جزوهٔ دوره

جامپ یادگیری ماشین Quera College

تهیه شده توسط علی فاضل نیا

دانشجوی علوم کامپیوتر

راه‌های ارتباط:

Email: alifazelniya.1384@gmail.com
Telegram: @Norbert_desu
GitHub: github.com/AliFazelniya

اسفند ماه ۱۴۰۴

فهرست مطالب

پیشگفتار

اینجا هدف، دامنه، و نحوه‌ی استفاده از جزوه را بنویس. (می‌توانی همین متن را بعداً دقیقاً با متن خودت جایگزین کنی.)

فصل ۱

مقدمه

۱.۱ سلام!

سلام؛ ورود شما را به کالج «یادگیری ماشین ۲ | جامپ تکنیکال» خوش‌آمد می‌گوییم. از اینکه در کوئرا کالج افتخار میزبانی شما را داریم، به خود می‌بالیم

«یادگیری ماشین ۲ | جامپ تکنیکال» کالج سوم از مسیر علم داده و یادگیری ماشین کوئراست که پس از «یادگیری ماشین ۰ | دروازه ورود» و «یادگیری ماشین ۱ | تحلیل داده با پایتون» طراحی شده است.

هدف ما از تدوین کالج این است که شما را به شکل اصولی و گام‌به‌گام با الگوریتم‌های یادگیری ماشین آشنا کنیم؛ به‌طوری که در نهایت بتوانید الگوریتم‌ها را تحلیل کنید، نقاط ضعف و قوت آن‌ها را بشناسید و تشخیص دهید چگونه از آن‌ها برای حل مسائل دنیای واقعی کمک بگیرید! علاوه بر این‌که با الگوریتم‌های یادگیری ماشین آشنا می‌شوید، تکنیک‌هایی برای استفاده عملی از آن‌ها را نیز خواهید آموخت؛ به عبارت بهتر، «یادگیری ماشین ۲ | جامپ تکنیکال» آمیخته‌ای موزون از دانش علمی و مهارت عملی یادگیری ماشین کلاسیک است!

این کالج مناسب افرادی است که در حد متوسط با پایتون و کتابخانه‌های Numpy و Pandas آشنایی داشته باشند. داشتن دانش حداقلی از آمار احتمال و حسابان (مباحث مشتق) و جبر خطی (ماتریس‌ها) به شما در فراگیری محتوای این کالج کمک می‌کند. به صورت کلی می‌توان گفت اگر با ریاضیات در حد مقطع دبیرستان آشنا باشید، به راحتی می‌توانید به ماجراجویی در این کالج بپردازید! اگر با پیشنیازهای کالج آشنا نیستید، می‌توانید با گذراندن سایر دوره‌های کوئرا کالج، مهارت‌های لازم را کسب کنید.

شما با داشتن پیش‌نیازهای کالج شروع می‌کنید و در انتها پس از مطالعه‌ی درسنامه‌ها و حل تمرین‌ها، به «دانشمند داده» و «مهندس یادگیری ماشین» تبدیل می‌شوید که قطعاً راه طولانی اما شیرینی برای حرفه‌ای شدن پیشِ رو دارید!

البته برای حفظ تناسب کالج و افزایش احتمال یادگیری عمقی، از آموزش کامل یادگیری عمیق، که زیرمجموعه یادگیری ماشین است، پرهیز شده است؛ بلکه شبکه‌های عصبی مصنوعی و مقدمات یادگیری عمیق را در فصل «شبکه عصبی» آورده‌ایم. برای فراگیری یادگیری عمیق، می‌توانید منتظر کالج‌های بعدی مسیر علم داده و یادگیری ماشین کوئرا کالج باشید

امیدواریم با ارائه‌ی آموزش باکیفیت، گامی مثبت در افزایش دانش و پیشرفت شما برداریم. سختی‌های

مسیر نه تنها ناامیدمان نمی‌کند بلکه توان‌مان را بیشتر و تصمیم‌مان را راسخ‌تر می‌کند. مسیر رسیدن به قله‌ای که آرزویش را داریم؛ جوانانی توانمند، پرتلاش و ایرانی پیشرفته...

۲.۱ قالب کالج

قالب کالج روند آموزشی کوئرا کالج

امروزه با فراگیر شدن آموزش‌های آنلاین، قالب‌های متعددی برای یادگیری مفاهیم علوم کامپیوتر بصورت آنلاین پیاده‌سازی شده است. آموزش آنلاین این مفاهیم، فرصت برنامه‌نویسی به همراه یادگیری را فراهم می‌کند و هم‌چنین انجام تمرین‌های واقعی و استفاده از کتابخانه‌ها و قالب‌های نزدیک به صنعت را تسهیل می‌کند. اما آموزش آنلاین چالش‌هایی نیز به همراه دارد؛ زیرا تعاملی که استاد در سر کلاس با دانشجو دارد و نظمی که جلسات کلاس به یادگیری دانشجو می‌دهد به سختی در قالب‌های آموزش آنلاین گنجانده می‌شود. در کوئرا کالج تلاش کردیم قالبی برای برطرف شدن نیاز تعاملی بودن، نظم، و هم‌چنین عملی بودن آموزش آماده کنیم.

در کل، آموزش این دوره متشکل از چندین فصل می‌باشد، که هر فصل شامل تعدادی درسنامه، تمرین یا آزمون است.

درسنامه‌ها

پس از مطالعه‌ی هر درسنامه می‌توانید تیک "خواندم" آن را بزنید و پیش بروید. در پایین هر درسنامه بخش کامنت‌ها تعبیه شده تا بتوانید پرسش‌ها و نظرات عمومی خود را درباره‌ی درسنامه‌ی مربوطه با ما و سایر

شرکت‌کنندگان دوره مطرح کنید.

تمرین‌ها

تمرین‌ها نیز همگی توسط سامانه‌ی داوری خودکار Quera تصحیح شده، و پس از ارسال کد، توسط سیستم، نمره‌دهی می‌شود. داوری تمرین ممکن است شامل چندین تست مختلف باشد و سعی شده هنگامیکه در کسب نمره‌ی کامل یک تست دچار مشکل می‌شوید بازخورد مناسبی توسط سیستم خروجی داده شود.

در تمرین‌هایی که عملکرد مدل یادگیری ماشین شما سنجیده می‌شود یک حد آستانه (Threshold) تعریف شده و در صورتی‌که مدل شما عملکردی بهتر از آن حد داشته باشد تمرین با موفقیت گذرانده می‌شود. البته اگر عملکرد مدل شما از مقدار خواسته‌شده بهتر باشد امتیاز اضافه‌تری کسب خواهید کرد.

آزمون‌ها

در بعضی از فصل‌ها به منظور درک عمیق‌تر مباحث آموخته شده تعدادی آزمون چندگزینه‌ای در نظر گرفته شده است. پاسخ شما به آزمون‌ها نیز مشابه با تمرین‌ها توسط سامانه‌ی داوری خودکار Quera انجام خواهد گرفت. هرچند که از نظر تعداد ارسال پاسخ محدودیتی نخواهید داشت، با این حال امتیاز آزمون‌ها در مقایسه با تمرین‌ها کمتر در نظر گرفته شده است.

پشتیبانی

ما گام به گام در طول مسیر این کالج همراه‌تان هستیم! در صورت وجود هرگونه پرسش یا ابهام درباره‌ی هر بخشی از این کالج می‌توانید از طریق پیغام خصوصی با پشتیبانان کالج در ارتباط باشید.

۳.۱ سیستم امتیازات و دریافت گواهی اصلی

سیستم امتیازات و دریافت گواهی اصلی

با حل هر تمرین یا آزمون، مقداری امتیاز به شما تعلق می‌گیرد. با این امتیاز می‌توانید راه‌حل تمرین‌های دوره را دریافت کنید. پس از پایان دوره، بر اساس امتیاز کسب‌شده توسط شما، گواهی صادر خواهد شد. پس سعی کنید از امتیازات خود به نحو مناسبی استفاده کنید! گذراندن فصل‌ها

دوره‌ی «یادگیری ماشین ۲ |جامپ تکنیکال» از چندین فصل تشکیل شده است. پیشنهاد ما این است که برنامه خود را مطابق با سرفصل دوره تنظیم کرده و مباحث را به ترتیب پیش ببرید. چیدمان محتوا، مورد تایید مهندسان و استادان برتر داخلی و خارجی است و پیش‌رفتن طبق آن، یادگیری شما را تضمین می‌کند! توضیحات دریافت گواهی

شما با حل هر یک از تمرین‌های دوره، مقداری امتیاز دریافت می‌کنید و می‌توانید با استفاده از این امتیازات، جواب تمرین‌ها را خریداری کنید. زمانی که دوره را به اتمام می‌رسانید، مقداری امتیاز برای شما باقی می‌ماند و برحسب این مقدار به شما گواهی داده می‌شود.

اگر جواب هیچ تمرینی را خریداری نکنید و همه تمرین‌ها را حل کنید، مجموع امتیاز شما به ۳۲۰۰ می‌رسد.

در نهایت بر حسب امتیاز نهایی، یکی از چهار سطح زیر در گواهی گزارش می‌شود:

سطح Perfect برای نمرات بالای ۲۲۰۰

سطح Good Very برای نمرات کم‌تر از ۲۲۰۰ و بالای ۱۸۰۰

سطح Good برای نمرات کم‌تر از ۱۸۰۰ و بالای ۱۴۰۰

سطح Fair برای نمرات کم‌تر از ۱۴۰۰

برای هر فصل، یک آستانه تعریف شده است که شما برای دریافت گواهی، حتما باید بیشتر از آستانه، فصل را مطالعه کرده باشید. به عنوان مثال اگر فصلی دارای آستانه ۸۰ درصدی باشد، برای آنکه بتوانید گواهی را دریافت کنید، حداقل ۸۰ درصد آن فصل را باید مطالعه کنید. حتی اگر امتیازتان بیشتر از ۱۶۰۰ باشد ولی فصلی باشد که کمتر از آستانه مطالعه شده باشد، گواهی صادر نمی‌شود!

البته پس از پایان دوره شما می‌توانید با حل سؤالات دیگر داخل فصل‌های دوره، امتیاز خود را افزایش داده و گواهی Perfect را دریافت کنید.

۴.۱ یادگیری ماشین چیست؟

یادگیری ماشین چیست؟

فرض کنید دهه پنجاه میلادی است و چیزی به نام یادگیری ماشین وجود ندارد. در آن دوره یکی از مهندسان IBM که نامش آقای آرتور ساموئل (Arthur Samuel) بود، برای اولین بار از عبارت یادگیری ماشین (Machine Learning) استفاده کرد و تعریف زیر را برای آن ارائه داد: «یادگیری ماشین زمینه‌ای از تحقیقات است که به کامپیوترها توانایی یادگیری بدون برنامه‌نویسی صریح را می‌دهد.»

در دیدگاه آرتور ساموئل، یادگیری ماشین با برنامه‌نویسی صریح تفاوت دارد. در برنامه‌نویسی ساده، ما باید الگوهای متفاوت را خودمان تشخیص داده و به‌صورت دستی برای آن‌ها برنامه‌نویسی کنیم. اما در یادگیری ماشین، ما صرفاً مدلی را طراحی می‌کنیم که قادر به یادگیری و پیدا کردن خودکار الگوها از روی مجموعه‌داده (Dataset) است. در مواردی که مسئله‌ی مورد نظر پیچیده شود، پیدا کردن این الگوها اغلب برای انسان دشوار یا حتی غیر ممکن است. اما برای ماشین‌ها به دلیل قدرت پردازشی بالا و توانایی استفاده از الگوریتم‌ها، این کار بسیار ساده‌تر است. همین موضوع، دلیل اصلی شهرت یادگیری ماشین است. الگوریتم‌های یادگیری ماشین، مثل انسان به کمک تجربه یاد می‌گیرند. داده (Data) همان تجربه‌ای است که به عنوان ورودی به الگوریتم داده می‌شود.

