GeekBrains

Обучение с учителем. Библиотека Scikit-learn. Регрессия

Библиотеки Python для Data Science

Оглавление

Введение

Машинное обучение — это наука об изучении алгоритмов, которые автоматически совершенствуются под воздействием опыта.

Мы научились обрабатывать данные, проводить первичный и визуальный анализ. Наши данные готовы, теперь пришла пора посмотреть, как можно работать с математическими моделями.

Scikit-learn, также известный как sklearn — это одна из наиболее популярных библиотек машинного обучения на языке Python. Библиотека предоставляет широкий спектр алгоритмов машинного обучения, включая классификацию, регрессию, кластеризацию и многие другие, а также инструменты для предобработки данных, оценки моделей и настройки параметров.

Значимость sklearn заключается в том, что она предоставляет простой и удобный интерфейс для использования алгоритмов машинного обучения, что делает ее доступной для широкого круга пользователей, как начинающих, так и опытных специалистов. Библиотека также является открытым и бесплатным инструментом, что позволяет разработчикам и исследователям быстро применять и сравнивать различные методы машинного обучения в своих проектах.

Кроме того, scikit-learn является частью экосистемы инструментов машинного обучения на Python, что позволяет пользователю быстро интегрировать ее с другими библиотеками и инструментами для анализа данных. Ее большое сообщество пользователей и разработчиков также обеспечивает поддержку и развитие библиотеки в будущем.

В целом, scikit-learn является важным инструментом в области машинного обучения, который обеспечивает простоту и удобство использования, широкий спектр функций и возможных применений, а также открытость и гибкость для интеграции с другими инструментами и экосистемами на Python.

Обучение с учителем является одним из наиболее распространенных методов машинного обучения. В процессе обучения модели, аналитик использует данные, которые уже имеют определенную разметку для того, чтобы научиться прогнозировать новые значения.

Одной из задач обучения с учителем является задача регрессии, которая заключается в прогнозировании непрерывной переменной на основе заданных входных данных, а также классификация электронных писем на спам и не спам. Мы имеем размеченный датасет, где каждое сообщение помечено как спам или не спам (классы). Для обучения модели мы используем этот датасет и обучаем модель на классификацию электронных писем на спам и не спам. Примерами обучения без учителя кластеризация данных. Модель должна кластеризовать данные на несколько кластеров, в которых элементы данных внутри кластеров должны быть похожи друг на друга, а элементы в разных кластерах должны быть разными.

Мы не знаем, какие элементы данных относятся к какому кластеру, поэтому мы не можем дать модели точные ответы. Но можем определить число кластеров, границы и правила для разделения элементов данных на кластеры и оценивать качество кластеризации.

Линейная регрессия является одной из наиболее популярных и простых моделей регрессии, которую мы рассмотрим в данной лекции.

Эта тема актуальна, поскольку многие реальные задачи данных связаны с прогнозированием непрерывных переменных, таких как доход или цена на недвижимость. Поэтому, понимание линейной регрессии является необходимым для успеха в области машинного обучения и анализе данных.

На этой лекции вы узнаете:

- Что такое обучение с учителем.
- Что такое линейная регрессия
- Как считать метрики качества модели линейной регрессии.

Термины, используемые в лекции

Обучение с учителем или контролируемое обучение — подход, при котором машине заранее дают понять, какой ответ будет считаться правильным.

