

جزوه 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

فهرست مطالب

ت	پیشگفتار
۱	۱ مقدمه
۱	۱.۱ سلام!
۲	۲.۱ قالب کالج
۳	۳.۱ سیستم امتیازات و دریافت گواهی اصلی
۴	۴.۱ یادگیری ماشین چیست؟
۱۱	۵.۱ چرا پایتون؟
۱۳	۶.۱ آماده‌سازی محیط کار
۱۳	۷.۱ اجرای نوت‌بوک‌ها
۱۳	۸.۱ گوگل کولب
۱۳	۹.۱ معرفی مجموعه‌داده
۱۴	۲ مدیریت پروژه
۱۴	۱.۲ اهداف فصل
۱۵	۲.۲ چرخه پروژه
۱۵	۳.۲ اولویت‌بندی
۱۵	۴.۲ سازماندهی تیم
۱۵	۵.۲ چرا پروژه‌ها شکست می‌خورند؟
۱۶	۳ آماده‌سازی داده
۱۶	۱.۳ اهداف فصل
۱۶	۲.۳ سوالاتی درباره داده
۱۶	۳.۳ چالش‌های داده
۱۶	۴.۳ ویژگی‌های مجموعه‌داده‌ی خوب
۱۶	۵.۳ تقسیم‌بندی مجموعه‌داده
۱۷	۶.۳ داده‌های پرت
۱۷	۷.۳ مقادیر گم‌شده
۱۷	۸.۳ مجموعه‌داده نامتوازن
۱۸	۴ مهندسی ویژگی
۱۸	۱.۴ اهمیت
۱۸	۲.۴ مشخصات ویژگی خوب

۱۸	ویژگی‌های دسته‌ای	۳.۴
۱۸	مقادیر گم شده در ویژگی‌های دسته‌ای	۴.۴
۱۸	ویژگی‌های تقویمی	۵.۴
۱۸	سنتز ویژگی	۶.۴
۱۹	تغییر مقیاس ویژگی	۷.۴
۱۹	نشت داده	۸.۴
۱۹	فوت و فن‌های مهندسی ویژگی	۹.۴
۱۹	کاهش ابعاد	۱۰.۴
۱۹	انتخاب ویژگی	۱۱.۴
۱۹	خط لوله	۱۲.۴
۲۰	رگرسیون	۵
۲۰	اهداف فصل	۱.۵
۲۰	مقدمه	۲.۵
۲۰	مدل چیست؟	۳.۵
۲۰	تخمین، تابع هزینه و بهینه‌سازی	۴.۵
۲۰	رگرسیون خطی	۵.۵
۲۰	ارزیابی	۶.۵
۲۱	رگرسیون چندجمله‌ای	۷.۵
۲۱	عمومیت	۸.۵
۲۱	رگولاریزیشن	۹.۵
۲۲	دسته‌بندی	۶
۲۲	مقدمه	۱.۶
۲۲	رگرسیون لجستیک	۲.۶
۲۲	ارزیابی - قسمت اول	۳.۶
۲۲	ارزیابی - قسمت دوم	۴.۶
۲۲	کراس ولیدیشن	۵.۶
۲۲	نزدیکترین- k همسایه	۶.۶
۲۳	بیز ساده‌لوحانه	۷.۶
۲۳	ماشین بردار پشتیبان	۸.۶
۲۳	هایپرپارامترها	۹.۶
۲۳	آشنایی با کتابخانه $\text{O}2\text{H}$	۱۰.۶
۲۳	درخت تصمیم	۱۱.۶
۲۳	فوت و فن درخت تصمیم	۱۲.۶
۲۳	بیش‌برازش درخت تصمیم	۱۳.۶
۲۴	یادگیری تجمعی	۷
۲۴	اهداف فصل	۱.۷
۲۴	مقدمه	۲.۷
۲۴	جنگل تصادف	۳.۷

