دانشکده مهندسی کامپیوتر

ارائه یادگیری ماشین

بهنود قربانی - علیرضا لیمویی - امیررضا ارجمند

فهرست مطالب

۱ مقدمه ۲
۱.۱ تعریف یادگیری ماشین ۲
۱.۱.۱ یادگیری ماشین چیست؟ ۲
۲.۱.۱ مفاهیم کلیدی یادگیری ماشین ۲
۲.۱ هوش مصنوعی و طبیعی ۲
۱.۲.۱ تفاوت بین هوش مصنوعی و هوش طبیعی ۲
۲.۲.۱ نحوه یادگیری انسان‌ها در مقابل AI ۲
۳.۱ الگوریتم های یادگیری ماشین ۳
۱.۳.۱ Supervised learning ۳
۲.۳.۱ Unsupervised learning ۳
۳.۳.۱ Reinforcement learning ۳
۴.۱ یادگیری ماشین چگونه اتفاق می افتد ۴
۵.۱ مدل یادگیری ماشین چیست ۵
۶.۱ کاربرد های یادگیری ماشین ۶
۱.۶.۱ بهداشت و درمان ۶
۲.۶.۱ مالی ۶
۳.۶.۱ بازاریابی و فروش ۷
۴.۶.۱ خرده‌فروشی ۷
۷.۱ مباحث اخلاقی ۷
۱.۷.۱ شفافیت ۷
۲.۷.۱ عدالت و بی‌طرفی ۷
۳.۷.۱ حریم خصوصی و امنیت ۷
۴.۷.۱ فایده و رفاه ۷

۲ رگرسیون ۸
۱.۲ مقدمه ۸
۲.۲ کاربردهای رگرسیون ۸
۳.۲ ارزیابی مدل ۸
۴.۲ یادگیری مدل چگونه رخ می‌دهد ۹
۵.۲ بیش‌برازش و کم‌برازش (Overfitting & Underfitting) ۱۰
۶.۲ نقاط پرت (Outliers) ۱۲

۳ دسته بندی ۱۳
۱.۳ مقدمه ۱۳
۲.۳ انواع دسته بندی ۱۳
۱.۲.۳ دسته بندی دودوی ۱۳
۲.۲.۳ دسته بندی چند کلاسه ۱۳
۳.۲.۳ دسته بندی چند دسته‌ای ۱۵
۳.۳ نگاهی عمیق تر به دسته بندی دودوی ۱۵
۴.۳ ارزیابی مدل ۲۲

۱ مقدمه

۱.۱ تعریف یادگیری ماشین

۱.۱.۱ یادگیری ماشین چیست؟

یادگیری ماشین شاخه‌ای از هوش مصنوعی (AI) است که بر ساخت سیستم‌هایی تمرکز دارد که قادر به یادگیری و تصمیم‌گیری بر اساس داده‌ها هستند. برخلاف برنامه‌نویسی سنتی که در آن توسعه‌دهنده دستورات صریحی برای سیستم می‌نویسد، یادگیری ماشین شامل ایجاد الگوریتم‌هایی است که به سیستم اجازه می‌دهد الگوها را شناسایی کند، پیش‌بینی کند و با تجربه بهبود یابد.

۲.۱.۱ مفاهیم کلیدی یادگیری ماشین

الگوریتم‌ها: هسته یادگیری ماشین، الگوریتم‌ها مدل‌های ریاضی هستند که داده‌ها را پردازش کرده و از آن‌ها یاد می‌گیرند. الگوریتم‌های رایج شامل رگرسیون خطی، درخت‌های تصمیم‌گیری و شبکه‌های عصبی هستند.

داده‌ها: سوخت یادگیری ماشین. می‌تواند ساختاریافته (مثل پایگاه‌های داده و صفحات گسترده) یا غیرساختاریافته (مثل متن، تصاویر و ویدیوها) باشد. کیفیت و کمیت داده‌ها به شدت بر عملکرد مدل تأثیر می‌گذارد.

آموزش: فرآیند وارد کردن داده‌ها به الگوریتم یادگیری ماشین برای کمک به یادگیری آن. در طول آموزش، الگوریتم پارامترهای خود را تنظیم می‌کند تا خطاها را به حداقل برساند.

آزمون: پس از آموزش، مدل با داده‌های جدید آزمایش می‌شود تا دقت و عملکرد آن ارزیابی شود. این کمک می‌کند تا مشخص شود مدل چقدر به داده‌های نادیده تعمیم می‌دهد.

۲.۱ هوش مصنوعی و طبیعی

۱.۲.۱ تفاوت بین هوش مصنوعی و هوش طبیعی

هوش مصنوعی (AI) و هوش طبیعی (NI) هر دو به توانایی یادگیری و حل مسائل اشاره دارند، اما با تفاوت‌های بنیادینی در نحوه عملکرد و مکانیسم‌های زیرساختی همراه هستند:

هوش مصنوعی (AI) منشأ: توسط انسان‌ها ساخته و برنامه‌ریزی می‌شود.

ساختار: بر پایه الگوریتم‌ها و مدل‌های ریاضی استوار است.

یادگیری: از طریق پردازش حجم وسیعی از داده‌ها و استفاده از تکنیک‌های یادگیری ماشین.

انطباق‌پذیری: توانایی انطباق با شرایط جدید و به‌روزرسانی مدل‌ها براساس داده‌های جدید.

محدودیت‌ها: در محدوده داده‌های ورودی و طراحی الگوریتم‌ها عمل می‌کند؛ فاقد خلاقیت و درک واقعی است.

هدف: انجام وظایف خاص و حل مسائل مشخص.

هوش طبیعی (NI) منشأ: در موجودات زنده، به‌ویژه انسان‌ها، ذاتی است.

ساختار: بر پایه مغز و سیستم عصبی استوار است.

یادگیری: از طریق تجربه، آموزش، تعاملات اجتماعی و فرآیندهای شناختی.

انطباق‌پذیری: بسیار انعطاف‌پذیر، قادر به یادگیری و تطبیق در محیط‌های متغیر و نامشخص.

محدودیت‌ها: محدود به توانایی‌های فیزیکی و روانی انسان‌ها؛ ممکن است با خطا و تعصب همراه باشد.

هدف: بقا، تطبیق با محیط، خلاقیت و درک عمیق از جهان.

۲.۲.۱ نحوه یادگیری انسان‌ها در مقابل **AI**

یادگیری انسان‌ها

تجربه مستقیم: انسان‌ها از طریق تجربه‌های شخصی و تعاملات با محیط یاد می‌گیرند. مشاهده و تقلید: یادگیری از طریق مشاهده رفتارهای دیگران و تقلید آن‌ها. آموزش رسمی: از طریق آموزش‌های ساختاریافته مانند مدارس و دانشگاه‌ها. تکرار و تمرین: تمرین مداوم مهارت‌ها تا تسلط بر آن‌ها. انعکاس و تفکر: توانایی تأمل و ارزیابی تجربیات گذشته برای بهبود یادگیری. احساسات و انگیزه‌ها: نقش مهمی در فرآیند یادگیری دارند، به عنوان مثال، علاقه و انگیزه می‌توانند یادگیری را تسریع کنند.

یادگیریAI

داده‌های برچسب‌دار: مدل‌های AI از طریق داده‌های برچسب‌دار (supervised learning) آموزش می‌بینند، که به آن‌ها کمک می‌کند الگوها و روابط را شناسایی کنند. الگوریتم‌ها: استفاده از الگوریتم‌های یادگیری ماشین مانند رگرسیون، درخت‌های تصمیم، و شبکه‌های عصبی. بازخورد: مدل‌های AI از طریق بازخورد اصلاح می‌شوند؛ مثلاً الگوریتم‌های یادگیری تقویتی از طریق پاداش و جریمه بهبود می‌یابند. تکرار و به‌روزرسانی: مدل‌ها با داده‌های جدید به‌روزرسانی و مجدداً آموزش داده می‌شوند تا دقت و کارایی آن‌ها افزایش یابد. پردازش موازی: استفاده از قدرت محاسباتی بالا برای پردازش حجم وسیعی از داده‌ها به طور همزمان.

۳.۱ الگوریتم های یادگیری ماشین

۱.۳.۱ Supervised learning

یادگیری نظارت‌شده چیست؟

یادگیری نظارت‌شده (Supervised Learning) یکی از روش‌های یادگیری ماشین است که در آن مدل با استفاده از داده‌های برچسب‌دار آموزش داده می‌شود. این به این معناست که هر ورودی در مجموعه داده‌ها با خروجی صحیح (برچسب) مرتبط است. هدف از یادگیری نظارت‌شده این است که مدل بتواند با دیدن داده‌های جدید و نامعلوم، خروجی‌های صحیح را پیش‌بینی کند.

مراحل یادگیری نظارت‌شده

جمع‌آوری داده‌ها: جمع‌آوری مجموعه‌ای از داده‌های ورودی و خروجی‌های مرتبط با آن‌ها. این داده‌ها به عنوان داده‌های آموزشی استفاده می‌شوند . تقسیم داده‌ها: تقسیم داده‌ها به دو مجموعه: داده‌های آموزشی (training data) و داده‌های آزمون(test data).

انتخاب مدل: انتخاب الگوریتم مناسب برای یادگیری. مثال‌هایی از الگوریتم‌ها عبارتند از رگرسیون خطی، درخت تصمیم‌گیری، ماشین بردار پشتیبان (SVM)، و شبکه‌های عصبی. آموزش مدل: تغذیه داده‌های آموزشی به مدل و تنظیم پارامترهای آن به گونه‌ای که خطا بین خروجی پیش‌بینی شده و خروجی واقعی به حداقل برسد.

ارزیابی مدل: استفاده از داده‌های آزمون برای ارزیابی عملکرد مدل و اطمینان از این که مدل به خوبی تعمیم یافته است.

استفاده از مدل: استفاده از مدل آموزش دیده برای پیش‌بینی خروجی‌های داده‌های جدید و نامعلوم.

۲.۳.۱ Unsupervised learning

یادگیری بدون نظارت چیست؟

یادگیری بدون نظارت (Unsupervised Learning) یکی از روش‌های یادگیری ماشین است که در آن مدل بدون استفاده از داده‌های برچسب‌دار آموزش داده می‌شود. در یادگیری بدون نظارت، داده‌ها تنها شامل ورودی‌ها هستند و خروجی‌های مشخصی به مدل داده نمی‌شود. هدف از این نوع یادگیری کشف الگوها، ساختارها یا ویژگی‌های مهم در داده‌ها است.

مراحل یادگیری بدون نظارت

جمع‌آوری داده‌ها: جمع‌آوری مجموعه‌ای از داده‌های ورودی بدون برچسب . انتخاب مدل: انتخاب الگوریتم مناسب برای یادگیری. مثال‌هایی از الگوریتم‌ها عبارتند از خوشه‌بندی (clustering) و کاهش ابعاد (dimensionality reduction).

آموزش مدل: تغذیه داده‌ها به مدل و استفاده از الگوریتم‌های یادگیری بدون نظارت برای شناسایی الگوها و ساختارها.

