МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/ специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(БАКАЛАВРСКАЯ РАБОТА)

на тему «Применение технологий смешанного машинного обучения при решении задач регрессии»

Студент Н.Н. Стрежнев _____
(И.О. Фамилия) (личная подпись)

Руководитель к.тех.н., В.С. Климов
(ученая степень, звание, И.О. Фамилия)

Консультант М.В. Дайнеко
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема бакалаврской работы: «Применение технологий смешанного машинного обучения при решении задач регрессии».

В работе исследуются пути развития ансамблевых технологий машинного обучения.

Объект исследования – ансамблевые алгоритмы машинного обучения.

Предмет исследования – обеспечение совместной работы произвольно выбранного набора методов машинного обучения

Цель работы – разработка подхода смешанного машинного обучения и программного обучения для его тестирования.

Задачи исследования:

1. Анализ существующих подходов по совместному использованию методов машинного обучения.
2. Разработка подхода смешанного машинного обучения позволяющего совмещать различные алгоритмы машинного обучения в заданной пользователем комбинации.
3. Разработка программной обеспечения, позволяющего тестировать смешанное обучение на произвольно заданных данных.

В первой главе работы анализируются существующие ансамблевые технологии машинного обучения. Во второй главе описывается разработанный подход смешанного машинного . В третьей главе описывается программная реализация предложенного подхода и результаты его тестирования. В заключении представлены выводы по проделанной работе.

В работе присутствует 22 рисунка, 1 таблица. Список литературы состоит из 25 литературных источников. Общий объем выпускной квалификационной работы составляет 44 страницы.

Abstract

The topic of the bachelor's thesis is «Application of Mixed Machine Learning Technologies in Regression Problems Solving».

The paper investigates the development of ensemble machine learning technologies.

The object of the research is ensemble algorithms of machine learning.

The subject of research is to ensure that a randomly selected set of machine learning methods work together.

The purpose of the work is to develop a mixed machine learning approach and software learning to test it.

Research objectives:

1. To analyze existing approaches for collaborative machine learning methods.

2. Development of a mixed machine learning approach that allows combining different machine learning algorithms in a user-defined combination.

3. Development of software that allows to test mixed learning on arbitrary data.

The first chapter of the paper analyzes existing ensemble machine learning technologies. The second chapter describes the developed mixed machine learning approach. The third chapter describes the software implementation of the proposed approach and its testing results. The conclusion of the work done is presented in the conclusion.

The paper contains 22 figures, 1 table. The reference list consists of 25 literature sources. The total volume of the graduate qualification work is 44 pages.

Оглавление

Введение 5
Глава 1 Обзор ансамблевых технологий машинного обучения при решении задач регрессии 7
1.1 Разновидности задач регрессии при анализе данных 7
1.2 Обзор ансамблевых технологий машинного обучения 8
Глава 2 Разработка технологии смешанного использования алгоритмов машинного обучения 17
2.1 Схема применения смешанного машинного обучения 17
2.2 Данные для апробации смешанного машинного обучения 21
Глава 3 Программная реализация смешанного машинного обучения 22
3.1 Описание программного кода 22
3.2 Интерфейс программного обеспечения 32
3.3 Результаты тестирования 35
Заключение 37
Список используемой литературы 39

Введение

Основой искусственного интеллекта с точки зрения математики и информатики являются алгоритмы машинного обучения. С помощью этих алгоритмов, с использованием обучающего набора данных, производится обучение прогнозных моделей, описывающих закономерности в данных.

Машинное обучение используется во многих научных отраслях: при создании диагностических производственных систем, при проектировании аналитических систем оценки рисков в экономике, при создании систем поддержки принятия решений в медицине, в системах обеспечения общественного порядка используемых правоохранительными органами и т.д.

Актуальной задачей для исследователей алгоритмов машинного обучения является повышение точности получаемых прогнозных моделей. Помимо разработки новых алгоритмов, исследователи из данной области разрабатывают новые способы применения существующих алгоритмов. Например, одно из современных достижений в данной области – методы ансамблевого использования существующих алгоритмов машинного обучения. Эти методы основаны на идее использования одного метода машинного обучения для настройки с его помощью нескольких прогнозных моделей, работающих (с целью повышения точности получаемых прогнозов) совместно. При этом результат прогнозирования формируется на основе совокупности выводов совместно работающих прогнозных моделей. Совместно работающие модели в этом случае называют ансамблями (ensemble methods).

К таким ансамблевым методам относятся Бэггинг (Bagging), Случайный лес (Random Forest), Бустинг (Boosting), Градиентный бустинг (Gradient Boosting), параллельная реализация градиентного бустинга (XGBoost).

Все ансамблевые методы объединяет то, что они работают на основе одного выбранного базового метода машинного обучения.

В данной бакалаврской работе делается предположение, что возможно развить ансамблевый подход до использования не одного базового метода, а сколь угодно большого числа базовых методов «смешивая» их в любых комбинациях на выбор пользователя. Очевидно, что чтобы доказать эффективность смешанного машинного обучения требуется провести большое количество вычислительных экспериментов, а для этого требуется программное обеспечение, реализующее данную технологию.

Поэтому, цель работы – разработка подхода смешанного машинного обучения и программного обучения для его тестирования.

Для достижения поставленной цели в работе решаются следующие задачи:

1. Анализ существующих подходов по совместному использованию методов машинного обучения.
2. Разработка подхода смешанного машинного обучения позволяющего совмещать различные алгоритмы машинного обучения в заданной пользователем комбинации.
3. Разработка программной обеспечения, позволяющего тестировать смешанное обучение на произвольно заданных данных.

Глава 1 Обзор ансамблевых технологий машинного обучения при решении задач регрессии

1.1 Разновидности задач регрессии при анализе данных

Задача регрессии – один из типов математической задачи, в которой требуется найти и объяснить взаимосвязь между характеристиками рассматриваемых объектов и целевым значением, выраженным в виде вещественного числа. С помощью регрессионных моделей можно получать различные прогнозные модели [7, 17].

Регрессионный анализ применяется в различных областях науки. На производстве регрессионные модели используются в составе диагностических систем, когда на основе параметров технологического процесса прогнозируются количественные характеристики получаемого результата, например, прочность изделия или химический состав. В медицине регрессионные модели используются, например, для определения вероятности успешного исхода назначаемого лечения в зависимости от медицинских параметров пациента. Также регрессионные модели используются для определения вероятности наличия заболевания у пациентов. В финансовой области регрессионные модели используются для оценки рисков при реализации инвестиционных проектов [1, 5, 10].

Как видно из проведенного обзора, регрессионный анализ повсеместно применяется для решения практических задач различного типа. Поэтому актуальными можно признать исследования направленные на развитие методов для построения регрессионных моделей данных.

С учетом того, что при построении таких моделей чаще всего используются алгоритмы машинного обучения, именно наибольшей значимостью обладают исследования на развитие способов применения данных алгоритмов. В настоящее время получили распространение

ансамблевые методы построения моделей, так как получаемые с их помощью прогнозы отличаются более высокой точностью.

1.2 Обзор ансамблевых технологий машинного обучения

Рассмотрим существующие подходы для ансамблевого применения алгоритмов машинного обучения.

Ансамбль алгоритмов – это подход по пользованию одного метода машинного обучения для настройки с его помощью нескольких прогнозных моделей, работающих совместно с целью повышения точности получаемых прогнозов. Результат прогнозирования формируется на основе совокупности выводов совместно работающих прогнозных моделей. При построении ансамбля применяет один тип алгоритмов машинного обучения [2, 6].

Наиболее изученной областью применения ансамблей является построение регрессионных моделей на основе деревьев решений (Decision Tree’s algorithm).

Дерево решений (decision tree) – структура данных, в процессе обхода которой в зависимости от условия в каждом узле принимается решение по переходу по той или иной ветке от корня к конечным вершинам [8].

Алгоритм построения дерева включает в себя следующие элементы:

1. Дерево строится сверху вниз от корня.
2. Для каждого значения атрибута создается ветка дерева, набор наблюдений разделяется в соответствии со значением в каждой ветке.
3. Процесс повторяется рекуррентно для каждой ветки.

Пример дерева принятия решений показан на рисунке 1.

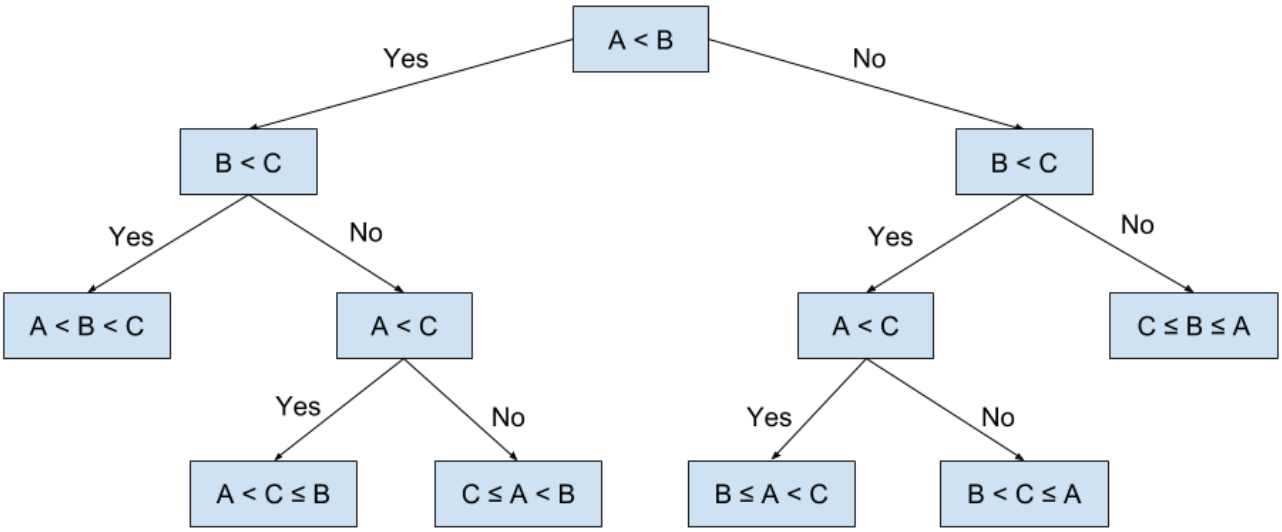

Рисунок 1 – Пример дерева принятия решений

При построении дерева используется критерий разделения множества данных, который которые можно разделить на 2 типа [12]:

– критерии, основанные на теории информации (цель - снизить неопределенность);

– критерии, основанные на расстоянии (цель – получить в листах дерева максимально отличающиеся друг от друга значения целевого признака).