۲۴	الگوریتم AdaBoost	۴.۷
۲۴	الگوریتم Boosting Gradient	۵.۷
۲۴	الگوریتم XGboost	۶.۷
۲۵	روش Stacking	۷.۷
۲۶		پروژه اول	۸
۲۶	مقدمه	۱.۸
۲۶	یادداشت‌ها و راه حل	۲.۸
۲۷		شبکه عصبی	۹
۲۷	اهداف فصل	۱.۹
۲۷	پرسپیترون	۲.۹
۲۷	آموزش پرسپیترون	۳.۹
۲۷	پرسپیترون چندلایه	۴.۹
۲۷	عمومیت	۵.۹
۲۸		یادگیری ناظارت‌نشده	۱۰
۲۸	مقدمه	۱.۱۰
۲۸	PCA	۲.۱۰
۲۸	الگوریتم t-SNE	۳.۱۰
۲۸	k-means خوشبندی با	۴.۱۰
۲۸	k-modes خوشبندی با	۵.۱۰
۲۸	k-prototype خوشبندی با	۶.۱۰
۲۹		پروژه دوم	۱۱
۲۹	اهداف فصل	۱.۱۱
۲۹	تعابیری متن	۲.۱۱
۲۹	فاصله‌ی ویرایش	۳.۱۱
۲۹	RBO معیار شباهت	۴.۱۱
۳۰		بیشتر بدانید	۱۲
۳۰	NearMiss نمونه‌کاهی با	۱.۱۲
۳۰	SMOTE نمونه‌افزایی با	۲.۱۲
۳۰	درخت رگرسیون	۳.۱۲
۳۱		واژه‌نامه	
۳۲		منابع	

پیشگفتار

اینجا هدف، دامنه، و نحوه استفاده از جزو 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 می‌شویم، 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 انتهای پس از مطالعه درسنامه‌ها و حل تمرین‌ها، به «دانشمند داده» و «مهندس یادگیری ماشین» تبدیل می‌شوید که قطعاً راه طولانی اما شیرینی برای حرفه‌ای شدن پیش رو دارید!

البته برای حفظ تناسب کالج و افزایش احتمال یادگیری عمیق، 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، پرتلash و ایرانی پیشرفته...



۲.۱ قالب کالج

قالب کالج روند آموزشی کوئرا کالج

امروزه با فرآگیر 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 می‌شود بازخورد مناسبی توسط سیستم خروجی داده شود.

در تمرین‌هایی که عملکرد مدل یادگیری ماشین شما سنجیده می‌شود یک حد آستانه (Threshold) تعريف شده و در صورتی که مدل شما عملکردی بهتر از آن حد داشته باشد تمرین با موفقیت گذرانده می‌شود. البته اگر عملکرد مدل شما از مقدار خواسته شده بهتر باشد امتیاز اضافه‌تری کسب خواهد 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 راه حل تمرین‌های دوره را دریافت کنید. پس از پایان دوره، بر اساس امتیاز کسب شده توسط شما، گواهی صادر خواهد شد. پس سعی کنید از امتیازات خود به نحو مناسبی استفاده کنید! گذراندن فصل‌ها

دوره‌ی «یادگیری ماشین ۲ اجامپ 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 بر حسب 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 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 به صورت 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 مجموعه داده 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 Mitchell) در کتاب یادگیری ماشین خود، یادگیری ماشین را از دید مهندسی به این شکل تعریف کرده است: «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 مجموعه داده یا نمونه‌ها استفاده

کرد. معیار عملکرد (Performance)

هر مدل یادگیری ماشینی که طراحی کنیم، همواره به طور 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 به طور 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 ناظارت شده Learning) (Supervised

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 بدھیم 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 ناظارت شده Learning) (Supervised 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.