تفسیر نتایج: تفسیر و تحلیل نتایج به دست آمده برای استخراج دانش مفید از داده‌ها.

استفاده از مدل: استفاده از مدل آموزش دیده برای اعمال به داده‌های جدید و شناسایی الگوهای مشابه.

۳.۳.۱ Reinforcement learning

یادگیری تقویتی چیست؟

یادگیری تقویتی (Reinforcement Learning) یکی از روش‌های یادگیری ماشین است که در آن یک عامل (Agent) از طریق تعامل با محیط، یاد می‌گیرد که چگونه با انجام یک سری از اعمال (Actions) در موقعیت‌های مختلف (States) بیشترین پاداش (Reward) ممکن را دریافت کند. هدف یادگیری تقویتی یافتن سیاست بهینه (Optimal Policy) است که حداکثر پاداش بلندمدت را تضمین می‌کند.

عناصر کلیدی یادگیری تقویتی

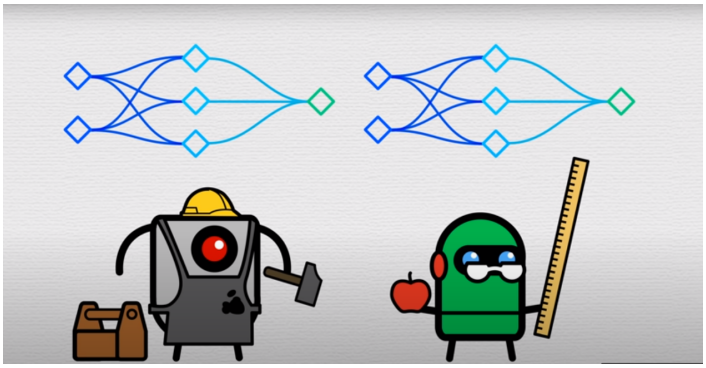
عامل (Agent): موجودیتی که تصمیم‌گیری می‌کند و اعمال را انجام می‌دهد.

محیط (Environment): جایی که عامل در آن عمل می‌کند و از آن پاداش و بازخورد دریافت می‌کند.

وضعیت‌ها (States): نمایشگر وضعیت فعلی محیط.
اعمال (Actions): مجموعه‌ای از اقدامات که عامل می‌تواند انجام دهد.
پاداش (Reward): بازخوردی که عامل پس از انجام یک عمل دریافت می‌کند. پاداش می‌تواند مثبت یا منفی باشد.
سیاست (Policy): استراتژی‌ای که عامل برای انتخاب اعمال بر اساس وضعیت‌ها استفاده می‌کند.
تابع ارزش (Value Function): تخمین ارزش بلندمدت وضعیت‌ها یا وضعیت-عمل‌ها.
مراحل یادگیری تقویتی
آغاز وضعیت: عامل در یک وضعیت اولیه شروع می‌کند.
انتخاب عمل: عامل بر اساس سیاست فعلی یک عمل را انتخاب می‌کند.
اجرای عمل و دریافت پاداش: عامل عمل را اجرا کرده و پاداش فوری را دریافت می‌کند. همچنین وضعیت جدید را مشاهده می‌کند.
به‌روزرسانی سیاست: عامل سیاست خود را بر اساس تجربه جدید به‌روزرسانی می‌کند تا پاداش بلندمدت خود را بهبود بخشد.
تکرار: مراحل فوق تا زمانی تکرار می‌شوند که سیاست بهینه یافت شود یا زمان معین شده پایان یابد.

۴.۱ یادگیری ماشین چگونه اتفاق می افتد

مثال "ربات سازنده" و "ربات معلم "
سناریو
فرض کنید که ما دو ربات داریم: ربات سازنده (Builder Bot) و ربات معلم (Teacher Bot). هدف این است که ربات سازنده یاد بگیرد چگونه سازه‌های مختلفی را با استفاده از قطعات ساختمانی بسازد. ربات معلم نقش مربی را ایفا می‌کند و داده‌های آموزشی را برای ربات سازنده فراهم می‌کند.
مراحل انجام کار
جمع‌آوری داده‌ها توسط ربات معلم:
ربات معلم یک سری سازه‌های کامل شده با قطعات ساختمانی دارد. برای هر سازه، یک مجموعه از قطعات (ورودی‌ها) و ترتیب ساخت (برچسب‌ها) موجود است. داده‌های آموزشی شامل مجموعه‌های قطعات به همراه ترتیب صحیح ساخت آن‌ها می‌شود.
تقسیم داده‌ها:
داده‌های جمع‌آوری شده به دو قسمت تقسیم می‌شوند: داده‌های آموزشی برای آموزش ربات سازنده و داده‌های آزمون برای ارزیابی عملکرد آن.
انتخاب مدل:
یک مدل یادگیری نظارت‌شده، مثل شبکه عصبی یا درخت تصمیم‌گیری، برای آموزش انتخاب می‌شود.
آموزش مدل:
ربات سازنده داده‌های آموزشی را از ربات معلم دریافت می‌کند. مدل یادگیری با استفاده از این داده‌ها آموزش داده می‌شود تا الگوها و ترتیب‌های صحیح ساخت سازه‌ها را بیاموزد.
ارزیابی مدل:
ربات سازنده مدل آموزش دیده را با داده‌های آزمون ارزیابی می‌کند تا عملکرد آن را بسنجد. دقت مدل در پیش‌بینی ترتیب صحیح ساخت سازه‌ها بررسی می‌شود. استفاده از مدل:
ربات سازنده حالا می‌تواند با استفاده از مدل آموزش دیده، سازه‌های جدیدی را با ترتیب صحیح بسازد. ربات سازنده قادر است با دریافت مجموعه‌ای از قطعات، ترتیب صحیح ساخت را پیش‌بینی کرده و سازه مورد نظر را بسازد. مثال عددی
فرض کنید داده‌های آموزشی شامل ۱۰ نوع سازه مختلف است که هر کدام از آن‌ها با قطعات مختلفی ساخته شده‌اند. به عنوان مثال:
سازه ۱: قطعاتA، B، C (ترتیب ساخت: $C \rightarrow B \rightarrow A$)
سازه ۲: قطعات D، E، F (ترتیب ساخت: $F \rightarrow E \rightarrow D$)
ربات معلم این داده‌ها را به ربات سازنده می‌دهد. ربات سازنده مدل خود را با استفاده از این داده‌ها آموزش می‌دهد. سپس، ربات سازنده با دریافت مجموعه‌ای از قطعات جدید (مثلاً G، H، I) قادر خواهد بود ترتیب صحیح ساخت (مثلاً $I \rightarrow H \rightarrow G$) را پیش‌بینی کند و سازه را بسازد.
خلاصه اختار: استفاده از منطق، قواعد و الگوریتم‌های قطعی (Deterministic).
منطق: بر پایه برنامه‌نویسی صریح و تعریف قواعد دستوری است.
مثال‌ها: سیستم‌های خبره (Expert Systems)، موتورهای قانون‌گذاری (Rule Engines)، منطق فازی (Fuzzy Logic).



شکل ۱: Builder robot and Teacher robot

قابلیت یادگیری: به طور کلی، سیستم‌های هوش مصنوعی کلاسیک توانایی یادگیری از داده‌ها را ندارند و بر اساس قواعد از پیش تعیین‌شده عمل می‌کنند.

کاربردها: اتوماسیون فرآیندها، سیستم‌های تشخیص خطا، بازی‌های شطرنج اولیه، سیستم‌های توصیه‌گر ساده.

۲. یادگیری ماشین (Machine Learning) یادگیری ماشین زیرمجموعه‌ای از هوش مصنوعی است که در آن سیستم‌ها از طریق داده‌ها و تجربیات یاد می‌گیرند و بهبود می‌یابند.

ساختار: استفاده از الگوریتم‌ها و مدل‌های آماری برای تحلیل و یادگیری از داده‌ها.

منطق: بر پایه الگوهای موجود در داده‌ها و تجربه‌های گذشته است.

مثال‌ها: رگرسیون خطی و لجستیک، درخت‌های تصمیم‌گیری، شبکه‌های عصبی، ماشین‌های بردار پشتیبان (SVM).

قابلیت یادگیری: سیستم‌های یادگیری ماشین با استفاده از داده‌ها آموزش می‌بینند و توانایی بهبود و تطبیق با داده‌های جدید را دارند.

کاربردها: تشخیص تصویر و صدا، پردازش زبان طبیعی، توصیه‌گرهای پیچیده، سیستم‌های پیش‌بینی، خودروهای خودران.

تفاوت‌های کلیدی

۱. روش‌های حل مسئله

هوش مصنوعی کلاسیک: مشکلات را با استفاده از قواعد و منطق صریح حل می‌کند. به عنوان مثال، یک سیستم خبره پزشکی ممکن است بر اساس مجموعه‌ای از قواعد از پیش تعریف‌شده برای تشخیص بیماری‌ها عمل کند.

یادگیری ماشین: مشکلات را با استفاده از الگوهای موجود در داده‌ها حل می‌کند. به عنوان مثال، یک مدل یادگیری ماشین می‌تواند با تحلیل داده‌های پزشکی، الگوهایی را شناسایی کرده و پیش‌بینی‌هایی درباره بیماری‌ها انجام دهد.

۲. انعطاف‌پذیری و تطبیق‌پذیری

هوش مصنوعی کلاسیک: انعطاف‌پذیری کمتری دارد زیرا بر اساس قواعد سخت‌گیرانه‌ای که از پیش تعریف شده‌اند عمل می‌کند.

یادگیری ماشین: انعطاف‌پذیری بیشتری دارد زیرا می‌تواند از داده‌های جدید یاد بگیرد و خود را با تغییرات تطبیق دهد.

۳. نیاز به داده‌ها

هوش مصنوعی کلاسیک: به داده‌های برچسب‌دار و زیادی نیاز ندارد زیرا بر اساس قواعد ثابت عمل می‌کند.

یادگیری ماشین: به داده‌های برچسب‌دار و حجیم نیاز دارد تا بتواند الگوها را شناسایی کند و مدل‌های دقیقی بسازد.

۴. عملکرد در شرایط نامعین

هوش مصنوعی کلاسیک: عملکرد ضعیفی در شرایط نامعین و پیچیده دارد زیرا به قواعد از پیش تعریف‌شده وابسته است.

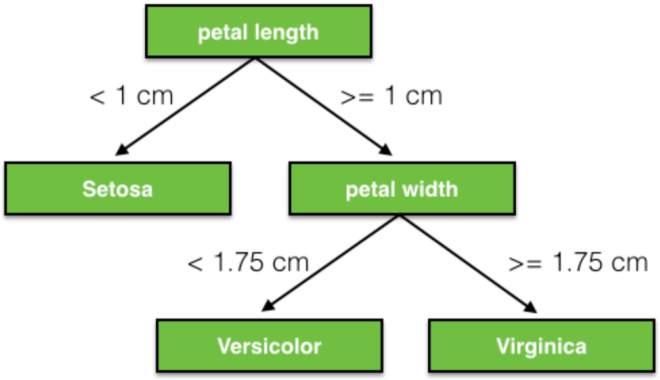
یادگیری ماشین: عملکرد بهتری در شرایط نامعین و پیچیده دارد زیرا می‌تواند از داده‌ها یاد بگیرد و خود را تطبیق دهد.

