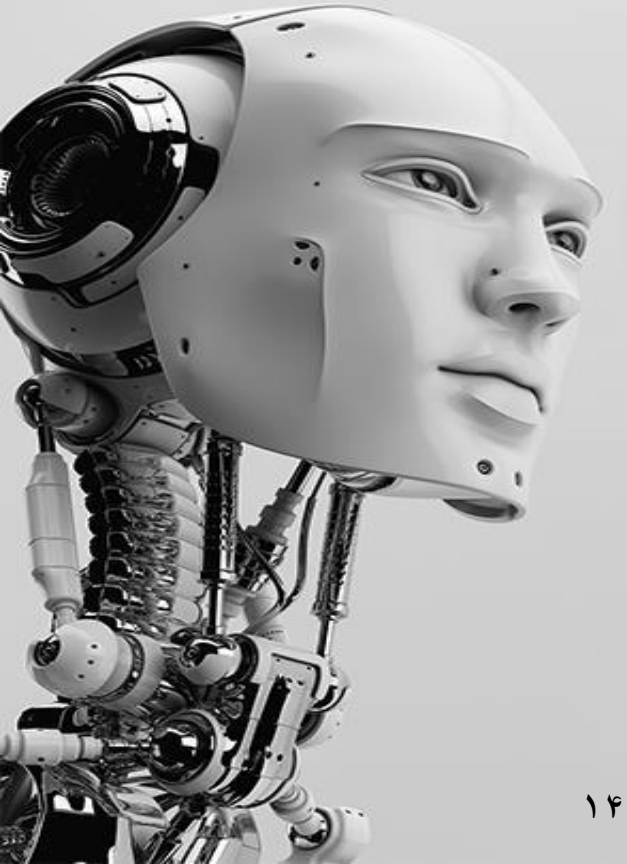
دوره مقدمات داده‌کاوی DATA MINING و یادگیری ماشین MACHINE LEARNING

05

سید ایمان غفوریان - عضو هیات علمی دانشگاه آزاد مشهد -
زمستان ۱۴۲۰

دوره مقدمات داده‌کاوی DATA MINING و یادگیری ماشین MACHINE LEARNING

05



تمرکز این بخش بر آشنایی مقدماتی با داده‌کاوی data mining و یادگیری ماشین machine learningاست. به صورت انتزاعی با مباحث مختلف داده‌کاوی و کاربردهای آن آشنای می شویم. همچنین به معرفی انواع روش‌های داده‌کاوی به صورت کاربردی‌تر خواهیم پرداخت. سعی شده با اصطلاحات مختلف این حوزه آشنا شویم و یک بررسی سطح بالا در داده‌کاوی و یادگیری ماشین machine learningدر این بخش داشته باشیم.

سید ایمان غفوریان - عضو هیات علمی دانشگاه آزاد مشهد-زمستان ۱۴۰۲

داده‌کاوی Data mining چیست؟

داده‌کاوی فرآیندِ **تبدیلِ** یک سری **داده**، به یک سری **دانش**، توسط فرآیندهای مختلف است.

با یک مثال شروع می‌کنیم. فرض کنید شما مدیرِ یک بانک هستید. و میخواهید از بینِ ۱۰۰۰۰۰ مشتری که متقاضی **وام** هستند، به ۱۰۰۰ نفر وام دهید. پس لازم است که از بین این ۱۰۰۰۰۰ نفر، ۱۰۰۰ نفری را انتخاب کنید که **اطمینانِ** بیشتری برای برگرداندنِ وام دارند. ولی این مدیر بانک فرصت ندارد که تمامیِ ۱۰۰۰۰۰ نفر را یکی یکی ارزیابی کند. علاوه بر این هر روز افراد جدیدی از راه می‌رسند و بایستی یکی یکی آن‌ها را هم ارزیابی کند. این مدیر، تصمیم می‌گیرد به جای فرآیندهای **سنتی**، از **روش‌های داده‌کاوی** برای حل این مسئله استفاده کند.

در فرآیند داده‌کاوی، **ابتدا** مدیر بانک بایستی یک تعدادِ کمی از افراد مثلاً ۲۰۰ فرد را به عنوان افرادِ **مطمئن** و ۲۰۰ نفر دیگر را به عنوان افرادِ **غیر مطمئن** برای سیستم مشخص کند. این کار توسط **هوشِ طبیعیِ مدیر** بانک قابل انجام است.

اینجاست که داده‌کاوی وارد عمل میشود و ۲۰۰ فرد مورد اطمینان و ۲۰۰ فرد غیرمطمئن که مدیر بانک **برچسب زده** بود را مشاهده کرده و **الگوهای رفتاریِ** این افراد را مورد بررسی قرار می‌دهد. در واقع سیستم متوجه می شود که کدام الگوی رفتاری، **منجر** به **اطمینان** و کدام الگو منجر به **عدم اطمینان** می‌شود . در اینجاست که سیستم، **یاد** می گیرد **learn** و می‌تواند بین افرادِ مطمئن و غیرمطمئن ، تمایُز قائل شود. البته برای تشخیص این الگو، مدیر بانک بایستی **ویژگی‌های** مشتریان را در اختیار الگوریتم بگذارد.

داده‌کاوی Data mining چیست؟

حال این سیستم که فرآیند را یادگرفته است، می‌تواند هر مشتری دیگری علاوه بر این ۴۰۰ نفر که در موردِ یادگیری قرار گرفته‌اند را نیز، در دسته مطمئن‌ها و غیرمطمئن‌ها، تقسیم کند. اینجاست، که تمامی ۱۰۰۰۰۰ نفر را به سیستم وارد می‌کنیم و خروجی این سیستم، می‌تواند افرادی را مشخص کند که مطمئن هستند و می‌توان به آن‌ها وام داد.

این یک مثال، از داده‌کاوی بود، که به یادگیریِ نظارت‌شده نیز معروف است. در این جا، ناظر (همان مدیر بانک) یک مجموعه‌ی کم از داده‌ها را برای سیستم، به اصطلاح برچسب label زد. یعنی مشخص کرد که کدام مشتری مطمئن و کدام مشتری نامطمئن است، سپس سیستم از روی این داده‌های برچسب زده شده و ویژگی‌های آن‌ها، یادگیری را انجام داد.
همان طور که مشاهده می‌کنید، از یک مجموعه‌ی داده (مشتری‌های بانک)، به یک سری دانش (به چه شخصی وام بدهیم به چه شخصی وام ندهیم) رسیدیم.
علم داده‌کاوی، بسیار گسترده تر از بحثِ مثالِ اینجاست. به جز یادگیری نظارت‌شده که در مثال بالا گفته شد (یادگیری با مجموعه آموزشی داده)، یادگیریِ غیرنظارت‌شده (خوشه‌بندی) یا همان clustering، قواعد وابستگی، یادگیری تقویت شده و... نیز از زیر حوزه‌های علم داده‌کاوی هستند.

یادگیری ماشینMachine Learningچیست؟

یادگیری ماشین، **زیر مجموعه‌ای** از هوش مصنوعی است. با استفاده از تکنیک‌های یادگیری ماشین، کامپیوتر، الگوهای موجود در **داده‌ها** (اطلاعات **پردازش** شده) را یادگرفته و می‌تواند از آن استفاده کند. توجه داشته باشید که در این **تکنیک‌ها**، **یادگیریْ** در یک سیستمِ کامپیوتری، **بدون برنامه‌نویسیِ** صریح (explicit programming)صورت می‌پذیرد. پاسخ به این سوالات که برنامه‌نویسیِ صریح چیست و یادگیری ماشین چطور کار می‌کند مواردی هستند که در این درس به آن پاسخ می‌دهیم.

مثالِ کلاسیکِ زیر را در نظر بگیرید:

مثال :فرض کنید در یک فروشگاهِ بزرگِ خرده‌فروشی به صورت اینترنتی در حال خرید هستید. در زمان خرید، سه محصول مختلف را به سبد خرید خود اضافه می‌کنید. فرض کنید این سه محصول به صورت زیر است:

لپ تاپ سری N– موس بیسیم – یک عدد تمیز کننده مانیتور

حال، سیستم می‌خواهد به صورت **هوشمند**، به شما **چند محصول** دیگر را پیشنهاد دهد. **مدل برنامه‌نویسی صریح**، به این صورت است که برای مثال، سیستم، محصولاتی در یک دسته (مثلاً یک سری محصولاتی که مربوط به حوزه‌ی ITاست) را به شما نمایش بدهد. در این حالت، **هوشمندیِ خاصی** در سیستم مشاهده **نمی‌شود** و در واقع سیستم، یادگیریِ خاصی **انجام نمی‌دهد**.

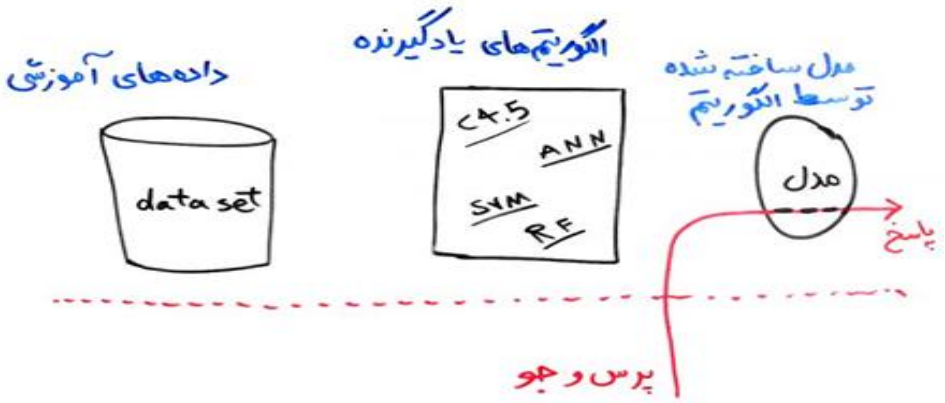
حال فرض کنید، سیستم از **طریق الگوریتم‌های یادگیری‌ماشین**، بتواند مشتریانِ قبلیِ خود را به گروه‌های مختلف تقسیم‌بندی کند (به این کار در **اصطلاح خوشه‌بندی** یا Clustering گفته می‌شود). با این کار، شما با تکمیلِ سبد خرید خود، به دسته‌ای از مشتریانِ قبلی متعلق می‌شوید. با تعلقِ شما به گروهِ خاصی از مشتریان، محصولاتی که آن‌ها قبلاً خریداری کرده‌اند و شما در سبد خرید خود ندارید، به شما پیشنهاد داده می‌شود.

یادگیری ماشینMachine Learningچیست؟

فرآیند کلیِ یادگیری ماشین را می‌توان به صورت شکل زیر مدل کرد. داده‌های آموزشی (در مثال ما: مشتریانی که قبلا خرید کرده اند و سابقه خرید آن‌ها) به الگوریتم‌های یادگیری ماشین تزریق می‌شوند. این الگوریتم‌ها، وظیفه‌ی یادگیری و واکشیِ الگوها ی patternsمختلف، در داده‌ها را دارند. بعد از به دست آوردنِ الگوها توسط الگوریتم‌ها، معمولا یکی از الگوریتم ها مورد استفاده قرار میگیرد، و یک مدل model ساخته می‌شود. این مدل می تواند در حافظه ذخیره شود. بعد از ذخیره‌ی مدل، سیستم توانایی پیش‌بینیِ رفتار را دارد. در مثال بالا، شما (شخصی که چند محصول را در سبد خرید خود دارد)، به عنوان یک پرس‌وجو به مدلی که یادگرفته است، داده می‌شوید. این مدل، می‌تواند خروجیِ پیشبینی (در این مثال، محصولی که باید به شما بر اساس خریدهای مشتریانِ هم‌دسته‌ی شما توصیه شود) را برگرداند.

به این مدل، در سیستم‌های کامپیوتری،یادگیری ماشین machine learning گفته می‌شود. مدلی که شاید بتوان آن را برنامه‌نویسیِ برنامه‌نویسی! دانست.

در واقع برنامه‌نویسی، فرآیندها را خودکار (اتوماتیک) می کند، این در حالی است که یادگیری ماشین همین فرآیندهای خودکار را به صورت خودکار تولید می کند.



یادگیری ماشین Machine Learning چیست؟

یادگیری ماشین با فرآیندهای داده‌کاوی، بسیار شبیه (و از نگاه کاربردی تقریباً یکسان) است. در فرآیندهای یادگیری‌ماشین دو نوع یادگیری وجود دارد: یادگیریِ نظارت‌شده supervised learning و یادگیریِ غیرنظارت شده unsupervised learningالبته انواع دیگری مانند یادگیری نیمه نظارت‌شده semi supervised learningیا یادگیری تقویتی reinforcement learning نیز وجود دارند.

در فرآیندهای یادگیری ماشین، داده‌ها، بسیار اهمیت دارند.در واقع این داده‌ها هستند که به الگوریتم تزریق می‌شوند و الگوریتم از روی آن‌ها یادگیری را انجام می‌دهد. در مثال بالا، داده‌های سابقه‌ی فروشِ فروشگاه اینترنتی و مشتریانی که این خریدها را انجام داده‌اند، به سیستم داده می‌شود.
اصطلاحِ معروفی در این حوزه وجود دارد :.

