МИНОБРНАУКИ РОССИИ
федеральное государственное бюджетное образовательное
учреждение высшего образования
«Национальный исследовательский университет «МЭИ»

Отчет по научно-исследовательской работе

по теме:

«Обзор методов классического машинного обучения для задач регрессии»

Работу выполнил
студент группы А-13б-20
Бегунов Никита
Научный руководитель:
Кожевников Антон Вадимович

Москва 2023

Оглавление

1. Введение

Машинное обучение играет ключевую роль в современном информационном обществе, проникая в различные сферы деятельности человека. В контексте развития этой области особое внимание уделяется методам классического машинного обучения, основанным на статистических и алгоритмических подходах. Одним из важных направлений в машинном обучении является задача регрессии, где осуществляется прогнозирование непрерывных значений на основе имеющихся данных.

Цель данной научно-исследовательской работы состоит в обзоре различных методов классического машинного обучения, применяемых для решения задачи регрессии. Анализ и сравнительная оценка эффективности данных методов позволят выявить их особенности, преимущества и ограничения, что в свою очередь способствует более глубокому пониманию применимости различных подходов к решению задач регрессии.

В контексте быстрого развития технологий и увеличения объема данных становится крайне важным понимание преимуществ и недостатков методов классического машинного обучения для эффективного применения их в реальных прикладных задачах.

2. Описание задачи регрессии

Регрессия в машинном обучении – это тип контролируемого обучения, при котором модель обучается прогнозировать непрерывный числовой вывод на основе заданного набора входных признаков. Она используется для прогнозирования значения непрерывной переменной на основе значений других переменных [1]. Основная цель регрессии состоит в том, чтобы построить модель, которая может предсказать значения целевой переменной (откликов) на основе входных данных (факторов).

Задача регрессии заключается в том, чтобы найти функциональную зависимость между независимыми (факторными) переменными и зависимой переменной. Пример: предсказание стоимости дома на основе его площади, количества комнат и расположения. Здесь мы имеем три независимые переменные (площадь, количество комнат, расположение) и одну зависимую переменную (стоимость дома) [2].

Этаж	Этажей в доме	Площадь	Количество комнат	Цена
8	10	82.6	3	6050000
5	24	69.1	2	5120000
5	9	66	3	4000000
12	16	38	2	2000000

3. Алгоритмы классического машинного обучения для задачи регрессии

3.1. Линейная и полиномиальная регрессия

Линейная и полиномиальная регрессия — это два распространенных статистических метода решения задачи регрессии в машинном обучении, которые используются для моделирования отношений между независимыми переменными (факторами) и зависимой переменной (откликом) [3].

Линейная регрессия предполагает, что зависимость между переменными линейна, то есть можно представить ее в виде прямой линии в пространстве признаков и целевой переменной. Простая линейная регрессия определяется линейной функцией:

$$Y = \beta_0 X + \beta_1 + \varepsilon,$$

где β_0 и β_1 – две независимые константы, представляющие наклон регрессии, тогда как ε – член ошибки [4]. Пример модели для простой линейной регрессии приведен на рисунке 1.

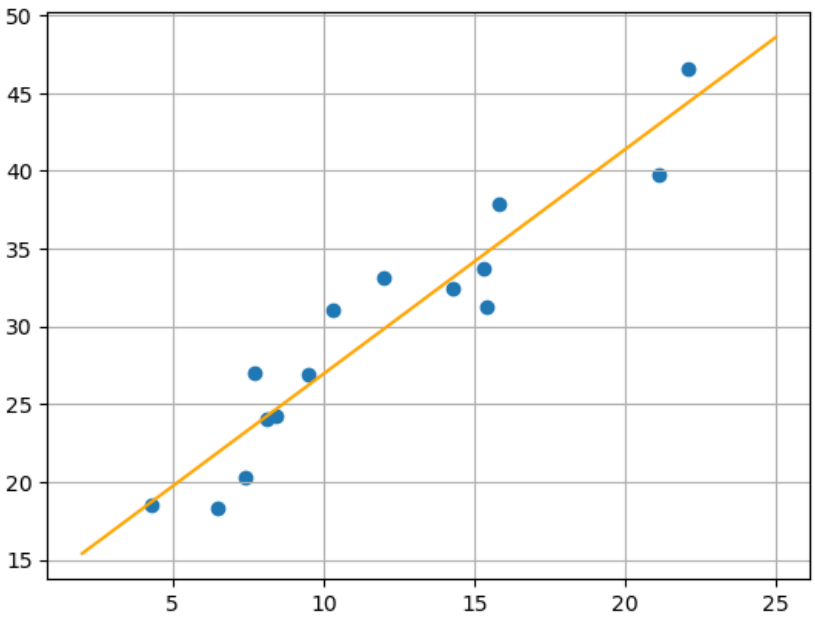

Рис. 1. Простая линейная регрессия.