آقای تام میشل (Tom M. Mitchell) در کتاب یادگیری ماشین خود، یادگیری ماشین را از دید مهندسی به این شکل تعریف کرده‌است: «اگر کارایی برنامه در انجام تکلیف TT که با معیار عملکرد PP ارزیابی می‌شود، با تجربه‌ی EE افزایش یابد، می‌گوییم که برنامه یاد گرفته است از تجربه‌ی EE با توجه به تکلیف TT و معیار عملکرد PP استفاده کند.» تکلیف (Task)

تکلیف در واقع همان مسئله‌ای است که ما انتظار داریم بتوانیم با یادگیری ماشین حل کنیم. برای مثال بانکی را تصور کنید که می‌خواهد تصمیم بگیرد آیا به یک مشتری وام اختصاص بدهد یا خیر. انتخاب وام دادن یا ندادن به مشتری را تکلیف TT می‌گوییم. تجربه (Experience)

برای انجام فرآیند یادگیری که منجر به حل تکلیف TT می‌شود، نیازمند تعدادی نمونه (Sample) هستیم که اطلاعات مورد نیاز در مورد مسئله را به ما می‌دهند. برای مثال در مسئله‌ی وام دادن بانک، می‌توان از سابقه‌ی مشتریان پیشین و این که وام خود را پرداخت کرده‌اند یا خیر برای مجموعه‌داده یا نمونه‌ها استفاده

کرد. معیار عملکرد (Performance)

هر مدل یادگیری ماشینی که طراحی کنیم، همواره به‌طور قطعی و ۱۰۰ درصدی نتیجه‌ی درست و مناسبی را ارائه نمی‌دهد؛ بنابراین به معیاری برای بررسی و اندازه‌گیری عملکرد آن نیاز داریم تا در صورت عملکرد نامناسب بتوانیم با تغییر پارامترها به مدل بهتری دست یابیم. به این معیار، معیار عملکرد PP می‌گوییم. انواع یادگیری ماشین

یادگیری ماشین را می‌توان به‌طور کلی به سه دسته تقسیم کرد. البته بعضی از منابع تقسیم‌بندی‌های دیگری را نیز برای یادگیری ماشین تصور کرده‌اند، اما اکثر آن‌ها همین سه قسمت کلی را به عنوان زیرشاخه‌های یادگیری ماشین معرفی می‌کنند. در ادامه به تعریف هر کدام از این دسته‌ها می‌پردازیم. ۱. یادگیری نظارت‌شده (Supervised Learning)

فرض کنید که کامپیوتر یک بچه است و ما ناظر (supervisor،) به طور مثال پدر یا مادر او هستیم. ما می‌خواهیم به این کودک یاد بدهیم که یک خروس چه شکلی‌ست. برای این کار، ما تعدادی عکس که بعضی از آن‌ها عکس خروس و بعضی حیوانات دیگری هستند را به بچه نشان می‌دهیم. وقتی که عکس خروس را نشان می‌دهیم، جمله‌ی «این خروس است» را گفته و وقتی عکس‌هایی که خروس نیستند را نشان می‌دهیم، جمله‌ی «این خروس نیست» را می‌گوییم. به این ترتیب، بچه‌ی ما یاد خواهد گرفت که عکس‌های خروس را از غیر خروس تشخیص دهد. به این روش یادگیری، یادگیری نظارت‌شده (Supervised Learning) می‌گوییم.

در این نوع از یادگیری، نمونه‌هایی که برای آموزش مدل استفاده می‌شوند، دارای برچسب (Label) هستند. به این معنی که مدل یادگیری ماشین با استفاده از داده‌هایی که از قبل برچسب مشخصی دارند («خروس بودن» و «خروس نبودن» در این مسئله)، الگوهای اساسی را تا زمانی که به عملکرد رضایت‌بخشی برای ما برسند، پیدا می‌کند.

به‌عنوان مثال، جدول زیر را در نظر داشته باشید. این جدول اطلاعاتی از خانه‌های شهر پکن به ما می‌دهد.

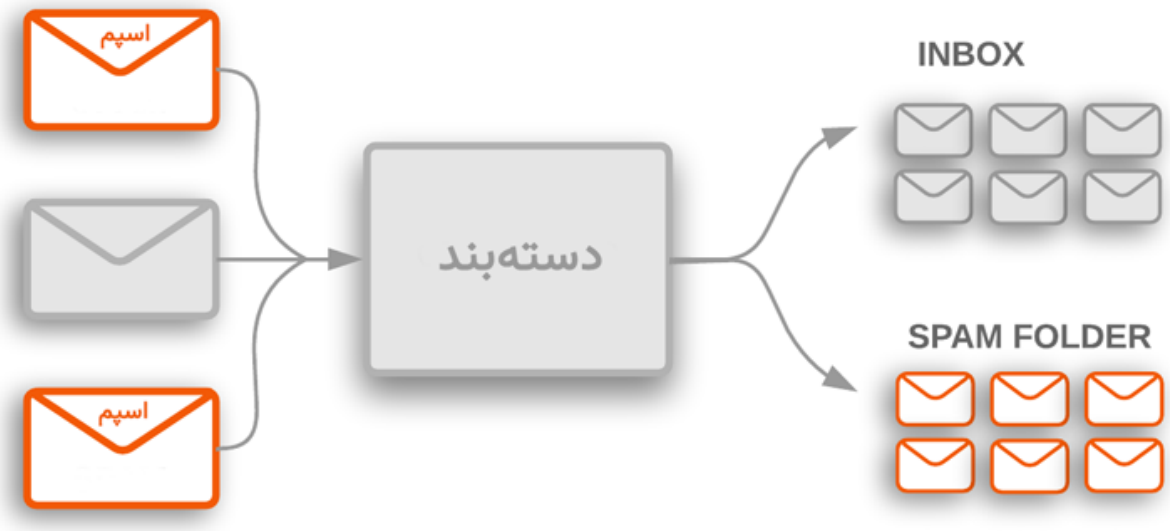
Lng	Lat	constructionTime	elevator	Price
116.69	39.8706	2010	0	82762
116.406	39.9577	2005	1	82762
116.471	39.9011	2005	1	42473
116.337	39.8941	1985	0	41586
116.359	39.9321	1992	1	87161
116.353	40.0502	2012	1	69539
116.296	39.8484	1996	0	19176
116.336	39.9322	1980	0	80059
116.48	39.9115	1985	0	42761
116.576	39.8620	2001	0	21109

هر سطر این جدول، مربوط به یک خانه است و ستون‌های آن، اطلاعاتی از هر خانه را نشان می‌دهند. اگر بخواهیم به کمک یادگیری ماشین، قیمت هر متر مربع خانه‌ها را پیش‌بینی کنیم، ستون Price همان برچسبی است که مدل سعی می‌کند به کمک سایر ستون‌ها، آن را پیش‌بینی کند.

الگوریتم‌های یادگیری نظارت‌شده را می‌توان به دو بخش دسته‌بندی (Classification) و رگرسیون (Regres-sion) تقسیم کرد که در ادامه به معرفی بیشتر هرکدام می‌پردازیم. دسته‌بندی (Classification)

در دسته‌بندی، هدف ما پیدا کردن دسته (Class) یا برچسب مناسب برای نمونه‌های بدون برچسب است.

برای این کار، ما مدل یادگیری ماشینی خود را با استفاده از نمونه‌های برچسب‌دار، آموزش می‌دهیم. بر اساس این آموزش، مدل ما یاد می‌گیرد که داده‌ها را به دسته‌های مختلف تقسیم کند. به‌عنوان مثال، دسته‌بندی ایمیل‌ها به دو دسته‌ی اسپم (Spam) و غیر اسپم را در نظر بگیرید. برای این کار، شما مجموعه‌داده‌ای شامل میلیون‌ها متن ایمیل، موضوع ایمیل و دیگر ویژگی‌هایی که ممکن است مهم باشند را جمع‌آوری می‌کنید. سپس، بر اساس اینکه هر ایمیل اسپم بوده است یا خیر، آن‌ها را برچسب می‌زنید. اکنون، با استفاده از یکی از الگوریتم‌های دسته‌بندی، مدلی را روی نمونه‌های برچسب‌دار، آموزش می‌دهید. مدل شما در نهایت می‌تواند یک ایمیل اسپم را از غیر اسپم تشخیص دهد.

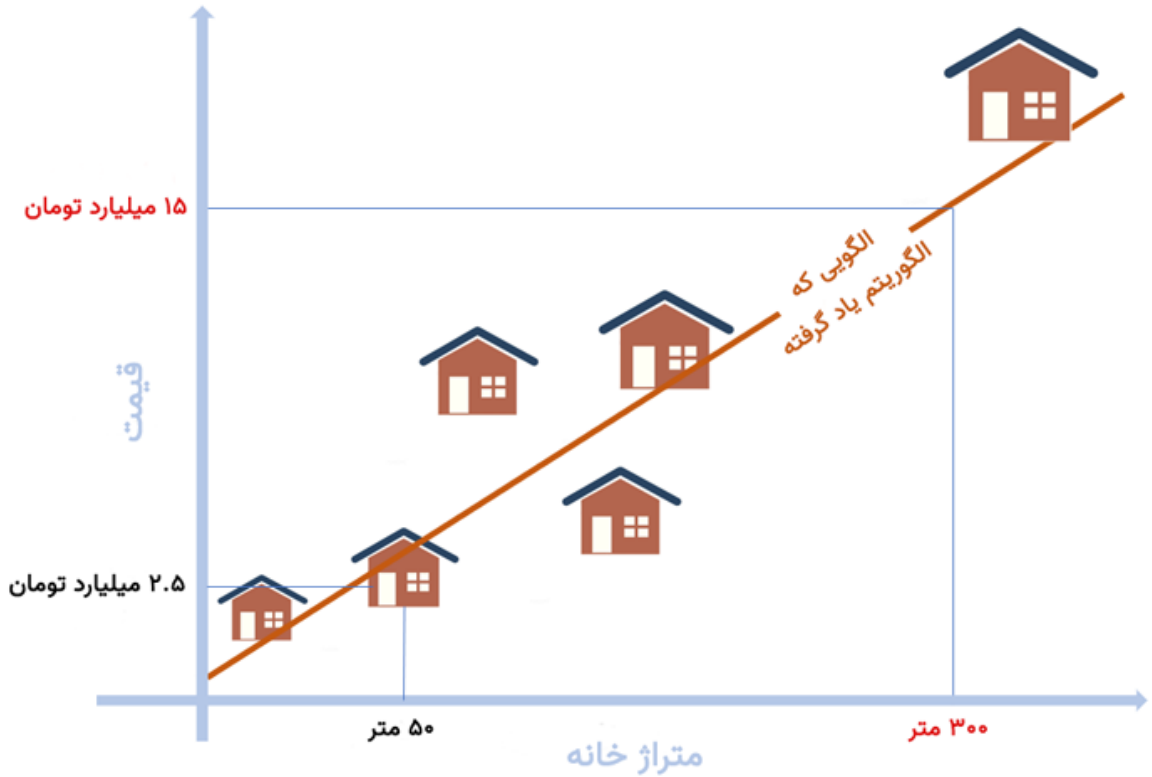


رگرسیون (Regression)

در رگرسیون، هدف ما تخمین مقدار یک ویژگی (این بار مقداری عددی/پیوسته) برای یک نمونه است. این الگوریتم‌ها برای پیش‌بینی روند بازار، قیمت خانه و دیگر برچسب‌های عددی به کار می‌روند.

به‌طور مثال، برای پیش‌بینی قیمت خانه، می‌توان از اطلاعات خانه‌های دیگر برای تخمین قیمت یک خانه استفاده کرد. ویژگی‌هایی مانند متراژ، تعداد اتاق، داشتن یا نداشتن پارکینگ، داشتن یا نداشتن حیاط و دیگر ویژگی‌های تاثیرگذار بر قیمت یک خانه، می‌توانند به عنوان اطلاعات ورودی در نظر گرفته شوند.

در مدل زیر، فقط از متراژ خانه‌ها برای ساخت مدل یادگیری ماشین استفاده شده است. هر نقطه یک خانه را نشان می‌دهد. برای مثال، خانه‌ی ۵۰ متری، ۵.۲ میلیارد تومان ارزش دارد. از نظر مدل ما، خانه‌ی ۳۰۰ متری که قیمت آن مشخص نیست، ۱۵ میلیارد تومان ارزش دارد.



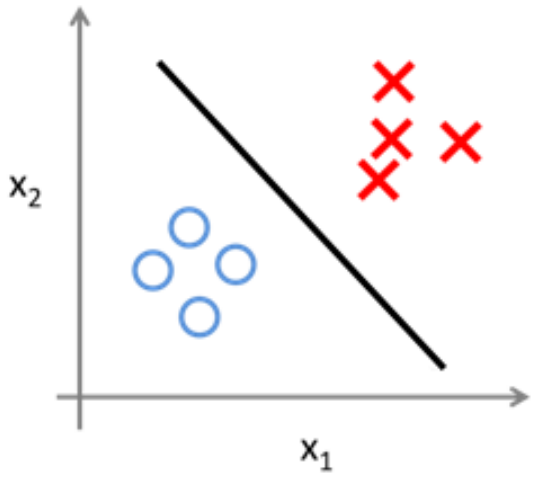
تفاوت دسته‌بندی و رگرسیون

همان‌طور که دیدید در هر دو رویکرد مدل یاد می‌گیرد تا برچسب داده‌ها را پیش‌بینی کند. تفاوت میان این دو، در نوع این برچسب است. در الگوریتم‌های دسته‌بندی برچسبی که می‌خواهیم پیش‌بینی کنیم به صورت متغیری دسته‌ای/گسسته است اما در الگوریتم‌های رگرسیون متغیر هدف از نوع عددی/پیوسته است.

