Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К КУРСОВОМУ ПРОЕКТУ

НА ТЕМУ:

Система предсказания успеваемости студентов

Студент ИУ7-23М _____ **Е. А. Варламова**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель курсового проекта _____ **Солодовников В.И.**
(Подпись, дата) (И.О.Фамилия)

2024 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ИУ-7
(Индекс)
_____ И. В. Рудаков
(И.О.Фамилия)
«15» февраля 2024 г.

З А Д А Н И Е
на выполнение курсовой работы

по дисциплине Методы машинного обучения

Студент группы ИУ7-23М

Варламова Екатерина Алексеевна
(Фамилия, имя, отчество)

Тема курсовой работы
Система предсказания успеваемости студентов

Направленность КР (учебная, исследовательская, практическая, производственная, др.)
учебная
Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения работы: 25% к ___ нед., 50% к ___ нед., 75% к __ нед., 100% к ___ нед.

Задание:

Решить задачу регрессии, состоящую в определении выпускного балла студентов, основываясь на данных об их образе жизни. Для этого необходимо: провести анализ возможных методов решения задачи, выбрать модель для решения задачи и функционал качества модели; описать набор данных и визуализировать его; осуществить предобработку данных и обучить модель; провести анализ полученного метода решения исходной задачи, приведя значения функционала качества и оценив обобщающую способность.

Оформление курсовой работы:

Расчетно-пояснительная записка на 25-35 листах формата А4. Расчетно-пояснительная записка должна содержать постановку введение, аналитическую часть, конструкторскую часть, технологическую часть, заключение, список литературы.

Дата выдачи задания «15» февраля 2024 г.

Руководитель курсовой работы _____ **Солодовников В.И.**
(Подпись, дата) (И.О.Фамилия)

Студент _____ **Варламова Е. А.**
(Подпись, дата) (И.О.Фамилия)

Содержание

Введение

В современном образовании одним из ключевых моментов является определение выпускного балла студентов, который отражает их успеваемость и готовность к завершающему этапу обучения. Однако, при анализе факторов, влияющих на формирование этого показателя, становится ясно, что образ жизни студентов играет существенную роль.

Исследование данных об образе жизни студентов может дать ценную информацию о их поведении, привычках, уровне активности и здоровье, которые в свою очередь могут оказывать влияние на их успеваемость и результаты обучения. Факторы, такие как режим дня, питание, физическая активность, уровень стресса и т.д., могут быть важными при определении выпускного балла студентов.

Таким образом, анализ данных об образе жизни студентов может помочь выявить взаимосвязи между их поведением и успехами в учебе, что в свою очередь может быть использовано для более точного определения критериев успешности обучения и разработки эффективных стратегий поддержки студентов. В данном контексте, использование методов анализа данных и моделей регрессии может стать мощным инструментом для выявления закономерностей и прогнозирования выпускного балла студентов на основе данных об их образе жизни.

Цель работы – решить задачу регрессии, состоящую в определении выпускного балла студентов с помощью информации об их образе жизни, на данных о студентах двух португальских школ, взятых с kaggle [1].

Для достижения поставленной цели требуется решить следующие задачи:

— описать набор данных и визуализировать его;

— проанализировать существующие модели регрессии для решения задачи и выбрать наиболее подходящую, а также выбрать функционал качества модели;

— описать алгоритм предобработки данных;

— описать общий алгоритм работы ПО, осуществляющего решение задачи регрессии;

— разработать ПО, решающее задачу регрессии;

— выбрать гиперпараметры модели, с которыми модель работает наилучшим образом с точки зрения выбранного функционала качества, оценить обобщающую способность;

— провести исследование ПО с целью сравнения полученной модели с другими моделями.

1 Аналитический раздел

В данном разделе описывается и визуализируется набор данных, формализуется задача, проводится анализ существующих моделей регрессии осуществляется выбор наиболее подходящей для решения поставленной задачи, а также выбирается функционал качества модели.

1.1 Описание и анализ набора данных

Набор данных, на основе которого разрабатывается система предсказания выпускного балла по образу жизни студентов, состоит из следующих признаков:

1. school – название школы (бинарный: 'GP' – Gabriel Pereira, 'MS' - Mousinho da Silveira);
2. sex – пол (бинарный: 'F' – женский; 'M' – мужской);
3. age – возраст (числовой: от 15 до 22);
4. addres – тип места жительства (бинарный: 'U' – город, 'R' - деревня);
5. famsize – размер семьи (бинарный: 'LE3' – меньше 3, 'GT3' – больше 3);
6. pstatus – статус родителей (бинарный: 'T' – живут вместе, 'A' – отдельно);
7. medu – образование матери (числовой: 0 – нет образования, 1 – начальное, 2 – базовое среднее, 3 – полное среднее, 4 – высшее);
8. fedu – образование отца (числовой: 0 – нет образования, 1 – начальное, 2 – базовое среднее, 3 – полное среднее, 4 – высшее);
9. mjob – работа матери (номинальный: 'teacher' – учитель, 'health' – связан со здровоохранением, 'services' – гражданские сервисы, 'at_home' – не трудоустроен, 'other' – другое);

10. fjob – работа отца (номинальный: 'teacher' – учитель, 'health' – связан со здровоохранением, 'services' – гражданские сервисы, 'at_home' – не трудоустроен, 'other' – другое);

11. reason – причина выбор школы (номинальный: 'home' – близко к дому, 'reputation' – репутация школы, 'course' – предпочтительная образовательная программа, 'other' – другое);

12. guardian – опекун (номинальный: 'mother' – мать, 'father' – папа, 'other' – другое);

13. traveltime – время пути в школу (числовой: 1 – меньше 15 минут, 2 – от 15 минут до 30 минут, 3 – от 30 минут до 45 минут, 4 – больше часа);

14. studytime – время на самостоятельную подготовку в неделю (числовой: 1 – меньше 2 часов, 2 – от 2 до 5 часов, 3 – от 5 до 10 часов, 4 – больше 10 часов);

15. failures – количество проваленных контрольных (числовое: если n в диапазоне 1-3, то n, иначе 4);

16. schoolsup – дополниьельные занятия (бинарный: 'yes' – да, 'no' – нет);

17. famsup – помощь семьи в учёбе (бинарный: 'yes' – да, 'no' – нет);

18. paid – дополнительные оплачиваемые занятия (бинарный: 'yes' – да, 'no' – нет);

19. activities – наличие хобби (бинарный: 'yes' – да, 'no' – нет);

