دانشگاه ولی عصر (عج)
رفسنجان

نام پروژه : کاربرد آمار در یادگیری ماشین

نام و نام خانوادگی دانشجو : محمد حسین زارعی

استاد راهنما: دکتر عباس مهدوی

پاییز ۱۴۰۴

فهرست مطالب

۱ مقدمه **۲**

۲ بررسی عملکرد مدل **۳**
۱.۲ اورفیتنگ(overfiting) ۳
۲.۲ اندرفیتینگ (Underfitting) ۴
۳.۲ فیتینگ خوب (Good Fitting) ۵

۳ انواع یادگیری **۶**
۱.۳ رگرسیون در یادگیری نظارت شده ۷
۱.۱.۳ رگرسیون خطی ساده ۷
۲.۱.۳ رگرسیون خطی چند متغیره ۱۰
۳.۱.۳ گرادیان(gradeint) ۱۴
۴.۱.۳ رگرسیون لجستیک ۱۶
۵.۱.۳ نظم‌دهی (Regularization) ۱۸
۶.۱.۳ رگرسیون لاسو ۱۸
۷.۱.۳ رگرسیون ریج ۲۰
۲.۳ یادگیری بدون نظارت ۲۰
۱.۲.۳ K-means ۲۱

۱ مقدمه

یادگیری ماشین شاخه‌ای از هوش مصنوعی است که به ماشین‌ها امکان می‌دهد از داده‌ها یاد بگیرند و رفتارهای هوشمندانه‌ای مشابه انسان‌ها بروز دهند، بدون اینکه به‌صورت صریح برنامه‌ریزی شوند. این فناوری به ماشین‌ها اجازه می‌دهد با استفاده از الگوریتم‌های یادگیری، الگوها و روابط موجود در داده‌ها را شناسایی کرده و بر اساس آن‌ها تصمیم‌گیری کنند یا به محیط اطراف خود تأثیر بگذارند.
در یک نگاه ساده، یادگیری ماشین می‌تواند به دو صورت عمل کند: در روش سنتی، برنامه‌نویسان با استفاده از عملیات منطقی و ریاضی، برنامه‌ای مشخص را برای ماشین طراحی می‌کنند. برای مثال، در مدیریت موجودی کارخانه، برنامه‌نویس با دریافت فرمول مقدار اقتصادی سفارش، برنامه نویسی می کند. که با تحلیل داده‌های موجودی و پارامترهای محیطی، مقدار بهینه سفارش را محاسبه و اعلام می‌کند. اما در رویکرد مدرن یادگیری ماشین، به جای ارائه برنامه‌های از پیش تعیین‌شده، ماشین با قرار گرفتن در معرض داده‌ها و بهره‌گیری از الگوریتم‌های یادگیری، به‌تدریج مدل‌های مورد نیاز خود را شکل می‌دهد. این فرایند مشابه یادگیری انسان‌ها، مانند کودکی است که به‌تدریج زبان را می‌آموزد و توانایی صحبت کردن را کسب می‌کند.
یادگیری ماشین (Machine Learning) در سال‌های اخیر پیشرفت‌های چشمگیری داشته و در حوزه‌های مختلف کاربردهای پیشرفته‌ای پیدا کرده است. در ادامه، به برخی از مثال‌های پیشرفته و نوآورانه یادگیری ماشین اشاره می‌کنم که نشان‌دهنده قدرت و تنوع این فناوری هستند:

۱. تشخیص و درمان پزشکی پیشرفته: - تشخیص سرطان با دقت بالا: الگوریتم‌های یادگیری عمیق (Deep Learning) مانند شبکه‌های کانولوشنی (CNN) برای تحلیل تصاویر پزشکی (MRI، CT، و ماموگرافی) استفاده می‌شوند. برای مثال، مدل‌های AI مانند Google Health's DeepMind در تشخیص سرطان پستان از تصاویر ماموگرافی، گاهی دقت بیشتری نسبت به رادیولوژیست‌های انسانی دارند. - **پیش‌بینی بیماری‌های ژنتیکی**: ابزارهایی مثل AlphaFold DeepMind ساختار پروتئین‌ها را با دقت بی‌سابقه‌ای پیش‌بینی می‌کنند که در توسعه داروهای جدید برای بیماری‌های پیچیده مانند آلزایمر یا سرطان نقش مهمی دارند.

۲. پردازش زبان طبیعی (NLP): مدل‌های زبانی پیشرفته: مدل‌هایی مانند GPT-4 یا LLaMA با تحلیل حجم عظیمی از داده‌های متنی، توانایی تولید متن‌های شبیه انسان، پاسخ به سؤالات پیچیده، ترجمه زبان‌ها، و حتی نوشتن کد را دارند. این مدل‌ها در چت‌بات‌ها، دستیارهای مجازی، و تحلیل احساسات (Sentiment Analysis) کاربرد دارند. - ترجمه بلادرنگ: سیستم‌هایی مانند Google Translate با استفاده از یادگیری ماشین، ترجمه‌های صوتی و متنی را به‌صورت بلادرنگ با دقت بالا انجام می‌دهند.

۳. بینایی کامپیوتری: -تشخیص اشیا و چهره: الگوریتم‌های یادگیری عمیق در سیستم‌های امنیتی برای شناسایی چهره‌ها یا تشخیص اشیا در تصاویر و ویدئوها استفاده می‌شوند. مثلاً، سیستم‌های نظارت شهری از این فناوری برای شناسایی رفتارهای مشکوک بهره می‌برند. - خودروهای خودران: شرکت‌هایی مثل Tesla و Waymo از یادگیری ماشین برای پردازش داده‌های حسگرها (LiDAR، دوربین‌ها) استفاده می‌کنند تا خودروها بتوانند موانع، علائم راهنمایی، و مسیرها را شناسایی کرده و رانندگی ایمن انجام دهند.

۴. یادگیری تقویتی در رباتیک: - ربات‌های خودآموز: الگوریتم‌های یادگیری تقویتی (Reinforcement Learning) به ربات‌ها امکان می‌دهند تا از طریق آزمون‌وخطا وظایف پیچیده‌ای مانند راه رفتن، گرفتن اشیا، یا حتی انجام جراحی را یاد بگیرند. برای مثال، ربات‌های Boston Dynamics از این فناوری برای حرکات پویا و تعادل استفاده می‌کنند. - **بازی‌های استراتژیک**: مدل‌هایی مثل AlphaGo از DeepMind4 با استفاده از یادگیری تقویتی، استراتژی‌های پیچیده‌ای را در بازی‌های تخته‌ای مانند Go توسعه داده‌اند که حتی قهرمانان انسانی را شکست داده‌اند.

۵. توصیه‌گرهای هوشمند: - سیستم‌های پیشنهاددهنده: پلتفرم‌هایی مانند Netflix، Spotify، و Amazon از الگوریتم‌های یادگیری ماشین (مانند Collaborative Filtering و یادگیری عمیق) برای تحلیل رفتار کاربران و پیشنهاد محتوا یا محصولات شخصی‌سازی‌شده استفاده می‌کنند. - تبلیغات هدفمند: شرکت‌های تبلیغاتی از یادگیری ماشین برای تحلیل داده‌های کاربران و ارائه تبلیغات متناسب با علایق آن‌ها بهره می‌برند.

۶. پیش‌بینی و تحلیل داده‌های کلان: - پیش‌بینی بازارهای مالی: الگوریتم‌های یادگیری ماشین برای تحلیل روندهای بازار، پیش‌بینی قیمت سهام، و مدیریت ریسک در مؤسسات مالی استفاده می‌شوند. - **مدیریت زنجیره تأمین**: شرکت‌هایی مانند Walmart از یادگیری ماشین برای بهینه‌سازی موجودی، پیش‌بینی تقاضا، و مدیریت لجستیک استفاده می‌کنند.

۷. کشف دارو و زیست‌فناوری: - طراحی مولکول‌های جدید: AI با شبیه‌سازی تعاملات شیمیایی، ترکیبات دارویی جدید را پیشنهاد می‌دهد. برای مثال، شرکت Insilico Medicine از یادگیری ماشین برای کشف داروهای جدید در زمان کوتاه‌تر استفاده کرده است. - تحلیل داده‌های ژنومی: الگوریتم‌ها برای شناسایی جهش‌های ژنتیکی و پیش‌بینی پاسخ بیماران به درمان‌های خاص استفاده می‌شوند.

۸. کشاورزی هوشمند: - کشاورزی دقیق: یادگیری ماشین با تحلیل داده‌های حسگرها، تصاویر ماهواره‌ای، و داده‌های آب‌وهوایی، به کشاورزان کمک می‌کند تا زمان مناسب کاشت، آبیاری، و برداشت را تعیین کنند. مثلاً، سیستم‌های AI می‌توانند آفات یا بیماری‌های گیاهی را زودهنگام تشخیص دهند. - اتوماسیون مزارع: ربات‌های مجهز به AI برای کاشت، برداشت، و حتی جمع‌آوری داده‌های خاک استفاده می‌شوند.

۹. امنیت سایبری: - تشخیص تهدیدات: الگوریتم‌های یادگیری ماشین برای شناسایی حملات سایبری، بدافزارها، و رفتارهای غیرعادی در شبکه‌ها استفاده می‌شوند. مثلاً، سیستم‌های تشخیص نفوذ (IDS) از AI برای تحلیل ترافیک شبکه بهره می‌برند. - احراز هویت بیومتریک: AI در سیستم‌های تشخیص صدا یا اثر انگشت برای افزایش امنیت کاربرد دارد.

۱۰. هنر و خلاقیت: - تولید محتوا: ابزارهایی مانند DALL-E یا MidJourney با استفاده از یادگیری عمیق، تصاویر، موسیقی، یا حتی داستان‌هایی خلاقانه تولید می‌کنند که مشابه آثار انسانی هستند. - تحلیل سبک‌های هنری: AI می‌تواند آثار هنری را تحلیل کرده و سبک‌های جدید خلق کند یا حتی در مرمت آثار باستانی کمک کند.

ویژگی‌های پیشرفته این کاربردها: - یادگیری عمیق: استفاده از شبکه‌های عصبی عمیق با لایه‌های متعدد برای پردازش داده‌های پیچیده. - تحلیل داده‌های کلان: توانایی پردازش حجم عظیمی از داده‌ها در زمان کوتاه. - تطبیق‌پذیری: قابلیت یادگیری و بهبود مستمر با داده‌های جدید. -تعامل چندحوزه‌ای: ترکیب یادگیری ماشین با حسگرها، رباتیک، و اینترنت اشیا (IoT).