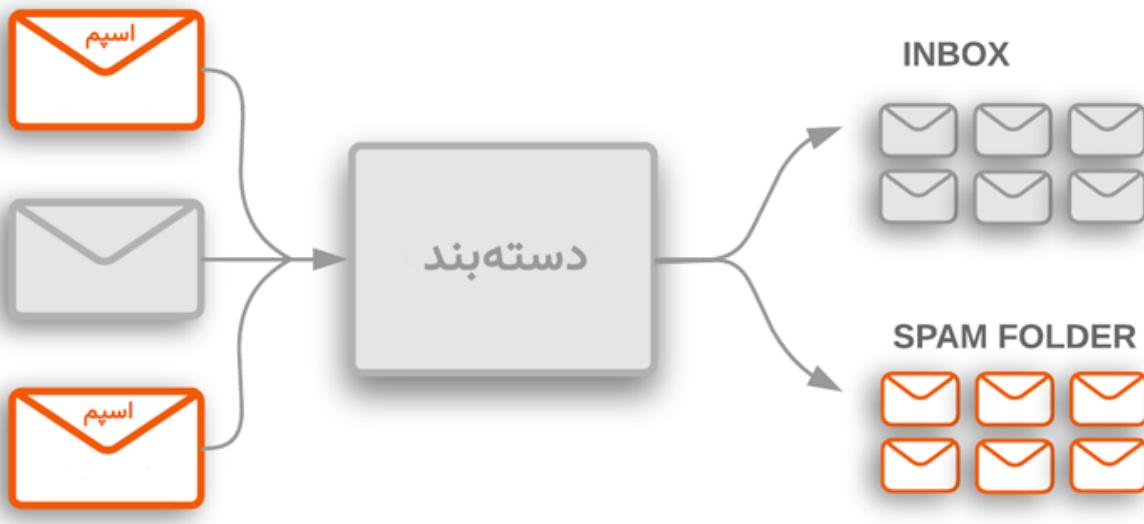
به عنوان مثال، جدول زیر را در نظر داشته باشید. این جدول اطلاعاتی از خانه‌های شهر پکن به ما می‌دهد.

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 ناظارت شده را می‌توان به دو بخش دسته‌بندی (Classification) و رگرسیون-Regression تقسیم کرد که در ادامه به معرفی بیشتر هر کدام 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. به عنوان مثال، دسته‌بندی ایمیل‌ها به دو دسته‌ی اسپم (Spam) و غیر اسپم است. 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.

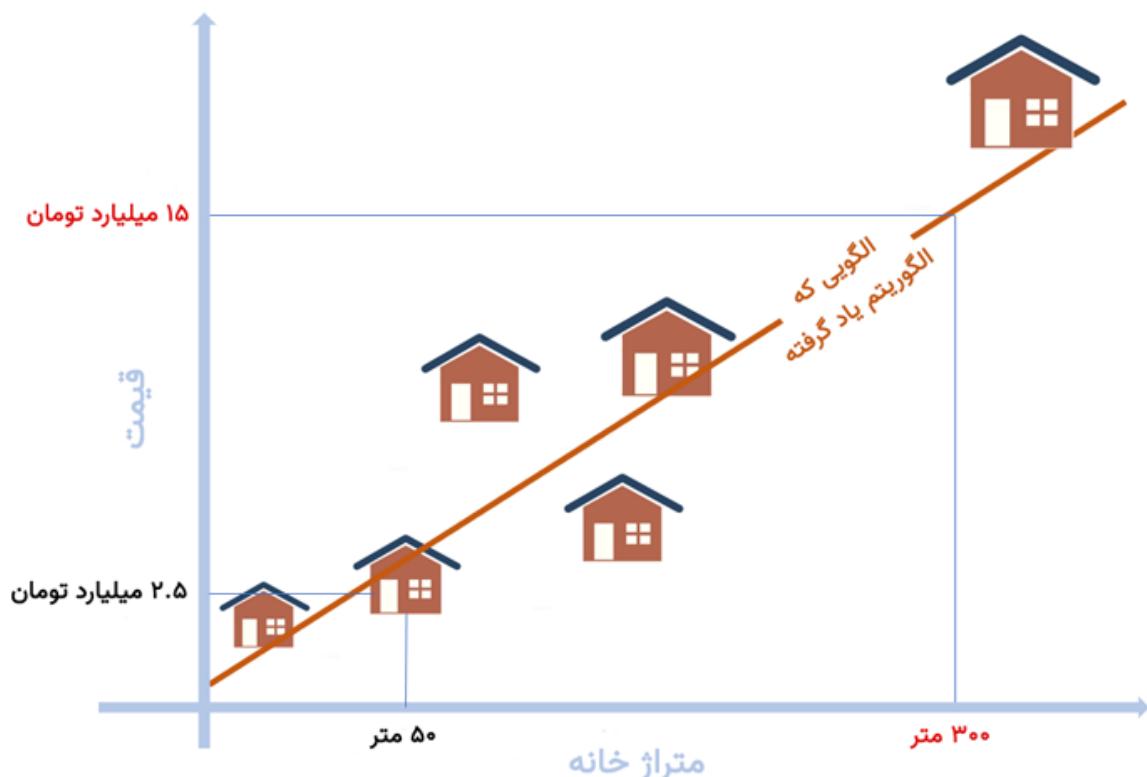


رگرسیون (Regression)

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.