Достоинствами алгоритмов для построения деревьев решений является:

– простота алгоритмов;

– легкость интерпретирования работы алгоритмов;

– возможность работы с различными данными (например, числовые, категориальные);

– возможность обрабатывать данные с пропущенными значениями;

– высокая скорость работы (построение регрессионной модели и прогнозирования).

Недостатками алгоритмов для построения деревьев решений является [14]:

– переобучение при сложных конструкциях представления данных;

– разделяющая граница, построенная деревом решений, имеет свои ограничения и на практике уступает некоторым другим методам;

– проблема построения оптимального дерева - NP-полная.

Теперь рассмотрим алгоритм машинного обучения для построения ансамблевой модели – случайного леса (random forest).

Случайный лес (random forest) — это композиция независимых деревьев решений [16].

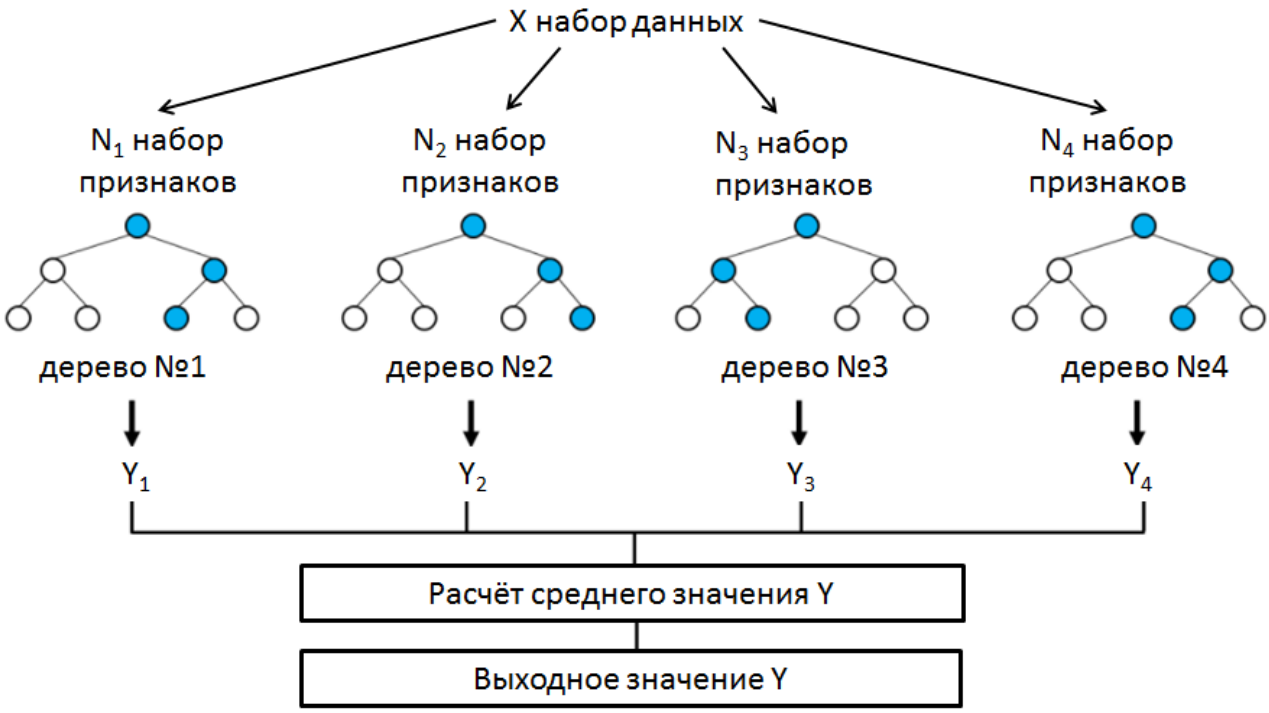

Рисунок 2 – Схема работы случайного леса

Алгоритм построения случайного леса сочетает в себе две основные идеи: метод бэггинга (Bagging) предложенный Лео Брайманом (Leo Breiman), и метод случайных подпространств, предложенный Тим Кам Хо (Tin Kam Ho) [3].

Основная идея заключается в использовании большого ансамбля решающих деревьев, каждое из которых само по себе даёт высокую точность

прогнозов, но за счёт их большого количества результат получается хорошим.

Схема работы случайного леса показана на рисунке 2.

Преимущества при использовании случайного леса [4]:

– эффективно обрабатывает данные с большим числом признаков;

– нечувствительность к масштабированию значений признаков;

– распараллеливание вычислений (так как тренировка деревьев происходит независимо);

– одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки.

Недостатки использования случайного леса:

– в отличие от одного дерева, результаты случайного леса сложнее интерпретировать;

– большой размер получающихся моделей. Требуется O(K) памяти для хранения данных, где K — число деревьев.

Рассмотрим еще один алгоритм для построения ансамблевой модели - градиентный бустинг (Gradient Boosting). Алгоритм предполагает решение задачи регрессии методом построения комитета (ансамбля) «слабых» предсказывающих деревьев принятия решений [9, 25].

На первой итерации строится ограниченное по количеству узлов дерево принятия решений, после чего считается разность между тем, что предсказало полученное дерево, умноженное на learn rate (коэффициент «слабости» каждого дерева), и искомой переменной на этом шаге.

По этой разнице строится следующая итерация. Так продолжается, пока результат не перестанет улучшаться. Таким образом, на каждом шаге мы пытаемся исправить ошибки предыдущего дерева.

Схема формирования ансамбля регрессионных моделей при градиентном бустинге показана на рисунке 3.

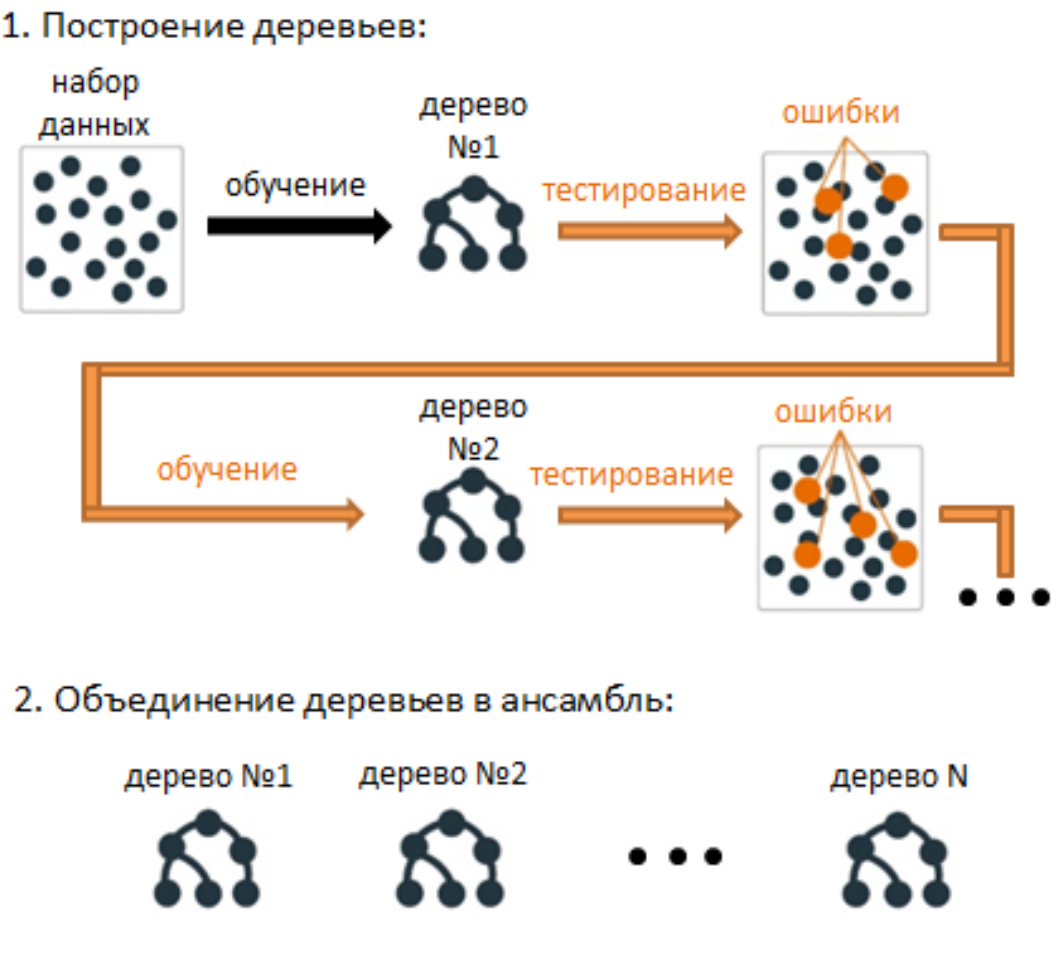

Рисунок 3 – Схема работы градиентного бустинга

Лучше использовать проверочные данные, которые не участвовали в построении, так как на обучающих данных возможно переобучение.

Преимущества градиентного бустинга [23]:

– высокое качество результата, особенно для данных с большим количеством наблюдений и малым количеством переменных;

– ссравнительно с алгоритмом построения случайного леса – быстрое время построение оптимальной модели.

Недостатки градиентного бустинга [24]:

– требуется тестовая выборка (либо кросс-валидация).

– невозможность хорошо распараллелить (т. к. последующее построение дерева зависит от предыдущего);

– относительно слабая устойчивость к ошибочным данным и переобучению;

– сложная интерпретация модели (так же как и в random forest).

Дальнейшим развитием алгоритма градиентного бустинга стало появление его модификации под названием XGBoost.

Градиентный бустинг — это техника машинного обучения для задач классификации и регрессии, которая строит модель предсказания в форме ансамбля слабых предсказывающих моделей, обычно деревьев решений [11, 13].

