ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»
Институт компьютерных наук и кибербезопасности

Отчет о прохождении учебной (научно-исследовательская работа (получение первичных навыков научно-исследовательской работы)) практики

Торновской Софьи-Александры Владимировны

(Ф.И.О. обучающегося)

1 курс, 5130203/40001

(номер курса обучения и учебной группы)

02.03.03 Математическое обеспечение и администрирование информационных систем

(направление подготовки (код и наименование))

Место прохождения практики: ФГАОУ ВО «СПбПУ», ИКНиК, ВШТИИ,

(указывается наименование профильной организации или наименование структурного подразделения

г. Санкт-Петербург, ул. Обручевых, д. 1, лит. В

ФГАОУ ВО «СПбПУ», фактический адрес)

Сроки практики: с 20.06.2025 по 17.07.2025

Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»: Пак Вадим Геннадьевич, к.ф.-м.н., доцент ВШТИИ

(*Ф.И.О., уч. степень, должность)*

Консультант практической подготовки от ФГАОУ ВО «СПбПУ»: нет

(*Ф.И.О., уч. степень, должность)*

Руководитель практической подготовки от профильной организации: нет

Оценка:

Руководитель практики
от ФГАОУ ВО «СПбПУ»: /Пак В.Г./

Обучающийся: /Торновская С.В./

Дата: 17.07.25

ВВЕДЕНИЕ

Машинное обучение для табличных данных — это ключевой инструмент в анализе структурированных данных, который используется для построения моделей прогнозирования и принятия решений в различных областях, таких как медицина, финансы, маркетинг и другие. Регрессионные модели являются основой большинства задач машинного обучения, так как они позволяют находить зависимости между переменными и строить прогнозы на основе этих зависимостей.

Одними из самых популярных методов являются линейная и логистическая регрессия. Линейная регрессия используется для предсказания непрерывных значений, тогда как логистическая регрессия применяется в задачах классификации, где необходимо предсказать вероятность принадлежности объекта к одному из классов. Однако обе модели имеют свои ограничения, например, линейная регрессия плохо справляется с нелинейными зависимостями, а логистическая регрессия требует соблюдения предположений о распределении данных.

Цель данного исследования – сравнение линейной и логистической регрессии на простых наборах данных. В процессе работы будет проведен анализ их точности, скорости обучения, а также интерпретируемости моделей. Также будут рассмотрены метрики качества моделей, такие как точность, AUC-ROC, R^2 для линейной регрессии и precision, recall, F1-score для логистической регрессии.

Актуальность работы объясняется необходимостью эффективного применения регрессионных моделей для решения различных практических задач, таких как предсказание цен на товары, анализ качества продукции, диагностика заболеваний и многое другое. Сравнительный анализ этих моделей поможет определить, какая из них лучше всего подходит для конкретных типов данных и задач.

Объект исследования — регрессионные модели машинного обучения, а именно линейная регрессия и логистическая регрессия.

Предмет исследования — сравнительный анализ эффективности этих методов в задаче классификации и регрессии на простых наборах данных по точности, скорости обучения и интерпретируемости.

Цель исследования — провести сравнительный анализ линейной и логистической регрессии на примере простых наборов данных и оценить их эффективность в различных аспектах.

Задачи исследования:

A. Изучение математических основ линейной и логистической регрессии.
B. Сравнение моделей по точности и скорости обучения на простых наборах данных.
C. Оценка интерпретируемости моделей.
D. Анализ результатов и выводы о применимости каждой модели для различных типов задач.

1. ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

1.1. Введение в проблематику

Регрессионные модели являются основой анализа данных, позволяя находить зависимости между переменными и делать прогнозы. Они широко применяются в различных областях: в экономике для прогнозирования финансовых показателей, в здравоохранении для диагностики заболеваний, в маркетинге для предсказания поведения потребителей.

Одной из задач, с которой сталкиваются исследователи и практики, является правильный выбор модели для решения той или иной задачи. Для задач с числовыми предсказаниями часто используются линейная регрессия и логистическая регрессия, которые являются классическими моделями машинного обучения. Линейная регрессия применяется для предсказания непрерывных значений, а логистическая регрессия используется для задач классификации, где важно определить принадлежность объекта к одному из классов.

Однако несмотря на широкое применение этих моделей, существует ряд ограничений, связанных с их использованием. Например, линейная регрессия предполагает линейную зависимость между переменными, что не всегда соответствует реальности, особенно в задачах с сложными нелинейными зависимостями. Логистическая регрессия, в свою очередь, эффективна при бинарной классификации, но она также может иметь трудности при работе с сильно несбалансированными данными или в случае многоклассовых задач.

