МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_ Кувайскова Ю.Е.

( подпись, ФИО )

“08” июня 2023 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Прогнозирование класса объекта на основе анализа измерений беспроводных сенсорных узлов

Автор работы \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_Мазур А.Е.\_ПМбд-41\_\_\_\_\_\_

(подпись) (ФИО, группа)

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_Алексеева В.А\_\_\_\_\_\_\_\_\_

(подпись) (ФИО)

Ульяновск

2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_ Кувайскова Ю.Е.

( подпись, ФИО )

“29” декабря 2022 г.

**ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ**

студенту группы ПМбд-41 ФИСТ

\_\_\_\_\_\_\_\_\_Мазур А.Е.\_\_\_\_\_\_\_\_\_\_

(Фамилия, И.О.)

1.Тема: Прогнозирование класса объекта на основе анализа измерений беспроводных сенсорных узлов.

утверждена приказом по УлГТУ №4129 от “29” декабря 2022 г.

2. Срок представления работы к защите “15” июня 2023 г.

3. Исходная информация к работе

Данные с сенсоров о температуре, освещенности, звуке, СО2, движении, данные о занятости помещения. Задача: выявление наиболее точных методов машинного обучения для решения задачи многомерной классификации.

4. Содержание работы

Введение; методы машинного обучения: метод опорных векторов, метод наивного байеса, метод k-ближайших соседей, случайный лес, нейронные сети, градиентный бустинг, стэкинг, бэггинг, адаптивный бустинг, метрики оценки; обзор разработанной программы; описание и предобработка исходных данных; анализ результатов; заключение; список используемых источников.

5. Графические материалы

Рисунки 1-27\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

6. Приложения

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Алексеева Венера Арифзяновна

(подпись) (ФИО)

“29” декабря 2022 г.

Задание принял к исполнению “29” декабря 2022 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

УТВЕРЖДАЮ

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_Алексеева В.А «29» декабря 2022 г.

(подпись, ФИО)

**КАЛЕНДАРНЫЙ ПЛАН**

выполнения выпускной квалификационной работы

по теме «Прогнозирование класса объекта на основе анализа измерений беспроводных сенсорных узлов»

студентом группы ПМбд - 41 Мазур Александром Евгеньевичем

(ФИО)

1. Подбор и систематизации литературы, поиск в Internet, подготовка обзора, подготовка исходных данных для анализа - 14.03
2. Изучение основных закономерностей исследуемого процесса, анализ или вывод зависимостей, разработка математических моделей - 15.03
3. Разработка программного обеспечения - 20.04
4. Тестирование и отладка программы - 05.05
5. Обработка и анализ данных - 20.05
6. Подготовка выводов и рекомендаций - 25.05
7. Оформление текста работы - 01.06
8. Оформление графических материалов - 05.06
9. Рассмотрение работы на заседании кафедры - 08.06
10. Защита работы - 15.06

Студент Мазур А.Е. “29” декабря 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение

 высшего образования

«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра «Прикладная математика и информатика»

**ОТЗЫВ**

на бакалаврскую работу

обучающегося ПМбд-41  Мазур Александра Евгеньевича

группа Ф.И.О. полностью

Тема ВКР: **Прогнозирование класса объекта на основе анализа измерений беспроводных сенсорных узлов.**

В выпускной квалификационной работе Мазур А.Е. применил различные методы машинного обучения для создания модели, которая способна определять занятость помещения на основе показаний беспроводных сенсорных узлов.

Цель выпускной квалификационной работы – исследование и сравнение эффективности методов машинного обучения для решения задачи прогнозирования класса занятости помещения на основе анализа показателей беспроводных сенсоров.

Наряду с изучением теоретических основ машинного обучения в рамках выпускной квалификационной работы были изучены возможности библиотек scikit-learn, pandas, matplotlib, tkinter для создания приложения, обучающего модели машинного обучения.

В процессе работы проводилась предобработка данных с использованием библиотеки *Pandas,* было написано приложение на языке Python в среде разработки Visual Studio Code, в котором из предобработанных данных выбираются параметры на усмотрение пользователя, производится обучение и оценка работы методов машинного обучения и ансамблей методов, выводятся графики начальных данных. Итоговый результат на тестовой выборке по метрике F1Score составляет 0,99 для методов «Случайный лес» и «Градиентный бустинг деревьев решений».

Структура и выполнение выпускной квалификационной работы Мазур А.Е. полностью соответствуют заданию. Работа выполнена аккуратно и грамотно, содержит большое количество иллюстраций и расчетов, что позволяет наглядно представить ее основные результаты.

При выполнении выпускной квалификационной работы Мазур А.Е. проявил самостоятельность, упорство и настойчивость в достижении поставленной цели, способности к анализу и обобщению информации.

Считаю, что Мазур А.Е. может быть присвоена квалификация бакалавр по направлению 01.03.04 «Прикладная математика», а ВКР заслуживает оценки «отлично».

Руководитель к.т.н., доцент кафедры ПМИ \_\_\_\_\_\_\_\_\_\_\_\_\_В.А. Алексеева

« »\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

**Реферат**

**Цель** выпускной квалификационной работы – исследование и сравнение эффективности методов машинного обучения для решения задачи прогнозирования класса занятости помещения на основе анализа показателей беспроводных сенсоров.

Выпускная квалификационная работа по теме «Прогнозирование класса объекта на основе анализа показаний беспроводных сенсорных узлов» содержит 47 с., 6 ч., 27 рис., 30 источников.

Ключевые слова: МАШИННОЕ ОБУЧЕНИЕ, МЕТРИКИ, МНОГОМЕРНАЯ КЛАССИФИКАЦИЯ, МЕТОД ОПОРНЫХ ВЕКТОРОВ, МЕТОД К-БЛИЖАЙШИХ СОСЕДЕЙ, НЕЙРОННЫЕ СЕТИ, ГРАДИЕНТНЫЙ БУСТИНГ, СЛУЧАЙНЫЙ ЛЕС, АНСАМБЛИ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ, ЗАНЯТОСТЬ ПОМЕЩЕНИЯ, СИСТЕМЫ ОТОПЛЕНИЯ.

В процессе работы проводилась предобработка данных с использованием библиотеки *Pandas,* было написано приложение на языке Python в среде разработки Visual Studio Code, в котором из предобработанных данных выбираются параметры на усмотрение пользователя, производится обучение и оценка работы методов машинного обучения и ансамблей методов, выводятся графики начальных данных.

В результате предобработки были убраны штампы даты и времени из данных. Были обучены и оценены различные методы машинного обучения с использованием различных комбинаций начальных данных.

Оглавление

[Введение 7](#_Toc137555788)

[Глава 1 – Методы и метрики 9](#_Toc137555789)

[Описание используемых методов 9](#_Toc137555790)

[Наивный Байесовский классификатор 9](#_Toc137555791)

[Метод опорных векторов 10](#_Toc137555792)

[Метод k-ближайших соседей 14](#_Toc137555793)

[Нейронные сети 16](#_Toc137555794)

[Случайный лес 16](#_Toc137555795)

[Бустинг деревьев решений 19](#_Toc137555796)

[Стекинг 20](#_Toc137555797)

[Бэггинг 21](#_Toc137555798)

[Адаптивный бустинг (AdaBoost) 22](#_Toc137555799)

[Описание метрик 23](#_Toc137555800)

[Глава 2 – Программная реализация 26](#_Toc137555801)

[Краткое описание используемых библиотек 26](#_Toc137555802)

[Описание программы 26](#_Toc137555803)

[Стартовая вкладка 26](#_Toc137555804)

[Вкладка выбор данных 28](#_Toc137555805)

[Вкладка методы классификации. 31](#_Toc137555806)

[Вкладка с графиками 34](#_Toc137555807)

[Глава 3 – Данные и результаты 36](#_Toc137555808)

[Данные и результаты 36](#_Toc137555809)

[Постановка задачи 36](#_Toc137555810)

[Описание данных 36](#_Toc137555811)

[Анализ результатов 38](#_Toc137555812)

[Заключение 44](#_Toc137555813)

[Список литературы 45](#_Toc137555814)

# **Введение**

Машинное обучение – наука о разработке статистических моделей и алгоритмов, используемых компьютерными системами для выполнения задач без инструкций, полагаясь вместо этого на логические выводы и образцы. Компьютерные системы обрабатывают большие объемы данных и информации за короткие промежутки времени, помогая человеку выполнять задания с большой эффективностью, а также автоматизировать многие задачи.

Машинное обучение помогает компаниям стимулировать рост, открывать новые источники дохода и решать сложные проблемы. **Одной из таких проблем** является оптимизация потребления энергии, тратящейся на системы отопления, вентиляции и кондиционирования, в зданиях и помещениях [1]. Ранние подходы к определению занятости помещения предполагали использование камер [2], Wi-Fi [3], умной одежды [4] и RFID-меток [5], которые считаются нарушающими личное пространство человека. В последние годы исследования перешли на неинтрузивные беспроводные сенсорные узлы для определения занятости, считывающие выделения углекислого газа [6-14], угарного газа [9-10], летучих органических веществ [9-10], уровни температуры [6-7], [9-14], освещенности, звука, движения, влажности, давления [12-13], объема воздуха [14]. Алгоритмы машинного обучения и данные с сенсоров позволяют обучить компьютерную систему распознавать количество людей в помещении, и тем самым регулировать теплоподачу в помещение в зависимости от количества человек в любой момент времени.