При множественном линейном регрессионном анализе набор данных содержит одну зависимую переменную и несколько независимых переменных. Функция линии линейной регрессии изменяется и включает в себя большее количество факторов:

$$Y = \beta_0 X_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon$$

По мере увеличения количества переменных-предикторов константы β также соответственно увеличиваются [4].

Модель линейной регрессии стремится найти оптимальную прямую линию (в случае одной переменной) или гиперплоскость (в случае нескольких переменных), которая наилучшим образом соответствует данным. Для этого минимизируется сумма квадратов разностей между предсказанными значениями и фактическими значениями целевой переменной. Это достигается с помощью метода наименьших квадратов или других оптимизационных методов.

Полиномиальная регрессия расширяет линейную модель, позволяя моделировать нелинейные отношения между переменными. Вместо прямой линии полиномиальная регрессия может использовать кривые, соответствующие полиномиальным функциям высших степеней. Это позволяет более гибко моделировать данные, которые не соответствуют линейной зависимости. Для полиномиальной регрессии формула принимает вид [5]:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \cdots + \beta_n X^n + \varepsilon.$$

Полиномиальная регрессия с многочленом второй степени графический изображена на рисунке 2.

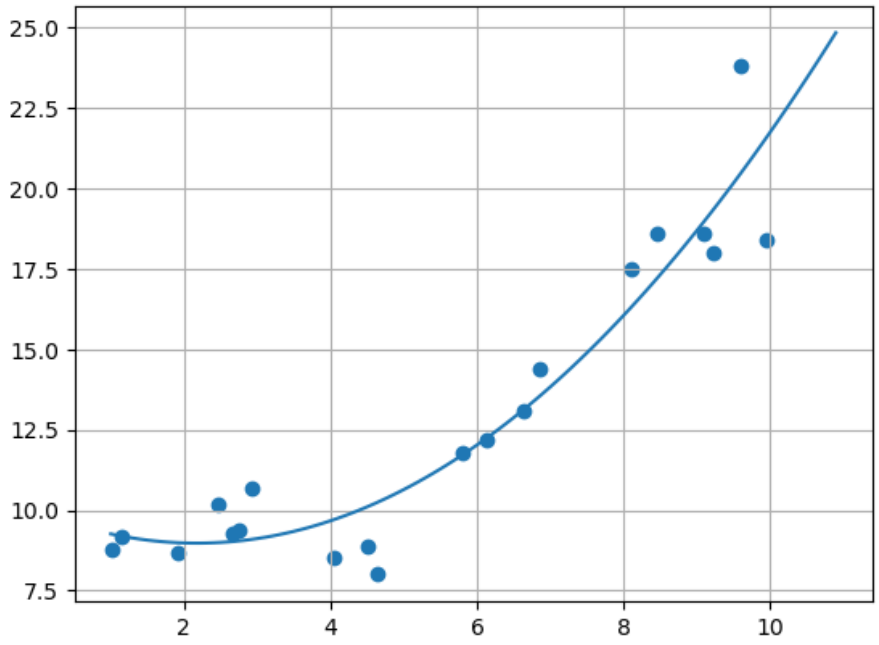

Рис. 2. Полиномиальная регрессия с многочленом второй степени.

Оба метода могут быть использованы для прогнозирования значений целевой переменной на основе новых данных, а также для понимания природы взаимосвязей между переменными в данных. Однако важно помнить, что полиномиальная регрессия более склонна к переобучению на данных из-за большей гибкости в моделировании нелинейных отношений.

3.2. Решающее дерево (Decision Tree)

Решающие деревья представляют собой метод машинного обучения, который может использоваться для задач классификации и регрессии. Они работают путем разделения данных на подгруппы на основе определенных признаков, чтобы предсказать целевую переменную [6].