20. nursery – посещал ли детский сад (бинарный: 'yes' – да, 'no' – нет);

21. higher – хоачет получить высшее образование (бинарный: 'yes' – да, 'no' – нет);

22. internet – есть доступ в интернет дома (бинарный: 'yes' – да, 'no' – нет);

23. romantic – состоит в романтических отношениях (бинарный: 'yes' – да, 'no' – нет);

24. famrel – качество отношений в семье (числовое: от 1 – плохо до 5 – хорошо);

25. freettime – свободное время после занятий (числовое: от 1 – мало до 5 – много);

26. goout – ходит на прогулки с друзьями (числовое: от 1 – редко до 5 – часто);

27. dalc – употребляет алкоголь в рабочие дни (числовое: от 1 – редко до 5 – часто);

28. walc – употребляет алкоголь в выходные дни (числовое: от 1 – редко до 5 – часто);

29. health – здоровье (числовое: от 1 – слабое до 5 – хорошее);

30. absences – прогулы (числовое: от 0 до 93);

31. G1 – оценка за 1 семестр (числовое: от 0 до 20);

32. G2 – оценка за 2 семестр (числовое: от 0 до 20);

33. G3 – итоговая оценка (числовое: от 0 до 20).

Выпускной балл, который необходимо предсказать, ялвяется признаком с названием G3. Распределение этого признака показано на рисунке 1.1.

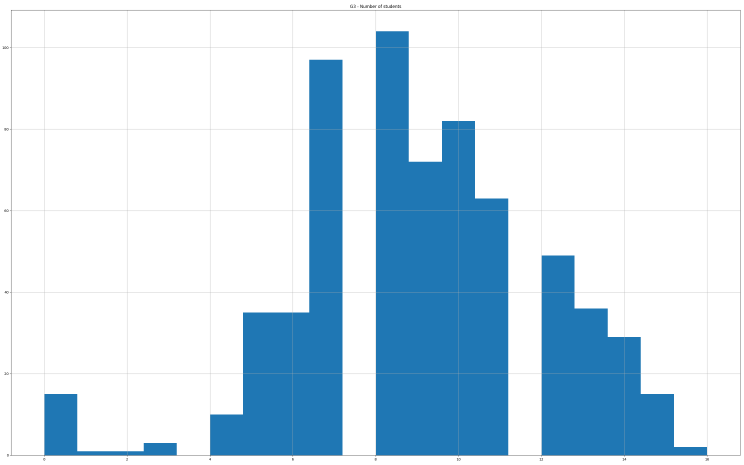

Рис. 1.1: Распределение G3

Из описания признаков предлагается исключить признак school, поскольку он не является показательными с точки зрения образа жизни студентов. Кроме того, все номинальные и бинарные признаки необходимо предобработать, превратив их в числовые.

На рисунке 1.2 приведена матрица корреляции признаков.

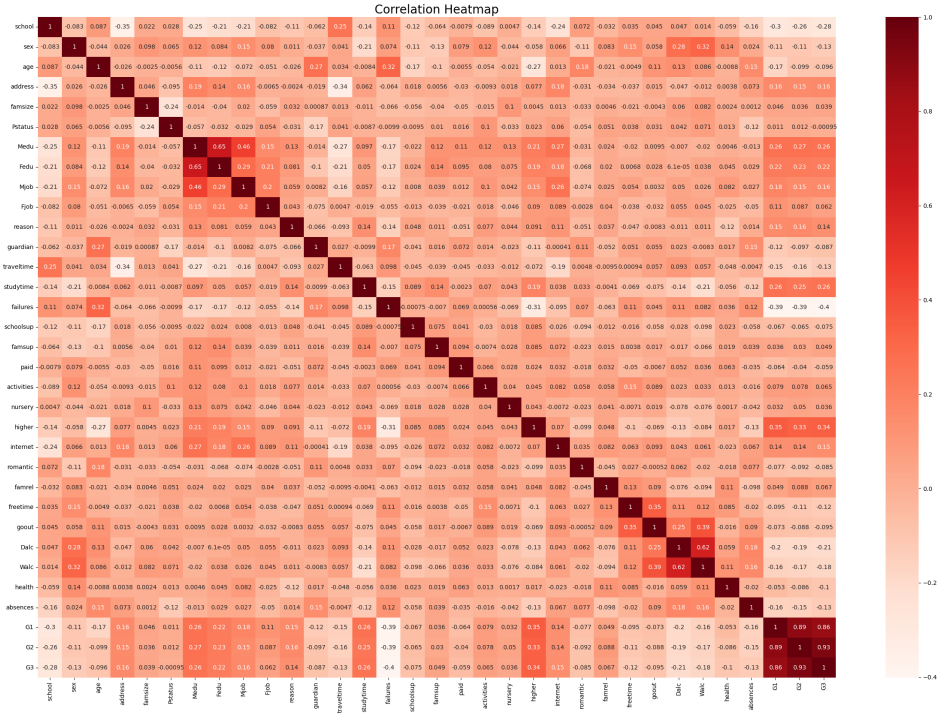

Рис. 1.2: матрица корреляции признаков

Видно, что признаки G1 и G2 сильно коррелируют с G3, что логично, ведь G1 и G2 являются оценками за экзамены предыдущих семестров. Это значит, что любая модель будет принимать во внимание только эти признаки, что нежелательно, поэтому исключим их из признаков перед обучением модели.

1.2 Формализация задачи

Требуется разработать систему предсказания выпускного балла студентов (признак G3) на основе их образа жизни (остальные признаки, которые не были исключены в анализе набора данных X). Математически это может быть описано следующим образом.

Дано обучающее множество:

$$(x^1, y^1), (x^2, y^2), \ldots, (x^m, y^m) \tag{1.1}$$

где:

— $x^i = (x_1^i, x_2^i, \ldots, x_n^i)$ – вектор входных признаков для i-го студента;

— y^i – соответствующее выходное значение (значение G3).

Цель – найти функцию $f(x)$ (модель), которая будет предсказывать выходное значение y по входным признакам x. Обычно эта функция задается параметрическим семейством моделей. Для настройки параметров модели используется метод оптимизации, например, метод наименьших квадратов или градиентный спуск. Цель состоит в том, чтобы минимизировать функцию потерь, которая измеряет разницу между предсказанными и фактическими значениями.

После настройки параметров модели на обучающем наборе данных, можно использовать полученную модель для предсказания выходных значений для новых входных данных.

Формальная постановка задачи совпадает с классической задачей регрессии.