**Цель** выпускной квалификационной работы – исследование и сравнение эффективности методов машинного обучения для решения задачи прогнозирования класса занятости помещения на основе анализа показателей беспроводных сенсоров.

В процессе выполнения работы были поставлены и выполнены следующие **задачи**:

* Изучение литературы и анализ результатов исследований в данной области.
* Рассмотрение методов машинного обучения как возможного способа распознавания количества людей в помещении.
* Программная реализация ряда методов машинного обучения.
* Анализ и сравнение результатов решения задачи классификации занятости помещения с применением различных алгоритмов машинного обучения.
* Сравнительный анализ влияния различного набора параметров на результаты работы алгоритмов машинного обучения.

**Актуальность** данной работы обусловлена тем, что большой объем энергии в зданиях приходится на обслуживание систем отопления, вентиляции и кондиционирования. Один из способов оптимизации затрат – сделать потребление энергии зависимым от количества человек в здании/помещении. Исследования в 2019 году показали, что возможна экономия до 30% электроэнергии в зданиях, где известна закономерность пребывания людей в помещении [17].

**Практическая значимость заключается** в возможности применения алгоритмов машинного обучения в компьютерных системах для оптимизации энергопотребления в зданиях.

Автоматизация и системы управления – затронутая **область исследования. Объект исследования –** способы снижения энергопотребления на отопление в помещении. **Предмет исследования** – набор алгоритмов машинного обучения, как способ обучения компьютерных систем, отвечающих за системы отопления.

# **Глава 1 – Методы и метрики**

## **Описание используемых методов**

### **Наивный Байесовский классификатор**

Наивный байесовский классификатор (Naive Bayes classifier) – это очень популярный в машинном обучении алгоритм, который в основном используется для получения базовой точности набора данных.

Теорема Байеса позволяет рассчитать апостериорную вероятность P(A | B) на основе P(A), P(B) и P(B|A):

(1.1)

где:

* P(A | B) – апостериорная вероятность (что A из B истинно)
* P(A) – априорная вероятность (независимая вероятность A)
* P(B | A) – вероятность данного значения признака при данном классе. (что B из A истино)
* P(B) – априорная вероятность при значении нашего признака. (независимая вероятность B)

**Плюсы:**

* Алгоритм быстро и легко определяет класс на тестовой выборке данных и хорошо справляется с задачей многоклассовой классификации.
* Скорость обучения и классификации наивного байесовского классификатора лучше, чем у других простых алгоритмов. Более того, обучающих данных требуется меньше.
* По сравнению с числовыми признаками, наивный Байесовский классификатор хорошо работает с категориальными. Для числовых признаков предполагается нормальное распределение, что может быть серьезным допущением в точности нашего алгоритма.

**Минусы:**

* Если переменная в тестовом наборе данных имеет категорию, не наблюдавшуюся в обучающем наборе данных, то модель присвоит нулевую вероятность и не сможет сделать предсказание. Чтобы решить эту проблему, используются техники сглаживания, например, оценка Лапласа, являющийся одним из самых простых методов сглаживания.
* Метод *predict\_proba,* возвращающий значения спрогнозированных вероятностей, не всегда выдает достаточно точные показания.
* Предположение о независимости признаков является ограничением данного алгоритма. Однако полностью независимые признаки в реальных задачах встречаются крайне редко.

### **Метод опорных векторов**

Метод опорных векторов (Support Vector Machine) [28] – это алгоритм машинного обучения, который проецирует наблюдения в n-мерном пространстве признаков с целью нахождения гиперплоскости, разделяющей наблюдения на классы:

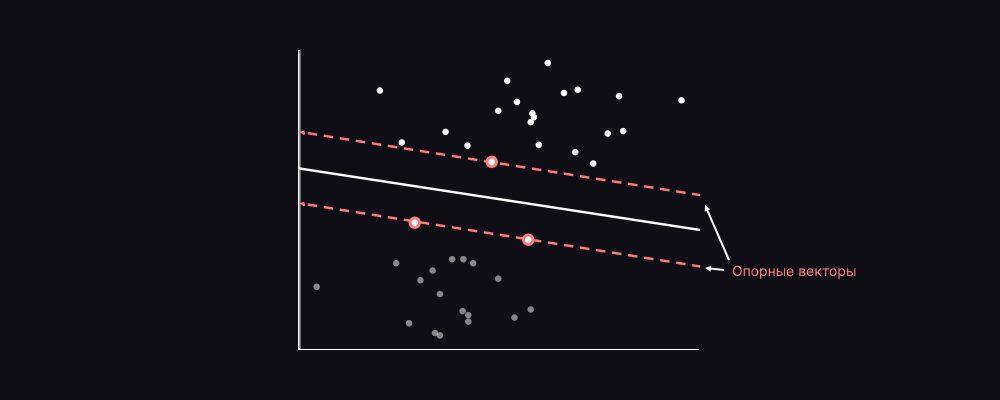


Рис. 1 – схема метода опорных векторов

Метод используется как для задач классификации, так и для задач регрессии. Наиболее часто он применяется в задачах классификации. Каждый элемент данных изображается как точка в n-мерном пространстве, где n – количество признаков, причем значение каждого элемента наблюдения является координатой в соответствующей плоскости. Затем выполняется классификация, находя разделяющую гиперплоскость, разграничивающую два класса.

Опорные векторы (обозначены на рис.1) сформированы за счет пограничных наблюдений. Классификатор SVM – это граница, лучше всего разделяющая два класса. Удаление всех точек, кроме опорных векторов, не приведет к изменению алгоритма метода опорных векторов. Это свойство отличает SVM от других методов, такие как метод k-ближайших соседей и наивный Байесовский классификатор, где каждая точка обучающего набора используется для оптимизации функции. Теоретическое различие приводит к значительному отличию между опорными векторами и другими методами в практическом плане.

Этот подход обобщается на многомерный случай.

**Линейный метод**

Даны наблюдения для обучения D – набор, состоящий из n объектов:

, (1.2)

где y принимает значения -1 или 1, определяя, к какому классу относится каждая точка xi – вектор размерности p.

Мы хотим найти гиперплоскость максимальной разности, которая разделяет наблюдения, имеющие yi = 1 от объектов yi = -1.

Любая гиперплоскость может быть записана как множество точек х, удовлетворяющих:

(1.3)

где \* - скалярное произведение нормали к гиперплоскости на вектор х.

Параметр определяет смещение гиперплоскости от начала координат вдоль нормали w.

Если обучающая выборка является линейно разделимой, можно выбрать две параллельные гиперплоскости таким образом, что они разделят множество точек на два класса, и точек между ними не останется.

Затем пытаются максимизировать дистанцию между ними, одновременно делая поворот и сдвиг параллельных прямых. Область, ограниченная 2 гиперплоскостями, называется "разностью (маржей)". Эти гиперплоскости описываются уравнениями:

(1.4)

(1.5)

Используя геометрию, находятся расстояния между гиперплоскостями - . Для того, чтобы дистанция была максимальной, нужно минимизировать . Чтобы исключить все точки из полосы, следует убедиться, что для всех наблюдений справедливо:

(1.6)

(1.7)

Далее, решается задача оптимизации:

(1.8)

**Нелинейный классификатор**

Создание нелинейного классификатора происходит путем использования произвольной функция ядра. Каждое скалярное произведение будет заменено на нелинейную функцию ядра. Таким образом, гиперплоскость максимальной разности может быть найдена в трансформированном пространстве функций. Изменение также может быть нелинейным и трансформироваться в пространство с более высокой размерностью. Даже учитывая, что классификатор является гиперплоскостью в многомерном пространстве функций, в исходном пространстве обучающей выборки он может являться нелинейным.

Функция классификации *F* принимает вид:

.

*Ядро* классификатора – выражение . С точки зрения математики, ядром может являться любая положительно определенная симметричная функция двух переменных. Положительная определенность обязательна для того, чтобы соответствующая функция Лагранжа ограничивалась снизу в задаче оптимизации, т.е. задача оптимизации была бы правильно решена. Выбор ядра диктует точность классификатора.

На практике чаще всего используются следующие ядра:

1. Полиномиальное: .
2. Радиальная базисная функция: .
3. Гауссова радиальная базисная функция: .
4. Сигмоид: .

**Плюсы:**

* это наиболее быстрый способ поиска решающих функций;
* метод сводится к решению задачи квадратичного программирования в выпуклой области, которая всегда имеет единственное решение;
* метод позволяет осуществить более надежную классификацию за счет нахождения разделяющей полосы максимальной ширины.

**Минусы:**

* стандартизация данных и шумы оказывают сильное влияние на метод;
* в случае линейной неразделимости классов, затруднительно выполнить автоматический выбор ядра, это следует делать вручную.

### **Метод k-ближайших соседей**

Метод k-ближайших соседей (k Nearest Neighbors, или kNN) – популярный алгоритм классификации, который используется в разных типах задач машинного обучения. Является одним из самых понятных методов решения задач классификации, наравне с деревом решений.