این مثال‌ها تنها بخشی از پتانسیل یادگیری ماشین را نشان می‌دهند. با پیشرفت الگوریتم‌ها و افزایش قدرت محاسباتی، انتظار می‌رود که کاربردهای پیشرفته‌تری در آینده ظهور کنند که زندگی بشر را بیش از پیش متحول کنند.

در یادگیری ماشین، به ویژه در مدل‌های نظارت‌شده مانند رگرسیون خطی، مفاهیم Overfitting (اورفیتینگ)، Underfitting (اندرفیتینگ) و Good Fitting (فیتینگ خوب) نقش کلیدی در ارزیابی و بهبود عملکرد مدل ایفا می‌کنند. این مفاهیم بخشی از مفهوم کلی Bias-Variance Tradeoff هستند و نشان‌دهنده تعادل بین پیچیدگی مدل و تعمیم‌پذیری آن می‌باشند. در ادامه، تعاریف رسمی هر کدام را بررسی می‌کنیم.

۲ بررسی عملکرد مدل

۱.۲ اورفیتنگ(overfiting)

Overfitting یکی از چالش‌های اصلی در مدل‌های یادگیری ماشین (به ویژه در یادگیری نظارت‌شده مثل رگرسیون خطی) است. به زبان ساده، Overfitting زمانی رخ می‌دهد که مدل بیش از حد به جزئیات و نویزهای داده‌های آموزشی "چسبیده" و الگوهای واقعی را نمی‌تواند به داده‌های جدید تعمیم دهد. یعنی مدل روی داده‌های آموزشی عالی عمل می‌کند (دقت بالا)، اما وقتی داده‌های تست (جدید) می‌آید، عملکردش افت می‌کند مثل دانش‌آموزی که فقط کتاب درسی را حفظ می‌کند، اما سؤال‌های جدید را نمی‌تواند حل کند.

بررسی دقیق تر :

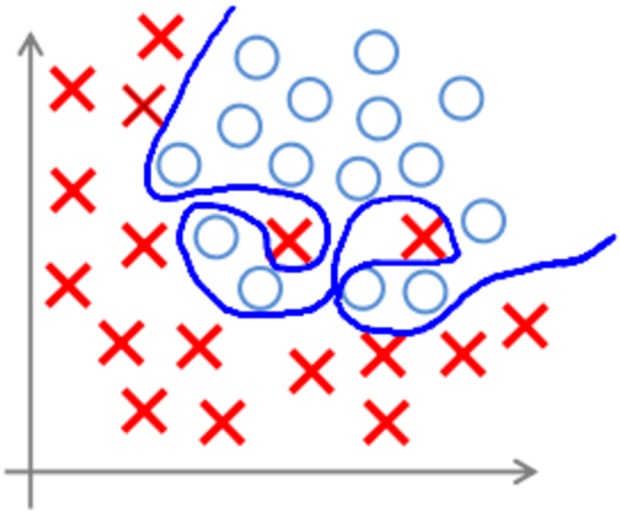
- **چه زمانی این اتفاق رخ می دهد؟** مدل پیچیده (مثل رگرسیون با ویژگی‌های زیاد یا درخت تصمیم عمیق) سعی می‌کند همه نقاط‌های آموزشی را "کامل" فیت کند، حتی نویزها (outliers) یا الگوهای تصادفی. این باعث می‌شود مدل "یاد بگیرد" اما "درک نکند".

- **نشانه‌ها**:

  - دقت روی داده‌های آموزشی (Train Accuracy) بالاست (مثل ٪۹۹)، اما روی داده‌های تست (Test Accuracy) پایین است (مثل ٪۷۰).
  - منحنی یادگیری: خط Train Error کم است، اما Test Error بالاست.

- **ارتباط با آمار**: در رگرسیون خطی، اگر مدل بیش از حد به داده‌های آموزشی وابسته شود، MSE روی Train کم می‌شود اما روی Test زیاد. آمار این را با مفهوم "واریانس بالا" (High Variance) توضیح می‌دهد.

مثال عملی فرض کنید می‌خواهیم قیمت خانه را بر اساس مساحت پیش‌بینی کنیم: با مدل اورفیت، دقت Train: ٪۹۸، اما برای خانه جدید پیش‌بینی عجیبی می‌کند (دقت Test: ٪۶۰).
شکل زیر نمونه‌ای واضح از Overfitting (بیش‌برازش) را نشان می‌دهد. در این مدل، خط رگرسیون (الگو) بیش از حد به داده‌های آموزشی (نقاط آبی) وابسته شده و منحنی را طوری فیت کرده که جزئیات و نویزهای تصادفی آن‌ها را نیز بازتاب می‌دهد، اما الگوهای کلی داده‌های تست (نقاط قرمز) را به درستی شناسایی و تعمیم نمی‌دهد. این وضعیت، مانند حفظ طوطی‌وار اطلاعات بدون درک واقعی، باعث می‌شود مدل نتواند پیش‌بینی دقیقی برای داده‌های جدید ارائه کند و عملکرد آن در محیط‌های واقعی افت کند.



شکل ۱: overfit

۲.۲ اندرفیتینگ (Underfitting)

Underfitting زمانی رخ می‌دهد که مدل **بیش از حد ساده** باشد و نتواند الگوهای اصلی داده‌های آموزشی را هم یاد بگیرد. یعنی مدل نه تنها به داده‌های جدید تعمیم نمی‌دهد، بلکه حتی روی داده‌های آموزشی هم ضعیف است. این مثل دانش‌آموزی است که درس را اصلاً نخوانده و نه سؤال‌های کتاب را بلد است، نه سؤال‌های امتحان. **:بررسی دقیق تر**

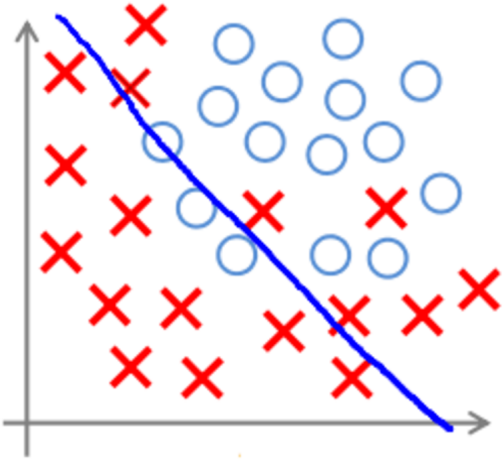
- **چرا اتفاق می‌افتد؟** مدل پیچیدگی کافی ندارد (مثل رگرسیون خطی ساده برای داده‌های غیرخطی)، یا ویژگی‌های ورودی کم/نامناسب هستند.

- **نشانه‌ها**:

- خطای Train Error و Test Error هر دو بالا (مثل MSE > ۱۰).
- دقت پایین روی هر دو مجموعه (Accuracy < ۵۰٪).

- **ارتباط با آمار**: واریانس کم (Low Variance) اما بایاس بالا (High Bias) مدل فرضیات اشتباهی دارد.

مثال عملی: در پیش‌بینی قیمت خانه، اگر مدل فقط مساحت را در نظر بگیرد و عوامل دیگر را نادیده، پیش‌بینی‌ها برای همه داده‌ها ضعیف است (Train Error: ۲۰٪، Test: ۲۲٪).
نمودار زیر نمونه‌ای واضح از Underfitting (اندرفیتینگ یا کم‌برازش) را نشان می‌دهد. در این مدل، خط رگرسیون (الگو) آن‌قدر ساده است که نه تنها الگوهای اصلی داده‌های آموزشی (نقاط آبی) را به خوبی شناسایی نمی‌کند، بلکه داده‌های تست (نقاط قرمز) را نیز به درستی تفکیک و تعمیم نمی‌دهد. این وضعیت، که ناشی از کمبود پیچیدگی مدل است، باعث کاهش شدید دقت کلی (مانند MSE بالا) و تولید پیش‌بینی‌های پرت (outlier) می‌شود. برای حل این مشکل، می‌توان ویژگی‌های بیشتری اضافه کرد یا به مدل‌های پیچیده‌تری (مانند رگرسیون چندمتغیره) مهاجرت نمود. . برای جلوگیری از این مشکل، می‌توان از تکنیک‌هایی مانند منظم‌سازی (Regularization) یا اعتبارسنجی متقابل (Cross-Validation) استفاده کرد.



شکل ۲: underfit

۳.۲ فیتینگ خوب (Good Fitting)

Good Fitting حالت ایده‌آل مدل است، جایی که **تعادل بین پیچیدگی و تعمیم‌پذیری** برقرار است. مدل الگوهای اصلی را خوب یاد می‌گیرد، بدون اینکه به نویز بچسبد. یعنی دقت روی Train و Test مشابه و بالاست مثل دانش‌آموزی که درس را خوب فهمیده و هم تمرین‌ها را حل می‌کند، هم امتحان را.
بررسی دقیق تر:

- **چرا مهم است؟** مدل واقعی و قابل اعتماد می‌شود، بدون هدررفت منابع.

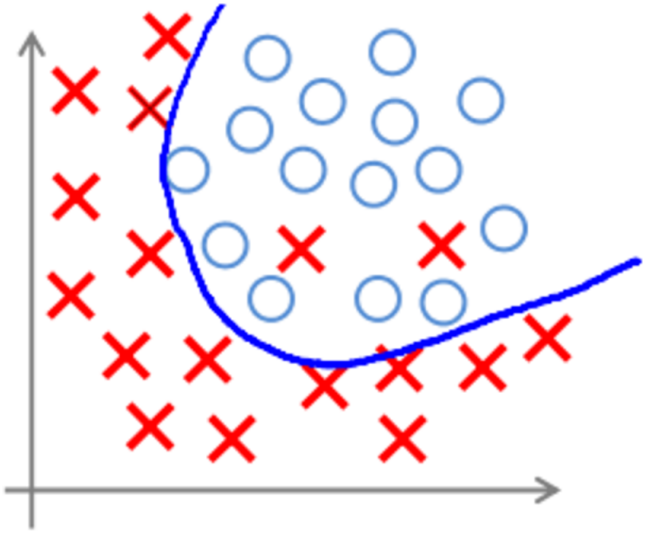
- **نشانه‌ها**:

  - Train Error و Test Error نزدیک هم و کم .
  - R^2 بالا (نزدیک ۱) و تعمیم‌پذیری خوب.