1.3 Анализ существующих моделей регрессии

1.3.1 Линейная модель

Линейная модель для задачи регрессии может быть описана следующей формулой:

$$f(x) = j_0 + j_1 x_1 + j_2 x_2 + \ldots + j_n x_n \tag{1.2}$$

где $j_0, j_1, \ldots, j_n$ - параметры модели, которые необходимо настроить по обучающим данным.

1.3.2 Случайный лес

Случайный лес (Random Forest) – это ансамблевый метод машинного обучения, основанный на построении множества деревьев решений в процессе обучения. Каждое дерево строится независимо и случайным образом, а итоговое предсказание получается путем усреднения предсказаний всех деревьев.

Модель случайного леса для задачи регрессии может быть описана следующим образом:

1. Для построения случайного леса необходимо определить количество деревьев T и размер подвыборки признаков m.

2. Для каждого дерева $t = 1, 2, ..., T$ строится дерево решений на основе случайной подвыборки данных размера m. При построении каждого узла дерева выбирается случайное подмножество признаков размером m, и разбиение узла происходит наилучшим образом по одному из этих признаков.

3. После построения всех деревьев случайного леса, для предсказания нового объекта x происходит усреднение предсказаний всех деревьев:

$$y'(x) = \frac{1}{T} \sum_{t=1}^{T} f_t(x) \tag{1.3}$$

   где $f_t(x)$ - предсказание отдельного дерева.

Таким образом, случайный лес комбинирует предсказания множества деревьев, что позволяет улучшить качество предсказаний и снизить переобучение.

1.3.3 Метод k ближайших соседей

Метод k ближайших соседей (k-Nearest Neighbors, k-NN) также может быть использован для задачи регрессии. Основная идея метода заключается в том, что предсказание для нового объекта делается на основе значений целевой переменной ближайших к нему соседей.

Модель k-NN для задачи регрессии может быть описана следующим образом:

— Основным параметром модели k-NN является число соседей k, которые будут использоваться для предсказания.

— Предсказание для нового объекта x_new вычисляется путем усреднения (или взвешенного усреднения) значений целевой переменной ближайших к нему k соседей.

— Предсказание для нового объекта x_new вычисляется путем усреднения (или взвешенного усреднения) значений целевой переменной ближайших к нему k соседей.

$$y'\_new = \frac{1}{k}\sum_{i=1}^{k} y_i \tag{1.4}$$

где:

– y'_new - предсказанное значение целевой переменной для нового объекта;

– y_i - значение целевой переменной i-го ближайшего соседа объекта x_new,

– k - количество соседей, используемых для предсказания.

— В случае взвешенного усреднения можно использовать веса, зависящие от расстояния между новым объектом и его соседями.

$$y'\_new = \sum_{i=1}^{k} w_i \frac{y_i}{\sum_{i=1}^{k} w_i} \tag{1.5}$$

где:

— w_i - вес, присвоенный i-му соседу на основе расстояния до нового объекта.

Таким образом, модель k-NN для задачи регрессии предсказывает значение целевой переменной для нового объекта на основе значений целе-

вой переменной ближайших к нему соседей, используя усреднение или взвешенное усреднение.

1.3.4 Многослойный персептрон

Обучение нейросети состоит из нескольких эпох (итераций). В течение одной эпохи обучения нейронной сети происходит несколько этапов, которые повторяются для каждого обучающего примера:

1. прямое распространение:
   - входные данные подаются на входной слой нейронов;
   - данные передаются через скрытые слои, взвешиваются с использованием соответствующих весов и агрегируются;
   - агрегированные значения проходят через функции активации каждого нейрона в скрытых слоях и выходном слое, что приводит к формированию выходов сети;
2. оценка ошибки : вычисляется ошибка между выходами сети и ожидаемыми значениями (целевыми метками/метками классов);
3. обратное распространение ошибки:
   - ошибка распространяется обратно через сеть, начиная с последнего слоя и двигаясь к входному слою;
   - для каждого слоя вычисляется градиент функции потерь по весам и смещениям сети;
4. обновление весов, чтобы уменьшить ошибку модели: используя градиент ошибки, веса сети обновляются с использованием метода оптимизации, такого как стохастический градиентный спуск или его модификации;

Эти этапы повторяются для каждой эпохи обучения с тем, чтобы постепенно корректировать веса сети и уменьшать ошибку прогноза. Процесс обучения заключается в том, чтобы минимизировать ошибку модели и достичь желаемой производительности в решении конкретной задачи.

1.3.5 Градиентный бустинг

Метод градиентного бустинга (Gradient Boosting) является одним из популярных ансамблевых методов машинного обучения, который может быть использован для задачи регрессии. Основная идея метода заключается в построении ансамбля слабых моделей (например, деревьев решений), которые последовательно обучаются на остатках предыдущих моделей.

Функция модели градиентного бустинга для задачи регрессии может быть описана следующим образом:

— Градиентный бустинг строит ансамбль моделей $F(x) = \sum_{m=1}^{M} f_m(x)$, где каждая следующая модель $f_m(x)$ обучается на результате предыдущих моделей.

— Для построения новой модели $f_m(x)$, минимизируется функция потерь, которая определяет разницу между предсказанными значениями и реальными значениями целевой переменной.

— На каждом шаге градиентного бустинга строится новая модель, которая приближает антиградиент функции потерь.

$$F(x) = \sum_{m=1}^{M} f_m(x) \tag{1.6}$$

где:

– $F(x)$ – предсказанное значение целевой переменной для объекта x;

– M – количество моделей в ансамбле,

– $f_m(x)$ – m-ая модель в ансамбле.

Таким образом, функция модели градиентного бустинга для задачи регрессии представляет собой сумму прогнозов всех моделей в ансамбле, которые последовательно улучшают предсказания путем минимизации функции потерь на каждом шаге.

1.4 Выбор модели регрессии

Для сравнения методов регрессии по 6 критериям (точность, интерпретируемость, скорость обучения, устойчивость к выбросам, способность к работе с большими объемами данных, необходимость настройки гиперпараметров) и выбора наилучшего метода, составим таблицу 1.1.