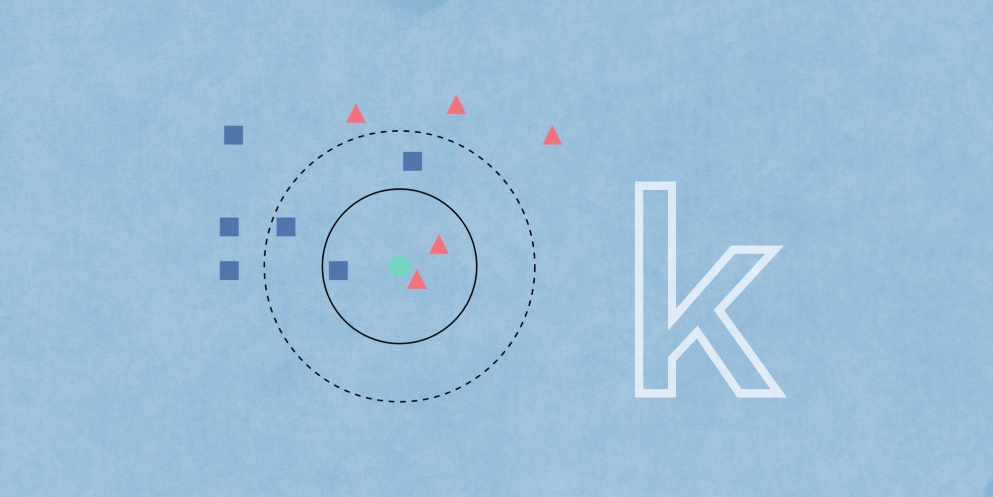


Рис. 2 – схема метода k-ближайших соседей

При использовании метода для решения задачи классификации, объекту присваивается тот класс, который является наиболее часто встречающимся среди **k ближайших** соседей данного элемента, классы которых уже известны.

Рассмотрим основные математические компоненты алгоритма **k-ближайших соседей**.

* + **Евклидова метрика** (евклидово расстояние, или же Euclidean distance) – метрика в евклидовом пространстве, расстояние между двумя точками евклидова пространства, вычисляемое по теореме Пифагора. Иными словами, это наименьшее возможное расстояние между точками **A** и **B**. Хотя евклидово расстояние хорошо применимо для малых измерений, оно не справляется с большими измерениями и категориальными переменными. Еще один недостаток евклидова расстояния заключается в игнорировании сходства между атрибутами. Каждый атрибут рассматривается как абсолютно отличный от всех остальных.

Формула вычисления Евклидова расстояния:

(1.9)

* **Нормализация** является второй важной составляющей метода. Разные атрибуты, как правило, в выборке ранжируются по-разному. К примеру, атрибут **А** находится в диапазоне от 0.1 до 0.5, а атрибут **Б** ранжирован в диапазоне от 50 до 100. В таком случае появляется сильная зависимость значения дистанции от атрибутов с бо́льшими диапазонами. Поэтому для большинства данных применяется нормализация. При кластерном анализе есть два основных способа нормализации данных: **MinMax-нормализация** и **Z-нормализация**.

**MinMax-нормализация** осуществляется следующим образом:

(1.10)

в этом случае все значения будут попадать в диапазон от 0 до 1; дискретные бинарные значения определяются как 0 и 1.

**Z-нормализация**:

(1.11)

где **σ** – среднеквадратичное отклонение. В данном случае большинство значений попадает в диапазон.

**Преимущества:**

* Алгоритм прост и легок в реализации.
* Не чувствителен к выбросам.
* Нет необходимости строить модель, настраивать несколько параметров или делать дополнительные допущения.
* Алгоритм универсален, и возможно его использование для задач как классификации, так и регрессии.

**Недостатки:**

* Скорость работы алгоритма значительно снижается при росте объема выборки, предикторов или независимых переменных.
* Следовательно, большие вычислительные затраты во время выполнения.
* Всегда необходимо находить оптимальное значение **k**.

### **Нейронные сети**

Долгое время учёные, черпая вдохновение из процессов, происходящих в мозге, пытались осуществить реверсивный инжиниринг центральной нервной системы и сделать имитацию работы человеческого мозга. Благодаря этому было рождено новое направление в машинном обучении — нейронные сети.

Математическая модель сети, которая состоит из одного нейрона, выполняет две последовательные операции:

* вычисляет сумму входных сигналов с учетом их весов (проводимости или сопротивления) связи:

(1.12)

* применяет активационную функцию к общей сумме воздействия входных сигналов:

(1.13)

Любая дифференцируемая функция может быть использована в качестве функции активации. Выбор активационной функции ложится на плечи оператора, и , как правило, этот выбор основывается на опыте решения похожих задач, или методом проб и ошибок.

### **Случайный лес**

Алгоритм случайного леса (**Random Forest**) — универсальный алгоритм машинного обучения, который строится вокруг использования ансамбля решающих деревьев. Одно решающее дерево показывает крайне низкую точность классификации, однако большое количество этих деревьев значительно улучшает результат. Также это один из немногих алгоритмов, который можно использовать в абсолютном большинстве задач.

Благодаря своей гибкости, Случайный лес применим для решения большинства проблем в области машинного обучения. Сюда можно отнести задачи классификации (**RandomForestClassifier**) и регрессии (**RandomForestRegressor**), а также задачи сложнее, вроде отбора признаков, поиска выбросов/аномалий и кластеризации.

Теория метода Случайного леса проста в сравнении с прочими методами машинного обучения, необходима только формула итогового классификатора a(x):

(1.14)

где

* **N** – количество деревьев;
* **i** – счетчик для деревьев;
* **b** – решающее дерево;
* **x** – сгенерированная нами на основе данных выборка.

*Важно отметить, что задачи классификации решаются путем голосования по большинству.*

**Преимущества:**

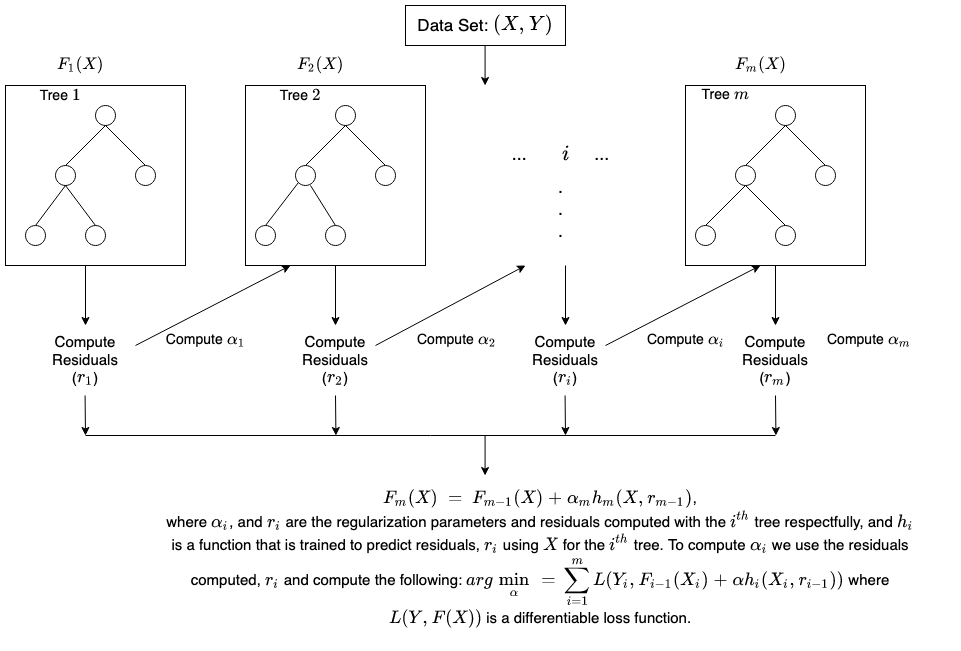
* Имеет высокую точность предсказания, которая сравнима с результатами градиентного бустинга.
* Не так важен тщательный тюнинг параметров, хорошо работает с настройками по умолчанию.
* Выбросы в данных мало влияют на работу метода благодаря случайному семплированию (**random sample**).
* Не чувствителен к масштабированию и к другим монотонным преобразованиям значений признаков.
* Редко переобучается. На практике добавление деревьев только улучшает композицию.
* Если происходит переобучение, оно преодолевается путем усреднения или объединения результатов нескольких деревьев решений.
* Эффективен в обработке данных с большим числом признаков и классов.
* Пропущенные данные не являются проблемой – метод сохраняет хорошую точность даже при их наличии.
* Одинаково хорошо обрабатывает как непрерывные, так и дискретные признаки
* Высокая параллелизуемость и масштабируемость.

**Недостатки:**

* Алгоритм случайного дерева требует высокий объем вычислительных ресурсов для реализации.
* Большой размер моделей.
* Построение случайного леса гораздо затратнее в плане времени, чем деревья решений или линейные алгоритмы.
* Зашумленные данные могут привести к переобучению алгоритма.
* Отсутствуют формальные выводы, такие как **p-values**, используемые для оценки значимости переменных.
* В отличие от более простых алгоритмов, результаты случайного леса сложнее интерпретировать.
* Когда в выборке очень много разреженных признаков, таких как тексты или наборы слов (**bag of words**), алгоритм работает хуже, чем линейные методы.
* В отличие от линейной регрессии, **Random Forest** не обладает возможностью экстраполяции. Однако в случае выбросов исчезают экстремальные значения, что можно считать плюсом.
* Есть вероятность недообучения модели при содержании в данных группы признаков с корреляцией, имеющих похожую значимость для меток, что ведет к преимуществу маленьких групп перед большими.
* Прогнозирование методом случайного леса гораздо более трудоемко по сравнению с другими алгоритмами.

### **Бустинг деревьев решений**

**Градиентный бустинг** – это продвинутый алгоритм машинного обучения для решения задач классификации и регрессии. Он делает предсказание в виде ансамбля слабых предсказывающих моделей, основу которых, как правило, составляют деревья решений. Из нескольких слабых моделей в итоге мы собираем одну, но уже точную. Общая идея алгоритма – последовательное применение предиктора (предсказателя) так, чтобы каждая следующая модель минимизировала ошибку предыдущей.