- **ارتباط با آمار**: بایاس و واریانس هر دو پایین (Bias-Variance Tradeoff) مدل بهینه است.

مثال عملی در پیش‌بینی قیمت خانه، مدل با ویژگی‌های مناسب که Train Error: ٪۵ و Test: ٪۶ پیش‌بینی دقیق برای خانه‌های جدید.

نمودار زیر نمونه‌ای واضح از Good Fitting (فیتینگ خوب یا برازش مناسب) را نشان می‌دهد. در این مدل، خط رگرسیون (الگو) با تعادل ایده‌آل، الگوهای اصلی داده‌های آموزشی (نقاط آبی) را به خوبی شناسایی و فیت می‌کند، و همزمان داده‌های تست (نقاط قرمز) را به درستی تفکیک و تعمیم می‌دهد. این وضعیت، که نشان‌دهنده درک واقعی مدل از روابط داده‌ها بدون وابستگی به نویز است، بیانگر آن است که مدل نه تنها "حفظ" کرده، بلکه "یاد گرفته" است با دقت کلی بالا (مانند R^2 نزدیک به ۱) و عملکرد پایدار در پیش‌بینی داده‌های جدید. برای دستیابی به این حالت، نظارت بر Bias-Variance Tradeoff و استفاده از تکنیک‌هایی مانند اعتبارسنجی متقابل (Cross-Validation) ضروری است.



شکل ۳: overfit

۳ انواع یادگیری

در یادگیری ماشین با دونوع یادگیری مواجه می شویم، که یک نوع آن یادگیری نظارت‌شده (Supervised Learning) یکی از شاخه‌های اصلی یادگیری ماشین است که در آن مدل با استفاده از داده‌های برچسب‌دار آموزش می‌بیند. در این روش، داده‌های ورودی (ویژگی‌ها) همراه با خروجی‌های متناظر (برچسب‌ها) به مدل ارائه می‌شوند تا مدل بتواند رابطه بین ورودی‌ها و خروجی‌ها را یاد بگیرد و از آن برای پیش‌بینی خروجی‌های جدید استفاده کند. مفاهیم اصلی:

- داده‌های برچسب‌دا: داده‌های آموزشی شامل دو بخش هستند:

  ویژگی‌ها، برچسب

- (Features): متغیرهای ورودی که مدل از آن‌ها برای یادگیری استفاده می‌کند (مثل قد، وزن، یا دما).
- (Labels) برچسب : خروجی‌های مورد انتظار که مدل باید پیش‌بینی کند (مثل "مثبت" یا "منفی" در طبقه‌بندی، یا یک عدد در رگرسیون). انواع مسائل در یادگیری نظارت‌شد:
- رگرسیون (Regression) : پیش‌بینی یک مقدار پیوسته، مثل پیش‌بینی قیمت خانه یا دمای هوا.
- طبقه‌بندی (Classification) : پیش‌بینی یک دسته یا کلاس، مثل تشخیص ایمیل به‌عنوان "هرزنامه" یا "غیرهرزنامه". مراحل یادگیری نظارت‌شده:

- جمع‌آوری داده: تهیه مجموعه داده‌ای با ویژگی‌ها و برچسب‌های دقیق.
- آموزش مدل: استفاده از الگوریتم‌هایی مثل رگرسیون خطی، درخت تصمیم، یا شبکه‌های عصبی برای یادگیری رابطه بین ویژگی‌ها و برچسب‌ها.
- ارزیابی مدل: بررسی عملکرد مدل با معیارهایی مثل دقت (Accuracy)، خطای میانگین مربعات (MSE) یا ماتریس درهم‌ریختگی (Confusion Matrix).
- پیش‌بینی: استفاده از مدل آموزش‌دیده برای پیش‌بینی برچسب داده‌های جدید. لگوریتم‌های رایج: - رگرسیون خطی (Linear Regression) - رگرسیون لجستیک (Logistic Regression) - ماشین بردار پشتیبان (SVM) - درخت تصمیم (Decision Tree) - جنگل تصادفی (Random Forest) - شبکه‌های عصبی (Neural Networks)

زایا: - دقت بالا در صورتی که داده‌های برچسب‌دار کافی و باکیفیت وجود داشته باشد. - کاربرد گسترده در مسائل واقعی مثل تشخیص پزشکی، پیش‌بینی مالی، و پردازش زبان طبیعی. چالش‌ها: - نیاز به داده‌های برچسب‌دار که ممکن است جمع‌آوری آن‌ها زمان‌بر و پرهزینه باشد. - حساسیت به داده‌های پرت (Outliers) یا نویز. - خطر بیش‌برازش (Overfitting) اگر مدل بیش از حد به داده‌های آموزشی وابسته شود.
مثال عملی: فرض کنید می‌خواهید یک مدل برای تشخیص تصاویر سگ و گربه بسازید: داده آموزشی مجموعه‌ای از تصاویر که هر کدام برچسب "سگ" یا "گربه" دارند. ویژگی‌ها: پیکسل‌های تصویر یا ویژگی‌های استخراج‌شده مثل رنگ و شکل. آموزش: مدل (مثل یک شبکه کانولوشنی) یاد می‌گیرد که الگوهای مربوط به سگ یا گربه را تشخیص دهد.

۱.۳ رگرسیون در یادگیری نظارت شده

رگرسیون یکی از تکنیک‌های اصلی در یادگیری نظارت‌شده (Supervised Learning) است که برای پیش‌بینی مقادیر پیوسته (Continuous) استفاده می‌شود. در یادگیری نظارت‌شده، مدل با استفاده از داده‌های ورودی (ویژگی‌ها) و خروجی‌های متناظر (برچسب‌ها) آموزش می‌بیند تا رابطه‌ای بین ورودی‌ها و خروجی‌ها را یاد بگیرد. در مسائل رگرسیون، خروجی یک مقدار عددی پیوسته است، مانند پیش‌بینی قیمت خانه، دمای هوا، یا میزان فروش. در رگرسیون، متغیر هدف (خروجی) مقداری پیوسته است، برخلاف مسائل طبقه‌بندی که خروجی گسسته (مانند کلاس‌ها) است. هدف: یافتن تابعی که رابطه بین ویژگی‌های ورودی (X) و خروجی (y) را به بهترین شکل مدل کند.
تابع پیش‌بینی: مدل رگرسیون یک تابع $f(X)$ تولید می‌کند که مقادیر پیش‌بینی‌شده ($\hat{y}$) را به ازای ورودی‌های جدید تخمین می‌زند.
خطا: تفاوت بین مقدار واقعی (y) و مقدار پیش‌بینی‌شده ($\hat{y}$) به عنوان خطا محاسبه می‌شود. هدف کمینه کردن این خطا است

۱.۱.۳ رگرسیون خطی ساده

رگرسیون خطی (Linear Regression) رگرسیون خطی (Linear Regression) یکی از بنیادی‌ترین و پرکاربردترین الگوریتم‌های یادگیری نظارت‌شده (Supervised Learning) در یادگیری ماشین است که به طور خاص برای پیش‌بینی مقادیر پیوسته (مانند قیمت، دما یا فروش) طراحی شده است. این مدل بر پایه فرض ساده‌ای بنا شده: رابطه بین متغیرهای ورودی (ویژگی‌ها، X) و خروجی (y) به صورت خطی است. برای مثال، اگر بخواهیم قیمت یک خانه را بر اساس دو ویژگی پیوسته مانند مساحت (x_1) و تعداد اتاق‌ها (x_2) پیش‌بینی کنیم، مدل رگرسیون خطی معادله‌ای مانند $\hat{y} = w_0 + w_1x_1 + w_2x_2$ می‌سازد، که w_0 عرض از مبدأ و w_1, w_2 ضرایب شیب هستند. این رویکرد نه تنها ساده و تفسیرپذیر است، بلکه در مسائل واقعی مانند تحلیل املاک (مانند تخمین در پلتفرم‌هایی مثل Zillow) یا پیش‌بینی تقاضای بازار، دقت بالایی ارائه می‌دهد
در یادگیری نظارت‌شده، هدف اصلی آموزش یک مدل است که با استفاده از داده‌های آموزشی (مجموعه‌ای از جفت‌های (X, y))، رابطه بین ویژگی‌های ورودی و خروجی را مدل‌سازی کند. داده‌های آموزشی شامل برچسب‌های واقعی (labels) هستند که مدل را هدایت می‌کنند تا الگوها را بیاموزد. رگرسیون خطی این کار را با کمینه‌سازی تابع هزینه (مانند میانگین مربعات خطا، MSE) انجام می‌دهد،

تا پیش‌بینی‌ها ($\hat{y}$) تا حد ممکن به مقادیر واقعی (y) نزدیک باشند. فرض کلیدی مدل، خطی بودن رابطه است یعنی تغییرات خطی در ورودی‌ها، تغییرات متناسب در خروجی ایجاد می‌کند. اگر این فرض برقرار نباشد (مثل روابط غیرخطی)، مدل‌های پیشرفته‌تری مانند رگرسیون چندجمله‌ای یا منظم‌سازی (Ridge/Lasso) پیشنهاد می‌شود. در ادامه، به طور کامل به جنبه‌های مختلف رگرسیون خطی می‌پردازیم: از فرمول‌های ریاضی و روش‌های بهینه‌سازی (مانند گرادیان نزولی) تا چالش‌هایی مانند بیش‌برازش (Overfitting) و راه‌حل‌های آماری آن.

کیمنه سازی: تابع هزینه معیاری برای سنجش میزان خطای مدل در پیش‌بینی خروجی‌ها نسبت به مقادیر واقعی است. در رگرسیون خطی، تابع هزینه معمولاً میانگین مربعات خطا (Mean Squared Error - MSE) است که تفاوت بین مقادیر واقعی (y_i) و پیش‌بینی‌شده ($\hat{y}_i$) را اندازه‌گیری می‌کند. دقت مدل را افزایش دهیم:

کمینه کردن تابع هزینه به معنای کاهش خطای پیش‌بینی است. این کار باعث می‌شود مدل پیش‌بینی‌هایی تولید کند که به مقادیر واقعی نزدیک‌تر باشند. پارامترهای بهینه را پیدا کنید. در رگرسیون خطی، پارامترها (مانند w_0 و w_1) تعیین‌کننده خط پیش‌بینی هستند. کمینه‌سازی تابع هزینه، مقادیر بهینه این پارامترها را پیدا می‌کند تا مدل بهترین تطابق را با داده‌ها داشته باشد.