به طور مثال، برای پیش‌بینی قیمت خانه، می‌توان از اطلاعات خانه‌های دیگر برای تخمین قیمت یک خانه استفاده کرد. ویژگی‌هایی مانند مترأژ، 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 مترأژ خانه‌ها برای ساخت مدل یادگیری ماشین استفاده شده است. هر نقطه یک خانه را نشان می‌دهد. برای مثال، خانه‌ی ۵۰ متری، ۵۰ میلیارد تومان ارزش دارد. از نظر مدل ما، خانه‌ی ۳۰۰ متری که قیمت آن مشخص نیست، ۱۵ میلیارد تومان ارزش دارد.



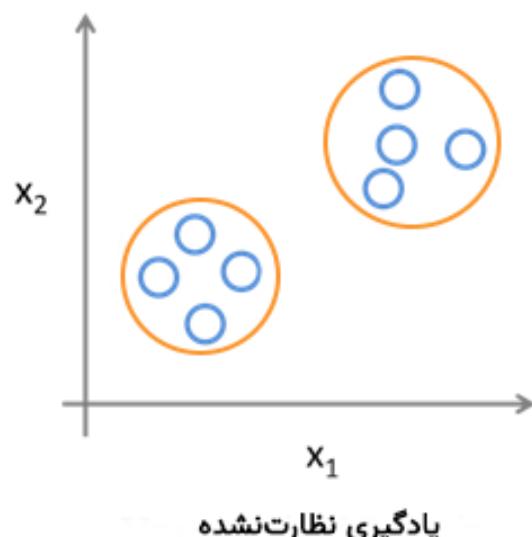
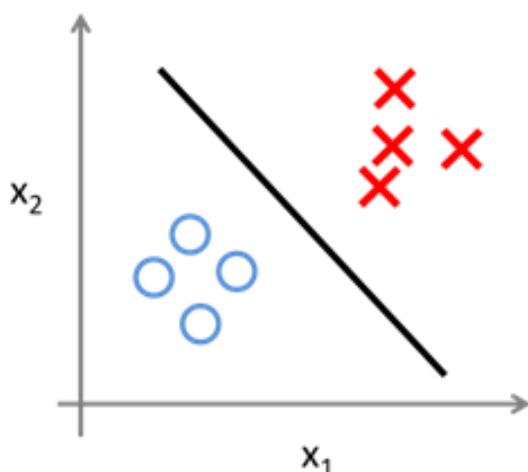
تفاوت دسته‌بندی و رگرسیون

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 ناظارت‌نشده (Unsupervised Learning)

تفاوت یادگیری ناظارت‌نشده (Unsupervised Learning) با یادگیری ناظارت‌شده، در نبودن برچسب‌ها است. به عبارت دیگر، هیچ ناظری (برچسب) به کامپیوتر نمی‌گوید که چه زمانی درست پیش‌بینی کرده و چه زمانی مرتكب خطا شده است. در این رویکرد یادگیری، مدل به‌نهایی و بدون کمک برچسب‌هایی که در روش ناظارت‌شده 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 ناظارت‌نشده 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.