*,* (1.15)

где и – регулирующие параметры, а остатки вычисляются вместе с i-ым деревом, - функция, обученная для предсказания остатков, используя X для i-ого дерева. Для вычисления мы используем вычисленные остатки и и вычисляем следующее:

(1.16)

где L(Y, F(X)) – дифференцируемая функция потерь.

**Плюсы:**

* Алгоритм работает с любыми функциями потерь.
* Предсказания в среднем лучше, чем у других алгоритмов.
* Самостоятельно справляется с пропущенными данными.

**Минусы:**

* Алгоритм крайне чувствителен к выбросам и при их наличии будет тратить большое количество ресурсов на их обработку. Однако, стоит отметить, что использование **Mean Absolute Error (MAE)** вместо **Mean Squared Error (MSE)** значительно снижает влияние выбросов на модель.
* Слишком большое количество деревьев привет к переобучению модели. Данная проблема присутствует в любом алгоритме, связанном с деревьями, но правильная настройка параметра **n\_estimators** устраняет эту проблему.
* Вычисления могут быть затратными по времени. Поэтому, при наличии большого набора данных, обязательно следует определять правильный размер выборки и корректно настроить параметр **min\_samples\_leaf**.

### **Стекинг**

Стекинг является первым из трех используемых ансамблевых методов машинного обучения. Работа этого типа ансамблей является достаточно элементарной. Обучающий набор подается на вход всех слабых моделей, каждый прогноз идёт к финальной модели, называемой мета-учеником, после чего та делает заключительный прогноз.

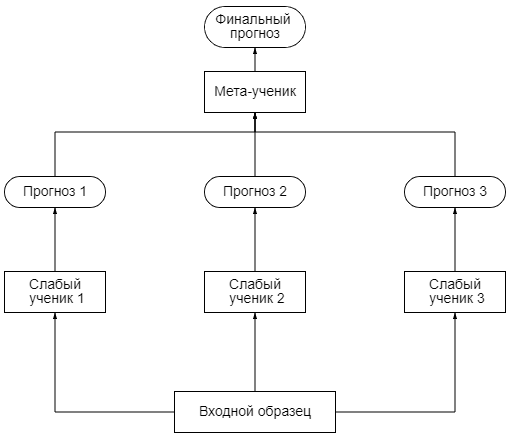


Рис. 4 – схема работы стекинга

При обучении мета-модели используется приём удерживаемого набора. Сначала обучающая выборка разделяется на 2 части. Слабые ученики поочередно обучаются на первой и второй половинах обучающей выборки. Затем, основываясь на прогнозах из первых двух частей выборки, создаётся новый обучающий набор. Таким образом, количество прогнозов на каждый входной набор соответствует количеству слабых учеников в ансамбле, а мета-модель делает прогнозы уже основываясь на новом наборе.

### **Бэггинг**

Основная идея бэггинга заключается в том, что несколько одних и тех же моделей обучаются на разных выборках. Распределение выборки неизвестно, поэтому каждая модель отличается от другой. Для начала генерируется несколько бутстрэп-выборок. Бутстрэп – это случайный выбор данных из набора данных и подстановка их в модель, затем данные возвращаются обратно и процесс повторяется. Основываясь на бутстрэп-выборках, модели делают свои предсказания. В случае регрессии происходит усреднение прогнозов. В случае же классификации применяется голосование.

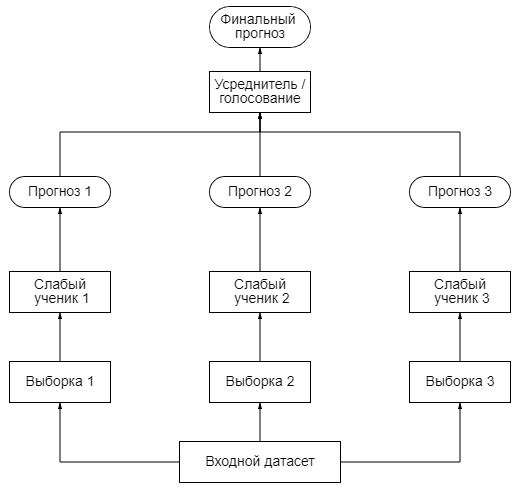


Рис. 5 – схема работы бэггинга

Есть 2 типа голосования для этого метода: жесткое и мягкое. При жестком голосовании, класс является результатом предсказания модели, если большинство слабых моделей предсказали именно этот класс. При мягком голосовании рассматриваются вероятности предсказывания каждого класса, затем вероятности усредняются и результатом является класс с большой вероятностью.

### **Адаптивный бустинг (AdaBoost)**

Данный алгоритм сначала обучает первую базовую модель (допустим деревья решений) на обучающей выборке и увеличивает относительный вес неправильно спрогнозированных значений. На вход второй базовой модели подаются обновлённые веса и модель обучается, после чего вырабатываются прогнозы и цикл повторяется.

Результат работы адаптивного бустинга – это средневзвешенная сумма каждой модели. Спрогнозировано будет то значение, которое получает большинство взвешенных голосов:

, (1.17)

где C - результат работы ансамбля, W - вес, X - значение прогнозатора.

Adaboost обновляет веса объектов на каждой итерации. Веса объектов, классифицированных правильно, уменьшаются относительно весов неправильно классифицированных объектов. Модели, которые работают лучше, имеют больший вес в окончательной модели ансамбля.

При адаптивном бустинге используется итеративный метод (добавление слабых учеников последовательно, исследуя каждую итерацию, чтобы найти наилучшее сочетание коэффициента и слабого ученика для прибавления к текущей модели ансамбля) изменения весов. Он работает быстрее, чем аналитический метод.

## **Описание метрик**

В результате работы алгоритма возможны 4 исхода:

* TP (True Positive) – истинно-положительное решение. В комнате 3 человека, и мы определили 3 человека.
* TN (True Negative) – истинно-отрицательное решение. В комнате 2 человека, и мы не стали определять это как 3 человека.
* FP (False Positive) – ложно-положительное решение. В комнате 2 человека, но мы определили как 3 человека.
* FN (False Negative) – ложно-отрицательное решение. В комнате 3 человека, но мы не стали определять как 3 человека.

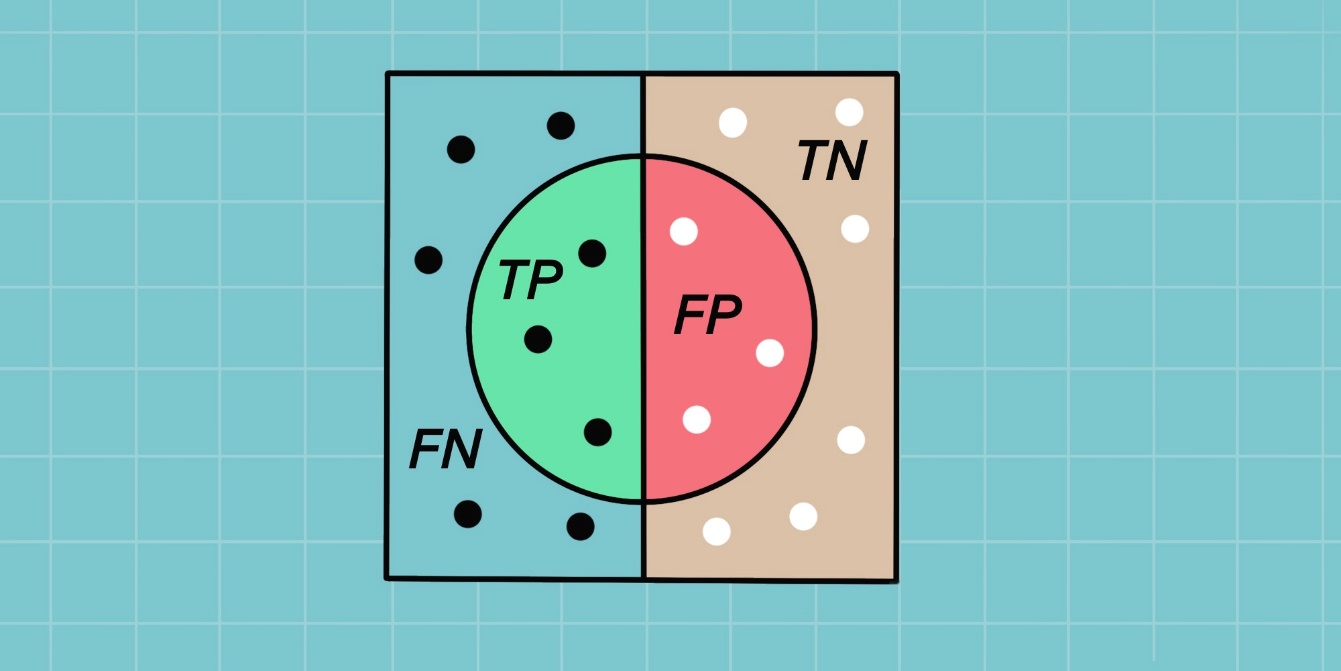


Рис. 6 – исходы работы алгоритма

Далее введем понятия Точность и Полнота:

Precision (точность) – насколько хорошо мы находим цель, стремясь к минимизации всех False срабатываний.

Recall (полнота) – как правильно мы обнаружим все True examples.