Процесс построения решающего дерева начинается с корня, который представляет собой весь набор данных. Далее дерево делит данные на подмножества на основе определенных признаков таким образом, чтобы в каждой из новых групп целевая переменная была как можно более однородной. Это делается путем выбора наилучшего признака и значения этого признака для разделения данных.

Процесс разделения повторяется рекурсивно для каждого получившегося подмножества, пока не будет выполнен некоторый критерий останова, например, достижение определенной глубины дерева или минимального числа образцов в узле.

Когда дерево полностью построено, для новых данных оно использует свою структуру, чтобы определить путь от корня до листового узла, в котором будет содержаться предсказание для данного наблюдения. Для задачи регрессии это обычно является средним значением целевой переменной в листовом узле. Приведем пример: необходимо предсказать значение функции f(x), изображенной на рисунке 3. Дерево решения для такой функции изображено на рисунке 4.

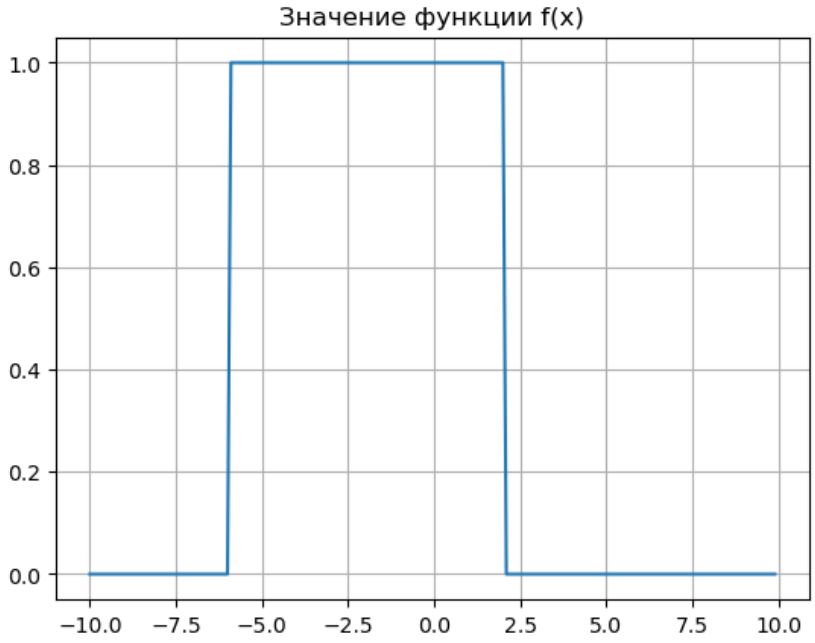

Рис. 3. Функция, для которой необходимо построить решающее дерево.

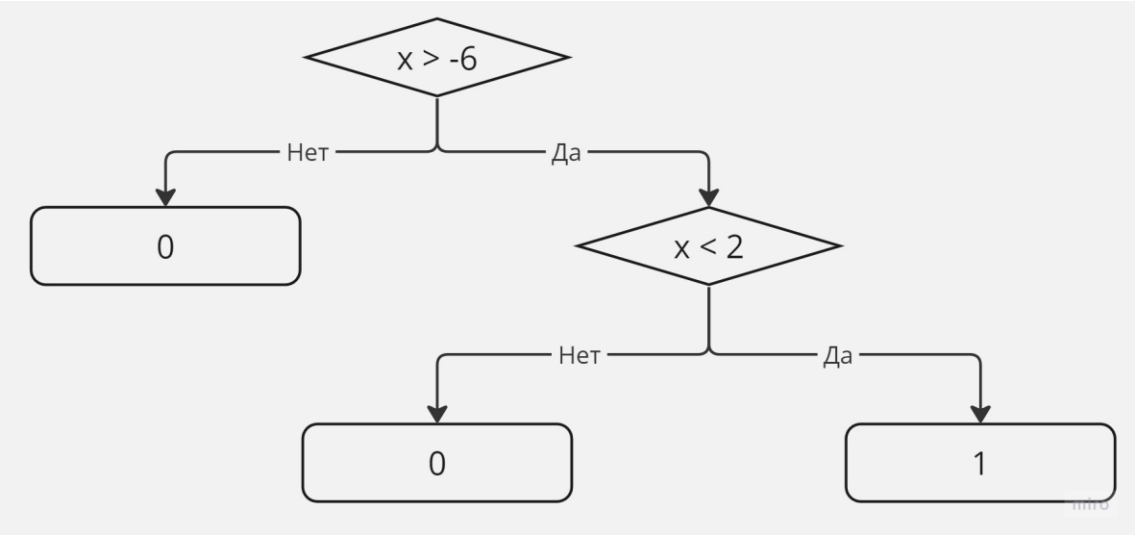

Рис. 4. Решающее дерево для f(x).