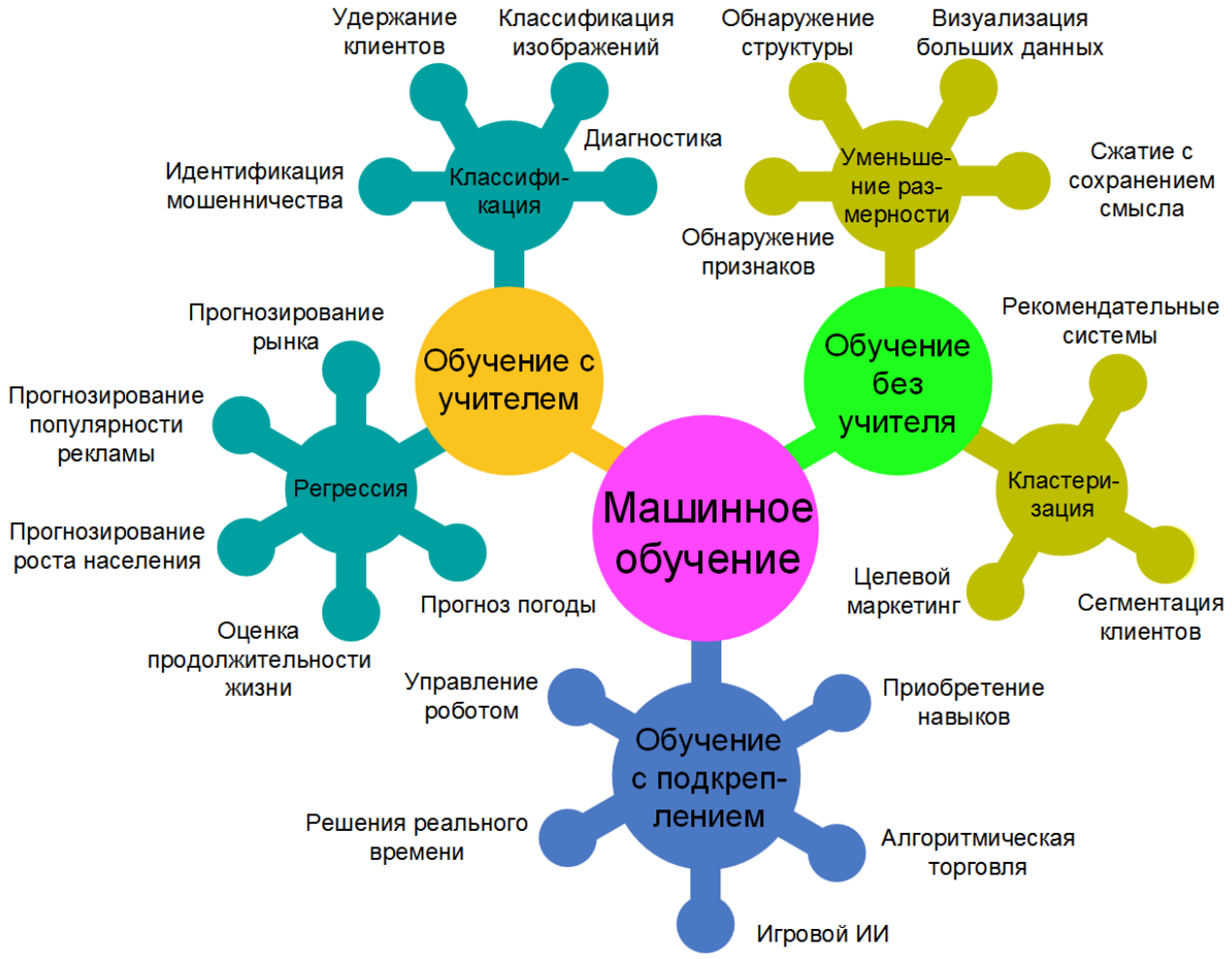
Регрессия – это метод контролируемого машинного обучения, который помогает нам находить корреляцию между переменными и позволяет прогнозировать непрерывные выходные переменные на основе одной или нескольких переменных-предикторов.

Линейная регрессия – задача линейной регрессии заключается в нахождении линии, которая наилучшим образом соответствует данным.

Ранжирование в машинном обучении относится к задаче упорядочивания объектов по их значимости или полезности для конкретной цели. Обычно это происходит с помощью алгоритмов машинного обучения, которые обучаются на обучающей выборке, содержащей пары объектов и их оценок значимости или полезности. В результате обучения алгоритмы могут использоваться для ранжирования новых объектов, которые не входили в обучающую выборку.

Пропуски (отсутствующие значения) - объекты или атрибуты с отсутствующими значениями.

Обучение с учителем



На данной диаграмме представлены некоторые типовые задачи машинного обучения.

Слева на диаграмме отображены задачи классификации. В таких задачах модель обучается относить объекты к одной из заранее заданных категорий. Примеры классификации – определение письма как спам или не спам, определение, принадлежит ли человек к определенной группе по заданным признакам и т.д.

В центре диаграммы представлены задачи регрессии. В регрессионных задачах модель обучается построить функциональную зависимость между входными признаками и выходными значениями. Примеры задач регрессии –

предсказание цены на недвижимость, предсказание количества продаж товара и т.д.

Справа на диаграмме представлены задачи кластеризации. В таких задачах модель обучается группировать объекты на основе их признаков. Примеры кластеризации – группировка пользователей по их предпочтениям, группировка изображений по схожести и т.д.

Каждая из этих задач машинного обучения имеет свои специфические методы и алгоритмы для решения, и диаграмма позволяет увидеть общую классификацию их по типу.

Задачи машинного обучения имеют много общего.

Во-первых, их решения можно описать как функции, которые отображают объекты или примеры (samples) в предсказании (target). Например, пациенты должны быть сопоставлены с диагнозами, а документы - с оценками релевантности.

Во-вторых, эти задачи вряд ли имеют единственное идеальное решение. Даже профессиональные переводчики могут по-разному перевести одно и то же предложение, и оба перевода будут правильными. Другими словами, лучшее в решении этих задач — враг хорошего.