تعمیم مدل با کمینه‌سازی تابع هزینه روی داده‌های آموزشی، مدل می‌تواند الگوهای واقعی داده‌ها را یاد بگیرد و روی داده‌های جدید (داده‌های آزمون) عملکرد بهتری داشته باشد، به شرطی که از بیش‌برازش (Overfitting) جلوگیری شود.

معیاری برای مقایسه مدل‌ها: مقدار تابع هزینه می‌تواند به‌عنوان معیاری برای مقایسه عملکرد مدل‌های مختلف (مثلاً رگرسیون خطی در مقابل رگرسیون چندجمله‌ای) استفاده شود.

پایه‌ای برای الگوریتم‌های پیچیده‌تر: مفهوم کمینه‌سازی تابع هزینه در رگرسیون خطی، پایه‌ای برای یادگیری الگوریتم‌های پیچیده‌تر مانند شبکه‌های عصبی است که از روش‌های مشابه (مثل گرادیان نزولی) برای بهینه‌سازی استفاده می‌کنند.

اثبات کمینه‌سازی تابع هزینه در رگرسیون خطی

$$J(w_0, w_1) = \sum_{i=1}^{n} (y_i^{(i)} - \hat{y}^{(i)})^2 = \sum_{i=1}^{n} (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \tag{۱}$$

که در آن y_i مقدار واقعی، $\hat{y}_i = w_0 + w_1 x_i$ مقدار پیش‌بینی‌شده. می باشد.

کمینه‌سازی تابع هزینه برای کمینه کردن $J(w_0, w_1)$، مشتق‌های جزئی نسبت به w_0 و w_1 را محاسبه و صفر می‌کنیم.

مشتق نسبت به w_0

$$\frac{\partial J}{\partial w_0} = \frac{\partial}{\partial w_0} \left[\sum_{i=1}^{n} (y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2 \right]$$

$$-2 \sum_{i=1}^{n} (y_i^{(i)} - w_0 - w_1 x_i^{(i)})^2 = 0$$

$$n w_0 + w_1 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$$

$$w_0 = \frac{\sum_{i=1}^{n} y_i - w_1 \sum_{i=1}^{n} x_i}{n} = \bar{y} - w_1 \bar{x}$$

که در آن $\bar{x} = \frac{\sum x_i}{n}$ و $\bar{y} = \frac{\sum y_i}{n}$ میانگین داده‌ها هستند.

مشتق نسبت به w_1

$$\frac{\partial J}{\partial w_1} = \frac{\partial}{\partial w_1}\left[\sum_{i=1}^{n}(y_i^{(i)} - (w_0 + w_1 x_i^{(i)}))^2\right]$$

$$= \sum_{i=1}^{n} 2(y_i^{(i)} - w_0 - w_1 x_i^{(i)})(-x_i^{(i)}) = \sum_{i=1}^{n} x_i(y_i^{(i)} - w_0 - w_1 x_i^{(i)})$$

صفر کردن مشتق و جایگذاری $w_0 = \bar{y} - w_1\bar{x}$:

$$\sum_{i=1}^{n} x_i^{(i)}\left(y_i^{(i)} - (\bar{y} - w_1\bar{x}) - w_1 x_i^{(i)}\right) = 0$$

$$\sum_{i=1}^{n} x_i^{(i)}(y_i^{(i)} - \bar{y} - w_1(x_i^{(i)} - \bar{x})) = 0$$

$$\sum_{i=1}^{n} x_i^{(i)}(y_i^{(i)} - \bar{y}) - w_1\sum_{i=1}^{n} x_i(x_i^{(i)} - \bar{x}) = 0$$

حل برای w_1:

$$w_1 = \frac{\sum_{i=1}^{n} x_i^{(i)}(y_i^{(i)} - \bar{y})}{\sum_{i=1}^{n} x_i^{(i)}(x_i^{(i)} - \bar{x})}$$

برای ساده‌سازی، از فرم معادل استفاده می‌کنیم:

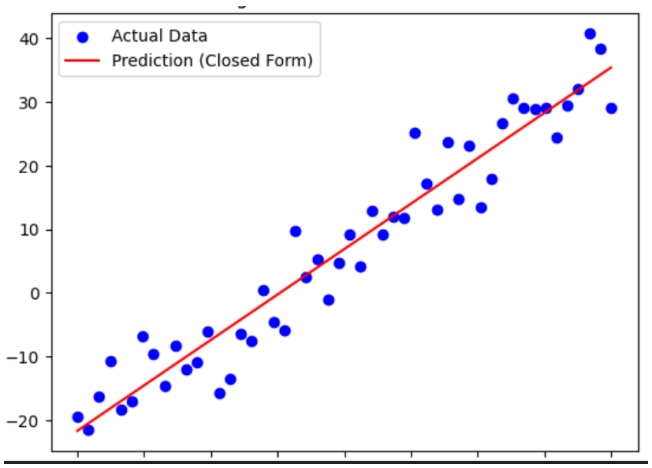
$$w_1 = \frac{\sum_{i=1}^{n}(x_i^{(i)} - \bar{x})(y_i^{(i)} - \bar{y})}{\sum_{i=1}^{n}(x_i^{(i)} - \bar{x})^2}$$

برآوردگرهای بهینه بنابراین، برآوردگرهای بهینه برای پارامترها عبارت‌اند از:

$$\hat{w_1} = \frac{\sum_{i=1}^{n}(x_i^{(i)} - \bar{x})(y_i^{(i)} - \bar{y})}{\sum_{i=1}^{n}(x_i^{(i)} - \bar{x})^2} \tag{۲}$$

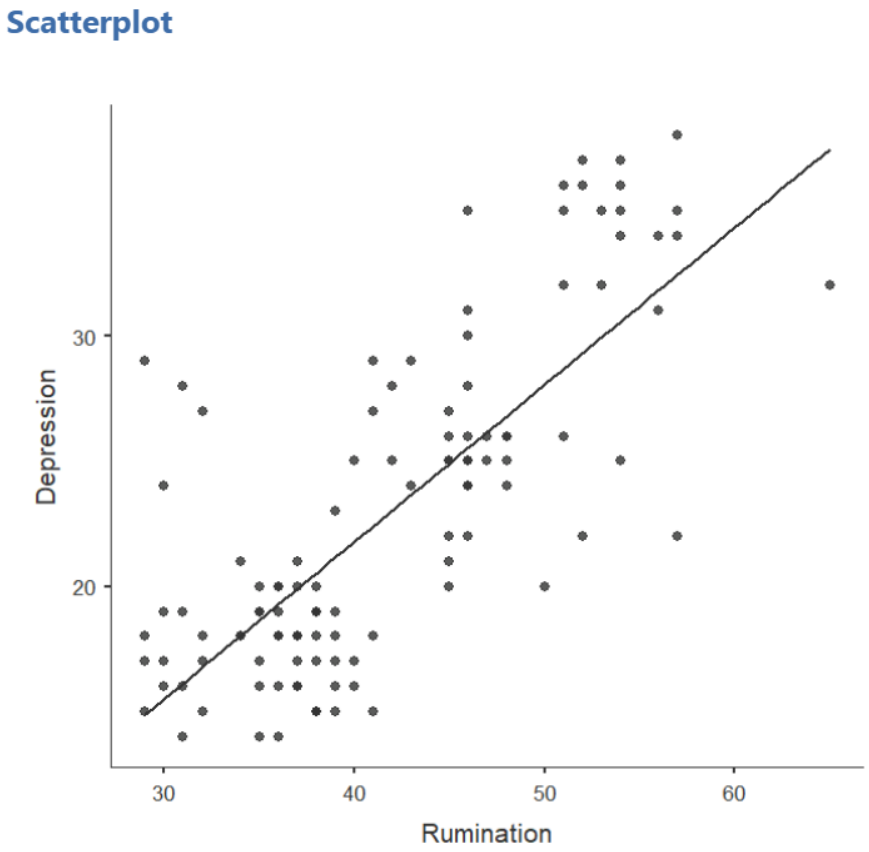
$$\hat{w_0} = \bar{y} - w_1\bar{x} \tag{۳}$$

تاثیر کمینه سازی بر مدل خطی: نمودار زیر نمونه‌ای واضح از تأثیر مثبت کمینه‌سازی در رگرسیون خطی ساده را نشان می‌دهد. در این مدل، خط پیش‌بینی (قرمزبا استفاده از روش کمینه‌سازی Least Squares، الگوهای اصلی داده‌های واقعی (نقاط آبی) را به خوبی شناسایی و فیت می‌کند. این رویکرد، که معادلات نرمال را حل می‌کند، تعادل مناسبی بین دقت و تعمیم‌پذیری ایجاد کرده و از Overfitting یا Underfitting جلوگیری می‌نماید.



شکل ۴: overfit

نمودار زیر تأثیر کمینه‌سازی را در داده‌های واقعی‌تر (مانند رابطه افسردگی و نشخوار فکری) نشان می‌دهد. در اینجا، خط رگرسیون (سیاه) با استفاده از کمینه‌سازی استاندارد، الگوهای کلی را شناسایی می‌کند، اما به دلیل سادگی مدل، نتوانسته داده‌ها را به طور کامل تفکیک کند نشانه‌ای از Underfitting. این وضعیت، که در آن MSE هنوز بالاست، بیانگر نیاز به مدل‌های پیشرفته‌تر (مانند رگرسیون چندمتغیره یا منظم‌سازی) است.



شکل ۵: overfit

۲.۱.۳ رگرسیون خطی چند متغیره

. (Multiple Linear Regression) یکی از روش‌های کلیدی در یادگیری نظارت‌شده است که برای پیش‌بینی یک متغیر وابسته (خروجی پیوسته) بر اساس چندین متغیر مستقل (ویژگی‌ها) استفاده می‌شود. این روش زمانی کاربرد دارد که بخواهیم رابطه‌ای خطی

بین چندین متغیر ورودی و یک خروجی پیوسته را مدل کنیم. در یادگیری نظارت‌شده، داده‌های آموزشی شامل جفت‌های ورودی (ویژگی‌ها) و خروجی (برچسب‌ها) هستند، و رگرسیون خطی چندگانه با یادگیری این رابطه، پیش‌بینی‌های دقیقی برای داده‌های جدید ارائه می‌دهد. در ادامه، به کاربردهای این روش و فرضیات آن می‌پردازیم.

