МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ «МИСиС»

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И КОМПЬЮТЕРНЫХ НАУК

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

НАПРАВЛЕНИЕ 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

на тему: «Разработка модели идентификации целенаправленной физической активности человека при выполнении производственных работ»

Студент _____ *Кириллов Павел Сергеевич*

Руководитель работы _____ *Дерябин Сергей Андреевич*

Нормоконтроль проведен _____ *Агабубаев А.*

Проверка на заимствования проведена _____ *Давыденко А.А.*

Работа рассмотрена кафедрой и допущена к защите в ГЭК

Заведующий кафедрой _____ *Темкин И.О.*

Директор института _____ *Солодов С.В.*

Москва, 2022

АННОТАЦИЯ

В данной работе решается задача разработки средства автоматизации мониторинга целенаправленной физической активности людей на основе собираемой телеметрии о их действиях в целях повышения эффективности процессов управления производством.

В ходе работы был проведено исследование предметной области, обзор существующих способов анализа, обработка данных и построение моделей классификации целенаправленной физической активности. Были построена модель функционирования процесса управления целенаправленной физической активность, а также строились модели подпроцесса мониторинга функционирования в настоящий момент (AS-IS) и при внедрении предложенного решения (TO-BE). Также была показана информационная модель решения задачи и разработан алгоритм программной реализации в общем виде.

В результате работы было построено программное решение, которое позволило идентифицировать производственную активность рабочих посредством различных методов обработки телеметрии и настройки модели машинного обучения.

Работа содержит: 83 страницы, 36 рисунков, 12 таблиц, 27 источников

СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

BoxPlot – график визуализации статистических свойств данных.

DHCP - протокол динамической настройки узла

DMZ - демилитаризованная зона - сегмент сети, содержащий общедоступные сервисы и отделяющий их от частных

DNS - компьютерная распределённая система для получения информации о доменах

DPI - количество пискелей на дюйм

DT – алгоритм дерева решений

FDR – false discovery rate

FPS - количество сменяемых кадров за единицу времени

GB – гигабайт, единица измерения количества информации

GNU – аббревиатура “GNU's Not UNIX”, окружение ядра Linux

HDD - запоминающее устройство произвольного доступа

IEEE - международная ассоциация специалистов, занимающаяся разработкой стандартов по информационным технологиям

IQR – межквартильный размах

KNN – алгоритм k-ближайших соседей

LogReg – алгоритм логистической регрессии

NAT - механизм в сетях, позволяющий преобразовывать адреса транзитных пакетов

NN – нейросетевой алгоритм

OS – операционная система

Q-Q plot – график сравнения квантилей наблюдаемого распределения с ожидаемым

RF – алгоритм случайного леса

SQL – язык программирования для управления базами данных

SVM – метод опорных векторов

ТВ – терабайт, единица измерения количества информации

VLAN - виртуальная локальная компьютерная сеть

ГГЦ – гигагерц единица измерения частоты периодических процессов

Гц – герц, единица частоты периодических процессов

НТИ – научно-технические источники

ОЗУ – оперативное запоминающее устройство

НИР – научно-исследовательские работы

ПО – программное обеспечение

ВВЕДЕНИЕ

В современном мире крайне сильна конкуренция между компаниями в различных экономических областях, в которых они функционируют. Для соответствия рыночным стандартам компаниям требуется увеличивать количество выпускаемой продукции и уменьшать издержки производства. Достижению этих целей способствует модернизация внутренних процессов компании посредством средств автоматизации. Их внедрение ведет к уменьшению необходимого штата сотрудников и к увеличению производительности рабочих процессов.

Рассмотрим процесс управления целенаправленной физической активностью сотрудников. От него зависит рабочая деятельность, предполагающая участие человека, например: контроль качества изделий, разработка производственного плана работ, его декомпозиция на задачи, выполнение производственных работ и пр. Хорошая реализация процесса управления целенаправленной физической активностью в компании позволяет уменьшить время и затраты на рабочую деятельность с сохранением или увеличением качества итогового продукта.

Для производства продукции высокого качества необходимо контролировать этапы рабочего процесса сотрудников, работающих по технологическим заданиям (например, строителей). Этот контроль осуществляется посредством мониторинга целенаправленной физической активности. Он строится на анализе производственной активности рабочих, посредством которой вводятся корректировки в рабочий процесс управляющими низшего звена (например, прорабами). В случае неэффективной работы процесса мониторинга рабочие могут сильно отдаляться от поставленного производственного плана. Более того, с ростом количества сотрудников уровня выполнения увеличивается потребность в людях, осуществляющих их контроль, что ведет к новым тратам ресурсов. Для уменьшения трат количество контролирующего персонала сокращают при помощи средств автоматизации: стационарные, портативные камеры, умных-часов. Телеметрия с этих средств может использоваться для идентификации рабочей активности управляющими низшего звена, однако это неэффективно, поскольку этот процесс занимает много рабочего времени. Для решения этой задачи были рассмотрены программные средства, которые при интегрировании в рабочий процесс способны обеспечить автономию процесса распознавания целенаправленной физической активности рабочих уровня исполнения.

Программное решение основано на современных исследованиях на стыке математики и информатики, которые определили класс алгоритмов (алгоритмов машинного обучения),

способных распознавать производственную активность рабочих на основе поиска закономерностей в данных, считанных с аппаратного решения.

Современные исследования позволяют говорить об эффективности применения алгоритмов машинного обучения в процессе управления целенаправленной физической работой посредством их интеграции в подпроцесс мониторинга рабочей активности сотрудников [1-7]. Хоть эта тема не является новой, однако, в виду научного прогресса, такие алгоритмы стали гораздо эффективней, чем раньше, а благодаря современным компьютерам, их обучение не является столь долгим. Кроме того, задача классификации целенаправленной физической активности является крайне актуальной сама по себе и решается в различных сферах деятельности. Этот факт отражается в количестве научных работ по данной теме в процессах, отличных от управления целенаправленной физической работой людей [8-21].

Обзор научных работ в данной и смежных областях позволил сузить круг поиска наилучших способов построения программного решения. В результате работы было определены средства обработки данных и построения модели машинного обучения, интеграция которых в систему мониторинга позволит сократить время распознавания действий человека и уменьшить расходы на специализированный персонал.

1 Обзор научно-технических источников информации

Для успешного применения алгоритмов машинного обучения и способов обработки данных для распознавания деятельности рабочих требуется рассмотреть работы, в которых решалась схожая задача для того, чтобы понимать специфику предметной области более полно. Если коснуться сфер, в которых такие технологии применяются, то можно рассмотреть медицинскую область. Она примечательна тем, что в ней алгоритмы машинного обучения применяются достаточно давно и в настоящий момент ведутся широкие исследования применения таких алгоритмов для классификации целенаправленной активности человека [11, 21], что говорит о высоком уровне доверия к алгоритмам машинного обучения. Кроме того, требуется выделить возможные способы построения системы распознавания, для которой требуются: способы анализа и обработки данных, выбор моделей машинного обучения и метрик для оценки результатов обучения.

Рассмотрим способы решения задачи использования алгоритмов классификации физической активности для мониторинга деятельности рабочих:

Таблица 1 - НИР, в которых применялись алгоритмы машинного обучения для распознавания деятельности рабочих.

Индекс в списке источников	Область исследования / проблема	Цель исследования	Метод(ы) решения	Результаты
1.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Тестирование алгоритмов для распознавания по видео инструментов, которые находятся в руках рабочих в конкретные момент времени.	Видеокамеры, комплекс классификационных нейросетевых алгоритмов для распознавания инструментов в руках рабочих, предварительная обработка видеоданных, метрики качества.	Использованные алгоритмы показали высокую точность определения инструментов в руках человека, что позволяет строить дальнейшие модели классификации производственной деятельности.

Продолжение таблицы 1

Индекс в списке литературы	Область исследования / проблема	Цель исследования	Метод(ы) решения	Результаты
2.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Автоматизация построения системы мониторинга активности рабочих.	Носимые датчики, обработка и нормализация данных, классификационные алгоритмы машинного обучения, метрики качества.	Во время тестирования модели получилось достигнуть точности в 88%
3.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Рассмотрение возможностей машинного обучения для мониторинга активности экскаватора.	Android телефон с акселерометром и гироскопом, закрепленный в кабине экскаватора, методы обработки данных, классификационные алгоритмы машинного обучения, метрики качества.	Нейронная сеть показала себя хуже всего (53.29%), лучше всего показали себя KNN (73.61%) и дерево решений (73.47%).
4.	Анализ устройств мониторинга и способы обработки поступающих с них данных	Рассмотрение возможностей использования переносных гироскопов для мониторинга рабочей активности	3 типа гироскопов, закрепленных на рабочих, алгоритмы обработки данных.	Акселерометр можно использовать в строительных работах, данные поступают непрерывно.

Продолжение таблицы 1

Индекс в списке литературы	Область исследования / проблема	Цель исследования	Метод(ы) решения	Результаты
5.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Автоматизация построения системы мониторинга активности рабочих.	Обработка данных, метрики качества, алгоритмы машинного обучения: NN, DT, KNN, logreg, SVM	Лучше всего оказался нейросетевой алгоритм, после которого идет дерево решений
6.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Автоматизация построения системы мониторинга активности рабочих.	Трехосевой наручный датчик, обработка данных, алгоритмы машинного обучения: logreg, SVM, DT и KNN	SVM оказался лучше всего с точность в 90%, в logreg и DT точность составляла примерно 87%
7.	Интеграция алгоритмов машинного обучения в мониторинг активности рабочих.	Сравнение всевозможных способов мониторинга активности рабочих	Алгоритмы машинного обучения, способы считывания данных и их обработки	Не существует универсального способа построения модели распознавания. Все зависит от структуры данных

На сегодняшний момент наиболее распространены два подхода к распознаванию активности рабочих:

1) Подходы, основанные на считывании видео активности (vision-based) [1]
2) Подходы, основанные на считывании двигательной активности (kinematic-based), например, работы: [2, 3, 4, 5, 6]

Vision-based методы основываются на алгоритмах computer-vision и используют для считывания информации видеокамеру. Несмотря на кажущуюся легкость интеграции

подобных методов, данное семейство решений имеет существенные недостатки. Использование видеокамер влечет за собой технические проблемы, ограничивающие надежность и практичность моделей: камеры чувствительны к факторам окружающей среды (таким как пыль, снег, дождь, туман), плохо работают в темноте или под прямыми солнечными лучами и угол обзора этих устройств зачастую не охватывает все рабочее пространство, что требует установки дополнительных видеокамер и влечет за собой новые расходы. Более того, алгоритмы computer vision могут не справиться в случае большого количества информации, поступающей с видеокамер, так как это ведет к обработке большого количества зашумленных данных. Также такие системы требуют хранения большого памяти для обработки изображений.[2]

Kinematic-based методы считывания используют физическую информацию об активности рабочих за счет использования акселерометров и гироскопов в носимых устройствах. Такие методы представляются крайне гибкими и надежными. Для считывания данных могут использоваться:

1) Носимые кистевые устройства (например, умные часы)
2) Смартфоны
3) Специальные системы, носимые человеком.

Рисунок 1 - пример носимой системы, разработанной в работе [10].

Kinematic-based методы являются самыми популярными методам распознавания активности на сегодняшний день, так как являются очень эффективными и подавляющее число научных работ по данной тематике используют эти методы. При их использовании ставится задача разработки классического классификатора целенаправленной активности рабочих методами машинного обучения, вместо использования крайне сложных нейросетевых алгоритмов распознавания computer vision.

В работах для анализа данных используются: различные квантили [25], qq-plot [24], boxplot [24, 25], тест Шапиро-Уилка [27], корреляционный анализ [26]. Такие статистические показатели как математическое ожидание, оценки стандартного отклонения и дисперсии используется повсеместно практически во всех работах, предполагающих анализ данных.

Для корреляционного анализа можно использовать как критерий Пирсона, так и критерий Спирмена. Ведь, согласно работе [26], не совсем верно утверждать, что критерий Пирсона работает только с нормально распределенными данными. Однако, он определенно плохо работает с данными, содержащими выбросы, так что критерий Спирмена уместнее в задаче более всего, хоть он и слабее критерия Пирсона в общем случае.

Для обработки выбросов можно использовать boxplot принцип [25], предполагающий использование IQR для удаления аномалий в данных, так как теоретически это способно повысить стабильность модели. Однако, для успешного использования этого принципа необходимо условие нормальности данных, для доказательства которого рекомендуется, согласно работе [24], использование Q-Q графика для визуального и теста Шапиро-Уилка для критериального тестирования [27]. Другой пример использования IQR представлен в работе [27], где он используется для визуализации данных.

Касательно алгоритмов машинного обучения можно сделать вывод, что разные модели актуальны для разных данных и выбрать универсальную нельзя. Более того, в рассмотренных источниках слабо представлены ансамблевые алгоритмы машинного обучения, такие как RF (случайный лес) или реализации градиентного бустинга. Если говорить об алгоритме RF, то он представляется более эффективным, чем DT, который, хоть и имеет такие достоинства как простота интерпретации и высокая скорость обучения, но является достаточно слабым алгоритмом из-за сильного переобучения на обучающей выборке. В пользу RF говорит работа [11], в которой решалась задача распознавания целенаправленной физической активности. Согласно ей, алгоритм RF обладает следующими свойствами:

- стабильное качество результатов обучения в задаче распознавания;
- не требует предварительной нормализации данных, поскольку не является метрическим алгоритмом;
- крайне эффективен без нужды в настройке гиперпараметров;
- устойчив к выбросам в данных, что делает его пригодным для работы со считывающими устройствами, которые могут давать отклонения при передаче;

Согласно работе [20], применительно к данной задаче, лучшими алгоритмом машинного обучения является RF из класса ансамблевых алгоритмов, после которого идет SVM из класса одиночных алгоритмов.

В работе [18], в которой решается задача, аналогичная той, которая ставилась в статье [20], ситуация схожая: алгоритмы SVM и Bagged Trees (алгоритм, схожий с RF) показали наилучшую точность классификации из выбранного списка. Стоит отметить, что для алгоритма SVM наилучшим образом показало себя полиномиальное ядро третьей степени. Однако, стоит отметить, что SVM - метрический алгоритм и требует нормализации данных, что является его минусом. Отдельно интересно рассмотреть алгоритм градиентного бустинга, поскольку он потенциально способен улучшить результаты обучения RF и SVM, особенно интересна улучшенная его реализация в виде CatBoost классификатора, которая является одной из самых мощных и быстрых на сегодняшний день [23]. Существенным минусом является сложность ее реализации, которая ведет к затруднению в интерпретации работы алгоритма для третьих лиц. Однако, этот алгоритм активно применяется в Европейской организация по ядерным исследованиям “Cern”, что характеризует его как надежный метод машинного обучения.

Отдельно интересно рассмотреть вопрос конструирования данных: во всех рассмотренных работах, где ставится задача разработки модели машинного обучения для классификации целенаправленной физической деятельности исходные признаки для предсказания (предикторы), преобразуются в новые признаки, представляющие собой статистические выкладки с информацией об интервалах в данных. Этот интервал может быть как скользящим, так и определенным [2, 10, 11]. В случае скользящего интервала размерность матрицы данных не меняется, поскольку новые предикторы перезаписывают исходные методом скользящего окна, однако, в силу появления новых признаков вероятность нахождения случайных взаимосвязей в данных возрастает. В случае фиксированного интервала количество точек данных уменьшается, так как данные агрегируются по некоторому фиксированному числу наблюдений, но полезная информация увеличивается, так появляются новых признаки и уменьшается количество шумов. Размер этих интервалов зависит от особенностей изначальных данных. Пример сконструированных признаков из работы [18]: стандартное отклонение, асимметрия, минимум, максимум, среднее, медианное абсолютное отклонение, эксцесс, межквартильный диапазон, энтропия, средняя частота, диапазон значений между минимум и максимумом. Для обработки зависимого столбца используется мода [10]. Метрики для теста алгоритмов машинного обучения в большинстве работ одинаковы: accuracy, precision, recall, f1-score. В дополнение возможно использовать out-of-bag score для RF, как это сделано в работе [10].

1.1 Метрики качества алгоритмов:

В данной пункте приведены наиболее важные и используемые метрики качества в задачах классификации. Отдельно выделяется oob-score - метрика, которая актуально только для случайного леса, она может оказаться полезна для подбора гиперпараметров случайного леса [10].

1.1.1 Accuracy

$$accuracy(\alpha, X) = \frac{1}{l}\sum_{i=1}^{l}[\alpha(x_i) = y_i] \ , \quad (1)$$

где α - алгоритм машинного обучения,

$\alpha(x_i)$ - предсказание на i-ом объекте данных,

y_i - истинное значение на i-ом объекте,

[выражение] - скобки Айверсона (число 0, если условие ложно и 1, если истинно, l - длина выборки)

1.1.2 Precision

$$precision(\alpha, X) = \frac{TP}{TP + FP} \ , \quad (2)$$

где TP - верно классифицируемые как положительный класс,

FP - ложно классифицируемые как положительный класс

1.1.3 Recall

$$recall(\alpha, X) = \frac{TP}{TP + FN} \ , \quad (3)$$

где TP - верно классифицируемые как положительный класс, FN - ложно классифицируемые как отрицательный класс.

Рассмотрим матрицу ошибок, в которой TN - правильно классифицируемый отрицательный класс.

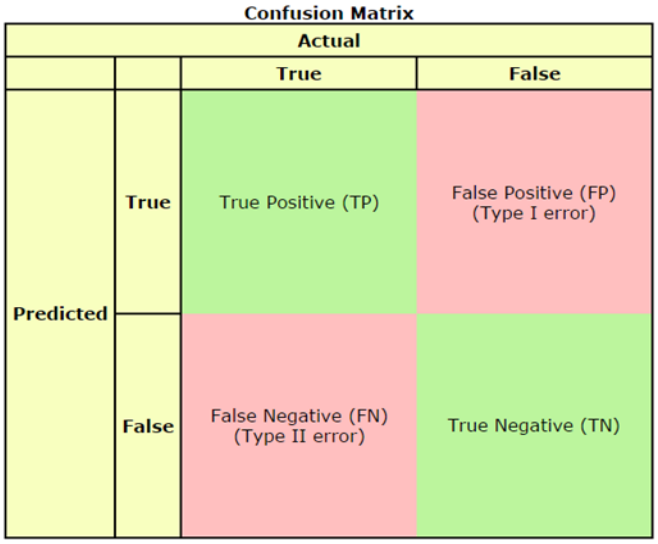

Рисунок 2 - Пример матрицы ошибок.

1.1.4 F1-мера

F1-мера является гармоническим средним между точностью и полнотой:

$$F1 = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

В случае многоклассовой классификации матрица ошибок расширяется на необходимое количество меток. Информация о точности, полноте и F1-мере дается по каждому прогнозируемому классу отдельно.

1.1.5 Out-of-bag score

Каждое дерево из случайного леса обучается на бутстрапированной выборке, можно показать, что для каждого дерева в обучающую выборку попадают примерно 63% всех данных, остальные не используются. Идея в том, чтобы проверить качество алгоритма на 37% неиспользованных данных по формуле выше. Согласно работе [22], использование oob score для задачи классификации вместо кросс-валидации при условии большой выборки может быть

оправдано. Хотя, в общем случае, она менее устойчива (в частности, может быть завышение ошибки) при сравнении кросс-валидацией, но измерение этой ошибки занимает гораздо меньше времени, поскольку не нужно обучать алгоритм n-ое количество раз.