Рассчитываются они следующим образом:

(1.18)

(1.19)

Очевидно, что чем выше Precision (точность) и Recall (полнота), тем лучше. Однако, в реальной жизни невозможно достичь максимума обоих показателей одновременно.

Таким образом, необходима метрика, которая объединяет в себе информацию о точности и полноте нашего алгоритма. Ей является F-score:

(1.20)

Библиотека **scikit-learn** предоставляет оценку точности работы с помощью f-score.

Classification report выводит сводную таблицу с результатами точности, полноты, f-score и количеством элементов в классе для каждого из исследуемых классов (рис. 7).

Ниже приводится среднее значение f-score (macro avg) из всех f-score для каждого из классов. Этот метод измеряет все классы одинаково, независимо от количества элементов в каждом классе.

Weighted avg принимает во внимание количество элементов в каждом классе, присваивая каждому из них веса, пропорциональные количеству элементов относительно к общему количеству элементов.

Micro avg, или точность (accuracy), высчитывает глобальный f-score, путем вычисления сумм всех истинно-положительных, ложно-негативных и ложно-положительных значений и подстановки их в уравнение f-score.

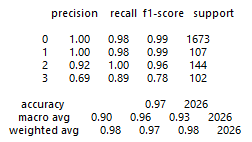


Рис. 7 – classification report из библиотеки scikit-learn

# **Глава 2 – Программная реализация**

## **Краткое описание используемых библиотек**

Программа писалась на языке Python в среде разработки Visual studio code.

Было задействовано множество библиотек, ключевой является **scikit-learn** [15] – она предоставляет алгоритмы машинного обучения и способы оценивания точности их работы.

Библиотека **pandas** предоставила способы чтения и предобработки исходных данных из csv-файла.

Библиотека **mathplotlib** использовалась для построения различных графиков и матриц ошибок.

Библиотека **tkinter** была задействована для оформления программы в виде приложения с несколькими окнами и главным меню.

## **Описание программы**

### **Стартовая вкладка**

Написанная программа представляет собой приложение, состоящее из нескольких окон. Перед написанием были импортированы все необходимые библиотеки и модули используемых библиотек, преимущественно методы машинного обучения (МО) из библиотеки scikit-learn (рис. 8). Также загружаются данные из csv-файла с помощью строки   
data = pd.read\_csv('occup.csv'), где read\_csv – функция модуля pd библиотеки pandas.

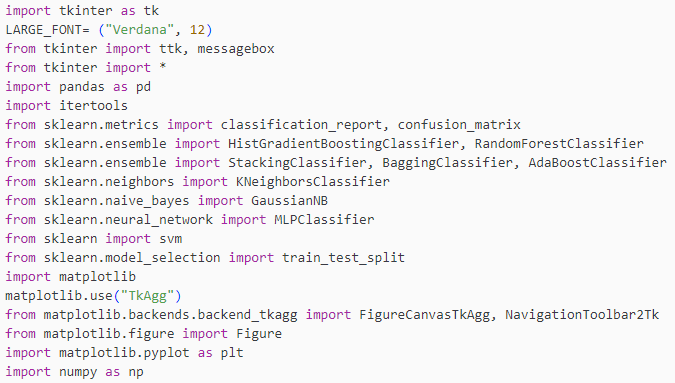


Рис. 8 – импорт в программу всех необходимых библиотек и компонентов

Реализация программы в виде вкладочного приложения стала возможна благодаря использованию библиотеки tkinter. Сначала был реализован класс Application, который стал шаблоном для всех созданных окон (рис. 9).



Рис. 9 – класс Application

Затем была создана стартовая страница, которая содержит 4 вкладки, переключающие на другие функции приложения (рис. 10). В коде описано содержание первой вкладки. (рис. 11).

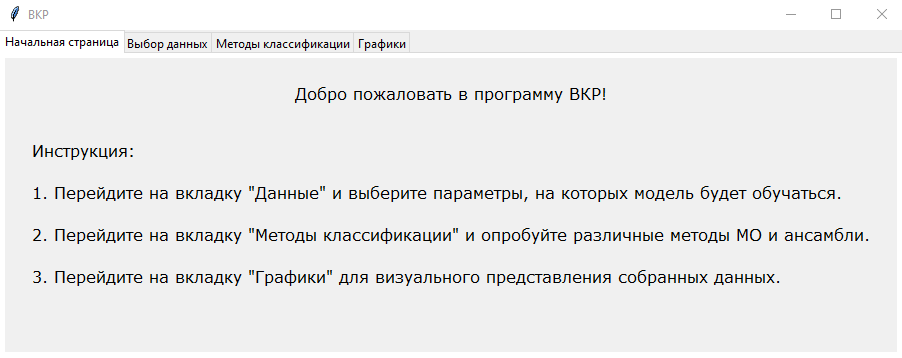


Рис. 10 – Стартовое окно приложения

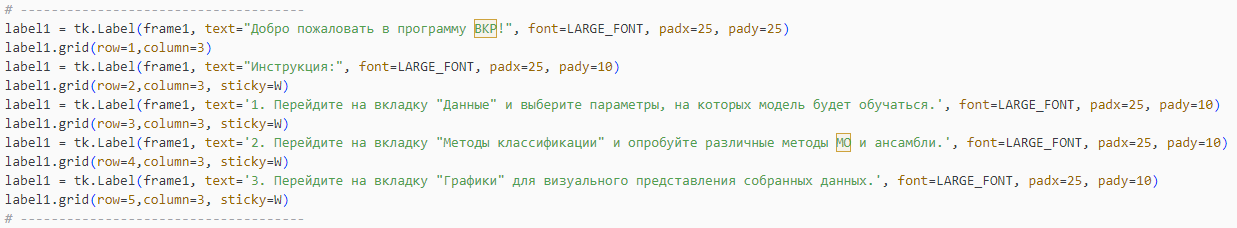


Рис. 11– Программная реализация стартовой страницы

### **Вкладка выбор данных**

Перед тем, как обучать модели машинного обучения, следует выбрать данные, которые станут независимыми параметрами (рис. 12). Данные разделены по типу, и каждый тип содержит несколько столбцов значений, соответствуя количеству сенсоров, с которых они были получены. После выбора следует нажать кнопку «Подтвердить». У каждого чекбокса есть своя переменная, считывающая его состояние, а кнопка «Подтвердить» собирает и передает эти состояния в функцию формирования данных (рис. 13). Программа поддерживает любую комбинацию данных, а при отсутствии выбора каких-либо данных будет выведено сообщение об ошибке с просьбой выбрать хоть бы один тип данных (рис. 14). Выбор данных реализован через функцию choose\_data, в которую передаются 1 и 0 с соответствующих чекбоксов в приложении, далее выбирается необходимая комбинация данных. Через свойство «iloc» в библиотеке pandas выбираются необходимые столбцы, за исключением первых двух колонок данных, отвечающих за дату и время наблюдения, а также последней колонки, являющуюся зависимой величиной. С помощью функции train\_test\_split происходит разделение всех данных на обучающую и тестовую выборки, которые затем возвращаются из функции (рис. 15).

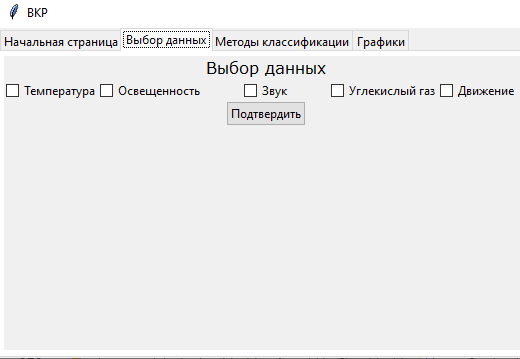


Рис. 12 – вкладка выбора данных перед обучением моделей



Рис. 13 – Программная реализация первой страницы

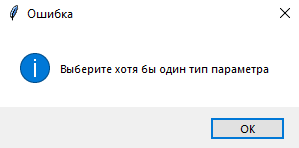


Рис. 14 – сообщение об ошибке



Рис. 15 – функция choose\_data

### **Вкладка методы классификации.**

Окно (рис. 17) содержит все методы машинного обучения, которые используются в данной программе и позволяет обучать модели с последующим выводом результатов обучения. Перед обучением были определены параметры для используемых методов машинного обучения (рис. 16).

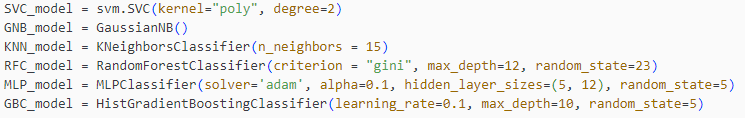


Рис. 16 – задание параметров методам МО

Так же задействованы ансамбли методов машинного обучения, с выбором необходимых параметров: для работы стэкинга выбираются 3 модели, каждая из них должна быть уникальной и не может повторяться. Бэггинг и адаптивный бустинг используют выбранную модель машинного обучения и количество простых моделей, на котором будет основываться работа алгоритма. Результаты выводятся в стандартном формате classification report (рис. 18), который является встроенной функцией библиотеки scikit-learn, и представляют собой показатели точности, полноты, f-меры и количества экземпляров в каждом классе, а также макро-среднее и взвешенное среднее. После того, как окно с результатами закрывается, пользователь видит построенную с помощью библиотеки matplotlib матрицу ошибок (рис. 19).