رگرسیون چندگانه به ما کمک می‌کند تا اثر هر متغیر مستقل بر خروجی را درک کنیم. ضرایب مدل نشان‌دهنده میزان تأثیر هر ویژگی هستند.

مثال: در بازاریابی، می‌توان تأثیر بودجه تبلیغات، نوع رسانه، و فصل فروش بر میزان فروش را بررسی کرد.

کاربردهای گسترده در حوزه‌های مختلف:

اقتصاد: پیش‌بینی رشد اقتصادی یا تورم بر اساس متغیرهایی مانند نرخ بهره، تولید ناخالص داخلی، و بیکاری.

پزشکی: پیش‌بینی فشار خون بیمار بر اساس سن، وزن، سطح فعالیت بدنی، و رژیم غذایی.

علوم طبیعی: پیش‌بینی دمای هوا با استفاده از متغیرهایی مانند رطوبت، فشار جو، و سرعت باد.

بازاریابی و تجارت: پیش‌بینی فروش محصول بر اساس قیمت، تبلیغات، و رفتار مشتری.

مهندسی: پیش‌بینی مصرف انرژی یک ساختمان بر اساس اندازه، تعداد ساکنان، و نوع عایق‌بندی. پایه‌ای برای مدل‌های پیچیده‌تر:

رگرسیون خطی چندگانه به دلیل سادگی و تفسیرپذیری، پایه‌ای برای درک مدل‌های پیشرفته‌تر مانند رگرسیون منظم‌شده (ریج، لاسو) یا شبکه‌های عصبی است.

این روش به‌عنوان یک معیار اولیه (Baseline) برای مقایسه با مدل‌های پیچیده‌تر استفاده می‌شود.

تصمیم‌گیری مبتنی بر داده:

با ارائه پیش‌بینی‌های دقیق، رگرسیون چندگانه به تصمیم‌گیری در حوزه‌هایی مانند مدیریت ریسک، برنامه‌ریزی مالی، و بهینه‌سازی منابع کمک می‌کند.

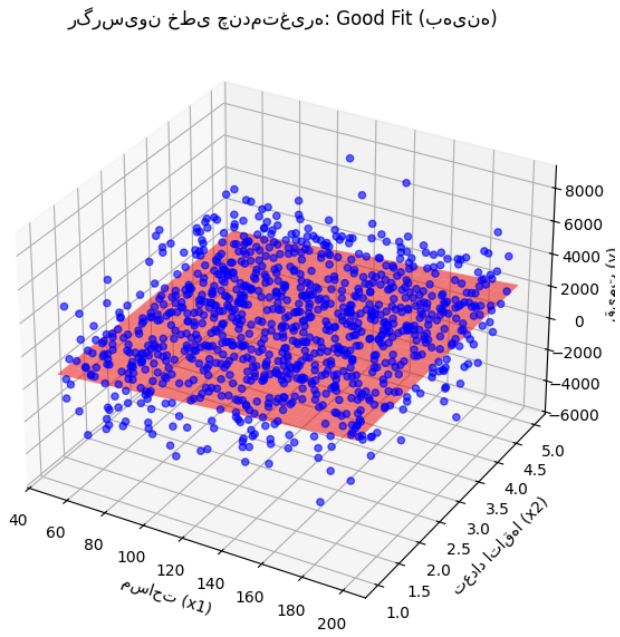
مثال: پیش‌بینی تقاضای محصول برای مدیریت موجودی در زنجیره تأمین.

تحلیل روابط پیچیده:

این روش امکان مدل‌سازی تعاملات بین متغیرهای مختلف را فراهم می‌کند، به‌ویژه زمانی که یک متغیر به‌تنهایی نمی‌تواند خروجی را توضیح دهد.

مثال: پیش‌بینی عملکرد تحصیلی دانش‌آموز بر اساس ساعات مطالعه، کیفیت تدریس، و سطح استرس.

بررسی عملکرد در مدل رگرسیون خطی نمودار زیر نمونه‌ای از اندرفیتینگ (Underfitting) در رگرسیون خطی چندمتغیره را نشان می‌دهد. در این مدل، سطح مدل (صفحه قرمز) تقریباً صاف است و تغییرات پیچیده دیتای سه‌بعدی را دنبال نمی‌کند. نقاط آبی (داده‌های واقعی) در اطراف این صفحه پراکندگی زیادی دارند و مدل الگوی واقعی داده را یاد نگرفته است. اگر مدل خوب بود، صفحه باید کمی خمیدگی یا شیب متناسب با داده نشان می‌داد، اما اینجا فقط یک صفحه ساده رسم شده است.



شکل ۶: اندرفیتینگ در رگرسیون خطی چندمتغیره: صفحه قرمز صاف (مدل ساده) و پراکندگی نقاط آبی (residuals بزرگ).

این وضعیت، MSE بالا و R^2 پایین را نتیجه می‌دهد، که با مفهوم High Bias در آمار همخوانی دارد **کمینه‌سازی تابع هزینه در رگرسیون خطی چندگانه** رگرسیون خطی چندگانه رابطه بین متغیر وابسته (y) و چندین متغیر مستقل ($x_1, x_2, \ldots, x_n$) را به صورت زیر مدل می‌کند:

$$y = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n \tag{۴}$$

به صورت ماتریسی: $y = Xw$، که در آن:

- $X = [1, x_1, x_2, \ldots, x_n]$: ماتریس ویژگی‌ها (ستون اول برای w_0 پر از ۱ است).
- $w = [w_0, w_1, \ldots, w_n]^T$: بردار ضرایب.
- y: بردار خروجی‌ها.

تابع هزینه تابع هزینه میانگین مربعات خطا (MSE) به صورت زیر تعریف می‌شود:

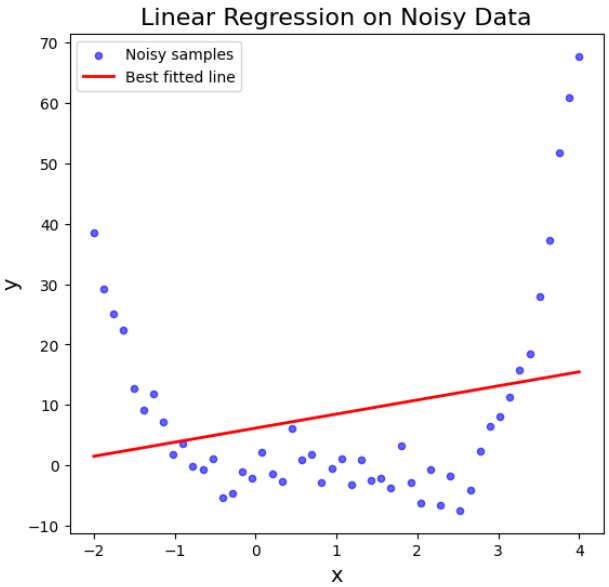
$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1x_{i1} + \cdots + w_nx_{in}))^2 \tag{۵}$$

یا به صورت ماتریسی:

$$J(w) = \frac{1}{n}(y - Xw)^T(y - Xw) \tag{۶}$$

کمینه سازی رگرسیون چند جمله ای برای کمینه‌سازی $J(w)$، مشتق تابع هزینه نسبت به w محاسبه و صفر می‌شود. تابع هزینه را گسترش می‌دهیم:

$$J(w) = \frac{1}{n}\left[(y^Ty - 2y^TXw + w^TX^TXw)\right]$$



شکل ۷: استفاده از رگرسیون خطی با که باعث نویز شده است.

مشتق نسبت به w:

$$\frac{\partial J}{\partial w} = \frac{\partial}{\partial w}\left[\frac{1}{n}(y^T y - 2y^T X w + w^T X^T X w)\right] \tag{۷}$$

$$= \frac{1}{n}\left(-2X^T y + 2X^T X w\right) = \frac{2}{n}\left(X^T X w - X^T y\right)$$

صفر کردن مشتق:

$$X^T X w = X^T y \tag{۸}$$

برآوردگر w:

$$\hat{w} = (X^T X)^{-1} X^T y \tag{۹}$$

این برآوردگر بهینه، تابع هزینه را کمینه می‌کند، مشروط بر اینکه $X^T X$ معکوس‌پذیر باشد (یعنی ویژگی‌ها هم‌خطی کامل نداشته باشند).

تعمیم حالت خطی. ؟؟ رگرسیون خطی ساده (Simple Linear Regression) برای مدل‌سازی رابطه بین یک متغیر مستقل (x) و یک متغیر وابسته (y) به کار می‌رود، اما در بسیاری از مسائل واقعی، خروجی (y) تحت تأثیر چندین متغیر مستقل است. در چنین مواردی، رگرسیون خطی چندگانه (Multiple Linear Regression) به دلیل توانایی در نظر گرفتن چندین متغیر مستقل به طور همزمان، انتخاب مناسب‌تری است. در ادامه، به دلایل اصلی استفاده از رگرسیون چندگانه به جای رگرسیون خطی ساده، استفاده خواهد شد.

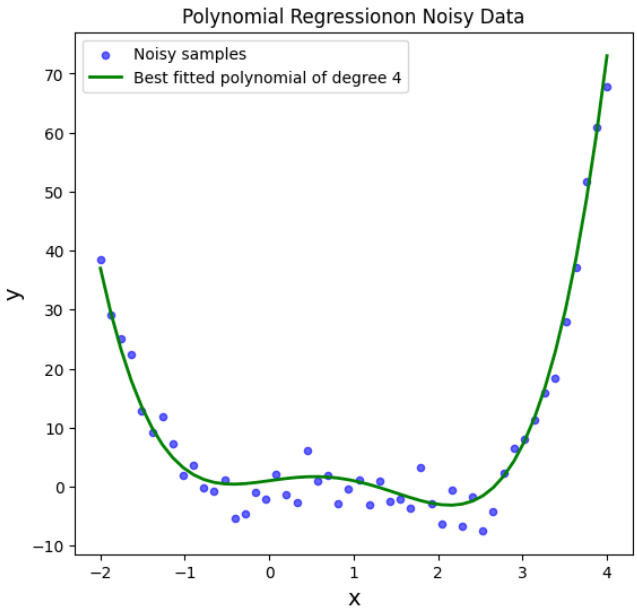
همان طوری که در بالا هم مشاهده می شود وقتی از رگرسیون خطی روی داده های چند گانه می شود باعث این می شود یک نویز یا خطایی به وجود آید حتی برای آن یک تخمین اولیه را نداشته باشیم. برای رفع این مشل می توانیم از رگرسیون های چند گانه با درجه هایی مختلفی استفاه کنیم و این نویز را از بیین ببریم. به تصویر عدی نگاه کنید.