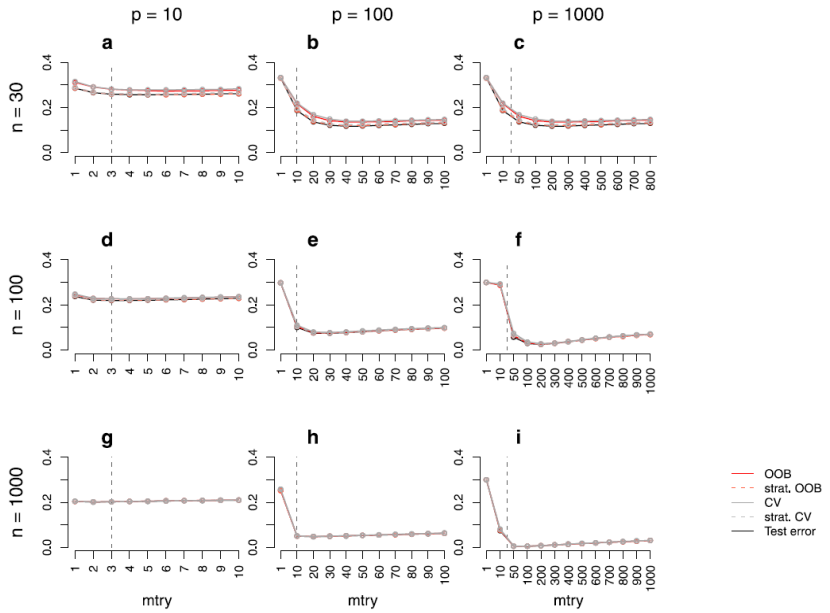

Рисунок 3 - Результаты обучения случайного леса на разных метриках при различных mtry в задаче бинарной классификации при несбалансированных метках (30 % наблюдений меньшего класса) [22].

Данные о работе oob-score представлены на рисунке 3, где mtry - подмножество признаков для выбора разбиения на каждом шаге деревьев случайного леса, p - общее количество предикторов, n - количество точек.

Данная метрика уже использовать в задаче распознавания целенаправленной физической деятельности в работе [10].

1.2 Статистические методы анализа и обработки данных:

1.2.1 Выборочное математическое ожидание

Математическое ожидание будет находиться по выборке данных, а значит имеет следующую формулу:

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \quad , \tag{5}$$

где x_i – элемент выборки, n – ее длина. Часто формула отображается в виде оператора $E[X^n]$.

1.2.2 Оценка стандартного отклонения (выборочное отклонение)

Вычисление оценки стандартного отклонения основано на вычислении выборочной дисперсии (смещенная оценка дисперсии) и вычисляется по следующей формуле:

$$\sigma = \sum_{i=1}^{n} \sqrt{(x_i - \bar{x})^2} \tag{6}$$

1.2.3 Коэффициент асимметрии:

Величина, которая характеризует асимметрию распределения выборки данных относительно нормального, мера скошенности распределения влево или вправо.

$$A_s = E[(\frac{X^n - E[X^n]}{\sigma})^3] \tag{7}$$

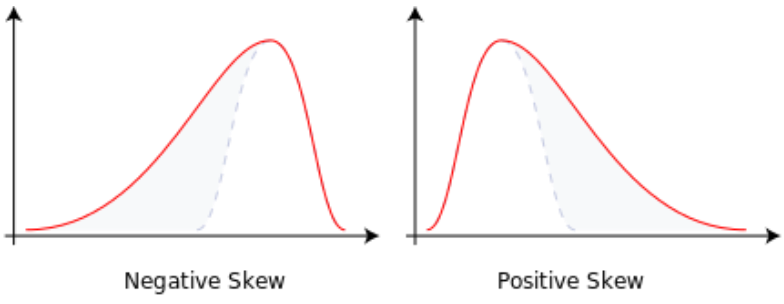

Рисунок 4 - Поведение коэффициента асимметрии.

1.2.4 Коэффициент эксцесса

Определяет меру остроты пика выборочного распределения относительно нормального распределения. Мера скошенности по вертикали.

$$\gamma_2(X) = \frac{E[(X^n - E[X^n])^4]}{E[(X^n - E[X^n])^2]^2} \quad (8)$$

1.2.5 Энтропия

Энтропия характеризует меру неопределенности в данных.

$$H(X^n) = \sum_{i=1}^{N}(p_i * log(p_i)) \ , \quad (9)$$

где p_i рассчитывается как:

$$p_i = \frac{x_i}{\sum_{j=1}^{N} x_j} \quad (10)$$

1.2.6 Квантили

x_α, - α квантиль (квантиль порядка α) случайной величины X, если

$$P(X < x_\alpha) \geq \alpha \quad (11)$$

$$P(X \geq x_\alpha) \geq 1 - \alpha \ , \quad (12)$$

где оператор P(condition) обозначает вероятность события, заключенного в скобках значение $\alpha \in (0, 1)$

1.2.7 Мода

Обозначает значение во множестве наблюдений, которое встретилось чаще остальных.

1.2.8 QQ-plot

Визуальный способ предположений о проверке нормальности выборочных данных.

Этапы построения:

1. Отсортировать выборку по неубыванию
2. Сопоставить каждому объекту выборку точку на графике:
   2.1. Значение вертикальной оси - значение элемента выборки
   2.2. Значение по горизонтальной оси - математическое ожидание квантиля стандартного нормального распределения, подсчитанной по выборке такого же объема.

Если выборка взята из нормального распределения, то точки на графике должны лежать на одной прямой:

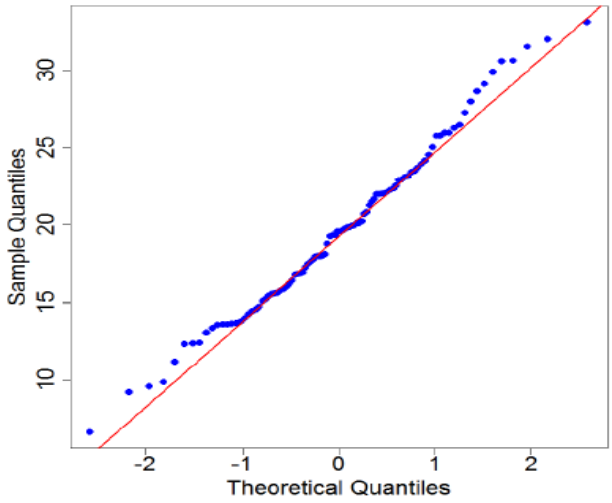

Рисунок 5 - Пример Q-Q графика

1.2.9 Критерий Шапиро-Уилка

Критерий основан на Q-Q графике, формально проверяет соответствии распределения выборки нормальному:

Определяется выборка X^n содержащая числа от x_1 до x_n.

Нулевая гипотеза проверяет предположение о генерации выборки из нормально распределения. Альтернатива утверждает, что выборка было получения из другого распределения.

Статистика рассчитывается следующим образом:

$$W(X^n) = \frac{(\sum_{i=1}^{n} a_i x_i)^2}{\sum_{i=1}^{n}(x_i - E[X^n])^2} \quad , \tag{13}$$

где a_i- величина, основанная на математических ожиданиях порядковых статистик из нормального распределения, для нее не существует аналитического решения.

Нулевое распределение рассчитываемой статистики табличное, так как его невозможно рассчитать аналитически.

1.2.10 Boxplot

Boxplot – удобный способ визуализации выборок, сгенерированных из нормального распределения. Он позволяет посмотреть на статистических метрики, структуру данных и выбросы.

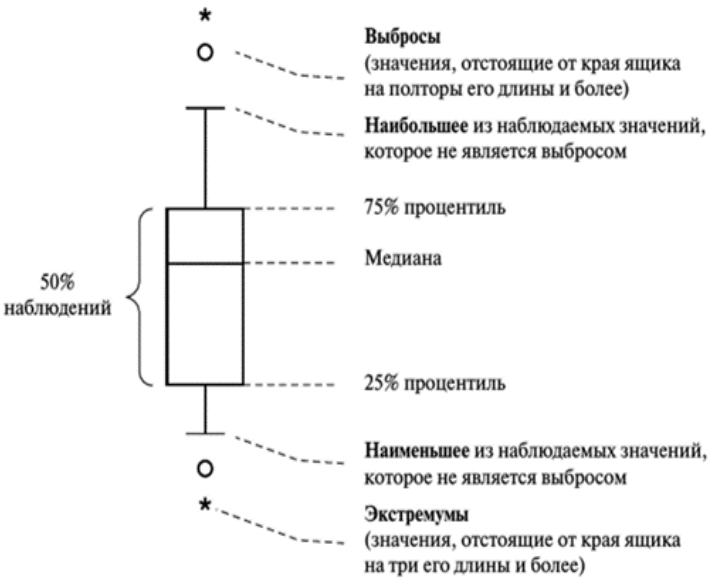

Рисунок 6 - Устройство BoxPlot.

1.2.11 Корреляционный анализ

Для оптимальных моделей рекомендуется отсутствие корреляции (или слабая корреляция) между предикторами. Если корреляция будет сильна, то модель может переобучаться на тренировочной выборке и выдавать низкий результат на тестовой

По этим причинам часто для анализа используется корреляция Спирмена.

Корреляция Спирмена - мера монотонной взаимосвязи между двумя случайными величинами.

Формула расчета корреляции Спирмена:

$$\rho_{X_1^n X_2^n} = 1 - \frac{6}{n^3 - n} * \sum_{i=1}^{n} (rank(x_1^i) - rank(x_2^i)) \ , \quad (14)$$

где $rank(x_1^i)$ - ранг i-го элемента относительно выборки x_1^n,

x_1^i – i-ый элемент первой выборки.

1.2.12 Обучающая, валидационная, тестовая выборки

Данные разделяются на 3 подвыборки: обучающую (большая часть данных 70%), валидационную (15 %) и тестовую (15 %).

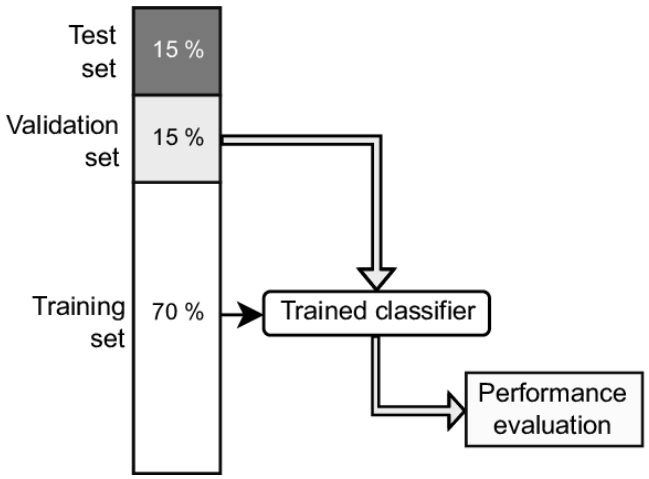

Рисунок 7 - Пример использования разбиения

Также выделяют стратифицированную выборку, в которой пропорции между распознаваемыми классами сохраняется в трех видах подвыборок. Такие разбиения могут использоваться для подбора гиперпараметров модели или для сравнения модели с целью выбрать лучшую из набора.

1.2.13 Межквартильный размах (IQR)

IQR - число, показывающее разброс средней половины (в которую входят 50 % данных от середины). Разница между первым (0.25 квантиль) и третьим квартилем (0.75 квантиль).

$$IQR = x_{0.75} - x_{0.25} \quad , \tag{15}$$

где $x_{0.25}$ (Q1) - первый квартиль выборки,

$x_{0.75}$ $(Q3)$ - третий квартиль выборки.

Для очистки данных от выбросов и аномальных значений признаков будет использован интервал 1.5 * IQR. Важно, что для применения данного алгоритма требуется нормальность

распределения выборок, иначе данный метод не будет иметь никакого смысла, поскольку удаление аномалий осуществляется при предположении о нормальности данных

Выбросами будут классифицироваться данные, находящиеся:

– ниже Q1 - 1.5 * IQR;
– выше Q3 + 1.5 * IQR;

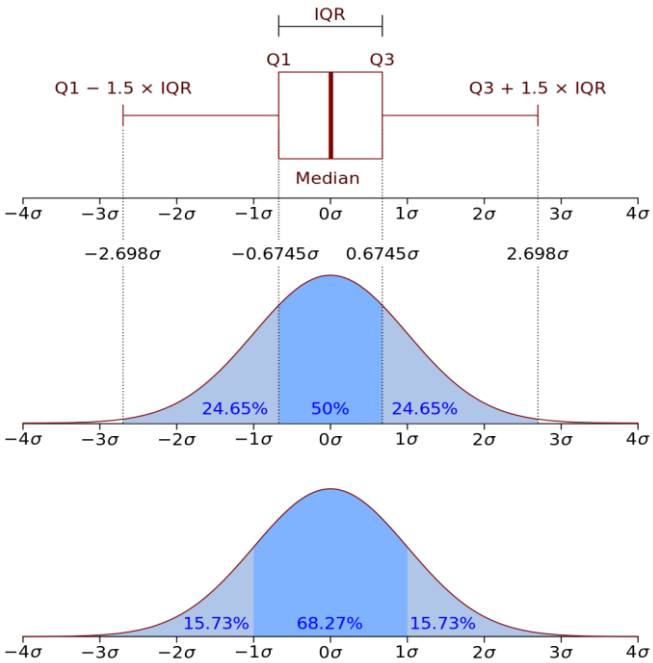

Рисунок 8 - Boxplot, показывающий квартили, IQR, 1.5 * IQR нормально распределенной случайной величины.

Также будут удаляться данные, которые являются неинформативными, например, если признак содержит метку, отличную от нуля только в паре случаев из большого множества, то, очевидно, что такие данные непригодны для обучения.

1.2.14 Кодирование зависимого столбца

Требуется для кодирования категориальных столбцов числами для подачи в некоторые модели машинного обучения, например, SVM и Random Forest. Этот метод кодирования обозначает словосочетанием Label Encoder.

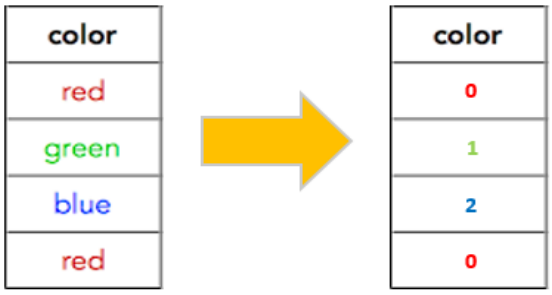

Рисунок 9 - Label Encoder. Пример работы.

Для классификатора CatBoost данный алгоритм не нужен, так как он уже реализован внутри модели.

1.2.15 Методы скользящего и фиксированного окна:

Для того, чтобы преобразовать данные с целью извлечения новых признаков или сглаживания аномалий используется метод скользящего окна. Его преимущество заключается в том, что при такой обработке практически не теряется полезной информации.

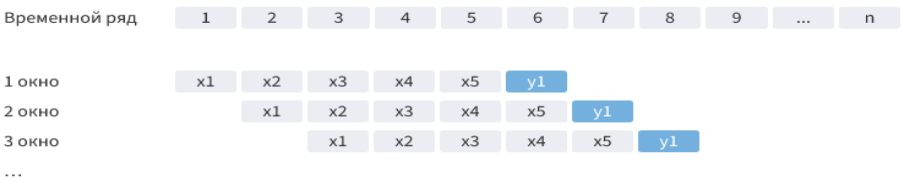

Рисунок 10 - Метод скользящего окна.

Метод скользящего окна использует различные функции, позволяющие извлечь из данных нужную информацию.

Метод фиксированного окна (non-overlapping) отличается лишь разбиение окон на непересекающиеся интервалы

1.2.16 Поправка на множественную проверку гипотез методом Бенджамини-Хохберга

Во время множественной проверки гипотез шанс отклонить нулевую гипотезу сильно возрастает (вероятность ошибки первого рода становится существенно больше, чем 0.05), по этой причине используются поправка на множественную проверку гипотез. Для расчёта метода Бенджамини-Хохберга используется мера FDR

$$FDR = E[\frac{V}{\max(R,1)}] \quad , \tag{16}$$

где R - число отвергнутых гипотез,

V - число неверно отвергнутых гипотез (число ошибок первого рода),

оператор max(R, 1) определяет большее число из предложенного в скобках.

В таком случае, уровень значимости α в следующей формуле:

$$FDR = E[\frac{V}{\max(R,1)}] \leq \alpha \quad , \tag{17}$$

Метод Бенджамини-Хохберга работает с нисходящим вариационным рядом уровней значимости. На первом шаге самый большой p-value p_m сравнивается с константой α_m, если $p_m \leq \alpha_m$, то соответствующая уровню значимости нулевая гипотеза H_m и все остальные (от H_1 до H_{m-1}) отвергаются, иначе гипотеза H_m принимается, и процедура продолжается до тех пор, пока не закончатся уровни значимости в ряду

Константы рассчитываются следующим образом:

$$\alpha_i = \frac{i * \alpha}{m} \tag{18}$$

где i – число, проходящее все значения от 1 до m

1.3 Алгоритмы машинного обучения:

1.3.1 Алгоритм градиентного бустинга CatBoost

Catboost классификатор - метод градиентного бустинга, реализованный над симметричными деревьями.

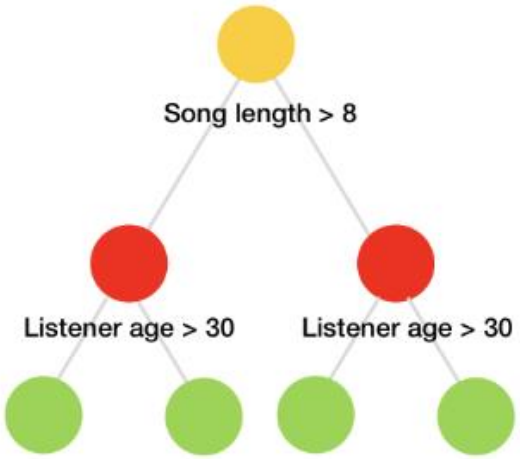

Рисунок 11 - Пример симметричного дерева.

Градиентный бустинг - алгоритм построения композиции простых деревьев, где каждое следующее дерево в композиции исправляет ошибку предыдущих, двигаясь в сторону, обратную градиенту целевой функции ошибки на данных. Если говорит в общем, то прогноз каждого следующего дерева берется с определенным коэффициентом λ, который обычно равен числу в интервале от 0 до 1. Это нужно для предотвращения переобучения и поиска более сложных закономерностей в данных.

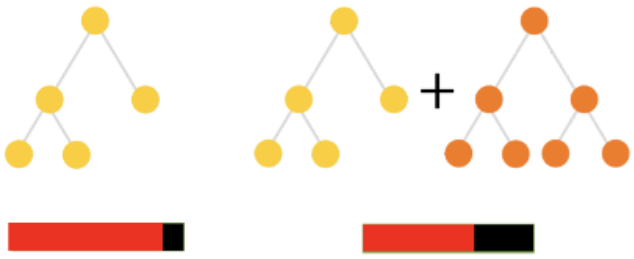

Рисунок 12 - Пример работы градиентного бустинга.
Шкала - ошибка на функции потерь.