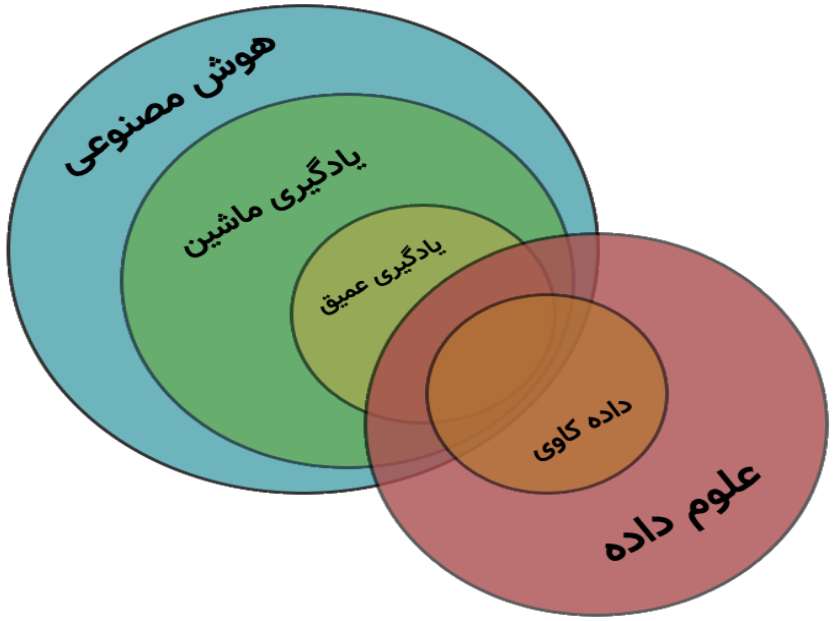
اگر داده‌ی بد، به سیستم تزریق شود، خروجی نیز، خروجی بدی خواهد بود

به این معنی که، هر چقدر الگوریتم‌های مختلفِ یادگیری ماشین، قوی و جامع طراحی شوند، اگر داده‌های خوبی به سیستم وارد نشود (مثلاً داده‌های غلط یا داده‌های ناکافی به سیستم تزریق شوند)، سیستم، پاسخی غیرِدقیق و ناصحیح ارائه می‌دهد.

تفاوت هوش مصنوعی، یادگیری ماشین، داده‌کاوی، یادگیری عمیق و علم داده

بسیاری از افراد (حتی افراد متخصص در حوزه‌ی هوش مصنوعی یا علوم داده) ممکن است تفاوت دقیق واژه‌های مورد استفاده در حوزه‌ی هوش مصنوعی و علوم داده را ندانند.دلیل آن هم شاید این باشد که واقعاً مرز روشنی بین این تعاریف وجود ندارد و در بسیار از مواقع این حوزه‌ها با یکدیگر ترکیب می‌شوند. با این حال در این درس قصد داریم تا حد ممکن تفاوت این عبارات را با یکدیگر بیاموزیم و کاربرد هر یک را ببینیم.

طبق تحقیقی که توسط گروهی از علاقه‌مندان به این حوزه در سال ۲۰۱۸ انجام شد، نزدیک به ۴۰ درصد شرکت‌های استارتاپی در اروپا که ادعا کرده بودند از هوش مصنوعی در کارهای خود استفاده می‌کنند، واقعاً این کار را انجام نمی‌دادند و این خود نشان می‌دهد که شاید بسیاری از افراد (حتی افراد شاغل در حوزه‌ی آی‌تی) تفاوت مفاهیم را در این زیر حوزه‌ها درک نکرده باشند.



برای شروع به شکل زیر بنگرید:

ابتدا مفهوم هوش مصنوعی artificial intelligence بررسی می‌کنیم . طبق یک تعریف معروف، هوش مصنوعی یک مفهوم کلی است که به کامپیوتر این ویژگی را می‌دهد تا بتواند مانند انسان فکر کرده و عمل کند. پس هوش مصنوعی یک ایده‌ی کلی است که در برگیرنده‌ی بسیاری از مفاهیم دیگر شده است.

یادگیری ماشین Machine Learning چیست؟

یادگیری ماشین machine learning ، قسمت کاربردی هوش مصنوعی است که با کمک آن می‌توان به کامپیوتر از روی داده‌ها، الگوها patterns را یاد داد. در واقع کامپیوتر با یادگیری ماشین می‌تواند یادگیری را از روی داده‌ها انجام دهد و مسئله‌ای که برایش تعریف کرده‌ایم را حل کند.

یادگیری عمیق deep learning، به مجموعه‌ای از الگوریتم‌های یادگیری ماشین گفته می‌شود که غالباً می‌توانند الگوهای پیچیده‌تری را از میان داده‌ها کشف کنند.در واقع این دسته از الگوریتم‌ها مسائلِ سخت‌تری را درحوزه‌ی یادگیری ماشین و داده‌کاوی حل کنند. همان‌طور که در شکل قبل مشاهده می‌کنید، زیر مجموعه‌ی یادگیری ماشین بوده و اشتراک‌هایی با داده‌کاوی دارند.

علوم داده data science به مجموعه‌ی روش‌ها و فعالیت‌هایی گفته می‌شود که بر روی داده‌ها انجام می‌دهیم. در علوم داده، تحلیل اولیه داده‌ها، پیش‌پردازش داده‌ها و تفسیر و نمایش داده‌ها وجود دارد. واکشی داده‌ها و ذخیره‌سازی آن‌ها نیز جزو فعالیت‌های علوم داده قرار می‌گیرد.

اما شاید سخت‌ترین بخش، بیان تفاوت داده‌کاوی data mining و یادگیری ماشین machine learning باشد. همان‌طور که در شکل مشاهده می‌کنید این دو نقاط مشترک زیادی با یکدیگر دارند ولی شاید اساسی‌ترین تفاوت بین آن‌ها این باشد که در داده‌کاوی، دخالت انسان معمولاً بیشتر است.یعنی داده‌کاوی بدون تحلیل و کمک یک انسان معمولاً انجام ناپذیر است. این در حالیست که یادگیری ماشین می‌تواند به صورت خودکار self-learning نیز کارها را انجام دهد.

البته بخش‌های دیگر مانند بهینه‌سازی‌ها optimizations ، یادگیری تقویتی reinforcement learning و کلان داده big data نیز وجود دارد که هر کدام می‌توانند زیر مجموعه‌ی یک قسمت از شکل بالا باشند. برای مثال الگوریتم‌های بهینه‌سازی و یادگیری تقویتی را می‌توان داخل قسمت هوش مصنوعی قرار داد و کلان داده را نیز می‌توان جزو زیر دسته‌های علم داده گذاشت.

طبقه‌بندی Classification چیست؟

فرض کنید مدیریتِ یک بانک را برعهده دارید که ۱۰۰ هزار مشتری دارد و می‌خواهید به یک سری از مشتریانِ خود وام دهید. طبیعتاً به افرادی وام را خواهید داد که شانسِ برگرداندن وام توسط آن‌ها بیشتر باشد. هر کدام از این افراد نیز، دارای خصوصیات مختلفی هستند (ویژگی‌های آن‌ها). برای مثال، آیا این شخص خانه دارد یا نه؟ این شخص دارای اتومبیل شخصی هست یا خیر؟ حقوق دریافتیِ این شخص چقدر است؟ و... .

حال فرض کنید این بانک دارای یک سابقه‌ی ۱۰ هزار تایی از مشتریانی است که وام گرفته‌اند که یا توانسته‌اند برگردانند یا خیر. این افراد به دو دسته (۲ کلاس) تقسیم شده اند، یا توانسته اند وام خود را **بازگردانند** (**کلاسِ ۱**) یا **خیر** (**کلاسِ ۲**). همان طور که گفتیم این افراد خصوصیات یا ویژگی‌های مختلفی داشته‌اند. پس نگاهی به جدول زیر بیندازید:

مجموعه داده‌ی مشتری‌های بانک

	خانه دارد؟	تعداد فرزندان	اتومبیل دارد؟	حقوق دریافتی	وام را پس داده است؟
#1	1	2	1	800	بلی
#2	0	1	0	750	بلی
#3	0	2	1	700	بلی
#4	1	0	1	650	خیر
...	...	...	...	...	...

تفسیر این جدول که نوعی ماتریس نیز هست، ساده است. همان‌طور که مشاهده می‌کنید: شخصِ شماره‌ی ۱، دارای منزل است، تعداد ۲فرزند دارد،حقوق ماهیانه معادل ۸۰۰ هزارتومان دارد و یک اتومبیل از خود دارد. در **ستون آخر** ستون **برچسب** یا **lable** مشاهده می کنید که این شخص توانسته وام خود را برگرداند. شخص شماره ۲ و ۳ هم به همین ترتیب است یعنی توانسته‌اند وامِ خود را **برگردانند**. ولی شخصِ شماره‌ی ۴، با ویژگی‌هایی که دارد، **نتوانسته** وام دریافتیِ خود را بازگرداند. این **سه** مورد از **۱۰ هزار** مشتریِ مختلفی است که در پایگاه داده‌ی ذخیره شده‌اند.

یادگیری ماشینMachine Learningچیست؟

همان‌طور که مشاهده می‌کنید، در جدولِ بالا (که در داده‌کاوی به ماتریس معروف است)، هر سطر نمایشگرِ یک فرد خاص است. به این فرد خاص، یک رکورد یا یک نمونه یا یک sampleیا یک tupleگفته می‌شود. و هر ستون نمایشگرِ یک ویژگی یا همان featureاست. به ویژگی‌ها در داده‌کاوی اصطلاحاً بُعد dimension نیز گفته می‌شود. مثلاً داده‌های موجود در تصویرِ بالا، ۴بعدی است چون ۴ ویژگی (ستون) دارد. توجه کنید که ستونِ آخر، ستونِ برچسب‌ها یا همان lableهای ماست که مشخص میکند یک نمونه‌ی خاص، در هر سطر به کدام دسته class تعلق دارد. در این مثال ما ۲ دسته یا ۲ طبقه class داریم. کسانی که وام خود را پس داده‌اند، و کسانی که وام خود را پس نداده.اند.

به طور کلی به مسئله‌هایی که ستون طبقه یا classرا داشته باشند، مسائل طبقه‌بندی یا classificationگفته می‌شود. این دست از مسائل به یاگیریِ با ناظر supervised learning نیز معروف هستند، چون در واقع یک ناظر وجود دارد که ستون آخر را برای ما برچسب‌زنی کند (مثلاً در این‌جا مدیر بانک، تعدادی مشخصی از مشتریان را برای ما برچسب زده است).

الگوریتم‌های یادگیری ماشین و داده‌کاوی که عملِ طبقه‌بندی را انجام می‌دهند (مانند SVM، Random Forest، Naive Bayesو...) می‌توانند این جدول یا همان ماتریس را به عنوانِ ورودی قبول کنند و از این ماتریس و ویژگی‌های آن، الگوی موجود در هر طبقه یا classرا یاد بگیرند. سپس اگر یک نمونه‌ی جدید (مثلاً یک مشتریِ جدید) – که طبقه‌ی آن را نمی‌دانیم – به الگوریتمی که یادگرفته است داده شود، این الگوریتم می‌تواند این نمونه‌ی جدید را به طبقه‌های احتمالاً درست (که قبلا دیده است) طبقه‌بندی یا classificationکند. مثلاً یک مشتریِ جدید با ۴ ویژگی، به الگوریتم داده می‌شود، و الگوریتم می‌تواند با توجه به داده هایی که یادگرفته است پیش‌بینی کند که این مشتریِ جدید می‌تواند وام خود را پس دهد یا خیر؟

خوشه‌بندیClustering چیست؟

فرض کنید، شما یک فروشگاه بزرگِ موادغذایی دارید و مشتریانِ این فروشگاه که بالغ بر ۱۰۰ هزار نفر هستند ویژگی‌های مختلفی دارند. اجازه دهید، سه ویژگیِ زیر را برای یک مشتریِ خاص از مشتریان این فروشگاهِ بزرگ موادغذایی در نظر بگیریم (بقیه‌ی مشتریان نیز این ویژگی‌ها را دارند):

#	R	F	M
#1	3	4	5000
#2	7	14	15000
#3	7	10	7000
#4	1	15	25000
#5	25	30	100000
⋮	⋮	⋮	⋮

۱. این مشتری آخرین خریدِ خود را چند روز پیش انجام داده است؟ (که با Rنام گذاری می‌کنیم)

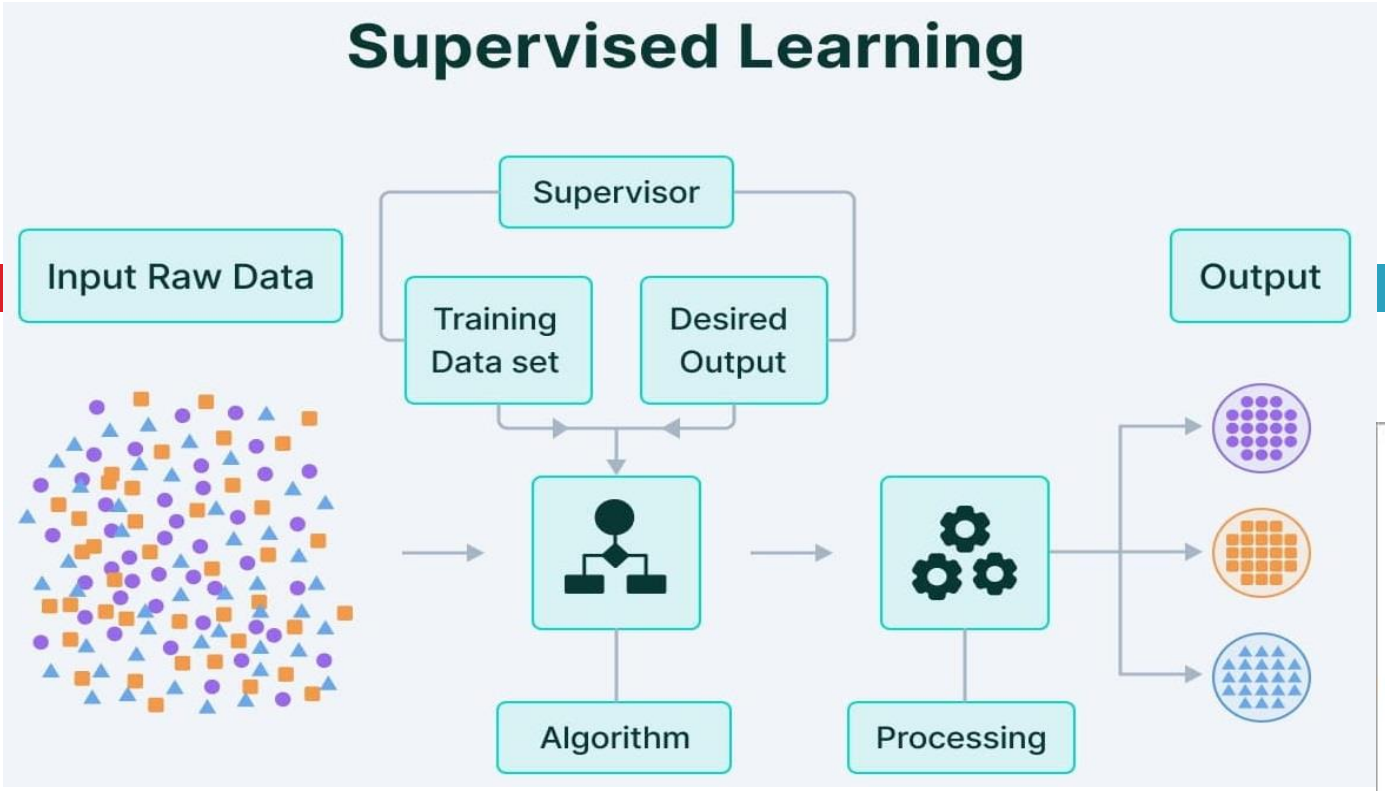
۲. این مشتری در یکسالِ گذشته، به طورِ میانگین چند روز یک بار از فروشگاه ما خرید کرده است؟ (که با Fنام گذاری میکنیم)