Обучение ансамбля проводится последовательно в отличие, например от бэггинга (Bagging). На каждой итерации вычисляются отклонения предсказаний уже обученного ансамбля на обучающей выборке. Следующая модель, которая будет добавлена в ансамбль, предназначена для предсказывания эти отклонений.

Таким образом, добавив предсказания нового дерева к предсказаниям обученного ансамбля, мы можем уменьшить среднее отклонение модели, которое является таргетом оптимизационной задачи. Новые деревья добавляются в ансамбль до тех пор, пока ошибка уменьшается, либо пока не выполняется одно из правил "ранней остановки".

Преимущества алгоритма XGBoost [18, 22]:

– эффективно применение в задачах предсказания, которые используют неструктурированные данные (например, изображения, текст);

– алгоритм использует свой собственный метод кросс-валидации на каждой итерации.

Недостатки алгоритма XGBoost [19]:

– при работе со структурированными или табличными данными небольших размеров эффективнее оказываются алгоритмы, основанные на построении одного дерева принятия решений;

– сложная интерпретация модели.

Сравнение эффективности одиночной модели Decision Trees и ансамблевых моделей (Boosting, XGBoost, Bagging, Random Forest) по скорости обучения и точности прогнозирования показано на рисунке 4.

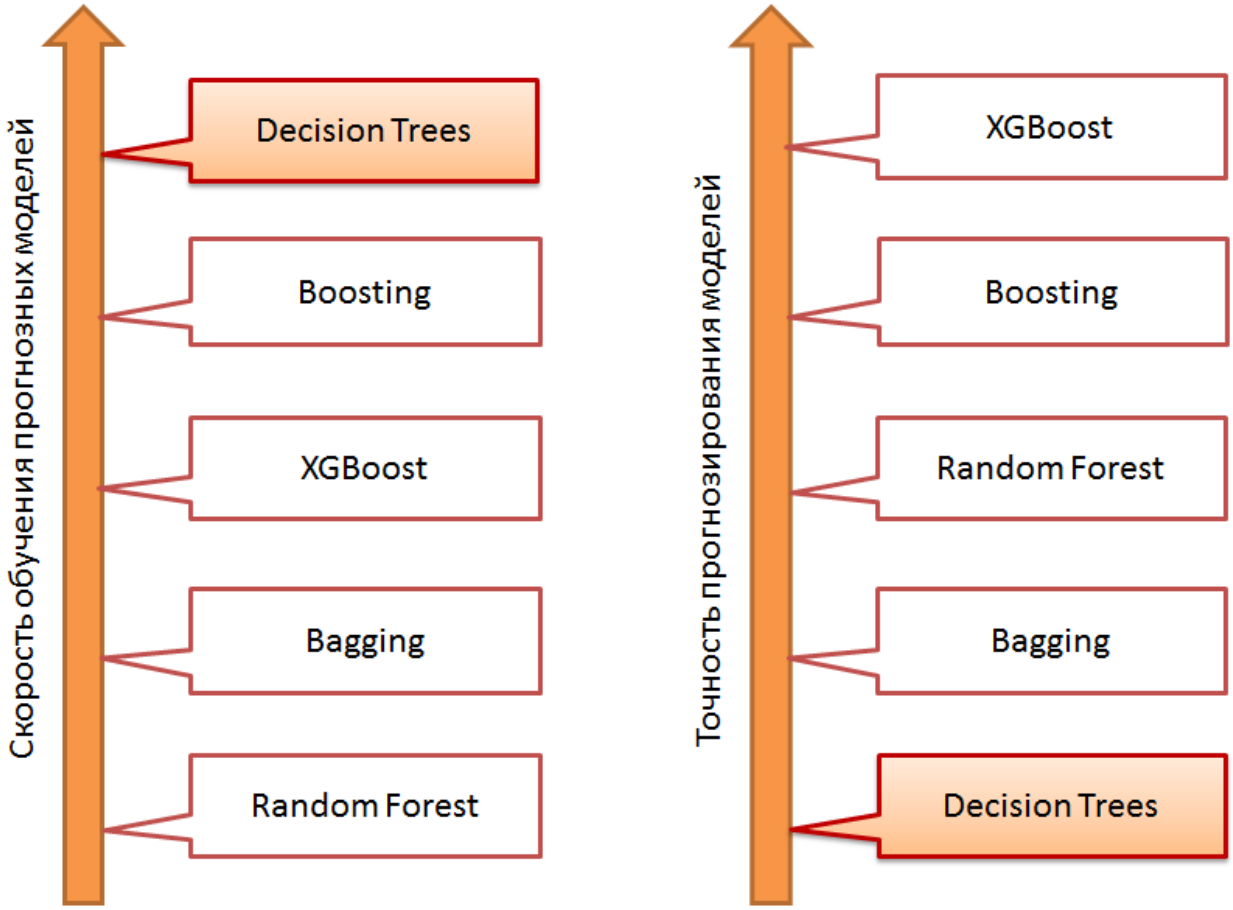

Рисунок 4 – Сравнение эффективности одиночной модели Decision Trees и ансамблевых моделей (Boosting, XGBoost, Bagging, Random Forest) по скорости обучения и точности прогнозирования

Ключевые особенности моделей описанных моделей сведены в таблицу 1.

Таблица 1 – Ключевые особенности одиночной модели Decision Trees и ансамблевых моделей Boosting, XGBoost, Bagging, Random Forest [15, 20, 21]

Название алгоритма	**Ключевые особенности**
Алгоритмы построения деревьев решений (Decision Tree’s algorithm)	Базовый алгоритм, который может применяться в для построения составных прогнозных моделей. Обеспечивает представление регрессионной модели в виде дерева.
Бэггинг (Bagging)	Ансамблевый мета-алгоритм, основанный на учете предсказания от нескольких деревьев решений и использующий механизм голосования.
Случайный лес (Random Forest)	Ансамблевый алгоритм, основанный на пакетировании данных. Исходные данные случайным образом делятся на подмножества. Затем для каждого подмножества данных строится свое регрессионное дерево.
Бустинг (Boosting)	Ансамблевый алгоритм, подразумевающий последовательно построение регрессионных деревьев. Каждое последующее дерево направлено на компенсацию ошибок предыдущего дерева
Градиентный бустинг (Gradient Boosting)	Ансамблевый алгоритм, отличающийся тем, что для минимизации ошибок в последовательных моделях используется алгоритм градиентного спуска.
Параллельный градиентный бустинг (XGBoost)	Оптимизированная версия ансамблевого алгоритма Gradient Boosting. Оптимизация достигается за счет параллельной обработки, обрезки деревьев, обработки пропущенных значений и регуляризации.

Таким образом, в настоящее время существуют подходы по использованию одного метода машинного обучения для настройки с его

помощью нескольких прогнозных моделей, работающих совместно с целью повышения точности получаемых прогнозов. Результат прогнозирования формируется на основе совокупности выводов совместно работающих прогнозных моделей. Все ансамблевые методы объединяет то, что они работают на основе одного выбранного базового метода машинного обучения.

Для совершенствования идеи ансамблевых методов предложено разработать подход смешанного машинного обучения, исключающего ограничение на использование единственного базового алгоритма. При этом также будут обучаться составные прогнозные модели, но разнообразие состава будет обеспечиваться за счет применения разных алгоритмов машинного обучения (а не одного, как в ансамблевых методах).

Выводы по главе

Было произведено сравнение подходов по использования алгоритмов машинного обучения для получения составных прогнозных моделей (ансамблей). При сравнении рассматривались такие ансамблевые методы, как бэггинг (Bagging), случайный лес (Random Forest), бустинг (Boosting), градиентный бустинг (Gradient Boosting), параллельный градиентный бустинг (XGBoost). По результатам сравнения была составлена схема, показывающая, как соотносятся скорости обучения ансамблевых методов и точности прогнозирования получаемых моделей.

В ходе анализа ансамблевых методов установлено, что их использование основано на применении одно базового алгоритма, например, например алгоритма построения дерева принятия решения. Поэтому предложено разработать технология смешанного машинного обучения позволяющего использовать не один, а сразу несколько базовых алгоритмов при построении составных прогнозных моделей.

Глава 2 Разработка технологии смешанного использования алгоритмов машинного обучения

2.1 Схема применения смешанного машинного обучения

Предложенная технология смешанного машинного обучения позволяет согласовать работу отдельно построенных регрессионных моделей для увеличения точности формируемых прогнозов.

Алгоритм получения составной регрессионной модели с использованием предложенной технологии состоит из следующих этапов:

1. Подготовка данных для обучения моделей.

2. Использование любого количества методов машинного обучения для построения множества регрессионных моделей с использованием тренировочной выборки данных.

3. Обучение полученных регрессионных моделей совместной работе для формирования итогового прогноза.

На первом этапе проводится разделения исходных данных тренировочную и тестовую выборку.

На втором этапе проводится параллельное обучение регрессионных моделей на одних и тех же тренировочных данных с использованием различных методов машинного обучения. В результате выполнения данного этапа мы получаем множество регрессионных моделей работающих с различной точностью (рисунок 5).

Так как методы машинного обучения основаны на различных математических принципах, то результаты предсказаний регрессионных моделей, полученных с их использованием, также будут отличаться.