В связи с этим важно проводить тщательное сравнение этих моделей, чтобы выбрать наиболее подходящую для конкретной задачи, а также оценить их эффективность с точки зрения точности, скорости обучения и интерпретируемости.

1.2. Введение в регрессионные модели

Регрессия — это статистический метод, используемый для анализа зависимости между переменными и построения модели, которая может предсказать значения зависимой переменной на основе известных независимых переменных. В машинном обучении существует несколько типов регрессионных моделей, среди которых выделяются линейная регрессия и логистическая регрессия.

Линейная регрессия используется для моделирования отношений между независимыми переменными и зависимой переменной, когда эта зависимая переменная является непрерывной. Она основывается на линейной зависимости, что делает ее простым и понятным инструментом для прогнозирования числовых значений.

Логистическая регрессия применяется в задачах классификации, когда результатом является категория (например, "да" или "нет", "болен" или "не болен"). Несмотря на название, это не регрессия в традиционном понимании, а классификационный метод, который использует логистическую функцию для оценки вероятностей.

1.3. Линейная регрессия: математическая модель и алгоритм обучения

Линейная регрессия — это метод, используемый для предсказания числовых значений на основе линейных зависимостей между независимыми переменными (признаками) и зависимой переменной (целевой переменной). Этот метод находит широкое применение в различных областях, таких как экономика, медицина, инженерия, где важно предсказать количественные параметры.

Математическая модель линейной регрессии предполагает, что зависимость между целевой переменной y и признаками $x_1, x_2, \dots, x_n$ можно выразить через линейную функцию:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

где:

y — зависимая переменная (целевой признак),

$x_1, x_2, \ldots, x_n$— независимые переменные (признаки),

$w_0, w_1, \ldots, w_n$ — коэффициенты модели, которые обучаются на основе обучающих данных.

Процесс обучения линейной регрессии заключается в нахождении таких коэффициентов $w_0, w_1, \ldots, w_n$ которые минимизируют ошибку предсказания, обычно измеряемую с помощью среднеквадратичной ошибки (MSE):

$$MSE = \frac{1}{Ni} \sum_{i=1}^{N} (yi - \hat{y}i)^2$$

где:

yi — истинные значения,

$\hat{y}i$— предсказания модели,

Модели могут обучаться с использованием различных методов оптимизации, включая метод наименьших квадратов или градиентный спуск. Процесс продолжается до тех пор, пока ошибка не станет минимальной (или не будет достигнут определенный порог) или пока не будет выполнено заданное количество итераций.

1.4 Логистическая регрессия: математическая модель и алгоритм обучения

Логистическая регрессия — это метод классификации, используемый для предсказания вероятности принадлежности объекта к одному из классов. Математически она моделирует зависимость вероятности ppр принадлежности к классу 1 следующим образом:

Математическая модель логистической регрессии основывается на логистической функции, которая преобразует линейную комбинацию признаков в значение вероятности:

$$p = \frac{1}{1 + e^{-(w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n)}}$$

где:

p — вероятность того, что объект принадлежит классу 1,

$x_0, x_1, \ldots, x_n$ — признаки,

$w_0, w_1, \ldots, w_n$ — коэффициенты модели.

Модель обучается с использованием **логистической функции потерь**, которая минимизирует разницу между предсказанными вероятностями и реальными метками классов.

Для логистической регрессии алгоритм обучения схож с линейной регрессией, но используется логистическая функция потерь:

$$Loss = -\frac{1}{N} \sum_{i=1}^{N} [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)]$$

где:

y_i — истинная метка класса (0 или 1),

p_i — предсказанная вероятность принадлежности к классу 1.

Обучение модели логистической регрессии сводится к минимизации этой функции потерь с использованием методов оптимизации, таких как градиентный спуск.

1.5. Отличия между линейной и логистической регрессией

Линейная и логистическая регрессия — это два различных метода, которые используются для решения разных типов задач машинного обучения. Несмотря на схожесть в названии, эти методы имеют принципиальные различия как в математической основе, так и в области применения.

1. Тип задачи:

Линейная регрессия применяется для решения задач регрессии, где целью является предсказание непрерывной переменной. Примером может служить задача прогнозирования стоимости недвижимости на основе различных признаков, таких как площадь, расположение, возраст дома и т. д.