Одна из основных преимуществ решающих деревьев в регрессии состоит в их способности моделировать нелинейные зависимости между признаками и целевой переменной без необходимости предварительной обработки данных. Однако деревья могут быть склонны к переобучению, особенно если не ограничивать их глубину или не используется правильная настройка параметров. Чтобы справиться с этой проблемой, часто применяют ансамбли деревьев, такие как случайный лес или градиентный бустинг, которые комбинируют несколько деревьев для получения более устойчивых и точных прогнозов.

3.3. Случайный лес (Random Forest)

Случайный лес (Random Forest) — это мощный метод машинного обучения, который обычно используется для задач как классификации, так и регрессии. Он основан на идее создания множества деревьев решений в процессе обучения и усреднения их прогнозов для получения более точного и устойчивого результата.

Для решения задачи регрессии случайный лес использует несколько деревьев решений, каждое из которых обучается на различном подмножестве данных и с использованием различных признаков. Когда нужно сделать прогноз для нового объекта, каждое дерево выдает свой прогноз, а затем их прогнозы усредняются для получения окончательного результата [7]. Схематически работу случайного леса можно изобразить следующим образом (рис. 5) [8]:

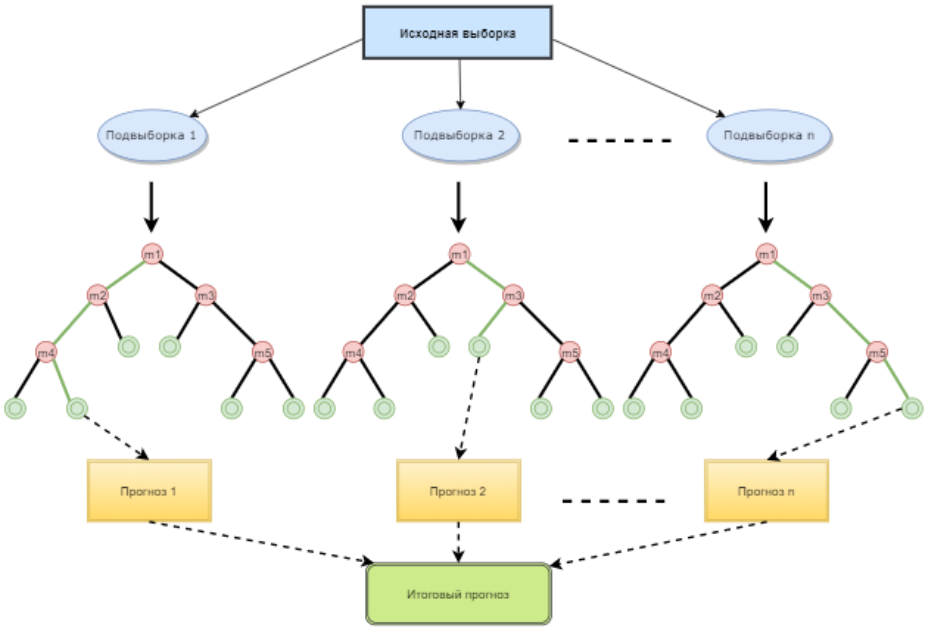

Рис. 5. Схематическое изображение случайного леса.

Преимущества случайного леса для задач регрессии включают в себя:

- Устойчивость к переобучению: благодаря использованию нескольких деревьев и случайного выбора признаков для обучения каждого дерева, случайный лес обычно более устойчив к переобучению, чем отдельные деревья решений.
- Хорошая обобщающая способность: случайный лес способен обобщать данные, работая с выборками и подмножествами признаков, что часто приводит к более точным прогнозам на новых данных.
- Способность работать с большим количеством признаков: он может эффективно обрабатывать большое количество признаков без необходимости в предварительной обработке данных.
- Высокая точность: за счет усреднения прогнозов нескольких деревьев случайный лес часто обеспечивает высокую точность предсказаний.