В-третьих, существует множество примеров правильных ответов (например, переводы текстов на другие языки или подписи к заданной картинке), а примеры неправильных ответов (при необходимости) обычно легко построить. Функция, которая сопоставляет объекты с предсказаниями, называется **моделью**, а набор доступных примеров — **обучающей выборкой или набором данных**. Обучающая выборка состоит из объектов (например, фотографий, загруженных из Интернета, истории болезни пациента, активности пользователя услуг и т.д.) и ответов (например, подписи к фотографиям, диагнозы, информация о пользователях, покидающих сервис), которые иногда называют таргетами.

Задача обучения с учителем

Приведенная выше описание является примером задачи контролируемого обучения или обучения с учителем, поскольку правильный ответ для каждого объекта в обучающем множестве известен заранее. В зависимости от содержания обучающего множества задачи контролируемого обучения могут быть следующих типов:

Задача регрессии – предсказания вещественного значения: примерами задач регрессии является предсказание продолжительности поездки на каршеринге, спрос на конкретный товар в конкретный день или погода в вашем городе на завтра (температура, влажность и давление — это несколько вещественных чисел, которые формируют вектор нашего предсказания).

Задача классификации – предсказания категориального ответа (метки класса) с конечным количеством вариантов: например, мы можем предсказывать, кликнет ли пользователь по рекламному объявлению, вернёт ли клиент кредит в установленный срок, сдаст ли студент сессию, случится ли определенное заболевание у пациента, есть ли на картинке банан.

Ранжирование Основным примером является задача ранжирования в поисковой системе. Здесь для любого заданного запроса все возможные документы должны быть ранжированы в соответствии с их релевантностью запросу. Оценка релевантности имеет смысл только в контексте сравнения двух документов друг с другом; ее абсолютное значение не имеет смысла.

Задание: Определите тип задачи: Предсказание курса евро к доллару на следующий день.

Ответ: Это задача регрессии. Модель предсказывает вещественное число

Критерии качества.

Мы хотим построить модель с достаточно высокими прогностическими значениями на основе обучающего примера. Что мы подразумеваем под достаточно высокими прогностическими значениями? Необходимо внимательно отнестись к выбору показателей качества, поскольку невозможно предоставить хорошее решение, не понимая, чего вы хотите добиться.

Машинное обучение начинается с данных. Для этого важно, чтобы данных было много и они были достаточно качественными. Некоторые проекты приходится откладывать на неопределенный срок из-за невозможности собрать данные.

Чем сложнее задача, тем больше данных необходимо для ее решения.
Однако важен не только объем данных, но и то, насколько эти данные пригодны для анализа. Давайте рассмотрим, что это значит и какие проблемы возникают в связи с этим.

Чтобы работать с объектами, модель должна основываться на некоторых их свойствах. Например, доход человека, цвет левого верхнего пикселя на изображении или частота употребления слова "интеграл" в тексте. Эти свойства

часто называют **признаками**, а набор свойств, извлеченных из объекта — **описанием этого свойства**.

Вот некоторые простые и распространенные типы свойств

- Числовые — например, рост, доход и т.д. Можно выделить вещественные и целочисленные атрибуты.

- Категориальные атрибуты принимают значения из дискретного набора. Например, профессия человека или день недели.

- Бинарные атрибуты принимают два значения: 0 и 1 или "да" и "нет". Его можно рассматривать и как числовой, и как категориальный атрибут.

- Категориальный атрибут иногда также называют порядковым атрибутом. Этот атрибут принимает значения из упорядоченного дискретного множества. Например, класс опасности химического вещества (1-4) или продолжительность обучения студента в магистратуре являются порядковыми атрибутами.

Необходимо также рассмотреть более сложные признаки. Например, описание ресторана может содержать текст и фотографии, или профиль человека в социальной сети может содержать список друзей. Было разработано множество методов извлечения признаков для многих подобных типов данных, включая изображения, видео, тексты, звуки и графику. В настоящее время это в основном методы нейронных сетей. Если встречаются более сложные данные, может потребоваться больше усилий для извлечения из них признаков. Этот процесс известен как **feature engineering.** Удобно бывает записать данные в виде таблицы, строки которой соответствуют объектам, а столбцы – признакам.

Создание информативных описаний признаков имеет решающее значение для дальнейшего анализа. Однако следует также обратить внимание на качество полученных данных. Например, могут возникнуть следующие проблемы

Пропуски (отсутствующие значения). Объекты или атрибуты с отсутствующими значениями могут быть пропущены из выборки, но если отсутствующих значений слишком много, может быть потеряно много информации. Кроме того, наличие пропусков может само по себе содержать информацию. Например, они могут указывать на систематические проблемы при сборе данных

для определенного сегмента выборки. Некоторые модели имеют собственные методы работы с выбросами, например, решающие деревья, в то время как другие, такие как линейные модели и нейронные сети, должны удалять выбросы или заменять их чем-то другим.