Так как заранее неизвестно, какие методы машинного обучения покажут лучшие результаты на анализируемом наборе данных, то при обучении нескольких регрессионных моделей возникают следующие ситуации:

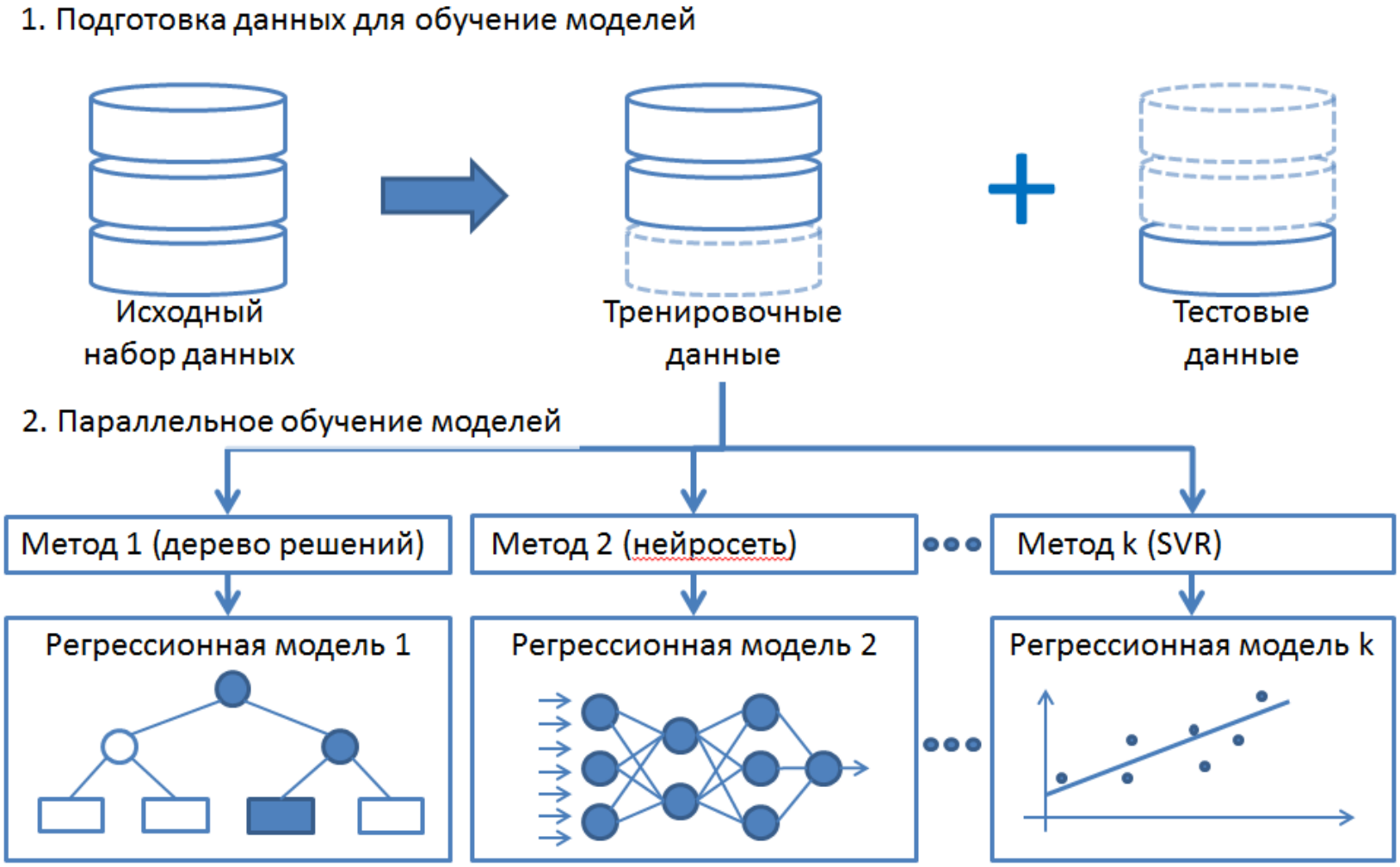

Рисунок 5 – Подготовка исходных данных и построение отдельных регрессионных моделей

– некоторые полученные регрессионные модели отличаются низкой точностью работы, поэтому доверие к их прогнозам будет минимальным;

– некоторые регрессионные модели хорошо сочетаются, так как позволяют друг другу компенсировать ошибки в прогнозах.

Для обеспечения согласованной работы всех полученных регрессионных моделей, необходимо связать выдаваемые ими прогнозы в единую систему. Это предложено сделать с помощью линейной функции вида (2.1):

$$Y(y_1,...,y_k) = y_1 \cdot k_1 + y_2 \cdot k_2 + ... + y_k \cdot k_k + b, \quad (2.1)$$

где, Y – прогнозное значение выдаваемое системой состоящей из множества регрессионных моделей; y_1 – прогноз, выдаваемый первой регрессионной моделью, y_2 – второй и т.д.; $k_1, k_2..., k_k, b$ – коэффициенты линейной модели.

Для согласования прогнозов моделей требуется на обучающей выборке определить значения коэффициентов $k_1, k_2..., k_k, b$. Найти требуемые значения коэффициентов можно, например, с помощью метода машинного обучения Linear Regression. При этом значение коэффициента близкое к нулю будет фактически отключать использование прогнозов от связанной с ним регрессионной модели при формировании итогового прогноза. Т.е. прогнозы от моделей с низкой точностью учитывать не будут.

Таким образом, в предложенном подходе достигается согласованная работа любого количества регрессионных моделей построенных с использованием различных алгоритмов машинного обучения.

Связывание результатов работы регрессионных моделей через линейную функцию показано на рисунке 6.

Разработка программного обеспечения, для тестирования предложенного рассматривается в 3 главе.

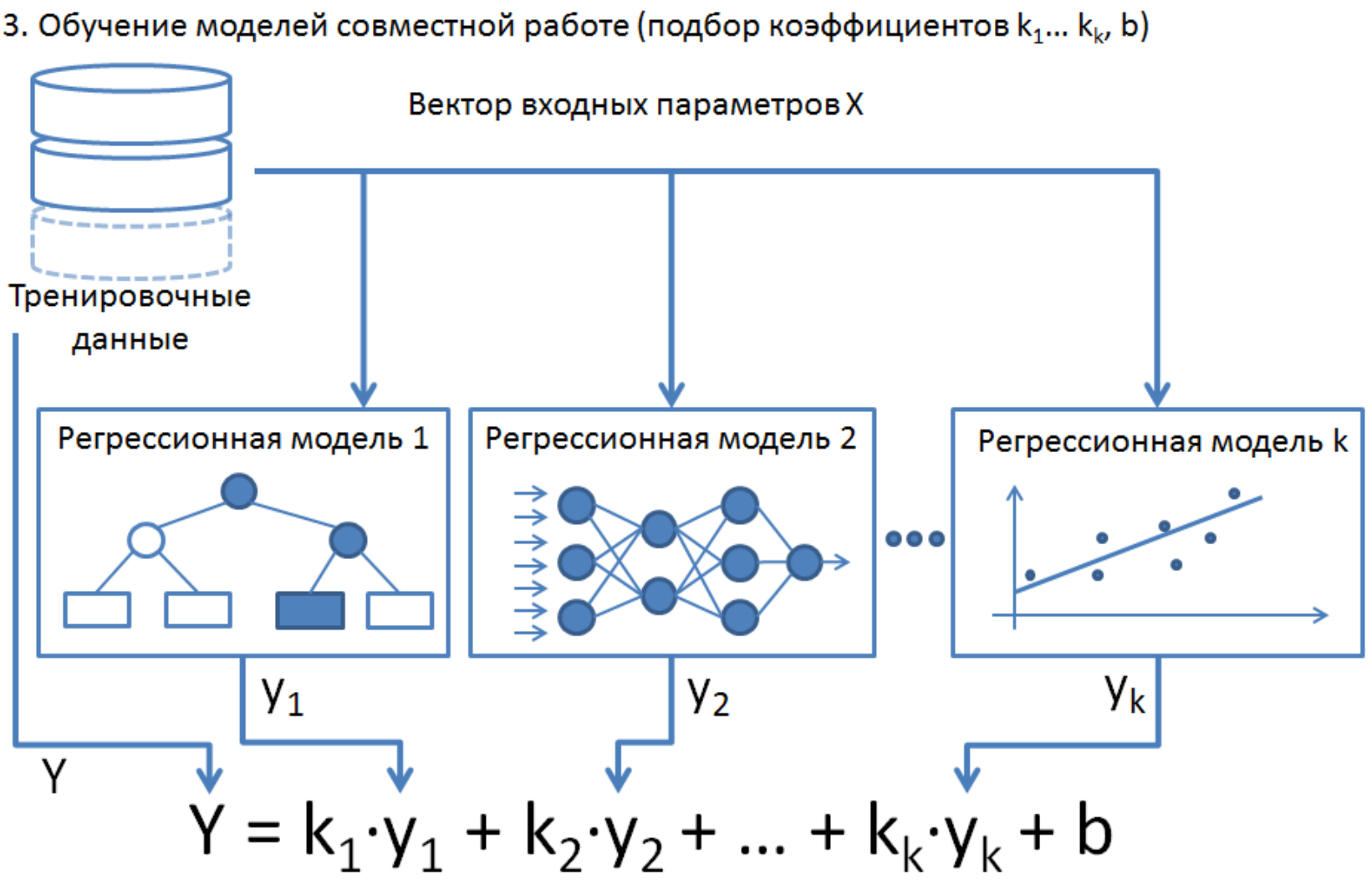

Рисунок 6 – Связывание результатов работы регрессионных моделей через линейную функцию

2.2 Данные для апробации смешанного машинного обучения

Для работы методов машинного обучения требуются обучающий набор данных. В открытых репозиториях существуют большое количество подготовленных для анализа данных. Однако стоит заметить, что эффективность методов машинного обучения во многом зависит от свойств анализируемых данных. Так точность некоторых методов машинного обучения зависит от полноты данных, количества информативных признаков и т.д. Поэтому с целью соблюдения объективности, тестирование предложенного подхода будет производиться на синтетических данных. Это является обычной практикой при проведении научных исследований.

В настоящее время существует большое количество библиотек для генерирования синтетических данных, предназначенных для тестирования методов машинного обучения. В данной работе будет использован генератор make_regression(), входящий в состав библиотеки sklearn. Данный генератор подробно рассматривается в 3 главе.

Выводы по главе

Предложена технология смешанного машинного обучения для построения составных регрессионных моделей. Алгоритм обучения составных регрессионных моделей включает в себя следующие этапы: разделение исходных данных на тренировочную и тестовую выборку, использование тренировочной выборки данных для построения множества регрессионных моделей с использованием выбранного набора методов машинного обучения и настройка полученных моделей для совместной работы. При этом задача согласования совместной работы моделей сведена к задаче поиска коэффициентов линейной функции, связывающей выходное значение составной модели (Y) с выходными значениями исходного набора регрессионных моделей ($y_1, y_2, \ldots, y_k$).