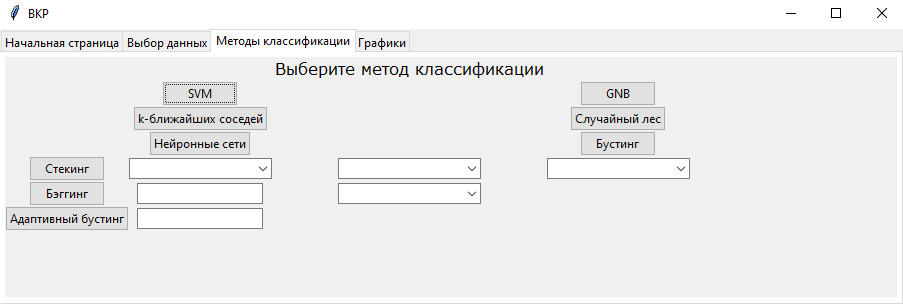


Рис. 17– Третья вкладка приложения с выбором методов МО

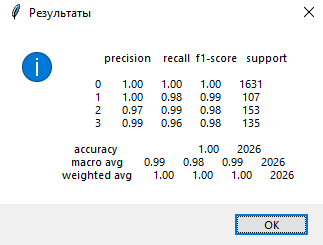


Рис. 18 – Информационное окно с результатами работы метода МО

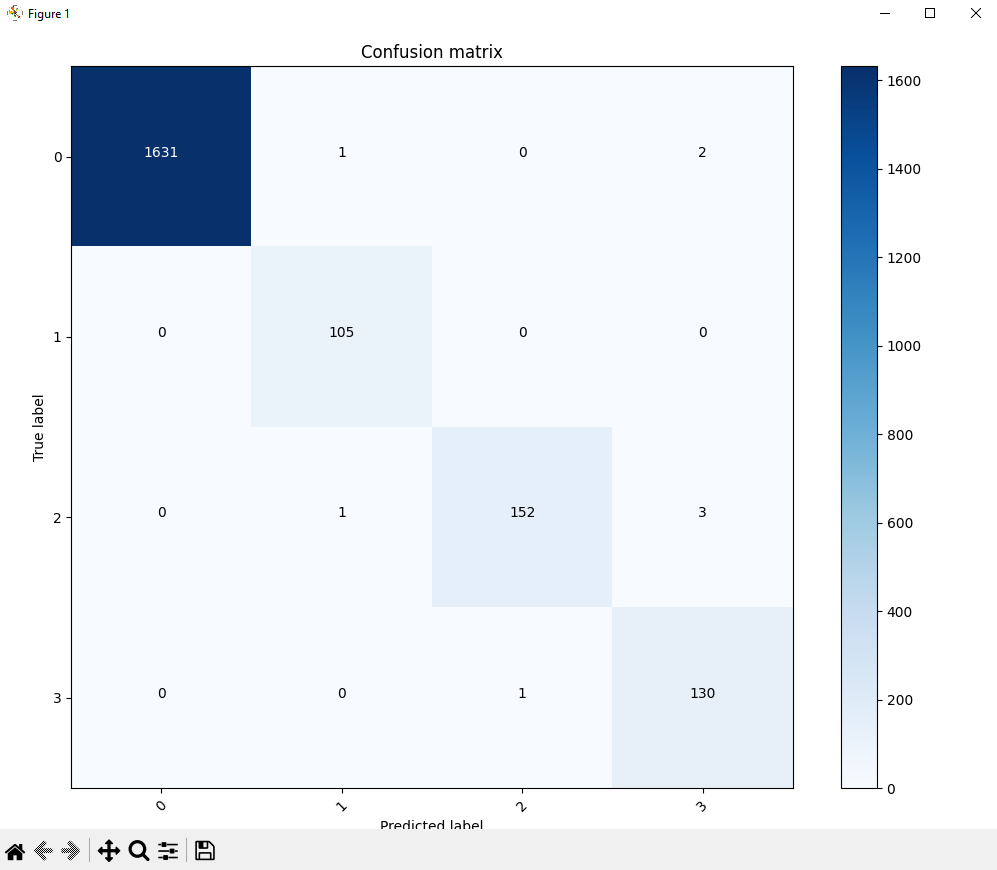


Рис. 19– Матрица ошибок для методов МО

### **Вкладка с графиками**

Третья вкладка (рис. 21) дает возможность пользователю выбрать и построить визуальное представление данных с сенсоров, полученных на протяжении исследования. Они включают все параметры, и распределены по типу параметра: температура (рис. 22), освещенность, звук и т.д. Инструменты ниже графика позволяют манипулировать графиком, например, приблизить (рис. 23). После построения графика, следует нажать кнопку «очистить график» для построения следующего графика.

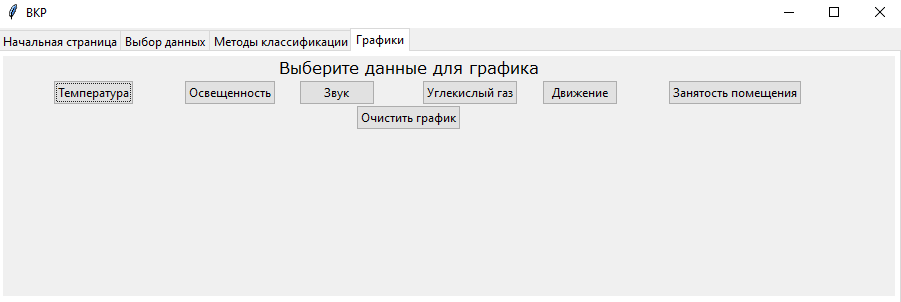


Рис. 21 – четвертая вкладка приложения с графиками

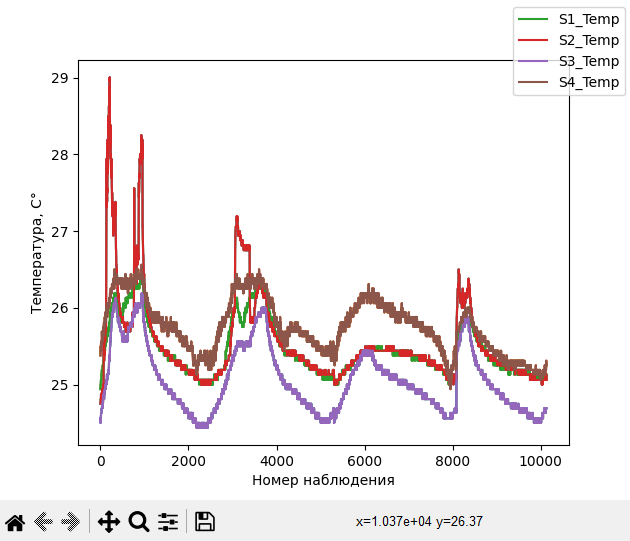


Рис. 22 – Пример графика: температура

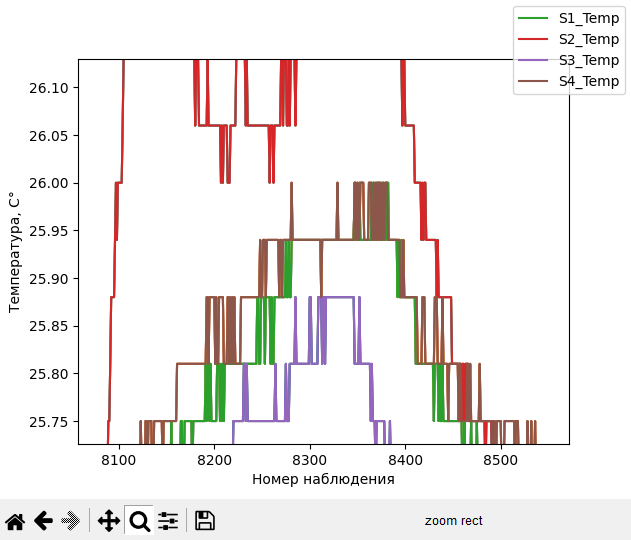


Рис. 23 – Приближенный график температуры

# **Глава 3 – Данные и результаты**

## **Данные и результаты**

### **Постановка задачи**

Главной задачей данного исследования является выявление наиболее точных методов машинного обучения для решения задачи многомерной классификации, в частности прогнозирования класса (занятости) исследуемого объекта (помещения) исходя из показаний сети беспроводных сенсорных узлов, расположенных в исследуемом помещении.

### **Описание данных**

Данные были взяты с сайта Machine Learning Repository, из исследования Adarsh Pal Singh из Интернационального Института Информационных Технологий в Хайдарабаде в 2018 году [1]. Имеется 1 зависимая переменная – количество человек в помещении и 16 параметров – показатели температуры, света, звука, углекислого газа и движения, полученные с беспроводных сенсоров.

Данные поступали с сенсоров [6] (рис. 24) каждые 30 секунд в течение 4 дней, для датчиков движения проверка на наличие движения проводилась каждые 2,5 секунды, на наличие звука – каждые 3 секунды. Исследуемое помещение: лаборатория площадью 6 х 4.6 метров с 4 столами. Под Сенсором 7 находится окно, под Сенсором 6 – дверь. Во время проведения эксперимента системы отопления, вентиляции и кондиционирования были отключены.

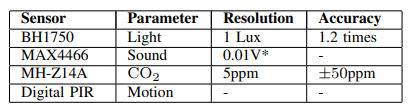


Рис. 24 – Характеристики сенсоров

\*Уровень звука измерялся в вольтах, а не в децибелах.