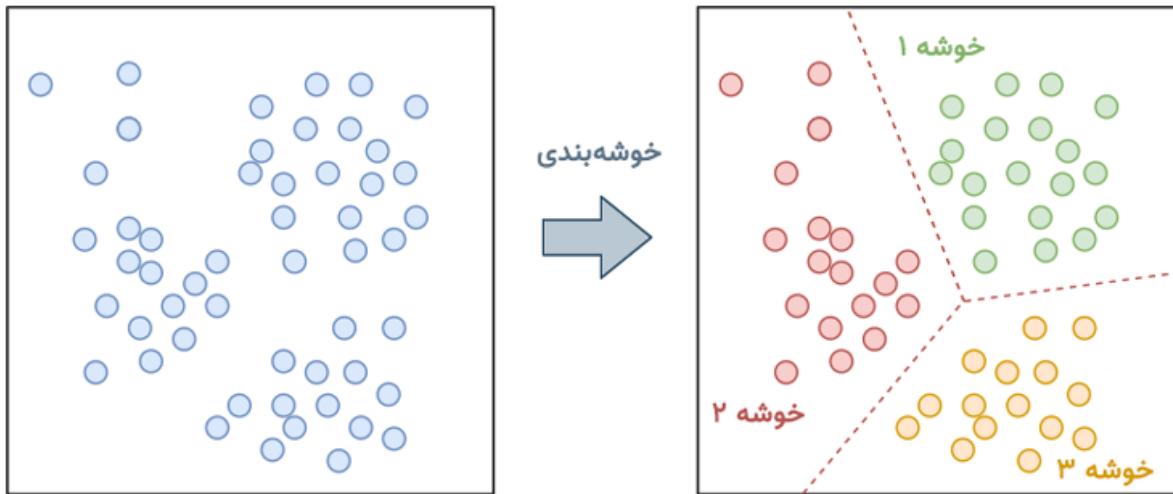
در تصویر بالا سمت چپ دو نوع داده وجود دارد، دادهایی با علامت ضربدر قرمز و دادهایی با دایره‌های آبی. دایره‌ی آبی و ضربدر قرمز همان برچسب‌ها هستند. این تصویر در واقع همان مسئله‌ی دسته‌بندی (نوعی از یادگیری نظارت شده) را نشان می‌دهد و وظیفه‌ی مدل پیدا کردن خط سیاه رنگ است که به کمک آن بتوان دسته‌ی «دایره‌های 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 نظارت نشده، 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 مجموعه داده‌ی 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 Extraction) 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 معماهی 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. زمانی که 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.

روندي 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، 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 خطمسی (Policy) خود را تغییر دهد.

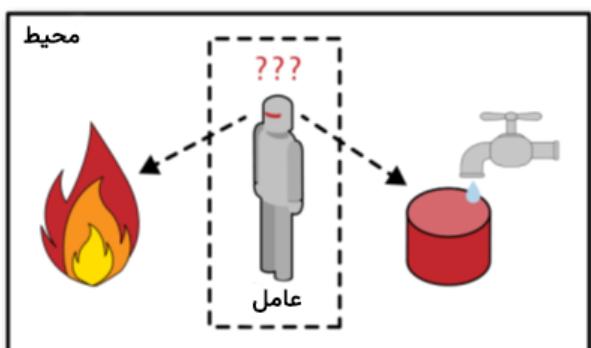
توضیح اضافه

در مثال هزارتو:

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

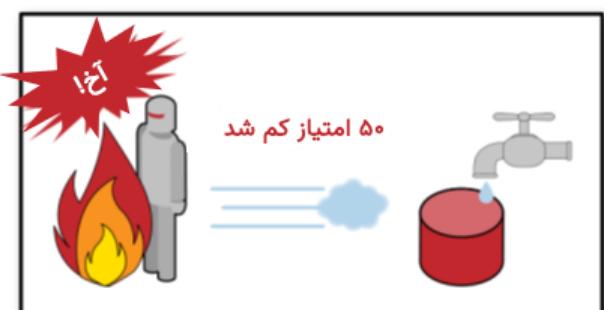
مکان فعلی شما در هزارتو و اطلاعات بیشتری در مورد محیط که در آن به یادگیری می‌پردازید را نشان می‌دهد. حرکت یا عمل (Action) گامی است که در یک جهت بر می‌دارید. بازخورد (Reward) (امتیاز مثبت یا منفی‌ای است که دریافت می‌کنید تا بفهمید آیا در مسیر درستی قرار دارید یا نه! خطمشی (Policy) مشخص می‌کند که در هر 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 TensorFlow and Keras Scikit-Learn, with Learning Machine On Hands 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.



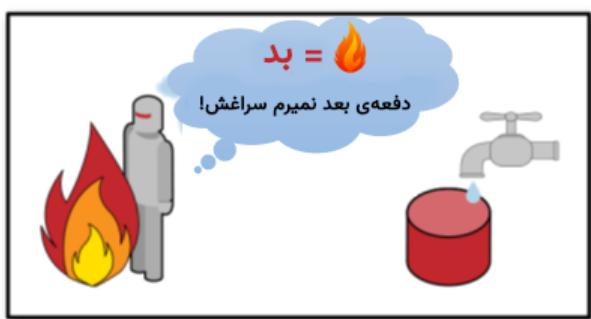
۱. مشاهده

۲. انتخاب حرکت با استفاده از خطمشی



۳. انجام حرکت!