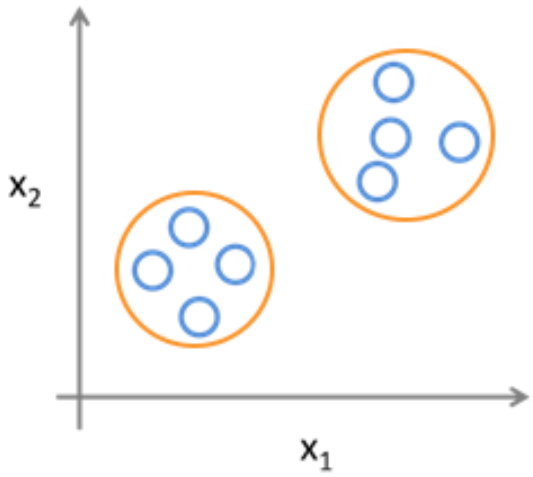
۲. یادگیری نظارت‌نشده (Unsupervised Learning)

تفاوت یادگیری نظارت‌نشده (Unsupervised Learning) با یادگیری نظارت‌شده، در نبودن برچسب‌ها است. به عبارت دیگر، هیچ ناظری (برچسب) به کامپیوتر نمی‌گوید که چه زمانی درست پیش‌بینی کرده و چه زمانی مرتکب خطا شده است. در این رویکرد یادگیری، مدل به‌تنهایی و بدون کمک برچسب‌هایی که در روش نظارت‌شده دیدیم، باید الگوها را شناسایی کند.

برای مثال در تصویر پایین سمت راست، نقاط هیچ برچسبی ندارند (از لحاظ ظاهری تفاوتی ندارند) اما فاصله‌ی نقاط از هم است که آن‌ها را متمایز می‌کند و در دسته‌های مجزا قرار می‌دهد. از یادگیری نظارت‌نشده برای تحلیل اکتشافی و خوشه‌بندی مجموعه‌داده استفاده می‌شود. توجه داشته باشید که بیشتر مجموعه‌داده‌های موجود، بدون برچسب هستند و به این دلیل، این روش‌ها بسیار کاربردی هستند.



یادگیری نظارت‌شده



یادگیری نظارت‌نشده

در تصویر بالا سمت چپ دو نوع داده وجود دارد، داده‌هایی با علامت ضربدر قرمز و داده‌هایی با دایره‌های آبی. دایره‌ی آبی و ضربدر قرمز همان برچسب‌ها هستند. این تصویر در واقع همان مسئله‌ی دسته‌بندی (نوعی از یادگیری نظارت‌شده) را نشان می‌دهد و وظیفه‌ی مدل پیدا کردن خط سیاه رنگ است که به کمک آن بتوان دسته‌ی «دایره‌های آبی» را از دسته‌ی «ضربدرهای قرمز» تفکیک کرد. هر نقطه‌ای که سمت راست خط مشکی باشد، متعلق به دسته‌ی «ضربدرهای قرمز» است. هر نقطه‌ای هم که سمت چپ خط قرار بگیرد، متعلق به دسته‌ی «دایره‌های آبی» خواهد بود.

اما در تصویر سمت راست هیچ گونه برچسبی وجود ندارد. در این حالت ماشین تشخیص داده است که مجموعه‌داده‌ی ما، قابل تقسیم به دو دسته است: یکی پایین سمت چپ و دیگری بالا سمت راست صفحه.

الگوریتم‌های یادگیری ماشین نظارت‌نشده، که در آن مدل بدون دخالت انسان و بدون برچسب، الگوهای پنهان را پیدا می‌کند، به سه دسته‌ی زیر تقسیم می‌شود:

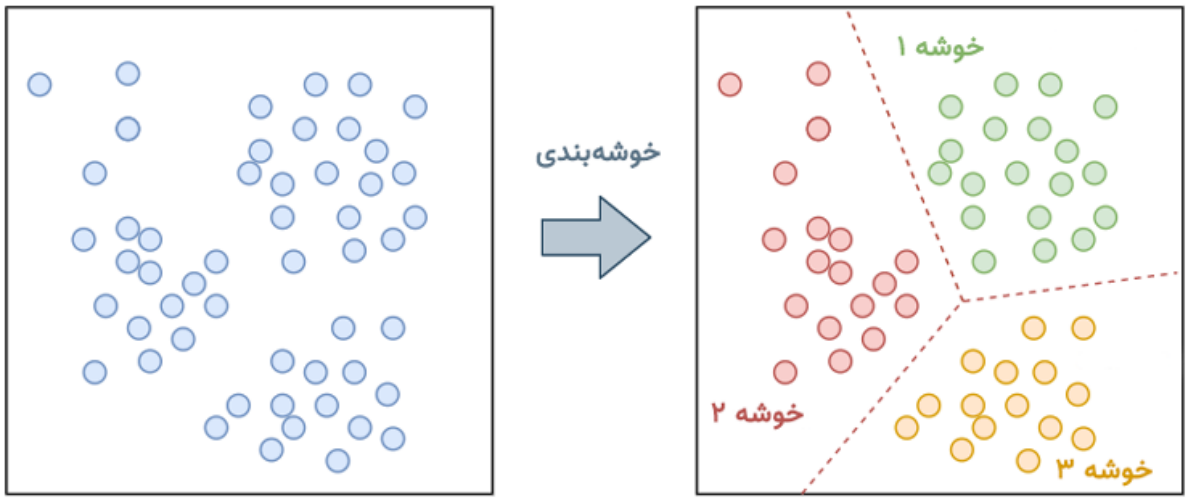
خوشه‌بندی (Clustering)

کاهش ابعاد (Dimensionality Reduction)

استخراج قانون وابستگی (Association Rule Mining)

در این درسنامه، به معرفی «خوشه‌بندی» و «کاهش ابعاد» می‌پردازیم و از توضیح «استخراج قانون وابستگی» به دلیل آن که خارج از حوزه‌ی این دوره است، صرف‌نظر می‌کنیم. خوشه‌بندی (Clustering)

خوشه‌بندی به معنی دسته‌بندی خودکار مجموعه داده، به خوشه‌های همگن است؛ به‌گونه‌ای که نمونه‌های هر خوشه، ویژگی‌های یکسانی داشته باشند. اولین گام، انتخاب معیاری برای سنجش فاصله بین نمونه‌ها مانند فاصله اقلیدسی است که یکی از پرکاربردترین معیارهای سنجش فاصله است. به عنوان مثال، عکس پایین نمونه‌ای از خوشه‌بندی است که نمونه‌ها را بر اساس معیار فاصله اقلیدسی به سه دسته گروه‌بندی کرده است. اما لازم است بدانیم که معیارهای مناسب برای فاصله، تنها به فاصله اقلیدسی محدود نمی‌شوند.



کاهش ابعاد (Dimensionality Reduction)

در ساده‌ترین حالت، روش کاهش ابعاد یعنی کاهش دادن تعداد ویژگی‌هایی که از آن‌ها برای آموزش مدل یادگیری ماشین خود استفاده می‌کنیم. به طور مثال، کاهش دادن تعداد ستون‌های یک مجموعه‌داده‌ی جدولی، حالتی از کاهش ابعاد است.

سوالی که مطرح می‌شود این است که «چه نیازی به این کار داریم؟ چرا لازم است تعداد ستون‌های یک مجموعه داده جدولی که مثلاً ۸۰ ستون دارد را کاهش دهیم؟ چرا به سادگی از تمام این ۸۰ ویژگی برای آموزش مدل خود استفاده نکنیم؟»

در درسنامه‌های فصل یادگیری نظارت‌نشده با معرفی تکنیک‌های مختلف استخراج ویژگی (Feature Extrac-tion) که زیرشاخه‌ای از کاهش ابعاد شناخته می‌شود پاسخ این پرسش‌ها را بررسی خواهیم کرد. ۳. یادگیری تقویتی (Reinforcement Learning)

فرض کنید در حال انجام یک بازی معمایی به‌طور مثال بازی هزارتو هستید. هدف شما خارج شدن از هزارتو است و هر بار که قدمی در مسیر خارج شدن از هزارتو بردارید، پاداشی دریافت می‌کنید. همچنین زمانی که در مسیری گام بردارید که شما را به خارج از هزارتو هدایت نکند، از امتیاز شما کم می‌شود (مجازات می‌شوید). در این بازی ممکن است تا زمانی که بتوانید از هزارتو خارج شوید به دفعات به بن‌بست برسید. زمانی‌که قدم‌های درستی بردارید، با گرفتن امتیاز متوجه خواهید شد که در مسیر درست قرار دارید و با سعی در ادامه‌ی این مسیر می‌توانید از هزارتو خارج شوید.

روندی که در مثال بالا در پیش گرفتید در واقع همان رویکرد یادگیری تقویتی (Reinforcement Learning) است. یادگیری تقویتی با ذهنیت آزمون و خطا کار می‌کند. عامل هوشمند (Agent) طبق حالت جاری (State)، حرکتی (Action) را انجام می‌دهد و بر اساس آن حرکت بازخورد (Reward) دریافت می‌کند؛ این بازخورد ممکن است مثبت یا منفی (پاداش یا تنبیه) باشد و متناسب با این بازخورد خط‌مشی (Policy) خود را تغییر دهد.

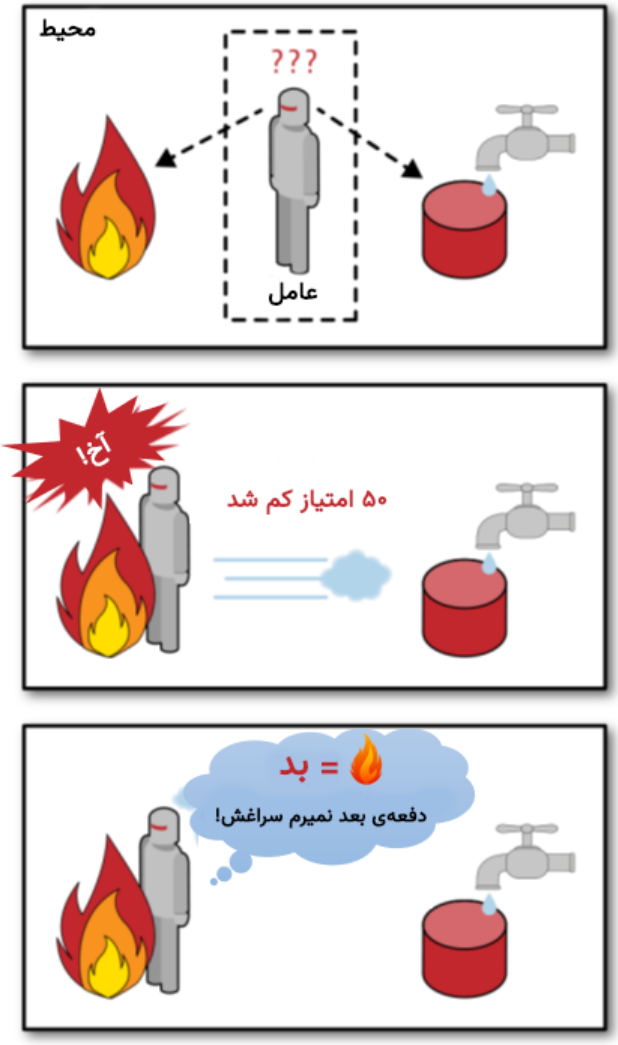
توضیح اضافه

در مثال هزارتو:

عامل هوشمند (Agent) شما هستید که سعی می‌کنید از هزارتو خارج شوید. حالت جاری (State) مختصات

مکان فعلی شما در هزارتو و اطلاعات بیشتری در مورد محیطی که در آن به یادگیری می‌پردازید را نشان می‌دهد. حرکت یا عمل (Action) گامی است که در یک جهت بر می‌دارید. بازخورد (Reward) امتیاز مثبت یا منفی‌ای است که دریافت می‌کنید تا بفهمید آیا در مسیر درستی قرار دارید یا نه! خط‌مشی (Policy) مشخص می‌کند که در هر حالت چه عملی را انتخاب کنید تا بهترین پاداش را بگیرید.