Проблемой большинства реализаций алгоритма градиентного бустинга является смещение оценки градиента, которая приводит к переобучению. Catboost классификатор исправляет эту проблему при помощи рандомизации на различных этапах создания деревьев. Например, в CatBoost классификаторе используется бутстрапированая подвыборка обучающей выборки для каждого дерева в композиции, что приводит к их рандомизации и, как следствие, избеганию переобучения.

1.3.2 Метод опорных векторов (SVM)

SVM (support vector machine) - линейный классификатор, максимизирующий расстояние между разделяющей прямой (гиперплоскостью) и крайними точками данных (опорными векторами) по разные стороны от этой прямой (в случае линейной разделимости данных).

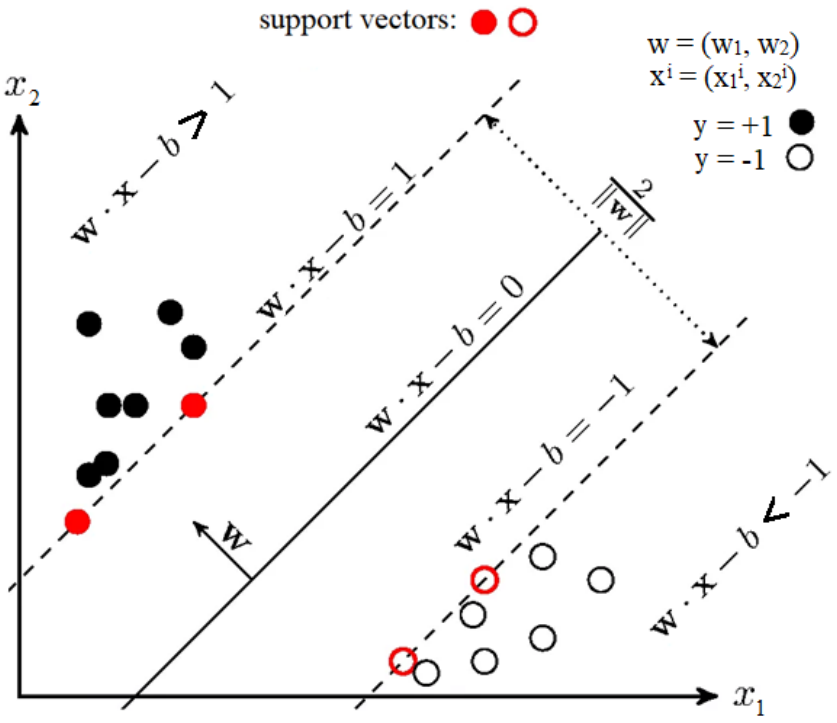

Рисунок 13 - Принцип работы SVM в случае линейно разделимых классов.

Пусть:

x_+- объект, отнесенный к положительному классу с самой низкой уверенностью из возможных (опорный вектор)

x_-- объект, отнесенный к отрицательному классу с самой низкой уверенностью из возможных (опорный вектор)

w - вектор весов классификации SVM

Для разделяющей полосы в случае линейно разделимых выборок справедливо выражение:

$$< (x_+ - x_-), \frac{w}{\|w\|} >= \frac{2}{\|w\|} \ , \tag{19}$$

где $\|w\| - l2$ норма вектора $\|w\|$,

оператор <v1, v2> - обозначает скалярное произведение между векторами v1 и v2

Тогда оптимизационная задача имеет следующий вид:

$$\begin{cases} < w, w > \rightarrow min \\ y_i * (< w, x > - w_0) \geq 1 \\ \forall i \in [1, n] \end{cases}, \tag{20}$$

где $< w, w > \rightarrow min$ обозначает задачу минимизации скалярного произведения вектора w на самого себя.

В случае линейной неразделимой выборки второе условие системы уравнений [20] не может быть выполнено для всех i по определению. Для этого задача обобщается на случай линейно неразделимых данных при помощи введения ошибок алгоритма $\xi_i \geq 0$ и их интеграции в минимизируемую функцию потерь:

$$\begin{cases} \frac{1}{2} * < w, w > + \mathrm{C} * \sum_{i=1}^{n} \xi_i \rightarrow min \\ y_i * (< w, x > - w_0) \geq 1 - \xi_i \\ \xi_i \geq 0 \\ \forall i \in [1, n] \end{cases}, \tag{21}$$

где C - гиперпараметр, задающий размер штрафов за ошибки.

В нелинейных случаях часто используются ядра. Ядро отображает пространство признаков в новое, где классы становятся почти или полностью линейно разделимы.

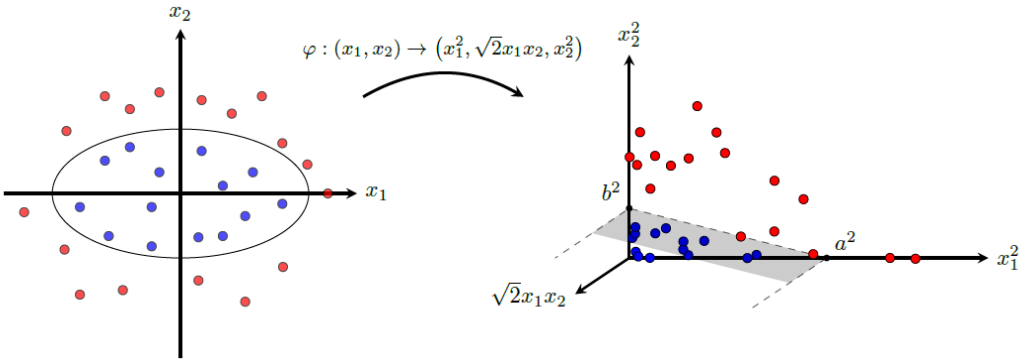

Рисунок 14 - Пример работы спрямляющего пространства в задаче классификации. Предикторы x_1 и x_2 переводятся в новые признаки $x_1^2, \sqrt{2}x_1x_2, x_2^2$.

Если вместо скалярного произведения использовать нелинейную симметричную функцию $K(w, x)$

$$K(w, x) = < \varphi(w), \varphi(x) > \ , \tag{22}$$

где φ - спрямляющее отображение.

То метод SVM может быть сформулирован в исходном пространстве с получением нелинейной разделяющей поверхности. В этом суть метода kernel trick.

В качестве ядра может быть использовано полиномиальное ядро, которое представляет собой пространство мночленов не выше степени d:

$$K(w, x) = << w, x > + r >^d \tag{23}$$

1.3.3 Алгоритм Random Forest (случайный лес)

Случайный лес - ансамблевый метод над простыми деревьями решений (Decision Tree), данный алгоритм машинного обучения исправляет главную проблему дерева решений - сильное переобучение путем усреднения предсказания по множеству деревьев.

Алгоритм основан на композиции решающих деревьев:

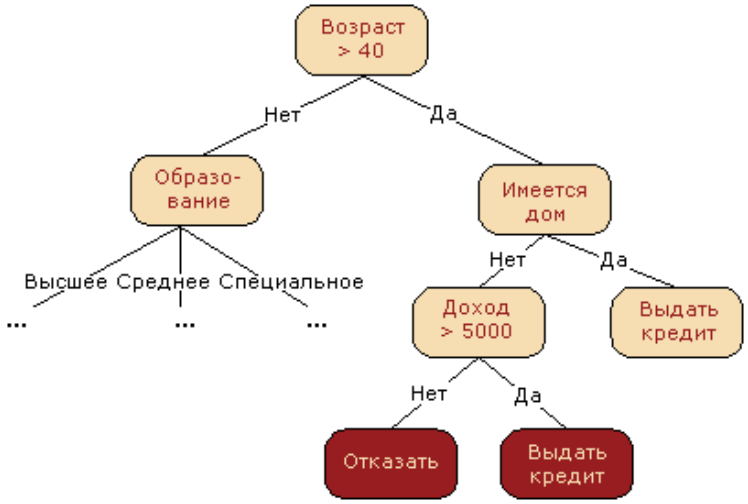

Рисунок 15 - Пример работы решающего дерева

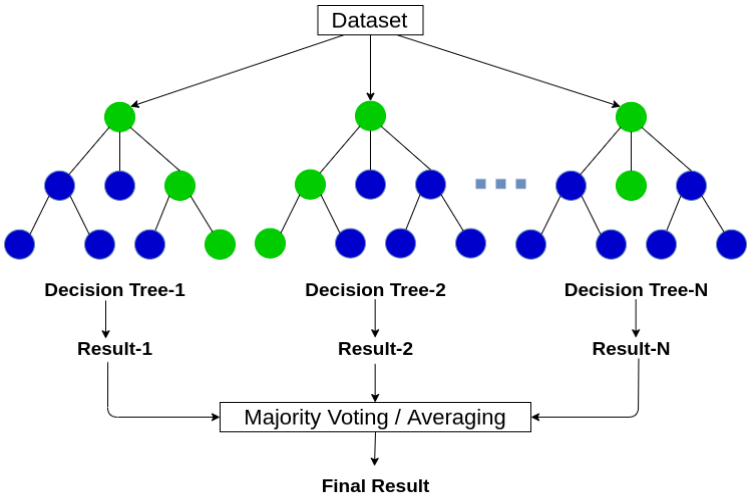

Рисунок 16 - Принцип работы случайного леса

Идея случайного леса состоит в том, чтобы объединить N алгоритмов деревьев решений (от b_1 до b_N) в один, после чего усреднить ответы:

$$a(x) = \frac{1}{N}\sum_{n=1}^{N} b_n(x) \quad (24)$$

Однако, для обучения базовых алгоритмов нельзя использовать одну и ту же выборку, причина этого в разложении ошибки в точке x:

$$Err(x) = Bias(\hat{f})^2 + Var(\hat{f}) + \sigma^2 \ , \quad (25)$$

где:

Квадрат смещения $Bias(\hat{f})^2$- средняя ошибка алгоритма на всевозможных наборах данных

Дисперсия $Var(\hat{f})$ - дисперсия алгоритма. Различие ответов при обучении на разных наборах данных

σ^2 - ошибки, которые невозможно устранить (например, погрешности замеров данных)

$\hat{f}$ – обученные алгоритм случайного леса

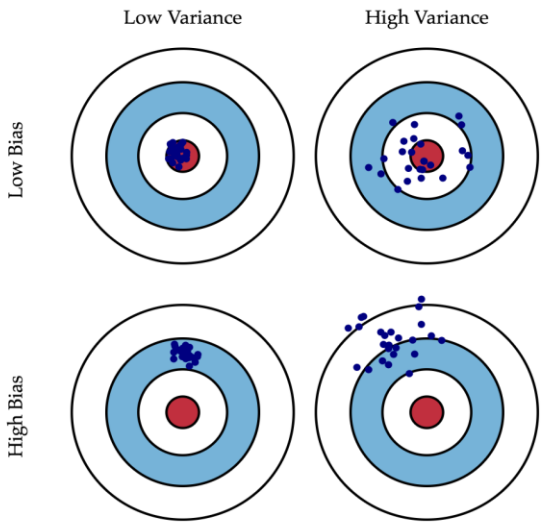

Рисунок 17 - Отличие компонентов ошибки.

Известно, что дерево решений имеет низкий квадрат смещения, но очень высокую дисперсию ошибки, в результате чего он постоянно переобучается. Для того, чтобы исправить эту проблему в случайном лесе используются:

- Бэггинг (бутстрапированная выборка) - каждый базовый алгоритм обучается на своей подвыборке данных, причем, чем более различны эти подвыборки, тем меньше будет дисперсия алгоритма.
- Каждое разбиение в деревьях выбирается из подвыборки всех признаков.

Алгоритм построения случайного леса из N решающих деревьев:

1. Построить с помощью бутстрапа сгенерировать N подвыборки X_n, где n принимает значения от 1 до N
2. Каждая получившаяся подвыборка используется как обучающая для построения соответствующего решающего дерева $b_n(x)$.

2.1. Дерево строится, пока в каждом листе окажется не более n_{min} объектов. Очень часто деревья строят до конца (n_{min} = 1), чтобы получить сложные и переобученные решающие деревья с предельно низким смещением.

2.2. Процесс построения дерева рандомизирован на этапе выбора оптимального признака, по которому будет происходить разбиение. Оно ищется среди случайного подмножества размера q, а не среди всего подмножества признаков.

2.3. Следует обратить особое внимание, что случайное подмножество размера q выбирается заново каждый раз, когда необходимо разбить очередную вершину. В этом состоит основное отличие такого подхода от метода случайных подпространств, где случайное подмножество признаков выбиралось один раз перед построением базового алгоритма.

3. Построенные деревья объединяются в композицию и ответ алгоритма будет вариант, за который “проголосовало” большинство базовых алгоритмов.

Для построения отдельных деревьев критерий ошибки при каждом разбиении записывается следующим образом:

$$Q(X_m, j, t) = \left(\frac{|X_l|}{|X_m|}H(X_l) + \frac{|X_r|}{|X_m|}H(X_r)\right) \rightarrow min \quad , \tag{26}$$

где $H(X_l)$ - критерий информативности,

$|X_l|$ - количество объектов, ушедших в левый лист,

$|X_r|$- количество объектов, ушедших в правый лист.

$|X_m|$ - общее количество объектов текущего разбиения,

j - номер признака, по которому разбиваются данные,

t - значение признака, по которому бьется выборка X_l.

Рассмотрим критерий, на основании которого совершается разбиение: критерий информативности Джини (Gini):

Если p_k - доля объектов класса k в выборке X^n, то рассчитывается он следующим образом:

$$p_k = \frac{1}{|X^n|}\sum_{i \in X^n}[y_i = k] \tag{27}$$

Тогда критерий информативности Gini записывается следующим образом:

$$H(X^n) = \sum_{k=1}^{K} p_k(1 - p_k) \quad (28)$$

Оптимум достигается в случае, когда объекты X^n все относятся к одному классу.

Кроме того, случайный лес способен рассчитывать важность признаков для обучения. Рассмотрим принцип работы:

Чем выше признак разбиения в дереве, тем важнее он для прогноза, поскольку он сильно уменьшает неопределенность в данных (уменьшает критерий Джини).

Для разбиения в дереве решений:

$$ni_j = \frac{|X_m|}{|X_{m-1}|} H(X_m) - \frac{|X_l|}{|X_m|} H(X_l) - \frac{|X_r|}{|X_m|} H(X_r)) \quad , \quad (29)$$

где $|X_{m-1}|$ - количество классифицируемых объектов в листе, который является родительским для X_m.

X_{m-1} – классифицируемые объекты в листе

ni_j - важность отдельного разбиения j в дереве по признаку i.

Для отдельного дерева в композиции важность признаков рассчитывается как:

$$fe_i = \frac{\sum_{j:\text{разбиение } j \text{ по признаку } i} ni_j}{\sum_{k \in \text{все разбиения дерева}} ni_k} \quad (30)$$

После чего величина fe_i нормируется:

$$norm(fe_i) = \frac{fe_i}{\sum_{j \in \text{все признаки}} fe_j} \quad (31)$$

Значимость признаков для всего случайного леса считается по следующей формуле:

$$RF(fi_i) = \frac{\sum_{j \in \in \text{все деревья}} norm(fi_i)}{T} \quad , \quad (32)$$

где T - количество деревьев в случайном лесе.

2 Структурный системный анализ исследуемого объекта

В рабочем процессе на предприятии существенную роль играет горизонтальная коммуникация между подчиненными, занимающимися производственными работами и начальством. Она может быть как прямая (например, диалог прораба с рабочим), так и косвенная (например, выдача технического задания рабочим).

За организацию такой коммуникации отвечает процесс управления целенаправленной физической работой персонала, использующийся для управления сотрудниками, которые задействованы преимущественно в ручном труде на предприятии (например, строители, дворники и пр.). Этот процесс представляется крайне важным для компаний, где существенная часть работ выполняется персоналом вручную, так как процесс напрямую влияет на физическую работу посредством таких производственных этапов как: разработка технического задания, контроль выполнения задач и корректировка результатов работы и.т.п. Хоть такое управление людьми может сильно различаться в зависимости от рассматриваемых экономических сфер и компаний, однако, существуют основные положения и свойства, которые не меняются.

Общую иерархию сотрудников предприятия можно представить в разбивке на следующей диаграмме:

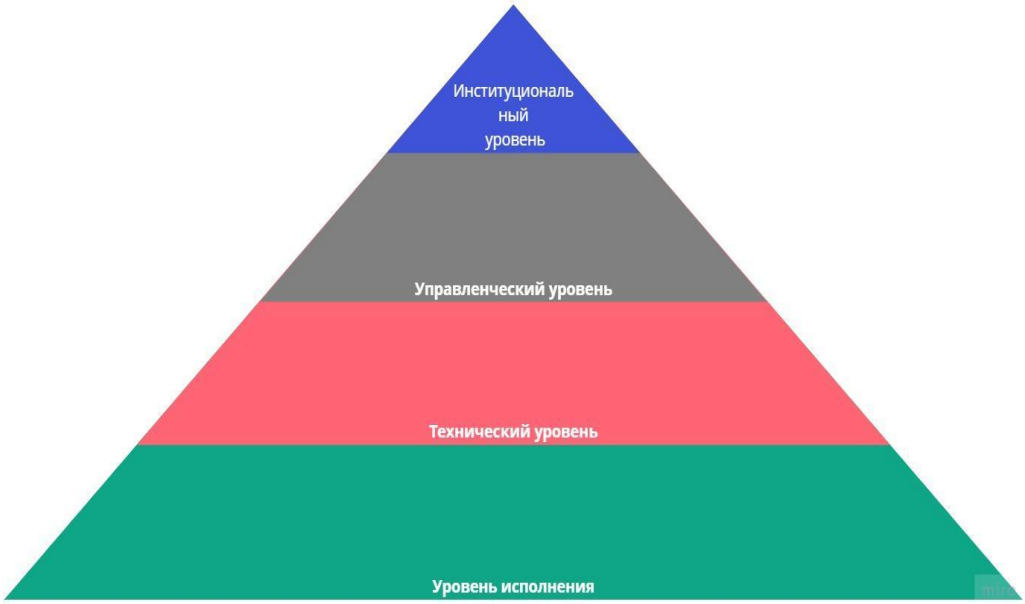

Рисунок 18 - Иерархия уровней сотрудников.

Распишем каждый из уровней подробнее:

- институциональный уровень - преимущественно занимается постановкой общей проектной задачи для работ на предприятии в рассматриваемой предметной области. На этом уровне задействовано только 3-7% от общей численности персонала. Рабочий персонал - руководители высшего звена (президент компании, совет директоров и пр.);
- управленческий уровень - формирует план на основе задачи, поставленной на институциональном уровне, в целях утверждения временных рамок и детализации требуемых работ, после чего он передается ниже по уровню. После завершения рабочего процесса на техническом уровне происходит проверка результатов рабочей деятельности и отправка их на верхний уровень в случае успешного окончания работ. Рабочий персонал - руководители среднего звена (руководители отделов/департаментов и пр.);
- технический уровень - осуществляет формирование конкретных производственных заданий на основе сформированного плана и контроль за выполнением этих заданий. Работа над ними происходит на уровне исполнения. Рабочий персонал - руководители низшего звена (прорабы, начальники участков и пр.);
- уровень исполнения - на этом уровне выполняются производственные работы непосредственно над реализуемым продуктом. Персоналом являются рабочие уровня исполнения (различные строители и прочий производственный персонал);

В процесс управления целенаправленной физической работой выделяются объекты и субъекты:

Объект управления персоналом - деятельность, выполняемая субъектами управления персоналом (например, разработка плана производственных работ, оценка работы сотрудников и пр.).