۴. دریافت پاداش یا جزا



۵. بهروزرسانی خطمشی (مرحله‌ی یادگیری)

۶. تکرار تا وقتی که 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.

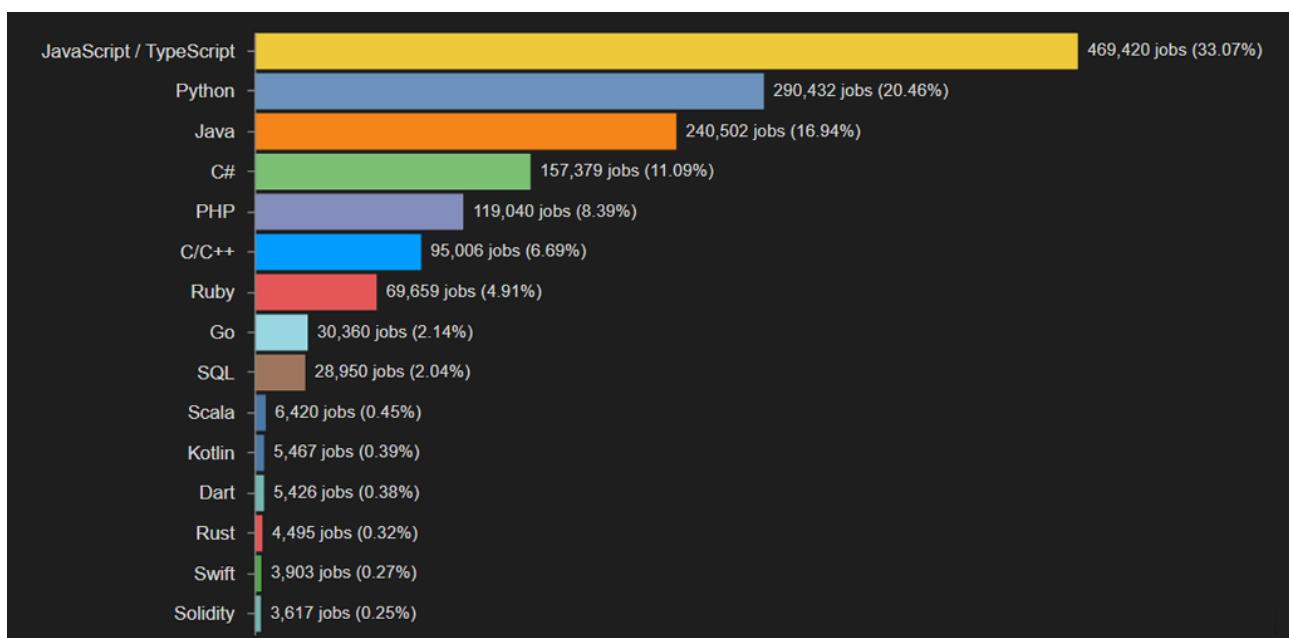
۵.۱ چرا پایتون؟

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 می‌کردند. 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 R MATLAB 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