شکل زیر نشان می‌دهد که یک ربات چگونه یاد می‌گیرد که به آتش نزدیک نشود. (برگرفته از کتاب Hands On Machine Learning with Scikit-Learn، and Keras TensorFlow) این روش از یادگیری، نزدیک‌ترین حالت به یادگیری انسان است. یک کودک برای آموختن چگونه راه رفتن، مدام تلاش می‌کند و پدر و مادرش با تشویق کردن سعی می‌کنند به او در این یادگیری کمک کنند. برای حرف زدن هم انسان‌ها فرآیند مشابهی را طی می‌کنند. یادگیری تقویتی برخلاف یادگیری نظارت‌شده و یادگیری نظارت‌نشده وابسته به داده نیست، بلکه به واسطه‌ی تعامل با محیط می‌آموزد.



۱. مشاهده
۲. انتخاب حرکت با استفاده از خط‌مشی
۳. انجام حرکت!
۴. دریافت پاداش یا جزا
۵. به‌روزرسانی خط‌مشی (مرحله‌ی یادگیری)
۶. تکرار تا وقتی‌که خط‌مشی بهینه پیدا شود

این روش از یادگیری، نزدیک‌ترین حالت به یادگیری انسان است. یک کودک برای آموختن چگونه راه رفتن، مدام تلاش می‌کند و پدر و مادرش با تشویق کردن سعی می‌کنند به او در این یادگیری کمک کنند. برای حرف زدن هم انسان‌ها فرآیند مشابهی را طی می‌کنند. یادگیری تقویتی برخلاف یادگیری نظارت‌شده و یادگیری نظارت‌نشده وابسته به داده نیست، بلکه به واسطه‌ی تعامل با محیط می‌آموزد.

۵.۱ چرا پایتون؟

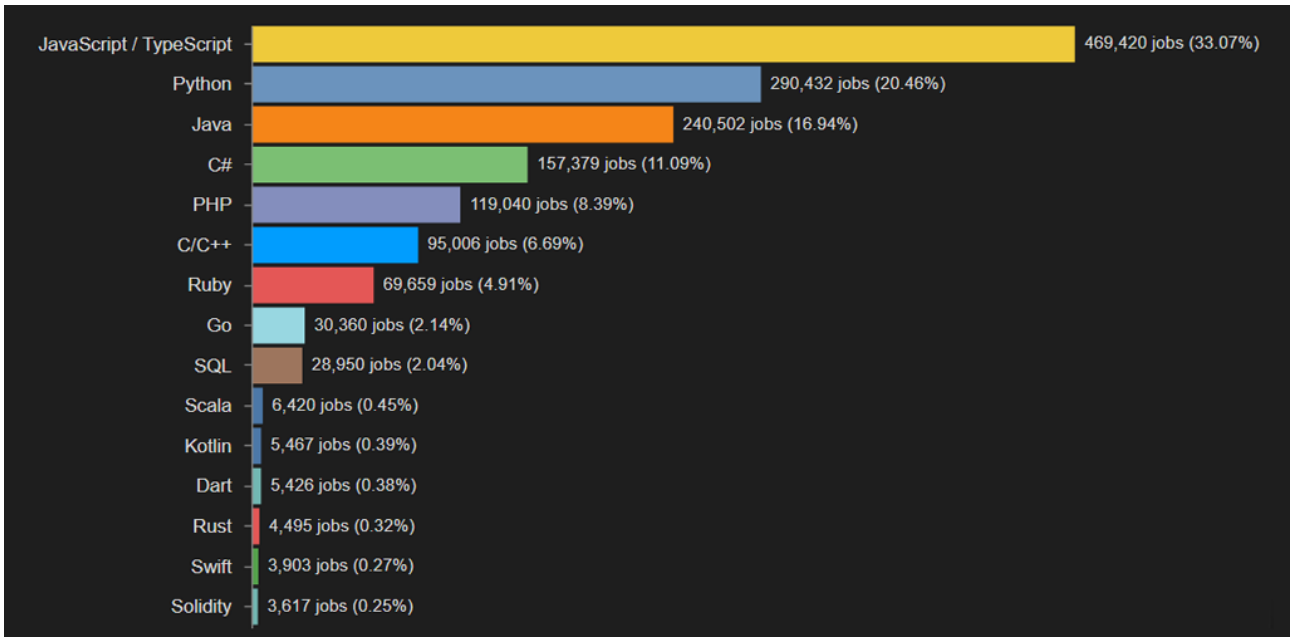
چرا پایتون؟

همان‌گونه که از اسم دوره انتظار می‌رود، زبان برنامه‌نویسی مورد استفاده‌ی ما، پایتون (Python) است. ممکن است تا به حال دوره‌های دیگر یا آگهی‌های شغلی زیادی دیده باشید که همگی از زبان برنامه‌نویسی پایتون استفاده می‌کرده‌اند. سوالی که اینجا مطرح می‌شود این است که دلیل محبوبیت این زبان چیست؟

به‌طور کلی، دلایل انتخاب پایتون در این دوره را می‌توان به چهار دسته تقسیم کرد. زبان غالب در صنعت

در اکثر شرکت‌هایی که با داده سر و کار دارند، زبان برنامه‌نویسی پایتون در حال استفاده است. چون می‌خواهیم دوره‌ی یادگیری ماشین، کاربردی باشد، نیاز است زبانی را انتخاب کنیم که شرکت‌های زیادتری متقاضی آن هستند. البته زبان‌های برنامه‌نویسی دیگر مانند MATLAB، R و Julia نیز در صنعت داده کاربرد خود را دارند، اما شاید بتوان گفت پایتون اولین انتخاب بسیاری از افراد است.

طبق گزارش تهیه‌شده توسط DevJobsScanner در سال ۲۰۲۲ که با تحلیل بیش از ۷ میلیون آگهی شغلی توسعه‌دهندگان به دست آمده، زبان پایتون توانسته رتبه‌ی دوم را کسب کند که خود نشان از اهمیت آن در صنعت دارد.



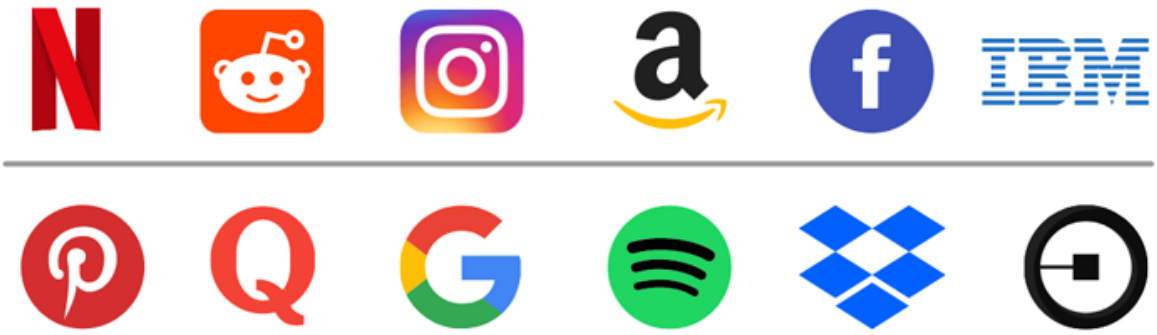
کتابخانه‌های متن‌باز فراوان

در پایتون کتابخانه‌های زیادی برای بارگذاری و دستکاری داده، مدل‌سازی، فعالیت‌های آماری، پردازش تصویر و پردازش متن وجود دارند. هر کدام از این کتابخانه‌ها، توابع زیادی در اختیار توسعه‌دهندگان قرار می‌دهند تا بتوانند به راحتی و در حالت بهینه، برنامه خود را توسعه دهند. به عنوان مثال Numpy برای کار با آرایه‌ها و اعمال ریاضی، pandas برای ذخیره‌سازی داده‌های جدولی و دستکاری داده، OpenCV برای کار با تصاویر و scikit-learn برای یادگیری ماشین در پایتون مورد استفاده قرار می‌گیرند.

تعامل با کد و داده

در علوم داده و یادگیری ماشین، تعامل با کد و توسعه تعاملی بسیار مرسوم و معمول است. در پایتون با یک تغییر جزئی در کد می‌توانیم از برنامه اجرا بگیریم، نتیجه را مشاهده و تحلیل کنیم و مجدداً کد را تغییر دهیم. این چرخه آن‌قدر تکرار می‌شود تا به نتیجه‌ی مطلوب برسیم. با کمک ترمینال یا ابزارهایی مانند جوپیتر، تعامل با کد در پایتون بسیار راحت خواهد بود. کاربردهای دیگر پایتون

پایتون یک زبان برنامه‌نویسی همه‌منظوره است. بنابراین کتابخانه‌های قدرتمندی برای ساخت واسط کاربری، کار با وب و دیگر کاربردها دارد. تصور کنید تنها با یادگیری یک زبان برنامه‌نویسی، می‌توانید یادگیری ماشین انجام داده و مدل را به کمک چارچوب‌هایی مانند جنگو، در وب قابل دسترس کرده و یا برای آن، واسط کاربری طراحی کنید! برخی از کمپانی‌های شناخته‌شده‌ای که از زبان پایتون در سرویس‌هایشان استفاده می‌کنند در تصویر زیر آمده است.



این دلایل تنها بخشی از عللی هستند که باعث می‌شوند اکثر دانشمندان داده یا مهندسان یادگیری ماشین، پایتون را به عنوان زبان اصلی خود انتخاب کنند.

۶.۱ آماده‌سازی محیط کار

آماده‌سازی محیط کار

اگر تصمیم دارید تا بتوانید کدها را بر روی سیستم خود اجرا کنید این درسنامه راه‌اندازی محیط مورد نیاز به کمک شما می‌آید. با آن‌که راه‌اندازی محیط توسعه (Development Environment) می‌تواند با توجه به مهارت‌ها و سلیقه‌های هر فرد به‌شکل متفاوتی صورت گیرد اما ما قصد داریم یک راه آسان و اصولی را پیش پای شما بگذاریم. در این درسنامه ابتدا با نحوه‌ی نصب پایتون و conda آشنا خواهید شد، سپس یک محیط مجازی خواهید ساخت و پکیج‌های پیش‌نیاز این دوره را بر روی محیط ساخته‌شده نصب خواهید کرد. در درسنامه‌های بعد نیز با نحوه‌ی راه‌اندازی یک محیط کدنویسی مناسب و حرفه‌ای آشنا خواهید شد.

۱. نصب پایتون

نسخه‌ی پیشنهادی ما برای این دوره، پایتون ۱۲.۳ است.

۷.۱ اجرای نوت‌بوک‌ها

[متن شما]

۸.۱ گوگل کولب

[متن شما]

۹.۱ معرفی مجموعه‌داده

[متن شما]

فصل ۲

مدیریت پروژه

۱.۲ اهداف فصل

اهداف فصل

فرض کنید تصمیم گرفته‌اید با دوستان‌تان یک سفر دسته‌جمعی آخر هفته داشته باشید. همه هیجان‌زده‌اند، اما هیچ‌کس دقیق نمی‌داند کجا می‌روید، کی حرکت می‌کنید، چه کسی ماشین می‌آورد و اصلاً چه چیزی باید با خود بیاورید! تنها چیزی که مشخص است، «هیجان سفر» است. صبح جمعه، عده‌ای خواب مانده‌اند، یکی یادش رفته بنزین بزند، و آن یکی وسط راه متوجه می‌شود مدارک ماشینش را جا گذاشته است. در آخر این سفر به‌جای خوش‌گذرانی تبدیل می‌شود به یک خاطره‌ی تلخ و خنده‌دار.

مدیریت پروژه هم درست همین‌طور است. اگر بدون برنامه‌ریزی، تعیین اولویت‌ها، تقسیم وظایف و فهمیدن دلایل شکست‌های قبلی وارد پروژه‌ای شویم، احتمال اینکه پروژه ما هم به سرنوشت آن سفر دچار شود، بسیار زیاد است.

در این فصل یاد می‌گیریم که چطور مثل یک رهبر حرفه‌ای پروژه را از ابتدا تا انتها مدیریت کنیم. در ادامه، خلاصه‌ای از مطالبی که در این فصل راجع به آن بحث می‌شود آورده شده است.

آشنایی با مفهوم چرخه‌ی پروژه و مراحل اصلی آن

بررسی روش‌های اولویت‌بندی وظایف و منابع در پروژه‌ها

آشنایی با اصول سازماندهی و مدیریت تیم پروژه

شناخت دلایل رایج شکست پروژه‌ها و راهکارهای پیشگیری از آن‌ها

۲.۲ چرخه پروژه

چرخه پروژه

هر پروژه‌ای (چه یادگیری ماشین باشد یا نباشد) برای پاسخ دادن به یک یا چند نیاز تعریف می‌شود. برای پاسخ‌دهی به این نیازها، هدفی مشخص می‌شود و برای دستیابی به آن هدف، پروژه شروع می‌شود. پس می‌توان اولین گام در هر پروژه‌ای را تعریف هدف آن در نظر گرفت.