Субъект управления персоналом - лицо, от которого могут зависеть подпроцессы предметной области (например, формирование плана работ, выдача производственных заданий работникам и.т.п.). Уровень важности решаемых им задач определяется относительно его положения в иерархии на предприятии. От него требуется воздействовать напрямую на соответствующий ему объект управления персоналом для достижения поставленных целей.

Основная функциональная задача предметной области - максимально эффективно использовать человеческий ресурс для достижения поставленных проектных задач.

Проектные задачи могут быть самыми разными в зависимости от предприятия и экономической сферы, в которой компании функционируют.

Дополнительные задачи предметной области:

- контроль качества производственных работ (соблюдение технологий производства);
- контроль безопасности на производстве (соблюдение ПБиОТ);
- эффективная постановка задач производственных работ на всех уровнях;
- качественное выполнение производственных работ;
- введение в эксплуатацию результатов рабочей деятельности;

3. Анализ технического, программного и информационного обеспечений

Рассмотрим применяемые технические, программные и информационные обеспечения в процессе управления персоналом.

Для процесса управления персоналом выделяются три вида обеспечения:

- техническое обеспечение - аппаратные решения, формирующие или используемые для формирования информационных баз на машинных носителях посредством программного обеспечения;
- программное обеспечение - системы, созданные при помощи программного кода и позволяющие взаимодействовать с техническим обеспечением для решения широкого круга рабочих задач;
- информационное обеспечение - решения в предметной области, касающиеся размещения, форм организации и объема информационных потоков, циркулирующих в процессе управления целенаправленной физической работой персонала;

3.1 Анализ технического обеспечения:

Основой технического обеспечения процесса управления персоналом составляет комплекс технических средств (КТС), представляющий собой взаимосвязанный набор аппаратных решений, позволяющих автоматизировать различные подпроцессы предметной области, а иногда сделать некоторые из них автономными. Задачи, выполняемые подсистемами КТС могут быть разными: сбор, регистрация, накопление, обработка, вывод и визуализация различной информации, однако, в общем, они должны способствовать решению задачи управления целенаправленной физической активностью рабочих с уменьшением финансовых затрат за счет повышения производительности труда сотрудников и автоматизации некоторых подсистем.

В качестве наиболее популярных аппаратных решений используется оргтехническое оборудование, включающее в себя системные блоки, принтеры, сканеры и пр. Эти средства используются на институциональном, управленческом и технологическом уровнях.

Таблица 2 - Общее техническое обеспечение для руководителей высшего, среднего и низшего звеньев.

№ п.п.	Наименование	Технические характеристики	Назначение
1.	Системный блок	Процессор: не менее двух ядер с частотой 1,6 ГГЦ ОЗУ: от 4 ГБ Устройство хранения информации: от 64 ГБ Сетевой адаптер: от 100 mbit OS: Windows, GNU Linux	Формирование отчетов, планов работ и документов; передачи информации между сотрудниками. В дополнение управляющие низшего уровня используют указанное оборудование для мониторинга активности рабочих (информация с видеокамер и прочих устройств)
3.	Монитор	60 Гц, 1024 на 768 пикселей разрешение	Визуальное взаимодействие с персональным компьютером
4.	Принтер	Печать: черно-белая Минимальное разрешение печати: 600*600 DPI	Печать документов, представленных на электронных носителях информации
5.	Сканер	Минимальное разрешение считывания: 2400*2400 DPI	Перенос документов в бумажном варианте на электронные носители

Количество системных блоков и мониторов обычно пропорционально количеству руководителей разных звеньев.

Для организации локальной сети используется древовидная топология вместе с инструментами, представленными в таблице 3.

Таблица 3 - Инструменты для организации сети.

№ п.п.	Наименование	Технические характеристики	Назначение
1.	Коммутатор	Режимы полного и половинного дуплекса, управление потоком (IEEE 802.3x), приоритет трафика (IEEE 802.1p), VLAN (стандарт IEEE 802.1q)	Используется для объединения нескольких устройств в пределах локальной сети
2.	Маршрутизатор локальной сети	Поддержка следующий технологий: DHCP, DNS, NAT, VLAN, DMZ	Используется для связи локальных сетей и выхода в интернет
3.	Сервер для хранения рабочей информации	Процессор: Intel Core от 4 ядер по 3.2 Ghz Оперативная память: от 8 gb Запоминающее устройство: от 1 TB HDD Операционная система: Windows Server, GNU Linux	Используется для хранения информации о различной рабочей активности
4.	Сервер для хранения и обработки информации о мониторинге рабочих	Процессор: Intel Core от 6 ядер по 3.2 Ghz Оперативная память: от 16 GB Запоминающее устройство: от 2 TB HDD Операционная система: Windows Server, GNU Linux	Используется для хранения мониторинговой информации о рабочих, задействованных в ручном труде.

Для сбора информации о рабочих с целью мониторинга их производственной активности обычно используются визуальные средства сбора телеметрии рабочих, представленные на таблице 4.

Таблица 4 - Перечень средств, которые зачастую используются для мониторинга.

№ п.п.	Наименование	Технические характеристики	Назначение
1.	Цифровая стационарная видеокамера	Способность записывать видео от 20 кадров в секунду на разрешении от 640x480	Следить за производственной активностью рабочих уровня исполнения посредством удаленной работы управляющих низшего звена
4.	Цифровая портативная видеокамера	1280x720, 20 FPS	Крепится на рабочих с целью слежения за производственной активностью рабочих уровня исполнения

Видеокамеры зачастую используется для мониторинга целенаправленной физической активностью руководителями низшего звена, поскольку позволяют распознавать действиях рабочих визуально. Однако, часто таких средств не хватает, например, в случае плохих погодных условий камеры могут передавать мало полезной информации. Кроме того, визуальная интерпретация способна занимать много времени и не является точной в большом количестве случаев (так как зависит сугубо от интерпретации управляющих низшего звена), особенно, когда производственная работа выполняется при помощи мелкой моторики. Также эти средства имеют ограничения по области считывания (угол обзора) и имеют высокую цену.

По вышеописанным причинам ставится задача интеграции нового средства мониторинга, которое позволит считывать телеметрию рабочих и помогать в те моменты, когда информации с видеокамер недостаточно по различным причинам.

Так как требуется конкретика относительно действий рабочих, то для считывания будут использованы закрепляемые на человеке устройства и считывающие информацию о физическом состоянии рабочих.

Сравнительный анализ приведен в таблице 5.

Таблица 5 - Сравнительная характеристика средств, которые могут быть внедрены в процесс мониторинга рабочих уровня исполнения.

№ п.п.	Наименование	Технические характеристики	Преимущества	Недостатки
1.	Смартфон.	Гироскоп, акселерометр, система андроид, Wi-Fi	Большая распространенность, простая интеграция	Высокая погрешность считывания, низкая надежность, уязвимость к погодным условиям, слабый аккумулятор, низкий охват считываемых данных, средняя цена
2.	Специально разработанная система считывания телеметрии рабочих	Гироскоп, акселерометр, шагомер, датчик сердцебиения, Wi-Fi	Крайне простая интеграция, устойчива к погодным условиям, погрешность измерений минимальна, надежная конструкция, мощный аккумулятор, высокий охват считываемых данных	Распространенность минимальная, стоимость крайне высокая
3.	Умные-часы	Гироскоп, акселерометр, шагомер, датчик сердцебиения, система андроид, Wi-Fi, IP54	Низкая стоимость, легкость интеграции, мощный аккумулятор, минимальная погрешность, высокая надежность, высокий охват физических данных, устойчивость к погодных условиям	Средняя распространенность

Продолжение таблицы 5

№ п.п.	Наименование	Технические характеристики	Преимущества	Недостатки
4.	Умные-часы, урезанные по функционалу	Гироскоп, акселерометр, система андроид, Wi-Fi, IP67	Минимальная стоимость, легкость интеграции	Средняя распространенность, средняя надежность, средний аккумулятор, средняя погрешность, средний охват физических данных, средняя устойчивость к погодным условиям

По результатам сравнения аппаратных средств считывания телеметрии в таблице 5 внедряются умные по следующим причинам:

- практически не сковывают движения рабочих во время работы;
- не требуют долгого процесса интеграции;
- так как устройства компактные и легкие, то шанс их повреждения крайне низок;
- дешевле смартфонов и специально разработанных систем считывания телеметрии рабочих;
- обладают наилучшей защитой от пыли и влаги из всех рассмотренных решений;
- охватывают большое количество данных для считывания;

3.2 Анализ программного обеспечения:

Рассмотрим программное обеспечение, которое может использоваться в рамках системы управления людьми во всех отделах управления.

Таблица 6 - Используемое программное обеспечение для управляющих всех звеньев:

№ п.п	Наименование	Функциональные возможности	Назначение
1.	Реляционная СУБД (Microsoft SQL, PostgreSQL, MySQL и.т.п.)	Система хранения и получения информации	Используется для хранения и получения информации о текущих рабочих задачах и информации о мониторинге рабочих уровня исполнения
2.	Набор офисных приложений (например, Word, Excel и.т.п.)	Подготовка отчетов, презентаций	Используется для создания отчетов и презентаций по рабочей деятельности
3.	Windows	Операционная система для компьютеров	Используется для взаимодействия с компьютером и установки требуемого ПО.
4.	Windows Server	Операционная система для серверов	Используется для взаимодействия с серверами и установки требуемого ПО.

Продолжение таблицы 6

№ п.п	Наименование	Функциональные возможности	Назначение
5.	GNU Linux	Операционная система для компьютеров и серверов	Используется для взаимодействия с компьютерами и серверами для работы и установки требуемого ПО. Также используется для работы: видеокамер, специально разработанных системы считывания данных, носимых человеком.
6.	Android	Операционная система для портативных устройств	Используется для работы смартфонов, урезанных смарт-часов, обычных смарт-часов.
7.	Mozila Firefox/ Internet Explorer	Взаимодействие с интранетом и интернетом	Связь с коллегами по локальной сети и выход в интернет
8.	Система видеонаблюдения для IP-камер «Линия»	Программа для организации видеоконтроля на любом объекте	Используется для мониторинга рабочей активности с камер управляющими низшего звена

3.3 Анализ информационного обеспечения

Информационное обеспечение процесса управления персоналом внемашинное и внутримашинное информационное обеспечение.

Внемашинное информационное обеспечение включает в себя системы классификации и кодирования информации, систему организации, хранения и внесения изменений в документацию. Информационная база представлена совокупностью сообщений, сигналов и документов в форме, воспринимаемых человеком непосредственно, без использования технических средств.

Внутримашинное информационное обеспечение содержит массивы данных, формирующие информационную базу системы на машинных носителях, а также систему программ организации, накопления, ведения и доступа к информации этих массивов.

Информация, считываемая смарт-часами, должна косвенно указывать на совершаемую производственную активность рабочими. Для этих целей обычно считываются количество шагов, пульс, ускорение руки по осям. На основе этих данных управляющие уровня выполнения распознают различные виды целенаправленной физической активности рабочих, например перемещение по территории с совершением работ руками, отсутствие каких-либо действий, совершение работ руками без перемещения по территории и пр.

Рассмотрим функционирование процесса управления людьми в общем виде:

В процессе управления целенаправленной физической активностью требуется постановка общей проектной задачи, она ставится руководством высшего звена. В ней указываются общие методы реализации задуманного проекта, после чего руководителями среднего звена формируется план работ, в котором указаны сроки сдачи, этапы работы и описание каждого из этапов.

На основе этой информации руководители низшего звена формируют задания для подчиненных им работников, в которых описываются технологии, которые должны быть задействованы, и задачи, которые должны быть выполнены. Задания должны быть максимально конкретизированы, чтобы избежать недопонимания с обеих сторон.

Дальше эти задачи выдаются работникам, после чего они выполняют требуемую от них работу, а руководители низшего уровня мониторят их рабочий процесс.

Во время контроля рабочего процесса крайне важно не упустить ошибки в работе на ранних этапах их появления, так как они могут привести к большим проблемам в будущем. Не менее важно получать актуальную и достоверную информацию о производственных процессах. Для обнаружения ошибок управляющие низшего звена общаются с рабочими о

проделанной ими работе, а также проводят внешний осмотр результатов их деятельности. Если производственный процесс реализует технологически сложную систему, то внешнего осмотра может не хватить для качественной оценки проведенной работы. По этой причине управляющие низшего звена используют специальные инструменты, которые способны достоверно показать качество результатов рабочей активности. Руководители не могут мониторить рабочих постоянно, так как проектов может быть множество. Работа над ними может происходить в различных местах. Кроме того, в больших компаниях приходится мониторить большое количество людей, что также накладывает ограничения. В связи с этим вводятся средства автоматизации для считывания телеметрии. Эти средства делятся на стационарные, портативные видеокамеры и смарт-часы. Они считывают полезную телеметрию с рабочих. Когда приходит время проверки, то эти данные обрабатываются управляющими низшего звена с целью распознавания той производственной активности, которой занимались сотрудники во время рабочего процесса.

По результатам вышеописанных процессов ставится вопрос о правильности выполненных работ. Если в работе были недочеты, то вносятся поправки в выданные задания. Если работа не закончена, то мониторинг продолжается до момента полного завершения производственного процесса.

По результатам мониторинга проводится оценка результатов всей рабочей деятельности на основе выданных заданий, после чего информация о выполненной работе направляется руководителям среднего звена, которые оценивают ее на предмет соответствия поставленному плану. В случае несоответствия итоговой реализации вносятся корректировки в план, который отдается руководителям низшего звена для доработки. Если ошибок не выявлено, то результаты работ отправляются руководителям высшего звена. Если они находят неточности между результатами рабочего процесса и поставленной проектной задачей, то проектная задача корректируется и отправляется обратно руководителям среднего звена для доработки.

Для передачи информации между руководителями в основном используются внутримашинное информационное обеспечение (передача информации посредством персональных компьютеров, находящихся в интранете). Иногда для передачи может использоваться внемашинное (передача различных документов, планов работ, технические задания в бумажном варианте), формируется оно часто при помощи средств оргтехники.

Визуализация процесса управления целенаправленной физической работой сотрудников выполнена в соответствии с нотацией диаграммы деятельности UML представлена на рисунке 19.

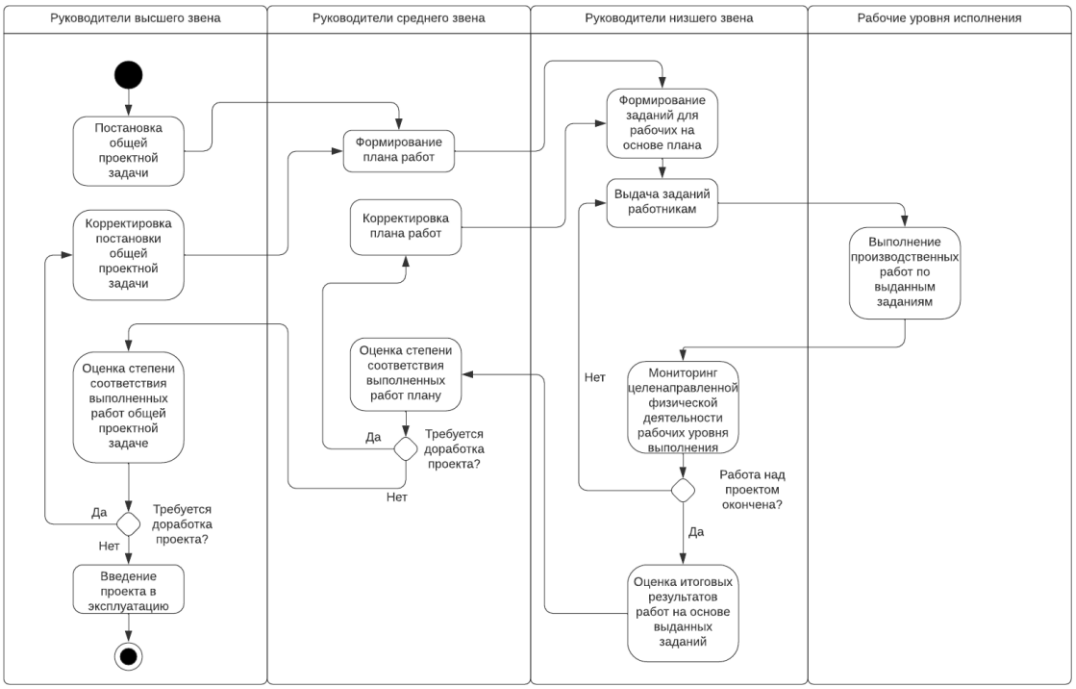

Рисунок 19 - Процесс управления целенаправленной физической активностью сотрудников в общем виде

Декомпозируем процесс “Мониторинг целенаправленной физической деятельности рабочих уровня выполнения” в соответствии с нотацией диаграммы деятельности UML (AS-IS).