Логистическая регрессия используется для решения задач классификации, где целью является предсказание категориальных значений (например, классов 0 или 1). Типичная задача — классификация на основе признаков, например, определение вероятности того, что пациент болен определенным заболеванием.

2. Выходные значения:

Линейная регрессия предсказывает непрерывные значения. Например, если мы прогнозируем цену недвижимости, результатом будет число, представляющее собой стоимость объекта.

Логистическая регрессия предсказывает вероятности (от 0 до 1), которые интерпретируются как вероятность принадлежности к одному из классов. После этого можно использовать порог (например, 0.5), чтобы классифицировать объект как принадлежавший к классу 0 или 1.

3. Функция потерь:

Для линейной регрессии используется среднеквадратичная ошибка (MSE), которая минимизируется во время обучения. Это функция потерь для задачи регрессии, которая измеряет среднее квадратное отклонение предсказанных значений от истинных.

Для логистической регрессии используется логистическая функция потерь, которая минимизирует отклонение предсказанных вероятностей от истинных меток классов.

4. Применение:

Линейная регрессия применяется для задач, где нужно предсказать количество или величину. Например, в экономике это может быть прогнозирование доходов, в здравоохранении — оценка риска на основе показателей здоровья.

Логистическая регрессия используется для задач, где необходимо классифицировать объекты в несколько категорий. Например, классификация изображений, диагностика заболеваний или анализ поведения клиентов.

5. Интерпретируемость:

Линейная регрессия легко интерпретируется, поскольку модель предполагает линейную зависимость между признаками и целевой переменной. Коэффициенты модели показывают, как изменение каждого признака влияет на предсказание.

Логистическая регрессия также имеет высокую интерпретируемость, однако, в отличие от линейной, результаты обычно интерпретируются через вероятности. Это означает, что для логистической регрессии можно легко объяснить, как каждый признак влияет на вероятность принадлежности объекта к одному из классов.

6. Ограничения:

Линейная регрессия плохо справляется с задачами, где данные имеют непостоянные или нелинейные зависимости, а также с несбалансированными классами, если задача классификации.

Логистическая регрессия ограничена в своих возможностях для задач с несколькими классами или сложными зависимостями, и требует дополнительных методов, таких как многоклассовая логистическая регрессия или регуляризация.

Линейная и логистическая регрессия имеют схожие концепции, но различаются в задачах применения и математических моделях. Линейная регрессия подходит для задач с непрерывной целевой переменной, в то время как логистическая регрессия используется для классификации объектов. Понимание этих различий поможет выбрать наиболее подходящий метод в зависимости от типа задачи и особенностей данных.

2. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МОДЕЛЕЙ ЛИНЕЙНОЙ И ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

2.1. Подготовка и выбор простых наборов данных для экспериментов

Для проведения экспериментов с моделями линейной регрессии и логистической регрессии был выбран набор данных Heart Disease, который широко используется в задачах машинного обучения. Этот набор данных содержит информацию о различных признаках, которые могут влиять на развитие сердечных заболеваний, такие как возраст пациента, уровень холестерина, артериальное давление и другие.

Целевая переменная в наборе данных — это "num", которая представляет собой категориальную переменную. В оригинальном наборе данных целевая переменная может принимать значения от 0 до 4, что указывает на различные степени выраженности сердечного заболевания. В нашей задаче мы используем задачу классификации для предсказания, имеет ли пациент заболевание сердца (классы 0 или 1).

В рамках эксперимента мы рассматриваем задачу бинарной классификации, где значения целевой переменной 0 (отсутствие заболевания) и 1 (наличие заболевания) являются целями классификации. При необходимости, для многоклассовой классификации можно использовать все пять классов, но в данном случае мы ограничимся бинарным вариантом для простоты анализа.

Перед обучением моделей необходимо провести несколько шагов по предобработке данных:

1. Заполнение пропусков: Для начала мы заменяем все пропуски в данных на средние значения соответствующих признаков. Это важно для обеспечения целостности набора данных.
2. Преобразование категориальных признаков: В наборе данных

имеются категориальные признаки, например, пол пациента. Мы преобразуем эти признаки в числовые значения, чтобы модель могла их обработать.

3. Масштабирование признаков: Признаки могут иметь разные масштабы (например, возраст и уровень холестерина), поэтому рекомендуется нормализовать данные. Однако в нашем случае, для простоты, мы не будем масштабировать признаки, так как линейная и логистическая регрессии могут справляться с данными без масштабирования.