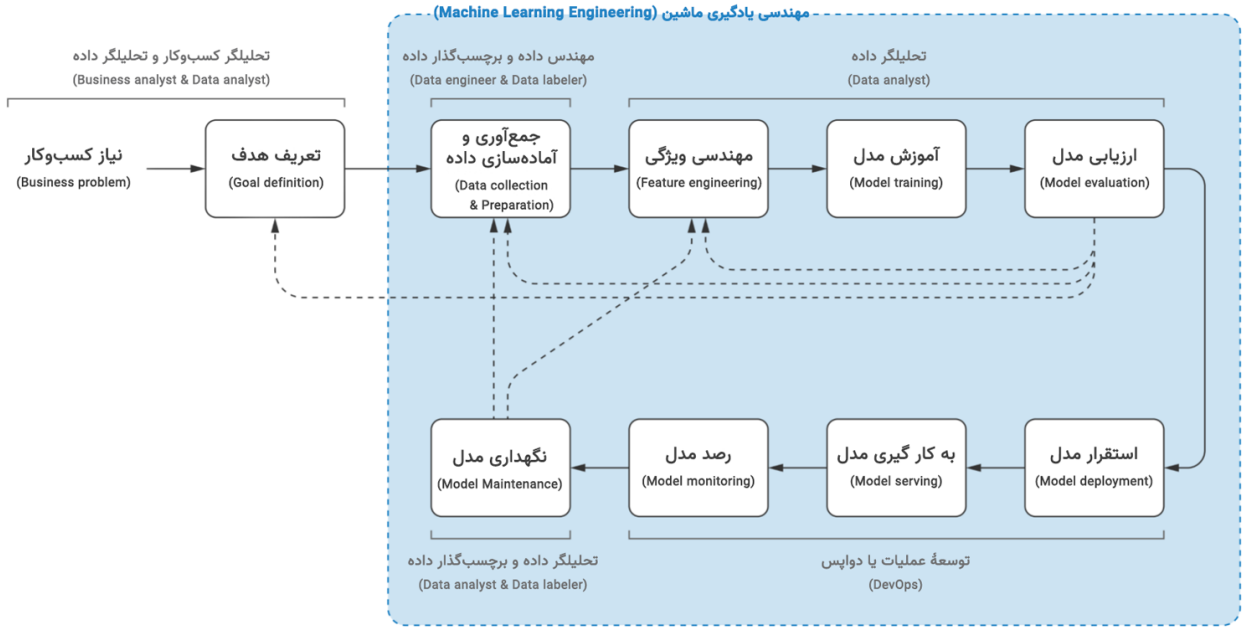
پروژه‌های یادگیری ماشین به خاطر یک نیاز کسب‌وکار شروع می‌شوند و این نیاز، هدف پروژه را تعریف می‌کند. متناسب با این هدف، تیم سازماندهی می‌شود و کار پروژه شروع می‌شود.

هدف تعریف شده پروژه، بایستی دارای ویژگی‌های زیر باشد:

میزان پیشرفت پروژه و معیار اتمام آن مشخص باشد. ورودی و خروجی مسئله واضح باشند.

توجه داشته باشید که لزوماً هدف یک پروژه‌ی یادگیری ماشین، همان هدف کسب‌وکار نیست. بلکه هدف کلی را به چند هدف کوچک‌تر می‌شکنیم و برای رسیدن به هر هدف پروژه‌ی مناسبی تعریف می‌کنیم. به عنوان مثال، کوئرا می‌خواهد رضایت کاربران از سامانه‌ی استخدام برنامه‌نویسان را افزایش دهد. یکی از پروژه‌های یادگیری ماشین که در رسیدن به این هدف به کوئرا کمک می‌کند، پیشنهاد آگهی‌های شغلی بر اساس رزومه به توسعه‌دهندگان است. یک پروژه دیگر، پیشنهاد توسعه‌دهنده‌ها با رزومه‌ی بهتر به شرکت‌هاست. چندین پروژه دیگر مانند دو مثالی که آورده شدند، تعریف و پیاده‌سازی شده‌اند تا هدف کسب‌وکار یعنی «افزایش رضایت کاربران» برآورده شوند.

به‌صورت کلی، یک پروژه‌ی یادگیری ماشین از ۹ گام زیر، تشکیل می‌شود. شکل زیر، مراحل مختلف یک پروژه یادگیری ماشین را نشان می‌دهد:



در این دوره، ما مراحل مختلف انجام یک پروژه را با هم می‌آموزیم. هنگام انجام تمرین‌ها و پروژه فرض می‌کنیم که مرحله‌ی اول یعنی تعیین هدف، از پیش مشخص شده و در اختیارمان قرار گرفته است. لیست زیر به‌صورت خلاصه، هر گام را توضیح می‌دهد. توضیحات مفصل‌تر را در هر فصل و درسنامه‌های آن مطالعه خواهید کرد. جمع‌آوری و پاک‌سازی (آماده‌سازی) داده‌ها

مهم‌ترین رکن در یادگیری ماشین، داده (data) است. در این مرحله، داده‌ی موردنیاز مسئله را از منابع موجود، استخراج می‌کنیم. داده‌های خام ممکن است نیاز به پاک‌سازی داشته باشند؛ به عنوان مثال، جنس ستون‌ها نیاز به تغییر داشته باشند یا مقادیر گم‌شده، بایستی که پر شوند. مهندسی ویژگی

پس از پاک‌سازی داده، داده‌ی خام به مجموعه‌ای تبدیل شده که فاقد خطا و مقادیر ناموجود است. اما همین کافی نیست. می‌توانیم با داشتن دانش زمینه‌ای از کسب‌وکار، مشاهدات و آزمون‌های آماری به مهندسی ویژگی (Feature Engineering) پرداخته ستون‌هایی را حذف یا ستون‌های جدیدی تولید کنیم تا مجموعه‌داده برای مرحله‌ی مدل‌سازی آماده شود. مدل‌سازی

با انتخاب الگوریتم‌های مناسب، می‌توانیم یک مدل (Model) یادگیری ماشین برای محاسبه‌ی رابطه بین متغیرهای مستقل (ویژگی‌ها) و متغیر وابسته (هدف) بسازیم. ارزیابی عملکرد مدل

وقت آن رسیده تا عملکرد مدل، مورد ارزیابی (Evaluation) قرار بگیرد. انتخاب معیار ارزیابی مناسب، مهم‌ترین تصمیمی است که بایستی در این مرحله گرفته شود. اگر که مدل نتواند امتیاز قابل قبولی از این معیار کسب کند، مهندس یادگیری ماشین بایستی با شناسایی علت، به دنبال راه‌کاری برای ساختن مدلی با عملکرد بهتر باشد. استقرار و به کار گیری مدل

پس از آن که مدلی ساختیم که حداقل عملکرد قابل قبول از لحاظ معیار ارزیابی را دارا بود، باید آن را برای استفاده‌ی عملی، مستقر (Deploy) کرد. منظور از مستقر کردن، قرار گرفتن در بستری است که کاربران بتوانند از آن استفاده کنند. در نهایت، باید درخواست‌های کاربران توسط کانالی مانند واسط کاربری، دریافت و به ورودی مورد استفاده مدل تبدیل شوند تا خروجی آن، برای کاربر برگردانده شود. رصد و نگهداری

منظور از رصد، پیگیری عملکرد مدل در طول زمان می‌باشد تا مطمئن شویم که مدل همچنان معتبر و دارای عملکرد قابل‌قبولی است. در صورتی که مدل، حداقل عملکرد قابل‌قبول را نداشته باشد، نیاز است که اصلاحاتی

مانند آموزش روی نمونه‌های جدید، روی مدل صورت بگیرد؛ به مجموعه کارهایی که به منظور حفظ کیفیت مدل انجام می‌شود، نگهداری (Maintenance) می‌گوییم.

۳.۲ اولویت‌بندی

اولویت‌بندی

هنگامی‌که به عنوان مهندس یادگیری ماشین، در یک شرکت یا سازمان مشغول به کار می‌شوید، احتمالاً ایده‌های مختلفی جهت استفاده از قدرت یادگیری ماشین، با شما به اشتراک گذاشته می‌شود و شما بایستی که آن‌ها را بر اساس یک اولویت‌بندی، انجام دهید. اما این اولویت‌بندی چگونه انجام می‌شود؟ این امر، می‌تواند تبدیل به یک تصمیم سلیقه‌ای شود.

در این درسنامه چارچوبی را معرفی خواهیم کرد که به کمک آن بتوانید به‌صورت ساختارمند، در اینباره تصمیم بگیرید. جهت اولویت‌بندی نیاز است برای هر ایده، دو مورد «ارزش «(Value) و «امکان‌پذیری «(Feasibility) آن را حساب کنید. در ادامه به معرفی کامل هرکدام از این اصطلاحات می‌پردازیم. ارزش (Value)

استفاده از یادگیری ماشین در یک پروژه، هنگامی دارای «ارزش» زیادی است که: ۱) یادگیری ماشین بتواند جایگزین بخش پیچیده‌ای از پروژه شود.

به عنوان مثال، بخش پیچیده‌ای از یک سامانه‌ی موجود، می‌تواند مبتنی بر قواعد یا به اصطلاح Rule-based باشد. در نتیجه، شما با قطعه کدی روبه‌رو می‌شوید که دارای تعداد زیادی دستور مانند if-else به صورت تودرتو، پیچیده و با استثنائات مختلفی هستند. نگه‌داری و توسعه‌ی چنین کدی در طول زمان می‌تواند دشوار، زمان‌بر و مستعد خطا باشد. همچنین این بخش سیستم، می‌تواند به قسمتی تبدیل شود که مهندسین نرم‌افزار از آن فراری باشند. آیا می‌شود که این دستورات، به جای آن که برنامه‌نویسی شوند، یاد گرفته شوند؟ (اگر جواب این سوال، بله است؛ استفاده از یادگیری ماشین، می‌تواند ارزشمند باشد.) ۲) دستیابی به پیش‌بینی‌های ارزان ولی احتمالاً با کمی خطا، دارای مزیت باشد.

به عنوان مثال، در سیستمی که تعداد زیادی درخواست (request) از کاربران دریافت می‌کند، فرض کنید که تعداد زیادی از این درخواست‌ها، «آسان» هستند و می‌توانند توسط اتوماسیون به سرعت انجام شوند. در نتیجه، فقط لازم است که درخواست‌های باقی‌مانده که «سخت» طبقه‌بندی می‌شوند، به‌صورت دستی مورد بررسی قرار بگیرند.

یک سیستم مبتنی بر یادگیری ماشین که درخواست‌های «آسان» را از «سخت» تشخیص می‌دهد، می‌تواند زمان زیادی را از وقت نیروی انسانی صرفه‌جویی کند. زیرا که نیروی انسانی فقط لازم است روی درخواست‌هایی که نیاز به دخالت دستی دارد، تمرکز کند. همچنین اگر که درخواست سختی به اشتباه آسان پیش‌بینی شود، اتوماسیون در پردازش آن‌ها، دچار خطا می‌شود. در نتیجه، چنین درخواست‌هایی می‌تواند توسط نیروی انسانی مورد ارزیابی مجدد قرار بگیرد. از طرف دیگر، اگر که یک نیروی انسانی، به اشتباه درخواست آسانی دریافت کند، هیچ مشکلی وجود ندارد. چون او می‌تواند تشخیص دهد که این درخواست آسان است و آن را به اتوماسیون جهت پردازش ارسال کند. ۳) آن ایده، بتواند ارزش مالی قابل توجهی برای سازمان بیاورد.

ارزش مالی می‌تواند به صورت کاهش هزینه‌ها (حقوق نیروی انسانی) و افزایش درآمد (میزان فروش) مورد حساب قرار گیرد. همچنین مواردی مانند افزایش رضایت مشتریان نیز می‌توانند به عنوان ارزش مالی لحاظ گردند. امکان‌پذیری (Feasibility)

امکان‌پذیری یک پروژه یادگیری ماشین، می‌تواند توسط ۳ عامل زیر، مشخص شود: ۱) سختی مسئله

آیا یک الگوریتم پیاده‌سازی شده و یا یک کتابخانه‌ی نرم‌افزاری برای حل این مسئله موجود است؟ اگر بله، تا حد خوبی مسئله آسان می‌شود. آیا نیاز به زیرساخت‌های محاسباتی خاصی برای ساختن مدل و یا استفاده از آن در عمل (production) است؟ اگر خیر، مسئله تا حد خوبی آسان می‌شود.