Выбросы — это объекты, которые значительно отличается от других. Например, в наборе данных, содержащем информацию о клиентах банка, 140-летний человек явно будет очень нетипичным. Выбросы могут быть результатом ошибок при сборе данных, но также могут представлять собой реальные выбросы. Отклонение от нормы обычно лучше устранить, но в некоторых случаях отклонения могут быть важными сущностями (например, очень богатые клиенты банка), и в этом случае может быть полезно зафиксировать отклонения и рассматривать их отдельно.

Ошибка разметки. Например, если вы собираете данные с помощью людей-разметчиков, вы должны быть готовы к тому, что некоторые из ваших таргетов будут обозначены неправильно.

Дрейф данных. Данные могут меняться с течением времени. Например, может измениться способ сбора данных, и данные могут поступать в формате, который модель никогда не обрабатывала. Распределение данных также может измениться, например, если вы предоставляете образовательные услуги для студентов, но теперь к вам приходят более зрелые люди. Дрейф данных - суровая реальность для систем, которые не могут решать проблемы немедленно, поэтому необходимо следить за распределением данных и при необходимости обновлять модель.

Существуют и другие проблемы. Нередко значительная часть данных отбрасывается, потому что что-то сломалось в процессе сбора или потому что система регистрации сервиса была изменена полгода назад, что делает невозможным объединение старых данных с новыми.

Линейная регрессия

Вторую часть нашей лекции начнем с самой простой и понятной модели машинного обучения: линейной модели. Мы рассмотрим, что такое линейные модели, почему они работают и когда их следует использовать, и как работает машинное обучение на относительно простых примерах.

Представьте, что у вас много объектов и вы хотите присвоить значение каждому объекту. Например, предположим, у вас есть набор транзакций по банковским картам, и вы хотите понять, какие из этих транзакций были совершены мошенниками. Если вы разделите все транзакции на два класса, где 0 означает законное действие, а 1 - мошенническое, то это будет простейшая задача классификации. Рассмотрим другую ситуацию. У вас есть данные геологоразведки, и вы хотите оценить вероятности различных вкладов. В этом случае модель будет предсказывать, например, годовую потенциальную прибыльность шахты на основе набора геологических данных. Это пример регрессионной задачи. Числа, которым мы хотим сопоставить объекты из нашего множества иногда называют таргетами (от английского target).

Другими словами, проблемы классификации и регрессии можно сформулировать как поиск соответствия из набора данных к набору возможных таргетов.

Очевидно, что просто сопоставлять некоторые объекты с некоторыми числами довольно бессмысленно. Мы хотим быстро обнаружить мошенников или решить, где строить шахты. Именно здесь необходимы определенные критерии качества. Мы хотим найти наиболее близкое к реальному соответствию между объектом и целью. Но что означает "наилучшее" - это сложный вопрос. Мы будем возвращаться к этому вопросу много раз. Но есть и более простой вопрос. Существует множество возможных сопоставлений, но давайте упростим задачу и скажем, что мы хотим найти решение только среди заранее определенного набора параметрических функций. В этом разделе мы сосредоточим все наши усилия на простейших семействах функций, а именно на линейных функциях следующего вида

$$y = w_1 x_1 + \ldots + w_D x_D + w_0,$$

где *y* - целевая переменная (таргет),

x1, ……xD - вектор, соответствующий объекту выборки (вектор признаков),

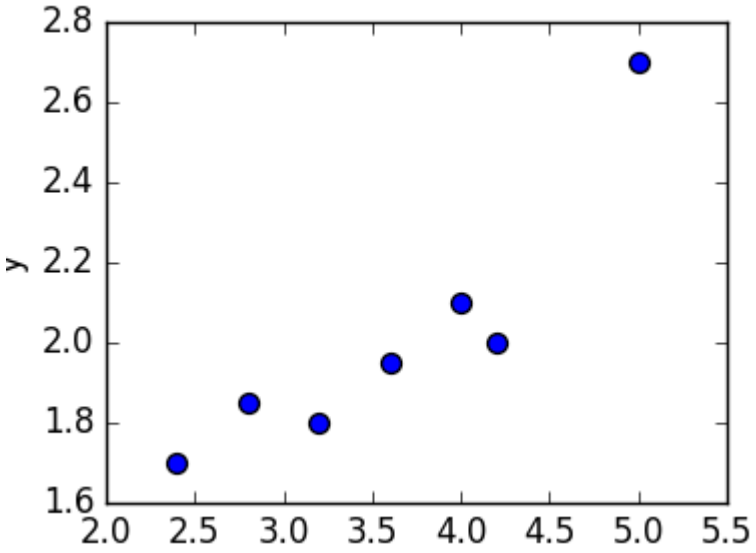
w1,……wD - – параметры модели

Признаки еще называют фичами (от английского features), Векторы *w* часто называют вектором весов. Поскольку прогнозы модели можно рассматривать как взвешенную сумму характеристик объекта, число *w0* - является свободным коэффициентом или смещением (bias).