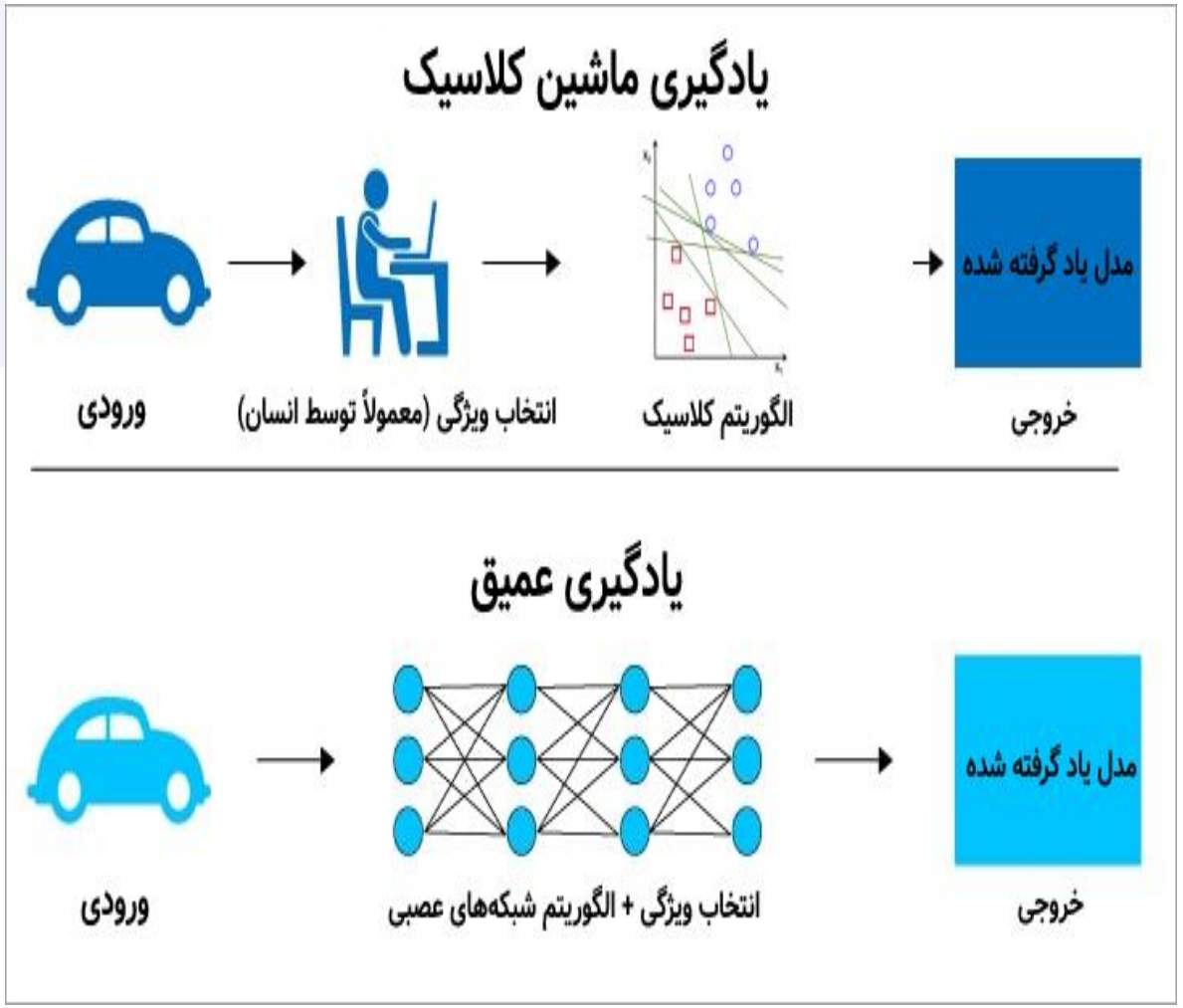
۳. این مشتری در یکسالِ گذشته به طورِ میانگین در هر بار خرید، چه مبلغی از فروشگاه خرید کرده است؟ (که با Mنامگذاری میکنیم)

حال به جدول زیر که نوعی ماتریس است نگاهی بیندازید. این‌ها قسمتی از داده‌های ما هستند:

هر سطر در این جدول، یک مشتری را نشان میدهد. ستون‌های Rو Fو Mبه ترتیب سه ویژگی یا سه بُعدِ مسئله ما را تشکیل می‌دهند که مطابق با سه ویژگیِ گفته شده در بالا است. این‌ها ۵ نمونه از ۱۰۰ هزار مشتریِ فروشگاه ما را تشکیل می دهند که در جدول بالا نمایش داده شده است. به فرد شماره ۱ توجه کنید: این فرد ۳ روز گذشته آخرین خرید خود را انجام داده است ویژگی Rدر یکسال گذشته به طور میانگین هر ۴ روز یکبار خرید انجام داده ویژگی .Fو به طورِ میانگین در یکسال گذشته در هر خرید ۵۰۰۰ تومان خرید کرده است. بقیه‌ی مشتریان را هم می‌توانید به همین ترتیب تفسیر کنید.

از لحاظِ کسب و کار قطعاً می‌دانید که نباید با تمامی مشتریان به یک صورت برخورد کنید. پس نیاز دارید تا بینِ گروهِ مشتریانِ مختلفِ خود تمایز قائل شوید. برای این کار می توانید از الگوریتم‌های خوشه‌بندی clustering یا همان یادگیری غیرنظارت شده unsupervised learning استفاده کنید. این الگوریتم‌ها می‌توانند با استفاده از ویژگی‌ها یا همان ابعادِ مسئله در اینجا Rو Fو Mگروه‌های مختلفی از نمونه‌هایی را که شبیه به هم هستند، پیدا کنند.


Supervised Learning
Supervisor
Input Raw Data
Training Data set
Desired Output
Output
Algorithm
Processing


یادگیری ماشین کلاسیک
مدل یاد گرفته شده
ورودی
انتخاب ویژگی (معمولاً توسط انسان)
الگوریتم کلاسیک
خروجی
یادگیری عمیق
مدل یاد گرفته شده
ورودی
انتخاب ویژگی + الگوریتم شبکه‌های عصبی
خروجی

خوشه‌بندی Clustering چیست؟

مثلا فرض کنید از الگوریتم معروف KMeans استفاده می‌کنیم. این الگوریتم تعداد گروه‌ها (خوشه‌ها) را از شما می‌خواهد، شما عدد ۸ را به الگوریتم می‌دهید، به این معنی که می‌خواهید الگوریتم ۱۰۰ هزار مشتریِ شما را به ۸ گروه یا همان ۸ خوشه تقسیم نماید، به صورتی که مشتریان در یک گروه، به یکدیگر شباهت‌های زیادی داشته باشند. مثلاً فرض کنید یکی از این ۸ گروه (که الگوریتم KMeans تقسیم بندی کرده است) حدوده ۱۵ هزار مشتری دارد که معمولاً دارای M و F بالایی هستند. به این معنی که این گروه پول زیادی در هر خرید خرج می‌کنند (مثلاً هر بار حدود ۱۲۰ هزار تومان ویژگی M، ولی دوره برگشتشان به فروشگاه طولانی است (مثلاً هر ۳۰ روز یکبار به فروشگاه مراجعه می‌کنند ویژگی F پس ما از میان داده‌هایمان توانستیم چندین خوشه یا گروه استخراج کنیم که یکی از این خوشه‌ها ویژگی F و M بالایی داشت. حال می‌توان برای این خوشه تصمیمِ مشخصی گرفت. برای مثال احتمالاً این خوشه بیشتر احتیاج به مایع ظرف‌شویی بزرگ دارد تا یک مایع ظرفشویی کوچک، زیرا معمولاً مشتریانِ این خوشه خریدهایی با مبلغ بالا برای مدت طولانی انجام می‌دهد، پس می‌توان در یک تبلیغ پیامکی، اجناسِ بزرگ (مانند مایعِ ظرف‌شوییِ چند کیلویی) را برای این گروه از مخاطبان فرستاد. در واقع نوعی هوشمندی در کسب و کار با استفاده از خوشه‌بندی ارائه شده است.

این یک مثال از خوشه‌بندی بود. همان‌طور که در درس طبقه‌بندی classification متوجه شدید، در طبقه‌بندی یک ستون برچسب lable داریم در حالی که در خوشه‌بندی این ستون برچسب وجود ندارد. در واقع الگوریتم‌ها و روش‌های خوشه‌بندی، می‌توانند به صورتِ غیرِنظارت‌شده و بدونِ استفاده از برچسب، عملیاتِ خوشه‌بندی را با استفاده از ویژگی‌های مختلفِ مسئله انجام دهند. به عبارت ساده‌تر خوشه‌بندی یک نوع تقسیم‌بندیِ داده‌ها با توجه به الگوها یا همان patterns ذاتیِ داده‌ها است.

الگوریتم‌های مختلف خوشه‌بندی مانند KMeans، DB Scan، OPTICS و... وجود دارند .

تفاوت طبقه‌بندی Classificationو خوشه‌بندی Clustering

کاربرد اصلیِ الگوریتم‌های یادگیری ماشین این است که اطلاعات موجود در داده‌ها را استخراج کنند یا از داده‌های موجود یاد بگیرند. مانند بچه‌ای که با مشاهده‌ی افراد و توصیه‌های والدین یادگیری را انجام می‌دهد. مثلا می‌تواند بفهمد که بخاری داغ است یا نباید به تنهایی از خیابان رَد شود. در واقع کار الگوریتم‌های یادگیری ماشین یا همان machine learningاین است که یک خلاصه summarizeاز داده‌ها را پیدا کنند یا در واقع یک مُدل model از داده‌ها با هر روشی ایجاد کنند. همان طور که می‌دانید یک مدل همواره شاملِ خلاصه‌ای از داده‌ها است. برای مثال نقشه جهان یک مدل از جهان است که کشورها و راه‌ها را بدون توجه به جزئیات زیاد (مثلاً اینکه چه مغازه‌ای در چه خیابانی است) نگاشت می‌کند.

البته این‌طور نیست که تمامیِ الگوریتم‌های یادگیری ماشین خلاصه‌سازیِ داده‌ها را انجام دهند. اگر به الگوریتم‌های خوشه‌بندی یا همان clusteringدر دوره خوشه‌بندی در درس‌های آتی نگاهی بیندازید، متوجه می‌شود که این الگوریتم‌ها می‌توانند داده‌ها را به صورت خودکار به گروه‌ها (خوشه‌ها) تقسیم بندی کنند. علاوه بر این، الگوریتم‌های خوشه‌بندی می‌توانند داده‌های جدید که از راه می‌رسند را به یکی از خوشه‌های قبلیْ که مدل کرده‌اند، ملحق کنند.

با این مقدمه اگر بخواهیم ساده نگاه کنیم، الگوریتم‌های یادگیری‌ماشین در دو دسته‌ی کلیْ قرار می‌گیرند. یکی از آن‌ها الگوریتم‌هایی است که به خوشه‌بندی clustering معروف هستند و در دوره‌ی خوشه‌بندی به آن‌ها خواهیم پرداخت و دیگری الگوریتم‌هایی که عملیات طبقه‌بندی classification را انجام می‌دهند و در دوره‌ی طبقه‌بندی به آن‌ها توجه خواهیم کرد.