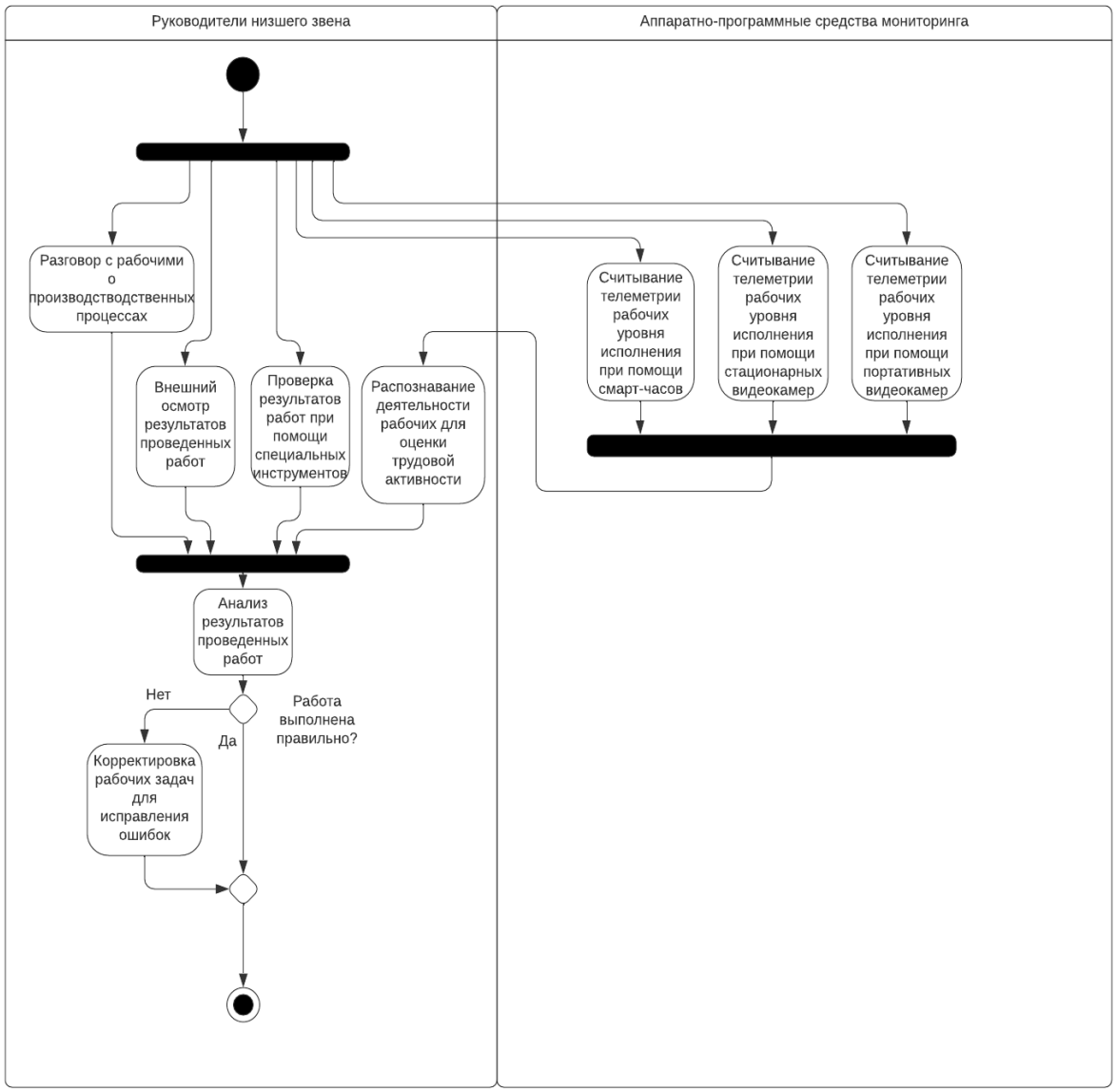

Рисунок 20 - Декомпозиция процесса “Мониторинг целенаправленной физической деятельности рабочих уровня выполнения”

4 Постановка задачи

На предприятиях руководителей низшего звена существенно меньше, чем работников уровня выполнения, поэтому в случае территориального распределения рабочих мест и большого количества подконтрольных рабочих руководить подчиненными сотрудниками в процессе управления целенаправленной физической активностью затруднительно. Проблема контроля наиболее актуальна для процесса мониторинга целенаправленной физической активности рабочих уровня исполнения, где управляющим требуется итерационно опрашивать каждого из подчиненных сотрудников, проводить осмотр результатов работ, корректировать рабочий процесс в случае ошибок, распознавать рабочую активность с устройств считывания телеметрии и.т.п. Если управляющих не хватает, то процесс мониторинга станет менее эффективным и результаты производственных работ будут чаще попадать с ошибками к более высокопоставленным управляющим, что существенно замедлит процесс работы над проектом в целом. В то же время, нежелательно нанимать слишком большой штат управляющих низшего звена, ведь это приведет к существенным тратам для компании.

По этим причинам вводятся средства автоматизации. Они позволяют сократить требуемый штат руководителей и увеличить эффективность процесса управления сотрудниками. Как видно из рисунка 20, процесс распознавания деятельности рабочих, который осуществляется на основе телеметрии, считанной с устройств мониторинга, выполняется руководителями низшего звена, а это требует существенных затрат рабочего времени с их стороны. Процесс распознавания целенаправленной физической активности на основе телеметрии с умных часов можно автоматизировать посредством программного обеспечения, включающего в себя конкретные технологии (методы, модели и алгоритмы) искусственного интеллекта.

Требования к разрабатываемому программному обеспечению:

- должно работать на серверах и компьютерах, представленных на таблицах 2 и 3;
- конфиденциальность данных рабочих уровня выполнения должна быть соблюдена; Личная информация о сотрудниках не должна быть использована построения решения.

Требуется разработать программную систему, которая будет в состоянии автоматически распознавать целенаправленную физическую активность с наименьшим уровнем погрешности. Для этого ставится задача выбора методов анализа данных, обработки данных, отбора модели машинного обучения на основе метрик, генерации статистики для формирования отчета о деятельности сотрудников;

Рассмотрим каждый этап построения решения по отдельности:

Этап анализа представляет собой визуализацию закономерностей данных посредством программных методов, основанных на математических и статистических алгоритмах. Анализ данных позволяет определить наилучшие способы обработки данных.

В качестве способов обработки данных могут использоваться различные методы конструирования, нормализации и редактирования признаков, а также способы разбиения данных для оценки работы алгоритма искусственного интеллекта. Обработка позволяет исключить из данных неинформативные закономерности или выделить важные свойства информации для этапа классификации. Эти процессы позволяют сделать датасеты более полезными для дальнейшего построения программной системы.

Выбор метрик качества обусловлен необходимостью проверки возможностей моделей машинного обучения. В качестве метрик выбираются функции различных видов, если говорить абстрактно, то они принимают обычно на вход 2 множества: множество предсказанных меток и множество изначальных меток. В некоторых случаях на вход принимается третья переменная - модель машинного обучения.

Для классификации деятельности используются алгоритмы машинного обучения, позволяющие на основе закономерностей в данных определять рабочий процесс, которым занят сотрудник. Современные алгоритмы машинного обучения имеют достаточную мощность, чтобы позволить классифицировать активность рабочих с высокой точностью на выбранных метриках и не занимать этим процессом много времени.

Формирование статистики способствует лучшему восприятию данных управляющими низшего уровня. Статистика формируется при помощи различных средств обработки результатов классификации.

Для программной реализации должен быть выбран язык программирования, который обладает достаточным функционалом, чтобы реализовать решение теми средствами, которые были выбраны и имел инструменты для интеграции этого решения в систему распознавания целенаправленной физической активности.

Итоговый программное решение должно быть достаточно мощным, чтобы точно классифицировать активность рабочих и предоставлять результаты работы модели в таком виде, чтобы управляющие низшего звена могли максимально быстро их интерпретировать. В случае соблюдения этих требований, средство автоматизации способно повысить эффективность процесса управления целенаправленной физической активностью рабочих и уменьшить необходимое количество контролирующего персонала.

5 Сущность решения задачи

Выбор инструментов для решения подзадач в процессе создания программного решения происходил в соответствии с анализом НТИ и структурным системным анализом. В качестве исследуемых данных была использована телеметрия, считанная с умных часов. Данные (датасеты) представлены в виде таблиц csv формата, где строки - наблюдения за сотрудниками в разные моменты времени, а столбцы - признаки, характеризующие рабочего уровня исполнения по определенным свойствам. Перечень этих свойств представлен на таблице 7.

Таблица 7- Признаки в таблицах телеметрии сотрудников.

Название признака	Описание признака
'Дата'	Время получения данных с устройства
'Дата_получения_локации'	Время получения данных со спутника
'Широта'	Географическая координата
'Долгота'	Географическая координата
'Высота'	Географическая координата
'Скорость'	Физическая скорость перемещения рабочего, считана посредством геолокации умных часов
'Точность'	Точность измерений
'Ax'	Ускорение по оси x
'Ay'	Ускорение по оси y
'Az'	Ускорение по оси z
'Сердечный_Ритм'	Ритм сердца человека
'Шаги'	Количество шагов, совершенное человеком

Продолжение таблицы 7

Название признака	Описание признака
'Разметка'	Метка класса, столбец, подлежащий распознаванию по признакам выше

Отдельно стоит отметить, что признак “Шаги” был получен с одинаковыми интервалами с устройства мониторинга. Так как данные вдоль строк - временной интервал, то шаги накапливались со временем. По этой причине признак в настоящий момент не является информативным и должен быть обработан.

Был выбран класс алгоритмов обучения с учителем (supervised learning) для классификации, так как он позволяет однозначно определять качество выбранных моделей машинного обучения путем сравнения предсказанных меток зависимого признака с существующей разметкой. Изначальная разметка считываемых данных была следующей:

Таблица 8 - Изначальная кодировка меток классов.

Наименование	Кодировка
Простой (стоит на месте)	a
Передвижение по территории	b
Вкручивание/выкручивание болтов/шурупов	c
Перенос крупных предметов по территории	d
Поднятие предмета	e
Положить предмет	f
Перенос мелких предметов по территории	g
Подключение/отключение проводов от компьютера	h

Продолжение таблицы 8

Наименование	Кодировка
Протирание предметов от пыли	i
Упаковке предметов в коробки	j
Переписка или звонок по телефону	k
Невозможно определить	l

Данные метки являются слишком узконаправленными и не подходят для унифицированной классификации целенаправленной физической деятельности рабочих, поэтому было решено перекодировать их для определения видов активности в общем виде.

Таблица 9- Перекодирование меток классов (группировка по активности).

Наименование (старое)	Кодировка (старая)	Кодировка (новая)	Наименование (новое)
Простой (стоит на месте)	a	a	Простой
Передвижение по территории	b	b	Перемещение
Вкручивание/выкручивание болтов/шурупов	c	e	Низкая мобильная активность
Перенос крупных предметов по территории	d	c	Низкая статическая активность
Поднятие предмета	e	e	Низкая статическая активность
Положить предмет	f	e	Низкая статическая активность
Перенос мелких предметов по территории	g	d	Высокая мобильная активность
Подключение/отключение проводов от компьютера	h	f	Высокая статическая активность

Продолжение таблицы 9

Наименование (старое)	Кодировка (старая)	Кодировка (новая)	Наименование (новое)
Протирание предметов от пыли	i	d	Высокая мобильная активность
Упаковка предметов в коробки	j	c	Низкая мобильная активность
Переписка или звонок по телефону	k	a	Простой
невозможно определить	l	При анализе не рассматривать	

Подробное описание новых меток классов представлено в таблице 10.

Таблица 10 - Значение новых меток классификации.

№ п.п.	Наименование	Характеристика	Критерий определения	Кодировка
1	Простой	Отсутствие каких-либо действий	Фиксация отсутствия интенсивных движений руками и перемещения в пространстве за интервал времени	a
2	Перемещение	Простое перемещение в пространстве	Фиксация перемещения в пространстве со средней интенсивностью движений руками	b

Продолжение таблицы 10

№ п.п.	Наименование	Характеристика	Критерий определения	Кодировка
3	Низкая мобильная активность	Перемещение в пространстве с выполнением простых (низкоамплитудных) действий ручного труда	Фиксация перемещения в пространстве с низкой интенсивностью или отсутствием движений руками	c
4	Высокая мобильная активность	Перемещение в пространстве с выполнением сложных (высокоамплитудных) действий ручного труда	Фиксация перемещения в пространстве с высокой интенсивностью движений руками	d
5	Низкая статическая активность	Выполнение простых (низкоамплитудных) действий ручного труда без перемещения в пространстве	Фиксация движений руками со средней интенсивностью и отсутствие перемещений в пространстве	e
6	Высокая статическая активность	Выполнение сложных (высокоамплитудных) действий ручного труда без перемещения в пространстве	Фиксация движений руками с высокой интенсивностью и отсутствие перемещений в пространстве	f

Средства для решения задачи выбирались посредством анализа НТИ:

Для анализа данных могут быть следующие решения: boxplot, Q-Q plot, гистограммы распределения данных, корреляционный анализ, вывод мер центральной тенденции (математическое ожидание, мода) и мер изменчивости (дисперсия, стандартное отклонение).

Для обработки данных могут быть использованы следующие средства: нормализация данных, обработка выбросов интервалом 1.5 * IQR, тест Шапиро-Уилка и метод Бенджамина-Хохберга. Данные разбиваются стратифицированные обучающую, валидационну и тестовую выборки, либо для разбивки может быть использована кросс-валидация. Кроме того, из изначальных данных для лучшей классификации целенаправленной активности рабочих требуется выделить новые признаки, которые наилучшим образом выделяют закономерности в данных. Способы создания новых признаков могут представлять собой как выделение статистик из каждого изначального предиктора (выделение: выборочного математическое ожидания, выборочного отклонения, коэффициента асимметрии, моды, коэффициента эксцесса, энтропии, средней частота, квантилей), так и перемешивания между собой первоначальных признаков (например, создание полиномов признаков). Первый способ осуществляется как по скользящему окну, так и методом агрегации данных по фиксированным отрезкам временного ряда. Для кодирования категориальных признаков может быть использован Label Encoder. Признак “Шаги” может быть как удален, так и обработан для извлечения из него полезной информации.

Возможные метрики accuracy, precision, recall, confusion matrix, oob-score, f1-score.

Для построения модели классификации могут использоваться алгоритмы машинного обучения, такие как Random Forest, SVM и CatBoost Classifier.

Для формирования итоговой статистики относительно классификации могут быть использованы различные виды группировок и агрегаций предсказанных меток классов.

При интеграции построенного программного решения в предметную область процесс мониторинга целенаправленной физической активности рабочих принимает следующий вид:

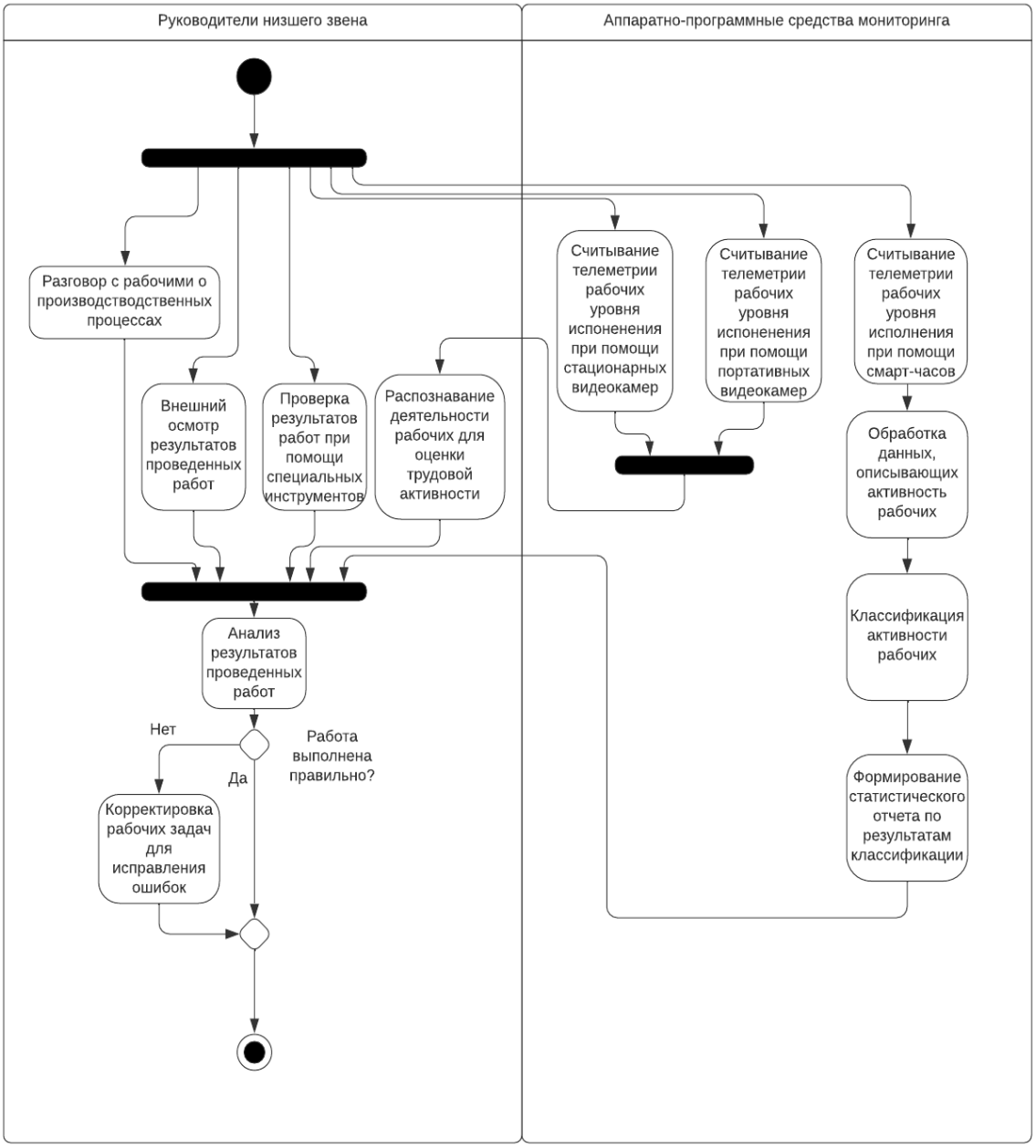

Рисунок 21 - TO-BE модель мониторинга целенаправленной физической деятельностью.

Как видно из диаграммы, программный комплекс при внедрении повышает уровень автоматизации на предприятии. Данные с умных часов предобрабатываются, после чего отправляются в модель машинного обучения для классификации, полученные метки классов используются для формирования статистического отчета, который передается управляющим низшего звена. Подзадача анализа данных, выбора моделей и метрик ставится только для разработки системы автоматического распознавания целенаправленной физической активности. При интеграции в предметную область в них нет необходимости.

Из диаграммы видно, что аппаратно-программный комплекс работает полностью автономно, что позволяет эффективней использовать рабочее время руководителей низшего звена.

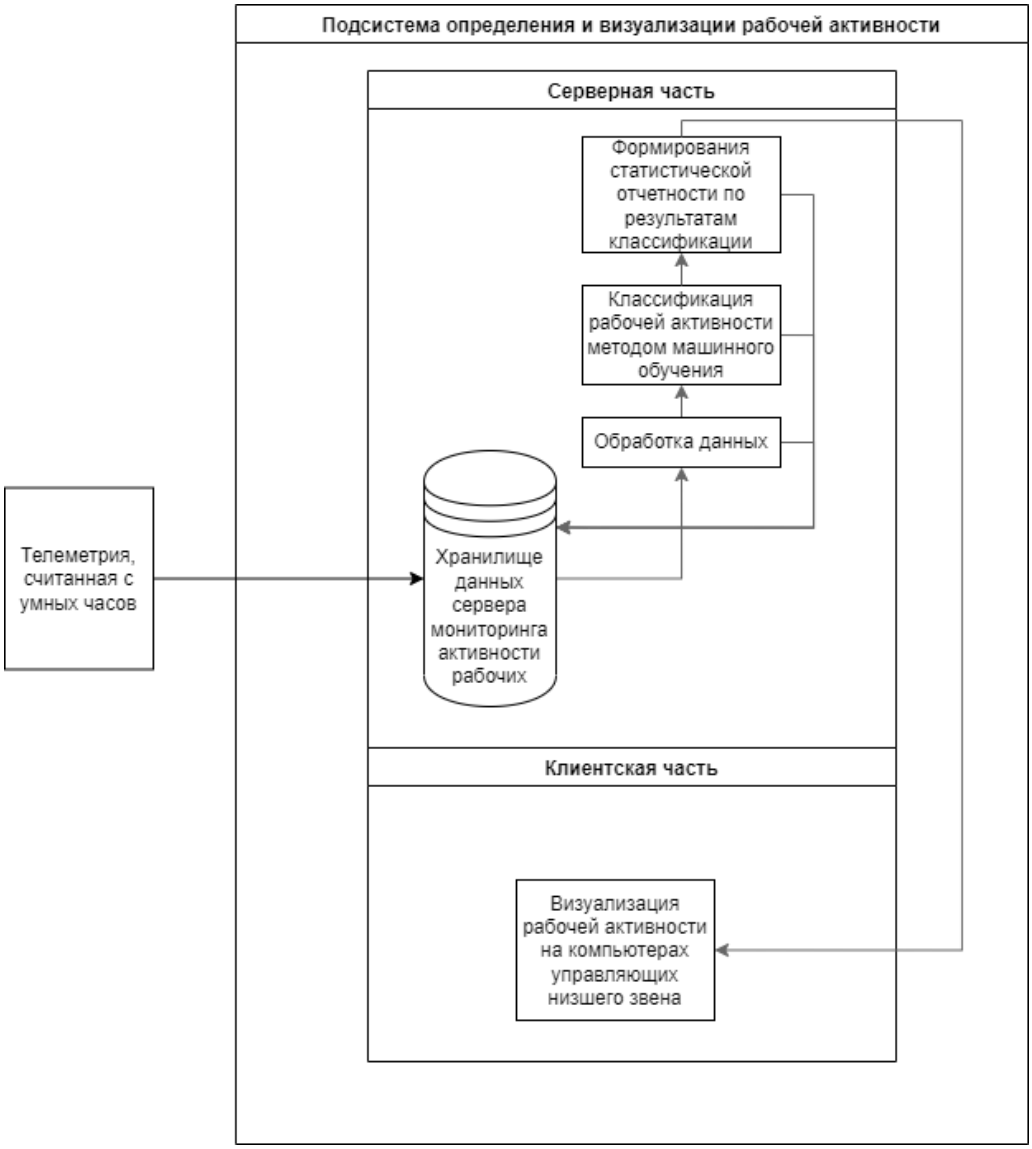

Рисунок 22 – Общий принцип работы системы идентификации целенаправленной физической активности рабочих при внедрении в предметную область.