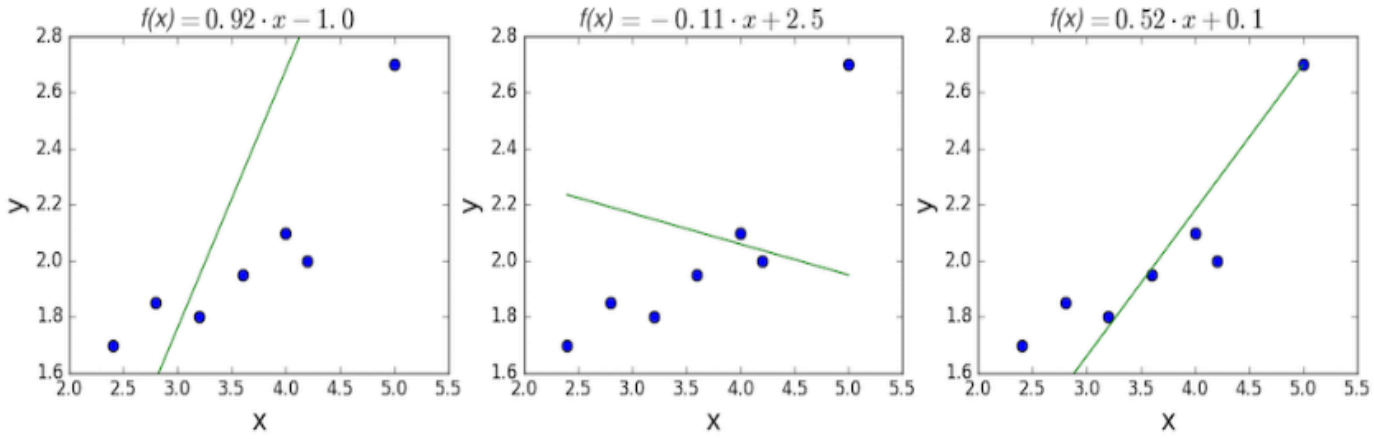
Для начала рассмотрим решение задачи регрессии с использованием линейной модели. Простейшим примером постановки задачи линейной регрессии является метод наименьших квадратов.

Линейная регрессия — это зависимость между переменной x и одной или несколькими другими переменными, моделируемая линейной функцией зависимости.

Предположим, нам задан набор из 7 точек



Цель линейной регрессии — найти линию, которая лучше всего подходит к этим точкам. Напомним, что общее уравнение для прямой линии - f (x) = m⋅x + b, где m - наклон линии, a b - смещение y. Таким образом, решение линейной регрессии определяет значения m и b так, чтобы f (x) была как можно ближе к y. Давайте попробуем несколько случайных кандидатов:



Совершенно очевидно, что первые две линии не соответствуют нашим данным; третья линия кажется лучше двух других. Как же мы можем это проверить? Формально нам нужно выразить, насколько хорошо подходят линии, и мы можем сделать это, определив функцию потерь.

Регрессионная модель — это функция, которая принимает на вход значения атрибутов данного объекта и выдает на выходе ожидаемое значение целевой переменной. В большинстве случаев предполагается только одна целевая

переменная. Если ставится задача предсказать несколько атрибутов, то они обычно рассматриваются как несколько независимых задач регрессии по одному и тому же атрибуту.

Мы еще ничего не сказали о том, как внутренне устроена регрессионная модель, она может быть какой угодно. Это может быть математическая формула, условный алгоритм, сложная программа с множеством ветвей и циклов или нейронная сеть — все это может быть представлено регрессионной моделью.

Единственным требованием к моделям машинного обучения является их параметричность. Это означает, что должны существовать внутренние параметры, от которых также зависят результаты вычислений. В простом случае наиболее часто используемые регрессионные модели — это аналитические функции. Таких функций бесчисленное множество, но часто используется самая простая функция — линейная, и мы начинаем исследование регрессии именно с нее.

Регрессионные модели делятся на парную регрессию и множественную регрессию. Парная регрессия используется, когда есть только один признак. Множественная регрессия используется, когда имеется более одного признака. Конечно, парная регрессия нечасто встречается на практике, но эти простые модели можно использовать в качестве примеров для понимания основных концепций машинного обучения.

Кроме того, парная регрессия может быть представлена графически очень удобным и наглядным способом: Когда переменных больше двух, графики строить немного сложнее, и модель нужно визуализировать другим способом.

🔥 Для реализации линейной модели каждый объект уже должен быть представлен вектором числовых признаков. Конечно, текст или графику нельзя поместить непосредственно в линейную модель, но сначала необходимо придумать их числовые свойства.