Таблица 1.1: Сравнительная таблица моделей

Метод регрессии	Точность	Интерпретируемость	Скорость обучения	Устойчивость к выбросам	Способность к работе с большими данными	Необходимость настройки гиперпараметров
Линейная регрессия	Средняя	Высокая	Высокая	Низкая	Высокая	Низкая
Случайный лес	Высокая	Низкая	Средняя	Высокая	Средняя	Средняя
Метод k соседей	Средняя	Низкая	Низкая	Низкая	Средняя	Высокая
Многослойный персептрон	Высокая	Низкая	Высокая	Низкая	Высокая	Высокая
Градиентный бустинг	Очень высокая	Низкая	Средняя	Высокая	Очень высокая	Высокая

По результатам сравнения по вышеуказанным критериям видно, что градиентный бустинг имеет высокую точность предсказаний, хорошую устойчивость к выбросам, способность работать с большими объемами данных и среднюю скорость обучения. В то же время он требует настройки гиперпараметров, что может потребовать дополнительных усилий при использовании.

Таким образом, градиентный бустинг является наилучшим методом регрессии в данном сравнении.

Бустинг, использующий деревья решений в качестве базовых алгоритмов, (GBDT) отлично работает на выборках с «табличными», неоднородными данными, что характеризует данные нашей задачи (при этом на однородных данных: текстах, изображениях, звуке лучше работают нейросетевые подходы). Такой бустинг способен эффективно находить нелинейные зависимости в данных различной природы. Этим свойством обладают все алгоритмы, использующие деревья решений, однако именно GBDT выигрывает в большинстве практических задач.

1.5 Выбор функционала качества модели

1. MSE (Mean Squared Error) вычисляется как среднее значение квадратов разностей между предсказанными значениями и истинными значениями:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2 \tag{1.7}$$

где y_i – истинное значение, y_i' – предсказанное значение, n – количество наблюдений;

2. RMSE (Root Mean Squared Error) представляет собой квадратный корень из MSE и показывает среднеквадратичное отклонение предсказанных значений от истинных:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{1.8}$$

3. MAPE (Mean Absolute Percentage Error) выражает среднее абсолютное процентное отклонение предсказанных значений от истинных:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - y_i'}{y_i}| \cdot 100\% \tag{1.9}$$

4. R2 (коэффициент детерминации) вычисляется по формуле:

$$R^2 = 1 - \frac{SS\_res}{SS\_tot}, \tag{1.10}$$

где SS_res – сумма квадратов остатков, SS_tot – общая сумма квадратов.

В таблице 1.2 приведено сравнение функционалов качества моделей.

Таблица 1.2: Сравнение функционалов качества моделей

Критерий	Интерпретация	Чувствительность к выбросам	Удобство использования
MSE	Среднеквадратичное отклонение между фактическими и прогнозными значениями	Чувствителен	Среднее
RMSE	Квадратный корень из MSE, возвращает значения в тех же единицах, что и исходные данные	Чувствителен	Высокое
MAPE	Средняя абсолютная ошибка в процентах	Не чувствителен	Высокое
R2	Отражает долю объясненной дисперсии в данных	Чувствителен	Низкое

Из представленных критериев качества модели регрессии наилучшим выбором является MAPE, поскольку он имеет простую интерпретацию и не чувствителен к выбросам.

Вывод

В данном разделе был описан и визуализирован набор данных, формализована задача, проведён анализ существующих моделей регрессии и осуществлён выбор наиболее подходящей для решения поставленной задачи – градиентный бустинг, а также выбран функционал качества модели – MAPE.

2 Конструкторский раздел

В данном разделе будет описан алгоритм предобработки данных и алгоритм работы ПО, осуществляющего решение задачи регрессии.

2.1 Алгоритм предобработки данных

1. Инициализируется пустой словарь, который будет использоваться для хранения соответствия между уникальными значениями и их кодами.
2. Поступает массив значений, который нужно закодировать.
3. Сначала необходимо пройти по всем значениям во входном массиве и вычислить уникальные значения.
4. Далее каждому уникальному значению присваивается уникальный код, начиная с 0 и увеличиваясь на 1 для каждого нового уникального значения.
5. Полученные соответствия между уникальными значениями и их кодами сохраняются в словаре.
6. Затем каждый элемента во входном массиве значений заменяется на соответствующий ему код из словаря.
7. В итоге возвращается массив закодированных числовых меток, где каждое значение заменено на соответствующий ему код.
8. Этот закодированный массив может быть использован для обучения моделей машинного обучения, которые требуют числовых данных в качестве входа.

Алгоритм предобработки данных приведён на рисунке 2.1.