На рисунке 22 визуализация рабочей активности осуществляются посредством Web интерфейса приложения, развернутого на сервере мониторинга активности рабочих.

6 Выбор и обоснование методов решения задачи

Для анализа данных в научных работах с похожей задачей телеметрия рабочих визуализируются посредством гистограмм, boxplot диаграмм и вывода статистической информации, такие же методы будут использованы в решаемой задаче.

Для анализируемых данных актуальна проблема накопления количества шагов со временем. По этой причине признак не является информативным, для решения этой проблемы нужно вычесть из записей признака “Шаги” количество шагов, которые были сделаны до последнего акта записи данных. После чего колонки предикторов нормализуются.

Обработка данных во всех работах со схожей задачей включает в себя выделение новых признаков при помощи извлечения статистических свойств из исходных колонок таблиц. Количество выделяемых статистических свойств в НТИ различается, поэтому выделим список самых частых признаков, которые будут способны охватить большинство полезной информации для классификации: 25 и 75 квантили, математическое ожидание, среднеквадратичное отклонение, коэффициент асимметрии, эксцесса, мода, минимум, максимум и энтропия. Решено было использовать непересекающиеся интервалы агрегации данных, как это сделано в большинстве рассмотренных работ. Помимо избавления от шумов и выделения полезной информации агрегация должна хорошо работать после интеграции разрабатываемого решения, где данных гораздо больше, чем в текущей работе. Для обработки зависимого признака используется нахождение моды в интервалах агрегации и перезапись изначальной метки полученной величиной. Для обоих видов обработки признаков размер окна выбран длиной в 5 строк данных. Для моделей, которые требуют кодирования зависимого столбца числами будет использован Label Encoder.

Статистическая обработка выбросов на основе гипотез нормальности не используется в задачах распознавания целенаправленной физической активности в НТИ и в данной применяться не будет, поскольку при удалении данных теряется потенциально полезная информация, включая метки классов. Так как данные размещались вручную, то потеря такой информации нежелательна. Однако, будут удаляться признаки, которые бессмысленны для прогноза (например, местонахождение сотрудников), также удаляются признаки, значения в которых не содержат в себе полезной информации (например, признак содержит одно уникальное значение - 0).

После вышеобозначенной обработки признаков две таблицы данных объединяются в одну для дальнейшего ее задействования в задаче отбора модели.

После этих средств используется разбиение на обучающую, валидационную и тестовую выборки (в пропорция разбиения 70%, 15%, 15% соответственно). Обучающая подвыборка используются для настройки моделей, валидационная нужна для их сравнения на метриках качества, а итоговая мощность выбранной модели рассчитывается на основе на тестовой. Кросс-валидация не будет использоваться, поскольку ее работа требует существенных временных затрат, в отличии от рассмотренного выше способа.

Как было показано в обзоре НТИ, наилучшими алгоритмами в среднем для задачи классификации целенаправленной физической активности являются: SVM, Random Forest. Несмотря на проявленную эффективность SVM, он имеет существенные минусы, которые мешают его использованию в решаемой задачи, а именно:

- сильная неустойчивость к аномалиям в данных;
- сложность выбора спрямляющего пространства для признаков (в т.ч. выбора ядра) и настройки гиперпараметров для увеличения мощности алгоритма;
- свойственно переобучение в виду принципов работы алгоритма;

Так как рассматриваемые данные, ввиду способа их получения, имеют много выбросов, то для качественной работы метода опорных векторов эти аномалии нужно будет удалять, а это может плохо сказаться на полезности датасетов для обучения модели. Алгоритм Random Forest не имеет вышеописанных недостатков, он переобучается гораздо меньше, крайне эффективен при гиперпараметрах по умолчанию и не чувствителен к аномалиям в данных. К сожалению, при изменении гиперпараметров существенного увеличения мощности модели не произойдет ввиду особенностей работы алгоритма.

Алгоритм CatBoost, основанный на градиентном бустинге, является потенциально мощнейшим из рассматриваемого списка, так как не имеет проблем, актуальных для SVM и RF и обладает потенциально большей предсказательной силой. Стоит отметить, что научных работ по теме классификации целенаправленной физической активности рабочих этим алгоритмом нет, однако, он довольно новый и вполне вероятно, что в ближайшем будущем такие работы появятся. Для рассмотрения были выбраны алгоритмы Random Forest и CatBoost Classifier с целью выявить наилучший.

Для сравнения моделей и несмещенной оценки выбранной использованы те метрики, которые встречаются в научных работах наиболее часто: accuracy, precision, recall, confusion matrix, f1-score.

Вывод статистической результирующей классификационной информации будет осуществлен посредством использования гистограммы распределения данных по предсказанным меткам.

7 Построение модели решения задачи

Рассмотрим модель построения решения задачи. В качестве входных потоков выступают два датасета с телеметрией рабочих.

Закодируем входные потоки в модель следующим образом:

f_i - i-ый датасет со всеми признаками.

Верхний индекс датасета означает первую букву последней совершенной над ним операции. Например, обозначение f_1^F означает 1-ый датасет, обозначение последней операции над которым начиналось с буквы F

Таблица 12 - Операции, совершаемые над входными данными.

Обозначение операции	Входные параметры	Выходные	Действие
Analysis	f_1, f_2	u_feature (бесполезные признаки)	Проверяет входные данные на наличие неинформативных признаков
Format	f_1,f_2,u_feature	f_1^F, f_2^F	Удаляет неинформативные признаки из наборов данных, обрабатывает признак “Шаги” и преобразует метки классов в соответствии с схемой __
Normalize	f_1^F, f_2^F	f_1^N, f_2^N	Нормализует столбцы датасетов с целью приведения данных к общему масштабу

Продолжение таблицы 12.

Обозначение операции	Входные параметры	Выходные	Действие
Gen Features	f_1^N, f_2^N	f_1^{GF}, f_2^{GF}	Генерация новых признаков из исходных и обработка зависимого столбца
Agg	f_1^{GF}, f_2^{GF}	f^A	Два датасета объединяются в один
Train Test Split	f^A	f_1^{TTS}, f_2^{TTS}, f_3^{TTS}	Разбиение исходных данных на 3 подвыборки: 1 - обучающая, 2 - валидационная, 3 - тестовая
CatBoost Learning Check	f_1^{TTS}, f_2^{TTS}	r^{CLC} - результаты классификации моделью тестовой выборки на выбранных метриках, y^{CLC}- предсказанные метки классов, m^{CLC}- модель catboost	Происходит обучение и тестирование модели CatBoost классификатора
Label Encoder	f_1^{TTS}, f_2^{TTS}, f_3^{TT}	f_1^{LE}, f_2^{LE}, f_3^{LE}	Кодирует строковые метки классов числами
Random Forest Learning Check	f_1^{LE}, f_2^{LE}	r^{RFLC} , y^{RFLC} , m^{RFLC}	Происходит обучение и тестирование модели случайного леса

Продолжение таблицы 12.

Обозначение операции	Входные параметры	Выходные	Действие
Compare Models	r^{CLC}, r^{RFLC}	i^{CM} (название лучшего алгоритма на метриках)	Сравнение выбранных алгоритмов на основе результатов обучения
Best Model Final Check	f_3^{TTS}, f_3^{LE}, m^{RFLC}, m^{CLC}, i^{CM}	m^{BMFC}, y^{BMFC}, r^{BMFC} (результаты классификации выбранной моделью)	Если i^{CM} - CatBoost, то m^{BMFC} - модель CatBoost, иначе m^{BMFC} - Random Forest. Аналогично для y^{BMFC} . Операция осуществляет несмещенную оценку модели на тестовой выборке для определения ожидаемой мощности.
Generate Predict Visualization	y^{BMFC}	v^{BMFC}	Генерирует визуализацию распределения классов для лучшей интерпретации результатов

8 Алгоритм решения задачи

На основе информационной модели, построенной выше, был составлен алгоритм, который описывает этапы решения задачи в нотации диаграммы деятельности UML:

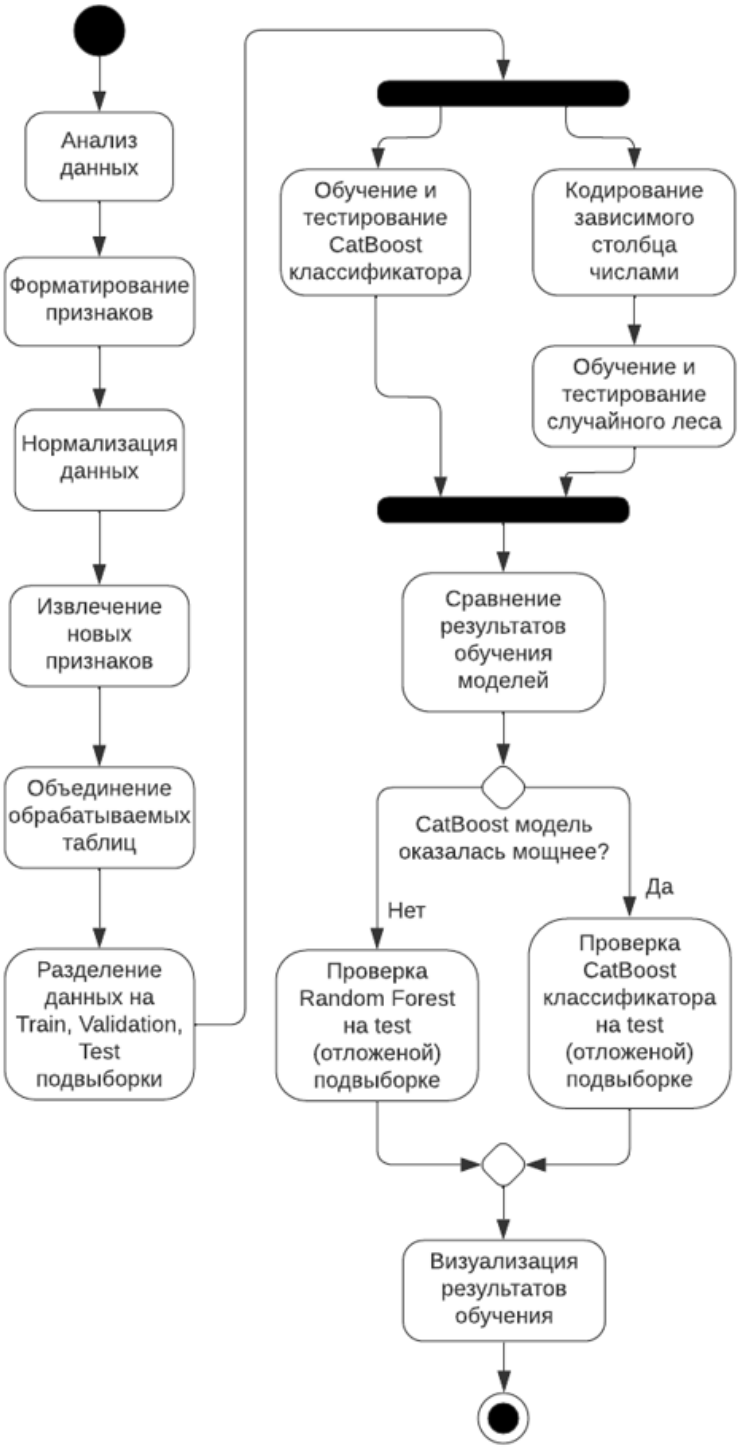

Рисунок 23 - Алгоритм решения задачи.

По результатам проведенной работы алгоритм программного решения, который будет функционировать в предметной области представлен на рисунке 36.

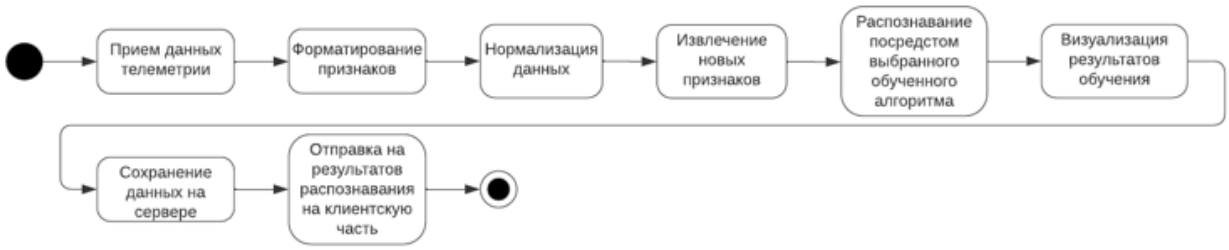

Рисунок 24 – Алгоритм функционирование программного решения в предметной области.

9 Программная реализация

Для реализации программного решения был использован: дистрибутив языка Python - Anaconda 3, так как он является одним из самых мощных инструментов анализа, обработки данных и построения моделей машинного обучения, так как обладает большим количеством высокоэффективных библиотек, предназначенными для этих целей:

Scipy - используется для генерирования новых признаков

Numpy - используется для матричных операций

CatBoost - использование моделей машинного обучения, которые строятся на основе алгоритмов градиентного бустинга

Scikit-learn - используется для построения моделей Random Forest, нормализации данных, деления данных на train, valid, test подвыборки, использования Label Encoder на зависимом столбце и выбора метрик

Pandas - работа с таблицами телеметрии: их форматирование, объединение, удаление столбов, применение метода агрегации данных по извлекаемым статистикам.

Statsmodels - для проверки статистических свойств данных

Seaborn и Matplotlib - библиотеки используется для визуализации confusion matrix, построения диаграммы предсказанных меток и анализа данных.

Mglearn – визуализация матрицы ошибок

Проведем анализ данных:

Загрузим первоначальные данные. Выведем статистические признаки:

Данные первого датасета:

	Широта	Долгота	Высота	Скорость	Точность	Ax
count	98836.0	98836.0	98836.0	98836.0	98836.0	98836.000000
mean	0.0	0.0	0.0	0.0	0.0	5.708649
std	0.0	0.0	0.0	0.0	0.0	5.028641
min	0.0	0.0	0.0	0.0	0.0	-29.400999
25%	0.0	0.0	0.0	0.0	0.0	2.049000
50%	0.0	0.0	0.0	0.0	0.0	6.148000
75%	0.0	0.0	0.0	0.0	0.0	9.807000
max	0.0	0.0	0.0	0.0	0.0	39.208000

	Ay	Az	Сердечный_Ритм	Шаги
count	98836.000000	98836.000000	98836.000000	98836.000000
mean	-4.455635	0.263577	69.864163	1033.377271
std	4.227270	4.912282	9.283915	726.247543
min	-39.228001	-39.208000	0.000000	0.000000
25%	-7.623000	-2.758000	64.007660	472.000000
50%	-4.654000	-0.114000	68.100480	752.000000
75%	-1.149000	3.466000	72.297104	1718.000000
max	39.208000	39.228001	101.698270	2424.000000

Данные второго датасета

	Широта	Долгота	Высота	Скорость
count	100490.000000	100490.000000	100490.000000	100490.000000
mean	55.552693	37.487491	3.921107	0.000006
std	3.120135	2.105499	25.518527	0.000097
min	0.000000	0.000000	0.000000	0.000000
25%	55.727449	37.606591	0.000000	0.000000
50%	55.727449	37.606591	0.000000	0.000000
75%	55.727449	37.606591	0.000000	0.000000
max	55.737360	37.610891	171.000000	0.001572

	Точность	Ax	Ay	Az
count	100490.000000	100490.000000	100490.000000	100490.000000
mean	483.860086	4.815008	-5.213053	1.522117
std	104.188903	4.874701	3.494547	4.193946
min	0.000000	-35.223999	-36.852001	-29.726999
25%	500.000000	1.398000	-8.083000	-1.206000
50%	500.000000	5.822000	-5.248000	1.896000
75%	500.000000	8.887000	-2.547000	4.137000
max	700.000000	39.151001	35.263000	35.435001

	Сердечный_Ритм	Шаги
count	100490.000000	100490.000000
mean	47.321643	1142.734183
std	43.411542	636.735857
min	0.000000	302.000000
25%	0.000000	564.000000
50%	66.011360	946.000000
75%	88.362970	1728.000000
max	115.662650	2348.000000

Рисунок 25 - Статистические характеристики изучаемых данных.

Как видно из сводной таблицы, признаки “Широта”, “Долгота”, “Высота”, “Точность” “Скорость” не имеют в первом датасете значений, отличных от нуля. Кроме того, их связь с активностью рабочих ставится под сомнение в виду смыслового значения этих признаков, за исключением признака “Скорость”. К сожалению, сохранение признака “Скорость” также не имеет смысла, поскольку в обоих датасетах его значения практически все нулевые. Признаки, кодирующие дату, как правило, используются для агрегации данных. Выведем графики признаков “Дата” и “Дата_получения_локации”:

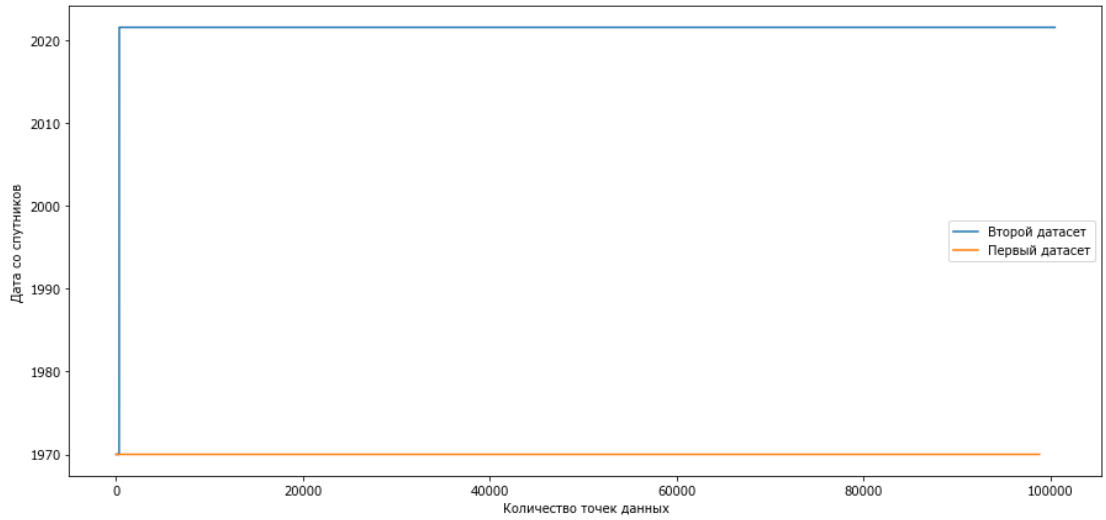

Рисунок 26 – Распределение признака “Дата_получения_локации” в датасетах.