همان طور که در تصویر بالا مشاهده می شود زمانی که درجه را زیاد می کنیم باعث می شود که نمودار به سمت چند جمله برود در

نهایت خطایی مدل ما خیلی کاهش یابد و از overfit جلوگیری شود. نکته ای باید مد نظر داشته باشید شما می توانید از چند تبدیل های مختلفی استفاده کنید و بهترین برارزش را برای داده خود انجام دهید. در ادامه به برخی از از تبدیل ها اشاره خواهیم کرد.

:انواع تبدیل‌ها

- تبدیل لگاریتمی: $y' = \log(y)$ یا $x'_i = \log(x_i)$. برای داده‌های با توزیع کج یا روابط نمایی.
- تبدیل چندجمله‌ای: $x'_i = x_i^2, x_i^3, \ldots$. برای مدل‌سازی روابط غیرخطی.
- تبدیل ریشه مربع: $y' = \sqrt{y}$ یا $x'_i = \sqrt{x_i}$. برای کاهش کجی یا اثر مقادیر بزرگ.
- تبدیل معکوس: $y' = \frac{1}{y}$. برای روابط معکوس یا مدیریت داده‌های پرت.



شکل ۸: استفاده نمودار غیر خطی .

از جمله تبدیل های هستند که می توانید برای برزاش داده استفاده کنید از overfit و underfit شدن جلوگیری کنید.

۳.۱.۳ گرادیان(gradeint)

گرادیان نزولی (Gradient Descent) یکی از الگوریتم‌های بهینه‌سازی پرکاربرد در یادگیری ماشین است که برای کمینه‌سازی تابع هزینه (Cost Function) در مدل‌هایی مانند رگرسیون خطی، رگرسیون لجستیک، و شبکه‌های عصبی استفاده می‌شود. این روش به‌ویژه در مواردی که داده‌ها حجم زیادی دارند یا محاسبات تحلیلی (مانند روش حداقل مربعات) هزینه‌بر هستند، بسیار مفید است. در ادامه، توضیح جامعی درباره گرادیان نزولی، نحوه کار، انواع آن، کاربردها، و مزایا و معایب آن ارائه می‌شود. گرادیان نزولی یک الگوریتم تکراری است که برای یافتن مقادیر بهینه پارامترهای مدل مانند ($w_0, w_1, \ldots$) به‌گونه‌ای که تابع هزینه ($J(w)$) کمینه شود، استفاده می‌شود. ایده اصلی این است که با حرکت در جهت مخالف گرادیان (مشتق تابع هزینه)، به سمت نقطه کمینه حرکت کنیم.
گرادیان: بردار مشتق‌های جزئی تابع هزینه نسبت به پارامترها، که جهت و شدت تغییرات تابع هزینه را نشان می‌دهد.
نزولی: حرکت در جهت مخالف گرادیان برای کاهش مقدار تابع هزینه.

شروع با مقادیر اولیه: مقادیر اولیه تصادفی یا صفر برای پارامترها (w) انتخاب می‌شود. محاسبه گرادیان: مشتق جزئی تابع هزینه نسبت به هر پارامتر محاسبه می‌شود:

$$\frac{\partial J}{\partial w_j}$$

به‌روزرسانی پارامترها: پارامترها در جهت مخالف گرادیان با گام مشخص (نرخ یادگیری، α) به‌روزرسانی می‌شوند:

$$w_j := w_j - \alpha \frac{\partial J}{\partial w_j}$$

تکرار : مراحل بالا تا زمانی تکرار می‌شود که تابع هزینه به مقدار کمینه برسد یا تغییرات آن ناچیز شود (همگرایی).

انواع گرادیان

گرادیان نزولی بر اساس نحوه استفاده از داده‌ها به سه نوع اصلی تقسیم می‌شود:

1. گرادیان نزولی دسته‌ای (Batch Gradient Descent): گرادیان با استفاده از کل داده‌های آموزشی محاسبه می‌شود.

مزایا : همگرایی پایدارتر به سمت کمینه. معایب : برای داده‌های بزرگ، محاسبات سنگین و کند است.

2. گرادیان نزولی تصادفی (Stochastic Gradient Descent - SGD): گرادیان برای هر نمونه داده به‌صورت جداگانه محاسبه و پارامترها به‌روزرسانی می‌شوند.

مزایا : سریع‌تر برای داده‌های بزرگ، می‌تواند از کمینه‌های محلی فرار کند. معایب: نوسانات زیاد در تابع هزینه، ممکن است به کمینه دقیق نرسد.

3. گرادیان نزولی مینی‌بچ (Mini-Batch Gradient Descent): داده‌ها به دسته‌های کوچک (mini-Batches) تقسیم می‌شوند و گرادیان برای هر دسته محاسبه می‌شود. مزایا: تعادل بین سرعت و پایداری، پرکاربرد در شبکه‌های عصبی. معایب: نیاز به تنظیم اندازه دسته.

کاربرد گرادیان نزولی در رگرسیون خطی در رگرسیون خطی، تابع هزینه معمولاً میانگین مربعات خطا (MSE) است:

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (w_0 + w_1 x_{i1} + \cdots + w_n x_{in}))^2$$

گرادیان نزولی برای یافتن $w_0, w_1, \ldots, w_n$ استفاده می‌شود:

مشتق‌های جزئی:

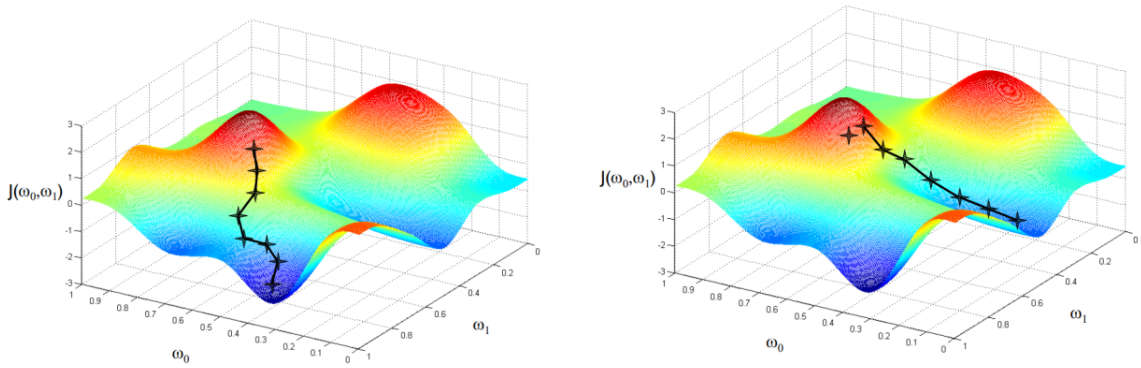
$$\frac{\partial J}{\partial w_0} = -\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)$$

$$\frac{\partial J}{\partial w_j} = -\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) x_{ij}, \quad j = 1, \ldots, n$$

به‌روزرسانی:

$$w_j := w_j + \alpha \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i) x_{ij}$$

cost function در این نمودار یک سطح سه بعدی است که محورهای آن به صورت زیر تعریف شده‌اند:



شکل ۹: گرادیان کاهشی

- محور x: θ_0 (پارامتر اول، مثلاً w_0).
- محور y: θ_1 (پارامتر دوم، مثلاً w_1).
- محور هر نمودار یک سطح سه بعدی است که محورهای آن به صورت زیر تعریف شده‌اند:

محور x: θ_0 (پارامتر اول، مثلاً w_0). محور y: θ_1 (پارامتر دوم، مثلاً w_1). محور z: $J(\theta_0, \theta_1)$ (مقدار تابع هزینه).
نمودار سمت چپ: مسیر گرادیان نزولی با نرخ یادگیری (α) بالاتر یا تعداد مراحل کمتر نشان داده شده است. مسیر دارای انحنا و نوسانات بیشتری است و به نظر می‌رسد هنوز به کمینه دقیق نرسیده است.
نمودار سمت راست: مسیر صاف‌تر و مستقیم‌تر است، که نشان‌دهنده نرخ یادگیری مناسب‌تر یا تعداد مراحل بیشتر است. این مسیر به طور واضح به سمت کمینه همگرا شده است.