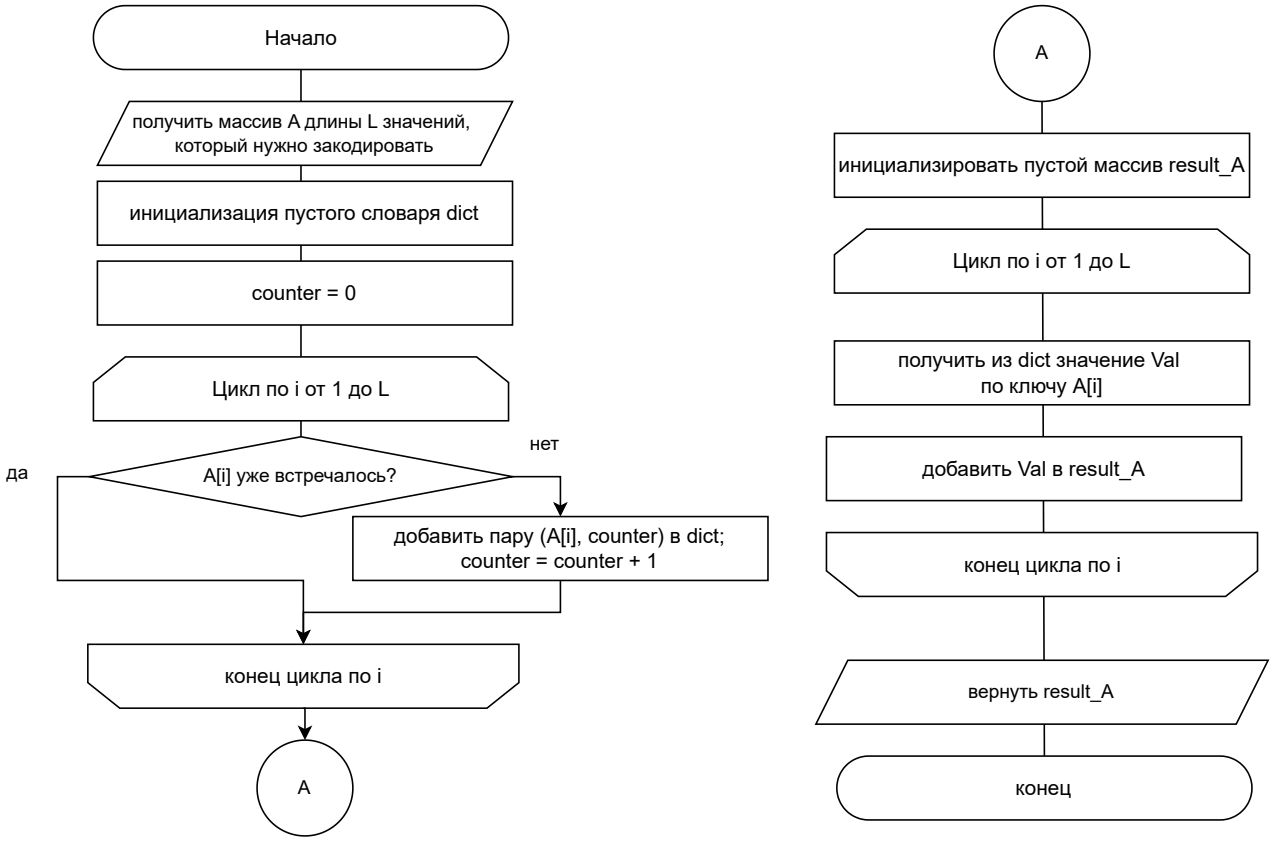

Рис. 2.1: Алгоритм предобработки данных

2.2 Алгоритм работы ПО

Основной алгоритм работы ПО состоит из 3 основных шагов:

— подготовки данных – преобразование всех нечисловых данные в числовой вид с помощью алгоритма, описанного в предыдущем пункте;

— обучение модели градиентного бустинга на обучающей выборке, полученной из исходного набора данных;

— вычисление значения метрики MAPE на тестовой выборке, полученной из исходного набора данных так, что она не пересекается с обучающей выборкой.

Алгоритм работы ПО приведён на рисунке 2.2.

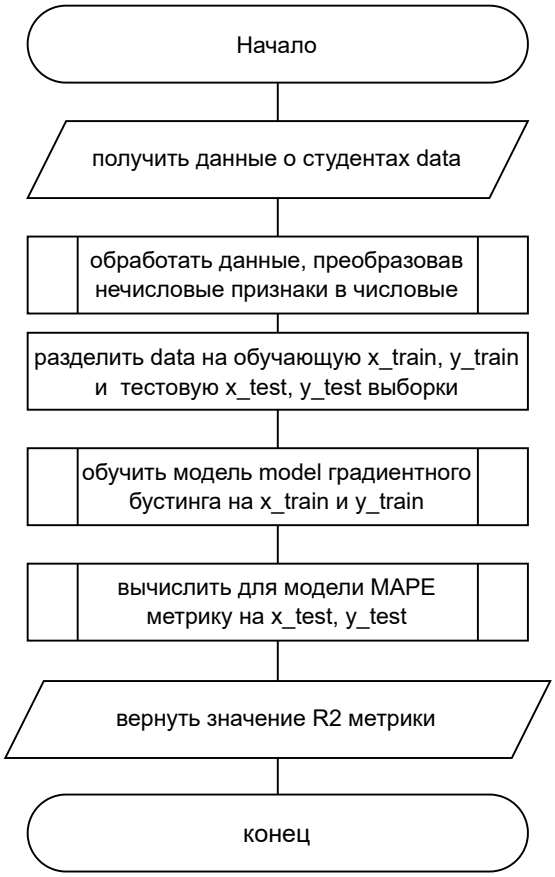

Рис. 2.2: Алгоритм работы ПО

Вывод

В данном разделе был описан алгоритм предобработки данных и алгоритм работы ПО, осуществляющего решение задачи регрессии.

3 Технологический раздел

В данном разделе выбраны средства разработки программного обеспечения, показаны детали реализации, а также выбраны гиперпараметры модели, с которыми модель работает наилучшим образом с точки зрения выбранного функционала качества.

3.1 Средства реализации ПО

В качестве языка программирования был использован язык Python [2], поскольку этот язык кроссплатформенный и для него разработано огромное количество библиотек и модулей, решающих разнообразные задачи.

В частности, имеется библиотека, включающая в себя алгоритм градиентного бустинга, а также вычисление метрики MAPE в библиотеке sklearn [3].

Для работы с табличными данными была выбрана библиотека pandas [4], так как она имеет мощные функции для обработки данных, интеграцию с другими библиотеками, а также поддержку чтения и записи различных форматов данных.

Для создания графиков были выбраны библиотеки matplotlib [5] и seaborn [6], доступные на языке Python, так как они предоставляет удобный интерфейс для работы с данными и их визуализации.

3.2 Листинг

В листинге 3.1 представлен код, решающий задачу регрессии.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import  GradientBoostingRegressor

data = pd.read_csv('students.csv')
le = LabelEncoder()
for col in data.columns:
    data[col] = le.fit_transform(data[col])
```