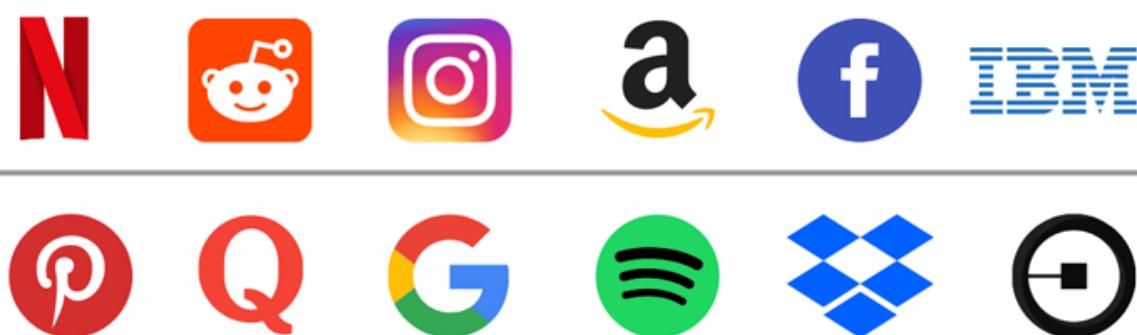
در پایتون کتابخانه‌های زیادی برای بارگذاری و دستکاری داده، مدل‌سازی، فعالیت‌های آماری، پردازش تصویر و پردازش متن وجود دارند. هر کدام از این کتابخانه‌ها، توابع زیادی در اختیار توسعه‌دهندگان قرار می‌دهند تا بتوانند به راحتی و در حالت بهینه، برنامه خود را توسعه دهند. به عنوان مثال Numpy برای کار با آرایه‌ها 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 مجددًا 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 علل 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 نحوه 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 نحوه 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 دوستان تان 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 به جای خوش‌گذرانی تبدیل می‌شود به یک خاطره‌ی تلخ و خنده‌دار.

مدیریت پروژه هم درست همین طور است. اگر بدون برنامه ریزی، 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.

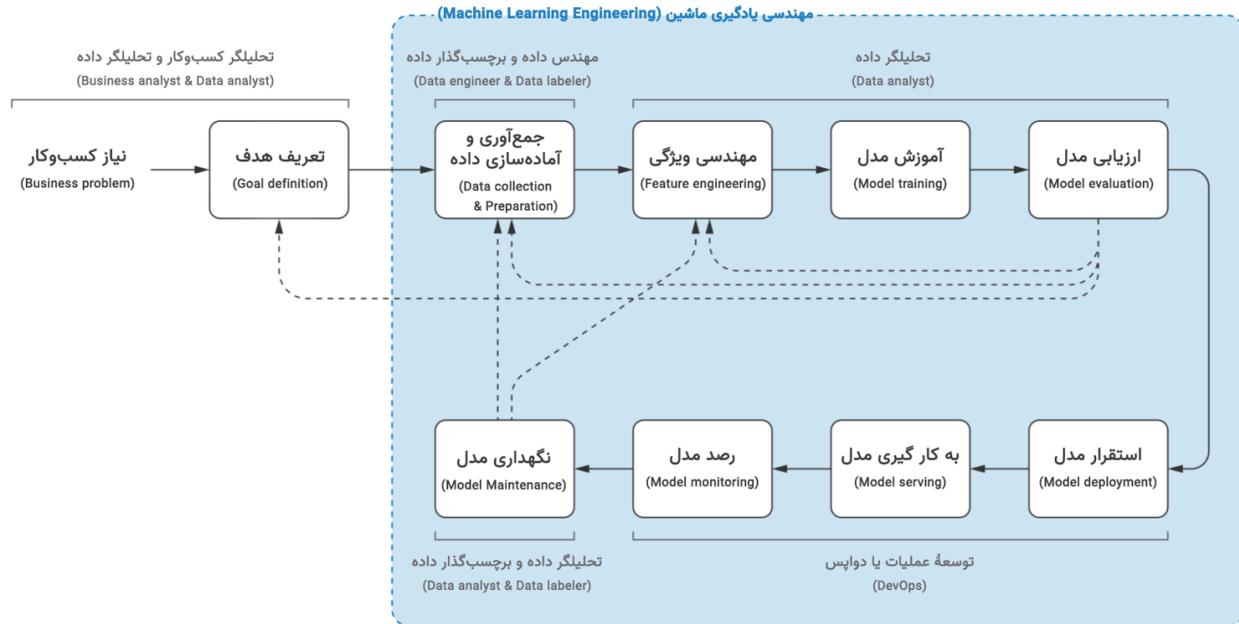
پروژه‌های یادگیری ماشین به خاطر یک نیاز کسب‌وکار شروع می‌شوند و این نیاز، هدف پروژه را تعریف می‌کند. متناسب با این هدف، تیم سازماندهی می‌شود و کار پروژه شروع می‌شود.

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.