Напомним, что в задачах регрессии пытаются получить достоверные значения целевой переменной, используя входные переменные. Любая функция, даже самая простая линейная функция, может давать очень разные значения для одних и тех же входных данных, если функция имеет разные параметры. Поэтому любая регрессионная модель — это не конкретная математическая функция, а набор функций. Задача алгоритма обучения состоит в том, чтобы подобрать значения параметров так, чтобы предсказанные значения (или теоретические

значения, значения, рассчитанные по модели) были как можно ближе к набору данных.

Итак, что нам делать в том случае, когда перед нами нет линейной зависимости, или закон поведения данных невозможно отразить с помощью линейной функции, нам нужна более сложная модель.

Полиномиальная регрессия может понадобиться в следующих случаях:

1. Когда взаимосвязь между зависимой и независимой переменными не линейная, а имеет кривую форму.
2. Когда простая линейная регрессия не может точно предсказать значения зависимой переменной.
3. Когда данные содержат выбросы или неоднородное распределение, и полиномиальная регрессия может дать более точные и устойчивые оценки.
4. Когда необходимо учесть взаимодействие и влияние нескольких независимых переменных на зависимую переменную.
5. В некоторых областях, таких как физика или инженерия, полиномиальная регрессия может быть применена для моделирования физических закономерностей или физических процессов.

В целом, полиномиальная регрессия используется, когда простые линейные модели недостаточно гибки и не могут адекватно описать взаимосвязь между переменными.

Полиномиальная регрессия — это метод машинного обучения, используемый для поиска нелинейных зависимостей между переменными. Он является расширением линейной регрессии, где модель является многочленом n-ой степени.

В общем случае модель полиномиальной регрессии выглядит следующим образом:

$$y = \beta 0 + \beta 1x + \beta 2x2 + \beta 3x3 + ... + \beta nxn + \varepsilon,$$

где y — зависимая переменная, x — независимая переменная, βi — коэффициенты, нужные для вычисления предсказанных значений, xn — степень x, а ε — ошибка.

Идея полиномиальной регрессии заключается в том, чтобы попытаться аппроксимировать данные многочленом наилучшей степени (градиентный спуск, метод наименьших квадратов и т. д.). В зависимости от формы данных, модель может быть квадратичной (n=2), кубической (n=3), квартиномиальной (n=4) и т. д.

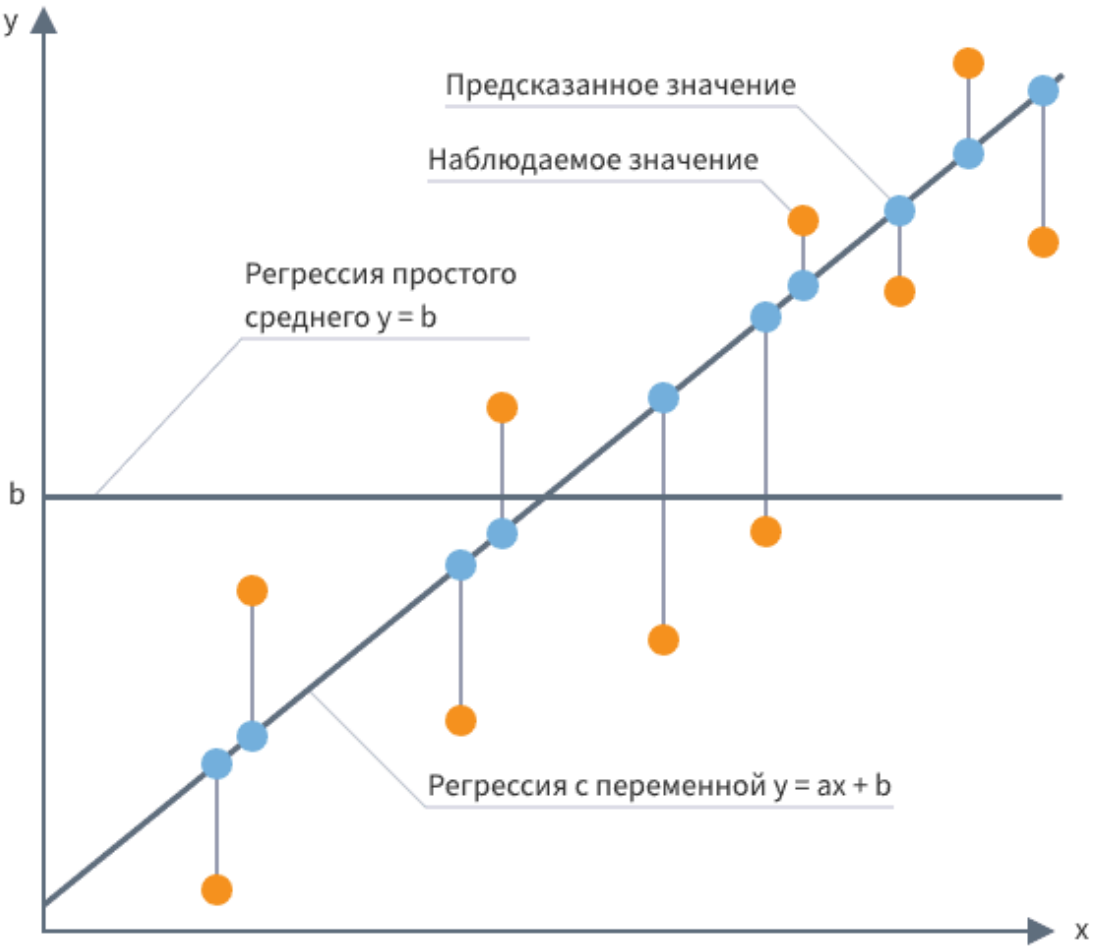
Иными словами, вместо прямой линии, используемой в обычной линейной регрессии, полиномиальная регрессия использует кривую или полином для описания данных. Это позволяет нам лучше описывать сложные отношения и предсказывать значения зависимой переменной на основе значений независимых переменных.