После подготовки данных, целевая переменная "num" будет преобразована для использования в задаче бинарной классификации: 0 — отсутствие заболевания, 1 — наличие заболевания.

2.2. Реализация линейной регрессии на Python

После подготовки данных, мы приступаем к обучению модели линейной регрессии. В случае задачи с бинарной целевой переменной, линейная регрессия может быть использована для прогнозирования вероятности, но для задачи классификации её использование может быть не так эффективно, как логистическая регрессия. Тем не менее, мы будем использовать её для сравнения.

Мы начинаем с разделения данных на обучающую и тестовую выборки, используя стандартное разделение 80/20. Это позволяет нам обучить модель на одной части данных и проверить её качество на другой, ранее не виденной.

Затем обучаем модель линейной регрессии на обучающих данных. Линейная регрессия пытается найти такие коэффициенты модели, которые минимизируют ошибку предсказания, обычно измеряемую с помощью среднеквадратичной ошибки (MSE).

После обучения модели, мы проверяем её производительность с использованием метрики R^2 и среднеквадратичной ошибки (MSE). R^2 показывает, насколько хорошо модель объясняет дисперсию в данных. Чем ближе R^2 к 1, тем лучше модель объясняет зависимость между признаками и

целевой переменной.

Мы оцениваем модель с использованием стандартных метрик:

- MSE (среднеквадратичная ошибка) показывает, насколько сильно предсказания модели отклоняются от реальных значений.
- R^2 (коэффициент детерминации) показывает, насколько хорошо модель объясняет изменчивость целевой переменной.

2.3. Реализация логистической регрессии на Python

После реализации линейной регрессии, мы переходим к более подходящей модели для классификации — логистической регрессии. Логистическая регрессия используется для решения задач классификации, где необходимо предсказать вероятность принадлежности объекта к одному из классов.

Во время предобработки разделяем целевую переменную на “0” и “1”, где “0” - нет заболеваний, а “1” - есть.

Подготовка данных аналогична предыдущему разделу: данные делятся на обучающую и тестовую выборки.

Обучение модели проводится с использованием метода оптимизации, который минимизирует логистическую функцию потерь, измеряя отклонение предсказанных вероятностей от истинных меток классов.

Для оценки качества логистической регрессии мы используем следующие метрики:

- Точность (Accuracy): показывает процент правильно классифицированных объектов.
- AUC-ROC: измеряет способность модели различать классы. Чем выше AUC, тем лучше модель.
- Матрица ошибок и отчет по классификации: для анализа точности предсказания каждого класса (precision, recall, F1-score).

2.4. Оценка точности моделей с использованием метрик

После реализации моделей линейной и логистической регрессии, мы сравниваем их производительность с использованием нескольких метрик.

Для линейной регрессии мы оценим MSE и R^2, чтобы понять, насколько точно модель предсказывает значения.

Для логистической регрессии мы используем accuracy, AUC-ROC, а также анализируем матрицу ошибок и отчет по классификации для более детального понимания качества классификации.

Метрики помогут нам понять, насколько эффективно каждая модель решает задачу, и определить, какая модель лучше подходит для данной задачи классификации.

3. СРАВНИТЕЛЬНЫЙ АНАЛИЗ МОДЕЛЕЙ ЛИНЕЙНОЙ И ЛОГИСТИЧЕСКОЙ РЕГРЕССИИ

3.1. Сравнение точности моделей на различных наборах данных

После реализации и оценки моделей линейной и логистической регрессии важно провести их сравнение, чтобы понять, какая модель более эффективно решает поставленную задачу. Для этого используем точность (accuracy) и другие метрики для оценки производительности моделей на тестовых данных.

Линейная регрессия: Точность для линейной регрессии измеряется с помощью среднеквадратичной ошибки (MSE) и R^2. Несмотря на то, что линейная регрессия может использоваться для задачи классификации, она не всегда хорошо справляется с бинарной классификацией, поскольку её цель — минимизация ошибки на непрерывных данных, а не на вероятностях классов.

Логистическая регрессия: Логистическая регрессия отлично подходит для классификационных задач и оценивает вероятность принадлежности объекта к одному из классов. Точность логистической регрессии измеряется через accuracy и AUC-ROC. Логистическая регрессия также использует метрики классификации (например, precision, recall, F1-score), что позволяет более точно оценить её способность правильно классифицировать объекты в различные категории (например, наличие или отсутствие заболевания).

Логистическая регрессия, как ожидается, будет показывать более высокие результаты по AUC-ROC и точности (accuracy), так как она специально предназначена для классификации. Линейная регрессия, с другой стороны, может не обеспечивать высокую точность при использовании её в задачах классификации, особенно когда классы сильно несбалансированы.