```
X = data.drop(['G3', 'G2', 'G1', 'school'], axis=1)
y = data['G3']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
    random_state=42)
model = GradientBoostingRegressor(learning_rate = 0.01, max_depth=9,
    n_estimators = 500, subsample = 0.5)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
s = 0
cnt = 0
y_test = y_test.values
y_pred = y_pred
for i in range(len(y_test)):
    if y_test[i] > 0.1:
        s += abs(y_test[i] - y_pred[i]) / y_test[i]
        cnt += 1
print(s / cnt * 100)
```

Листинг 3.1: Код для решения задачи регрессии

3.3 Выбор гиперпараметров модели

Опишем параметры модели GradientBoostingRegressor из библиотеки sklearn [3]:

1. learning_rate:
   - Определяет вклад каждого дерева в общий результат. Чем ниже learning rate, тем меньше влияние каждого дерева.
   - Диапазон: [0, +inf].
   - Более низкий learning rate требует большего количества деревьев для достижения хорошей точности, но может уменьшить переобучение.
2. n_estimators:
   - Определяет количество деревьев (или итераций), которые будут добавлены к модели.
   - Диапазон: любое положительное целое число.

— Большее количество деревьев может улучшить точность модели, но также увеличивает время обучения.

3. subsample

— Определяет долю обучающих данных, используемых для построения каждого дерева.

— Диапазон: (0, 1].

— Уменьшение значения subsample может уменьшить переобучение, но снизит скорость обучения.

4. max_depth

— Определяет максимальную глубину каждого дерева.

— Диапазон: любое положительное целое число.

— Большая глубина деревьев может привести к переобучению, но также может улучшить точность модели.

Параметры модели GradientBoostingRegressor позволяют настраивать модель для достижения оптимального баланса между точностью и предотвращением переобучения.

Проварьируем эти параметры с использованием следующих значений:

— learning_rate: [0.0001, 0.01, 0.1];

— n_estimators: [10, 500, 1000];

— subsample: [0.05, 0.5, 0.8];

— max_depth: [5, 9, 12].

В результате, получаем таблицы 3.1-3.2.

Таблица 3.1: Сравнение ошибки при разных гиперпараметрах (часть 1)

learning_rate	n_estimators	subsample	max_depth	MAPE
0.05	10	5	0.0001	24.2
0.05	10	5	0.0100	24.0
0.05	10	5	0.1000	22.7
0.05	10	9	0.0001	24.2
0.05	10	9	0.0100	23.7
0.05	10	9	0.1000	22.8
0.05	10	12	0.0001	24.2
0.05	10	12	0.0100	23.8
0.05	10	12	0.1000	22.9
0.05	500	5	0.0001	23.9
0.05	500	5	0.0100	21.1
0.05	500	5	0.1000	33.9
0.05	500	9	0.0001	23.9
0.05	500	9	0.0100	20.9
0.05	500	9	0.1000	32.9
0.05	500	12	0.0001	23.9
0.05	500	12	0.0100	21.6
0.05	500	12	0.1000	36.3
0.05	1000	5	0.0001	23.7
0.05	1000	5	0.0100	22.1
0.05	1000	5	0.1000	45.6
0.05	1000	9	0.0001	23.6
0.05	1000	9	0.0100	21.9
0.05	1000	9	0.1000	39.4
0.05	1000	12	0.0001	23.6
0.05	1000	12	0.0100	22.8
0.05	1000	12	0.1000	44.7
0.50	10	5	0.0001	24.2
0.50	10	5	0.0100	23.4
0.50	10	5	0.1000	21.0
0.50	10	9	0.0001	24.1
0.50	10	9	0.0100	23.2
0.50	10	9	0.1000	21.2

Таблица 3.2: Сравнение точности при разных гиперпараметрах (часть 2)

learning_rate	n_estimators	subsample	max_depth	MAPE