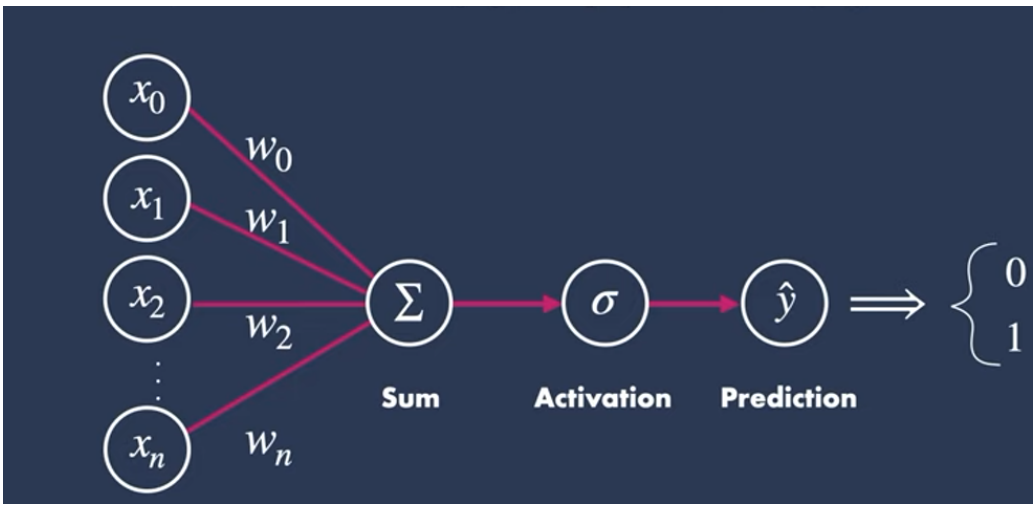
۴.۱.۳ رگرسیون لجستیک

می توان تابع لجستیک را به این شکل تعریف کرد.

$$\hat{Y} = \sigma(z) = \sigma(x_0 w_0 + x_1 w_1 + ... + x_n w_n)$$

تابع سیگومد به این شکل می باشد.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

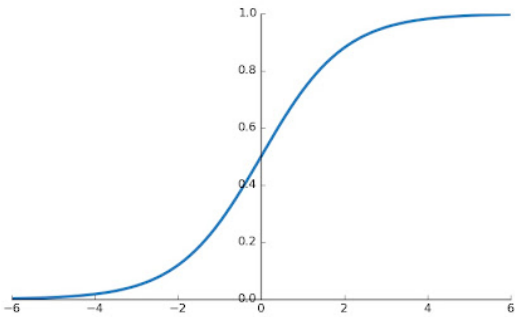


شکل ۱۰: تابع لجسیتک در یادگیری ماشین

و همچنین در مورد مشقق پذیری

$$\lim_{z\to-\infty} \sigma(x) = \frac{1}{1+\infty} = 0$$
$$\lim_{z\to\infty} \sigma(x) = \frac{1}{1+0} = 1$$
$$\lim_{z\to 0} \sigma(x) = \frac{1}{1+1} = 0.5$$

از آن جایی که $-\infty < x < \infty$ پس می توان نوشت $0 < \sigma(z) < 1$ می باشد. و نمودار به شکل زیر می باشد



شکل ۱۱: نمودار تابع لجسیتک

یکی دیگر از ویژگی های خوبی که این تابع دارد متشق پذیر است.

$$\frac{\partial}{\partial z}\left(\frac{1}{(1+e^{-z})}\right) = -\frac{e^{-z}}{(1+e^{-z})}\frac{1}{1+\sigma^{-z}} = (\sigma(z))(1-\sigma(z))$$

در یادگیری ماشین ما نیاز داریم که عددی که یا پیش بینی های که به ما باز گرداننده می شوند یک عدد بین صفر و یک باشد که برای

همین می توان از تابع لجسیتیک به شکل نووشت:

$$p(y=1|x,w)=\sigma(w^Tx)$$
$$p(y=0|x,w)=(1-\sigma(w^Tx))$$

به‌دست آوردن تابع هزینه در رگرسیون لجستیک

$$P(y_i=1 \mid x_i;w)=\sigma(w^\top x_i)=\frac{1}{1+e^{-w^\top x_i}}$$

برای $y_i \in \{0,1\}$، توزیع برنولی را داریم:

$$P(y_i \mid x_i;w)=\sigma(w^\top x_i)^{y_i}\cdot(1-\sigma(w^\top x_i))^{1-y_i}$$

با فرض مستقل بودن مشاهدات، تابع درست‌نمایی برابر است با:

$$L(w)=\prod_{i=1}^{n}\sigma(w^\top x_i)^{y_i}\cdot(1-\sigma(w^\top x_i))^{1-y_i}$$

برای ساده‌سازی، لگاریتم تابع درست‌نمایی را می‌گیریم:

$$\ell(w)=\log L(w)=\sum_{i=1}^{n}\left[y_i\log\sigma(w^\top x_i)+(1-y_i)\log(1-\sigma(w^\top x_i))\right]$$

حال برای پیدا کردن بردار وزن بهینه w، گرادیان تابع لگاریتم درست‌نمایی را محاسبه می‌کنیم:

$$\nabla_w\ell(w)=\sum_{i=1}^{n}(y_i-\sigma(w^\top x_i))x_i$$

این تابع شبیه به رگرسیون نیست که با استفاده از فرم بسته بتوان برای w پیدا کرد. پس با استفاده از گرایان مقدار بهینه را برای پارامتر پیدا خواهیم کرد.
برای بیشینه‌سازی تابع درست‌نمایی، از الگوریتم‌های عددی مانند گرادیان افزایشی (Gradient Ascent) یا گرادیان کاهشی (Gra-dient Descent) استفاده می‌شود. در عمل، ما معمولاً گرادیان کاهشی را برای کمینه‌سازی تابع هزینه معادل استفاده می‌کنیم:

$$J(w)=-\ell(w)=-\sum_{i=1}^{n}\left[y_i\log\sigma(w^\top x_i)+(1-y_i)\log(1-\sigma(w^\top x_i))\right]$$

مشتق منفی آن نیز همان است:

$$\nabla_w J(w)=-\sum_{i=1}^{n}(y_i-\sigma(w^\top x_i))x_i$$

۵.۱.۳ نظم‌دهی (Regularization)

یک تکینک مهم در ماشین لرنینگ است که باعث کاهش کمینه کردن و بهبود دقت می شود، با جلوگیری از بیش برازش (Overfitting-) حذف داده های پردت و باعث کاهش نویز می شود و کمک می کند مدل ساده تر شود و روی داده های جدید به خوبی عمل می کند. ابن عمل فقط با افزون یک ترم یا جریمه صورت خواهدد گرفت.

انواع(Regularization) مدلی رگرسیونی که از تکنیک نظم‌دهی L1 استفاده می‌کند، رگرسیون لاسو (LASSO) نام دارد که مخفف (Least Absolute Shrinkage and Selection Operator) است. در این روش، مقدار قدر مطلق ضرایب (ضریب‌های مدل) به‌عنوان یک جریمه (penalty) به تابع زیان (L) اضافه می‌شود. این جریمه می‌تواند باعث شود برخی ضرایب کاملاً به صفر برسند، که در نتیجه ویژگی‌های کم‌اهمیت حذف شده و تنها ویژگی‌های مهم باقی می‌مانند.

۶.۱.۳ رگرسیون لاسو

Lasso Regression یک روش رگرسیونی مبتنی بر تکنیک "کوچک‌سازی و انتخاب بر اساس قدر مطلق است که در تحلیل رگرسیون برای انتخاب متغیرها و اعمال نظم‌دهی Regularization استفاده می‌شود. این روش به حذف ویژگی‌های غیرمرتبط داده کمک می‌کند و از بیش‌برازش Overfitting جلوگیری می‌کند. در نتیجه، ویژگی‌هایی که تأثیر ضعیفی دارند به‌خوبی مشخص می‌شوند، چرا که ضرایب متغیرهای کم‌اهمیت به سمت صفر میل می‌کنند. در حقیقت می‌توان گفت که رگرسیون لاسو یکی از روش‌هایی است که با افزودن یک جمله‌ی جریمه به تابع هزینه مدل خطی، از بیش‌برازش جلوگیری می‌کند. تابع هزینه در این روش به صورت زیر تعریف می‌شود:

$$J(w) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) + \lambda \sum_{i=1}^{m} |w_i|$$

و همچنین در فرمول بالا $minRss = \sum(y_i - \hat{y}_i)^2$

y= مقدار مشاهده شده.

$\hat{y}$= مقدار پاسخ بیش بینی شده

و در قسمت جمله جریمه (penalty prem) $\lambda \sum |w|$ این تابع مشتق پذیر نمی باشد.

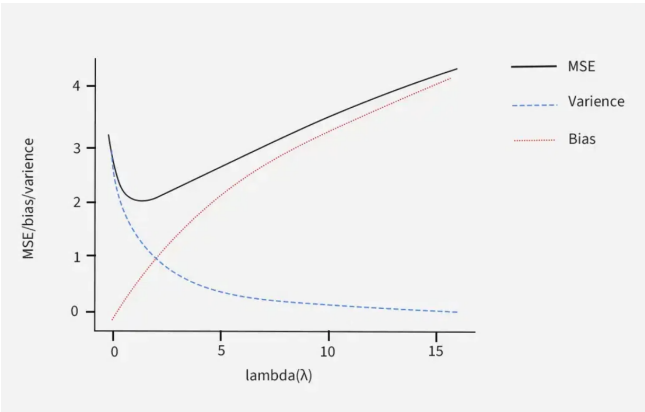
w= ضرایب

λ یک پارامتر تنظیم (uning parameter) است که شدت جریمه (penalty) را کنترل می‌کند. هرچه مقدار λ بیشتر شود، ضرایب بیشتری به سمت صفر رانده می‌شوند.

و می توان به شکل زیر فرمول بالا را بازنویسی کرد:

$$J(w) = min\ RSS + \lambda \sum_{i=1}^{m} |w_i|$$

تنظیم مقدار بهینه لاندا در ادامه دو معیار مهم در ماشین لرنینگ را مورد بررسی قرار خواهیم داد.

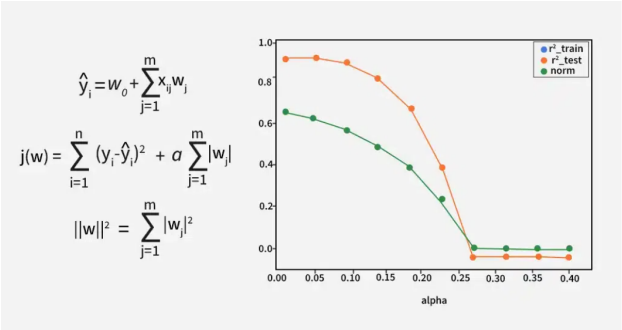


شکل ۱۲: biase and varince in lasso

Bias= وقتی یک مدل بسیار ساده باشد و داده های ما پیچده باشند نقاط پرت زیاد می شوند در نهایت خطا افزایش می یابد.

variance= در حقیقت وقتی مدل پیچیده باشد می تواند تنها روی داده های آموزش یا داده تست به خوبی فیت شود و در حالی در برگیرنده داده های آموزش نیست در نهایت می توان گفت که واریانس زیاد شده و اورفیت رخ داده است.

panlty= کاربردی که دارد باعث می شود ویژگی های که اهمیت کمی دارند حذف شوند. و این عمل باعث کاهش واریانس می شود و از اورفیت شدن جلوگیری میکند. **افزایش مقدار** λ باعث این می شود مدل بیش از حد ساده باشد و در نهایت باع ث این میشود که underfit رخ دهد. برای رسیدن به تعادل درست بین Bias و Variance باید مقدار مناسب λ را با استفاده از cross-validation پیدا کرد.