Как видно из графика, этот признак можно не использовать для дальнейшей обработки, так как он не содержит полезной информации. Рассмотрим признак “Дата”:

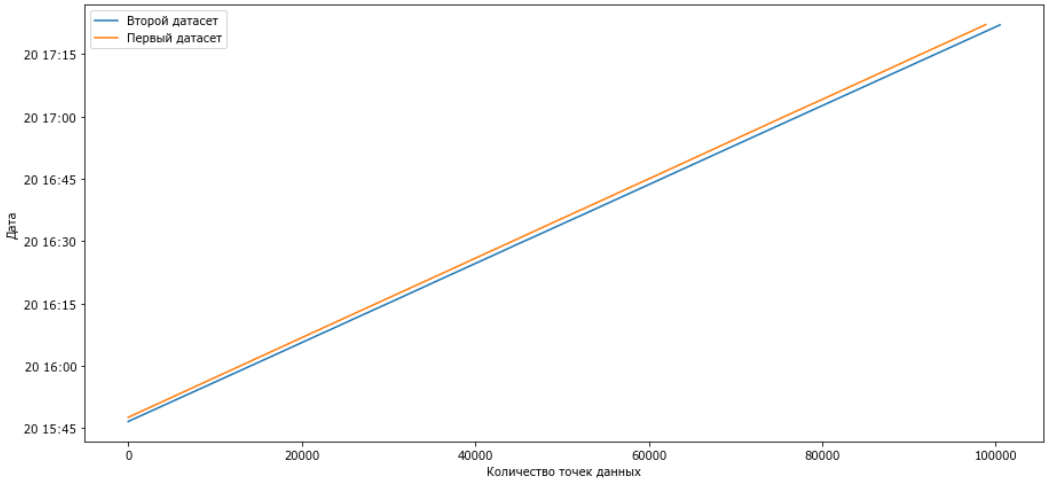

Рисунок 27 – Распределение признака “Дата” в датасетах.

Как видно из графика, данные двух датасетов записывались равномерно на протяжении некоторого времени. Временной масштаб не соответствует действительному, а значит считан устройством мониторинга неправильно. Однако, для агрегации данных можно использовать информацию о равномерности записи данных. Этой информации для дальнейшей работы достаточно и сохранять признак не имеет смысла.

Кроме того, в данных второго датасета 25 квантиль признака “Сердечный_ритм” равен нулю. Для большей наглядности рассмотрим признаки “Ax”, “Ay”, “Az”, “Сердечный_ритм” на графике в объединении по датасетам на Q-Q графике:

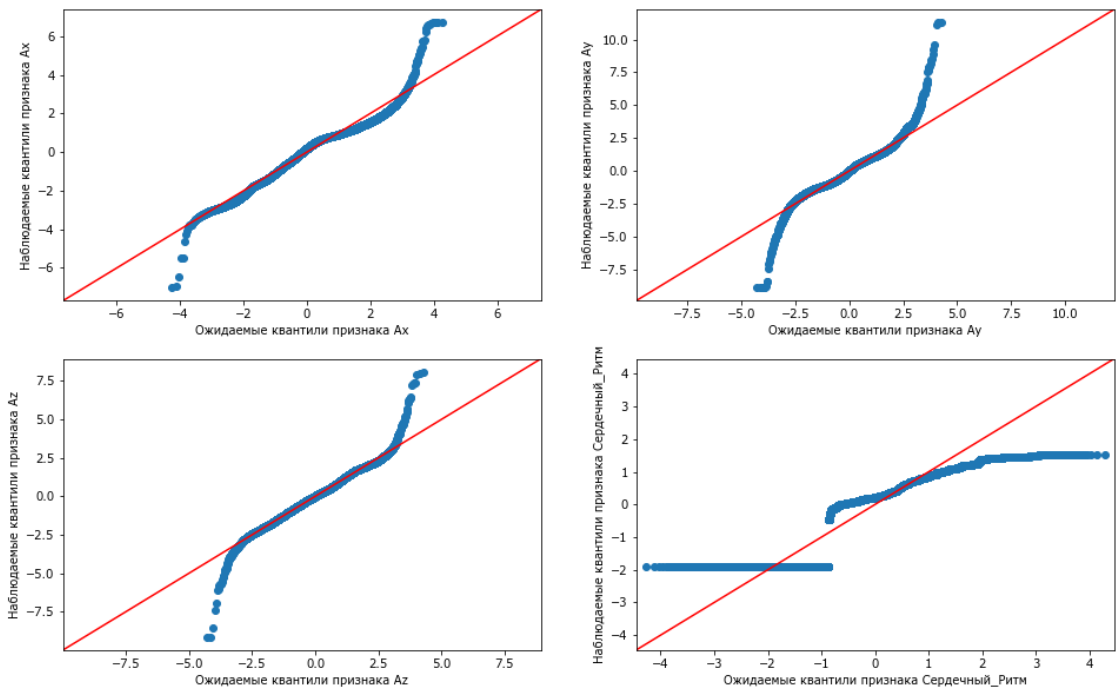

Рисунок 28 - Q-Q графики для признаков: “Ax”, “Ay”, “Az”, “Сердечный_ритм”.

Наблюдаемые данные сравнивались с ожидаемыми. Ожидаемые данные строились относительно предположения о нормально данных изначальных. Видно, что наблюдаемые данные слабо походят на нормально распределенные. Признак “Сердечный_ритм” имеет чуть меньше половины значений, равных нулю, что крайне плохо для предсказания активности рабочих.

Рассмотри признак “Шаги” отдельно:

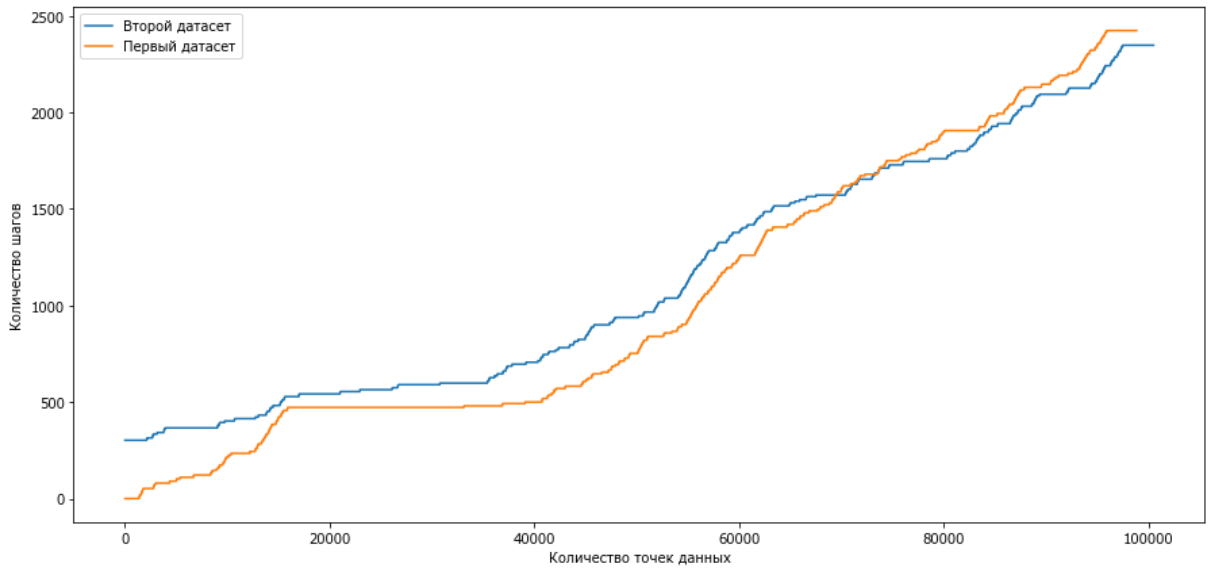

Рисунок 29 - Признак “Шаги” в двух датасетах.

Проведем форматирование признаков:

Перекодируем метки классов, в соответствии с таблицей 9. Удалим из данных те признаки, которые признаны неинформативными в соответствии с анализом данных. Обработаем признак “Шаги”. После чего нормализуем датасеты.

Для извлечение использована агрегация данных в окне, которое равно 5 строкам. Извлекаемые признаки: 25 и 75 квантили, математическое ожидание, среднеквадратичное отклонение, коэффициент асимметрии, эксцесса минимум, максимум и энтропия. Для обработки зависимого признака, в случае попадания нескольких меток классов в один интервал, используется мода, которая выбирает ту метку, которая чаще всего встречалась. Признак, по которому осуществляется агрегация, был сгенерирован из индексов строк данных. Размер окна составил 5 строк.

Обучим модель Random Forest на train подвыборке и проверим ее качество на валидационной:

Правильность на валидационном наборе: 0.72

	precision	recall	f1-score	support
0	0.63	0.84	0.72	503
1	0.86	0.66	0.75	1488
2	0.57	0.68	0.62	299
3	0.48	0.83	0.61	134
4	0.72	0.74	0.73	913
accuracy			0.72	3337
macro avg	0.65	0.75	0.69	3337
weighted avg	0.75	0.72	0.72	3337

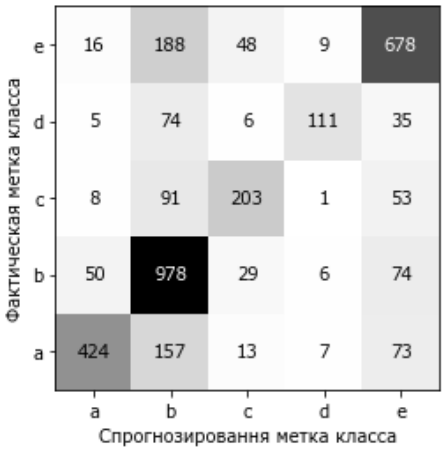

Рисунок 30 - Результаты обучения RF на валидационной выборке.

Результаты получились крайне высокими, посмотрим 10 самых важных признаков для предсказания:

```
Важность признака: ('Сердечный_Ритм', 'amax') для распознавания составляет: 0.040868761647798 32
Важность признака: ('Сердечный_Ритм', 'quantile_25') для распознавания составляет: 0.040085241853112984
Важность признака: ('Сердечный_Ритм', 'amin') для распознавания составляет: 0.03993670154615873
Важность признака: ('Сердечный_Ритм', 'mean') для распознавания составляет: 0.03920900479010118
Важность признака: ('Сердечный_Ритм', 'quantile_75') для распознавания составляет: 0.03888691089981775
Важность признака: ('Сердечный_Ритм', 'mode_int') для распознавания составляет: 0.03842620617688737
Важность признака: ('Ax', 'amax') для распознавания составляет: 0.026346881808633146
Важность признака: ('Ax', 'mean') для распознавания составляет: 0.0261099574596884
Важность признака: ('Ay', 'mean') для распознавания составляет: 0.026095774945780473
Важность признака: ('Ax', 'quantile_75') для распознавания составляет: 0.02562247974089227
```

Рисунок 31 - 10 важнейших признаков для обучения RF.

Обучим CatBoost классификатор:

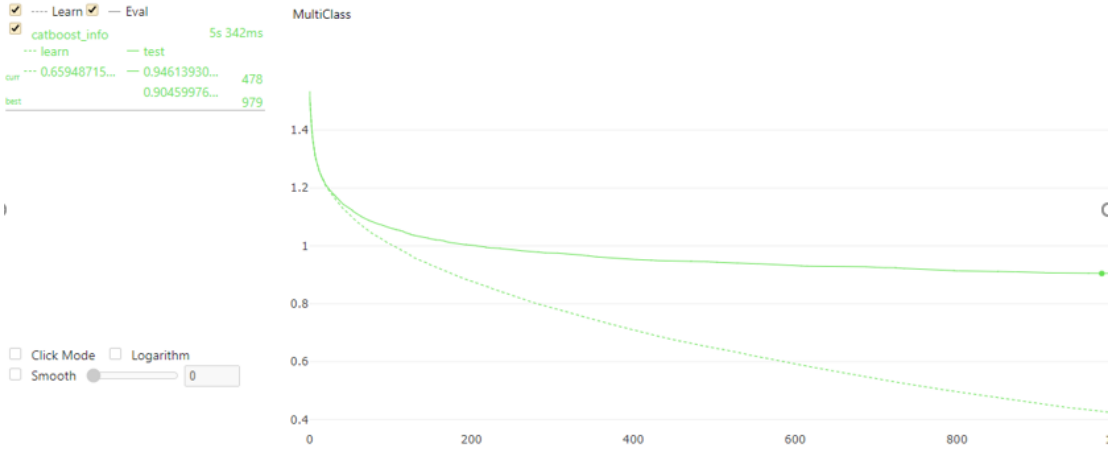

Рисунок 32 - Процесс обучения CatBoost классификатора. В качестве функции ошибки показана LogLoss.

Правильность на валидационном наборе: 0.66

	precision	recall	f1-score	support
a	0.57	0.75	0.65	508
b	0.79	0.61	0.69	1467
c	0.49	0.61	0.54	289
d	0.39	0.66	0.49	137
e	0.68	0.68	0.68	936
accuracy			0.66	3337
macro avg	0.58	0.66	0.61	3337
weighted avg	0.68	0.66	0.66	3337

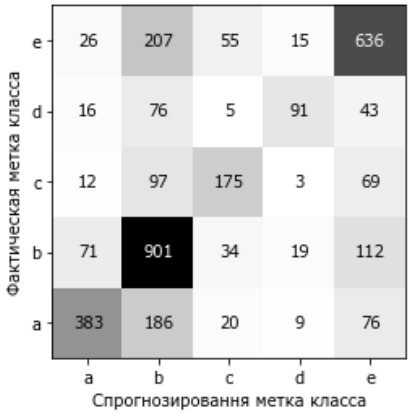

Рисунок 33 - Результаты обучения CatBoost классификатора на валидационной выборке.

10 важнейших признаков для обучения CatBoost представлены на рисунке 42.

```
Важность признака: ('Сердечный_Ритм', 'amax') для распознавания составляет: 6.034646509272406
Важность признака: ('Сердечный_Ритм', 'amin') для распознавания составляет: 5.869710484940761
Важность признака: ('Сердечный_Ритм', 'mean') для распознавания составляет: 3.491512459233916
Важность признака: ('Сердечный_Ритм', 'quantile_75') для распознавания составляет: 3.3877146429451064
Важность признака: ('Шаги', 'amax') для распознавания составляет: 3.219201959119424
Важность признака: ('Ax', 'std') для распознавания составляет: 3.0108764150074827
Важность признака: ('Ay', 'quantile_75') для распознавания составляет: 2.961757460726812
Важность признака: ('Шаги', 'mean') для распознавания составляет: 2.9158146844534842
Важность признака: ('Az', 'std') для распознавания составляет: 2.8175818810422197
Важность признака: ('Ay', 'amax') для распознавания составляет: 2.776177783792413
```

Рисунок 34 - 10 важнейших признаков для обучения CatBoost классификатора.

Как видно из результатов обучения, Random Forest в данной задаче оказался эффективней CatBoost классификатора.

Качество на тестовой (отложенной) выборке:

Правильность на тестовом наборе: 0.72

	precision	recall	f1-score	support
0	0.63	0.86	0.73	496
1	0.87	0.66	0.75	1456
2	0.56	0.69	0.62	304
3	0.45	0.71	0.55	151
4	0.72	0.73	0.73	864
accuracy			0.72	3271
macro avg	0.65	0.73	0.67	3271
weighted avg	0.75	0.72	0.72	3271

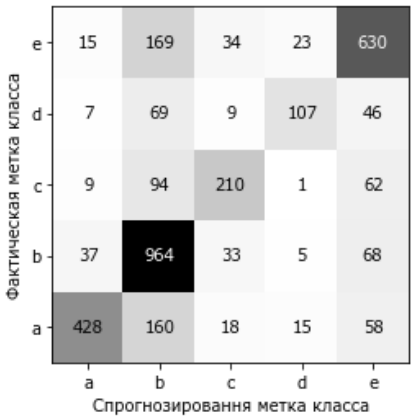

Рисунок 35 - Результаты случайного леса на тестовой выборке.

Результирующая статистика может отправляться в следующем виде:

```
Декодирование меток классов:
a - простой, b - перемещение,
c - низкая мобильная активность
d - высокая мобильная активность
e - низкая статическая активность,
f - высокая статическая активность
Метка - встречаемость:
b    1456
e     864
a     496
c     304
d     151
dtype: int64
```

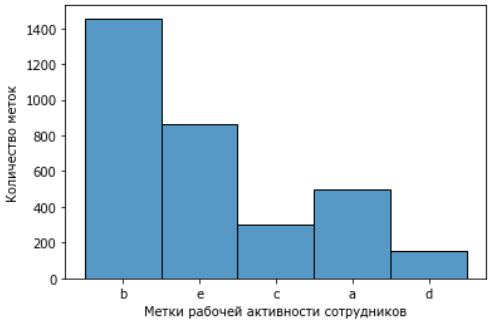

Рисунок 36 - Результирующая статическая характеристика классификации меток.

ЗАКЛЮЧЕНИЕ

В работе была поставлена задача разработки модели идентификации целенаправленной физической активности рабочих.

В ходе работы был проведен анализ предметной области и его подпроцесса, отвечающего за мониторинг целенаправленной физической активности, а также были раскрыты технические, программные и информационные обеспечения, используемые в предметной области. Были построены UML диаграммы процессов: управления целенаправленной физической активностью сотрудников в общем виде и мониторинга целенаправленной физической (AS-IS, TO-BE). TO-BE модель реализуется при интеграции предложенного программного решения.

Для теоретических основ решения задачи был проведен обзор НТИ и анализ предложенных в нем методов построения программного решения, относительно специфики представленной задачи. Были изучены способы анализа, обработки данных, выбраны метрики качества классификации, а также модели машинного обучения для сравнения.

По результатам построения теоретических основ был проведен анализ и обработка собранной телеметрии рабочих, сравнивалось качество моделей машинного обучения, по результатам которого выбиралась наилучшая модель. После чего ее качество измерялось на тестовой подвыборке и строилась гистограмма распределения меток, которую можно использовать управляющими низшего звена для анализа производственной активности.

Данные, считанные с устройств мониторинга, записывались в условиях реальной работы и размечались вручную. По этим причинам датасеты содержат много аномалий, особенно в признаках, кодирующих временную информацию и сердцебиение человека. Однако, алгоритм случайного леса дал точность прогноза, равную 72%. При уменьшении количества аномалий в данных и большем количестве телеметрии точность предсказания повысится значительно.

При внедрении разработанного решения в мониторинг производственной активности уровень автоматизации на предприятии повысится, результатом чего будет служить увеличение скорости производства и улучшение качества выпускаемой продукции в процессе управления целенаправленной физической активностью рабочих.