۵.۱ مدل یادگیری ماشین چیست

یک مدل یادگیری ماشین یک نمایش ریاضی یا الگوریتم است که برای یادگیری الگوها از داده‌ها و انجام پیش‌بینی‌ها یا تصمیم‌گیری‌ها بر اساس داده‌های جدید طراحی شده است. این مدل از طریق فرآیندی به نام آموزش ساخته می‌شود که در آن مدل با داده‌های تاریخی مواجه شده و توانایی شناسایی روابط و الگوها را پیدا می‌کند.

اجزای اصلی یک مدل یادگیری ماشین:

داده‌ها:



شکل ۲: Desicion tree

داده‌های آموزشی: برای آموزش مدل استفاده می‌شود و شامل ویژگی‌های ورودی و برچسب‌های خروجی متناظر است.
داده‌های اعتبارسنجی: برای تنظیم ابرپارامترهای مدل و جلوگیری از بیش‌برازش استفاده می‌شود.
داده‌های آزمایشی: برای ارزیابی عملکرد مدل بر روی داده‌های نادیده استفاده می‌شود.
ویژگی‌ها: متغیرها یا ورودی‌هایی که برای انجام پیش‌بینی‌ها استفاده می‌شوند. ویژگی‌ها معمولاً عددی هستند، اما می‌توانند شامل داده‌های دسته‌ای نیز باشند.
برچسب‌ها/اهداف: خروجی‌ها یا نتایجی که مدل قصد دارد پیش‌بینی کند.
انواع مدل‌های یادگیری ماشین:
رگرسیون خطی: نتایج پیوسته را بر اساس روابط خطی بین ویژگی‌ها و هدف پیش‌بینی می‌کند.
رگرسیون لجستیک: نتایج باینری را با استفاده از یک تابع لجستیک پیش‌بینی می‌کند.
درخت‌های تصمیم‌گیری: از یک مدل درخت‌مانند برای وظایف طبقه‌بندی یا رگرسیون استفاده می‌کند.
جنگل تصادفی: مجموعه‌ای از درخت‌های تصمیم‌گیری برای بهبود دقت و کاهش بیش‌برازش.
ماشین‌های بردار پشتیبان (SVM): صفحه فوق بهینه را برای جدا کردن دسته‌های مختلف پیدا می‌کند.
شبکه‌های عصبی: متشکل از لایه‌هایی از گره‌های متصل (نورون‌ها) که الگوهای پیچیده را یاد می‌گیرند.
خوشه‌بندی K-میانگین: یک روش بدون نظارت برای تقسیم‌بندی داده‌ها به خوشه‌ها بر اساس شباهت.
تحلیل مؤلفه‌های اصلی (PCA): تکنیکی برای کاهش ابعاد داده‌ها در حالی که اکثر واریانس را حفظ می‌کند.

۶.۱ کاربرد های یادگیری ماشین

یادگیری ماشین دارای کاربردهای گسترده‌ای در صنایع و زمینه‌های مختلف است. در اینجا برخی از نمونه‌های برجسته آمده است:

۱.۶.۱ بهداشت و درمان

تشخیص بیماری: مدل‌های یادگیری ماشین می‌توانند تصاویر پزشکی (مانند اشعه ایکس، MRI) را تحلیل کرده و بیماری‌هایی مانند سرطان، ذات‌الریه و رتینوپاتی دیابتی را تشخیص دهند.
تحلیل پیش‌بینی: پیش‌بینی نتایج بیماران و عوارض احتمالی، که منجر به مراقبت‌های بهداشتی پیشگیرانه می‌شود.
کشف دارو: تسریع فرآیند کشف داروهای جدید با پیش‌بینی رفتار ترکیبات مختلف.

۲.۶.۱ مالی

تشخیص تقلب: شناسایی تراکنش‌های تقلبی در زمان واقعی با شناسایی الگوهایی که از رفتار عادی انحراف دارند.
معاملات الگوریتمی: استفاده از داده‌های تاریخی برای توسعه استراتژی‌های معاملاتی و تصمیم‌گیری‌های معاملاتی در زمان واقعی.
امتیازدهی اعتباری: ارزیابی اعتبار افراد با تحلیل تاریخچه و رفتار مالی آن‌ها.

۳.۶.۱ بازاریابی و فروش

بخش‌بندی مشتریان: گروه‌بندی مشتریان بر اساس رفتار و ترجیحات آن‌ها برای تنظیم استراتژی‌های بازاریابی.
سیستم‌های توصیه‌گر: پیشنهاد محصولات، فیلم‌ها یا محتوا به کاربران بر اساس رفتار گذشته آن‌ها (مانند نتفلیکس، آمازون).
پیش‌بینی ترک مشتری: پیش‌بینی اینکه کدام مشتریان احتمالاً سرویس را ترک می‌کنند و اجرای استراتژی‌های نگه‌داری.

۴.۶.۱ خرده‌فروشی

مدیریت موجودی: پیش‌بینی تقاضا برای محصولات به منظور بهینه‌سازی سطح موجودی و کاهش هدررفت.
بهینه‌سازی قیمت: تنظیم قیمت‌های دینامیک بر اساس عواملی مانند تقاضا، رقابت و رفتار مشتری.
خدمات مشتری: پیاده‌سازی چت‌بات‌ها و دستیارهای مجازی برای پاسخگویی به سوالات مشتریان.

۷.۱ مباحث اخلاقی

اخلاق هوش مصنوعی (AI) به مجموعه‌ای از اصول و رهنمودها اشاره دارد که هدف آن‌ها اطمینان از استفاده مسئولانه و اخلاقی از فناوری‌های هوش مصنوعی است. پنج ستون اصلی اخلاق هوش مصنوعی عبارتند از:

۱.۷.۱ شفافیت

توضیح‌پذیری: اطمینان از اینکه تصمیمات و خروجی‌های سیستم‌های هوش مصنوعی قابل توضیح و درک برای انسان‌ها باشند. کاربران و ذی‌نفعان باید بتوانند بفهمند که چگونه و چرا یک مدل هوش مصنوعی به نتایج خاصی می‌رسد.
قابل مشاهده بودن: فرآیندهای داده‌پردازی و الگوریتم‌های مورد استفاده در سیستم‌های هوش مصنوعی باید به طور واضح و قابل مشاهده باشند تا از سوء استفاده جلوگیری شود.

۲.۷.۱ عدالت و بی‌طرفی

عدم تبعیض: اطمینان از اینکه سیستم‌های هوش مصنوعی بدون توجه به عوامل مانند نژاد، جنسیت، مذهب، یا وضعیت اجتماعی-اقتصادی، نتایج عادلانه و بی‌طرفانه ارائه دهند. این شامل شناسایی و کاهش هر گونه تعصب و تبعیض در داده‌های آموزشی و الگوریتم‌ها است.
دسترسی برابر: تلاش برای اطمینان از اینکه مزایای هوش مصنوعی به صورت عادلانه در میان افراد و گروه‌ها توزیع می‌شود.

۳.۷.۱ حریم خصوصی و امنیت

حفاظت از داده‌ها: اطمینان از اینکه داده‌های شخصی کاربران به طور ایمن ذخیره و پردازش می‌شوند و تنها برای اهداف مجاز استفاده می‌شوند. این شامل پیاده‌سازی روش‌های قوی برای حفاظت از داده‌ها در برابر دسترسی غیرمجاز و نقض امنیتی است.
حریم خصوصی: احترام به حریم خصوصی کاربران و اطمینان از اینکه سیستم‌های هوش مصنوعی با قوانین و مقررات مربوط به حفاظت از داده‌های شخصی مطابقت دارند.
۴. مسئولیت‌پذیری
پاسخگویی: اطمینان از اینکه سازندگان و استفاده‌کنندگان سیستم‌های هوش مصنوعی مسئولیت نتایج و پیامدهای ناشی از استفاده از این فناوری‌ها را بر عهده می‌گیرند. این شامل ایجاد سازوکارهایی برای پاسخگویی در صورت وقوع خطا یا سوء استفاده است.
ارزیابی و نظارت: پیاده‌سازی فرآیندهای مستمر برای ارزیابی عملکرد و تاثیر سیستم‌های هوش مصنوعی و نظارت بر رعایت اصول اخلاقی.

۴.۷.۱ فایده و رفاه

بهره‌وری انسانی: اطمینان از اینکه سیستم‌های هوش مصنوعی به بهبود کیفیت زندگی و رفاه انسانی کمک می‌کنند و به هیچ‌وجه ضرری به انسان‌ها نمی‌رسانند. این شامل طراحی سیستم‌هایی است که به نفع جامعه و ارتقاء بهره‌وری انسانی باشد.
رعایت حقوق انسانی: تضمین اینکه توسعه و استفاده از هوش مصنوعی مطابق با اصول حقوق بشر باشد و به حفاظت و ارتقاء حقوق و آزادی‌های انسانی کمک کند.
این اصول به منظور هدایت توسعه، استقرار و استفاده از هوش مصنوعی به نحوی که منجر به نتایج مثبت و اجتناب از پیامدهای منفی شود، تدوین شده‌اند. رعایت این اصول می‌تواند به ایجاد اعتماد و پذیرش بیشتر در میان کاربران و جامعه منجر شود.

۲ رگرسیون

۱.۲ مقدمه

هدف از رگرسیون، پیدا کردن رابطه یک یا چند متغیر مستقل یا نابسته، و یک متغیر وابسته است؛ برای مثال پیدا کردن قیمت خانه بر اساس متراژ، موقعیت مکانی، سال ساخت و غیره. یا تخمین قیمت سهام با استفاده از فاکتورهایی از جمله ارائه و تقاضا، نرخ بهره، عوامل سیاسی و غیره.

اگر بخواهیم به معنای کلمه رگرسیون بپردازیم، کلمه رگرسیون ریشه لاتین دارد که به معنای بازگشت می‌باشد و این به همان مفهوم بیان تغییرات یک متغیر بر اساس اطلاعات متغیر/متغیرهای دیگر اشاره دارد.

۲.۲ کاربردهای رگرسیون

۱. پیش‌بینی: رگرسیون برای پیش‌بینی مقادیر متغیر وابسته بر اساس متغیرهای مستقل استفاده می‌شود. به عنوان مثال، پیش‌بینی قیمت خانه بر اساس ویژگی‌هایی مانند متراژ، تعداد اتاق‌ها و موقعیت جغرافیایی.

۲. تحلیل روابط: رگرسیون به تحلیل و فهمیدن روابط بین متغیرها کمک می‌کند. به عنوان مثال، بررسی تأثیر تبلیغات بر فروش محصولات.