При использовании полиномиальной регрессии необходимо быть осторожными, чтобы избежать переобучения модели. Чем выше степень многочлена, тем больше свободных параметров модели вы имеете, и тем более точно вы можете аппроксимировать данные. В то же время, чем больше параметров, тем больше вероятность переобучения. Следовательно, необходимо проанализировать преимущества и недостатки при выборе определённой модели.

💡 “Хорошая” аналитическая модель должна удовлетворять двум зачастую противоречивым требованиям. То есть, она должна быть удобной для интерпретации и при этом как можно лучше соответствовать данным. На самом деле, увеличение соответствия данным часто связано с увеличением сложности модели (в случае регрессии - количества входных переменных для модели). Чем сложнее модель, тем ниже ее интерпретируемость.

Метрики линейной регрессии

Задача регрессии – предсказания вещественного значения. Это задача обучения с учителем, значит, у нас есть правильные ответы (истинные значения), с которыми можно сравнить предсказанные моделью.



Наклонная линия представляет собой линию регрессии с переменной, имеющей точки, соответствующие прогнозируемому значению выходной переменной (кружки синего цвета). Оранжевые кружки представляют фактические (наблюдаемые) значения *y*

MSE

Среднеквадратичная ошибка (Mean Squared Error) используется, когда вы хотите подчеркнуть большие ошибки и хотите выбрать модель с точно меньшим количеством больших ошибок. Большие значения ошибок подчеркиваются из-за квадратичной зависимости.

Действительно, предположим, что в двух случаях модель выдает 5 и 10 ошибок. По абсолютной величине это разница в два раза, но если возвести в квадрат 25 и 100, то разница будет четырехкратной. Следовательно, меньшее значение

MSE может точно уменьшить большую ошибку на большую величину.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

Таким образом, можно сделать вывод, что MSE корректируется для точного отражения влияния больших ошибок на качество модели.

Недостатки использования MSE является то, что если один или несколько случаев ошибок (возможно, включая выбросы) приводят к большим ошибкам, то их возведение в квадрат приводит к ошибочному выводу о низкой эффективности модели в целом. С другой стороны, если модель дает небольшие ошибки во многих случаях, это может иметь обратный эффект, т.е. недооценить слабость модели.

RMSE

Корень из среднеквадратичной ошибки (Root Mean Squared Error) вычисляется просто как квадратный корень из MSE:

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

Влияние каждой ошибки на RMSE пропорционально размеру квадрата ошибки. Поэтому большие ошибки оказывают непропорционально большое влияние на RMSE. В результате RMSE можно считать чувствительным к выбросам.

MAE

Средняя абсолютная ошибка (Mean Absolute Error) вычисляется следующим образом:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

То есть, MAE рассчитывается как среднее абсолютных значений разницы между наблюдаемыми и предсказанными значениями; в отличие от MSE и RMSE, это линейная оценка, поэтому все ошибки в среднем имеют одинаковый вес. Например, разница между 0 и 10 в два раза больше разницы между 0 и 5. Это не так в случае с MSE и RMSE, как обсуждалось выше.

MAPE

Средняя абсолютная процентная ошибка (Mean Absolute Percentage Error) вычисляется следующим образом:

$$MAPE = \frac{100}{n} \sum_{i=1}^{n} \frac{|y_i - \widehat{y_i}|}{|y_i|}$$

Эта ошибка не имеет размерности и очень легко интерпретируется. Она может быть выражена в виде дроби или процента. Например, MAPE=11,4 означает, что ошибка составляет 11,4% от истинного значения.

R-квадрат

Приведенные выше ошибки нелегко интерпретировать. Действительно, зная, что средняя абсолютная ошибка составляет, например, 10, мы не можем сразу сказать, хорошая это ошибка или плохая, и что мы можем сделать для улучшения модели.