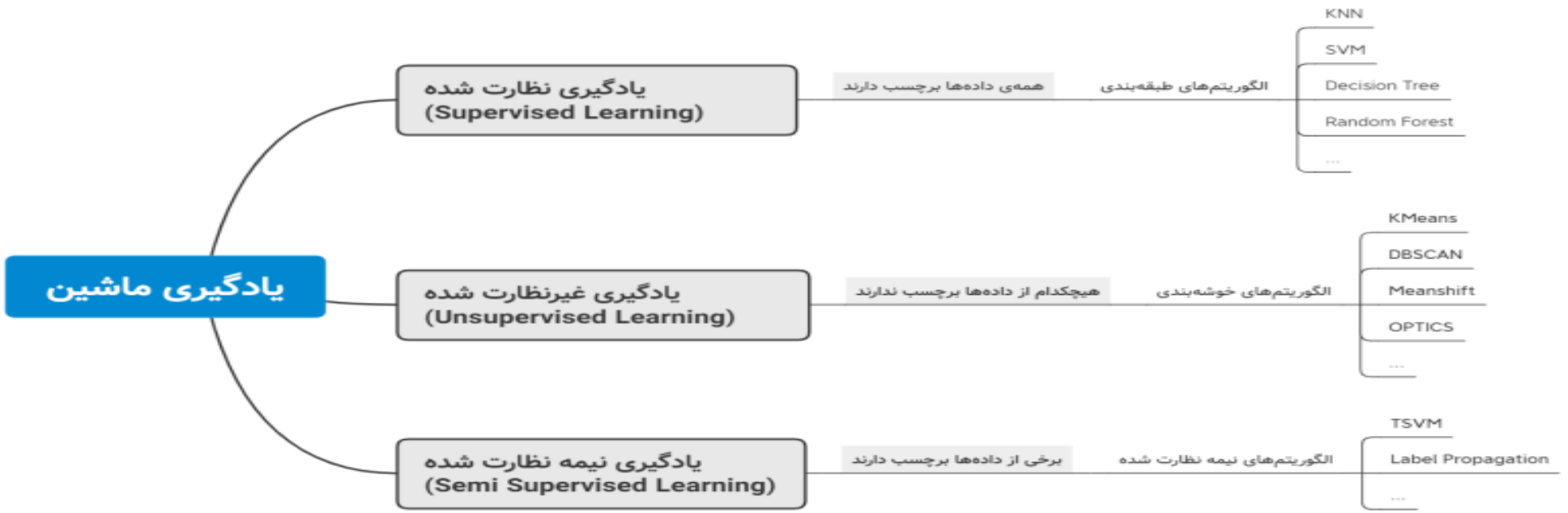
داده‌های مورد نیاز برای الگوریتم‌های خوشه‌بندی، نیاز به دانستنِ برچسبِ داده‌ها ندارند. یعنی به الگوریتم نمی‌گوییم که هر کدام از داده‌ها، در کدام دسته بندیْ قرار می‌گیرند. در واقع هیچ پیش فرضی در مورد اینکه داده‌های موجود در چه طبقه یا دسته یا گروهی قرار می‌گیرند نداریم و الگوریتم به صورت خودکار گروه‌بندیِ داده‌ها را کشف می‌کند. به همین دلیل به این دست از الگوریتم‌ها، الگوریتم‌های یادگیریِ غیرنظارت‌شده unsupervised learning می‌گویند.

تفاوت طبقه‌بندی Classificationو خوشه‌بندی Clustering

این در حالی است که گونه‌ی دیگرِ الگوریتم‌های یادگیری ماشین، الگوریتم‌های یادگیریِ ماشینِ نظارت‌شده supervised هستند. این الگوریتم‌ها، که به آن‌ها طبقه‌بندها نیز گفته می‌شود، داده‌هایی را دریافت می‌کنند که از قبلْ برچسب‌زده شده باشند. در مورد این الگوریتم‌ها نیز در دوره طبقه‌بندی بحث خواهیم کرد.

البته نوعی دیگر از الگوریتم‌ها نیز وجود دارند که چیزی بینِ دو گونه‌ی قبلی هستند. به این دسته، الگوریتم‌های یادگیری نیمه نظارت شده (semi supervised learning) گفته می‌شود. در این دسته از الگوریتم‌ها، برخی از نمونه‌ها برچسب دارند و برخی ندارند. این الگوریتم‌ها را در درس‌ها و دوره‌های آتی بررسی خواهیم کرد.

پس اگر بخواهیم یک نقشه‌ی ذهنی در مورد انواع الگوریتم‌های یادگیری ماشین داشته باشیم، می‌توانیم به شکل زیر برسیم:



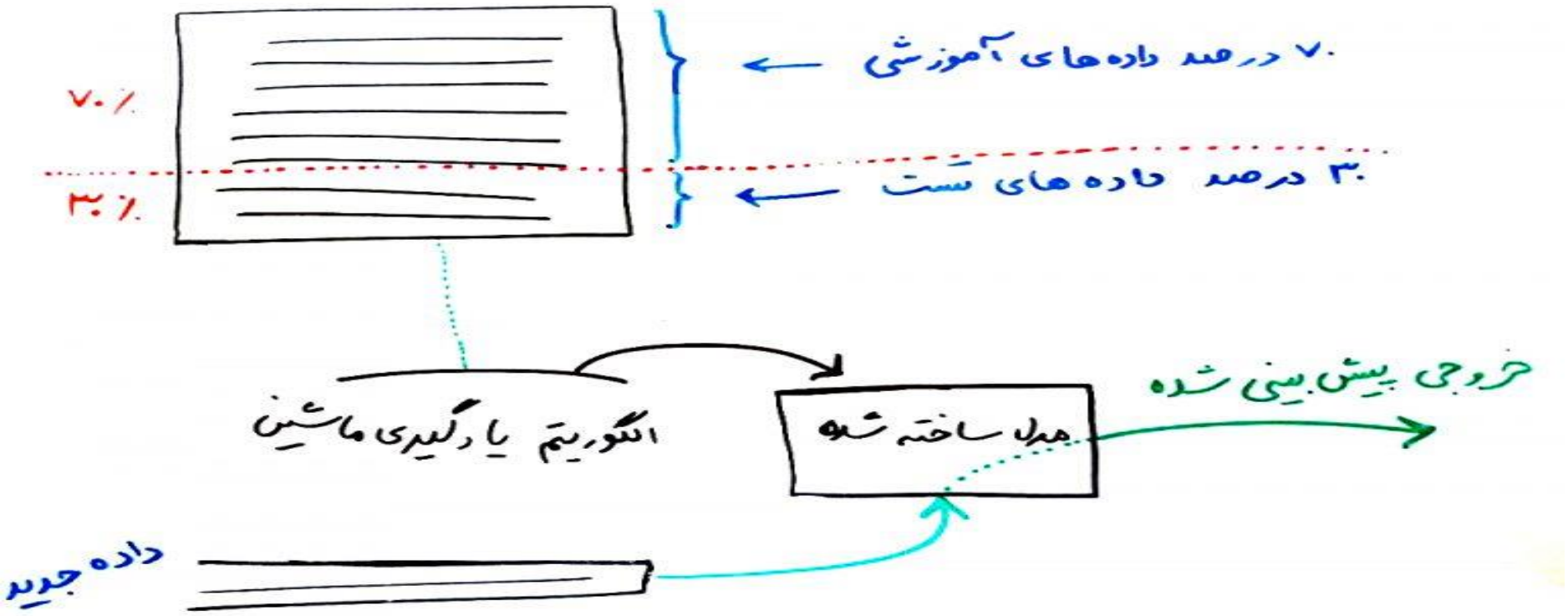
منظور از داده‌های آموزشیTraining Sets در طبقه‌بندی چیست؟

فرض کنید شما یک دانشجو هستید و معلم ۱۰۰ سوالِ نمونه همراه با جواب در اختیار شما قرار داده است. شما بایستی با خواندن این ۱۰۰سوالْ خود را برای امتحان آماده کنید. به این ۱۰۰ سوال به نوعی داده‌های آموزشی گفته می‌شود زیرا شما از این داده‌ها برای آموزش خود و آمادگی برای امتحانِ اصلی، استفاده می‌کنید. البته از آن‌جایی فرض بر این است که شما فقط ۱۰۰ سوال دارید و هیچ منبعِ دیگر در اختیار ندارید، نمی‌توانید خود را قبل از امتحان ارزیابی کنید. پس معقول است که به صورت تصادفی random ، از میان این ۱۰۰ سوال، مثلاً ۷۰ سوال را جدا کرده، آن‌ها را بخوانید و خوب یاد بگیرید. سپس ۳۰ سوالِ باقی‌مانده، داده‌های آزمایشی برای ارزیابی هستند که بایستی توسط آن‌ها، خود را قبل از آزمونِ واقعی بیازمایید. توجه کنید که ۷۰ سوال و ۳۰ سوالی که تقسیم بندی کرده‌اید، جواب‌هایش را دارید. در واقع با خواندن ۷۰ سوال و دیدن جواب‌های آن‌ها، یادگیری را انجام می‌دهید و سپس ۳۰ سوال باقی مانده را برای ارزیابیِ خود می‌گذارید. ۳۰ سوال را خوانده و برای خود جواب می‌دهید، سپس جواب‌های داده شده را با جواب‌هایی واقعیِ همان ۳۰ سوال، مقایسه می‌کنید و دقت و صحتِ خود را در پاسخ دادن به سوالات می‌سنجید. این همان کاری است که بایستی در یک الگوریتمِ یادگیری ماشین (معمولا الگوریتم‌های طبقه‌بندی) انجام شود.

شما یک مجموعه داده در اختیار دارید که هر کدام برچسبِ lableخود را دارند. حال این داده‌ها را به نسبت (مثلاً در این‌جا ۷۰ به ۳۰) تقسیم می‌کنید. الگوریتم از روی ۷۰ درصدِ داده‌ها، عملیاتِ یادگیری را انجام می‌دهد و از روی ۳۰ درصد بقیه، خود را ارزیابی می‌کند و نتیجه‌ی تست را به شما می‌گوید. به این ترتیب می‌توانید بفهمید که این الگوریتم چه مقدار دقت دارد. در واقع هنگامی که داده‌های واقعی از راه می‌رسند، می‌خواهیم بدانیم که این الگوریتم چقدر می‌تواند دقت داشته باشد (منظور از داده‌های واقعی، داده‌هایی است که در مجموعه داده‌های آموزشی نیستند و می‌خواهیم واقعا عملیاتِ داده‌کاوی و طبقه‌بندی را بر روی آن‌ها انجام دهیم).

منظور از داده‌های آموزشی Training Sets در طبقه‌بندی چیست؟

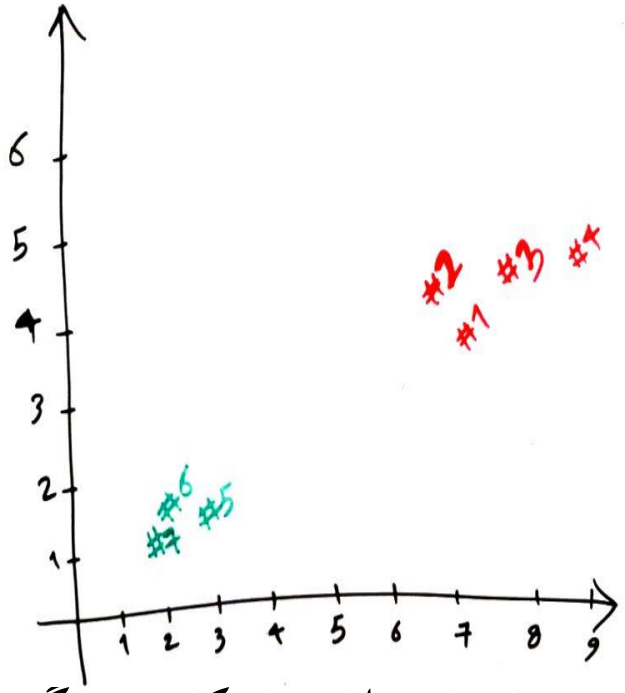
در هنگامِ آموزش، داده‌ها را به دو دسته‌ی آموزشی و ازریابی تقسیم‌بندی می‌کنیم. حال الگوریتم از روی داده‌های آموزشی یادگیری را انجام می‌دهد و از روی داده‌های ارزیابی یا همان داده‌های تست می‌توانید بفهمید که الگوریتم و مدلِ ساخته شده توسطِ آن، چقدر دقت داشته است. وقتی الگوریتمْ عملیاتِ یادگیری را انجام داد و در واقع یک مدل را از روی این داده‌ها ساخت، حالا می‌توان از روی این مدل، عملیاتِ داده‌کاوی را بر روی داده‌های جدید انجام داد.



ویژگی یا همان بُعد Dimension Feature در داده‌کاوی چیست؟

ویژگی Feature یا بُعد Dimension در واقع پایه‌ی بسیاری از عملیاتِ داده‌کاوی و یادگیری‌ماشین است. در این درس می‌خواهیم این مفاهیمِ ساده را با یکدیگر مرور کنیم تا در ادامه راه، بتوانیم ادبیاتِ مشترکی در حوزه داده‌کاوی و یادگیری‌ماشین داشته باشیم.فرض کنید شما یک مجموعه‌ی داده را در اختیار دارید که می‌خواهد تفاوتِ بین اتوبوس و پراید را بر حسب دو ویژگیِ طول و ارتفاع درک کند. مثال را خیلی ساده در نظر بگیرید. ما یک سری ماشین داریم که از هر کدام از آن‌ها فقط دو ویژگی را در نظر گرفته‌ایم. جدول زیر نشان دهنده‌ی همین موضوعات است، همان‌طور که مشاهده می‌کنید، ۷ عدد ماشین، دو ویژگی دارند.ویژگیِ اول طول و ویژگیِ دوم ارتفاع است. حال همین دو ویژگی را می‌توان بر روی محور مختصاتِ دو بعدی نمایش داد. مانند شکل زیر:

#	طول	ارتفاع	نوع
1	7	4	اتوبوس
2	6/5	4/5	اتوبوس
3	7/5	4/5	اتوبوس
4	9	4/5	اتوبوس
5	3	1/5	پراید
6	2/5	1/7	پراید
7	2	1/6	پراید



محورِ افقی بیانگر طول و محور عمودی بیانگر ، ارتفاع اتومبیل می باشد. همان‌طور که می‌بینید، نمونه‌ی اول که دارای طولِ ۷ و ارتفاعِ ۴ است بر روی محورِ مختصات نمایش داده شده است، و بقیه‌ی اتومبیل‌ها هم به همین ترتیب.

در واقع ما دو ویژگی داریم که به هر کدام از آن‌ها یک بُعد نیز گفته می‌شود. پس مجموعه داده‌های فعلی ما دو بُعدی است. ممکن است داده‌ها برای مثال ۳ ویژگی داشته باشند که آنوقت می‌توانیم آن ها را در یک فضای ۳بعدی رسم کنیم. داده‌هایی با بیشتر از ۳ ویژگی نیز بسیار متداول هستند که رسمِ آن ها سخت‌تر است ولی در ذهن می‌توانید آن ها تصور کنید. برای مثال یک مجموعه داده می‌تواند ۱۰۰ بُعدی باشد. یعنی دارای ۱۰۰۰ ویژگی باشد.

ویژگی‌ها و بُعدها در مسایل داده‌کاوی و یادگیری ماشین بسیار مهم هستند و در واقع پایه‌ی بسیاری از عملیات داده‌کاوی و یادگیری ماشین به حساب می‌آیند.