۳. مدل‌سازی: رگرسیون برای ساخت مدل‌های ریاضی استفاده می‌شود که رفتار سیستم‌های پیچیده را توصیف می‌کنند. به عنوان مثال، مدل‌سازی رشد جمعیت یا مدل‌سازی تغییرات آب و هوا.

۴. کنترل کیفیت: در صنایع مختلف، رگرسیون برای کنترل کیفیت و بهبود فرآیندها به کار می‌رود. به عنوان مثال، تحلیل داده‌های تولید برای شناسایی عوامل مؤثر بر کیفیت محصول.

۵. تحلیل سری‌های زمانی: رگرسیون برای تحلیل داده‌های سری زمانی و پیش‌بینی روندهای آینده استفاده می‌شود. به عنوان مثال، پیش‌بینی تقاضای برق در یک بازه زمانی مشخص.

۶. تحلیل بقا: در علوم پزشکی و اجتماعی، رگرسیون برای تحلیل داده‌های بقا و بررسی عواملی که بر زمان بقا تأثیر می‌گذارند، استفاده می‌شود. به عنوان مثال، بررسی تأثیر درمان‌های مختلف بر بقای بیماران.

برای این کار نیاز به دسترسی به داده‌های نمونه متشکل از متغیرهای مستقل و وابسته را داریم تا بتوانیم تاثیر هرکدام را روی خروجی الگوریتم بدست بیاوریم. این کار آموزش دادن مدل تلاقی می‌شود.

رگرسیون در واقع یک الگوریتم واحد نیست بلکه گروهی از تحلیل‌ها و الگوریتم‌ها که به دنبال حل مسائلی که در بالا ذکر شد هستند را الگوریتم‌های رگرسیون می‌نامیم. رگرسیون خطی معروف‌ترین و شناخته شده‌ترین این الگوریتم‌ها می‌باشد که معمولا در کلاس‌ها و کتب یادگیری ماشین به عنوان اولین الگوریتم تدریس می‌شوند. ما هم اینجا به یادگیری آن می‌پردازیم. بسیاری از الگوریتم‌های دیگر رگرسیون از رگرسیون خطی الگو می‌گیرند و با یادگیری رگرسیون خطی فهم آن‌ها نیز آسان‌تر می‌شود. برای مثال فرض کنید می‌خواهیم قیمت خانه را بر اساس متراژ آن تخمین بزنیم. اطلاعات تعدادی از خانه‌ها را داریم که در شکل ۷ به صورت نقاط آبی نمایان شده‌اند. با استفاده از اطلاعاتی که در اختیار داریم می‌توانیم خطی رسم کنیم که به ما قیمت خانه را برای هر متراژی مشخص کند.

مانند هر خط دیگری، خطی که رسم کردیم دارای یک شیب و یک عرض از مبدا می‌باشد که با استفاده از آن‌ها می‌توانیم معادله خط را بنویسیم. در اینجا شیب ما ۶۵۳.۷ و عرض از مبدا ۱۰۲۵۲۲.۴ می‌باشد و معادله خط به شرح زیر می‌باشد.

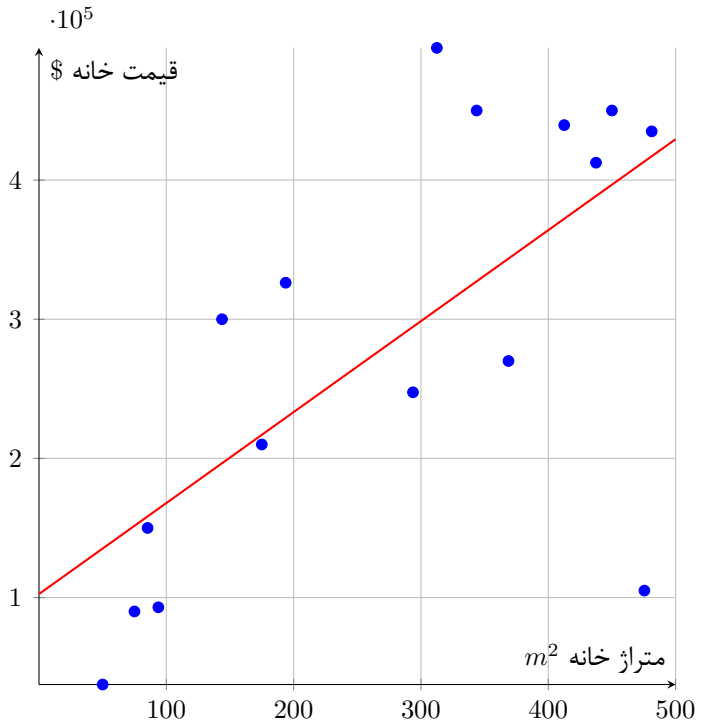
$$Price = 653.7 \times Size + 102,522.4 \quad (۱)$$

با استفاده از تابع ۱ می‌توانیم قیمت خانه‌ها را با هر متراژ دلخواهی بدست بیاوریم. برای نمونه:

$$Price(324.1m^2) = 653.7 \times 624.1 + 102,522.4 = 314,372.4 \quad (۲)$$

۳.۲ ارزیابی مدل

ممکن است افراد مختلف به روش‌های مختلف، به پارامترهای متفاوتی برسند. در آن صورت بهتر است کدام مدل را در نظر بگیریم؟ برای این کار باید بتوانیم مدلمان را ارزیابی کنیم. روش‌های مختلفی برای ارزیابی مدل رگرسیون خطی وجود دارد. یکی از ساده‌ترین و متداول‌ترین آن‌ها MSE می‌باشد. فرمول MSE به شرح زیر است:



شکل ۳: وابستگی قیمت خانه به متراژ آن

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{۳}$$

که در آن $\hat{Y}$ مقدار خروجی مدل ما و Y مقدار واقعی داده می‌باشد. طبیعتا هرچه فاصله مقدار پیش‌بینی شده توسط مدل ما با واقعیت بیشتر باشد، مقدار MSE نیز بیشتر می‌شود و به طور کلی مل همیشه به دنبال مدلی با MSE کمتر می‌رویم. توابع خطای دیگر از جمله MAE هم برای مدل‌های رگرسیون خطی ارائه شده اما بزرگترین مشکل آن‌ها مشتق‌پذیر نبودن است که در اینجا به این مساله نمی‌پردازیم.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{۴}$$

۴.۲ یادگیری مدل چگونه رخ می‌دهد

برای اینکه بهترین پارامترها را برای مدلمان پیدا کنیم، باید به دنبال کاهش تابع خطا که پیش‌تر برای ارزیابی مدر استفاده شد باشیم. یک راه تغییر دادن پارامترهای مدل (در اینجا شیب و عرض از مبدا) به صورت مکرر و با نرخ ثابت (که به آن learning rate گفته می‌شود و با نماد α یا η نمایش داده می‌شود.) می‌باشد؛ به نحوی که با هر تغییر خطای مدلمان کم شود.

برای این کار می‌توانیم از تابع خطای خود، نسبت به وزن‌ها (شیب و عرض از مبدا) مشتق بگیریم و با نرخ ثابت α به سمت شیب منفی آن برویم تا خطای مدل با هر تکرار کمتر شود. به مجموعه‌ی مشتقات تابع نسبت به وزن‌های آن gradient و به این الگوریتم پایین رفتن از مشتقات gradient descent گفته می‌شود. مشتق هر متغیر که به آن مشتق جزئی نیز گفته می‌شود با نماد ∂ نمایش داده می‌شود و مشتق کیری برای یک مدل با یک متغیر مستقل و یک متغیر وابسته به شرح زیر می‌باشد:

$$\begin{aligned}\frac{\partial f}{\partial w} &= \frac{1}{2N}\sum -2x((wx+b)-y) = \frac{-1}{N}\sum x((wx+b)-y)\\ \frac{\partial f}{\partial b} &= \frac{1}{2N}\sum -2((wx+b)-y) = \frac{-1}{N}\sum((wx+b)-y)\end{aligned} \tag{۵}$$

سپس برای به‌روزرسانی وزن‌ها از دو فرمول ۶ استفاده می‌کنیم:

$$w = w - \alpha \cdot \frac{\partial f}{\partial w}$$
$$b = b - \alpha \cdot \frac{\partial f}{\partial b} \quad (۶)$$

به طور کلی الگوریتم Gradient Descent به شکل زیر می‌باشد:

الگوریتم ۱.۲ Gradient Descent

ورودی: شیب اولیه w_0، عرض از مبدا اولیه b_0، تعداد تکرار T

خروجی: شیب w_T، عرض از مبدا b_T

۱. از $i = 1$ تا T

۲. $\frac{\partial f}{\partial w}$ را محاسبه کن

۳. $\frac{\partial f}{\partial b}$ را محاسبه کن

۴. $w_i = w_{i-1} - \alpha \cdot \frac{\partial f}{\partial w}$

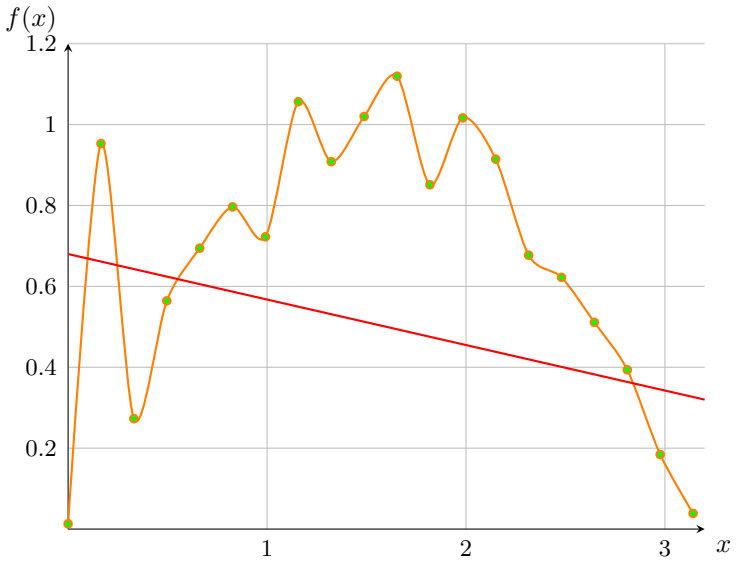
۵. $b_i = b_{i-1} - \alpha \cdot \frac{\partial f}{\partial b}$

۶. برگردان w_T, b_T

معمولا در یادگیری ماشین T را تعداد epoch خطاب می‌کنند.

