmax-регрессия. Её качество классификации чуть хуже, чем метод K ближайших соседей, но скорость её работы выше.

Список информационных источников

1. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones
2. Human Activity Recognition: Accelerometers Unveil Your Actions
3. Wearable Computing: Accelerometers' Data Classification of Body Postures and Movements
4. https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
5. https://keras.io/
6. http://groupware.les.inf.puc-rio.br/har#dataset
7. http://machinelearning.ru/wiki/images/9/9f/Ps_ts_ets.pdf
8. Кузьмин В. И., Гадзаов А. Ф. - Методы построения моделей по эмпирическим данным
9. Accelerometer signal pre-processing influence on human activity recognition
10. http://bazhenov.me/blog/2012/07/21/classification-performance-evaluation.html