سید ایمان غفوریان - عضو هیات علمی دانشگاه آزاد مشهد-زمستان ۱۴۰۲

بررسی چند الگوریتم یادگیری ماشین Machine Learning

درا ین درس سعی داریم تا به معرفیِ چند روش و الگوریتم در حوزه‌ی یادگیری ماشین با تمرکز بر بخش طبقه‌بندی classificationبه صورت خلاصه بپردازیم تا کمی فضای ذهنی خود را با این اسامی آشنا کنیم.

۱. درخت‌های تصمیم Decision Trees

درختانِ تصمیم که چند درس از دوره طبقه‌بندی را در چیستو به آن اختصاص دادیم، یکی از روش‌های ساده اما کاربردی در حوزه‌ی یادگیری‌ماشین هستند. این درخت‌ها با ایجادِ شاخه‌ها و برگ‌ها، یادگیری را انجام می‌دهند و می‌توانند در طیف وسیعی از مسائل، کاربرد داشته باشند. این الگوریتم‌ها معمولاً بیش‌برازش یا همان Overfitشده و شاید نتوانند دقت خوبی داشته باشند. البته نسخه‌های بهبود یافته‌ی این الگوریتم‌ها می‌توانند از overfittingدوری کنند.

۲. پرسپترون‌ها Perceptrons

در دوره‌ی شبکه‌های‌عصبی مقدمه‌ای بر پرسپترون گفتیم و به این نکته رسیدیم که پرسپترون می‌تواند یک حدِ آستانه‌ی خوب را پیدا کند و با این حد آستانهْ Threshold تفاوتِ نمونه‌های مختلفْ را درک کند (مثال اتوبوس و پراید در دوره شبکه‌های عصبی).

۳. شبکه‌های عصبی

در واقع شبکه‌های عصبی یک شبکه‌ای از پرسپترون‌ها هستند. شبکه‌ای بدون دور acyclicکه لایه‌های مختلف دارد. خروجی هر لایه می‌تواند ورودی لایه بعدی باشد تا به لایه‌ی آخر یا همان لایه‌ی خروجی برسیم.

۴. KNN

یادگیری بر اساس مورد Instance Base Learning. یک مثال ساده این است که فرض کنید شما ۱۰ دوست دارید که هر کدام زندگیِ متفاوتی دارند. این افراد در سه دسته‌ی دوستِ عادی، دوستِ صمیمی، دوستِ خیلی خیلی صمیمی برای شما قرار می‌گیرند

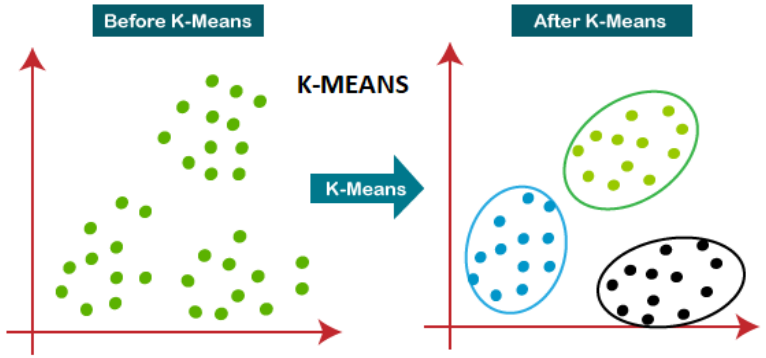
بررسی چند الگوریتم یادگیری ماشین Machine Learning

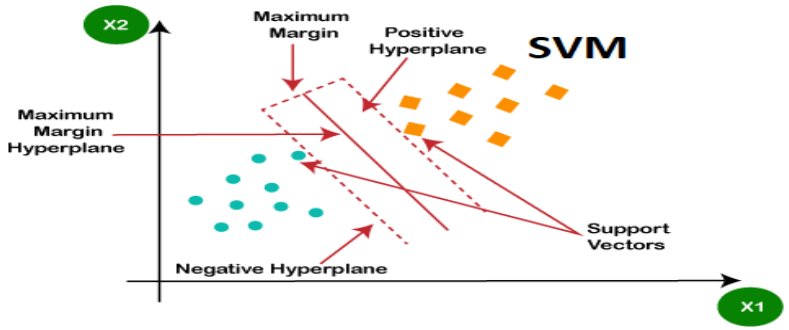
حالْ یک نفرِ جدید را پیدا می‌کنید که به نظرِ شما زندگیِ او مانند یکی از دوستان قبلی شماست که اتفاقا دوست خیلی خیلی صمیمی بوده است، پس احتمالاً شما به آن نفر (به خاطر نزدیکی شرایط زندگی‌اش به دوستِ خیلی خیلی صمیمی شما) حس خیلی خیلی خوبی دارید (که البته ممکن است درست نباشد). در واقع شما اینجا از KNNبا پارامترِ ۱ استفاده کرده‌اید. یعنی مشاهده کرده‌اید که این فرد جدید به کدام ۱ نفر از دوستان قبلی شما بیشتر شباهت دارد و این نفر جدید را در دسته‌ای قرار داده‌اید که دوست قبلی شما در آن دسته قرار داشته است.

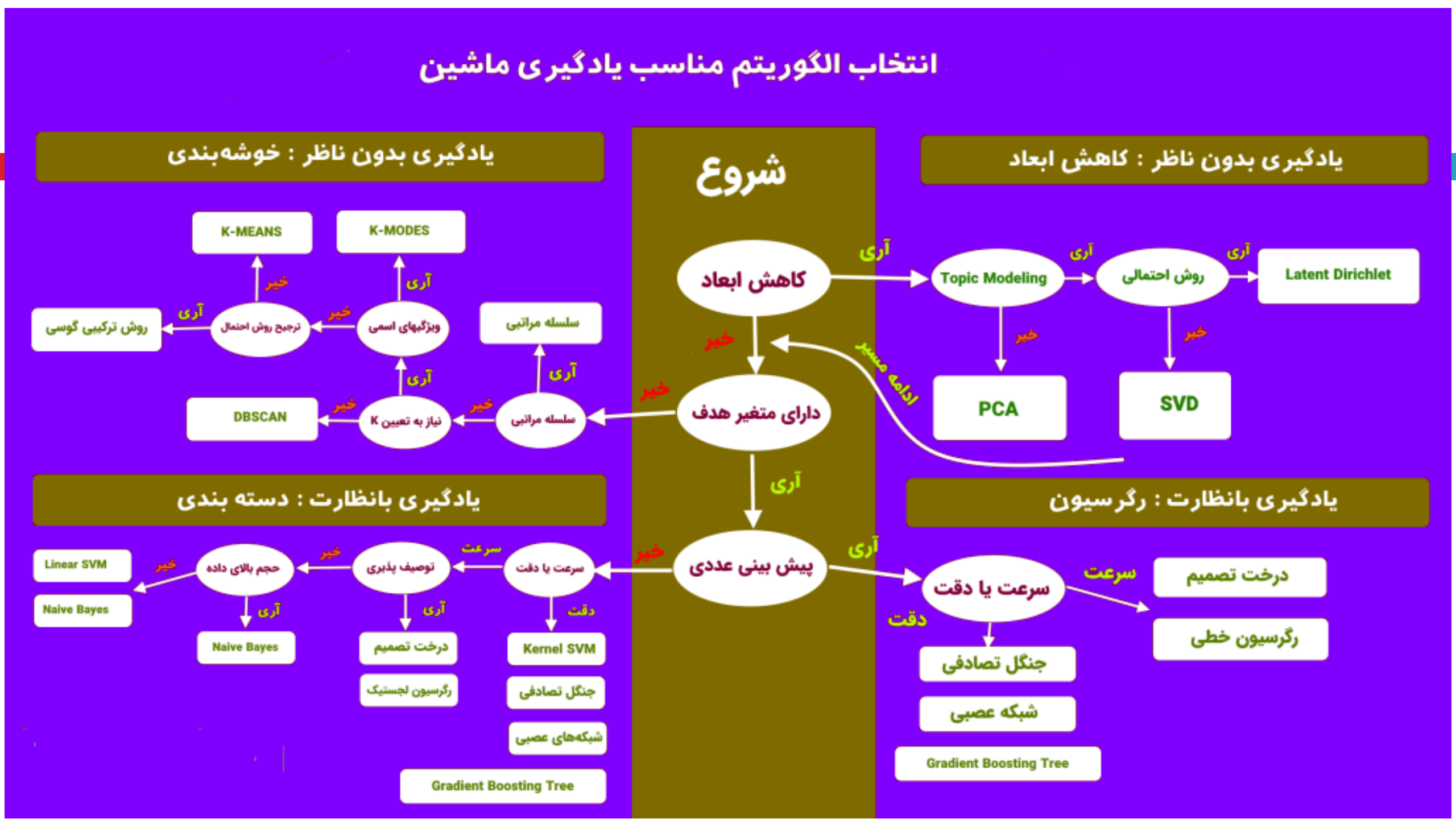
۵. ماشین بردار پشتیبان Support Vector Machines

این روش به مراتب پیشرفته‌تر از الگوریتم‌های قدیمی است که روشِ یادگیریِ آن شبیه پرسپترون است و بهتر می‌تواند توازن balanceبین bias وvariance را رعایت کند.

الگوریتم‌های گفته شده در بالا یک سری از الگوریتم‌های معروف در یادگیری ماشین، در زیر شاخه‌ی طبقه‌بندی داده‌ها هستند. تعداد بسیار زیاد دیگری از الگوریتم‌ها و روش‌ها وجود دارند که برای داده‌های مختلف می‌تواند مورد استفاده قرار گیرد. البته اینکه کدام الگوریتم را باید برای کدام یک از داده‌ها استفاده کرد نیز نیاز به تجربه و آزمایش بر روی داده‌های واقعی دارد.






انتخاب الگوریتم مناسب یادگیری ماشین
شروع
یادگیری بدون ناظر : کاهش ابعاد
کاهش ابعاد
آری
Topic Modeling
آری
روش احتمالی
آری
Latent Dirichlet
خیر
PCA
خیر
SVD
ادامه مسیر
خیر
دارای متغیر هدف
آری
یادگیری بدون ناظر : خوشه‌بندی
سلسله مراتبی
آری
ویژگیهای اسمی
آری
K-MODES
خیر
ترجیح روش احتمال
خیر
K-MEANS
آری
روش ترکیبی گوسی
خیر
سلسله مراتبی
خیر
نیاز به تعیین K
خیر
DBSCAN
یادگیری بانظارت : رگرسیون
پیش بینی عددی
آری
سرعت یا دقت
سرعت
درخت تصمیم
رگرسیون خطی
دقت
جنگل تصادفی
شبکه عصبی
Gradient Boosting Tree
یادگیری بانظارت : دسته بندی
خیر
سرعت یا دقت
سرعت
توصیف پذیری
خیر
حجم بالای داده
خیر
Linear SVM
Naive Bayes
آری
Naive Bayes
آری
درخت تصمیم
رگرسیون لجستیک
دقت
Kernel SVM
جنگل تصادفی
شبکه‌های عصبی
Gradient Boosting Tree

یادگیری دسته‌ای Batch Learning و یادگیری برخط Online Learning

مبحثِ یادگیری (چه یادگیری‌ماشین و چه داده‌کاوی) را از ابعاد گوناگون می‌توان نگاه کرد. یکی از این ابعاد می‌تواند تفاوت بین یادگیریِ دسته‌ای یا همان batch Learningو در مقابلِ آن، یادگیریِ برخط یا همان online learningباشد. در واقع درک تفاوت این دو روش، می‌تواند به حل مسائلِ مختلفِ این حوزه کمک کند.
تفاوتِ این دو دسته یادگیری را می‌توان با یک مثال ساده توضیح داد. فرض کنید یک دانش آموز می‌خواهد آمار و احتمالات را فرا بگیرد.
در نوع اول یادگیری، این دانش‌آموز می‌تواند یک مجموعه کتابِ آمار و احتمالات را تهیه کند، چند بار بخواند و یاد بگیرد. بعد از این‌که از این مجموعه کتابِ آمار، یادگیریِ خود را انجام داد، دیگر مطلبِ جدیدی یاد نگیرد و از این به بعد فقط از دانسته‌های خود استفاده کند.

این روش یادگیری، نوعی یادگیریِ دسته‌ای batch learningاست.در این روش یادگیری، تمامیِ داده‌ها در هنگام آموزش در دسترس هستند و بعد از فازِ آموزش، دیگر یادگیری نخواهیم داشت.
اما یادگیریِ دوم یا همان یادگیریِ برخط online learningمانندِ این است که این دانش‌آموز ابتدا کتاب‌های خود را مطالعه کند و از آن‌ها یاد بگیرد و سپس در حین استفاده از دانسته‌های خود، هر گاه کتاب جدیدی در حوزه‌ی آمار و احتمالات مشاهده کرد، تهیه کرده و با خواندنِ آن، یادگیریِ خود را بهبود ببخشد.

داده‌هایی که معمولا در دوره داده‌کاوی، دوره طبقه‌بندی و دوره خوشه‌بندی بر روی آن‌ها بحث می‌کردیم، داده‌هایی از نوع دسته ای batchبودند به این معنی که در زمانِ یادگیری train، تمامی داده‌ها در اختیار الگوریتم بودند و الگوریتم به هر نحوی می‌توانست بر روی داده‌ها عملیاتِ یادگیری را انجام دهد.
اما نوع دیگری از یادگیری نیز وجود دارد. برخی اوقات داده‌ها به صورت جریان‌داده می‌آیند و یا اینکه نیاز دارند به صورت مرتب یاد گرفته شوند.