0.50	10	12	0.0001	24.1
0.50	10	12	0.0100	23.3
0.50	10	12	0.1000	22.3
0.50	500	5	0.0001	23.7
0.50	500	5	0.0100	21.2
0.50	500	5	0.1000	24.4
0.50	500	9	0.0001	23.7
0.50	500	9	0.0100	20.9
0.50	500	9	0.1000	21.4
0.50	500	12	0.0001	23.7
0.50	500	12	0.0100	21.0
0.50	500	12	0.1000	23.2
0.50	1000	5	0.0001	23.4
0.50	1000	5	0.0100	22.0
0.50	1000	5	0.1000	22.6
0.50	1000	9	0.0001	23.3
0.50	1000	9	0.0100	21.2
0.50	1000	9	0.1000	22.9
0.50	1000	12	0.0001	23.3
0.50	1000	12	0.0100	21.2
0.50	1000	12	0.1000	23.2
0.80	10	5	0.0001	24.2
0.80	10	5	0.0100	23.3
0.80	10	5	0.1000	20.9
0.80	10	9	0.0001	24.2
0.80	10	9	0.0100	23.3
0.80	10	9	0.1000	21.5
0.80	10	12	0.0001	24.2
0.80	10	12	0.0100	23.4
0.80	10	12	0.1000	21.7
0.80	500	5	0.0001	23.7
0.80	500	5	0.0100	20.8
0.80	500	5	0.1000	22.5
0.80	500	9	0.0001	23.7
0.80	500	9	0.0100	20.8
0.80	500	9	0.1000	21.7
0.80	500	12	0.0001	23.7
0.80	500	12	0.0100	21.2
0.80	500	12	0.1000	22.5
0.80	1000	5	0.0001	23.4
0.80	1000	5	0.0100	21.7
0.80	1000	5	0.1000	22.6
0.80	1000	9	0.0001	23.4
0.80	1000	9	0.0100	21.2
0.80	1000	9	0.1000	21.4
0.80	1000	12	0.0001	23.4
0.80	1000	12	0.0100	21.2
0.80	1000	12	0.1000	22.4

Видим, что наименьшая ошибка модели с точки зрения выбранного функционала качества достигается при следующих параметрах модели:

— learning_rate: 0.01;

— n_estimators: 500;

— subsample: 0.80;

— max_depth: 9.

На рисунке 3.1 построена гистограмма важности признаков построенной модели (важность определяется как нормализованное общее снижение энтропии с помощью этого признака – индекс Джини). Видно, что наиболее важными признаками для опредлеления выпускного балла являются количество неудач на контрольных и количество прогулов, что логично.

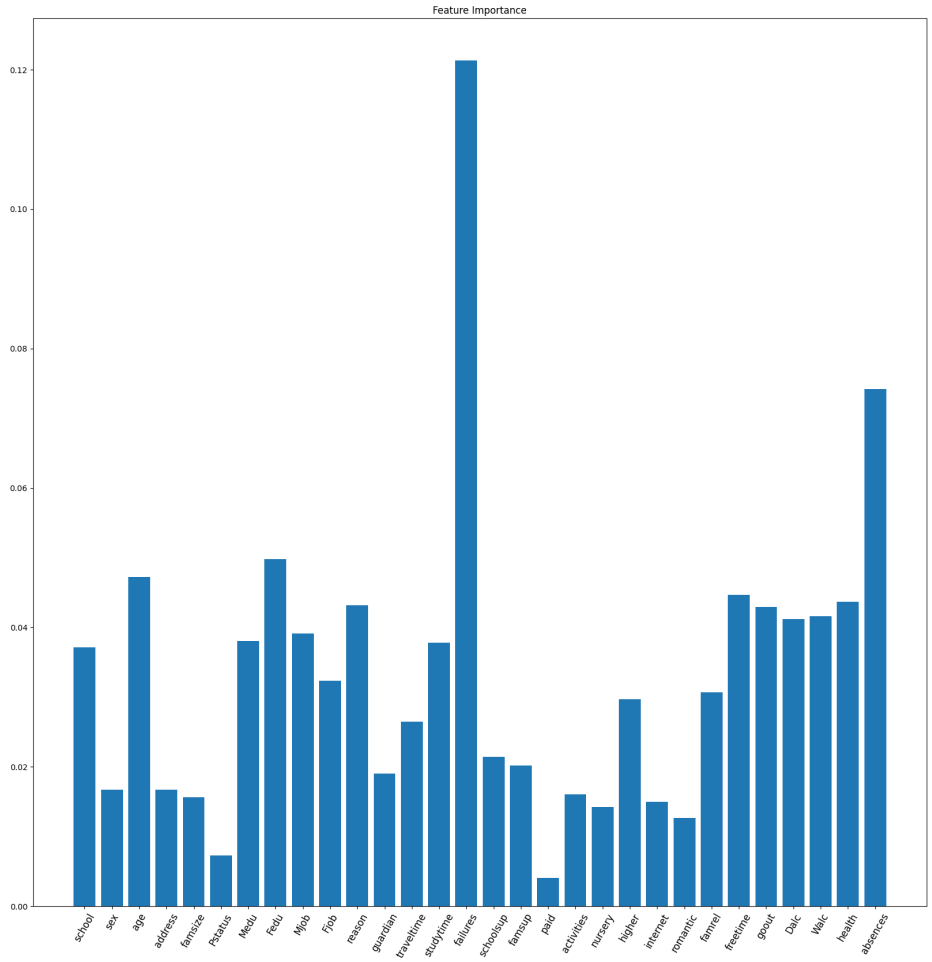

Рис. 3.1: Важность признаков

Итоговая ошибка модели на тестовой выборке составляет 20.8%.

3.4 Обобщающая способность

Оценим обобщающую способность модели. Для этого будем менять каждый из гиперпараметров так, что остальные остаются равными выбранным с помощью функционала качества на тестовой выборке в предыдщем пункте.

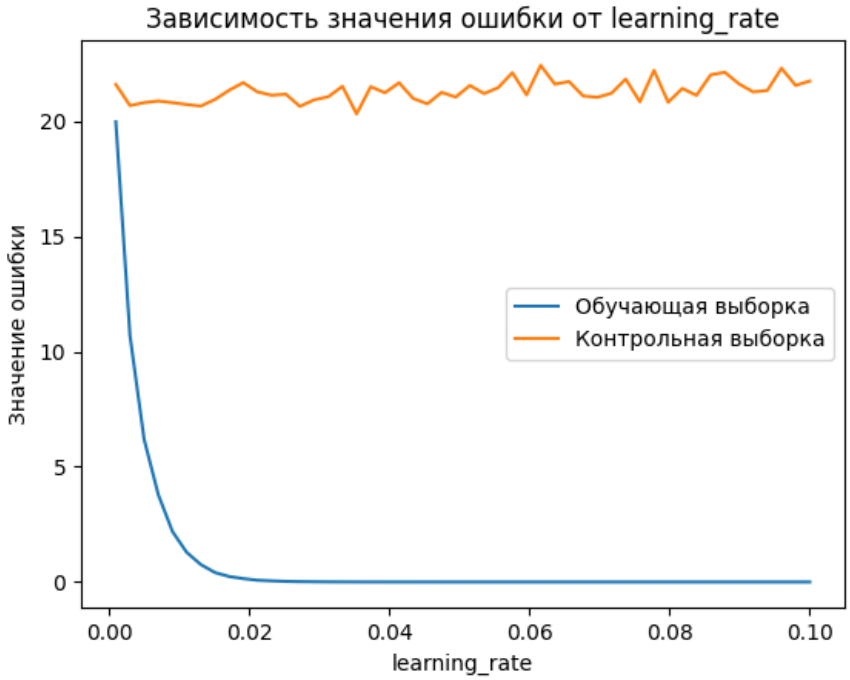

Рис. 3.2: Зависимость ошибки от learning_rate

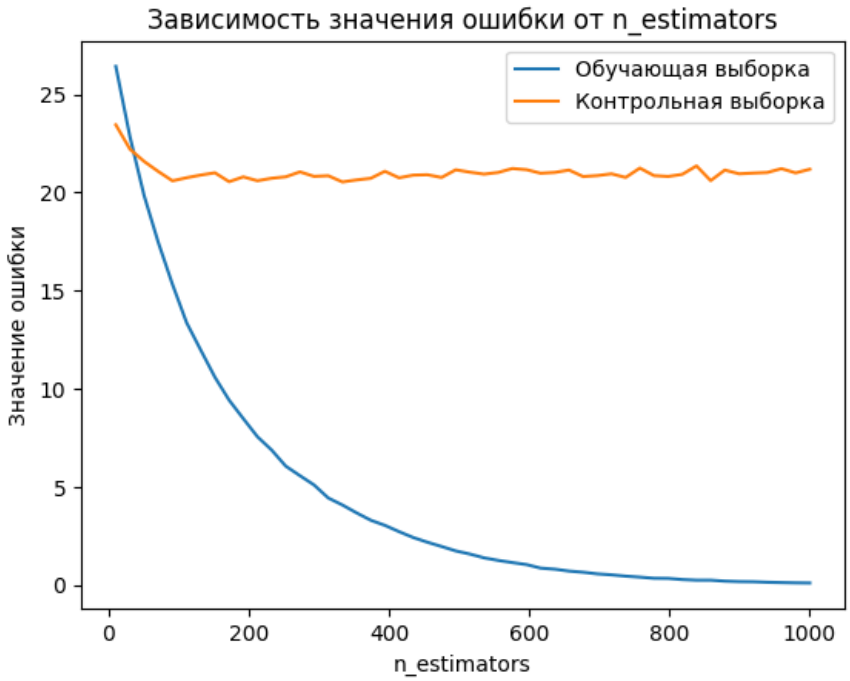

Рис. 3.3: Зависимость ошибки от n_estimators

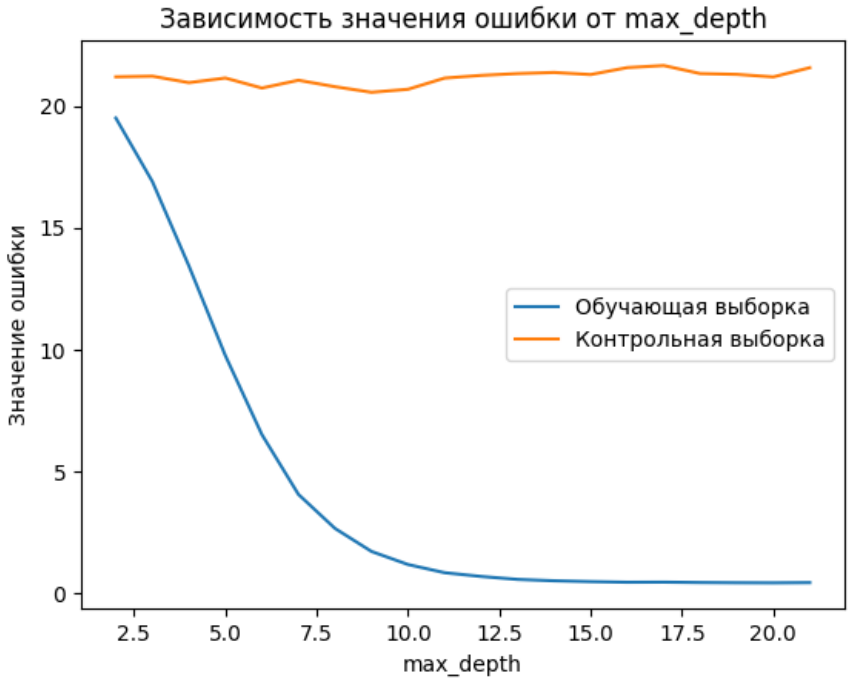

Рис. 3.4: Зависимость ошибки от max_depth

Видим, что обобщающую способность модели при выбранных парметрах можно считать удовлетворительной, поскольку при минимуме ошибки на обучающей выборке достигается минимум ошибки на тестовой (считаем, что при увеличении ошибки на тестовой выборке при минимуме на обучающей модель

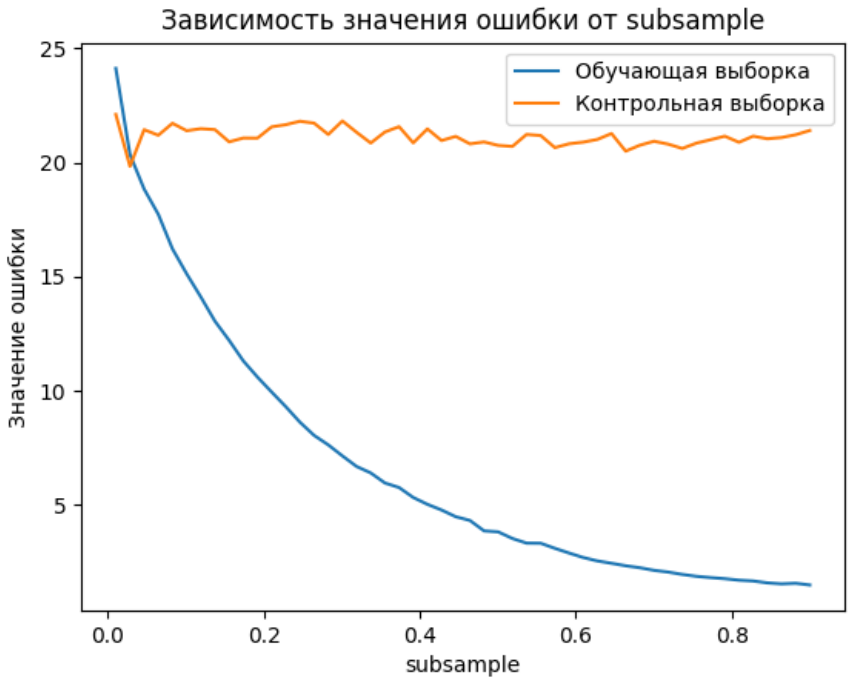

Рис. 3.5: Зависимость ошибки от subsample

начинает переобучаться).

Вывод

В данном разделе были выбраны средства разработки программного обеспечения, показаны детали реализации, а также выбраны гиперпараметры модели, решающей задачу регрессии.

4 Исследовательский раздел

В данном разделе проводится исследование значения функционала качества на разных моделях. Все исследования проводились на данных об успеваемости студентов.