Логистическая регрессия явно демонстрирует лучшие результаты по меткам точности, особенно для задач классификации с бинарными или многоклассовыми метками.

3.2. Сравнение скорости обучения и вычислительных затрат

Важно также оценить скорость обучения и вычислительные затраты моделей, поскольку время, необходимое для обучения модели, и её вычислительная эффективность могут быть критичными для реальных приложений.

Линейная регрессия обучается быстро, поскольку она основана на методе наименьших квадратов или градиентном спуске, который обычно требует малого количества итераций для сходимости, особенно при малом числе признаков.

Логистическая регрессия может потребовать больше времени на обучение, особенно для многоклассовой классификации или в случае сложных данных. Логистическая регрессия использует градиентный спуск для минимизации логистической функции потерь, что может быть более вычислительно затратным по сравнению с линейной регрессией, особенно при большом объеме данных.

Для простых задач и небольших наборов данных линейная регрессия будет обучаться быстрее, так как её алгоритм имеет меньшую вычислительную сложность.

Логистическая регрессия может требовать большего времени на обучение в случае многоклассовых данных или сложных зависимостей между признаками.

3.3. Оценка интерпретируемости моделей

Одним из важных аспектов машинного обучения является интерпретируемость моделей, то есть способность объяснить, как модель пришла к своему решению. Это особенно важно в таких областях, как медицина, финансы и другие, где понимание модели и её решений имеет решающее значение.

Линейная регрессия имеет высокую интерпретируемость. Коэффициенты модели показывают, как изменение каждого признака влияет на целевую переменную. Например, если коэффициент для признака "возраст" равен 0.2, это означает, что при увеличении возраста на 1 год предсказанное значение возрастает на 0.2 единицы.

Логистическая регрессия также является интерпретируемой моделью, но её коэффициенты показывают влияние признаков на логарифм отношения вероятностей (log-odds), а не на непосредственное значение целевой переменной. Это требует дополнительной интерпретации, хотя большинство пользователей может легко понять влияние каждого признака через коэффициенты модели и предсказанные вероятности.

3.4. Влияние шума и различных параметров моделей на результаты

Рассмотрим, как модели реагируют на шум в данных и на различные параметры, такие как настройка скорости обучения, количество итераций и другие гиперпараметры.

Линейная регрессия может быть чувствительна к шуму, особенно в случае, если данные содержат сильные выбросы или нелинейные зависимости. В таких случаях линейная регрессия может давать смещенные или неточные редсказания.

Логистическая регрессия может быть более устойчивой к шуму, так как онаиспользует логистическую функцию для предсказания вероятностей, что делает её менее чувствительной к экстремальным значениям.

Для линейной регрессии важным параметром является скорость обучения при использовании градиентного спуска. Слишком высокая скорость обучения может привести к расходимости модели.

Для логистической регрессии важно настроить параметры сигмоида и количество итераций градиентного спуска для правильной сходимости

модели.

ЗАКЛЮЧЕНИЕ

В ходе исследования и анализа моделей линейной регрессии и логистической регрессии на основе набора данных Heart Disease UCI были сделаны следующие ключевые выводы:

1. Точность и производительность моделей

Логистическая регрессия показала лучшие результаты по точности и AUC-ROC, особенно для задачи классификации, так как она специально разработана для работы с вероятностями и категориальными переменными. Это делает её особенно подходящей для бинарных и многоклассовых задач классификации, таких как предсказание наличия или отсутствия сердечных заболеваний.

Линейная регрессия, хотя и может быть использована для классификации, показала более низкую точность, особенно для задачи с бинарной целевой переменной. Линейная регрессия, ориентированная на минимизацию среднеквадратичной ошибки, не так хорошо справляется с задачами классификации, где результатом являются вероятности принадлежности к классам.

2. Интепретируемость

Линейная регрессия имеет высокую интерпретируемость, так как её коэффициенты показывают линейное влияние каждого признака на зависимую переменную. Это делает модель прозрачной для пользователя, особенно в таких областях, как медицина, где важно понимать, как изменения в признаках (например, возраст, уровень холестерина) влияют на предсказания.

Логистическая регрессия также остаётся интерпретируемой, но её коэффициенты показывают влияние признаков на логарифм отношения вероятностей (log-odds), что требует дополнительных усилий для их

понимания. Однако использование вероятностей позволяет более эффективно интерпретировать результаты в контексте классификации.