یادگیری دسته‌ای Batch Learning و یادگیری برخط Online Learning

پس یادگیریِ دسته‌ای مانند این است که شما یک مجموعه کتاب دارید و باید این کتاب‌ها را یاد بگیرید چون فردا امتحان دارید. تمام منبع شما همین کتاب‌ها است و در واقع تمامیِ داده‌ها را در اختیار دارید. اما فرض کنید در مسیر زندگی قرار دارید و روزانه با اطلاعات جدیدی که به شما داده می‌شود می‌بایستی هر روز چیزهای جدید را یاد گرفته و به دانش قبلیِ خود اضافه کنید. این موردِ دوم یادگیریِ برخط است، یعنی هنگامی که تمامی داده‌ها در حالِ حاضر موجود نیستند. در یادگیریِ برخطْ یک مدل ساخته می‌شود و بعد با رسیدن داده‌های جدیدتر، این مدل به‌روزرسانی می‌شود.

مدل یادگیری بر خط یا همان online learningدو مزیت اساسی دارد:

۱. با این روش می‌توان داده‌هایی با حجم بسیار بالا را آموزش داد. برای مثال داده‌هایی که به دلیل حجم بالای خود در حافظه جا نمی‌شوند.

۲. تغییراتی که ممکن است در ذات داده‌ها به وجود بیاید با این روش پوشش داده می‌شود. فرض کنید گوگل برای سیستمِ ایمیلِ خود یک الگوریتم توسعه داده باشد که ایمیل‌های هرزنامه spamرا به صورت هوشمند با الگوریتم‌های یادگیری‌ماشین تشخیص بدهد.همان‌طور که حدس می‌زنید محتوای ایمیل‌های هرزنامه مدام در حال تغییراست و انسان‌هایی که ایمیل‌های spamمی فرستند هر روز خود را در مقابل این الگوریتم‌های گوگل، بهینه می‌کنند. پس الگوریتمِ تشخیص ایمیلِ spam در گوگل می‌تواند به صورتِ برخط onlineیادگیری را انجام دهد تا بتواند ایمیل‌هایی را که در گذرِ زمان تغییر کرده‌اند و spamهستند تشخیص دهند. در واقع یادگیریِ الگوریتم ، با تغییر محتوا و شکل ایمیل‌های spam، بروز شده و مدلِ یادگرفته شده را مقاومت‌تر می‌کند.

یادگیری فعالActive Learningدر یادگیری ماشین

در یادگیریِ دسته‌ای، تمامی داده‌ها در هنگام یادگیری در اختیار الگوریتم قرار دارد ولی در یادگیری برخط، داده‌ها به صورت جریانی از داده از راه می‌رسند و تمامی داده‌ها در هنگام یادگیری در اختیار الگوریتم نیست.

در این بخش می‌خواهیم به مفهوم یادگیری فعال یا همان active learningدر داده‌کاوی و یادگیری ماشین بپردازیم که در واقع نوعی یادگیری برخط online learningاست.

با یک مثال ادامه دهیم، فرض کنید یک منشی قرار است نامه‌های رسیده به یک سازمان را به دو دسته‌ی نامه‌های بخش کارشناسی و نامه‌های بخش مدیریتی تقسیم‌بندی کند. در اولین روزهای استخدام چندین نمونه نامه‌ی کارشناسی و چندین نمونه نامه‌ی مدیریتی توسط سرپرست به این شخص آموزش داده شده است فرآیند learning و منشی این تفکیک کردنِ نامه‌ها را یادگرفته است. حالا این منشی می‌تواند نامه‌های جدید را طبقه‌بندی کند (مانند الگوریتم‌های طبقه‌بندی). حال فرض کنید بعد از مدتی یک نامه‌ی جدید به منشی می‌رسد که منشی احتمال می‌دهد این نامه به بخش کارشناسی تعلق دارد ولی مطمئن نیست. برای همین این نامه را نزدِ سرپرست می‌برد و از او سوال می‌پرسدکه این نامه برای کدام بخش است؟سرپرست مثلاً می‌گویداین نامه مربوط به بخش کارشناسی است.حالا منشی این نامه را به بخش کارشناسی طبقه‌بندی می‌کندو علاوه بر آن یاد می‌گیرد که این دست از نامه‌ها به بخش کارشناسی مربوط هستند.یعنی از این به بعد یادگیریِ او بهبود می‌یابد و نامه‌های جدید را دقیق‌تر طبقه‌بندی می‌کند. این نوعی یادگیریِ فعال یا همان active learningاست که در حین کار برای این منشی صورت گرفته است.

یادگیری فعالActive Learningدر یادگیری ماشین

روش‌های یادگیریِ فعال در داده‌کاوی هم به همین صورت هستند. در واقع الگوریتم‌هایی مانند طبقه‌بندها ، بعد از اینکه یادگیری را انجام دادند بایستی عملیاتِ پیش‌بینی و طبقه‌بندی را بر روی داده‌های جدید انجام دهند. حال اگر یک الگوریتم نتواند بگوید داده‌ی جدیدی که از راه رسیده به طور قطع متعلق به کدام طبقه است،می‌تواند این داده‌ی جدید را به یک ناظر supervisor یا همان کسی که بتواند برچسب دقیق بر روی داده بزند، بفرستدو از او برچسب واقعی داده‌ها را دریافت کرده ومانند یک الگوریتم برخط online این نمونه را که در طبقه‌بندیِ آن شک داشت، حالا با دانستنِ کلاسِ واقعیِ آن (که توسط ناظر برچسب خورده است) به مدلِ آموزشی خود اضافه کند و یادگیری را به صورت بر خط انجام دهد. در واقع این‌جا نوعی یادگیری فعال active learning برای الگوریتم اتفاق افتاده است.

برای نمونه می‌توان همان مثالِ شرکتِ گوگل در بخش قبل را دید. گوگل برای اینکه بتواند بفهمد یک ایمیل ارسالی، هرزنامه Spam هست یا خیر، یک الگوریتم توسعه‌داده است که بر اساس یک سری داده‌های آموزشی، یادگیری را انجام می‌دهد و حالا می‌تواند ایمیل‌های جدیدی که از راه می‌رسند را به طبقه‌های هرزنامه یا غیرهرزنامه طبقه‌بندی کند. یک ایمیل جدید می‌رسد که الگوریتم فکر می‌کند این ایمیل هرزنامه است ولی مطمئن نیست. پس این ایمیل را به یک فرد مختصص می‌فرستد تا این فرد پس از بررسی، مشخص کند این ایمیل جزو ایمیل‌های هرزنامه هست یا خیر؟
سپس طبقه‌ی اصلی این ایمیل توسط آن فرد خبره مشخص شده و حالا الگوریتم این نمونه را به صورت بر خط online به مدلِ یادگرفته‌شده‌ی خوداضافه می‌کندودر واقع مدلِ یادگرفته شده‌ی خودرا به صورت فعال تکمیل می‌کند.(البته شرکتی مانند گوگل راه‌های بهتر و پیچیده‌تری برای این کار دارد ، مثلا اگر شما یک ایمیل را از هزرنامه خود خارج کردید احتمالا الگوریتم یاد می‌گیرد که این ایمیل از این به بعد هرزنامه نیست)

انتخاب ویژگیFeature Selectionچیست؟

در مواقعی که بحث کار عملی (و صنعتی)بر روی داده‌ها پیش می آید واز مباحث تئوری و آکادمیکِ دانشگاهی فاصله می‌گیریم، شاید مهم‌ترین بخش برای عملیاتِ داده‌کاویْ عملیاتِ انتخابِ ویژگی است . در این بخش می‌خواهیم بیشتر به عملیاتِ انتخاب ویژگی یا همان Feature Selection که به نظر مهمترین بخش عملیات داده‌کاوی و یادگیری‌ماشین است بپردازیم.

مثال :شرکت گوگل می‌خواهد یک الگوریتم توسعه دهد که با آن بتواند بفهمد که یک ایمیل هرزنامه است یا خیر؟ برای این کار بایستی ویژگی‌های مختلفی را جمع آوری کند، برای مثال یکی از مجموعه ویژگی‌ها می‌تواند بردار TF-IDFباشد .بردار TF-IDFبرداری است که از روی کلمات می‌تواند ویژگی‌های مختلف را برای یک متن بسازد (در واقع متن را تبدیل به اعداد قابل فهم برای الگوریتم کند).همان‌طور که می‌دانید محتوای اصلی یک ایمیل متنِ آن است. پس گوگل از متن‌های موجود در ایمیل یک مجموعه ویژگی می‌سازد.مثلا اینکه تعداد تکرار کلمه‌ی "تبلیغ"در متن یک ایمیل چقدر بوده است؟ یا تعداد تکرار کلمه "جایزه" در یک ایمیل چقدر بوده است؟الگوریتم یادگیری‌ماشین با استفاده از این دست ویژگی‌ها می‌تواند به بفهمد یک ایمیل هرزنامه هست یا خیر.

ولی آیا تمامِ ویژگی‌ها برای طبقه‌بندیِ یک ایمیل می‌تواند صرفاً از روی متونِ آن به دست آید؟ در این مثال شاید بتوان ویژگی‌ها یا همان ابعادِ دیگری را نیز از ایمیل‌ها استخراج کرد و به الگوریتم یاد داد. مثلا اینکه IPارسال کننده کدام است؟ یعنی ممکن است IP ارسال کننده نیز در طبقه‌بندی تاثیر داشته باشد چون برخی از ارسال کننده‌های هرزنامه Spam از IPهای مشخصْ ایمیل‌های هرزنامه را ارسال می‌کنند و الگوریتمِ یادگیری‌ماشین می‌تواند این IP ها را در طبقه‌بندیِ ایمیل (به هرزنامه یا غیرِ هرزنامه) تاثیر دهد. مثال اتوبوس و پراید را به یاد بیاورید .در آن مثال دو ویژگیِ طول و ارتفاع ماشین جهت طبقه‌بندی در نظر گرفته شده بود. اگر بخواهیم ویژگی‌های دیگری به آن مسئله اضافه کنیم، چه ویژگی‌هایی می‌تواند باشد؟ کمی فکر کنید. برای مثال شاید تعداد سرنشینانِ ماشین، یکی از ویژگی‌هایی باشد که بتوان به دو ویژگی دیگر اضافه کرد و با کمک آن بتوان کیفیت طبقه‌بندی را بهبود بخشید.

در واقع با انتخاب و مهندسی ویژگی Feature Engineeringمی‌توان ویژگی‌هایی را به مسئله اضافه کرد که دقتِ عملیات داده‌کاوی (طبقه‌بندی یا خوشه‌بندی) را افزایش دهد

منظور از متغیر وابسته Dependent و مستقل Independent

	معدل کل	تعداد مقالات	مدرک IELTS زبان	سنوات تحصیلی	دکتری قبول شده؟
#1	19,5		1	3	بلی
#2	16,5	0		4	خیر
#3	15	0	0	3	خیر
#4	17	2	1	2,5	بلی
#5	18,5		0	2,5	بلی
#6	15,5	1		2,5	خیر
#7	19	3	1	3	بلی

از دروس گذشته، شکلی مانندِ شکلِ زیر را به یاد بیاورید:

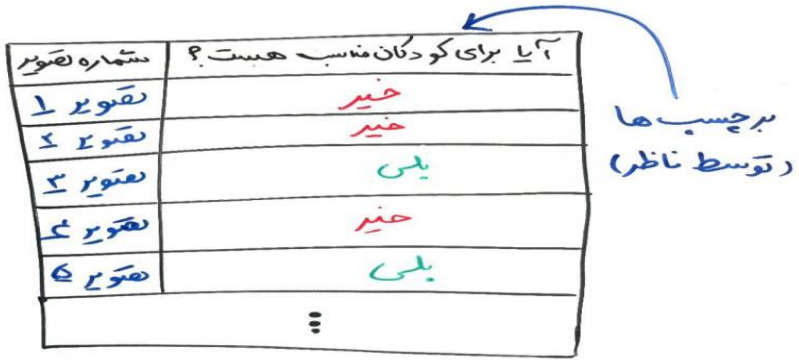
این یک مجموعه‌ی داده (دیتاست) شامل ۷ دانشجو بود که هرکدام ۴ ویژگی داشتند.می‌خواستیم از روی این ۴ ویژگی، یادگیری را انجام دهیم و بعد از یادگیری توسط الگوریتم، پیش‌بینی کنیم که یک دانشجوی جدید، می‌تواند دکتری قبول شود یا خیر. در این‌جا دو دسته داده داریم. داده‌هایی که ویژگی‌های ما را می‌سازند و عموماً به صورتِ X نمایش داده می‌شوند. در مثالِ مقابل چهار ستونِ اول (معدل کل، تعداد مقالات، مدرک IELTS زبان و سنواتِ تحصیلی) داده‌هایی هستند که به متغیر مستقل independent معروف هستند. چون وابسته به متغیرهای دیگری نیستند و در واقع مستقل از متغیرهای دیگر هستند.اما ستونِ آخر،ستون برچسب‌ها هستند که معمولاً با yنمایش می‌دهند،و به متغیرِ وابسته dependent معروف هستند. در این‌جا ستونِ دکتری قبول شده؟ یک متغیر وابسته است. زیرا ما از ۴ ستونِ قبلی استنتاج می‌کنیم که آیا شخصی با این ویژگی‌ها می‌تواند دکتری قبول شود یا خیر. برای مثال، شخص شماره #۱ با معدلِ کل ۱۹/۵، دارای مدرک زبان و ۳ سال سنوات تحصیلی توانسته دکتری قبول شود (حتماً توجه دارید که در موردِ شخصِ شماره‌ی ۱ مقدارِ ویژگیِ تعداد مقالات، خالی بود، یعنی برای این شخص داده‌ی تعداد مقاله مفقود شده است. همان‌طور که می‌بینید، این متغیرِ آخر (دکتری قبول شده یا خیر؟) وابسته به مقادیر مستقل یعنی همان Xها است. برای مثال در نمونه‌ی بالا ما می‌خواهیم از روی متغیر مستقل X ، متغیر وابسته y را پیش‌بینی کنیم.

از این مجموعه داده‌های مستقل و وابسته در عملیات یادگیری ماشین، طبقه‌بندی و رگرسیون بسیار زیاد استفاده می‌شود.