شکل ۱۳: cost function

در نمودار مربوط به Lasso Regression، تابع هزینه به‌صورت ترکیبی از مجموع مربعات باقیمانده‌ها (Residual Sum of Squares یا RSS) و یک جریمه‌ی L1 بر ضرایب β_j تعریف می‌شود. بخش اول یعنی RSS، اختلاف مربعی بین مقادیر واقعی و مقادیر پیش‌بینی‌شده را اندازه‌گیری می‌کند. بخش دوم، یعنی جریمه‌ی L1 (یا L1 penalty)، مقدار قدر مطلق ضرایب را جریمه می‌کند و باعث می‌شود برخی از ضرایب به صفر نزدیک شوند یا دقیقاً صفر شوند. این امر باعث ساده‌تر شدن مدل می‌گردد. شدت اثر این جریمه توسط پارامتر λ (یعنی lambda) کنترل می‌شود.

در نمودار، محور عمودی (محور y) مقدار تابع هزینه را نشان می‌دهد که Lasso Regression تلاش می‌کند آن را کمینه کند، و محور افقی (محور x) مقدار پارامتر λ را نمایش می‌دهد که شدت جریمه‌ی L1 را مشخص می‌کند. منحنی‌ای که از رنگ سبز به نارنجی تغییر می‌کند، نشان‌دهنده‌ی تغییرات مقدار تابع هزینه نسبت به تغییرات λ است. با افزایش λ، مقدار جریمه بیشتر شده و در نتیجه مقدار تابع هزینه نیز افزایش می‌یابد. این افزایش موجب می‌شود ضرایب بیشتری به سمت صفر رانده شوند و در نتیجه مدل ساده‌تر شود.

۷.۱.۳ رگرسیون ریج

رگرسیون ریج یک regularized است از رگرسسیون خطی. وبا اضافه کردن یک تابع جریمه به cost function کمک می کند یک به نقطه بهینه برسیم و هم چنین $overfit$ کمتری رخ بدهد.

$$J(\theta) = MSE(\theta) + \alpha \sum_{i}^{n} \theta_i^2$$

نکته: در رگرسیو ریج به این مورد توجه کنید که ویژگی های که اهیمت کمی دارند کوچیک می شوند یا نزیدک به صفر. نکته: از این ترم $\alpha \sum_{i=1}^{n} \theta_i^2$ استفاده می شود.

نکته: برای این که معیار عمل کرد مدل را تشخیص دهیم. یعنی وقتی که مدل را استفاده از تست دیتا عملکرد آن را مورد سیجش قرار می دهیم از این ترم استفاده خواهیم $j(\theta) = MSE(\theta)$ این قسمت برای این که متشق پذیر است و گرادیان آن راحت حساب می شود از آن استفاده خواهیم کرد.

توان منظم‌دهی (Regularization strength) با آلفا (α) مشخص می‌شود که یکی از hyperparameter و دو ویژگی دارد:
1. اگر $\alpha = 0$ باشد، هیچ منظم‌دهی اعمال نمی‌شود و مدل به رگرسیون خطی استاندارد (OLS) تبدیل می‌شود.

۲.۳ یادگیری بدون نظارت

تعریف ...یادگیری بدون نظارت (Unsupervised Learning) به تحلیل داده‌های بدون برچسب برای کشف الگوها یا ساختارهای مخفی در داده‌ها اشاره دارد

۱.۲.۳ K-means

۱. **خوشه‌بندی (Clustering):**

- داده‌ها را بر اساس شباهت به گروه‌هایی تقسیم می‌کند.
- نقاط داده در یک خوشه به یکدیگر شبیه‌ترند تا به نقاط خوشه‌های دیگر.
- شباهت بسته به نوع مسئله تعریف می‌شود (مثلاً رفتار خرید در تقسیم‌بندی بازار).

۲. **کاهش ابعاد :(Dimensionality Reduction)**

- تعداد ویژگی‌های داده را کاهش می‌دهد، اما ویژگی‌های مهم و اطلاعاتی را حفظ می‌کند.

۳. **تشخیص ناهنجاری (Anomaly Detection):**

- شناسایی نقاط داده‌ای که به طور قابل‌توجهی از حالت عادی منحرف هستند (مثلاً تشخیص تقلب).

۴. **مدل‌سازی مولد (Generative Modeling):**

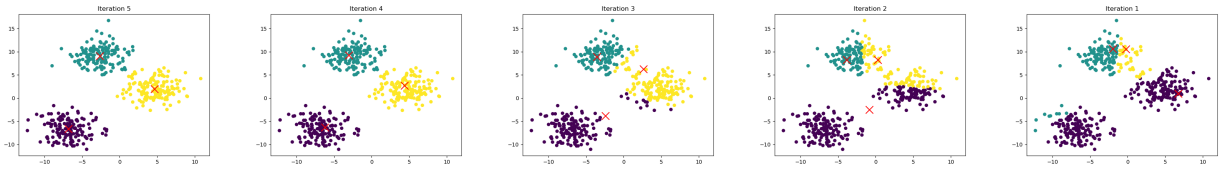
- یادگیری توزیع داده‌ها برای تولید نمونه‌های جدید و مشابه.

کاربردهای خوشه‌بندی

- **تقسیم‌بندی مشتریان (Customer Segmentation):** در بازاریابی برای گروه‌بندی مشتریان بر اساس رفتار.
- **تقسیم‌بندی تصویر و تشخیص اشیا (Image Segmentation and Object Detection):** در بینایی کامپیوتری.
- **تشخیص ناهنجاری:** در امنیت سایبری و امور مالی.
- **ژنتیک و بیوانفورماتیک:** تحلیل داده‌های زیستی.
- **تحلیل شبکه‌های اجتماعی و تشخیص جوامع:** شناسایی گروه‌های مشابه در شبکه‌ها.

الگوریتم خوشه‌بندی K-Means

- پرکاربردترین الگوریتم خوشه‌بندی.
- داده‌ها را به **K** گروه مجزا بر اساس شباهت ویژگی‌ها تقسیم می‌کند.
- با تخصیص تکراری نقاط داده به نزدیک‌ترین مرکز (میانگین گروه) کار می‌کند و سپس مراکز را بر اساس عضویت‌های جدید گروه‌ها بازمحاسبه می‌کند.
- این فرآیند تا زمانی که تخصیص‌ها دیگر تغییر نکنند، تکرار می‌شود.



شکل ۱۴: K-means

- هدف: نقاط داده در یک خوشه باید به یکدیگر شبیه باشند.
- در K-Means، این هدف به‌صورت تابع هزینه زیر بیان می‌شود:

$$J = \sum_{j=1}^{K} \sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu_j\|^2$$

- انتخاب f و $\mu = \{\mu_1, \mu_2, \ldots, \mu_K\}$ برای کمینه کردن تابع J.
- این مسئله NP-hard است. الگوریتم K-Means یک راه‌حل هیوریستیک است که تضمین نمی‌کند حتماً به راه‌حل بهینه برسد.
- ابتدا هر نمونه به نزدیک‌ترین مرکز تخصیص داده می‌شود:

$$f(x) := \arg\min_j \|x - \mu_j\|^2$$

- تخصیص هر نمونه تا زمانی که مرکز نزدیک‌تری پیدا نشود، ثابت می‌ماند.
- هر بار که یک نمونه بازتخصیص می‌شود، مجموع فاصله بین نمونه‌ها و مراکز آن‌ها کاهش می‌یابد.
- تعداد تخصیص‌های ممکن نمونه به مرکز محدود است.
- الگوریتم زمانی خاتمه می‌یابد که هیچ نمونه‌ای مرکز تخصیص‌یافته خود را تغییر ندهد.
- در مرحله به‌روزرسانی، با ثابت نگه داشتن $f(x)$، تابع J به‌صورت یک تابع درجه دوم از μ_j (مانند مجموع مربعات خطا) است و با مشتق‌گیری می‌توان آن را کمینه کرد:

$$\frac{\partial J}{\partial \mu_j} = 0 \implies \sum_{x^{(i)} \in C_j} -2\left(x^{(i)} - \mu_j\right) = 0$$

- این به این معناست که هر μ_j باید به‌عنوان میانگین خوشه C_j به‌روزرسانی شود:

$$\mu_j = \frac{\sum_{x^{(i)} \in C_j} x^{(i)}}{|C_j|}$$

- برای هر خوشه، میانگین نقاط آن خوشه مقدار مجموع فاصله‌های مربعی را کمینه می‌کند.

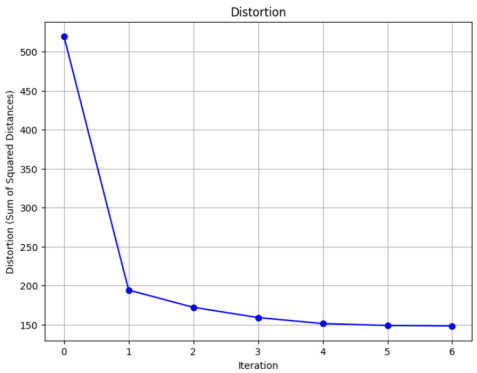
- برای خوشه‌ی C_j، اگر μ'_j مرکز قبلی باشد، داریم:

$$\sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu'_j\|^2 \ \geq \ \sum_{x^{(i)} \in C_j} \|x^{(i)} - \mu_j\|^2$$

یعنی انتخاب میانگین جدید هرگز مقدار خطا را افزایش نمی‌دهد.

- در نتیجه، $J_{\text{new}} \leq J_{\text{old}}$ خواهد بود.
- از آنجایی که تابع هزینه J غیرمنفی است و تعداد تقسیم‌بندی‌های ممکن نیز محدود می‌باشد، دنباله مقادیر J نمی‌تواند تا بی‌نهایت کاهش یابد.
- بنابراین الگوریتم K-Means در نهایت همگرا می‌شود.
- ویژگی‌های همگرایی الگوریتم K-Means نخستین‌بار توسط مک‌کوئین (MacQueen) در سال ۱۹۶۷ بررسی شد.

cost همان طوری که در تصویر مشاهده می شود در نهایت کمتر و کمتر می شود. و به یک نقطه صفر می رسد که بهترین جواب را پیدا خواهد کرد.



شکل ۱۵: نمودار تابع هزینه در k-means