۵.۲ بیش‌برازش و کم‌برازش (Overfitting & Underfitting)

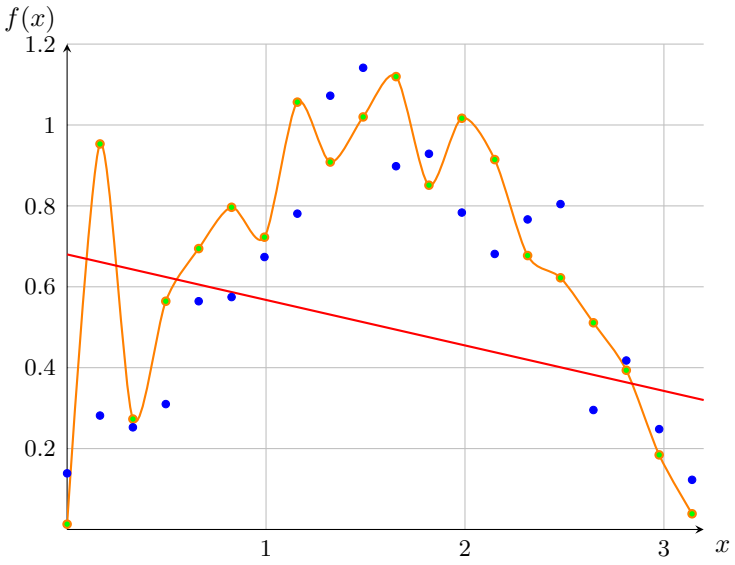
در شکل ۶ مشاهده می‌نمایید که دو مدل بر روی داده‌ها که با نقاط سبز مشخص شده‌اند آموزش داده‌ایم. مدلی که با خط قرمز نمایش داده شده است همان مدل خطی است که قبلا هم راجب آن صحبت شده. مدلی که با خط زرد رنگ نمایش داده شده را مدل Polynomial Regression می‌نامند که دارای جملاتی با درجه بیشتر از ۱ می‌باشد. به نظر شما کدام مدل بهتر است؟ می‌توانیم برای مقایسه خط قرمز را مدل ۱ بنامیم و خط زرد را مدل ۲ و سپس MSE را برای هرکدام محاسبه کنیم:



شکل ۴: مثالی از بیش‌برازش و کم‌برازش؛ خط قرمز مدل کم‌برازش شده است و خط زرد مدل بیش‌برازش شده

$$Error = 5.3$$
$$Error = 0.0$$

به نظر می‌رسد که مدل زرد رنگ خطای بسیار کمتری دارد. اما آیا در واقعیت هم همین قضیه صدق می‌کند؟ برای این کار می‌توانیم بخشی از داده‌های خود را کنار گذاشته و برای تست مدل استفاده کنیم که در به رنگ آبی نمایش دادیم:



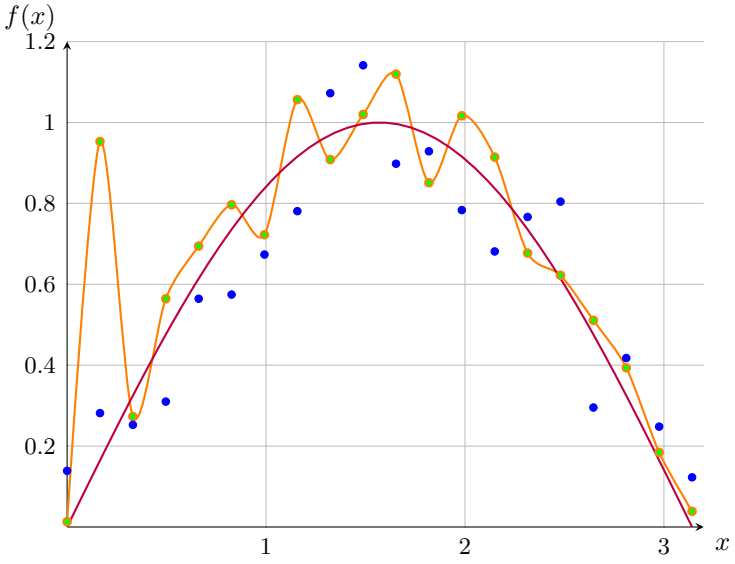
شکل ۵: استفاده از داده‌های تست برای ارزیابی مدل

$$Error_{train} = 5.3, Error_{test} = 5.6$$
$$Error_{train} = 0.0, Error_{test} = 3.6$$

همانظور که مشاهده می‌کنید، با وجود اینکه خطای مدل زرد در داده‌های آموزش ۰ می‌باشد، هنگامی که با داده‌هایی مواجه می‌شویم که مدل تا به حال آنها را ندیده، به یک بازه خطا به شدت افزایش پیدا می‌کند. این یعنی مدل داده‌های آموزش را "حفظ کرده" و "یاد نگرفته". اگر بخواهیم یک مثال ملموس‌تر بزنیم، می‌توانیم اینگونه فرض کنیم: تصور کنید که شما به شهر جدیدی رفته‌اید و نیاز به تاکسی پیدا می‌کنید. تاکسی از شما پول زیادی می‌گیرد و از آن پس فکر می‌کنید همه تاکسی‌های آن شهر همین رفتار را دارند؛ این می‌شود بیش برازش. در مقابل آن فرض کنید که حتی پس از ۲۰۰ بار سوار تاکسی شدن در شهر خودتان و زمانی که همه آن تاکسی‌ها پول زیاد و ناحقی از شما دریافت کردند هنوز هم یاد نگرفته‌اید که نباید در آن شهر از تاکسی استفاده کنید.

برای اینکه از بیش‌برازش جلوگیری کنیم نباید مدلمان بیش از حد پیچیده باشد و برای اینکه از کم‌برازش جلوگیری کنیم نباید مدلمان بسیار ساده. پیدا کردن مدل با پیچیدگی مناسب بستگی به نوع داده‌ها و مهارت شخص درحال طراحی مدل دارد. البته برای این کار ابزارهایی مانند Regularization هم وجود دارد که از پیچیده شدن بیش‌ازحد مدل اجتناب کند. به نظر می‌رسد برای این نوع داده‌ها تابعی درجه ۲ کافی باشد.

به نظر می‌رسد برای داده‌های ما یک مدل درجه ۲ می‌تواند نتیجه خوبی را بدهد:



شکل ۶: مدل مناسب

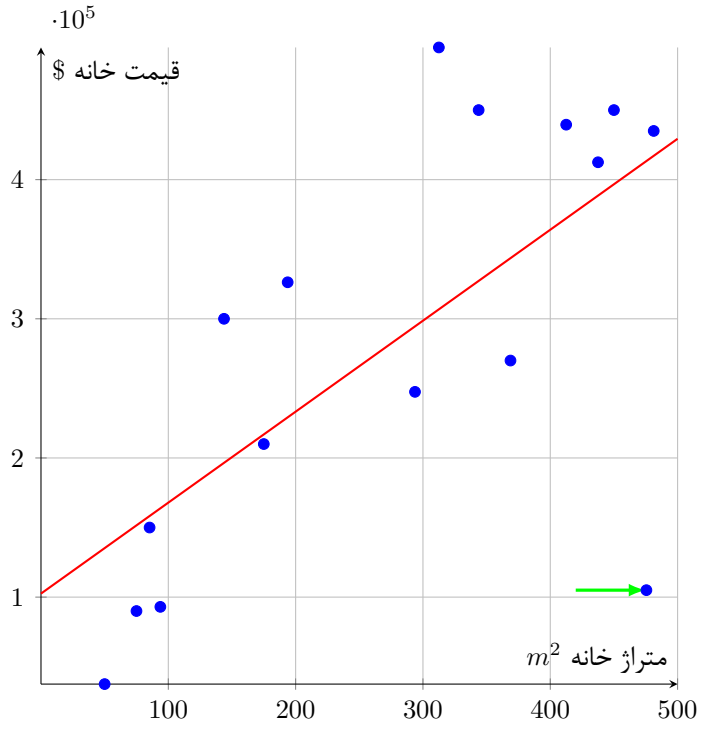
$$Error_{train} = 5.3, Error_{test} = 5.6$$
$$Error_{train} = 0.0, Error_{test} = 3.6$$
$$Error_{train} = 2.2, Error_{test} = 2.8$$

همانطور که مشاهده می‌کنید، با وجود اینکه خطای مدل بنفش برای داده‌های آموزش ۲.۲ می‌باشد. برای داده‌های آزمون خطای این مدل از هر دو مدل بیش‌برازش شده و کم‌برازش شده، کمتر است.

۶.۲ نقاط پرت (Outliers)

به اشکال خاصی از داده‌ها اطلاق می‌شود که از الگوهای عمومی یا میانگین معمول داده‌ها بیرون می‌افتند. به عبارت دیگر، این داده‌ها به دلایلی مثل خطای اندازه‌گیری یا وقوع رویدادهای نادر (از جمله خطا یا اطلاعات اشتباه) از سایر نمونه‌ها متمایز می‌شوند. این اشکال داده‌ها می‌توانند تحلیل‌های آماری را تحت تأثیر قرار داده و به تصمیمات نادرستی منجر شوند. بنابراین، شناسایی و مدیریت outliers در تحلیل داده‌ها از اهمیت بسیاری برخوردار است. یکی دیگر از دلایل دیگر به وجود آمدن outlierها می‌تواند این باشد که ما ابعاد دیگر داده را درنظر نگرفتیم. مثلا برای قیمت خانه‌ها تنها فاکتور مشخص کننده قیمت خانه متراژ آن نیست، بلکه فاکتورهای دیگر از جمله موقعیت مکانی، امکانات دیگر خانه و غیره در این تصمیم‌گیری نقش دارند. بدون داشتن ابعاد دیگر داده نمی‌توانیم تشخیص دهیم که دلیل به وجود آمدن آن‌ها چه چیزیست.

در شکل زیر دور یکی از outlierها خط کشیده شده است.



شکل ۷: نمایش یک نقطه پرت

۳ دسته بندی

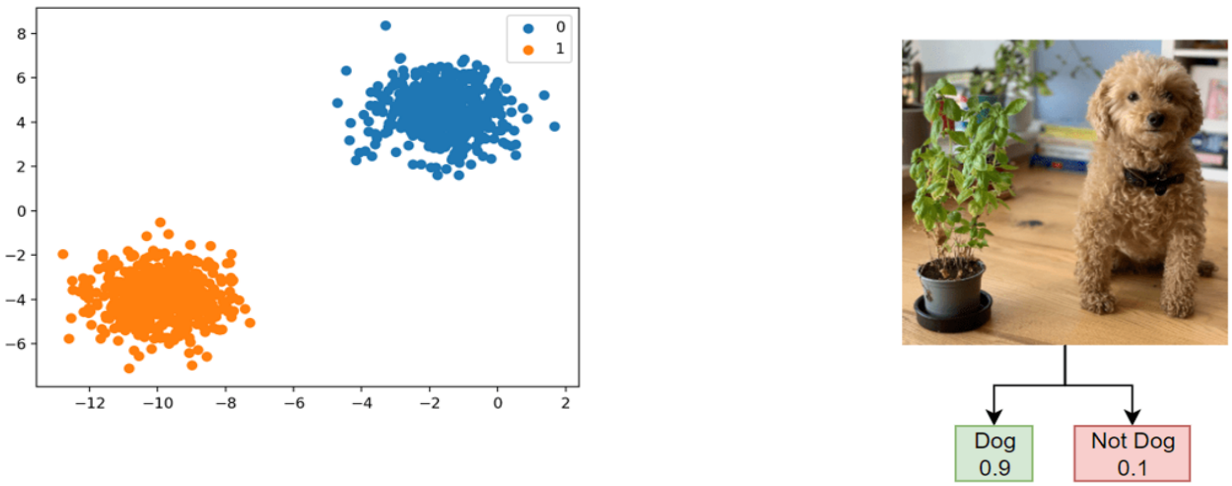
۱.۳ مقدمه

اکنون وارد مبحث Classification (دسته بندی) میشوم. در Supervised Classification هدف ما یادگیری تابع‌ای است که بتواند ورودی را به گروه‌های گسسته دسته بندی کند.