Глава 3 Программная реализация смешанного машинного обучения

3.1 Описание программного кода

Для тестирования технологии смешанного машинного обучения на языке программирования Python был разработано программное обеспечение, обладающее следующими ключевыми особенностями:

– наличие программной реализации подхода смешанного машинного обучения для обучения регрессионных моделей;

– реализована поддержка следующих базовых алгоритмов машинного обучения: Support Vector Regression, K-Neighbors Regressor, Descision Tree Regressor, Linear Regression;

– программа снабжена графическим интерфейсом с возможностью задания параметров для генерирования данных, выбора базовых алгоритмов машинного обучения, пропорции разделения исходных данных на тренировочную и тестовую выборку;

– реализованы средства визуализации для сравнения по точности получаемой составной регрессионной модели с моделями, построенными на основе отдельного использования алгоритмов машинного обучения;

– средства визуализации данных – табличное представление данных, полученных в результате тестирования регрессионных моделей и столбчатая диаграмма для сравнения точности регрессионных моделей по показателю MAE (mean absolute error).

При разработке программного обеспечения использовался компонентный подход, при котором реализации математических функций и алгоритмов машинного обучения из внешних библиотек.

В программном обеспечении использовались следующие свободно распространяемые библиотеки:

– библиотека numpy, обеспечивающая поддержку работы с массивами;

– библиотека sklearn, содержащая в себе реализации алгоритмов машинного обучения, генератор данных для построения регрессионных моделей, а также функции для разделения исходных данных на тренировочную и тестовую выборку;

– библиотека matplotlib, содержащая в себе методы для визуализации данных в виде столбчатой диаграммы;

– библиотека pandas, содержащая в себе методы для представления данных в табличном виде.

Программный код для подключения основных библиотек показан на рисунке 7, остальные библиотеки подключается в коде непосредственно перед их использованием.

```
from numpy import hstack
from sklearn.datasets import make_regression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.svm import SVR
```

Рисунок 7 – Программный код для экспорта библиотек и отдельных методов

Генерация данных, предназначенных для построения регрессионной модели, осуществляется с помощью метода make_regression(), оформленной в виде функции get_dataset().

В качестве параметров методу предаётся количество записей в наборе данных (n_samples), количество признаков в описании каждой записи (n_features), количество признаков (n_informative) влияющих на целевое

значение, диапазон случайного отклонения значений признаков (noise), целочисленной значение для случайной генерации данных (random_state). Программный код функции get_dataset() показан на рисунке 8.

```
def get_dataset():
  X, y = make_regression(n_samples=10000, n_features=20,
                         n_informative=10, noise=0.3, random_state=7)
  return X, y
```

Рисунок 8 – Программный код функции для генерирования данных регрессионной задачи

Так как смешанное машинное обучение основано на совместном использовании нескольких регрессионных моделей, то необходима функция, собирающая список используемых методов для построения этих моделей. Такой функцией является get_models(). Внутри функции сознается список models, который заполняется с использованием метода append(). В качестве параметров методу задается зазывание объекта и название одной из модели машинного обучения, реализуемой библиотекой sklearn(). Функция get_models() возвращает список models, программный код функции представлен на рисунке 9.

```
def get_models():
  models = list()
  models.append(('lr', LinearRegression()))
  models.append(('knn', KNeighborsRegressor()))
  models.append(('cart', DecisionTreeRegressor()))
  models.append(('svm', SVR()))
  return models
```

Рисунок 9 – Программный код функции для формирования набора базовых методов машинного обучения

Теперь необходимо обучить регрессионные модели с использованием методов, представленных в списке models. Для этого производится деление исходных данных (X, y), сгенерированных функцией get_dataset() на тренировочную выборку (X_train_full, y_train_full) и тестовую выборку (X_test, y_test). С помощью инструкции print() выводится информация о размере тренировочной и тестовой выборок данных.

Затем с помощью цикла for осуществляется последовательный переход по моделям списка models их обучение с помощью встроенного метода sklearn метода fit(). Для сбора статистической информации сразу после обучения регрессионной модели на тренировочных данных проверяется ее точность работы на тестовых данных посредством функции mean_absolute_error(). Рассчитанное значение точности выводится на экран. Программный код для выполнения данного этапа показан на рисунке 10.

```
X, y = get_dataset()
X_train_full, X_test, y_train_full,
y_test = train_test_split(X, y, test_size=0.5, random_state=1)
print('Train: %s, Test: %s' % (X_train_full.shape, X_test.shape))
models = get_models()
for name, model in models:
  model.fit(X_train_full, y_train_full)
  yhat = model.predict(X_test)
  score = mean_absolute_error(y_test, yhat)
  print('>%s MAE: %.3f' % (name, score))
```

Рисунок 10 – Программный код для тестирования методов машинного обучения отдельно друг от друга

В процессе обучения регрессионных моделей на экране пользователя будет отображаться информация о достигнутом значении MAE. Пример отображения информации по результатам обучения четырех регрессионных моделей показан на рисунке 11.

```
Train: (5000, 20), Test: (5000, 20)
>lr MAE: 1.236
>knn MAE: 100.169
>cart MAE: 133.536
>svm MAE: 138.195
```

Рисунок 11 – Результат обучения регрессионных моделей и их тестирования для определения точности

Для настройки согласованной работы полученных регрессионных моделей обходимо, как это показано на рисунке 6, подобрать коэффициенты линейного уравнения. Данное уравнение связывает выходное значение выдаваемой составной моделью и выходные значения регрессионных моделей по отдельности.

Настройка коэффициентов уравнения решается как задача обучения линейной регрессионной модели. Для обеспечения настройки коэффициентов разработана функция fit_ensemble(), принимающей в качестве параметров набор models настроенных регрессионных моделей, а также тренировочные (X_train, y_train) и тестовые (X_val, y_val) данные.

```
def fit_ensemble(models, X_train, X_val, y_train, y_val):
  meta_X = list()
  for _, model in models:
    model.fit(X_train, y_train)
    yhat = model.predict(X_val)
    yhat = yhat.reshape(len(yhat), 1)
    meta_X.append(yhat)
  meta_X = hstack(meta_X)
  blender = LinearRegression()
  blender.fit(meta_X, y_val)
  return blender
```

Рисунок 12 – Программный код для формирования регрессионной модели с использованием технологии смешанного машинного обучения

Составная модель храниться в переменной blender. Входными параметрами составной модели являются выходные значения отдельных регрессионных моделей, собранных в массив meta_x. Программный код для обучения составной модели показан на рисунке 12.

Для расчёта выходных значений составной модели blender на наборе входных данных X_test разработана функция predict_ensemble(). Данная функция для входных значений последовательно собирает предсказания регрессионных моделей, хранящихся в списке models, а затем, с учетом найденных коэффициентов линейного уравнения, рассчитывает выходное значение составной модели. Программный код функции представлен на рисунке 13.

```
def predict_ensemble(models, blender, X_test):
  meta_X = list()
  for _, model in models:
    yhat = model.predict(X_test)
    yhat = yhat.reshape(len(yhat), 1)
    meta_X.append(yhat)
  meta_X = hstack(meta_X)
  return blender.predict(meta_X)
```

Рисунок 13 – Функция для расчёта выходного значения составной модели

Теперь, когда все основные функции для обучения составной регрессионной модели определены, можно переходить к описанию кода, отвечающего за расчёт эффективности модели.

Для того, чтобы оценить точность составной модели необходимо обеспечить расчёт предсказания на основе вектора входных параметров.

Программный код, демонстрирующий пример расчёта предсказания составной модели на основе вектора входных значений, с использованием разработанных функций, показан на рисунке 14.

```
X, y = get_dataset()
X_train, X_val, y_train,
y_val = train_test_split(X, y, test_size=0.33, random_state=1)
print('Train: %s, Val: %s' % (X_train.shape, X_val.shape))
models = get_models()
blender = fit_ensemble(models, X_train, X_val, y_train, y_val)
row = [-0.24038754, 0.55423865, -0.48979221, 1.56074459,
       -1.16007611, 1.10049103, 1.18385406, -1.57344162,
       0.97862519, -0.03166643, 1.77099821, 1.98645499,
       0.86780193, 2.01534177, 2.51509494, -1.04609004,
       -0.19428148, -0.05967386, -2.67168985, 1.07182911]
yhat = predict_ensemble(models, blender, [row])
print('Predicted: %.3f' % (yhat[0]))
```

Рисунок 14 – Программный код, рассчитывающий выходное значение составной модели для конкретного вектора входных значений

В результате выполнения программного кода, показанного на рисунке 14, на экран выведется предсказание составной регрессионной модели в виде текстовой строки (рисунок 15).

```
Train: (6700, 20), Val: (3300, 20)
Predicted: 359.988
```

Рисунок 15 – Результат расчета выходного значения составной модели для конкретного вектора входных значений

Для того, чтобы протестировать точность составной модели, необходимо собрать все предсказания на основе массива данных X_test. Данный шаг осуществляется с помощью функции predict_ensemble(), при этом предсказания объединяются в вектор и помещаются в переменную yhat.

Чтобы рассчитать показатель MAE точности составной модели используется метод mean_absolute_error(), в который, в качестве параметров предаются предсказанные составной моделью значения yhat и ожидаемые

значения y_test, взятые из тестовых данных. Программный код и результат выполнения данного шага показан на рисунке 16.

```
X, y = get_dataset()
X_train_full, X_test, y_train_full,
y_test = train_test_split(X, y, test_size=0.5, random_state=1)
X_train, X_val, y_train, y_val =
train_test_split(X_train_full, y_train_full, test_size=0.33, random_state=1)
print('Train: %s, Val: %s, Test: %s' %
      (X_train.shape, X_val.shape, X_test.shape))
models = get_models()
blender = fit_ensemble(models, X_train, X_val, y_train, y_val)
yhat = predict_ensemble(models, blender, X_test)
score = mean_absolute_error(y_test, yhat)
print('Blending MAE: %.3f' % score)
```

```
Train: (3350, 20), Val: (1650, 20), Test: (5000, 20)
Blending MAE: 0.237
```

Рисунок 16 – Программный код для обучения составной модели на тренировочных данных и определения точности ее работы на тестовых данных

Теперь, когда у нас есть показатель точности MAE как для составной модели, так и для каждой отдельно взятой модели, для удобства сравнения объединим их в таблицу, и выведем на экран.

```
import pandas as pd

df = pd.DataFrame([lr_mae, knn_mae, cart_mae, svm_mae, blend_mae],
                  columns = ['Mean absolute error'],
                  index=['Linear Regression',
                         'KNeighbors Regressor',
                         'Decision Tree Regressor',
                         'Support Vector Regressor',
                         'Смешанное машинное обучение'])
df
```

Рисунок 17 – Программный код для визуализации результатов по сравнению точности в виде таблицы

Для этого воспользуемся библиотекой pandas, обеспечивающей возможность табличного представления массивов данных. В переменной df создается табличное представление данных путем вызова метода DataFrame() с передачей значений MAE, полученных при тестировании регрессионных моделей по отдельности и составной модели, основанной на смешанном машинном обучении. Для удобства восприятия таблицы также необходимо передать строковые списки (columns и index), которые будут использованы для подписи, соответственно, столбцов и строк.