В этом контексте было бы интересно оценить качество регрессионной модели не по величине ошибки, а по тому, насколько хорошо она работает по сравнению с моделью с одной константой и без входных переменных или с коэффициентом регрессии, равным нулю.

Это как раз и есть показатель коэффициента детерминации, который указывает на долю дисперсии зависимой переменной, объясненную регрессионной моделью. Наиболее распространенные формулы, используемые для расчета коэффициента детерминации, следующие:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\overline{y}_i - y_i)^2}$$

На практике числитель этого уравнения — стандартная ошибка оцениваемой модели, а знаменатель — модель с одной лишь константой.

Основное преимущество коэффициента детерминации перед мерами, основанными на ошибках, заключается в том, что он инвариантен по отношению к масштабу данных. Кроме того, он всегда находится в диапазоне от -∞ до 1. Значения, близкие к 1, указывают на то, что модель хорошо согласуется с данными. Это очевидно, когда коэффициент в уравнении близок к нулю, то есть когда ошибка модели с переменными намного меньше, чем ошибка модели с постоянными.

Если значение коэффициента близко к нулю (то есть ошибка модели с переменными примерно равна ошибке модели только с константами), это говорит, что модель плохо подходит к данным, и модель с переменными работает хуже, чем модель с постоянными.

Кроме того, коэффициент принимает отрицательное значение (обычно небольшое). Это происходит, когда ошибка средней модели меньше, чем ошибка модели с переменными. В этом случае видно, что добавление некоторых переменных в модель с постоянными сделает модель хуже (т.е. регрессионные модели с переменными работают хуже, чем прогнозирование с простыми средними).

Скорректированный R-квадрат

Основная проблема с использованием коэффициента детерминации заключается в том, что при добавлении в модель новой переменной коэффициент детерминации увеличивается (или, по крайней мере, не уменьшается), даже если эта переменная вообще не связана с зависимой переменной.

С этим связаны две проблемы. Первая заключается в том, что не все переменные, добавленные в модель, делают точность модели значимой и всегда увеличивают ее сложность. Вторая проблема заключается в том, что коэффициент детерминации нельзя использовать для сравнения моделей с разным количеством переменных. Для преодоления этих проблем используются альтернативные показатели, одним из которых является скорректированный коэффициент детерминации.

$$R^2_{adj} = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2/(n-k)}{\sum_{i=1}^{n}(\overline{y}_i - y_i)^2/(n-1)}$$

Скорректированный коэффициент детерминации всегда меньше единицы, но теоретически он может быть меньше нуля только в том случае, если нормальный коэффициент детерминации очень мал, а количество переменных в модели велико.

Сегодня мы познакомились с понятием обучения с учителем. Узнали, какие задачи решают алгоритмы данного типа. Узнали самую первую, наиболее простую модель Линейной регрессии. Понимание линейной регрессии является ключом к пониманию более сложных моделей, вплоть до глубоких нейронных сетей.

Давайте посмотрим, как мы можем использовать модель Линейной регрессии:
https://colab.research.google.com/drive/18-8n_FcXPmmEU991GfTcQegq002QuIsi?usp=sharing

Итоги урока:

Понимание работы модели Линейной регрессии поможет в дальнейшем обучении
Модель линейной регрессии широко используется в экономике, финансах, маркетинге, социологии и других областях, где необходимо оценивать взаимосвязи между переменными.

Например, модель линейной регрессии может использоваться для:

1. Прогнозирования: Модель можно использовать для прогнозирования будущих значений зависимой переменной на основе известных значений независимых переменных. Например, модель может использоваться для прогнозирования выпуска продукции на следующий год на основе данных о производственных мощностях и рекламных затратах.

2. Оценки эффекта: Модель может использоваться для оценки эффекта изменения одной или нескольких независимых переменных на зависимую переменную. Например, модель может использоваться для оценки того, насколько изменятся продажи товара, если будет увеличена его цена.

3. Идентификации факторов, влияющих на зависимую переменную: Модель может использоваться для выявления факторов, которые влияют на зависимую переменную, и определения, насколько сильно каждый фактор влияет на нее. Например, модель может использоваться для идентификации тех факторов, которые наиболее сильно влияют на социальную мобильность населения.

4. Оптимизации: Модель может использоваться для оптимизации различных процессов и операций. Например, модель можно использовать для оптимизации производства, путем определения оптимальных значений параметров производства на основе их влияния на выход продукции.

Что можно почитать еще?

1. Базовые принципы машинного обучения на примере линейной регрессии
2. Введение в алгоритмы машинного обучения: линейная регрессия
3. Обучение нейросети с учителем, без учителя, с подкреплением — в чем отличие? Какой алгоритм лучше?

Используемая литература

1. Регрессия как задача машинного обучения
2. Линейная регрессия в машинном обучении
3. Метрики качества линейных регрессионных моделей