Однако, как и у любого метода, у случайного леса есть свои ограничения. Он может тратить больше времени на обучение, особенно на больших наборах данных, чем деревья решений или линейные алгоритмы, за счет увеличения количества деревьев, которые необходимо обучить, а также за счет комбинации прогнозов из многих деревьев, и склонен к переобучению при работе с шумными данными.

Для применения случайного леса в задаче регрессии важно настроить параметры модели, такие как количество деревьев, максимальная глубина деревьев и минимальное количество объектов в листе дерева, чтобы достичь наилучшей производительности и предсказательной силы модели.

3.4. Бустинг

Бустинг — это метод машинного обучения, который объединяет слабые модели (например, деревья решений) в сильную предсказательную модель. Бустинг строит ансамбли моделей путем последовательного объединения нескольких слабых деревьев решений. Выходным данным отдельных деревьев присваиваются веса. Затем неправильным классификациям из первого дерева решений присваивается больший вес, после чего данные передаются в следующее дерево. После многочисленных циклов бустинг объединяет слабые классификаторы в один мощный алгоритм прогнозирования [9].

Один из наиболее популярных алгоритмов бустинга для регрессии — градиентный бустинг. Он работает итеративно, добавляя новую модель к уже имеющимся, исправляя ошибки предыдущих моделей. Схематически работу градиентного бустинга можно следующим образом [10]:

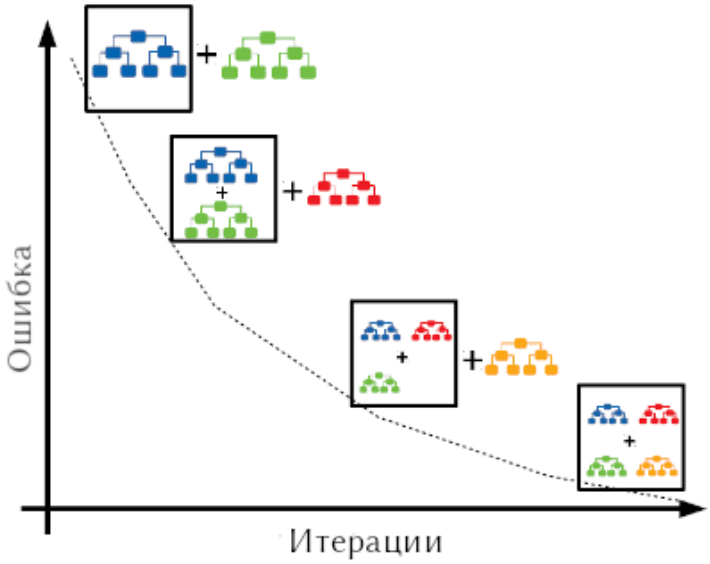

Рис. 6. Схематический принцип работы градиентного бустинга.

На каждой итерации модель анализирует ошибки предсказания предыдущей модели. Она фокусируется на этих ошибках, стремясь минимизировать их в новой модели. Обычно используется градиентный спуск для нахождения оптимальных параметров новой модели. Бустинг создает сильную модель, комбинируя множество слабых моделей. Каждая новая модель в финальной комбинации исправляет ошибки предыдущих, повышая общее качество предсказания.

Преимущества бустинга для задач регрессии:

- Высокая точность: благодаря комбинации моделей бустинг может достигать высокой точности предсказаний.

- Устойчивость к переобучению: благодаря итеративному процессу обучения, бустинг способен избегать переобучения, контролируя сложность модели.
- Работа с различными типами данных: бустинг хорошо работает с разными типами данных, обрабатывая как числовые, так и категориальные признаки.

Однако бустинг может быть вычислительно затратным и требовать тщательной настройки гиперпараметров для достижения оптимальной производительности. Несмотря на это, он остается мощным инструментом для задач регрессии благодаря своей способности к повышению качества предсказаний путем комбинирования простых моделей в более сложные и улучшенные алгоритмы.

3.5. Метод k-ближайших соседей (KNN)

Метод k-ближайших соседей – один из методов машинного обучения, который используется для задач классификации и регрессии. Основная идея метода KNN заключается в том, чтобы предсказать значение целевой переменной для новой точки данных путем усреднения значений ближайших к ней соседей из обучающего набора данных. KNN использует расстояние между точками данных для определения ближайших соседей. В случае регрессии, где требуется предсказать числовое значение, обычно используются метрики расстояния, такие как евклидово расстояние, манхэттенское расстояние (расстояние в городских кварталах) или другие. Выбор метрики расстояния зависит от особенностей данных и самой задачи. Например, если данные имеют различную важность для разных признаков, может быть предпочтительно использовать взвешенные расстояния или применять различные метрики для разных признаков [11].