3. Обработка шума и выбросов

Линейная регрессия может быть чувствительна к шуму и выбросам в данных, что может повлиять на её точность. В случае сильного шума или ненадежных данных модель может давать смещённые или неадекватные результаты.

Логистическая регрессия более устойчива к шуму, так как использует логистическую функцию для моделирования вероятностей. Это позволяет модели лучше справляться с выбросами и нереалистичными значениями, особенно в задачах классификации.

4. Скорость обучения и вычислительные затраты

Линейная регрессия обучается быстрее, поскольку она использует относительно простой алгоритм, основанный на методе наименьших квадратов или градиентном сплеше. Эта модель идеально подходит для быстрого прототипирования, особенно при небольших наборах данных и линейных зависимостях.

Логистическая регрессия, хотя и обучается медленнее, требует больше вычислительных затрат, особенно в случае многоклассовых задач или больших наборов данных. Однако её преимущества в точности классификации и способности работать с вероятностями делают её более подходящей для задач с категоричными целевыми переменными.

5. Применимость моделей

Линейная регрессия идеально подходит для задач, где нужно предсказать непрерывные значения, такие как предсказание концентрации загрязняющих веществ в воде или других количественных показателей. Она

хорошо работает, если зависимость между признаками и целевой переменной линейна.

Логистическая регрессия используется в задачах классификации, таких как предсказание наличия или отсутствия сердечного заболевания, анализ поведения пользователей или диагностика заболеваний. Она подходит для всех задач, где целевая переменная является категориальной (бинарной или многоклассовой).

На основе проведённого исследования, можно сделать несколько рекомендаций по выбору модели в зависимости от типа задачи и данных:

1. Задачи с бинарной или многоклассовой классификацией

Для задач, где целевая переменная является категориальной, например, предсказание наличия заболевания, рекомендуется использовать логистическую регрессию. Она специально предназначена для работы с вероятностями и классификацией, что делает её более подходящей для задач с несколькими категориями или бинарными метками. Логистическая регрессия будет более точной и эффективной в решении таких задач по сравнению с линейной регрессией.

2. Задачи с непрерывными целевыми переменными

Если задача требует предсказания непрерывных значений, таких как прогнозирование уровня загрязнителей в воде или других количественных переменных, линейная регрессия будет более подходящим выбором. Линейная регрессия работает быстро и эффективно для таких задач, при условии, что между признаками и целевой переменной существует линейная зависимость.

3. Устойчивость к шуму в данных

Если данные содержат много шума или выбросов, для задач классификации предпочтительнее использовать логистическую регрессию,

так как она более устойчива к этим проблемам благодаря своей способности работать с вероятностями. Логистическая регрессия менее чувствительна к выбросам, чем линейная регрессия, что делает её более надежной в условиях сложных данных.

4. Время обучения и вычислительные ресурсы

Если время на обучение модели критично или данные ограничены, и задача является простой (например, задача линейной зависимости), то линейная регрессия будет предпочтительнее. Она обучается быстрее и требует меньших вычислительных ресурсов, что делает её отличным выбором для быстрых прототипов и начальных этапов анализа данных.

5. Интерпретируемость моделей

Если важна интерпретируемость модели (например, в медицинских исследованиях или при принятии бизнес-решений), линейная регрессия будет предпочтительнее, так как её коэффициенты легко интерпретировать. Логистическая регрессия также предоставляет интерпретируемые результаты, но требует дополнительных усилий для понимания влияния признаков на вероятности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гудфеллоу И., Бенжио Й., Курвилл А. Глубокое обучение. — М.: ДМК Пресс, 2017. — 800 с.
2. Рашка С. Python и машинное обучение. — М.: БХВ-Петербург, 2018. — 416 с.
3. Хасти Т., Тибширани Р., Фридман Дж. Элементы статистического обучения. — М.: Финансовый университет при Правительстве РФ, 2012. — 760 с.
4. Джеймс Г., Уиттен Д., Хасти Т., Тибширани Р. Введение в статистическое обучение с применением R. — М.: Диалектика, 2015. — 512 с.
5. МакКинни У. Python для анализа данных. — М.: БХВ-Петербург, 2017. — 560 с.
6. Короткевич В. Машинное обучение: алгоритмы и приложения. — М.: Научная книга, 2019. — 432 с.
7. Харт П. Машинное обучение. Научный подход. — М.: Вильямс, 2018. — 320 с.
8. Бёрнс Р. Л. Статистика и машинное обучение. Методология и практика. — М.: Наука, 2016. — 450 с.
9. UCI Machine Learning Repository: Heart Disease Data Set. URL: https://archive.ics.uci.edu/ml/datasets/heart+disease (дата обращения: 10.07.2025).
10. Scikit-learn Documentation. URL: https://scikit-learn.org/stable/modules/linear_model.html (дата обращения: 12.07.2025).
11. Рашка С., Мирали В. Машинное обучение с использованием Python. — М.: БХВ-Петербург, 2020. — 352 с.
12. Патрушев С. А. Машинное обучение. Курс лекций. — М.: Высшая школа, 2021. — 300 с.
13. Нг А. Машинное обучение. Курс лекций [Электронный ресурс]. URL: https://www.coursera.org/learn/machine-learning (дата обращения: 10.07.2025).