Использование нескольких различных сенсоров предполагалось для более надежного сбора данных в больших помещениях, чем использование одного сенсора. Расположение сенсоров представлена на схеме:

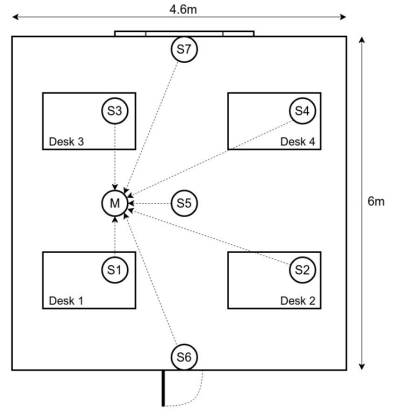


Рис. 25 – расположение сенсоров в исследуемом помещении

Узел М – это мастер-узел, на него поступали и записывались данные со всех сенсоров-подчиненных.

Сенсоры S1-S4 были закреплены на столах, и собирали информацию о температуре, свете. Звук регистрировался каждые 3 секунды. Так как предполагалось использование нескольких сенсоров, было принято решение установить бюджетные сенсоры.

Сенсор S5 регистрировал повышение и понижение уровней углекислого газа в помещении. Этот сенсор был установлен на потолке в центре помещения для наиболее оптимального покрытия помещения.

Сенсоры S6-S7 передавали информацию о движении. Если сенсор регистрировал движение каждые 2.5 секунды, он передавал значение «1» узлу-мастеру. Сенсоры движения были установлены на потолке над дверью и окном под углом, максимизирующим угол обзора сенсоров.

Также, каждый из участников исследования записывал в журнал время, когда он заходил в помещение и покидал его.

По окончанию периода сбора данных был составлен набор данных, состоящий из штампов даты и времени, 4 показателей температуры, освещенности и звука для каждого из четырех сенсоров на столах, показатели углекислого газа и движения. В конце вручную добавлялись данные о занятости помещения. На основе этих данных в дальнейшем обучались методы машинного обучения.

### **Анализ результатов**

Обучение моделей проходило в 2 этапа: на первом этапе методы обучались на каждом параметре по отдельности (Таблица 1). Это позволит увидеть, насколько хорошо модели машинного обучения могут оценивать занятость исходя из одного параметра. В качестве главных метрик использовались точность, полнота и macro F1Score, считающий F1Score для каждого класса отдельно, т.к. набор данных преимущественно соответствует пустой комнате (нулевая занятость), и являющийся более надежным, чем micro F1Score [16].

Таблица 1 – Результаты обучения для каждого параметра индивидуально

|  |  |  |  |
| --- | --- | --- | --- |
| Температура | | | |
|  | Precision | Recall | f-score |
| SVM | 0.53 | 0.50 | 0.51 |
| GNB | 0.51 | 0.43 | 0.46 |
| k-ближ. Соседей | 0.93 | 0.94 | 0.94 |
| Случайный лес | 0.95 | 0.95 | 0.95 |
| Нейронные сети | 0.25 | 0.20 | 0.22 |
| Бустинг | 0.96 | 0.94 | 0.95 |
| Освещенность | | | |
|  | Precision | Recall | f-score |
| SVM | 0.90 | 0.96 | 0.93 |
| GNB | 0.77 | 0.73 | 0.71 |
| k-ближ. Соседей | 0.96 | 0.98 | 0.97 |
| Случайный лес | 0.96 | 0.99 | 0.97 |
| Нейронные сети | 0.93 | 0.97 | 0.94 |
| Бустинг | 0.98 | 0.97 | 0.98 |
| Звук | | | |
|  | Precision | Recall | f-score |
| SVM | 0.49 | 0.62 | 0.53 |
| GNB | 0.51 | 0.61 | 0.54 |
| k-ближ. Соседей | 0.63 | 0.64 | 0.67 |
| Случайный лес | 0.65 | 0.68 | 0.66 |
| Нейронные сети | 0.62 | 0.66 | 0.64 |
| Бустинг | 0.65 | 0.69 | 0.67 |
| СО2 | | | |
|  | Precision | Recall | f-score |
| SVM | 0.37 | 0.37 | 0.36 |
| GNB | 0.54 | 0.51 | 0.52 |
| k-ближ. Соседей | 0.62 | 0.72 | 0.65 |
| Случайный лес | 0.84 | 0.86 | 0.85 |
| Нейронные сети | 0.40 | 0.33 | 0.36 |
| Бустинг | 0.85 | 0.88 | 0.86 |
| Движение | | | |
|  | Precision | Recall | f-score |
| SVM | 0.39 | 0.32 | 0.35 |
| GNB | 0.39 | 0.32 | 0.35 |
| k-ближ. Соседей | 0.37 | 0.32 | 0.34 |
| Случайный лес | 0.39 | 0.32 | 0.35 |
| Нейронные сети | 0.39 | 0.32 | 0.35 |
| Бустинг | 0.39 | 0.32 | 0.35 |

Как можно заметить, из индивидуальных параметров Освещенность дает лучший результат: 5 из 6 методов демонстрируют точность и F1Score больше 0,9. Однако, стоит отметить, что, несмотря на высокую точность показаний, объясняющуюся тем, что участники эксперимента включали лампы на рабочем столе, когда приходили и выключали их, когда уходили, использование только освещенности как независимого параметра оставляет систему уязвимой к ложным срабатываниям. Например, если человек покинул помещение, оставив свет включенным. Поэтому в дальнейшем анализе, когда при обучении моделей параметры добавлялись один за другим, освещенность добавлялась последней.

Диаметрально противоположно результатам Освещенности, Движение как индивидуальный независимый параметр дало худшие результаты, указывая, что его следует использовать только в комбинации с другими показателями.

На данном этапе исследования было выявлено, что методы «Случайный лес» и «Градиентный бустинг деревьев решений» дают наилучшие результаты по сравнению с другими методами машинного обучения.

Во втором этапе исследования обучение моделей началось с комбинации Температуры и СО2, с дальнейшим добавлением Звука, Движения и Освещенности в конце. Результаты можно видеть в Таблице 2.

Таблица 2 – Результаты обучения для комбинаций параметров

|  |  |  |  |
| --- | --- | --- | --- |
| Темп + СО2 | | | |
|  | Precision | Recall | f-score |
| SVM | 0.35 | 0.35 | 0.35 |
| GNB | 0.59 | 0.55 | 0.55 |
| k-ближ. Соседей | 0.68 | 0.76 | 0.71 |
| Случайный лес | **0.98** | **0.98** | **0.98** |
| Нейронные сети | 0.61 | 0.69 | 0.64 |
| Бустинг | **0.97** | **0.97** | **0.97** |
| Темп + СО2 + Звук | | | |
|  | Precision | Recall | f-score |
| SVM | 0.34 | 0.35 | 0.34 |
| GNB | 0.74 | 0.71 | 0.72 |
| k-ближ. Соседей | 0.67 | 0.75 | 0.71 |
| Случайный лес | **0.97** | **0.97** | **0.97** |
| Нейронные сети | 0.63 | 0.63 | 0.51 |
| Бустинг | **0.98** | **0.98** | **0.98** |
| Темп + СО2 + Звук + Движение | | | |
|  | Precision | Recall | f-score |
| SVM | 0.34 | 0.35 | 0.34 |
| GNB | 0.75 | 0.74 | 0.74 |
| k-ближ. Соседей | 0.66 | 0.74 | 0.69 |
| Случайный лес | **0.97** | **0.97** | **0.97** |
| Нейронные сети | 0.60 | 0.71 | 0.59 |
| Бустинг | **0.98** | **0.99** | **0.98** |
| Темп + СО2 + Звук + Движение + Освещенность | | | |
|  | Precision | Recall | f-score |
| SVM | 0.90 | 0.96 | 0.93 |
| GNB | 0.92 | 0.89 | 0.91 |
| k-ближ. Соседей | 0.99 | 0.98 | 0.99 |
| Случайный лес | **0.99** | **0.99** | **0.99** |
| Нейронные сети | 0.94 | 0.97 | 0.96 |
| Бустинг | **0.99** | **0.99** | **0.99** |

Как и ожидалось, полный набор данных, включающий все параметры, дал наилучшие результаты, оценивая занятость помещения точно. Случайный лес и Градиентный бустинг дают результаты F1Score = 0,99. Матрицы ошибок подтверждают точность обученных моделей (рис. 26 и рис. 27).