Базовая регрессия ближайших соседей использует одинаковые веса: то есть каждая точка в локальной окрестности вносит единообразный вклад в классификацию точки запроса (weights = ‘uniform’ присваивает всем точкам одинаковые веса). При некоторых обстоятельствах может быть выгодно взвесить точки так, чтобы близлежащие точки вносили больший вклад в регрессию, чем удаленные точки (например, weights = ‘distance’ назначает веса, обратно пропорциональные расстоянию от точки запроса) [12]. Примеры построения моделей регрессии методом KNN изображены на рисунке 7.

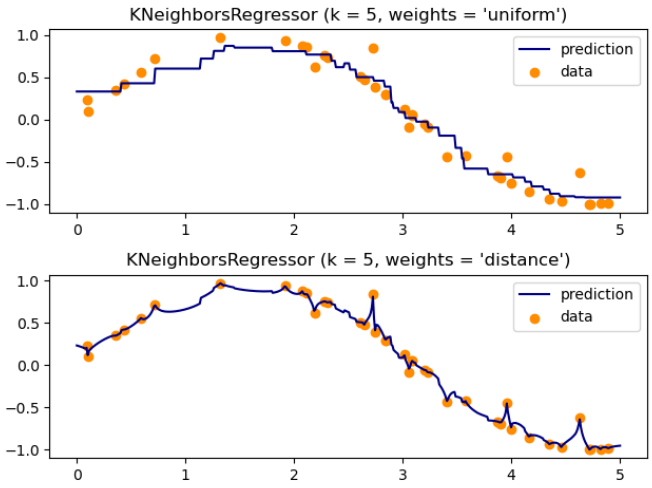

Рис. 7. Модели регрессии k-ближайших соседей.

Принцип работы KNN для задачи регрессии:

1. Определение ближайших соседей: для новой точки данных модель ищет k ближайших соседей в обучающем наборе данных. Расстояние между точками обычно измеряется с использованием метрики, такой как евклидово расстояние.
2. Вычисление значения: для регрессии значения целевой переменной вычисляются как среднее или взвешенное среднее значений целевой переменной всех k ближайших соседей.
3. Прогнозирование: после вычисления средних значений модель прогнозирует значение для новой точки данных как результат.

К преимуществам KNN относят простоту реализации и понимая, эффективности в задачах с небольшим количеством признаков или данных, а также адаптивность к изменениям в данных.

Недостатки метода k-ближайших соседей:

- Алгоритм работает значительно медленнее при увеличении объема выборки, предикторов или независимых переменных.
- Из аргумента выше следуют большие вычислительные затраты во время выполнения.
- Всегда нужно определять оптимальное значение k.

Для использования KNN в задаче регрессии необходимо выбрать оптимальное значение параметра k (количество ближайших соседей). Это может быть сделано с помощью кросс-валидации или других методов оценки производительности модели. Важно также масштабировать признаки перед использованием KNN, так как он чувствителен к масштабу из-за измерения расстояний между точками данных.

3.6. Метод опорных векторов (SVM)

Метод опорных векторов (Support Vector Machines, SVM) в основном используется для задач классификации, но также может быть адаптирован для решения задач регрессии. Идея SVM в регрессии состоит в том, чтобы найти линию или гиперплоскость в пространстве признаков, которая наилучшим образом соответствует данным. Вместо того чтобы точно воспроизвести значения, как это делается в типичных методах регрессии, SVM стремится минимизировать ошибку предсказания, допуская отклонение, называемое "предельным зазором" (margin) [13]. На рисунке 8 приведено абстрактное изображение для SVM метода для задачи регрессии для линейной и нелинейной зависимости в данных [14].