Программный код для создания табличного представления данных показан на рисунке 17.

	Mean absolute error
Linear Regression	1.236
KNeighbors Regressor	100.169
Decision Tree Regressor	133.536
Support Vector Regressor	138.195
Смешанное машинное обучение	0.237

Рисунок 18 – Визуализации результатов по сравнению точности в виде таблицы

Результат табличного представления данных показан на рисунке 18.

Для визуального сравнения данных, представленных в таблице, было предложено выполнить построение столбчатой диаграммы. Для построения столбчатой диаграммы использован метод bar(), реализация которого содержится в библиотеке matplotlib.

Для построения столбчатой диаграммы сначала формируется два списка хранящихся в переменных x и y. В первом списке строковые переменные для подписей элементов столбчатой диаграммы, а во втором – визуализируемые числовые значения. Формирование диаграммы осуществляется с помощью инструкции ax.bar(x, y). Затем с помощью команды set_ylabel задается подпись вертикальной оси, а с помощью команд set_figwidth и set_figheight задается ширина и высота столбчатой диаграммы в условных единицах.

Программный код для построения столбчатой диаграммы показан на рисунке 19.

```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt

x = ['Linear\nRegression', 'KNeighbors\nRegressor',
     'DecisionTree\nRegressor', 'SupportVector\nRegressor',
     'Смешанное\nмаш. обуч.']
y = [lr_mae, knn_mae, cart_mae, svm_mae, blend_mae]

fig, ax = plt.subplots()

ax.bar(x, y)
ax.set_ylabel('Mean absolute error')
ax.set_facecolor('seashell')
fig.set_facecolor('floralwhite')
fig.set_figwidth(7)    #  ширина
fig.set_figheight(5)    #  высота

plt.show()
```

Рисунок 19 – Программный код для визуализации результатов по сравнению точности в виде столбчатой диаграммы

Результат отображения столбчатой диаграммы показан на рисунке 20.

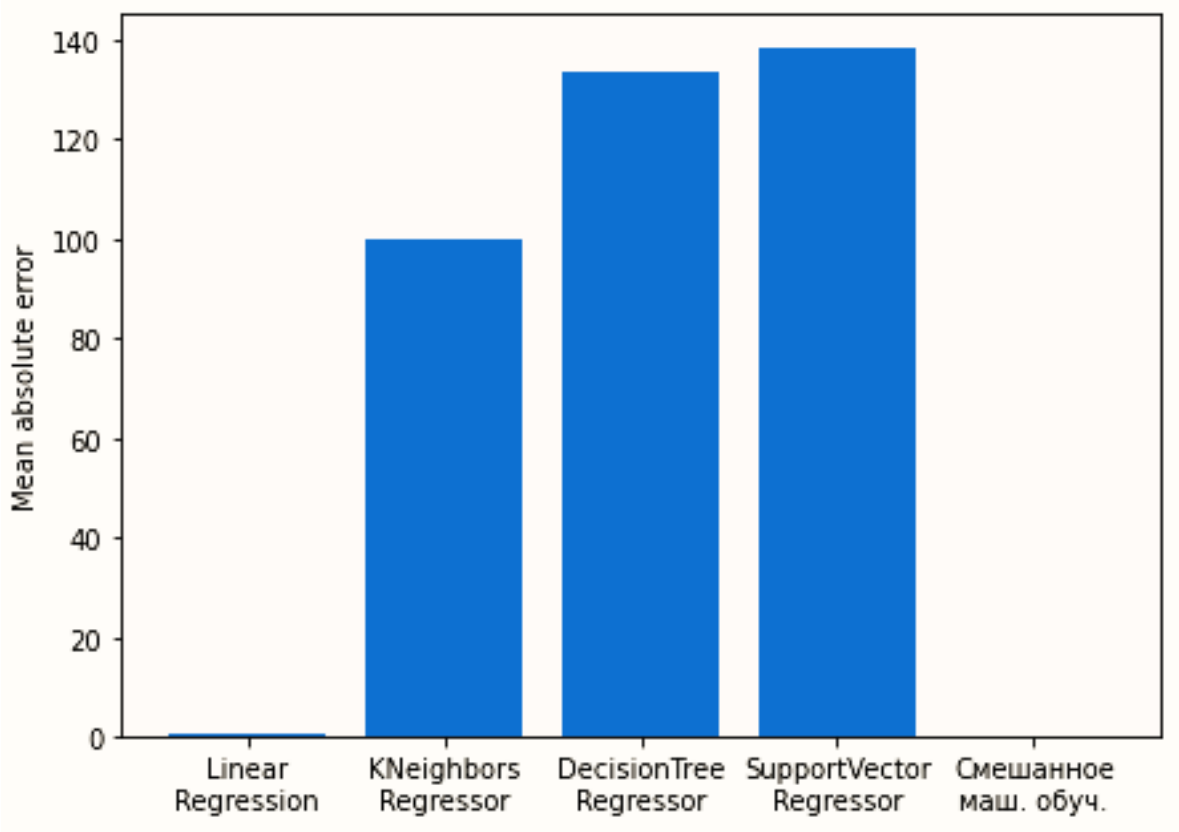

Рисунок 20 – Визуализации результатов по сравнению точности в виде столбчатой диаграммы

Для удобства работы с программным обеспечением был разработан графический интерфейс пользователя.

3.2 Интерфейс программного обеспечения

Разработанный интерфейс разделен на 5 логических частей:

– генерирование данных для построения регрессионной модели;
– планирование вычислительного эксперимента;
– запуск вычислений;
– визуализация результатов виде таблицы;
– визуализация результатов в виде столбчатой диаграммы.

Первые три части показаны на рисунке 21. В первой части интерфейса можно задать такие параметры для генерирования данных, как количество записей в наборе данных (n_samples), количество признаков в описании каждой записи (n_features), количество признаков (n_informative) влияющих на целевое значение, диапазон случайного отклонения значений признаков (noise), целочисленной значение для случайной генерации данных (random_state).

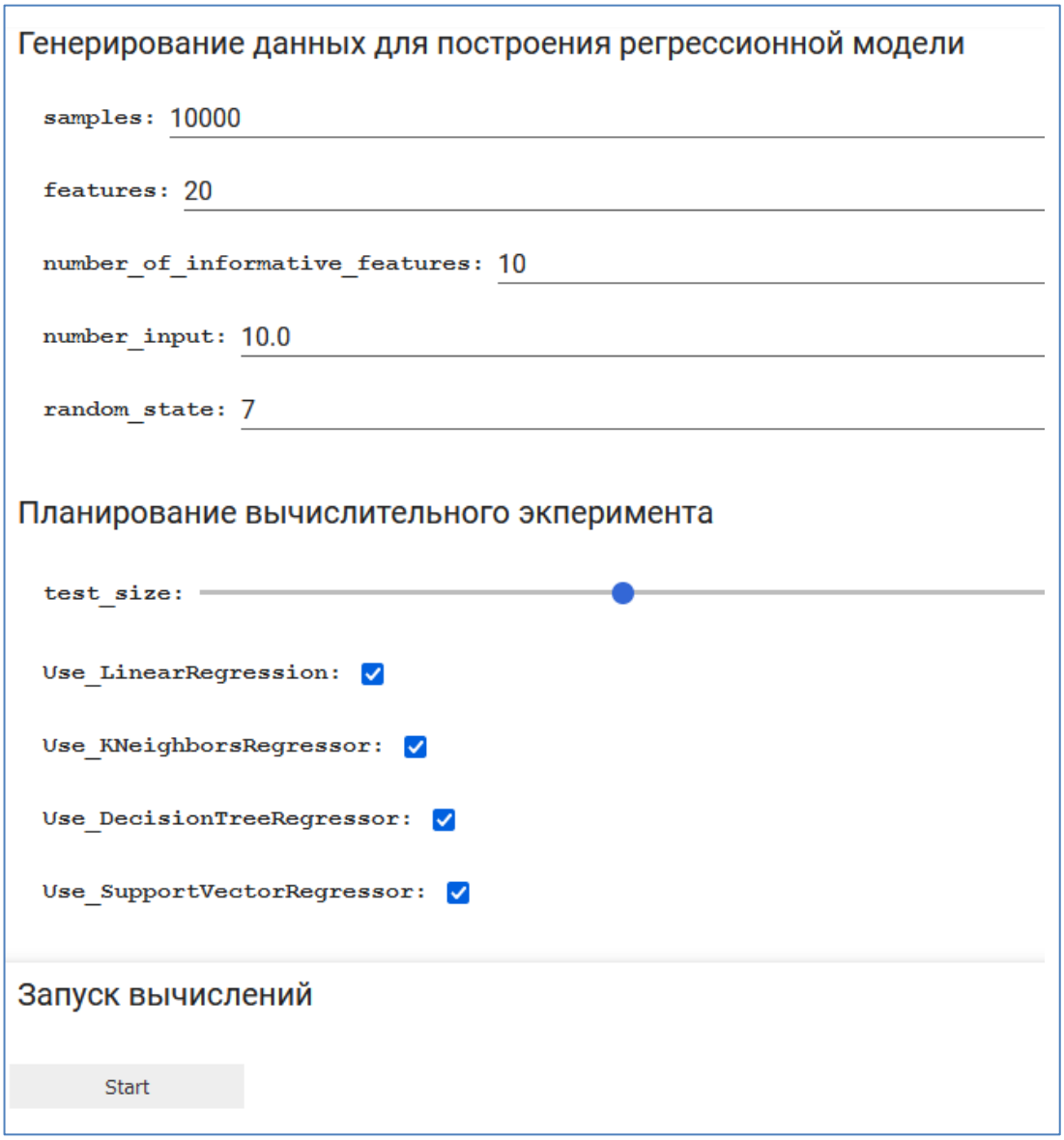

Рисунок 21 – Блоки графического интерфейса «Генерирование данных для построения регрессионной модели», «Планирование вычислительного эксперимента» и «Запуск вычислений»

Во второй части интерфейса («Планирование вычислительного эксперимента») можно задать пропорцию разделения исходных данных на тестовую и тренировочную выборку. Здесь также можно выбрать методы, которые будут использованы при настройке составной регрессионной модели (с применением предложенного подхода смешанного машинного обучения).