۲.۳ انواع دسته بندی

۱.۲.۳ دسته بندی دودوی

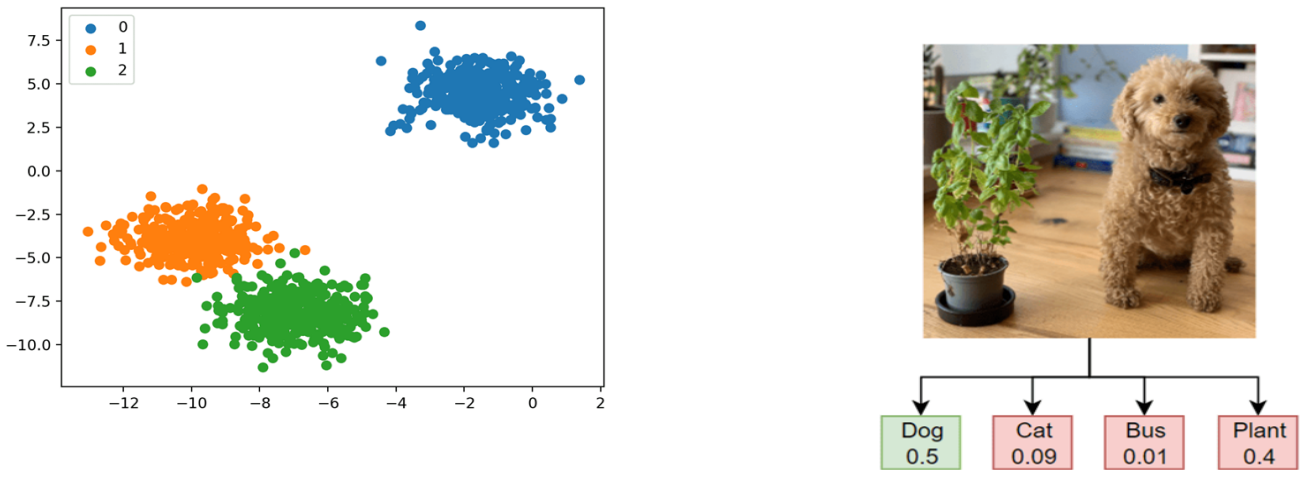
در ابتدا بهتر است که به انواع مختلف Classification اشاره کنیم. ساده ترین نوع Binary Classification است که در آن بودن و یا نبودن مورد بررسی قرار میگیرند و ورودی حداکثر دو نوع دسته خروجی دارد.



شکل ۸: Binary Classification.

۲.۲.۳ دسته بندی چند کلاسه

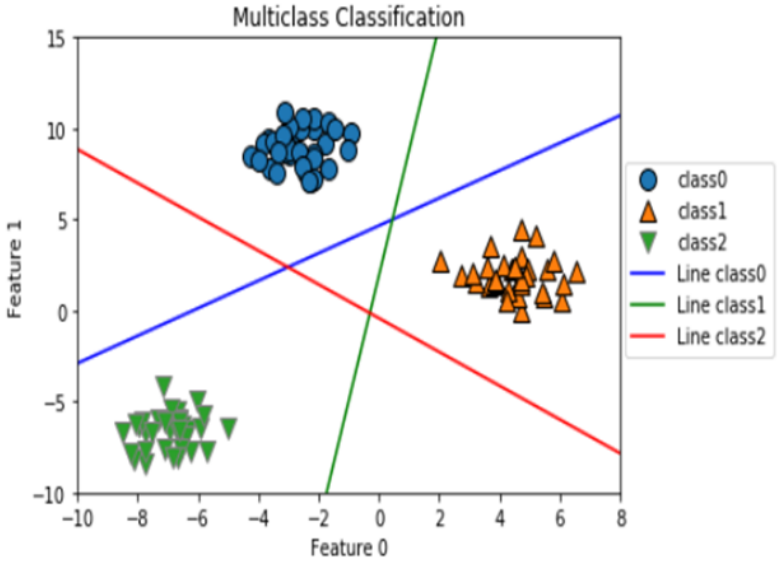
نوع دوم MultiClass Classification است که در آن تعداد دسته‌های که ورودی میتواند در آنها قرار گیرد بیشتر از ۲ است.



شکل ۹: MultiClass Classification.

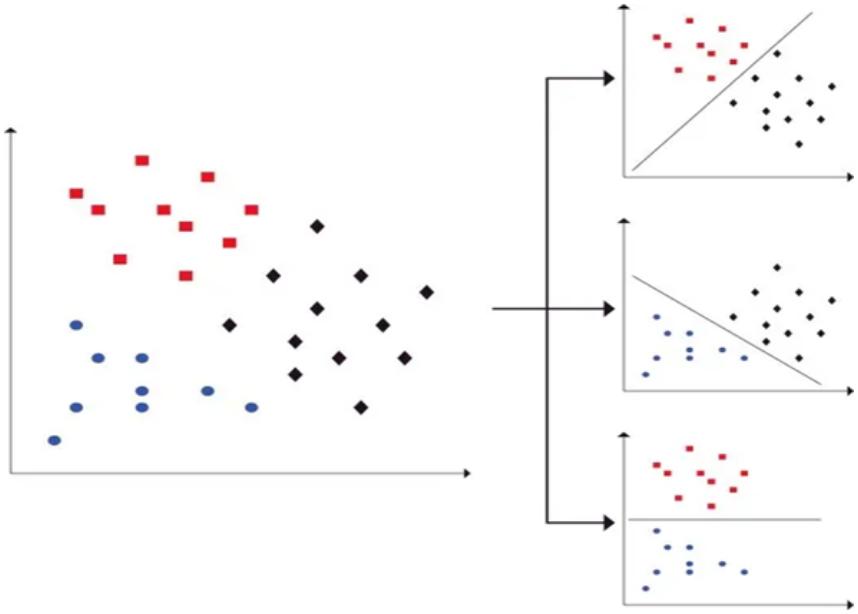
مسائل MultiClass Classification به یکی از دو صورت زیر مدل سازی میشوند:

یک در مقابل همه (باقی): در این روش مسئله را به چند زیر مسئله کوچک تر تبدیل میکنیم، به این صورت که هر زیر مسئله خود یک مسئله Binary Classification است و در آن عضویت هر ورودی برای هر دسته در مقابل سایر دسته‌ها بررسی میشود.



شکل ۱۰: MultiClass Classification.

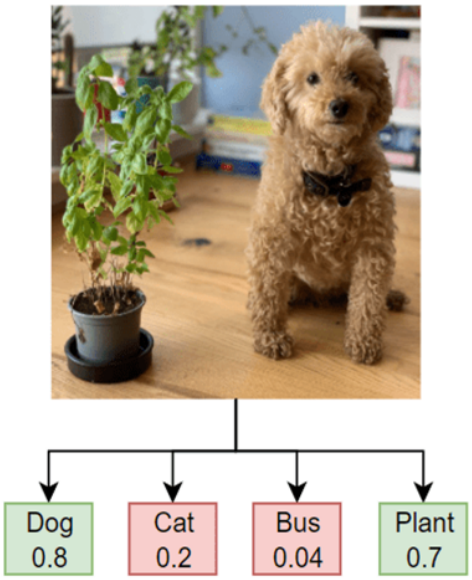
یک در مقابل یک: در این روش نیز مسئله را به چند زیر مسئله کوچک تر تبدیل میکنیم، به این صورت که هر زیر مسئله خود یک مسئله Binary Classification است و در آن مانند مثال زیر برای سه دسته مسئله به سه مسئله Binary Classification تقسیم بندی میشود و ورودی را با هر سه مدل ایجاد شده بررسی میکنیم.



شکل ۱۱: MultiClass Classification.

۳.۲.۳ دسته بندی چند دسته‌ای

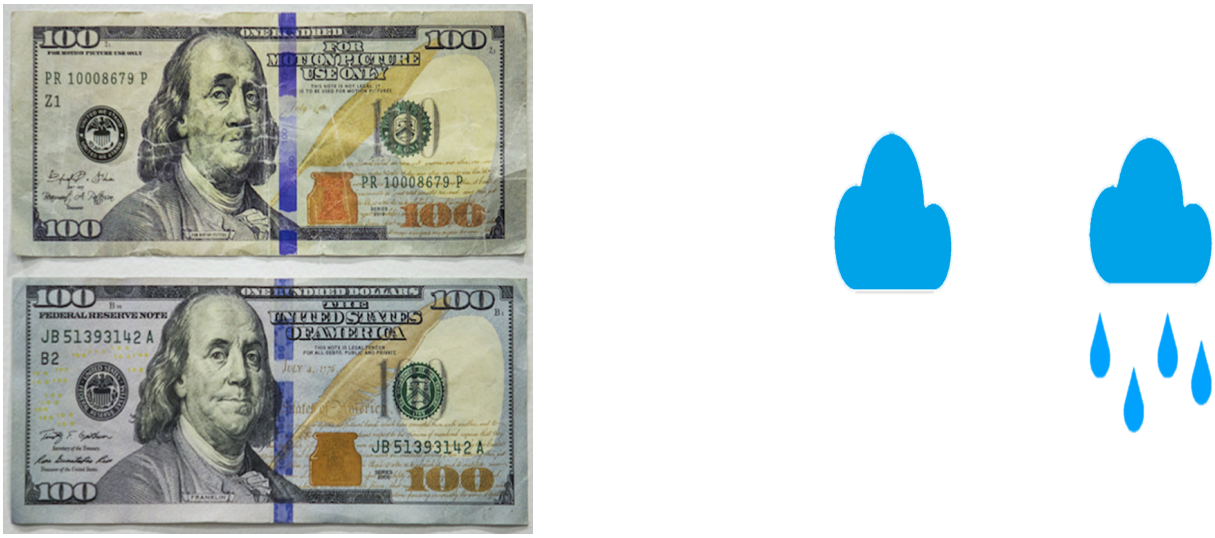
آخرین دسته از مسائل Classification مسائل Multilable Classification هستند که مشابه با مسائل MultiClass Classification میباشند با این تفوت که هر ورودی میتواند به بیش از یک دسته تعلق داشته باشد.



شکل ۱۲: Multilable Classification.

۳.۳ نگاهی عمیق تر به دسته بندی دودوی

در اینجا به دلیل ذات آسان‌تر مسائل Binary Classification و زمان محدود ارائه تمرکز خود را بر روی مسائل Binary Classification می‌گذاریم. نمونه‌های از مسائل Binary Classification مانند تشخیص اسکناس‌های اصلی از تقلبی و یا تشخیص روزهای بارانی و غیر بارانی هستند.