همچنین برای تخمین سختی یک مسئله، تعدادی مجهول وجود دارد که اگر در پروژه‌های مشابه کار کرده باشید و یا نتایج مشابهی را مطالعه کرده باشید، می‌توانید آن‌ها را حدس بزنید. این مجهولات عبارتند از:

آیا حداقل عملکرد مطلوب در عمل قابل دستیابی است؟ برای دسترسی به عملکرد مورد نظر، چه میزان داده مورد نیاز است؟ چه ویژگی‌هایی و چندتا از آن‌ها لازم است تا بتوانیم یک مدل قابل اتکای یادگیری ماشین برای استفاده در عمل بسازیم؟ الگوریتم استفاده شده برای آموزش مدل باید چقدر بزرگ باشد؟ آموزش مدل و زمان پیش‌بینی کردن آن، چقدر طول می‌کشد؟ چقدر زمان و چند مرحله آموزش لازم است تا که مدل شما به حداقل دقت مطلوب برسد؟

ساده‌سازی مسئله‌ی اصلی نیز یک روش حدس زدن است. برای مثال، فرض کنید که در مجموعه‌داده‌ی خانه‌های شهر پکن، شما می‌خواهید قیمت یک سری خانه را پیش‌بینی کنید. به منظور ساده‌سازی این مسئله، می‌توانید ابتدا قیمت‌های خانه‌ها را به گروه‌هایی مانند گران، متوسط و ارزان دسته‌بندی کنید. بدین صورت، صورت‌مسئله‌ی شما به دسته‌بندی قیمت خانه‌ها تبدیل می‌شود و این مسئله حتی برای یک انسان نیز ساده‌تر است. در نتیجه، مسئله‌ی ساده‌تری برای حل توسط یادگیری ماشین داریم.

در صورتی‌که یادگیری ماشین بتواند این مسئله را حل کند، می‌توان امید داشت که بتواند مسئله‌ی اصلی که سخت‌تر است را نیز حل کند. همچنین اطلاعاتی که در هنگام حل مسئله‌ی ساده‌تر کسب می‌کنیم به ما کمک می‌کند که تخمین بهتری از حل مسئله‌ی اصلی پیدا کنیم. توجه داشته باشید که نمی‌توانید به‌سادگی فقط با یک ضرب، تخمین مورد نظر را روی مسئله‌ی اصلی کسب کنید ولی معمولاً افزایش تعداد گروه‌های مورد نیاز برای پیش‌بینی در یک مسئله، نیازمند میزان بیشتری نمونه برای مجموعه‌داده‌ی آموزش خواهد بود و این رابطه خطی نیست!

همچنین، شما می‌توانید با استفاده از تعاریف موجود در داده‌ها، مسئله اصلی را به مسائل کوچک‌تر و ساده‌تر تبدیل کنید. فرض کنید که در مجموعه‌داده‌ی خانه‌های شهر پکن، به‌جای آن‌که یک مدل برای قیمت‌گذاری تمامی املاک مناطق مختلف داشته باشید، برای هر منطقه، یک مدل مجزا بسازید چون که به‌صورت طبیعی انتظار می‌رود که مثلاً بازار املاک شمال شهر با بازار املاک جنوب شهر، رفتار متفاوتی داشته باشند. ۲) هزینه دسترسی به داده

به دست آوردن مجموعه‌داده‌ی مناسب و در مقدار مناسب می‌تواند بسیار هزینه‌بر باشد. به خصوص وقتی‌که آن داده‌ها، نیاز به برچسب زدن دستی داشته باشند. در نتیجه، نیاز است که ما به سوالات زیر در مورد هزینه‌ی دسترسی به داده‌ها پاسخ دهیم:

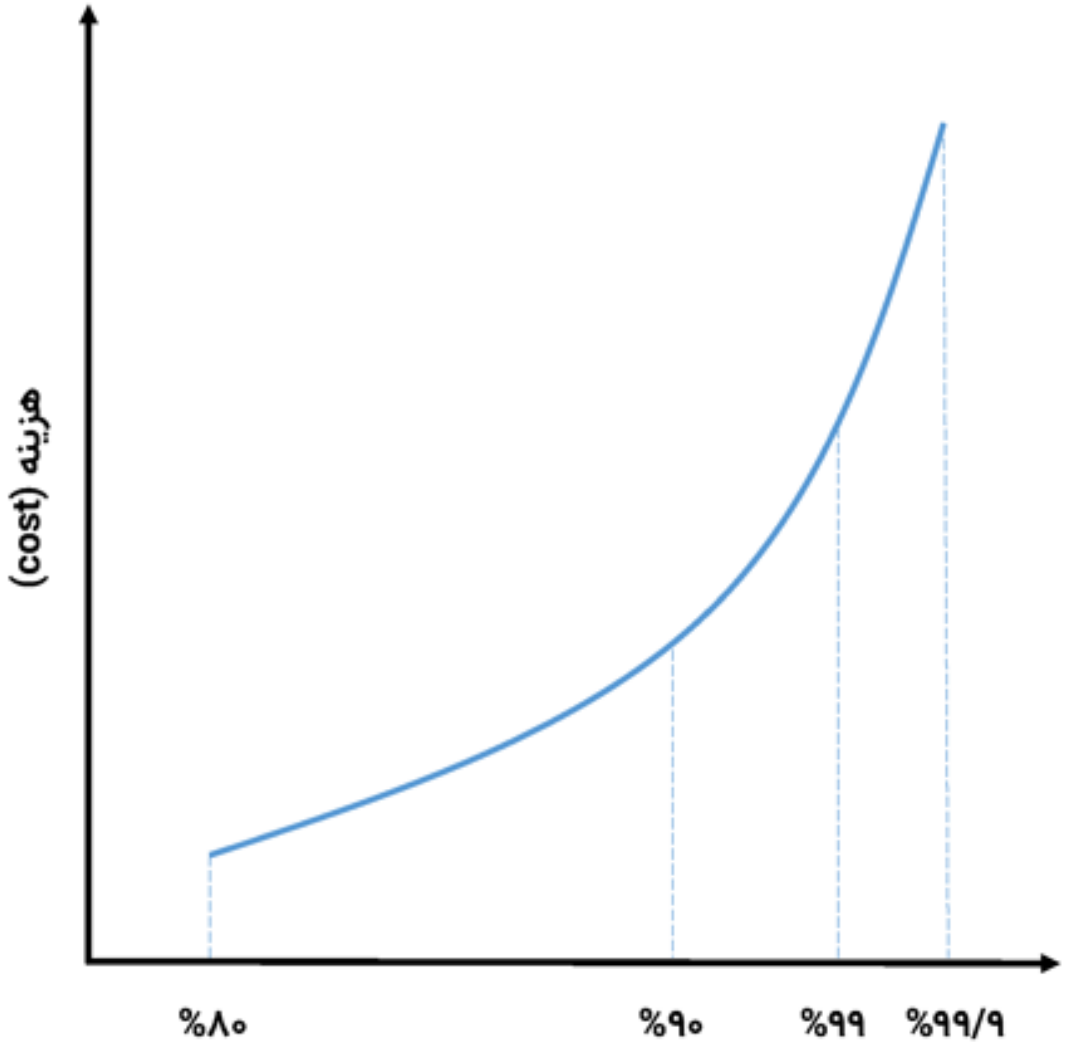
آیا می‌توان مجموعه‌داده‌ی مورد نیاز را به‌صورت خودکار تولید کرد؟ اگر بله، دسترسی به داده‌ها تا حد زیادی، «ساده» می‌شود. هزینه‌ی برچسب‌زدن دستی به مجموعه‌داده‌ای بدون برچسب چقدر است؟ چه میزان داده مورد نیاز است؟ (معمولاً نمی‌توان این عدد را از قبل از شروع پروژه دانست، اما می‌توان بر اساس نتایج مشابه منتشر شده و یا تجربیات قبلی، آن را تخمین زد.)

۳) حداقل عملکرد مطلوب

داشتن یک سیستم یادگیری ماشین با عملکرد بالا، می‌تواند به معنی نیاز به داده‌های بیشتر یا استفاده

از مدل‌های پیچیده‌ای باشد که از منظر سخت‌افزاری یا طراحی هزینه‌ی بالایی دارند. به همین دلیل، در محاسبه‌ی امکان‌پذیری، بایستی که حداقل عملکرد مطلوب سیستم یادگیری ماشین را لحاظ کنیم.

طبق تصویر زیر، می‌توان گفت که هزینه‌ی یک پروژه یادگیری ماشین با عملکرد مطلوب آن، یک رابطه‌ی نمایی دارد. به عبارت دیگر، هرچه که حداقل عملکرد مطلوب بیشتری نیاز داشته باشیم، امکان‌پذیری پیاده‌سازی چنین سیستمی، کاهش پیدا کرده و سخت‌تر می‌شود.



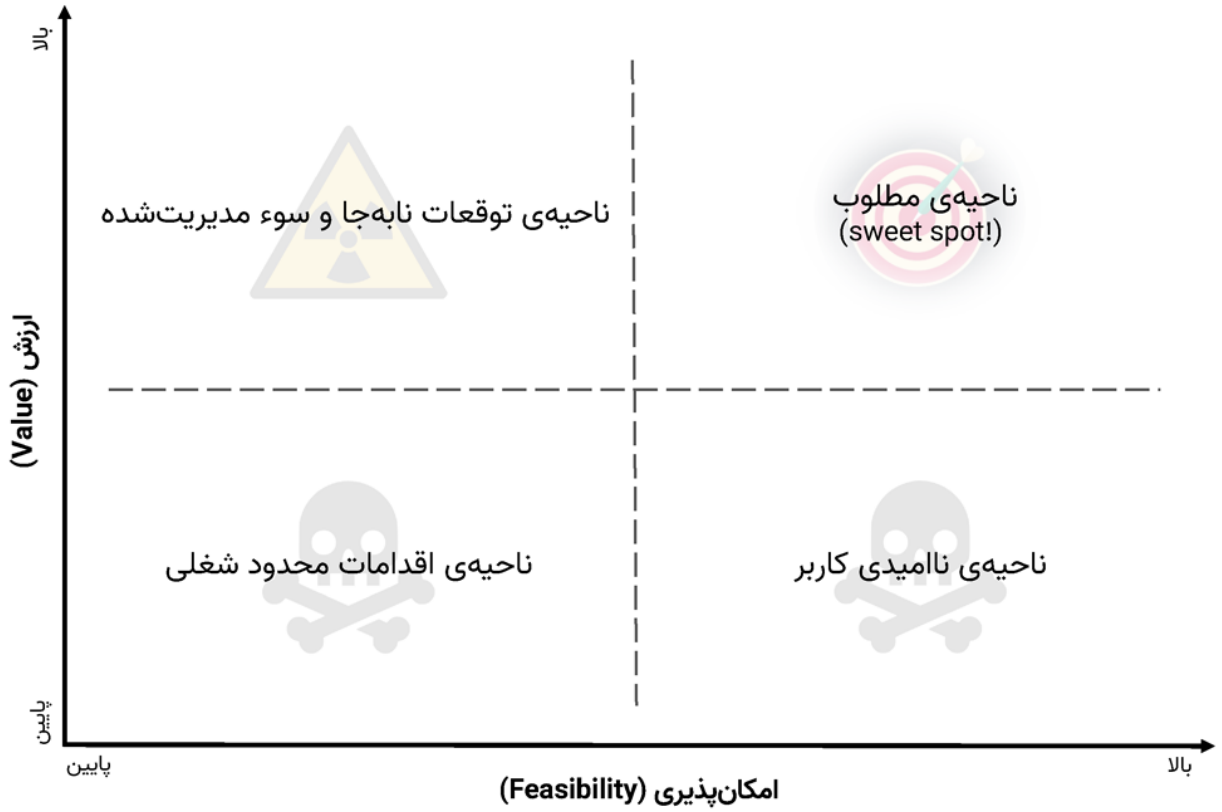
همچنین باید در نظر داشته باشیم که پیشرفت یک پروژه‌ی یادگیری ماشین، غیرخطی است. در هفته‌های ابتدایی، سرعت افزایش عملکرد مدل بالا است ولی بعد از آن، بهبود عملکرد خیلی با کندی سرعت می‌گیرد. به همین دلیل، شما بایستی که قبل از شروع پروژه، این مورد را در نظر داشته باشید و بتوانید انتظارات مدیر محصول را مدیریت کنید.