مجموعه داده‌هایی با ابعاد زیاد High Dimensional

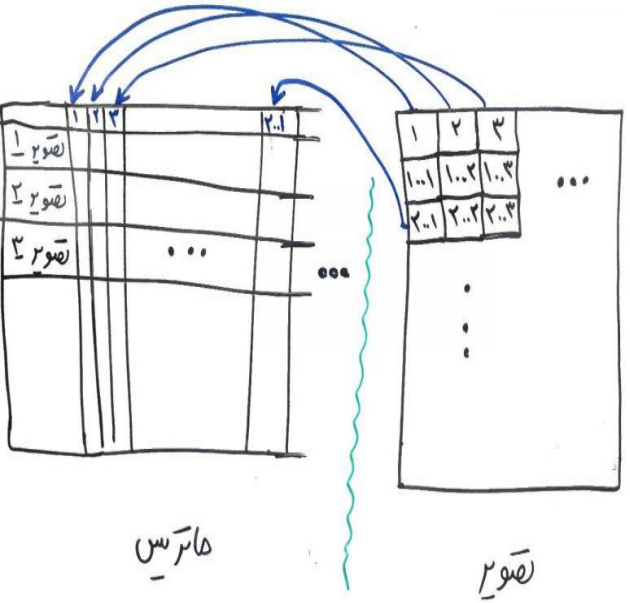
در داده‌کاوی و یادگیری ماشین، بسیاری از مواقع، داده‌ها دارای ویژگی‌های مختلفی هستند که آن‌ها را ابعاد نیز می‌نامند. مثلاً در مثال های قبلی دیدیم که برای تعیین نوع اتومبیل، دو ویژگیِ طول و ارتفاع را در نظر گرفتیم که هر کدام از این‌ها یک بُعد در فضا بودند. پس مسئله در آن درس ۲بُعدی بود. اما ممکن است یک مجموعه‌ی داده دارای ابعاد بیشتری نیز باشد که می‌خواهیم در مورد آن، در این بخش بیان نماییم.

فرض کنید مجموعه‌ای از تصاویر دارید و می‌خواهید هر کدام از آن‌ها را با توجه به ماهیتِ این تصاویر شناسایی کنید، به این صورت که آیا این تصویر مناسب کودکان هست یا خیر؟می‌دانید که این مسئله یک مسئله‌ی طبقه‌بندی است.برای این‌کار بایستی یک مجموعه‌ی تقریباً بزرگ از تصاویر را توسط یک شخصِ ناظر supervisor یعنی شخصی که مناسب یا نامناسب بودن تصویر برای کودکان را متوجه می‌شود برچسب بزنید. مثلاً یک تصویر را به این شخصِ ناظر بدهید و به او بگویید که این تصویر برای کودکان مناسب هست یا خیر؟ هر جوابی که اوداد به عنوان برچسب برای آن تصویر قرار می‌گیرد و به این صورت مجموعه‌ی آموزشی training set که خوراک الگوریتم طبقه‌بندی هست را می‌سازید. در واقع وظیفه‌ی اصلیِ الگوریتم طبقه‌بندی، این است که از روی این تصاویرِ برچسب‌زده شده توسط ناظر، یادگیری را انجام دهد و به حدی یاد بگیرد که از این به بعد خود (الگوریتم بدون دخالت شخصِ ناظر) بتواند تصاویرِ جدید را برچسب بزند. شکل زیر را نگاهی بیندازید:



مجموعه داده‌هایی با ابعاد زیاد High Dimensional

همان‌طور که مشاهده می‌کنید در این شکل، تعدادی تصویر به یک شخصِ ناظر داده شده است تا این شخص با توجه به تجربه‌ی خود، هر کدام از تصاویر را (با توجه به مناسب بودن برای کودکان) برچسب بزند. در نهایت الگوریتمِ یادگیری ماشین بایستی از این مجموعه‌ی داده، یادگیری را انجام دهد تا بتواند داده‌های جدیدتر را بدون نیاز به شخصِ ناظر طبقه‌بندی classification کند.می‌دانید که بایستی داده‌ها را برای کامپیوتر و الگوریتم‌هایی مانند الگوریتم‌های طبقه‌بندی، آماده کرده و برای این کار بایستی داده‌ها به فُرمت قابل فهم برای الگوریتم تبدیل شوند.پس در مثال بالا نیاز است تا تصاویر به یک فرمتی مانند ماتریس، که خود از بردار تشکیل شده و بردار نیز از عدد تشکیل شده است تبدیل شود (درس ماتریس و بردار را خوانده باشید). به این صورت تصاویر به فُرمتِ قابل فَهم برای الگوریتم تبدیل می‌شوند. شکل زیر را نگاه کنید:رض کنید تصاویر سیاه و سفید هستند یعنی رنگی نیستند. در این شکل ما می‌خواهیم یک تصویر مانند تصویر سمت راست را به بردارهای سمت چپ تبدیل کنیم. برای این‌کار هر پیکسل از تصویر یک خانه از بردار می‌شود و این تصاویر در کنارِ هم ماتریس مجموعه‌ی داده را تشکیل می‌دهند (هر سطر از ماتریس یک تصویر است و هر ستون یک پیکسل از آن تصویر است). برای مثال ستونِ شماره‌ی ۱ در ماتریسِ بالا، نشان‌دهنده‌ی پیکسل اول (از چپ بالا) برای هر تصویر است که می‌تواند عددی از ۰ تا ۲۵۵ بسته به روشنایی آن پیکسل در بازه‌ی سفید تا سیاه داشته باشد (چون تصویر سیاه و سفید است) و به همین ترتیب بقیه‌ی ستون‌ها. پس در این ماتریس به تعداد پیکسل‌های تصاویر، ستون داریم. ما این ماتریس را برای تزریق به الگوریتم‌هایی مانند الگوریتم‌های طبقه‌بندی نیاز داریم. اگر درس ویژگی و بُعد چیست را خوانده باشید، متوجه می‌شوید که هر کدام از این ستون‌ها یک ویژگی یا یک بُعد است. حالا به این نکته می‌رسیم که اگر در مثال بالا هر تصویر ۱۰۰۰ در ۱۰۰۰ پیکسل باشد یعنی مجموعاً ۱ میلیون پیکسل در هر تصویر موجود است. این یعنی که برای ساخت ماتریسِ ویژگی (مانند ماتریسِ ساخته‌شده در شکل بالا) نیاز به ۱ میلیون ستون (بُعد) داریم. به این دست از مسائل داده‌هایی با ابعاد بالا گفته می‌شود که طبیعتاً نمی‌توان آن‌ها را بر روی محورهای دوبُعدی یا سه‌بُعدی رسم کرد.



این مثال بالا، یک نمونه از داده‌هایی با ابعاد زیاد یا همان high dimensional data setبود که در مباحثِ مختلفِ داده‌کاوی و یادگیریِ ماشین برخوردهای متفاوتی با این دست از داده‌ها انجام می‌دهند.

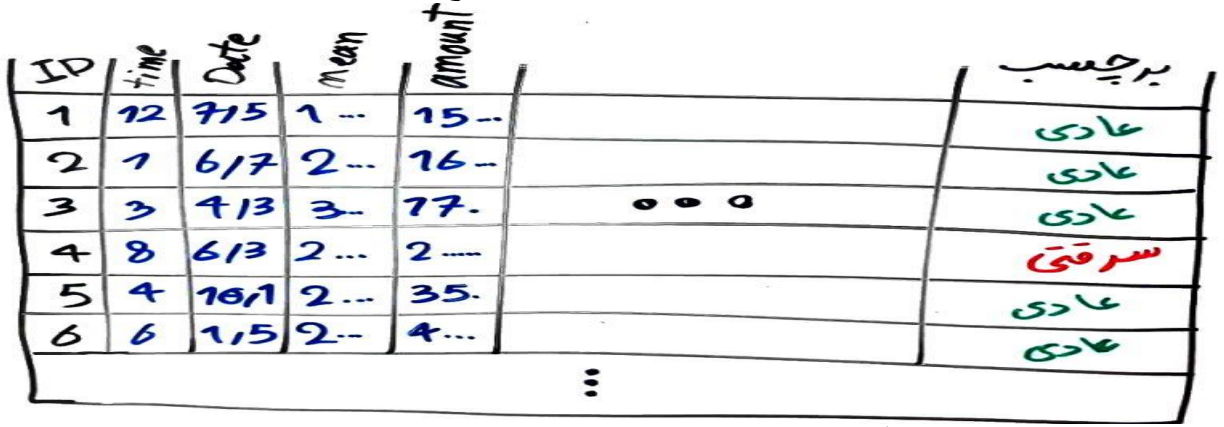
سید ایمان غفوریان - عضو هیات علمی دانشگاه آزاد مشهد-زمستان ۱۴۰۲

مجموعه داده‌هایی با ابعاد زیاد High Dimensional

اگر درس طبقه‌بندی Classificationرا خوانده باشید، می‌دانید که منظور ما از طبقه یا کلاس یا همان برچسب چیست. برای مثال در همان درسِ طبقه‌بندی دیدیم که مدیرِ یک بانک می‌خواست از روی ویژگی‌های مختلفِ مشتری‌ها، تصمیم بگیرد که به آن‌ها وام بدهد/یا خیر. پس مجموعه‌ی داده‌ای از مشتری‌های قبلی آماده می‌کرد و ویژگی‌های آن‌ها به همراه بازپرداخت وام را برای هر یک به دست می‌آورد و در مجموعه‌ی داده‌ی آموزشی قرار می‌داد. فرض کنید، از بین ۱۰هزار مشتری بانک، ۵هزار نفر آن‌ها توانسته باشند وام را پس دهند و ۵هزار نفر نتواسته باشند وام خود را پس دهند. پس در این‌جا یک مجموعه‌ی داده‌ی متوازن داریم به صورتی که هر کدام از طبقه‌ها به صورت تقریبی یک اندازه داده دارند و الگوریتمِ طبقه‌بندی می‌تواند الگوهای هر دسته را پیدا کرده و یادگیری خود را از روی این مجموعه‌ی داده انجام دهد. اما همه‌ی مجموعه‌ی داده‌ها به این صورت متوازن نیستند.

فرض کنید مجموعه‌ی داده‌ای از تراکنش‌های بانکی دارید. برخی از این تراکنش‌ها، تراکنش‌های سرقتی هستند. برای مثال کارت بانکی به همراه رمز آن، دزدیده شده است و شخصِ سارق، می‌خواهد از آن حساب برداشت کند. حال به الگوریتمی نیاز داریم تا بتواند هر تراکنش را بررسی کرده و با توجه به داده‌های گذشته، بفهمد که این تراکنش جدید، آیا تراکنش سالمی است یا تراکنش سرقتی.

برای انجام این پروژه بانک مرکزی به همراه نیروی انتظامی، در مدت یک سال تراکنش‌های مختلف را ارزیابی کرده‌اند و برخی از این تراکنش‌ها را به عنوان تراکنش سرقتی (با استفاده از اطلاعات انتظامی) برچسب گذاری کرده‌اند. برای هر تراکنش نیز، ویژگی‌هایی نیز در نظر گرفته شده است (برای مثال ساعت انجام تراکنش، تاریخ تراکنش، میانگین حساب شخص و مقدار تراکنش). مجموعه‌ی داده چیزی مانند شکل زیر است:



مجموعه داده‌هایی با ابعاد زیاد High Dimensional

همان‌طور که مشاهده می‌کنید درصدِ بسیار کمی از تراکنش‌ها (شاید کمتر از یک دهم درصد) تراکنش‌های سرقتی هستند و اکثر تراکنش‌ها را تراکنش‌های عادی تشکیل می‌دهند. ما مجموعه‌ی داده‌ی بالا را به یک الگوریتم طبقه‌بندی تزریق می‌کنیم تا یادگیری را انجام دهد.

در این مواقع نباید انتظار داشته باشیم که الگوریتم‌های طبقه‌بندی به درستی یاد بگیرد زیرا این الگوریتم‌ها عموماً به سمتِ طبقه‌هایی با برچسب اکثریت، تمایل پیدا می‌کنند. برای نمونه، در مثال بالا یک الگوریتم طبقه‌بندی عادی مانند SVM بعد از یادگیری از مجموعه‌ی داده‌ها، تمامیِ تراکنش‌های جدید را به عنوان تراکنش عادی طبقه‌بندی می‌کند!

در این شرایط معیار ارزیابی طبقه‌بندی نیز بایستی تغییر کند و نمی‌توان از معیارهای عادی مانند معیار دقت Accuracy استفاده کرد. چون برای مثال در همان مجموعه‌ی داده‌ی بالا، اگر یک الگوریتم تمامیِ نمونه‌های جدید را به عنوان تراکنش عادی تقسیم بندی کند، دقت تقریباً ۹۹/۹۹ درصد می‌شود ولی حتماً می‌دانید که در این دست از مسائل، نمی‌توانیم به این معیار دقت Accuracy اعتماد کنیم.

در این شرایط بایستی از روش‌های مختلف متعادل‌سازی داده‌ها Data Balancing و یا الگوریتم‌های خاصی استفاده کرد. البته این داده‌های نامتوازن فقط در دسته‌ی مسائل طبقه‌بندی نیستند و ممکن است در مسائل خوشه‌بندی نیز داده‌های نامتوازن داشته باشیم که راه‌حل‌های مربوط به خود را دارد.

فرآیند کریسپ CRISPجهت انجام پروژه‌های داده‌کاوی

پروژه‌های مختلف صنعتی، هر کدام روش‌ها و فرآیندهای خاص خود را دارند. برای مثال در فرآیند مهندسی ساخت و تولید یک نرم افزار، می‌توان از روش‌های گوناگونی مانند روش آبشاری، روش حلقوی یا روش چابک استفاده کرد. برای اجرای فرآیندهای داده‌کاوی نیز، روش‌های مختلفی تولید شده است که یکی از محبوب‌ترین آن‌ها روش «فرآیند استاندارد صنعتی متقاطع» است که مخفف شده و لاتینِ آن به CRISPمعروف است.