به صورت 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 به صورت 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 قادر 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 قابل قبولی است. در صورتی که مدل، حداقل عملکرد قابل قبول 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

[متن شما]

۴.۲ سازماندهی تیم

[متن شما]

۵.۲ چرا پروژه‌ها شکست می‌خورند؟

[متن شما]

فصل ۳

آماده‌سازی داده

۱.۳ اهداف فصل

[]

۲.۳ سوالاتی درباره داده

[]

۳.۳ چالش‌های داده

[]

۴.۳ ویژگی‌های مجموعه‌داده‌ی خوب

[]

۵.۳ تقسیم‌بندی مجموعه‌داده

[]

۶.۳ داده‌های پرت

] [

۷.۳ مقادیر گم شده

] [

۸.۳ مجموعه داده نامتوازن

] [

فصل ۴

مهندسی ویژگی

۱.۴ اهمیت

] [

۲.۴ مشخصات ویژگی خوب

] [

۳.۴ ویژگی‌های دسته‌ای

] [

۴.۴ مقادیر گم شده در ویژگی‌های دسته‌ای

] [

۵.۴ ویژگی‌های تقویمی

] [

۶.۴ سنتز ویژگی

] [

۷.۴ تغییر مقیاس ویژگی

] [

۸.۴ نشت داده

] [

۹.۴ فوت و فن‌های مهندسی ویژگی

] [

۱۰.۴ کاهش ابعاد

] [

۱۱.۴ انتخاب ویژگی

] [

۱۲.۴ خط لوله

] [

فصل ۵

رگرسیون

۱.۵ اهداف فصل

] [

۲.۵ مقدمه

] [

۳.۵ مدل چیست؟

] [

۴.۵ تخمین، تابع هزینه و بهینه‌سازی

] [

۵.۵ رگرسیون خطی

] [

۶.۵ ارزیابی

] [

۷.۵ رگرسیون چندجمله‌ای

] [

۸.۵ عمومیت

] [

۹.۵ رگولاریزیشن

] [

فصل ۶

دسته‌بندی

۱.۶ مقدمه

] [

۲.۶ رگرسیون لجستیک

] [

۳.۶ ارزیابی - قسمت اول

] [

۴.۶ ارزیابی - قسمت دوم

] [

۵.۶ کراس ولیدیشن

] [

۶.۶ نزدیک‌ترین- k -همسایه

] [

۷.۶ بیز ساده لوحانه

] [

۸.۶ ماشین بردار پشتیبان

] [

۹.۶ هایپرپارامترها

] [

۱۰.۶ آشنایی با کتابخانهی $\text{O}2\text{H}$

] [

۱۱.۶ درخت تصمیم

] [

۱۲.۶ فوت و فن درخت تصمیم

] [

۱۳.۶ بیشبرازش درخت تصمیم

] [

فصل ٧

يادگیری تجمعی

١.٧ اهداف فصل

] [

٢.٧ مقدمه

] [

٣.٧ جنگل تصادفی

] [

٤.٧ الگوريتم AdaBoost

] [

٥.٧ الگوريتم Boosting Gradient

] [

٦.٧ الگوريتم XGboost

] [

Stacking روش ۷.۷

]

فصل ۸

پروژه اول

۱.۸ مقدمه

] [

۲.۸ یادداشت‌ها و راه حل

] [

فصل ۹

شبکه عصبی

۱.۹ اهداف فصل

] [

۲.۹ پرسپترون

] [

۳.۹ آموزش پرسپترون

] [

۴.۹ پرسپترون چندلایه

] [

۵.۹ عمومیت

] [

فصل ۱۰

یادگیری ناظارت‌نشده

۱.۱۰ مقدمه

] [

۲.۱۰ الگوریتم PCA

] [

۳.۱۰ الگوریتم t-SNE

] [

۴.۱۰ خوشبندی با k-means

] [

۵.۱۰ خوشبندی با k-modes

] [

۶.۱۰ خوشبندی با k-prototype

] [

فصل ۱۱

پروژه دوم

۱.۱۱ اهداف فصل

] [

۲.۱۱ تعبیهی متن

] [

۳.۱۱ فاصلهی ویرایش

] [

۴.۱۱ معیار شبه است RBO

] [

فصل ۱۲

بیشتر بدانید

۱.۱۲ نمونه کاهی با NearMiss

] [

۲.۱۲ نمونه افزایی با SMOTE

] [

۳.۱۲ درخت رگرسیون

] [

واژه‌نامه

() .

منابع

() . / / /