14. Stepik: Курс "Машинное обучение" [Электронный ресурс]. URL: https://stepik.org/course/486 (дата обращения: 12.07.2025).

ПРИЛОЖЕНИЕ 1

Листинг программного кода

Листинг П1.1

logistic_heart.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,
roc_auc_score
```

In [86]:

```
#Загрузка данных
column_names = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num']
data = pd.read_csv('processed.cleveland.data', header=None, names=column_names,
na_values='?')
print("Вывод первых 5 строк для проверки:")
data.head()

#Проверка на пропуски
print("Пропуски в данных:")
data.isnull().sum()
```

In [88]:

```
#Просмотр общей информации о данных
data.info()
```

In [89]:

```
#Заполнение пропусков средним значением (для числовых признаков)
data.fillna(data.mean(), inplace=True)
```

In [90]:

```
#Преобразуем целевую переменную
data['num'] = data['num'].apply(lambda x: 1 if x > 0 else 0)#Если были диагнозированы
сердечные заболевания - 1, иначе - 0
#Проверим результат
data['num'].value_counts()
```

In [91]:

```
X = data.drop('num', axis =1)
y = data['num']  # Целевая переменная: 0 – отсутствие заболевания, 1 – наличие
```

In [92]:

```
#Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
random_state=42)
```

In [93]:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [94]:

```
#Создание и обучение модели логистической регрессии
model = LogisticRegression(max_iter=1000)
model.fit(X_train_scaled, y_train)
```

In [95]:

```
#Предсказание на тестовой выборке
y_pred = model.predict(X_test_scaled)
```

In [96]:

```
#Метрики качества модели
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
class_report = classification_report(y_test, y_pred)
roc_auc = roc_auc_score(y_test, model.predict_proba(X_test)[:, 1])
```

In [97]:

```
print(f"Точность модели: {accuracy:.2f}")
print(f"AUC-ROC: {roc_auc:.2f}")
print("Матрица ошибок:")
print(conf_matrix)
print("Отчет по классификации:")
print(class_report)
```

In [98]:

```
from sklearn.dummy import DummyClassifier
from sklearn.metrics import accuracy_score, classification_report, roc_auc_score

#Обучаем dummy-классификатор (предсказывает наиболее частый класс)
dummy_clf = DummyClassifier(strategy='most_frequent')
dummy_clf.fit(X_train_scaled, y_train)

#Предсказания dummy-модели
y_dummy_pred = dummy_clf.predict(X_test_scaled)

#Метрики dummy-классификатора
print("Метрики dummy-модели:")
print(f"Accuracy: {accuracy_score(y_test, y_dummy_pred):.2f}")
print(f"AUC-ROC: {roc_auc_score(y_test, dummy_clf.predict_proba(X_test_scaled)[:,
1]):.2f}")
print("Отчет по классификации:")
print(classification_report(y_test, y_dummy_pred))
```

In [99]:

```
#График матрицы ошибок
plt.figure(figsize=(6, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Нет
заболевания', 'Есть заболевание'], yticklabels=['Нет заболевания', 'Есть заболевание'])
plt.xlabel('Предсказано')
plt.ylabel('Реальное')
plt.title('Матрица ошибок')
plt.show()
```

In [100]:

```
#Таблица коэффициентов
coefficients = pd.DataFrame({
    'Признак': features,
    'Коэффициент': model.coef_[0]
})
print("Влияние признаков на вероятность заболевания:")
print(coefficients)
```

In [102]:

```
#Возьмем произвольные значения для признаков
new_data = pd.DataFrame({
    'age': [63], #Возраст
    'sex': [1], #Пол (1 - мужской, 0 - женский)
    'cp': [3], #Тип боли (3 - острый)
    'trestbps': [145], #Систолическое давление
    'chol': [233], #Холестерин
    'fbs': [1], #Уровень сахара (1 - больше 120 мг/дл)
    'restecg': [0], #ЭКГ в покое (0 - нормальный)
    'thalach': [150], #Макс. пульс
    'exang': [0], #Стенокардия (0 - нет)
    'oldpeak': [2.3], #Снижение ST-сегмента
    'slope': [3], #Складка на пике (3 - крутой)
    'ca': [0], #Количество крупных сосудов
    'thal': [1] #Талассемия (1 - нормальная)
})
```

In [103]:

```
#Предсказание
prediction = model.predict(new_data)
print(f"Предсказанная вероятность заболевания: {'Есть заболевание' if prediction[0] == 1
else 'Нет заболевания'}")
```

Листинг П1.2

linear_heart.ipynb

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
```

In [69]:

```
#Загрузка данных
column_names = ['age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach',
'exang', 'oldpeak', 'slope', 'ca', 'thal', 'num']
data = pd.read_csv('processed.cleveland.data', header=None, names=column_names,
na_values='?')
print("Вывод первых 5 строк для проверки")
data.head()
```

In [70]:

```
#Проверка на пропуски
print("Пропуски в данных:")
data.isnull().sum()
```

In [71]:

```
#Просмотр общей информации о данных
data.info()
```

In [72]:

```
#Заполнение пропусков средним значением (для числовых признаков)
data.fillna(data.mean(), inplace=True)
```

In [73]:

```
#Вычисление корреляционной матрицы
corr_matrix = data.corr()

#Визуализация корреляции между признаками и целевой переменной (target)
```

```
plt.figure(figsize=(10, 6))
sns.heatmap(corr_matrix[['num']], annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title("Корреляция признаков с целевой переменной")
plt.show()
```

In [74]:

```
features = ['ca', 'thal', 'oldpeak', 'cp', 'exang', 'slope', 'age', 'sex', 'restecg',
'trestbps', 'thalach']
X = data[features]
y = data['num']
```

In [75]:

```
#Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

In [76]:

```
#Создание и обучение модели
model = LinearRegression()
model.fit(X_train, y_train)
```

In [77]:

```
#Предсказание на тестовой выборке
y_pred = model.predict(X_test)
```

In [78]:

```
#Метрики качества модели
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"MSE: {mse:.2f}")
print(f"R²: {r2:.2f}")
```

In [79]:

```
from sklearn.dummy import DummyRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

#Обучение dummy-модели (предсказывает среднее значение)
dummy_reg = DummyRegressor(strategy='mean')
dummy_reg.fit(X_train, y_train)

#Предсказания dummy-модели
y_dummy_pred = dummy_reg.predict(X_test)
```

In [80]:

```
#Метрики dummy-модели
print("Метрики дамми модели (линейная регрессия):")
print(f"MSE: {mean_squared_error(y_test, y_dummy_pred):.2f}")
print(f"R^2: {r2_score(y_test, y_dummy_pred):.2f}")
```

In [81]:

```
#График реальных и предсказанных значений
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel("Реальные значения")
plt.ylabel("Предсказанные значения")
plt.title("Заболевание сердца: реальные и предсказанные значения")
plt.plot([0, 4], [0, 4], color='red')  #Идеальная линия
plt.show()
```

In [82]:

```
#Таблица коэффициентов
coefficients = pd.DataFrame({
```

```
    'Признак': features,
    'Коэффициент': model.coef_
})
print("Влияние признаков на вероятность заболевания:")
print(coefficients)
```

In [83]:

```
#Возьмем произвольные значения для признаков
new_data = pd.DataFrame({
    'age': [63], #Возраст
    'sex': [1], #Пол (1 - мужской, 0 - женский)
    'cp': [3], #Тип боли (3 - острый)
    'trestbps': [145], #Систолическое давление
    'restecg': [0], #ЭКГ в покое (0 - нормальный)
    'thalach': [150], #Макс. пульс
    'exang': [0], #Стенокардия (0 - нет)
    'oldpeak': [2.3], #Снижение ST-сегмента
    'slope': [3], #Складка на пике (3 - крутой)
    'ca': [0], #Количество крупных сосудов
    'thal': [1] #Талассемия (1 - нормальная)
}, columns=X.columns)
```

In [84]:

```
prediction = model.predict(new_data)
print(f"Предсказанная вероятность заболевания: {prediction[0]:.2f}")
```