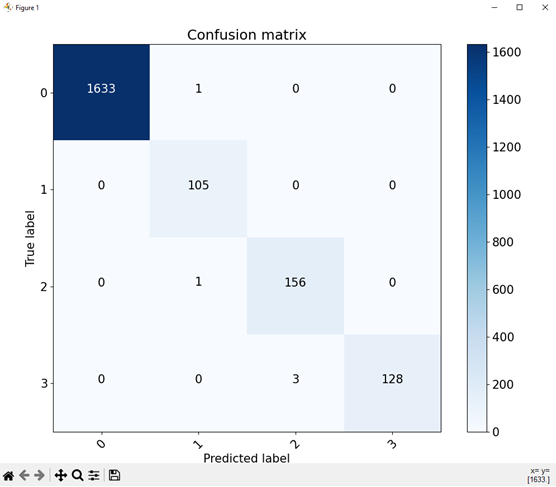


Рис. 26 – матрица ошибок случайного леса при обучении со всеми параметрами

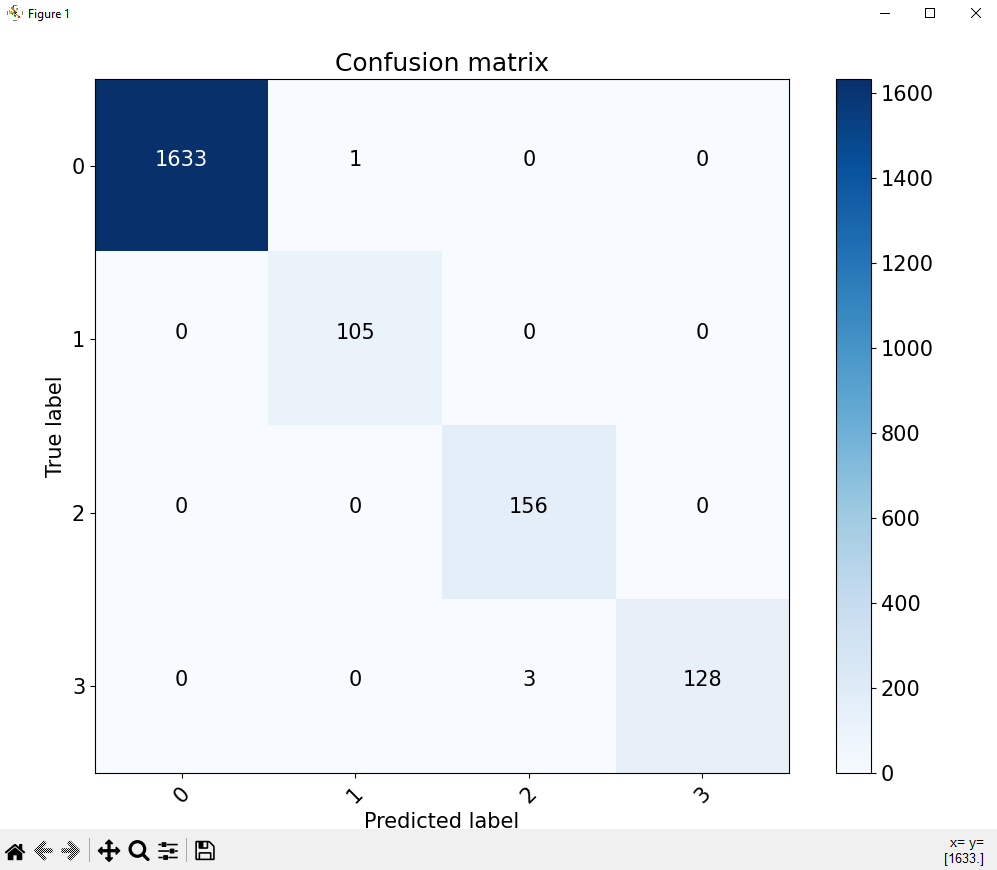


Рис. 27 – матрица ошибок градиентного бустинга при обучении со всеми параметрами

Другие алгоритмы работают менее точно, метод k-ближайших соседей имеет F1Score больший в большинстве случаев среди оставшихся методов.

# **Заключение**

В ходе работы были изучены труды на данную тему и разработана программа, способная принимать на вход файл с исходными данными, обучать следующие модели машинного обучения и оценивать точность их работы: Метод опорных векторов, Наивный Байесовский классификатор, Метод k-ближайших соседей, Случайный лес, Нейронные сети, Градиентный бустинг деревьев решений, Стекинг, Бэггинг, Адаптивный бустинг.

Получены результаты решения задачи прогнозирования занятости помещения с применением различных алгоритмов машинного обучения и проведено сравнение методов для различных наборов показателей сенсорных узлов. Как и ожидалось, полный набор данных, включающий все параметры, дал наилучшие результаты. Случайный лес и Градиентный бустинг дают результаты F1Score = 0,99.

Методы машинного обучения способны с большой точностью определять занятость помещения, основываясь только на показателях беспроводных сенсорных узлов, что открывает возможности для совершенствования и модернизаций систем отопления, вентиляции и кондиционирования. Использование машинного обучения для управления такими системами означает потенциальную экономию электроэнергии, а также повышение комфорта жизни людей.

Методы машинного обучения могут применяться и в других областях, близких к исследуемой в данной работе, открывая путь к оптимизации других систем.

Таким образом, алгоритмы машинного обучения могут помочь человечеству контролировать и оптимизировать расходы энергии в домах, делая потребление энергии зависимым от занятости помещения.

# **Список литературы**

1. Adarsh Pal Singh, Vivek Jain, Sachin Chaudhari, Frank Alexander Kraemer, Stefan Werner and Vishal Garg, “Machine Learning-Based Occupancy Estimation Using Multivariate Sensor Nodes”, in 2018 IEEE Globecom Workshops (GC Wkshps), pp. 20 – 21, 2018.
2. B. Balaji, J. Xu, A. Nwokafor, R. Gupta, and Y. Agarwal, “Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings,” in Proc. 11th ACM Conf. Embedded Networked
3. Sensor Systems. New York, NY, USA: ACM, 2013, pp. 17:1–17:14.
4. J. Scott et al., “Preheat: Controlling home heating using occupancy prediction,” in Proc. 13th Int. Conf. Ubiquitous Comput., ser. UbiComp. New York, NY, USA: ACM, 2011, pp. 281–290.
5. N. Li, G. Calis, and B. Becerik-Gerber, “Measuring and monitoring occupancy with an RFID based system for demand-driven HVAC operations,” Autom. construction, vol. 24, pp. 89–99, 2012.
6. I. Ang, F. Salim, and M. Hamilton, “Human occupancy recognition with multivariate ambient sensors,” in IEEE Int. Conf. Pervasive Comput. and Commun. Workshops (PerCom Workshops), Mar. 2016, pp. 1–6.
7. L. Candanedo and V. Feldheim, “Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models,” Energy and Buildings, vol. 112, pp. 28 – 39, 2016.
8. E. Hailemariam, R. Goldstein, R. Attar, and A. Khan, “Real-time occupancy detection using decision trees with multiple sensor types,” in Proc. Symp. Simulation Architecture and Urban Des. (SimAUD), San Diego, CA, USA, 2011, pp. 141–148.
9. K. Lam et al., “Occupancy detection through an extensive environmental sensor network in an open-plan office building,” IBPSA Building Simulation, vol. 145, pp. 1452–1459, 2009.
10. B. Dong et al., “An information technology enabled sustainability test-bed (itest) for occupancy detection through an environmental sensing network,” Energy and Buildings, vol. 42, no. 7, pp. 1038 – 1046, 2010.
11. Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, “A multisensor based occupancy estimation model for supporting demand driven HVAC operations,” in Proc. Symp Simulation Architecture and Urban Des. (SimAUD), San Diego, CA, USA, 2012, pp. 2:1– 2:8.
12. M. Masood, Y. Soh, and V. Chang, “Real-time occupancy estimation using environmental parameters,” in Int. Joint Conf. on Neural Netw. (IJCNN), Jul. 2015, pp. 1–8.
13. Z. Chen et al., “Building occupancy estimation with environmental sensors via CDBLSTM,” IEEE Trans. Ind. Electron., vol. 64, no. 12, pp. 9549–9559, Dec 2017
14. A. Dey et al., “Namatad: Inferring occupancy from building sensors using machine learning,” in IEEE 3rd World Forum Internet of Things (WF-IoT), Dec. 2016, pp. 478–483.
15. F. Pedregosa et al., “Scikit-learn: Machine learning in python,” J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Nov. 2011.
16. P. Perner, Machine Learning and Data Mining in Pattern Recognition: 10th International Conference, MLDM 2014, St. Petersburg, Russia, July 21-24, 2014, Proceedings. Springer International Publishing, 2014.
17. V. Garg and N. Bansal, “Smart occupancy sensors to reduce energy consumption,” Energy and Buildings, vol. 32, no. 1, pp. 81 – 87, 2000.
18. Айвазян С. А., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: исследование зависимостей. — М.: Финансы и статистика, 1985.
19. Айвазян С. А., Бухштабер В. М., Енюков И. С., Мешалкин Л. Д. Прикладная статистика: классификация и снижение размерности. — М.: Финансы и статистика, 1989.
20. Вапник В. Н. Восстановление зависимостей по эмпирическим данным. — М.: Наука, 1979.
21. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.
22. Загоруйко Н. Г. Прикладные методы анализа данных и знаний. — Новосибирск: ИМ СО РАН, 1999. ISBN 5-86134-060-9.
23. Шлезингер М., Главач В. Десять лекций по статистическому и структурному распознаванию. — Киев: Наукова думка, 2004. ISBN 966-00-0341-2.
24. Hastie, T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. — Springer-Verlag, 2009. — 746 p. — ISBN 978-0-387-84857-0..
25. Mitchell T. Machine Learning. — McGraw-Hill Science/Engineering/Math, 1997. ISBN 0-07-042807-7.
26. Ryszard S. Michalski, Jaime G. Carbonell, Tom M. Mitchell (1983), Machine Learning: An Artificial Intelligence Approach, Tioga Publishing Company, ISBN 0-935382-05-4[1].
27. Vapnik V.N. Statistical learning theory. — N.Y.: John Wiley & Sons, Inc., 1998. [1]
28. Bernhard Schölkopf, Alexander J. Smola Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. — MIT Press, Cambridge, MA, 2002 ISBN 978-0-262-19475-4 [2]
29. I.H. Witten, E. Frank Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). — Morgan Kaufmann, 2005 ISBN 0-12-088407-0 [3]
30. Liang Wang, Li Cheng, Guoying Zhao Machine Learning for Human Motion Analysis. — IGI Global, 2009. — 318 p. — ISBN 978-1-60566-900-7.