Третья часть интерфейса содержит кнопку для запуска вычислений.

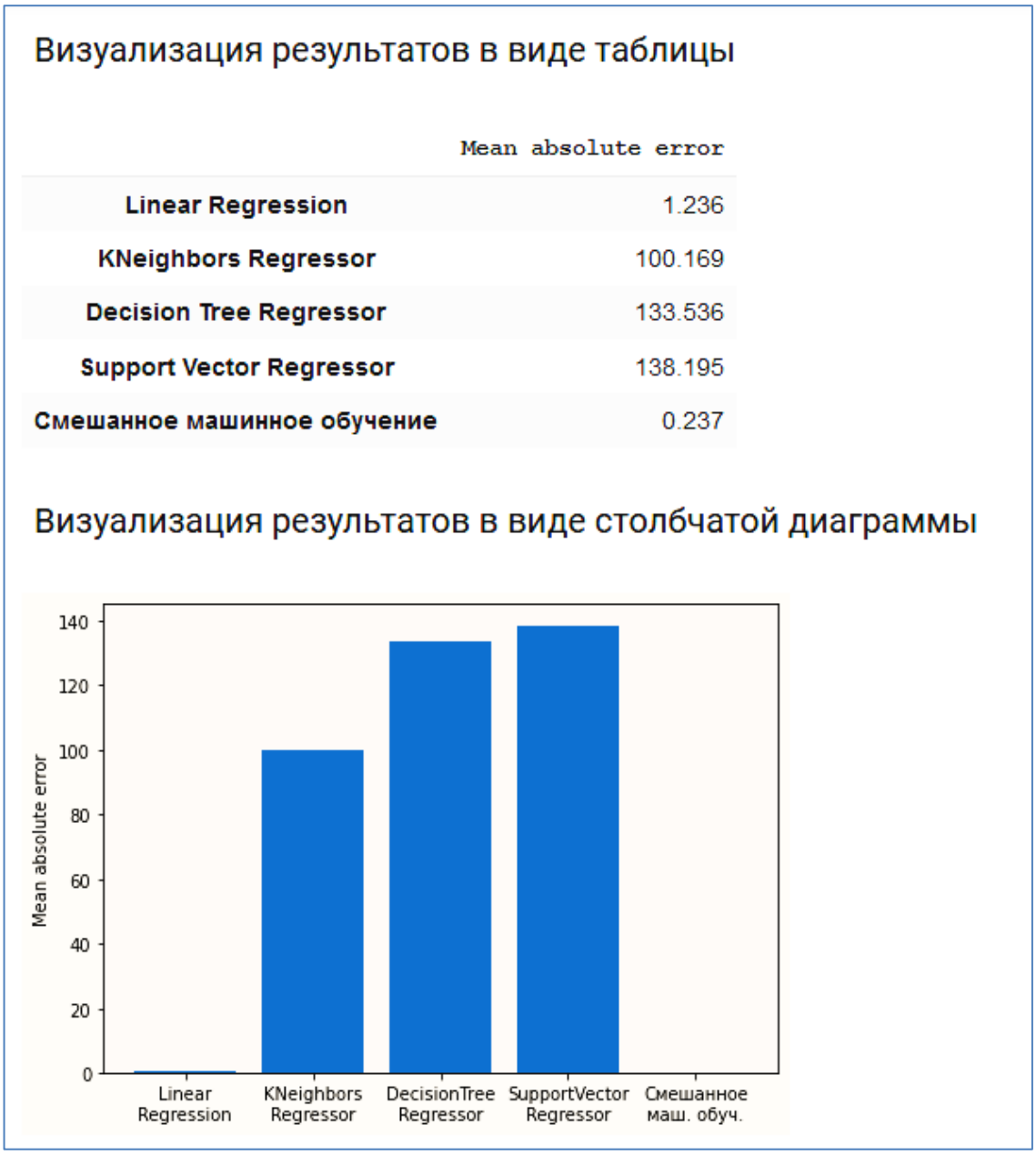

Рисунок 22 – Блоки графического интерфейса «Визуализация результатов в виде таблицы», «Визуализация результатов в виде столбчатой диаграммы»

Последние два блока содержат в себе элементы визуализации данных в виде таблицы и столбчатой диаграммы (рисунок 22).

Разработанный графический интерфейс обеспечивает удобство проведения вычислительных экспериментов для тестирования эффективности применения предложенной технологии смешанного машинного обучения.

3.3 Результаты тестирования

Тестирование технологии смешанного машинного обучения проводилось на данных, синтезированных с помощью функции make_regression() из библиотеки sklearn. В рамках одного вычислительного эксперимента на одних и тех же данных проводилась обучение не скольких регрессионных моделей:

- составная модель, основанная на смешанном машинном обучении;
- модель с применение метода Support Vector Regression;
- модель с применение метода K-Neighbors Regressor;
- модель с применение метода Descision Tree Regressor;
- модель с применение метода Linear Regression.

Точность всех моделей проверялась на тестовой выборке данных с расчётом ошибки MAE (средняя абсолютная ошибка).

Было проведено более 20 вычислительных экспериментов. В 19 экспериментах наибольшую точность (наименьшее значение ошибки MAE) показали регрессионные модели, основанные на смешанном машинном обучение.

Таким образом, доказана эффективность применения смешанного машинного обучения при построении регрессионных моделей.

Однако стоит отметить что, вычислительные эксперименты проводились толь ка синтезированных данных. Эффективность алгоритмов машинного обучения на конкретном реальном наборе данных следует проверять отдельно.

Выводы по главе

На языке программирования Python разработано программное обеспечение, позволяющее тестировать технологию смешанного машинного обучения и сравнивать ее эффективность (точность работы по значению MAE) с классическими методами машинного обучения (Support Vector Regression, K-Neighbors Regressor, Descision Tree Regressor, Linear Regression). Разработанное программное обеспечение обладает графическим интерфейсом и средствами визуализация результатов вычислительных экспериментов посредством таблиц и столбчатых диаграмм.

Было проведено более 20 вычислительных экспериментов на разных наборах данных. В 19 экспериментах наибольшую точность (наименьшее значение ошибки MAE) показали регрессионные модели, настроенные с использованием предложенной технологии смешанного машинного обучения. Это на практике доказывает эффективность предложенной технологии.

Заключение

В ходе выполнения бакалаврской работы были получены следующие результаты:

– Было произведено сравнение подходов по использования алгоритмов машинного обучения для получения составных прогнозных моделей (ансамблей). При сравнении рассматривались такие ансамблевые методы, как бэггинг (Bagging), случайный лес (Random Forest), бустинг (Boosting), градиентный бустинг (Gradient Boosting), параллельный градиентный бустинг (XGBoost). По результатам сравнения была составлена схема, показывающая, как соотносятся скорости обучения ансамблевых методов и точности прогнозирования получаемых моделей.

– В ходе анализа ансамблевых методов установлено, что их использование основано на применении одно базового алгоритма, например, например алгоритма построения дерева принятия решения. Поэтому предложено разработать технология смешанного машинного обучения позволяющего использовать не один, а сразу несколько базовых алгоритмов при построении составных прогнозных моделей.

– Предложена технология смешанного машинного обучения для построения составных регрессионных моделей. Алгоритм обучения составных регрессионных моделей включает в себя следующие этапы: разделение исходных данных на тренировочную и тестовую выборку, использование тренировочной выборки данных для построения множества регрессионных моделей с использованием выбранного набора методов машинного обучения и настройка полученных моделей для совместной работы. При этом задача согласования совместной работы моделей сведена к задаче поиска коэффициентов линейной функции, связывающей выходное значение составной модели (Y) с выходными значениями исходного набора регрессионных моделей ($y_1, y_2, \ldots, y_k$).

– На языке программирования Python разработано программное обеспечение, позволяющее тестировать технологию смешанного машинного обучения и сравнивать ее эффективность (точность работы по значению MAE) с классическими методами машинного обучения (Support Vector Regression, K-Neighbors Regressor, Descision Tree Regressor, Linear Regression). Разработанное программное обеспечение обладает графическим интерфейсом и средствами визуализация результатов вычислительных экспериментов посредством таблиц и столбчатых диаграмм.

– Было проведено более 20 вычислительных экспериментов на разных наборах данных. В 19 экспериментах наибольшую точность (наименьшее значение ошибки MAE) показали регрессионные модели, настроенные с использованием предложенной технологии смешанного машинного обучения. Это на практике доказывает эффективность предложенной технологии.

Список используемой литературы

1. Андриенко М.П. Понимание повторяющихся нейронных сетей: предпочтительная нейронная сеть для данных временных рядов / М.П. Андриенко, П.А. Юдин, Е.Ю. Вишневецкая // Актуальные вопросы современной науки и образования: сборник статей Международной научно-практической конференции : в 2 ч.. 2020. – Издательство: Наука и Просвещение (Пенза), 2020. – с. 96-98. – Текст : непосредственный.