با توجه به این که یک سیستم یادگیری ماشین در عمل مورد استفاده قرار خواهد گرفت، اگر که دارای عملکرد پایینی باشد می‌تواند ما را دچار ضرر و زیان کند. بنابراین بایستی که عملکرد مطلوب را بر اساس نکات زیر محاسبه کنیم:

هر پیش‌بینی اشتباه چقدر هزینه‌بر است؟ پایین‌ترین سطح عملکرد که زیر آن، پیش‌بینی‌های مدل غیرکاربردی می‌شوند، چقدر است؟

اولویت‌بندی

بعد از آن که برای هر ایده، ارزش و امکان‌پذیری آن را حساب کردید. ایده‌ها را همانند شکل زیر روی نمودار دو بعدی قرار دهید:



در این نمودار، محور افقی، نشان‌دهنده میزان امکان‌پذیری پروژه است و محور عمودی، میزان ارزش آن را نشان می‌دهد. بعد از این که ایده‌های مختلف به‌صورت نسبی از همدیگر در این نمودار دوبعدی قرار داده شدند، اولویت شروع با ایده‌هایی هست که هم ارزش بالایی برای شرکت/سازمان بیاورند و هم امکان‌پذیری بیشتری از بقیه داشته باشند.

۴.۲ سازماندهی تیم

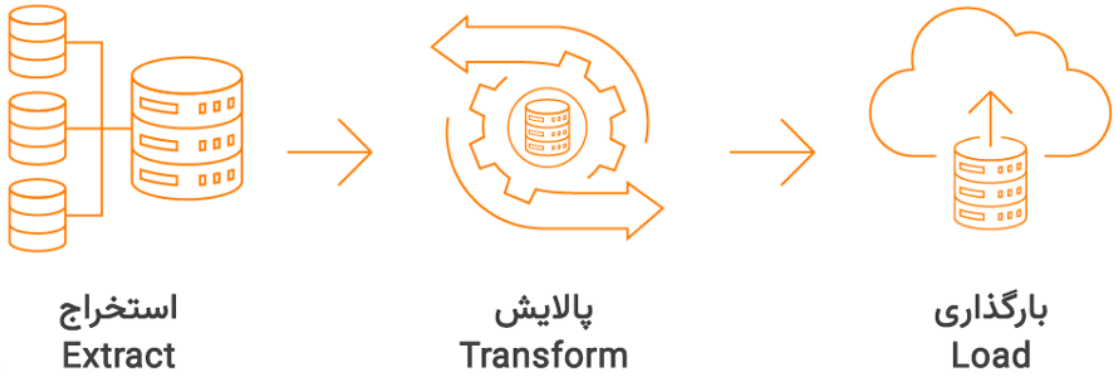
سازماندهی تیم

دو دیدگاه متفاوت در سازماندهی تیم‌های یادگیری ماشین وجود دارد. در دیدگاه نخست تیم یادگیری ماشین می‌تواند متشکل از متخصصینی با مهارت‌های مختلف (مثلاً تحلیلگر داده، مهندس نرم‌افزار و غیره) باشد که در کنار همدیگر کار می‌کنند و صرفاً زبان همدیگر را می‌فهمند. در دیدگاه دوم تمام افراد تیم باید ترکیبی از مهارت‌های مختلف را دارا باشند. معمولاً در شرکت‌های بزرگی همچون گوگل که افرادی با تخصص‌های مشخص جذب می‌کنند دیدگاه نخست در پیش گرفته می‌شود و دیدگاه دوم بیشتر مورد استفاده‌ی شرکت‌های کوچک و استارت‌آپ‌ها قرار می‌گیرد. در ادامه به شرح کامل‌تر هرکدام از این رویکردها خواهیم پرداخت. ۱) دیدگاه تخصص‌محور

در این دیدگاه، یک تیم یادگیری ماشین از افرادی تشکیل می‌شود که هر کدام وظیفه‌ی مشخصی دارند و فقط در وظیفه‌ی خود بسیار ماهر هستند. این تیم بسته به اندازه و حجم کار یک شرکت، می‌تواند از مهندسان نرم‌افزار، تحلیلگران داده، مهندسان داده، دانشمندان داده و مهندسان یادگیری ماشین، تشکیل شود.

به عنوان مثال، تیم یادگیری ماشین در یک کارگزاری قیمت مسکن را تصور کنید که در حال توسعه پروژه‌ای جهت تخمین قیمت خانه‌هاست. برای این پروژه، مهندسان داده، داده‌های خام را از پایگاه‌های داده استخراج کرده و تغییراتی روی آن اعمال می‌کنند تا آماده‌ی استفاده برای تحلیلگران داده شود.

فرآیند ETL چیست؟



در سازمان‌ها جهت جمع‌آوری، ترکیب و سنتز داده‌های خام از منابع مختلف معمولاً از فرآیندی به نام ETL استفاده می‌شود که هر حرف آن کوتاه‌شده‌ی کلمات زیر است:

استخراج :(Extract) در ابتدا داده‌ها از منابع مختلفی استخراج و جمع‌آوری می‌شوند.

پالایش :(Transform) در این مرحله داده‌ها مطابق با کسب‌وکار مربوطه به روش‌های مختلفی (همچون پاک‌سازی داده، مرتب‌سازی، فیلتر، حذف داده‌های تکراری و غیره) پالایش می‌شوند.

بارگذاری :(Load) در نهایت داده‌های پالایش‌شده در یک منبع داده ذخیره می‌شوند.

مهندسان داده همواره سعی می‌کنند فرآیند ETL را خودکار کنند تا دسترسی کسانی که از داده استفاده می‌کنند به آن راحت‌تر انجام شود. تحلیلگران داده، آمار و حقایقی از مولفه‌های مختلف خانه‌های موجود در پایگاه‌داده را استخراج می‌کنند و با ارائه‌ی نمودارها و گزارش‌ها، به تیم توسعه‌ی کسب‌وکار و تیم محصول در گرفتن تصمیمات داده‌محور کمک می‌کنند. دانشمندان داده و مهندسان یادگیری ماشین، مسئول پاک‌سازی داده، مهندسی ویژگی و مدل‌سازی هستند. آن‌ها مدل‌های توسعه‌یافته را در زیرساخت مناسب مستقر، رصد و نگهداری می‌کنند. مهندسان نرم‌افزار، برنامه‌ای مانند یک وبسایت، اپلیکیشن موبایل یا دسکتاپ، طراحی و پیاده‌سازی می‌کنند تا که از مدل مستقرشده استفاده و به کاربر نهایی، خدمت ارائه کند. ۲) دیدگاه تیم چابک (Agile)

در این دیدگاه از تعداد کمی از افراد که هر یک توانایی انجام تخصص‌های مختلفی را دارند، برای داشتن یک تیم چابک استفاده می‌شود. برای مثال افراد چنین تیمی می‌توانند هم کار مهندسی داده، هم کار مهندسی یادگیری ماشین و هم کار مهندسی نرم‌افزار را انجام دهند. در نتیجه، برای افراد محدودیتی در انجام کارها وجود ندارد، اما دست‌یابی به این هدف نیازمندِ عضویت افرادی با چند تخصص در تیم است. درست است که تیم، چابک و کوچک می‌شود، اما این به خاطر این است که این افراد حرفه‌ای‌تر هستند و می‌توانند چند

تخصص مختلف را با هم انجام دهند. بنابراین، تشکیل چنین تیمی هم سخت‌تر است و هم هزینه‌ی آن به ازای هر نفر، بیشتر از افراد تیم تشکیل‌شده در دیدگاه اول می‌شود.

هر دو دیدگاه مزایا و معایبی دارند. برخی در طرفداری از دیدگاه اول می‌گویند، هر کس باید در جایی که هست، بهترین باشد. مثلاً یک دانشمند داده و مهندس یادگیری ماشین بتواند به بهترین نحو ممکن مدل‌سازی کند و مدلی توسعه دهد که به بیشترین عملکرد ممکن دست پیدا کند. عده‌ای دیگر در طرفداری از دیدگاه دوم این‌گونه استدلال می‌کنند که معمولاً توسعه‌دهندگان مربوط به داده، اصول مهندسی را رعایت نمی‌کنند. به عنوان مثال، تمیز و ساختاریافته کد نمی‌نویسند. بیشتر به فکر بهتر کردن عملکرد مدل هستند و امکان استقرار مدل در محیط محصول (Production Environment) را در نظر نمی‌گیرند. بنابراین هنگام ترکیب کردن کد افراد با تخصص‌های داده‌محور با کد مهندسی‌شده و تمیز، هزینه‌ی زیادی مصرف می‌شود و گاهی حتی نیاز است تمام یا بخشی از کدها توسط مهندسان نرم‌افزار، بازنویسی شوند.

اگر تمایل دارید در مورد موقعیت‌های شغلی مرتبط با داده بیشتر مطالعه کنید، این درسنامه از دوره دروازه ورود به یادگیری ماشین، منبع خوبی است. همچنین اگر به دنبال شغل در این حوزه هستید، حتما به قسمت کاریابی کوئرا بروید و برای شغل‌های موجود این حوزه، اقدام بکنید.

۵.۲ چرا پروژه‌ها شکست می‌خورند؟

چرا پروژه‌ها شکست می‌خورند؟

طبق برآوردهای مختلف بین سالهای ۲۰۱۷ تا ۲۰۲۰ میلادی، از ۷۴ تا ۸۷ درصد از پروژه‌های یادگیری ماشین شکست می‌خورند یا به مرحله‌ی استفاده عملی توسط یک شرکت و سازمان نمی‌رسند. دلایل مختلفی برای شکست این پروژه‌ها وجود دارد و در این درسنامه، به مهم‌ترین آن‌ها اشاره خواهیم کرد. شاید اطلاع داشتن از این دلایل، بتواند مانع شکست پروژه یادگیری ماشین شما بشود. ۱) کمبود نیروی انسانی باتجربه

مهندسی یادگیری ماشین، هنوز رشته‌ی نسبتاً جدیدی است و روش استانداردی برای آموزش آن وجود ندارد. بیشتر سازمان‌ها نمی‌دانند چگونه در این حوزه، متخصص استخدام کنند. همچنین، اکثر نیروی انسانی موجود در بازار کار، افرادی هستند که فقط یک یا چند دوره آنلاین گذرانده‌اند و تجربه‌ی عملی قابل توجهی ندارند. در نتیجه، بخش قابل توجهی از نیروی کار موجود، فقط دارای تخصص سطحی و تجربه‌ی کار با داده‌های مورد استفاده در کلاس‌های درس هستند. لازم به ذکر است که بسیاری از این افراد، تجربه‌ی کار بر روی تمامی مراحل یک چرخه پروژه یادگیری ماشین را نیز ندارند. ۲) عدم پشتیبانی توسط مدیران ارشد

معمولاً مهندسین یادگیری ماشین و نرم‌افزار دارای اهداف، انگیزه‌ها و معیارهای موفقیت متفاوتی هستند. آن‌ها همچنین بسیار متفاوت عمل می‌کنند. در یک سازمان چابک، مهندسین نرم‌افزار از متدولوژی اسکرام (Scrum) استفاده کرده و وظایف خود را در اسپرینت‌هایی با خروجی‌های قابل تعریف و عدم قطعیت کم، جلو می‌برند. در حالی که، مهندسین یادگیری ماشین به علت ذات این حوزه که با اکتشاف و آزمایش‌های متعدد همراه است، با عدم قطعیت بیشتری، مواجه هستند. بسیاری از این آزمایش‌ها، به یک نتیجه‌ی قابل تحویل ختم نمی‌شود و یا بعد از پایان وظایف تعریف شده، دقت مورد نظر سیستم، بهبود مورد نظر را پیدا نمی‌کند و نیاز به بازتعریف کارها و انجام مراحل جدید می‌باشد. در نتیجه، این موارد می‌تواند توسط افراد خارج از تیم، اتلاف وقت تلقی شوند.

این موارد را در کنار این حقیقت قرار بدهید که در اکثر شرکت‌ها، مدیران رتبه بالای آن، دارای تجربه کار فنی و مهندسی نیستند. آن‌ها نمی‌دانند که هوش مصنوعی چگونه کار می‌کند و یا درک سطحی یا بیش از حد خوش‌بینانه‌ای دارند و فکر می‌کنند که با منابع فنی و انسانی، هوش مصنوعی بتواند در کوتاه مدت هر مشکلی را حل کند. هنگامی‌که پیشرفت سریع توسط تیم یادگیری ماشین اتفاق نمی‌افتد، آن‌ها به راحتی، اعضای تیم را سرزنش می‌کنند یا علاقه‌ی خود را به هوش مصنوعی به عنوان یک ابزار بی‌اثر با نتایج پیش‌بینی نشده و نامشخص به‌طور کامل از دست می‌دهند.