شکل ۱۳: Binary Classification.

در نظر بگیرید که برای برای حل مسائل روزهای بارانی به روش Supervised Classification یک جدول از روزهای گذشته داریم که در آن برای هر روز میزان رطوبت، فشار هوا و بارانی بودن و نبودن آن روز ثبت شده اند.

Date	Humidity	Pressure	Rain
January 1	93%	999.7	Rain
January 2	49%	1015.5	No Rain
January 3	79%	10.31.1	No Rain
January 4	65%	984.9	Rain
January 5	90%	975.2	Rain

جدول ۱: Rainy days

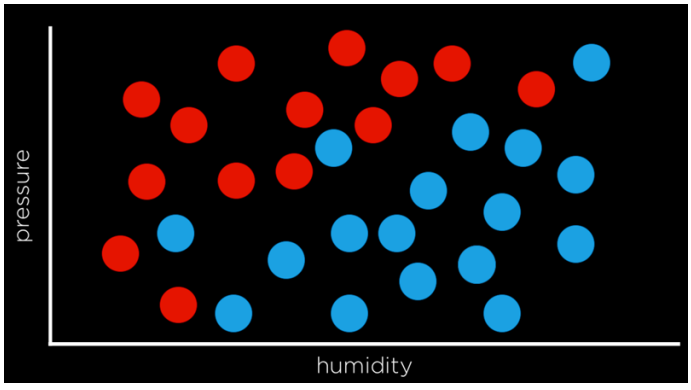
در طبیعت یک تابع وجود دارد که به صورت زیر کار میکند:

$$f(humidity,\ pressure) = Rain\ or\ Not\ Rain \tag{۷}$$

این تابع با دریافت رطوبت و فشار هوا برای هر روز مشخص میکند که آیا آن روز باید باران ببارد یا خیر.(در واقعیت عوامل بسیار بیشتر و پیچیده‌تری در تایین وضعیت آب‌وهوا دخیل هستند اما به دلیل جلوگیری از پیچیده شدن بیش از حد از آنها صرف نظر میکنیم) از آنجا که این تابع نامشخص است هدف ما در اینجا رسیدن به تابعی heuristic (تقریبی) است که تا حد امکان مشابه به تابع اصلی در طبیعت باشد و امکان تشخیص روزهای بارانی و غیربارانی را با دقت قابل توجهی فراهم کند.

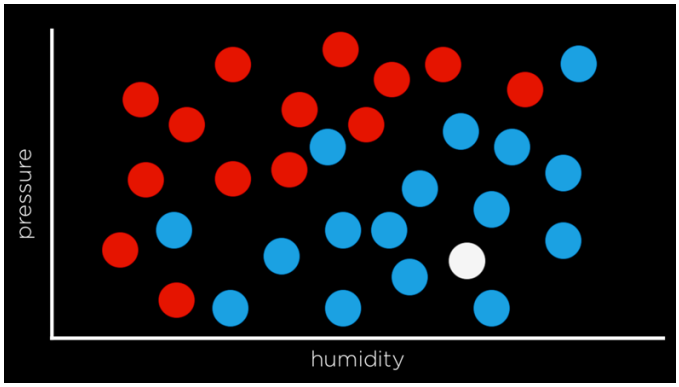
$$h(humidity,\ pressure) = Rain\ or\ Not\ Rain \tag{۸}$$

بهتر است داده‌های فرضی یک جدول رطوبت و فشار هوا را برای روزهای بارانی و غیربارانی در یک محور مختصات بر اساس رطوبت و فشار هوا مانند زیر ترسیم کنیم:



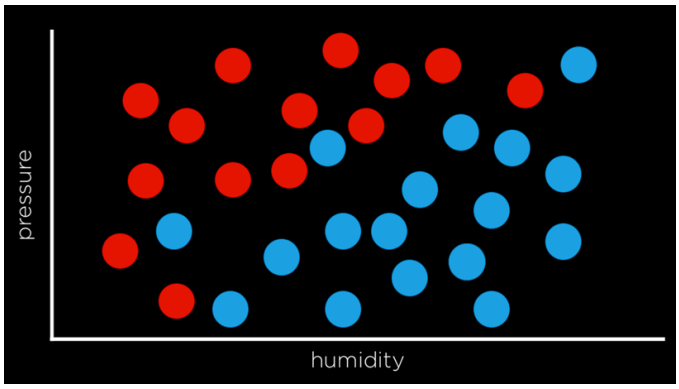
شکل ۱۴: Binary Classification.

در این شکل نقاط آبی نشان دهنده روز های بارانی و نقاط قرمز نشان‌دهنده روزهای غیربارانی هستند. اگر فردی تصویر زیر را به شما نشان دهد و از شما بخواهد که حدس خود را درباره رنگ نقطه سفید بیان کنید (تخمین بزنید که این روز بارانی است یا خیر) چه پاسخی میدهید؟



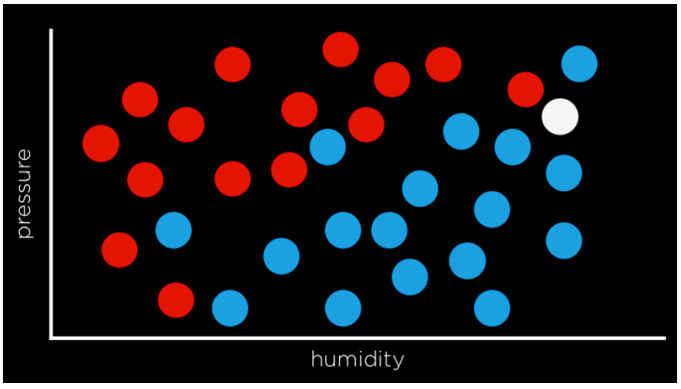
شکل ۱۵: Binary Classification.

تقریبا حدس تمامی افراد (عاقل) به این صورت خواهد بود که رنگ حقیقی این نقطه آبی است زیرا در میان نقاط آبی محاصره شده است. اگر حدس خود را بر اساس این منطق و این نکته که نزدیک ترین نقطه به سفید خود رنگ آبی دارد، منطق شما مشابه با تکنیک دسته‌بندی نزدیک‌ترین همسایه خواهد بود. در دسته‌بندی نزدیک‌ترین همسایه، در ابتدا نزدیک‌ترین نقطه به نقطه مورد سوال را یافته و به این گونه در نظر میگیریم که از آنجا که تمامی ویژگی‌های مستقل این دو نقطه مشابه هم هستند به احتمال فراوان ویژگی غیر وابسته آنان نیز (رنگ) یکسان خواهد بود.



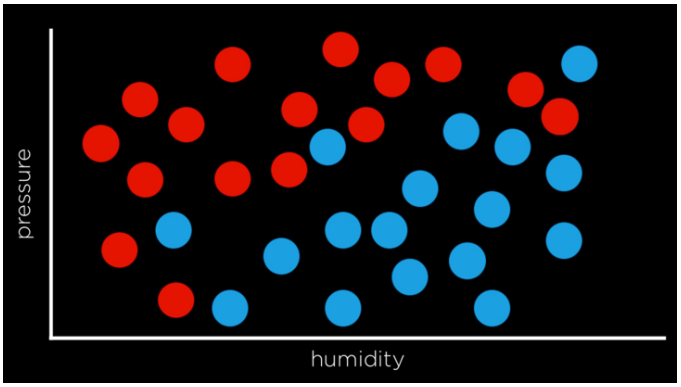
شکل ۱۶: Binary Classification.

حال مثال زیر را در نظر بگیرید:



شکل ۱۷: Binary Classification.

اکنون چه حدسی برای نقطه سفید دارید؟ اگر بر اساس اصل نزدیک‌ترین همسایه پیش برویم انتظار میرود که رنگ این نقطه قرمز باشد (غیر بارانی).

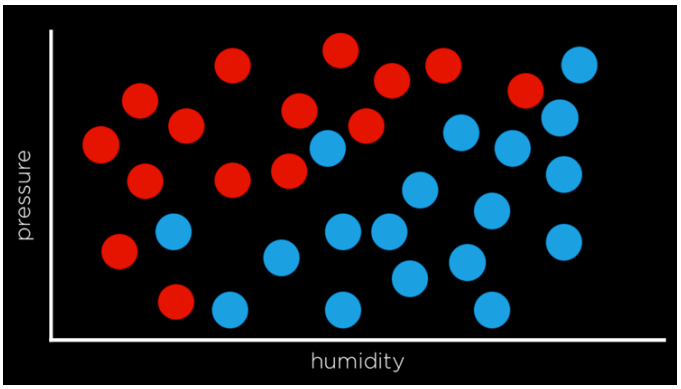


شکل ۱۸: Binary Classification.

اما همانطور که مشاهده میکنید در اطراف این نقطه تعداد بسیار بیشتری نقاط آبی قرار دارند و نقطه قرمز که نزدیک ترین نقطه به نقطه مورد نظر ما است یک داده outlier است، بر همین اساس انتظار میرود که احتمال اینکه رنگ نقطه مورد نظر آبی باشد بیشتر است (هرچند نمیتوان با قاطعیت بیان کرد).

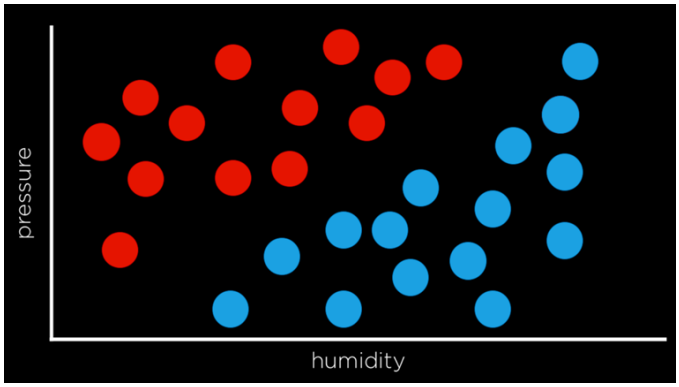
به منظور جلوگیری از بروز چنین موارد الگوریتم نزدیک‌ترین همسایه را کمی تغیر میدهیم، به این صورت که دیگر تنها بر اساس نزدیک ترین همسایه به نقطه مجهول تصمیم‌گیری نمیکنیم بلکه تعداد بیشتری از نقاط همسایه نزدیک را در نظر میگیرم و در میان آنها رای گیری میکنیم (به این منظور برای جلو گیری از برابر شدن تعداد آرا تعداد نقاط همسایه مورد بررسی را یک عدد فرد قرار میدهیم). روش بیان شده یک نسخه از الگوریتم نزدیک‌ترین همسایه است که k نزدیک‌ترین همسایه نام دارد (k تعداد همسایه‌های مورد بررسی است).

شکل ۱۹: Binary Classification.