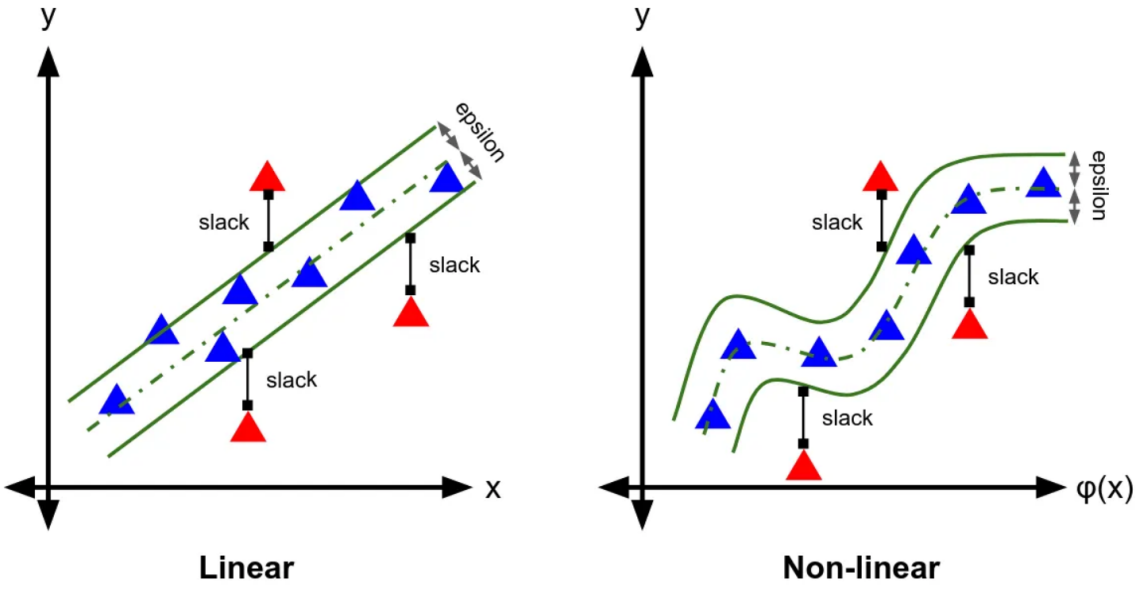

Рис. 8. SVM для регрессии для линейной и нелинейной зависимости.

Основные шаги при использовании SVM для регрессии:

1. Выбор ядра (Kernel): SVM использует ядра для преобразования данных в более высокие размерности, чтобы лучше разделить их. Популярные ядра включают линейное, полиномиальное и радиально-базисные функции (RBF).
2. Определение функции потерь: для регрессии используется функция потерь, которая штрафует модель за отклонение предсказанных значений от фактических. Обычно используется функция потерь эпсилон-интенсивности (epsilon-insensitive loss).
3. Настройка гиперпараметров: SVM имеет гиперпараметры, такие как параметр регуляризации C и параметры ядра (например, степень полинома, ширина RBF-ядра), которые нужно настроить для достижения лучшей производительности модели.

4. Оптимизация: Цель состоит в том, чтобы найти гиперплоскость (или линию) и оптимальную ширину предельного зазора, минимизирующие ошибку предсказания при учете допустимого отклонения (epsilon).
5. Подбор границ: SVM определяет "опорные вектора", которые являются точками данных, расположенными ближе всего к разделяющей гиперплоскости. Эти точки определяют формирование границы регрессии.

SVM для регрессии обладает свойствами, позволяющими учитывать выбросы и работать в многомерных пространствах признаков. Однако выбор ядра и настройка гиперпараметров могут потребовать некоторой экспериментальной работы для достижения наилучших результатов.

4. Сравнительный анализ методов

Резюмируя приведенные выше методы, можно составить таблицу со сравнением принципов их работы, а также преимуществами и недостатками каждого из них (таблица 1).

Таблица 1. Сравнительный анализ методов.

Метод	Описание	Преимущества	Недостатки
Линейная регрессия	Простой и интерпретируемый метод, который строит линейную модель зависимости между входными признаками и целевой переменной.	Интерпретируемость, быстрота обучения, хорошо работает, когда зависимость между признаками и целевой переменной линейная.	Не улавливает нелинейные зависимости между данными.
Полиномиальная регрессия	Расширение линейной регрессии, которое включает полиномиальные признаки для улавливания нелинейных зависимостей.	Может улавливать сложные нелинейные зависимости.	Сложность модели возрастает с увеличением степени полинома, что может привести к переобучению.
Решающее дерево (Decision Tree)	Структура, представляющая собой дерево решений, разбивающее	Может улавливать сложные зависимости, не требует	Склонно к переобучению, нестабильность при небольших