این مشکلات را در کنار ناتوانی یک مهندس یادگیری ماشین در مکاتبه‌ی نتایج و چالش‌های خود با مدیریت قرار دهید. دلیل این امر این است که این دو گروه، واژگان مشترکی ندارند و از لحاظ فنی دارای سطوح بسیار متفاوتی هستند. حتی موفقیتی که بد ارائه شود، می‌تواند به عنوان یک شکست توسط مدیران تلقی شود. به همین دلیل است که در سازمان‌های موفق در این حوزه، مدیران سطح بالا و مسئول هوش مصنوعی، اغلب دارای سابقه‌ی فنی یا علمی متناسب هستند. ۳) نبود زیرساخت داده

تیم یادگیری ماشین، با داده سر و کار دارند و کیفیت داده برای موفقیت یک پروژه یادگیری ماشین بسیار مهم است. به همین دلیل است که اعضای این تیم بایستی بتوانند به‌سادگی به وسیله‌ی زیرساخت داده‌ی یک شرکت/سازمان به داده‌هایی باکیفیت دسترسی داشته باشند. در عین حال، بایستی این اطمینان حاصل شود که در هنگام استفاده از مدلِ آموزش‌دیده برای پیش‌بینی روی نمونه‌های جدید، آن نمونه‌ها نیز دارای کیفیت مشابه باشند.

متاسفانه در اغلب موارد و به دلیل عدم وجود یک تیم مهندسی داده (Data Engineering)، این زیرساخت داده در یک شرکت و یا سازمان وجود ندارد. در نتیجه، تیم یادگیری ماشین مجبور می‌شود با استفاده از اسکریپت‌های مختلف موقت، مجموعه‌داده‌ی مربوط به آموزش هر پروژه را به‌دست آورند. این امر باعث افزایش پیچیدگی برای دسترسی به داده‌های با کیفیت در پروژه‌های جدید و همچنین سختی نگهداری کد و تکرار نتایج می‌شود. ۴) چالش برچسب‌گذاری داده

اکثر پروژه‌های یادگیری ماشین، از نوع یادگیری نظارت‌شده هستند و تیم یادگیری ماشین از مجموعه‌داده‌ی دارای برچسب برای آن‌ها استفاده می‌کنند. این مجموعه‌داده‌ها معمولاً سفارشی هستند، بنابراین برچسب‌زدن به‌طور خاص برای هر پروژه انجام می‌شود. بر اساس برخی گزارش‌ها تا سال ۲۰۱۹، حدود ۷۶ درصد تیم‌ها،

خود، مجموعه‌داده‌های مورد نیاز را برچسب‌گذاری می‌کردند و فقط ۶۳ درصد از آن‌ها، فرآیند برچسب‌گذاری را خودکار کرده‌اند.

این امر منجر به صرف زمان قابل توجهی توسط اعضای ماهر این تیم‌ها، بر روی برچسب‌زدن و توسعه‌ی ابزار برچسب‌گذاری می‌شود و این یک چالش بزرگ برای اجرای موثر یک پروژه یادگیری ماشین است. به همین دلیل است که برخی از شرکت‌ها، فرآیند برچسب‌گذاری داده را به شرکت‌های دیگر، برون‌سپاری می‌کنند. با این حال، این تیم‌ها بایستی که کیفیت داده‌های برچسب‌گذاری شده دریافتی را نیز تایید کنند تا از کیفیت پایین و یا اشتباه بودن آن‌ها جلوگیری کنند.

به منظور حفظ کیفیت و ثبات، بعضی از شرکت‌ها و سازمان‌ها اقدام به آموزش دادن و استخدام افراد برچسب‌زن (افراد داخل یا از شرکت بیرونی) کرده‌اند و این امر، به نوبه خود می‌تواند باعث کندی پیشرفت پروژه‌های یادگیری ماشین بشود. در نظر داشته باشید که بر اساس گزارشات مشابه، شرکت‌هایی که برچسب‌گذاری داده‌ها را برون‌سپاری می‌کنند، به احتمال زیاد و خوبی پروژه‌های یادگیری ماشین خود را به مرحله‌ی استفاده عملی می‌رسانند. ۵) عدم همکاری

مجموعه‌داده‌ی مورد نیاز برای یک پروژه یادگیری ماشین، اغلب در دپارتمان‌های مختلف یک شرکت/سازمان و تحت مالکیت و محدودیت‌های امنیتی آن‌ها هستند. همچنین، این داده‌ها می‌توانند هریک دارای قالب مختلفی باشند. در نظر داشته باشید که افراد مسئول داده در دپارتمان‌های مختلف، می‌توانند هریک دارای اهداف و اولویت‌های خود باشند. عدم اعتماد/شناخت و همکاری بین این افراد، می‌تواند باعث اصطکاک، کندی و یا توقف طولانی‌مدت پروژه شوند.

علاوه بر این، معمولاً دپارتمان‌های مختلف، بودجه‌های خاص خود را دارند و شاید علاقه‌ای به صرف آن بودجه برای کمک به دپارتمان دیگر نکنند. این موارد را در کنار عدم همکاری‌هایی که بین افراد مختلف یک تیم یادگیری ماشین می‌تواند رخ دهد، قرار دهید. ۶) پروژه‌های امکان‌ناپذیر

به‌خاطر هزینه نیروی انسانی و زیرساخت داده، بسیاری از پروژه‌های یادگیری ماشین دارای هزینه‌ی بالایی برای شرکت/سازمان هستند. به همین دلیل و برای جبران سرمایه‌گذاری، ممکن است که شرکت/سازمان، اهداف بسیار بلندپروازانه‌ای مانند تغییر کامل سازمان یا محصول یا وعده‌ی غیرواقعی بازگشت سرمایه بدهند.

در نتیجه، پروژه‌های بسیار بزرگ که شامل همکاری بین تیم‌های متعدد، دپارتمان‌ها و شرکت‌های بیرونی می‌شود، فشار زیادی را روی نیروی انسانی وارد می‌کنند. چنین پروژه‌های بیش از حد بلندپروازانه‌ای، ممکن است ماه‌ها یا حتی سال‌ها به طول انجامد. در طول این مدت، ممکن است که مدیران ارشد علاقه‌ی خود به این پروژه را از دست داده و به تمام حوزه‌ی یادگیری ماشین بدبین شوند و این حوزه از لیست پروژه‌های با اولویت‌های بالا خارج شود.

در نظر داشته باشید که اگر پروژه‌ی یادگیری ماشین خیلی طول بکشد تا تکمیل شود، ممکن است که دیر وارد بازار شود و در عمل به اهداف خود دست پیدا نکند. به همین دلیل است که پیشنهاد می‌کنیم در ابتدا، روی پروژه‌های امکان‌پذیر از لحاظ فنی که شامل همکاری‌های ساده بین تیم‌ها و دپارتمان‌ها هستند، تمرکز کنید تا که یک هدف تجاری ساده را پاسخ دهید. ۷) عدم هماهنگی بین تیم‌های فنی و تجاری

بسیاری از پروژه‌های یادگیری ماشین بدون درک روشن تیم فنی از اهداف تجاری آن، شروع می‌شوند. تیم یادگیری ماشین معمولاً، مسئله را به یک سوال دسته‌بندی یا رگرسیون تقلیل می‌دهند که باید دقت بالایی (خطای کمی) برای یک هدف فنی دیگر کسب کند. بدون تعامل با تیم تجاری برای کسب آگاهی از اهداف تجاری مانند افزایش نرخ کلیک یا حفظ کاربر که آن‌ها برای این پروژه تصور کرده‌اند، خروجی کار توسط تیم فنی، احتمالاً مورد قبول تیم تجاری قرار نمی‌گیرد. در چنین شرایطی، پروژه‌ها به دلیل صرف زمان و منابع به

پایان می‌رسند ولی تیم تجاری، از نتیجه‌ی کار راضی نیست.

فصل ۳

آماده‌سازی داده

۱.۳ اهداف فصل

[متن شما]

۲.۳ سوالاتی درباره داده

[متن شما]

۳.۳ چالش‌های داده

[متن شما]

۴.۳ ویژگی‌های مجموعه‌داده‌ی خوب

[متن شما]

۵.۳ تقسیم‌بندی مجموعه‌داده

[متن شما]

۶.۳ داده‌های پرت

[متن شما]

۷.۳ مقادیر گم‌شده

[متن شما]

۸.۳ مجموعه‌داده نامتوازن

[متن شما]

فصل ۴

مهندسی ویژگی

۱.۴ اهمیت

[متن شما]

۲.۴ مشخصات ویژگی خوب

[متن شما]

۳.۴ ویژگی‌های دسته‌ای

[متن شما]

۴.۴ مقادیر گم‌شده در ویژگی‌های دسته‌ای

[متن شما]

۵.۴ ویژگی‌های تقویمی

[متن شما]

۶.۴ سنتز ویژگی

[متن شما]

۷.۴ تغییر مقیاس ویژگی

[متن شما]

۸.۴ نشت داده

[متن شما]

۹.۴ فوت و فن‌های مهندسی ویژگی

[متن شما]

۱۰.۴ کاهش ابعاد

[متن شما]

۱۱.۴ انتخاب ویژگی

[متن شما]

۱۲.۴ خط لوله

[متن شما]

فصل ۵

رگرسیون

۱.۵ اهداف فصل

[متن شما]

۲.۵ مقدمه

[متن شما]

۳.۵ مدل چیست؟

[متن شما]

۴.۵ تخمین، تابع هزینه و بهینه‌سازی

[متن شما]

۵.۵ رگرسیون خطی

[متن شما]

۶.۵ ارزیابی

[متن شما]

۷.۵ رگرسیون چندجمله‌ای

[متن شما]

۸.۵ عمومیت

[متن شما]

۹.۵ رگولاریزیشن

[متن شما]

فصل ۶

دسته‌بندی

۱.۶ مقدمه

[متن شما]

۲.۶ رگرسیون لجستیک

[متن شما]

۳.۶ ارزیابی - قسمت اول

[متن شما]

۴.۶ ارزیابی - قسمت دوم

[متن شما]

۵.۶ کراس ولیدیشن

[متن شما]

۶.۶ نزدیک‌ترین-k همسایه

[متن شما]

۷.۶ بیز ساده‌لوحانه

[متن شما]

۸.۶ ماشین بردار پشتیبان

[متن شما]

۹.۶ هایپرپارامترها

[متن شما]

۱۰.۶ آشنایی با کتابخانه‌ی O2H

[متن شما]

۱۱.۶ درخت تصمیم

[متن شما]

۱۲.۶ فوت و فن درخت تصمیم

[متن شما]

۱۳.۶ بیش‌برازش درخت تصمیم

[متن شما]

فصل ۷

یادگیری تجمعی

۱.۷ اهداف فصل

[متن شما]

۲.۷ مقدمه

[متن شما]

۳.۷ جنگل تصادفی

[متن شما]

۴.۷ الگوریتم AdaBoost

[متن شما]

۵.۷ الگوریتم Boosting Gradient

[متن شما]

۶.۷ الگوریتم XGboost

[متن شما]

۷.۷ روش Stacking

[متن شما]

فصل ۸

پروژه اول

۱.۸ مقدمه

[متن شما]

۲.۸ یادداشت‌ها و راه‌حل

[متن شما]

فصل ۹

شبکه عصبی

۱.۹ اهداف فصل

[متن شما]

۲.۹ پرسپترون

[متن شما]

۳.۹ آموزش پرسپترون

[متن شما]

۴.۹ پرسپترون چندلایه

[متن شما]

۵.۹ عمومیت

[متن شما]

فصل ۱۰

یادگیری نظارت‌نشده

۱.۱۰ مقدمه

[متن شما]

۲.۱۰ الگوریتم PCA

[متن شما]

۳.۱۰ الگوریتم t-SNE

[متن شما]

۴.۱۰ خوشه‌بندی با k-means

[متن شما]

۵.۱۰ خوشه‌بندی با k-modes

[متن شما]

۶.۱۰ خوشه‌بندی با k-prototype

[متن شما]

فصل ۱۱

پروژه دوم

۱.۱۱ اهداف فصل

[متن شما]

۲.۱۱ تعبیه‌ی متن

[متن شما]

۳.۱۱ فاصله‌ی ویرایش

[متن شما]

۴.۱۱ معیار شباهت RBO

[متن شما]

فصل ۱۲

بیشتر بدانید

۱.۱۲ نمونه‌کاهی با NearMiss

[متن شما]

۲.۱۲ نمونه‌افزایی با SMOTE

[متن شما]

۳.۱۲ درخت رگرسیون

[متن شما]