4.1 Исследование функционала качества разных моделей

Для исследования были выбраны следующие модели:

— линейная модель;

— случайный лес с параметрами (100 деревьев, без обрезки);

— метод k ближайших соседей (k = 5);

— модель градиентного бустинга с параметрами, выбранными в предыдущем разделе (learning_rate: 0.01; n_estimators: 500; subsample: 0.80; max_depth: 9).

— многослойный персептрон с 2 слоями: 29-64-20 нейронов.

Результаты можно видеть в таблице 4.1.

Таблица 4.1: Результаты исследоавания

Модель	MAPE
линейная модель	22.457
k-соседей	26.374
случайный лес	21.381
многослойный персептрон	22.748
градиентный бустинг	20.958

Видно, что модель градиентного бустинга работает точнее остальных с точки зрения функционала качества.

Вывод

В данном разделе было проведено исследование значения функционала качества на разных моделях. В результате исследования было выяснено, что модель градиентного бустинга работает точнее остальных с точки зрения функционала качества.

Заключение

В результате работы была решена задача регрессии, состоящая в определении выпускного балла студентов с помощью информации об их образе жизни, на данных о студентах двух португальских школ, взятых с kaggle [1]. Таким образом, цель работы была достигнута.

Для достижения поставленной цели были решены следующие задачи:

— описан и визуализирован набор данных;

— проанализированы существующие модели регрессии для решения задачи и выбрана наиболее подходящая, а также выбран функционал качества модели;

— описан алгоритм предобработки данных;

— описан общий алгоритм работы ПО, осуществляющего решение задачи регрессии;

— разработано ПО, решающее задачу регрессии;

— выбраны гиперпараметры модели, с которыми модель работает наилучшим образом с точки зрения функционала качества, оценена обобщающую способность;

— проведено исследование ПО с целью сравнения полученной модели с другими моделями с точки зрения функционала качества.

В ходе выполнения экспериментально-исследовательской части было установлено, что модель градиентного бустинга работает точнее остальных с точки зрения функционала качества.

Список использованных источников

1. Набор данных об успеваемости и образе жизни студентов двух португальских школ [Электронный ресурс]. — (дата обращения: 14.04.2024). Режим доступа: URL: https://www.kaggle.com/datasets/larsen0966/student-performance-data-set/data.
2. *Python Core Team.* Python: язык программирования / Python Software Foundation. — 2019. — URL: `https://www.python.org/`.
3. Scikit-learn: Machine learning in Python / F. Pedregosa [и др.]. — 2011.
4. *McKinney W.* Data Structures for Statistical Computing in Python // Proceedings of the 9th Python in Science Conference. — 2010. — С. 51—56.
5. Библиотека визуализации данных matplotlib [Электронный ресурс]. — Режим доступа: URL: https://matplotlib.org (дата обращения: 13.04.2024).
6. *Waskom M. L.* seaborn: statistical data visualization // Journal of Open Source Software. — 2021. — Т. 6, № 60. — С. 3021. — URL: `https://doi.org/10.21105/joss.03021`.