	данные на подгруппы на основе правил.	предобработки данных.	изменениях в данных.
Случайный лес (Random Forest)	Ансамбль решающих деревьев, где каждое дерево обучается на случайной подвыборке данных и использует баггинг для уменьшения переобучения.	Устойчив к переобучению, хорошая обобщающая способность, способен работать с большими объемами данных.	Может быть медленным в обучении и предсказаниях на больших данных.
Бустинг	Ансамбль, который строит последовательность слабых моделей, каждая из которых исправляет ошибки предыдущей.	Высокая точность, способность работать с разнородными данными.	Может быть склонен к переобучению, более сложен в настройке параметров.
Метод k-ближайших соседей (KNN)	Основан на идее, что объекты с похожими признаками имеют похожие целевые значения.	Прост в реализации, хорошо работает для небольших наборов данных.	Чувствителен к выбросам, требует хорошей предобработки данных, может быть медленным для больших наборов данных.
Метод опорных векторов (SVM)	Строит гиперплоскость, максимально разделяющую классы в пространстве признаков.	Эффективен в пространствах высокой размерности, хорошо работает с небольшими выборками.	Может быть чувствителен к выбору ядра и параметров, требует нормализации данных.

Итоговый выбор метода для задачи регрессии зависит от многих факторов, таких как размер и природа данных, необходимая точность, склонность к переобучению и требования к скорости обучения/предсказания.

5. Заключение

Обзор классических методов машинного обучения для задач регрессии позволяет увидеть разнообразие подходов к решению задачи предсказания числовых значений. Линейные методы предоставляют простоту и интерпретируемость, в то время как ансамблированые методы демонстрируют высокую гибкость и способность адаптироваться к сложным данным.

Оценка преимуществ и недостатков каждого метода в контексте регрессии позволяет понять их эффективность в различных сценариях использования. Например, решающие деревья могут быть уязвимы к переобучению, в то время как метод опорных векторов демонстрирует хорошую устойчивость к выбросам.

Однако, несмотря на успехи, существует необходимость в дальнейших исследованиях и разработках. Перспективы развития методов регрессии включают в себя улучшение алгоритмов для работы с большими объемами данных, повышение устойчивости к шуму и выбросам, а также разработку новых подходов, объединяющих различные методы для получения еще более точных и интерпретируемых результатов.

Таким образом, обзор методов классического машинного обучения для задач регрессии подчеркивает их значимость, актуальность и перспективность в контексте современной науки и технологий

6. Список использованных источников

1. В. О. Алексенко, Д. Г. Буслович, Ц. Ло, С. В. Панин НАНОТЕХНОЛОГИИ. ИНФОРМАЦИЯ. РАДИОТЕХНИКА (НИР-23). Материалы Всероссийской молодежной научно-практической конференции (Омск, 18 апреля 2023 года). - ОмГТУ, 2023. - 478 с.
2. Что такое регрессия и классификация
   https://sky.pro/media/chto-takoe-regressiya-i-klassifikacziya/
3. Регрессия в машинном обучении
   https://proglib.io/p/ml-regression
4. Линейная регрессия
   https://aws.amazon.com/ru/what-is/linear-regression/
5. Модель полиномиальной регрессии
   https://habr.com/ru/articles/414245/
6. Деревья решений
   https://scikit-learn.ru/1-10-decision-trees/
7. Случайный лес
   https://tproger.ru/translations/python-random-forest-implementation
8. Тезисное описание алгоритма Random Forest
   https://www.mql5.com/ru/articles/3856
9. Бустинг в машинном обучении
   https://aws.amazon.com/ru/what-is/boosting/
10. Градиентный бустинг
    https://spark-school.ru/blogs/gradient-boosting-ml/
11. Метод ближайших соседей (KNN)
    https://learnmachinelearning.fandom.com/ru/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%B1%D0%BB%D0%B8%D0%B6%D0%B0%D0%B9%D1%88%D0%B8%D1%85_%D1%81%D0%BE%D1%81%D0%B5%D0%B4%D0%B5%D0%B9_(kNN)
12. Регрессия ближайших соседей
    https://scikit-learn.ru/1-6-nearest-neighbors/
13. Метод опорных векторов (SVM)
    https://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_%D0%BE%D0%BF%D0%BE%D1%80%D0%BD%D1%8B%D1%85_%D0%B2%D0%B5%D0%BA%D1%82%D0%BE%D1%80%D0%BE%D0%B2_(SVM)
14. Support Vector Machine: Regression
    https://medium.com/it-paragon/support-vector-machine-regression-cf65348b6345