Способы анализа и обработки данных, предложенные в данной работе, а также результаты сравнения моделей CatBoost и Random Forest в задаче классификации целенаправленной физической активности могут быть полезны для будущих работ по схожим темам.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Штехин С.Е. Карачёв Д.К. Иванова Ю.К. Разработка алгоритма распознавания движения человека методами компьютерного зрения в задаче нормирования рабочего времени // Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 1, 2020
2. Hamad Al Jassmi, Mahmoud Al Ahmad and Soha Ahmed. Automatic recognition of labor activity: a machine learning approach to capture activity physiological patterns using wearable sensors // Emerald publishing - Construction Innovation, vol. 21, issue 4, 2021
3. Reza Akhavian. Construction activity recognition for simulation input modeling using machine learning classifiers // 2014 Winter Simulation Conference - (WSC 2014)
4. Tomoyuki Gondo, Reiji Miura. Accelerometer-Based Activity Recognition of Workers at Construction Sites // Front. Built Environ., 09 September 2020
5. Reza Akhavian, Amir H. Behzadan. Smartphone-based construction workers activity recognition and classification // Elsevier Volume 71, Part 2, November 2016, Pages 198-209
6. Ibrahim Karataş, Abdulkadir Budak. Prediction of Labor Activity Recognition in Construction with Machine Learning Algorithms // Icontech international journal, vol. 5, No. 3, 38-47, 2021.
7. Behnam Sherafat; Changbum R. Ahn; Reza Akhavian; Amir H. Behzadan; Mani Golparvar-Fard; Hyunsoo Kim; Yong-Cheol Lee; Abbas Rashidi; and Ehsan Rezazadeh Azar. Automated Methods for Activity Recognition of Construction Workers and Equipment: State-of-the-Art Review // Journal of Construction Engineering and Management, vol. 146, no. 6, 2020
8. Muhammad Shoaib, Stephan Bosch, Ozlem Durmaz Incel, Hans Scholten, Paul J. M. Havinga. Fusion of Smartphone Motion Sensors for Physical Activity Recognition // Sensors 2014, vol. 14, no. 6, 10146-10176
9. Akram Bayat, Marc Pomplun, Duc A. Tran. A Study on Human Activity Recognition Using Accelerometer Data from Smartphones // Procedia Computer Science, vol. 34, 2014, pages 450-457
10. Pierluigi Casale, Oriol Pujol, Petia Radeva. Human Activity Recognition from Accelerometer Data Using a Wearable Device // Iberian Conference on Pattern Recognition and Image Analysis, IbPRIA 2011: Pattern Recognition and Image Analysis pp 289–296

11. Matthew N., Ahmadi Toby G. Pavey, и Stewart G. Trost. Preschool Children Machine Learning Models for Classifying Physical Activity in Free-Living // Sensors 2020, vol. 20, no 6, 4364
12. Jeffer Eidi Sasaki. Development and Validation of Accelerometer-Based Activity Classification Algorithms for Older Adults: A Machine Learning Approach // Doctoral Dissertations, 42, 2014
13. Andrea Mannini and Angelo Maria Sabatini. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers // Sensors 2010, 10(2), 1154-1175
14. Matthew Ahmadi. Comparison of hip and wrist accelerometers in a pre-adolescent population in free-living and semi-structured physical activity // 2016, Masters Theses, 404
15. Ahmed Younes Shdefat, Ahmed Abu Halimeh1 and Hee-Cheol Kim. Human Activities Recognition Via Smartphones Using Supervised Machine Learning Classifiers // Prim Health Care 8: 289, 2018
16. Thomas Bastian, Aurélia Maire, Julien Dugas, Abbas Ataya, Clément Villars, Florence Gris, Emilie Perrin, Yanis Caritu, Maeva Doron, Stéphane Blanc, Pierre Jallon, and Chantal Simon. Automatic identification of physical activity types and sedentary behaviors from triaxial accelerometer: Laboratory-based calibrations are not enough // Journal of Applied Physiology, volume 118, issue 6, march 2015, pages 716-722
17. Manuel Gil-Martín, Rubén San-Segundo, Fernando Fernández-Martínez, Javier Ferreiros. Time Analysis in Human Activity Recognition // Neural Processing Letters volume 53, pages 4507–4525, 2021
18. Е. С. Тарантова, К. В. Макаров. Выбор признаков и метода классификации видов физической активности в задаче построения телереабилитационной системы // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 43–57.
19. Е.С. Тарантова, К.В. Макаров, А.А. Орлов. Обзор подходов и практических областей применения распознавания видов физической активности // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. 2019. Т. 8, № 3. С. 43–57.

20. Alok Kumar Chowdhury, Dian Tjondronegoro, Vinod Chandran, Stewart G Trost. Ensemble Methods for Classification of Physical Activities from Wrist Accelerometry // Med Sci Sports Exerc. 2017, 49(9):1965-1973
21. И. Н. Стебаков, Д.В. Шутин, Н.А. Марахин. Машинное обучение в реабилитационной медицине и пример классификатора движений пальцев для кистевого тренажера // Инженерный вестник Дона, №6, 2020
22. Silke Janitza, Roman Hornung. On the overestimation of random forest's out-of-bag error // PLoS ONE 13(8): e0201904
23. Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin. CatBoost: gradient boosting with categorical features support // arXiv:1810.11363v1 [cs.LG] 24 Oct 2018
24. Asghar Ghasemi, Saleh Zahedias. Normality Tests for Statistical Analysis: A Guide for Non-Statisticians // Int J Endocrinol Metab. 2012;10(2):486-9.
25. Zhifeng Dai and Xiaoming Chang. Predicting Stock Return with Economic Constraint: Can Interquartile Range Truncate the Outliers? // Mathematical Problems in Engineering, volume 2021, article ID 9911986, 12 pages
26. Rovetta A. Raiders of the Lost Correlation: A Guide on Using Pearson and Spearman Coefficients to Detect Hidden Correlations in Medical Sciences // Cureus 12(11): e11794
27. Hulin Kuang, PhD; Wu Qiu; Anna M. Boers; Scott Brown; Keith Muir; Charles B.L.M. Majoie; Diederik W.J. Dippel; Phil White; Jonathan Epstein; Peter J. Mitchell; Antoni Dávalos; Serge Bracard; Bruce Campbell; Jeffrey L. Saver; Tudor G. Jovin; Marta Rubiera; Alexander V. Khaw; Jai J. Shankar; Enrico Fainardi; Michael D. Hill; Andrew M. Demchuk; Mayank Goyal; Bijoy K. Menon. Computed Tomography Perfusion–Based Machine Learning Model Better Predicts Follow-Up Infarction in Patients with Acute Ischemic Stroke // Stroke, january 2021 vol 52, issue 1

Приложение А

Листинг программы.

```
## Подключение библиотек
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib as mpl
import mglearn
from sklearn.preprocessing import MinMaxScaler, RobustScaler, LabelEncoder,
StandardScaler
from scipy.stats import normaltest, probplot, poisson, chisquare, shapiro,
ttest_ind, chi2_contingency
import scipy
from scipy import stats
import statsmodels.api as sm
from sklearn.model_selection import cross_val_score
plt.rc("font", family="Verdana") # кириллица
from sklearn.feature_selection import f_classif
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import classification_report, confusion_matrix,
roc_auc_score
from sklearn.model_selection import train_test_split
import catboost as cb
from sklearn.metrics import log_loss

## Считывание изучаемых данных
table_Pasha = pd.read_excel("./datasets/Разметка паша.xlsx")
table_Max = pd.read_excel("./datasets/Разметка макс.xlsx")
table_Pasha.columns = table_Max.columns = ["Дата", "Дата_получения_локации",
"Широта", "Долгота", "Высота", "Скорость", "Точность", "Ax",Ay", "Az",
"Сердечный_Ритм", "Шаги", "Разметка"]
print("Измененные колонки для \nПервого датасета {}, \nВторого датасета
{}".format(list(table_Pasha.columns),

list(table_Max.columns)))

## Статистические характеристики изучаемых данных
with pd.option_context('display.max_rows', None, 'display.max_columns',
None):
    print("Данные первого датасета:\n{}\n\n\nДанные второго датасета\n{}"
          .format(table_Max.describe(), table_Pasha.describe()))

## Распределение признака "Дата_получения_локации" в датасетах
plt.figure(figsize=(15, 7))
table_Pasha.Дата_получения_локации.plot(label="Второй датасет")
table_Max.Дата_получения_локации.plot(label="Первый датасет")
plt.legend();
plt.xlabel("Количество точек данных")
plt.ylabel("Дата со спутников");

## Распределение признака "Дата" в датасетах
plt.figure(figsize=(15, 7))
table_Pasha.Дата.plot(label="Второй датасет")
table_Max.Дата.plot(label="Первый датасет")
```

```
plt.legend();
plt.xlabel("Количество точек данных")
plt.ylabel("Дата");

## Q-Q графики для признаков: “Ax”, “Ay”, “Az”, “Сердечный_ритм”
corr_table = pd.concat([table_Pasha.drop(["Разметка", "Шаги"], axis=1),
table_Max.drop(["Разметка", "Шаги"], axis=1)], axis=0)
fig, axes = plt.subplots(nrows=2, ncols=2, figsize=(16, 10))
for idx, x in enumerate(corr_table):
    pp = sm.ProbPlot(corr_table[x].values, fit=True);
    pp.qqplot(line="45", ax=axes[idx // 2, idx % 2])
    axes[idx // 2, idx % 2].set_xlabel("Ожидаемые квантили признака {}
".format(x))
    axes[idx // 2, idx % 2].set_ylabel("Наблюдаемые квантили признака
{}".format(x))

## Признак “Шаги” в двух датасетах
plt.figure(figsize=(15, 7))
table_Pasha.Шаги.plot(label="Второй датасет")
table_Max.Шаги.plot(label="Первый датасет")
plt.legend();
plt.xlabel("Количество точек данных")
plt.ylabel("Количество шагов");

## Перекодирование меток классов в соответствии с таблицей 9
def change_class_inplace(df):
    df.replace({'Разметка': 'c'}, 'e', inplace=True)
    df.replace({'Разметка': 'd'}, 'c', inplace=True)
    df.replace({'Разметка': 'f'}, 'e', inplace=True)
    df.replace({'Разметка': 'g'}, 'd', inplace=True)
    df.replace({'Разметка': 'h'}, 'f', inplace=True)
    df.replace({'Разметка': 'i'}, 'd', inplace=True)
    df.replace({'Разметка': 'j'}, 'c', inplace=True)
    df.replace({'Разметка': 'k'}, 'a', inplace=True)
    loss = df.query('Разметка == "l"')
    df.drop(df[df.Разметка == "l"].index, inplace=True)
    df.reset_index(drop=True, inplace=True)
    return loss

pasha_loss = change_class_inplace(table_Pasha)
max_loss = change_class_inplace(table_Max)

## Удаление неинформативных признаков
table_Pasha.drop(axis=1, inplace=True, columns=["Широта", "Долгота", "Дата",
"Высота", "Скорость",
                                                "Точность",
'Дата_получения_локации'])
table_Max.drop(axis=1, inplace=True, columns=["Широта", "Долгота", "Дата",
"Высота", "Скорость",
                                              "Точность",
'Дата_получения_локации'])

## Обработка признака “Шаги”
table_Pasha_new = table_Pasha.copy()
table_Max_new = table_Max.copy()

p_f_u = [int(x) for x in table_Pasha.Шаги.unique()]
m_f_u = [int(x) for x in table_Max.Шаги.unique()]
```

```
def foot(c, f_u):
    c = int(c)
    if c != f_u[0]:
        return c - f_u[f_u.index(c) - 1]
    else:
        return f_u[0]

table_Pasha_new["Шаги"] = table_Pasha.Шаги.apply(foot, args=(p_f_u,))
table_Max_new["Шаги"] = table_Max.Шаги.apply(foot, args=(m_f_u,))
table_Pasha_new.replace({"Шаги": 502}, 2, inplace=True)

## Нормализация датасетов
scaled_features =
StandardScaler().fit_transform(table_Pasha_new.drop(["Разметка"], axis=1))
table_Pasha_new_s = pd.concat([pd.DataFrame(scaled_features,

columns=table_Pasha_new.drop(["Разметка"], axis=1).columns),
                              table_Pasha_new[["Разметка"]]], axis = 1)

scaled_features =
StandardScaler().fit_transform(table_Max_new.drop(["Разметка"], axis=1))
table_Max_new_s = pd.concat([pd.DataFrame(scaled_features,

columns=table_Max_new.drop(["Разметка"], axis=1).columns),
                            table_Max_new[["Разметка"]]], axis = 1)

## Генерация новых признаков
def agg_interval(df, sample_time="5S"):
    def mode_int(x):
        return stats.mode(x).mode

    def quantile_25(x):
        return np.quantile(x, .25)

    def quantile_75(x):
        return np.quantile(x, .75)

    def str_func(x):
        if len(x) <= 1:
            return "0" + x
        else:
            return x

    V = ""
    def mode_str(x):
        nonlocal V
        mode = str(*stats.mode(list(x)).mode)
        if not mode:
            mode = V
        else:
            V = mode
        return mode

    indexx = df.reset_index().index
    hours = indexx // 3600
    minutes = indexx // 60 - (indexx // 3600) * 60
```

```
    seconds = indexx - hours * 3600 - minutes * 60

    hms = pd.DataFrame({'hour': hours,
                        'minute': minutes,
                        'second': seconds})

    df = pd.concat([pd.DataFrame(pd.to_datetime(hms.iloc[:,
0].astype(str).apply(str_func) \
                                                + ":" + hms.iloc[:,
1].astype(str).apply(str_func) \
                                                + ":" + hms.iloc[:,
2].astype(str)\
                    .apply(str_func), format="%H:%M:%S"), columns=["Date"]), df],
axis=1)

    d = {'Разметка': mode_str}
    for x in set(df.columns) - set(["Разметка", "Date"]):
        d[x] = [mode_int, quantile_25, quantile_75, np.mean, np.std,
stats.skew, stats.kurtosis, \
                np.min, np.max, stats.entropy]
    new_df =  df.set_index("Date").resample(sample_time).agg(d)
    print("Количество пропущенных значений (столбец):", new_df.iloc[:,
3].isnull().values.sum())
    print("Всего значений (столбец):", new_df.shape[0])
    for i in new_df.columns: # итерируемся по столбцам
        new_df[i].fillna(method="ffill", inplace=True)

    return new_df

table_Pasha_new_s_feature = agg_interval(table_Pasha_new_s)
table_Max_new_s_feature = agg_interval(table_Max_new_s)

## Соединение двух обработанных таблиц
table_s_feature = pd.concat([table_Pasha_new_s_feature,
table_Max_new_s_feature], axis=0)

## Разделение данных на train, valid, test подвыборки
table_s_feature = table_s_feature.reset_index(drop=True)

X_train, X_test, y_train, y_test =
train_test_split(clean_dataset(table_s_feature.drop("Разметка", axis=1)),

table_s_feature["Разметка"],
                                                    random_state=1,
test_size=.15)
X_train, X_valid, y_train, y_valid = train_test_split(X_train, y_train,
random_state=2, test_size=.18)

## Обучение RF
y_enc = ['a', 'b', 'd', 'c', 'e']
enc = LabelEncoder()
enc.fit(y_enc)
y_train_rf = pd.DataFrame(enc.transform(y_train.values.ravel()),
columns=["Разметка"])
y_valid_rf = pd.DataFrame(enc.transform(y_valid.values.ravel()),
columns=["Разметка"])

forest = RandomForestClassifier(n_jobs=-1, random_state=1,
n_estimators=500).fit(X_train, y_train_rf.values.ravel())
```

```
print("Правильность на валидационном наборе:
{:.2f}".format(forest.score(X_valid, y_valid_rf.values.ravel())))
print(classification_report(forest.predict(X_valid), y_valid_rf))
score_image = mglearn.tools.heatmap(
                        confusion_matrix(y_valid_rf,
forest.predict(X_valid)),
                        xlabel="Спрогнозирования метка класса",
                        ylabel="Фактическая метка класса",

xticklabels=sorted(enc.classes_[sorted(y_valid_rf["Разметка"].unique())]),

yticklabels=sorted(enc.classes_[sorted(y_valid_rf["Разметка"].unique())]),
                        cmap=plt.cm.gray_r, fmt="%d")
```

10 важнейших признаков для обучения RF

```
d = {}
for i in zip(X_valid.columns, forest.feature_importances_):
    d[i[0]] = i[1]
d = dict(sorted(d.items(), key=lambda item: item[1], reverse=True))
i1 = 0
for i in d:
    i1 += 1
    print("Важность признака: {} для распознавания составляет: {}".format(i,
d[i]))
    if i1 == 10:
        break
```

Обучение CatBoost классификатора

```
model = cb.CatBoostClassifier(task_type="GPU")
model.fit(X_train,y_train, eval_set=(X_valid, y_valid), plot=True,
verbose=False)
```

Тестирование CatBoost на валидационной выборке

```
print("Правильность на валидационном наборе:
{:.2f}".format(model.score(X_valid, y_valid)))
print(classification_report(model.predict(X_valid), y_valid))
plt.figure()
score_image = mglearn.tools.heatmap(
                        confusion_matrix(y_valid, model.predict(X_valid)),
                        xlabel="Спрогнозирования метка класса",
                        ylabel="Фактическая метка класса",
                        xticklabels=sorted(y_valid.mode_str.unique()),
                        yticklabels=sorted(y_valid.mode_str.unique()),
                        cmap=plt.cm.gray_r, fmt="%d")
```

10 важнейших признаков для обучения CatBoost классификатора

```
test_pool = cb.Pool(X_valid, y_valid, feature_names=list(X_valid.columns))
d = {}
for i in zip(X_valid.columns, model.feature_importances_):
    d[i[0]] = i[1]
d = dict(sorted(d.items(), key=lambda item: item[1], reverse=True))
i1 = 0
for i in d:
    i1 += 1
    print("Важность признака: {} для распознавания составляет: {}".format(i,
d[i]))
    if i1 == 10:
```

break

```
## Тестирование RF на тестовой выборке
y_enc = ['a', 'b', 'd', 'c', 'e']
enc = LabelEncoder()
enc.fit(y_enc)
y_test_rf = pd.DataFrame(enc.transform(y_test.values.ravel()),
columns=["Разметка"])

print("Правильность на тестовом наборе: {:.2f}".format(forest.score(X_test,
y_test_rf.values.ravel())))
print(classification_report(forest.predict(X_test),
y_test_rf.values.ravel()))
score_image = mglearn.tools.heatmap(
                        confusion_matrix(y_test_rf.values.ravel(),
forest.predict(X_test)),
                        xlabel="Спрогнозированння метка класса",
                        ylabel="Фактическая метка класса",

xticklabels=sorted(enc.classes_[sorted(y_test_rf["Разметка"].unique())]),

yticklabels=sorted(enc.classes_[sorted(y_test_rf["Разметка"].unique())]),
                        cmap=plt.cm.gray_r, fmt="%d")

## Результирующая статическая характеристика классификации меток
f_p = enc.inverse_transform(forest.predict(X_test))
print("Декодирование меток классов:\na - простой, b - перемещение, \nc -
низкая мобильная активность" +
      "\nd - высокая мобильная активность\ne - низкая статическая
активность,\nf - высокая статическая активность")
print("Метка - встречаемость:\n{}\n".format(pd.value_counts(f_p)))
sns.histplot(f_p);
plt.xlabel("Метки рабочей активности сотрудников")
plt.ylabel("Количество меток");
```