روش کریسپ CRISP برای اجرای پروژه‌های داده‌کاوی در صنعت به کار گرفته می‌شود و دارای مراحل زیر است:

۱. فهم کسب و کار Business Understanding

در این مرحله، یک متخصص علم داده بایستی کسب و کاری که می‌خواهد بر روی آن پروژه داده‌کاوی انجام دهد را به خوبی بشناسد. در مرحله‌ی فهم کسب و کار، بایستی زوایای مختلف آن کسب و کار، محدودیت‌ها، شرایط موجود و اهداف آن کسب و کار از پروژه یا پروژه‌های جاری را بررسی نمود. این مرحله ذهن متخصص علم داده را برای کار بر روی پروژه آماده می‌کند و به او اجازه می‌دهد تا با شناخت بیشتر و بهتر به سراغ مراحل بعدی برود. در این مرحله، یک متخصص علم داده می‌تواند تا حدودی به کسب و کارِ موجود مسلط شده و فهم خود را از آن کسب و کار تا حد ممکن بالا ببرد.

۲. فهم داده‌ها Data Understanding

در مرحله‌ی فهم داده‌ها، متخصص علم داده، به سراغ داده‌های موجود کسب و کار رفته آن را برای شروع پروژه بررسی می‌کند. در این مرحله، عملیاتی مانند «آنالیز اکتشافی داده‌ها EDA و ساخت گزارش‌های اولیه از داده‌ها می‌تواند بسیار کمک کننده باشد. با فهم داده‌ها و درک ابعاد و ویژگی‌های مختلف آن، می‌توان ایده‌های مختلف را مطرح کرد و ساختار اصلی پروژه را تعیین نمود. در این مرحله می‌توان کیفیت داده‌ها را نیز ارزیابی کرد و در صورت نامناسب بودن داده‌ها، با مشورت و مشارکت قسمت‌های مختلف کسب و کار، این داده‌ها را بهبود بخشید.

فرآیند کریسپ CRISPجهت انجام پروژه‌های داده‌کاوی

۳. آماده‌سازی داده‌ها Data Preparation بعد از فهم کسب و کار و فهم داده‌ها، حال می‌توان داده‌ها را آماده‌ی تحلیل و مدل‌سازی کرد. اگر دوره‌ی پیش پردازش داده‌ها را در چیستیو مطالعه کرده باشید، احتمالاً به سادگی می‌توانید این مرحله را درک کنید. در این مرحله، داده‌های کثیف، تمیز می‌شوند و داده‌ها به صورت ساختاری، برای مرحله‌ی بعدی آماده‌سازی می‌شوند. در این مرحله همچنین می‌توان مجموعه داده‌های مختلف را با یکدیگر ترکیب کرد تا به مجموعه داده‌ی بهتر و با کیفیت‌تری رسید.

۴. مُدل‌سازی Modeling

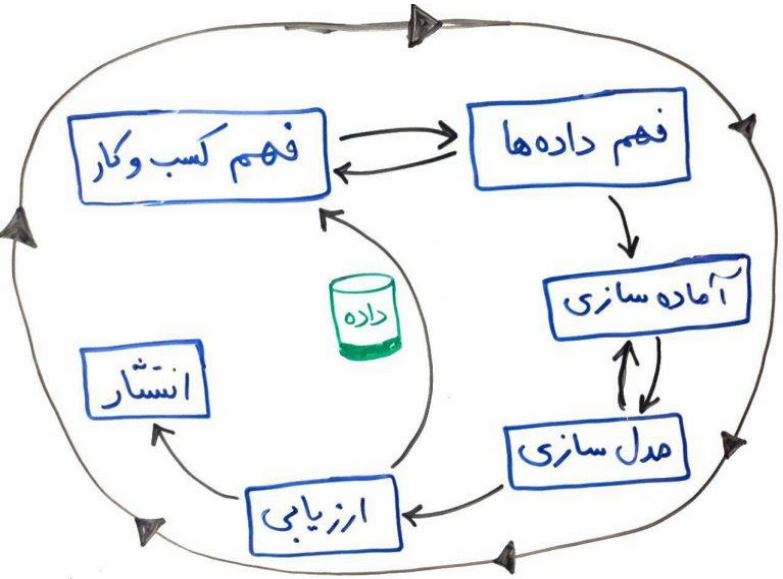
بسته به اینکه مسئله‌ی شما چه نوع مسئله‌ایست در این مرحله بایستی از الگوریتم‌ها و روش‌های مخصوص به خود استفاده کنید. مثلاً اگر مسئله‌ی شما طبقه‌بندی داده‌هاست، بایستی از الگوریتم‌های طبقه‌بندی برای یادگیری استفاده کنید و یا اگر مسئله‌ی شما در دسته‌ی خوشه‌بندی قرار می‌گیرد، می‌توانید یکی از الگوریتم‌های خوشه‌بندی را برای پروژه‌ی خود مورد استفاده قرار دهید. البته در یک پروژه‌ی داده‌کاوی، ممکن است مسائل مختلف و ترکیبی وجود داشته باشد که نیاز به عملیات پیچیده‌تری جهت مدل سازی دارند.

۵. ارزیابی Evaluation

چیزی که قابل ارزیابی نباشد، بهبود پیدا نمی‌کند. اگر در مراحل قبلی داده‌ها را آماده کردید و مدلی ساختید، بایستی بتوانید مدلِ خود را ارزیابی کنید. این ارزیابی بستگی به مدلِ انتخابی دارد. برای مثال اگر مسئله‌ی شما طبقه‌بندی بود، می‌توانید از روش‌های ارزیابی الگوریتم‌های طبقه‌بندی استفاده کنید. طبیعتاً اگر مدلِ شما به اندازه‌ی کافی کیفیت نداشت، بهتر است به مراحل قبلی بازگرید و مدل یا داده‌ها یا روش‌های آماده‌سازی داده‌هایتان را بهبود بخشیده و مجدداً ارزیابی را انجام دهید.

۶. پیاده‌سازی و انتشار Deployدر نهایت، بایستی نرم افزاری توسعه دهید تا کاربران بتوانند از زحمات شما استفاده کنند. این مرحله، معمولاً با کمک مهندسین نرم افزار و برنامه نویسان انجام می‌شود.

فرآیند کریسپ CRISPجهت انجام پروژه‌های داده‌کاوی



دیاگرام کلی روش کریسپ ((CRISPبه صورت زیر است:

همان‌طور که مشاهده می‌کنید این فرآیند به صورت چرخشی و تکراری انجام می‌شود. برای مثال اگر در مرحله‌ی «فهم داده» دچار مشکلی شدید به عقب برمی‌گردید و «کسب و کار را واکاوی» می‌کنید و یا اگر «مدل‌سازی» خوبی انجام ندادید، به عقب برگشته و در مرحله‌ی «آماده‌سازی داده‌ها» تجدید نظر می‌کنید. و یا اگر مدل شما بعد از «ارزیابی»، کیفیت مناسبی نداشت می‌توانید به مرحله‌ی اول برگشته و دوباره از «فهم کسب و کار» شروع کنید.

در ادامه مثالی کوتاه در حوزه‌ی بانکداری مبتنی بر روش کریسپ CRISP آورده می شود. فرض کنید می‌خواهید «فرآیند صلاحیت وام دادن» به متقاضیان وام را در یک بانک، به صورت هوشمند انجام دهید. این مثال را در درس طبقه‌بندی داده‌ها نیز ارایه شده، در مرحله‌ی اول به سراغ شناخت حوزه‌ی بانک، فرآیندهای موجود در بانک جهت اخذ وام و ویژگی‌های مختلف وام گیرنده و محل اعتبار وام می‌روید. بعد از آن به سراغ داده‌های موجود رفته و داده‌ها را بررسی می‌کنید. برای مثال ممکن است متوجه شوید که در داده‌ها کم و کاستی وجود دارد. مثلاً برخی از متقاضیان وام سن خود را به درستی وارد نکرده‌اند. ولی تصویر شناسنامه‌ی آن‌ها موجود است. می‌توانید از تصویر شناسنامه‌ی آن‌ها سن را استخراج کنید (که خود یک پروژه‌ی دیگر است) و یا کلاً ویژگی سن را از میان داده‌ها حذف کنید. بعد از آن به سراغ پیش پردازش بر روی داده‌ها می‌روید و داده‌ها را تمیز می‌کنید. سپس مدل طبقه‌بندی را بر روی این داده‌ها می‌سازید تا الگوریتم یاد بگیرد که به چه اشخاصی وام بدهد و به چه اشخاصی وام ندهد (با توجه به ویژگی‌ها یا ابعاد آن‌ها). بعد از آن مدل را با استفاده از معیارهای مختلف ارزیابی می‌کنید و در صورتی که کیفیت ارزیابی خوب بود، با کمک برنامه نویسان نرم افزاری منتشر می‌کنید که می‌تواند ورودی را از یک منبع داده بخواند، و بر اساس آن تشخیص دهد که آیا شخصِ جدیدِ متقاضی وام، صلاحیت دریافت وام را دارد یا خیر. در کل این فرآیند را نیز می‌توانید همواره بهبود بخشید و دوباره از اول بازنگری را بر روی مراحل مختلف انجام داده و نسخه‌های جدیدتر نرم افزار را منتشر کنید.

رانش یا گذار در داده‌ها Data Drift

. جهان در حال تغییر است و داده‌ها نیز در همین جهان زندگی می‌کنند، پس داده‌ها ممکن است پس از گذشت مدت زمانی تغییر کنند. این تغییرات در داده‌ها منجر به تغییر در الگوها نیز شده و احتمالاً مدل‌هایی که قبلاً بر روی داده‌ها یادگیری یا داده‌کاوی را انجام می‌دادند، با گذشت زمان، دقتی به مراتب پایین‌تر خواهند داشت.

رانش یا گذار در داده‌ها data drift ، به تغییرات ذاتی در داده‌ها با گذشت زمان گفته می‌شود. برای مثال فرض کنید یک الگوریتمِ داده‌کاوی یا یادگیری ماشین دارید که می‌تواند از روی متن پیامک، تشخیص دهد که این پیام هرز spam هست یا خیر؟ برای انجام این کار، مجموعه‌ای از پیامک‌ها (مثلاً ۱۰هزار پیامک) را انتخاب کرده و توسط یک متخصص هرزشناسی spam detector هر کدام از این پیامک‌ها را برچسب می‌زنیم. چیزی شبیه به جدول زیر:

مجموعه‌ی بالا یک مجموعه داده‌های آموزشی (training dataset) است که الگوریتم‌های داده‌کاوی و یادگیری ماشین از روی این داده‌ها یادگیری را انجام داده و سپس می‌توانند پیامک‌های جدید را به صورت خودکار برچسب‌زنی کنند.

SMS Dataset

	Text	Label
#1	سلام؟ خوبی؟ امشب می‌تونی مهمونی بیای؟	normal
#2	پوشاک نیک‌پوش، ۸۰درصد تخفیف ویژه تا آخر مهرماه تهران، میدان ونک، جنب بانک ملی	spam
#3	به این کارت بزن ۶۰۳۷۷۸۷۸۷۸۷۸۷۸	normal
#4	باشه، پس تا عصری خبرت می‌کنم	normal
#5	آخرین مهلت سپرده‌گذاری در موسسه کیان فردا هجده درصد سود تضمینی تماس با شماره‌ی: ۰۲۱۳۴۵۶۷۸	spam
...	...	...

فرض کنید الگوریتم بر روی این داده‌ها یادگیری را انجام داده و شما یک نرم‌افزار تشخیص پیامک هرز، با استفاده از همین الگوریتم ایجاد می‌کردید. بعد از گذشت چند ماه/یا چند سال، متن پیامک‌ها توسط شرکت‌های تبلیغاتی که پیامک ارسال می‌کنند، تغییر می‌کرد و یا ادبیات جدید وارد حوزه‌ی پیامکی بین مردم می‌شد. این تغییر در محتوا و متن پیامک‌ها باعث می‌شود که الگوریتمی که بر روی داده‌های قبلی (مجموعه‌ی آموزشی بالا) یادگیری را انجام داده، دیگر نتواند بر روی داده‌های جدید با همان دقت عمل کند. در واقع الگوریتم نتوانسته است تغییر یا همان رانش یا گذار در داده‌ها را تحمل کرده و همین امر باعث کاهش دقت الگوریتم بر روی داده‌ها (معمولاً بعد از گذشت مدت زمانی) می‌شود. این همان مفهوم رانش داده data drift است.

رانش یا گذار در داده‌ها Data Drift

برای مقابله با رانش داده‌ها روش‌های مختلف و متعددی موجود است. برای مثال می‌توان الگوریتم را با استفاده از روش‌های بر خط online بر روی داده‌ها برازش کنیم. در مورد یادگیری برخط online learning در درس مربوطه صحبت کرده‌ایم. همچنین با استفاده از راهکارهای آماری و مقایسه‌ی توزیع داده‌ها با روش‌هایی مانند KL Divergenceیا جنسون شنون Jenson Shannon نمونه‌ای از داده‌های جدید را به صورت دوره‌ای با داده‌های قبلی (که الگوریتم بر روی آن‌ها یادگرفته شده است)، مقایسه کرده تا بفهمیم که آیا داده‌ها دچار رانش شده است یا خیر. با این کار می‌توانیم به صورت دوره‌ای یک مجموعه‌ی داده‌ی آموزشی (مانند شکل زیر) ایجاد کرده و الگوریتم را هر چند وقت یک بار بر روی داده‌های جدید برازش کنیم.

SMS Dataset		
	Text	Label
#1	سلام؟ خوبی؟ امشب می‌تونی مهمونی بیای؟	normal
#2	پوشاک نیک‌پوش، ۸۰درصد تخفیف ویژه تا آخر مهرماه تهران، میدان ونک، جنب بانک ملی	spam
#3	به این کارت بزن ۶۰۳۷۷۸۷۸۷۸۷۸۷۸۷۸	normal
#4	باشه، پس تا عصری خبرت می‌کنم	normal
#5	آخرین مهلت سپرده‌گذاری در موسسه کیان فردا هجده درصد سود تضمینی تماس با شماره‌ی: ۰۲۱۳۴۵۶۷۸	spam
...	...	...