2. Аусабаев, Д.М. Использование машинного обучения в поддержке принятия решений / Д.М. Аусабаев, О.П. Волобуев // Прикладная математика и информатика: современные исследования в области естественных и технических наук – материалы III научно-практической всероссийской конференции (школы-семинара) молодых ученых. Тольятти, 24–25 апреля 2017 года. – Издатель Качалин Александр Васильевич, 2017. – с. 43-47. – Текст : непосредственный.

3. Власов, А.В. Машинное обучение применительно к задаче классификации семян зерновых культур в видеопотоке / А.В. Власов, А.С. Федеев // Молодежь и современные информационные технологии – сборник трудов XIV Международной научно-практической конференции студентов, аспирантов и молодых учёных, 07–11 ноября 2016. – Национальный исследовательский Томский политехнический университет (Томск), 2016. – с. 133-135. – Текст : непосредственный.

4. Грушевская, А.Л. Сравнительный анализ решения одной задачи классификации четырьмя типами нейронных сетей / А.Л. Грушевская, А.Н. Покровский // Современные методы прикладной математики, теории управления и компьютерных технологий (ПМТУКТ-2013) : сборник трудов VI международной конференции, 10–16 сентября 2013 года. – Издательство: Воронежский государственный университет (Воронеж), 2013. – с. 83-84. – Текст : непосредственный.

5. Дорогов, А.Ю. Нейронные сети глубокого обучения с управляемой коммутацией нейронных плоскостей / А.Ю. Дорогов // Дистанционные образовательные технологии: Материалы IV Всероссийской научно-практической конференции (с международным участием), 2019. – Издательство: Общество с ограниченной ответственностью «Издательство Типография «Ариал» (Симферополь), 2019. – с. 284-296. – Текст : непосредственный.

6. Жуков, Д.А. Формирование контрольных выборок при технической диагностике объекта с применением машинного обучения / Д.А. Жуков, А.С. Хорева, Ю.Е. Кувайскова, В.Н. Клячкин // Математические методы и модели: теория, приложения и роль в образовании – международная научно-техническая конференция : сборник научных трудов, 28–30 апреля 2016 года. – Ульяновский государственный технический университет (Ульяновск), 2016. – с. 44-48. – Текст : непосредственный.

7. Ибрагимов Р.М. Влияние функций активации нейронных сетей на скорость обучения на примере нейронной сети с обратным распространением ошибки / Р.М. Ибрагимов // Актуальные проблемы физической и функциональной электроники: материалы 21-й Всероссийской молодежной научной школы-семинара. 2018. – Издательство: Ульяновский государственный технический университет (Ульяновск), 2018. – с. 125-126. – Текст : непосредственный.

8. Иванников Ю.Ю. Применение методов машинного обучения для выявления бот-трафика среди запросов к веб-приложению / Ю.Ю. Иванников, Е.Ю. Митрофанова // Сборник студенческих научных работ факультета компьютерных наук ВГУ, Факультет компьютерных наук, 2017. – ФГБОУ ВО «Воронежский государственный университет», 2017. – с. 119-123. – Текст : непосредственный.

9. Клячин В.Н. Использование агрегированных классификаторов при технической диагностике на базе машинного обучения / В.Н. Клячин, Ю.Е. Кувайскова, Д.А. Жуков // Информационные технологии и

нанотехнологии (ИТНТ-2017) – сборник трудов III международной конференции и молодежной школы. Самарский национальный исследовательский университет имени академика С.П. Королева. 2017. – Предприятие "Новая техника" (Самара), 2017. – с. 1770-1773. – Текст : непосредственный.

10. Кононова, Н.В. Исследование подсистемы контентной фильтрации с использованием методов машинного обучения / Н.В. Кононова, Ю.А. Андрусенко, Т.А. Самокаева // Студенческая наука для развития информационного общества – сборник материалов VI Всероссийской научно-технической конференции. 22–26 мая 2017. – Северо-Кавказский федеральный университет (Ставрополь), 2017. – с. 268-270. – Текст : непосредственный.

11. Мелдебай, М.А. Анализ мнений покупателей на основе машинного обучения / М.А. Мелдебай, А.К. Сарбасова // Прикладная математика и информатика: современные исследования в области естественных и технических наук – материалы III научно-практической всероссийской конференции (школы-семинара) молодых ученых. 24–25 апреля 2017 года. – Издатель Качалин Александр Васильевич, 2017. – с. 360-363. – Текст : непосредственный.

12. Осколков, В.М. Применение параллельных вычислений для прогнозирования на основе алгоритма машинного обучения Random Forest / В.М. Осколков, Н.И. Шаханов, И.А. Варфоломеев, О.В. Юдина, Л.Н. Виноградова, Е.В. Ершов // Сборник трудов конференции Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации. Распознавание, Курск, 16–19 мая 2017 года. – Юго-Западный государственный университет (Курск), 2017. – с. 267-269. – Текст : непосредственный.

13. Синягов, А.И. Реализация искусственной нейронной сети на базе нейронной сети Петри / А.И. Синягов, А.А. Суконщиков // Шаг в будущее: искусственный интеллект и цифровая экономика. Материалы 1-й

Международной научно-практической конференции. Государственный университет управления. 2017. – Издательство: Государственный университет управления (Москва), 2017. – с. 130-135. – Текст : непосредственный.

14. Урубкин, М.Ю. Нейронные сети Кохонена и нечеткие нейронные сети в интеллектуальном анализе данных / М.Ю. Урубкин, А.В. Авакьянц // Совершенствование методологии познания в целях развития науки: сборник статей по итогам Международной научно-практической конференции: в 2 частях. 2017. – Издательство: Общество с ограниченной ответственностью "Агентство международных исследований" (Уфа), 2017. – с. 36-39. – Текст : непосредственный.

15. Федотов, И.А. Применение технологий машинного обучения для прогнозирования ситуации на финансовых рынках / И.А. Федотов // Студенческая наука для развития информационного общества – сборник материалов VI Всероссийской научно-технической конференции. 22–26 мая 2017. – Северо-Кавказский федеральный университет (Ставрополь), 2017. – с. 361-363. – Текст : непосредственный.

16. Якимчук, А.А. Глубокое обучение как эффективный метод машинного обучения / А.А. Якимчук // Научное сообщество студентов XXI столетия. Технические науки – сборник статей по материалам XCII студенческой международной научно-практической конференции. 2020. – ООО “Сибирская академическая книга” (Новосибирск), 2020. – с. 40-43. – Текст : непосредственный.

17. Ярыгин, А.А. Актуальные вопросы машинного обучения с подкреплением интеллектуальных агентов в задачах принятия решений / А.А. Ярыгин // Автоматизация: проблемы, идеи, решения - сборник статей по итогам Международной научно-практической конференции 2017. – ООО "Агентство международных исследований", 2017. – с. 62-68. – Текст : непосредственный.

18. Bartczuk, Ł. A New Version of the Fuzzy-ID3 Algorithm / Łukasz Bartczuk, Danuta Rutkowska // International Conference on Artificial Intelligence and Soft Computing – 8th International Conference, Zakopane, Poland, June 25-29, 2006. Proceedings: Artificial Intelligence and Soft Computing – ICAISC 2006. – Springer-Verlag Berlin Heidelberg 2006. – pp. 1060-1070. – Текст : непосредственный.

19. Filipczuk P. Automatic Breast Cancer Diagnosis Based on K-Means Clustering and Adaptive Thresholding Hybrid Segmentation [Text] / Paweł Filipczuk, Marek Kowal, Andrzej Obuchowicz // Image Processing and Communications Challenges 3 – Advances in Intelligent and Soft Computing – Springer-Verlag Berlin Heidelberg 2011. – pp. 295-302. – Текст : непосредственный.

20. Fu L. A Robust Text Segmentation Approach in Complex Background Based on Multiple Constraints [Text] / Libo Fu, Weiqiang Wang, Yaowen Zhan // Pacific-Rim Conference on Multimedia – 6th Pacific Rim Conference on Multimedia, Jeju Island, Korea, November 13-16, 2005, Proceedings, Part I: Advances in Multimedia Information Processing - PCM 2005. – Springer-Verlag Berlin Heidelberg 2005. – pp. 594-605. – Текст : непосредственный.

21. Huang Z. Chinese Historic Image Threshold Using Adaptive K-means Cluster and Bradley's [Text] / Zhi-Kai Huang, Yong-Li Ma, Li Lu, Fan-Xing Rao, Ling-Ying Hou // 12th International Conference, ICIC 2016, Lanzhou, China, August 2-5, 2016, Proceedings, Part III – ICIC 2016: Intelligent Computing Methodologies. – Springer International Publishing Switzerland 2016. – pp. 171-179. – Текст : непосредственный.

22. Jiang, S. A Combined Classification Algorithm Based on C4.5 and NB / ShengYi Jiang, Wen Yu // International Symposium on Intelligence Computation and Applications – Third International Symposium, ISICA 2008 Wuhan, China, December 19-21, 2008. Proceedings: Advances in Computation and Intelligence. – Springer-Verlag Berlin Heidelberg 2008. – pp. 350-359. – Текст : непосредственный.

23. Min, F. A Competition Strategy to Cost-Sensitive Decision Trees / Fan Min, William Zhu // International Conference on Rough Sets and Knowledge Technology – 7th International Conference, RSKT 2012, Chengdu, China, August 17-20, 2012. Proceedings: Rough Sets and Knowledge Technology. – Springer-Verlag Berlin Heidelberg 2012. – pp. 359-368. – Текст : непосредственный.

24. Zhao, C. Recommendation Based Heterogeneous Information Network and Neural Network Model / Cong Zhao, Yan Wen, Ming Chen, Geng Chen // International Conference on Wireless and Satellite Systems: 11th EAI International Conference, WiSATS 2020, Nanjing, China, September 17-18, 2020, Proceedings, Part II. – Springer Nature Switzerland AG, 2021. – pp. 588-598. – Текст : непосредственный.

25. Zheng, Z. Scaling up the rule generation of C4.5 / Zijian Zheng // Pacific-Asia Conference on Knowledge Discovery and Data Mining – Second Pacific-Asia Conference, PAKDD-98 Melbourne, Australia, April 15–17, 1998. Proceedings: Research and Development in Knowledge Discovery and Data Mining. – Springer-Verlag Berlin Heidelberg 1998. – pp. 348-359. – Текст : непосредственный. – Текст : непосредственный.