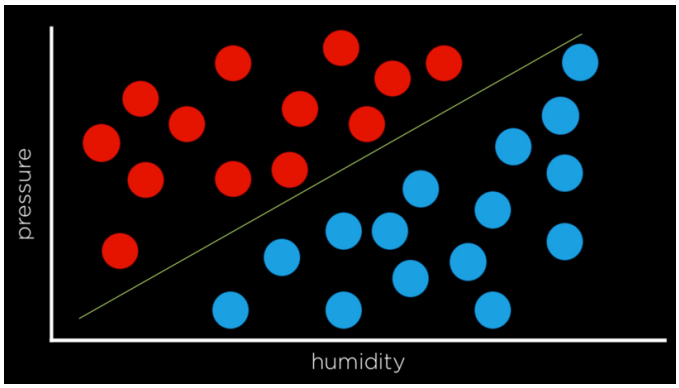
شکل ۲۰: Binary Classification.

الگوریتم k نزدیک‌ترین همسایه به دلیل آسانی استفاده‌های زیادی دارد اما به دلیل مشکلاتی نظیر اینکه برای به دست آوردن نزدیک ترین نقاط هماسیه به نقطه مجهول لازم است فاصله نقطه مجهول با تمامی نقاط را محاسبه کنیم کارای و عملکرد الگوریتم در دیتاست‌های بسیار بزرگ به شدت کاهش میابد. به این دلیل اکنون روش دیگری را برای حل مسائل Binary Classification بیان میکنیم. مثال زیر را در نظر بگیرد.



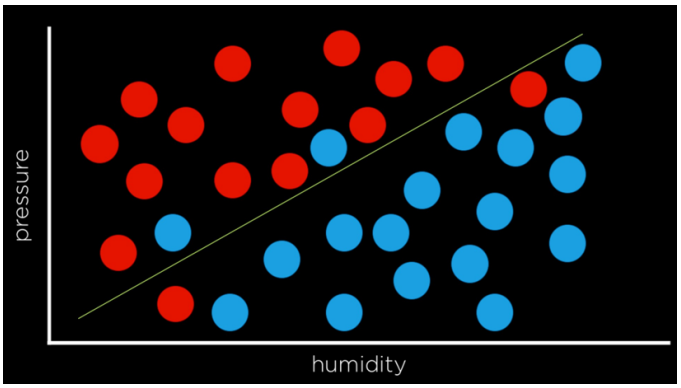
شکل ۲۱: Binary Classification.

آیا به نظر شما میتوان این نقاط را به وسیله یک خط مرزی از هم جدا کرد؟ پاسخ به این سوال مثبت است. میتوان مانند تصویر زیر عمل کرد.



شکل ۲۲: Binary Classification.

در واقعیت داده‌های ما به این صورت تمیز و خطی جداپزیر (linearly separable) نیستند.



شکل ۲۳: Binary Classification.

هدف ما این است که خطی را به دست آوریم که به بهترین شکل ممکن داده را از هم جدا کند (در آینده در رابطه با نحوه ارزیابی خطوط مختلف بحث خواهیم کرد). به منظور رسیدن به این هدف میتوان به این صورت تصور کرد که ما میخواهیم یک تابع heuristic را به گونه‌ای به دست آوریم که که مانند شکل زیر با دریافت مقادیر برای هر ورودی، مشخص کند که آن ورودی یک روز بارانی است یا خیر.

$$x_1 = Humidity \tag{۹}$$

$$x_2 = Pressure \tag{۱۰}$$

$$h(x_1,\, x_2) = Rain\ if\ w_0\ +\ w_1x_1\ +\ w_1x_2\ \geq\ 0\ otherwise\ No\ Rain \tag{۱۱}$$

در این تابع w ها وزن (wight) نام دارند. آنها را به صورت یک vector مینویسیم و ورودی را نیز به حالت vector در میآوریم (به vector ورودی مقدار یک را به عنوان ضریب برای w_0 اضافه میکنیم تا اندازه vector ورودی با اندازه vector وزن‌ها یکسان باشد و بتوانیم dot product آنها را محاسبه کنیم).

$$Weight\ Vector\ w : (w_0,\, w_1,\, w_2) \tag{۱۲}$$

$$Input\ Vector\ x : (1,\, x_1,\, w_2) \tag{۱۳}$$

$$w.x : w_0 + : w_1x_1 + : w_2x_2 \tag{۱۴}$$

$$h_{\mathrm{w}}(x) = 1\ if\ w\ .\ x\ \ =\ 0\ otherwise\ 0 \tag{۱۵}$$

حال به نحوه چگونگی مقادیر وزن‌ها میرسیم. برای محاسبه مقادیر وزن‌ها میتوانیم به صورت زیر عمل کنیم. هربار برای هر ورودی مقدار تابع را محاسبه میکنیم، اگر هردو یکسان بودند تغیری در وزن‌ها ایجاد نمیشود، اما اگر متفاوت بودند بنا بر میزان تفاوت ((alpha (learning rate)) مقادیر وزن‌های جدید را محاسبه میکنیم.

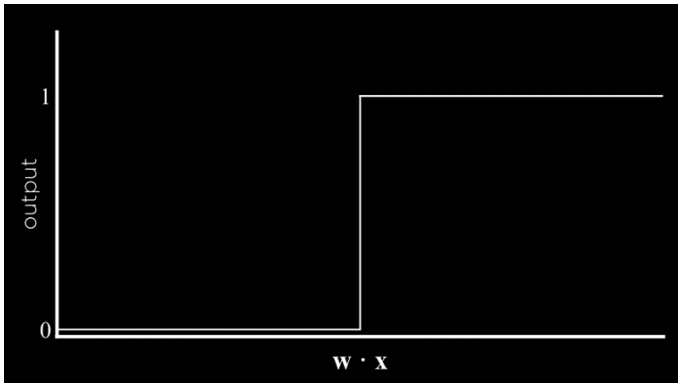
Perceptron Learning Rule

Given data point (x, y), update each weight according to:

$$w_{\mathrm{i}} = w_{\mathrm{i}} + \alpha(y - h_{\mathrm{w}}(x)) \times x_{\mathrm{i}} \tag{۱۶}$$

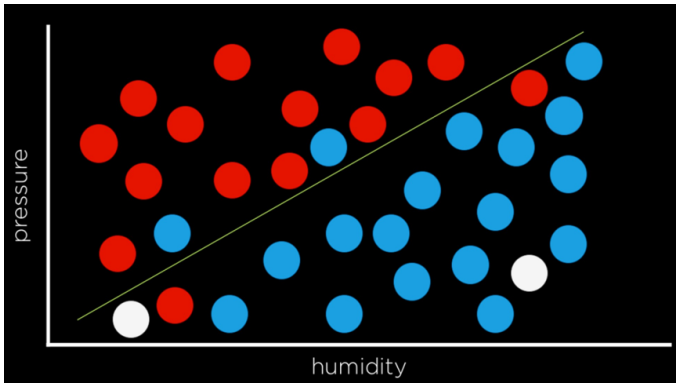
$$w_{\mathrm{i}} = w_{\mathrm{i}} + \alpha(actual\ value - estimate) \times x_{\mathrm{i}} \tag{۱۷}$$

اگر از این روش پیروی کنیم به یک تابع فعال کننده مانند شکل زیر میرسیم.



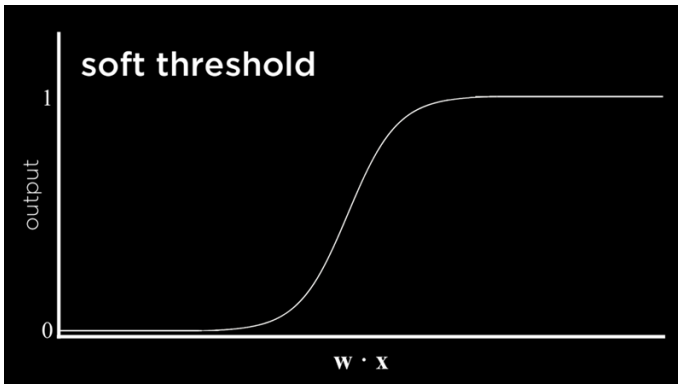
شکل ۲۴: Binary Classification.

در این تابع اگر مقدار ضرب vector های w و x (بر اساس مقدار ورودی x) از یک مقدار مشخصی کمتر باشد تابع آن ورودی را به عنوان روز غیر بارانی و اگر مقدار ضرب vector های w و x بیشتر از آن مقدار باشد آن ورودی را به عنوان یک روز بارانی در نظر میگیرد.
به تصویر زیر دقت کنید.



شکل ۲۵: Binary Classification.

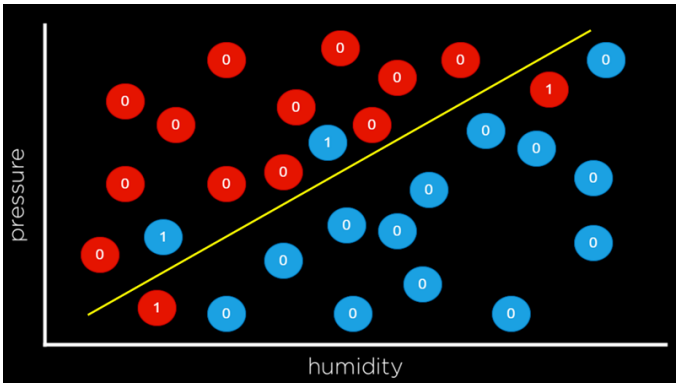
این مدل هر دو ورودی سفید رنگ را به عنوان نقاط بارانی در نظر میگیرد. اما با توجه بیشتر به تصویر مشاهده میکنید که میزان دقتی که برای تعیین دسته این دو ورودی وجود دارد دارای تفاوت فاهشی است و نقطه سفید رنگ در سمت چپ پایین صفحه میتواند در واقعیت یک روز غیر بارانی باشد. این نوع تابع hard threshold این امکان را به ما نمیدهد که میزان اطمینان مدل از گروه تعیین شده برای ورودی را به دست آوریم و به همین منظور اغلب از تابع logistic regression که این امکان را به ما میدهد استفاده میکنیم.



شکل ۲۶: Binary Classification.

۴.۳ ارزیابی مدل

برای ارزیابی مدل میتوان از روش‌های متنوعی استفاده کرد که در اینجا یکی از آنها را بیان میکنیم. ارزیابی مدل‌های دسته‌بندی را میتوان با روشی به نام 0-1 loss انجام داد، به این صورت که تعداد داده‌های که به اشتباه دسته بندی شده‌اند را محاسبه کرده و هدف خود را کمینه کردن مقدار این تابع قرار میدهیم.



شکل ۲۷: Binary Classification.

مراجع

[١] CS50's Introduction to Artificial Intelligence with Python

[٢] Learning - Lecture 4 - CS50's Introduction to Artificial Intelligence with Python 2020

[٣] Machine learning - Wikipedia

[٤] Regression analysis - Wikipedia

[٥] Linear regression - Wikipedia

[٦] How AIs, like ChatGPT, Learn

[٧] Linear Regression; A Visual Introduction To (Almost) Everything You Should Know