یادگیری ماشین

مؤلف:

مهندس سهیل کیا

انتشارات ملائک

سرشناسه	:	کیا، سهیل
عنوان و نام پدیدآور	:	یادگیری ماشین/ مولف سهیل کیا.
مشخصات نشر	:	تهران: ملائک، ۱۳۹۷.
مشخصات ظاهری	:	۱۶۵ ص.؛مصور، نمودار.
شابک	:	۹۷۸-۶۰۰-۷۹۵۸-۴۱-۴
وضعیت فهرست نویسی	:	فیپا
موضوع	:	فراگیری ماشینی
موضوع	:	Machine Learning
رده بندی کنگره	:	QC۳۲۵/۵/ک۹ی۲ ۱۳۹۷
رده بندی دیویی	:	۰۰۶/۳۱
شماره کتابشناسی ملی	:	۵۳۷۵۸۵۵

انتشارات ملائک

نام کتاب : یادگیری ماشین
مولف : سهیل کیا
ویراستار فنی : بهروز آذرخلیلی
ناشر : انتشارات ملائک
نوبت چاپ : اول / ۱۳۹۷
شمارگان : ۵۰۰ جلد
قیمت : ۳۷۰۰۰ تومان
شابک :۹۷۸-۶۰۰-۷۹۵۸-۴۱-۴

تقدیم به

بهروز آذر خلیلی

که صبورانه

فانوس به دست

روشنگر راهم بود.

فهرست

پیش گفتار ۷

فصل صفر: سرآغاز ۹

۰-۱ مقدمه ۱۱

۰-۲ یادگیری ماشین چگونه کار می‌کند؟ ۱۲

۰-۳ آماده سازی محیط ۱۵

۰-۴ جمع بندی ۱۶

فصل ۱: Regression ۱۷

۱-۱ مقدمه ۱۹

۱-۲ Linear Regression ۲۱

۱-۳ Linear Regression با مرتبه های بالاتر ۲۶

۱-۴ کار با داده‌های واقعی ۲۹

۱-۵ ورودی های n-بعدی ۳۵

۱-۶ Ridge، Lasso و Elastic Net ۳۶

۱-۷ داده‌های پرت (Outliers) ۴۳

۱-۸ Overfitting و Underfitting ۴۴

۱-۹ جمع بندی ۴۶

فصل ۲ : Classification ۴۹

۲-۱ مقدمه ۵۱

۲-۲ Logistic Regression یا Binary Classification ۵۳

۲-۳ Softmax Regression ۵۶

۲-۴ Support Vector Machines (SVM) ۶۱

۲-۵ درخت تصمیم (Decision Tree) ۶۸

۲-۶ KNN ۷۴

۲-۷ Naïve Bayes ۷۷

۲-۸ جمع بندی ۸۲

فصل ۳ : Ensembling ۸۳
۳-۱ مقدمه ۸۵
۳-۲ Ensemble خود را بسازید (Stacking) ۸۸
۳-۳ Bagging ۹۲
۳-۴ Boosting ۹۶
۳-۵ Random Forest ۹۸
۳-۶ Ensemble Regressors ۱۰۰
۳-۷ جمع بندی ۱۰۲
فصل ۴ : Clustering ۱۰۳
۴-۱ مقدمه ۱۰۵
۴-۲ K-Means ۱۰۷
۴-۳ Mean-Shift Clustering ۱۱۹
۴-۴ درخت خوشه‌بندی ۱۲۳
۴-۵ DBSCAN ۱۲۸
۴-۶ GMM و EM ۱۳۱
۴-۷ BIRCH ۱۳۶
۴-۸ جمع بندی ۱۳۸
فصل ۵ : Data Preprocessing ۱۳۹
۵-۱ مقدمه ۱۴۱
۵-۲ بررسی مجموعه داده ۱۴۱
۵-۳ داده‌های پرت (Outliers) ۱۴۷
۵-۴ مقادیر متنی ۱۵۰
۵-۵ استانداردسازی داده‌ها ۱۵۶
۵-۶ کاهش ابعاد ۱۵۷
۵-۷ جمع بندی ۱۶۲
کلام آخر ۱۶۳
منابع ۱۶۵

پیش گفتار

برای کسانی که اخبار را دنبال می‌کنند، شنیدن خبری تازه در حوزه هوش مصنوعی، دیگر چیز تازه‌ای به نظر نمی‌رسد. شرکت‌های بزرگ با شناخت پتانسیل موجود و سرمایه‌گذاری‌های کلان، فرصت رشد روزافزون را برای این حوزه فراهم آورده‌اند. دستیار صوتی گوشی همراه، موتورهای جستجو که تبلیغاتی مناسب سلیقه شما ارائه می‌کنند، سامانه‌های تشخیص جرم، اتومبیل‌های بدون راننده و ... همه و همه به پشتوانه هوش مصنوعی در کنار ما حضور دارند و به ما یاری می‌رسانند. در این میان، شاخه‌ای از هوش مصنوعی به نام یادگیری ماشین توجه بسیار زیادی را به خود جلب کرده است.

هنگامی که کتاب تاریخچه زمان نوشته استیفن هاوکینگ را مطالعه می‌کردم، ایده‌ای جالب را در مقدمه آن یافتم. هاوکینگ ادعا کرده بود که هر فرمول ریاضی، تعداد خوانندگان کتاب را نصف خواهد کرد و از این رو در کل کتاب تنها یک فرمول وجود دارد که به جای اثبات تئوریک آن، با توصیف نتایجش به خوبی کل فیزیک نوین را برای خواننده تشریح می‌کند.

متاسفانه در زمان نگارش این کتاب، جامعه بشری وی را از دست داد، اما این ایده آنقدر جذاب بود که در طول این کتاب تلاش داشتم تا بر همین منوال حرکت کنم و تنها در صورت لزوم به بیان فرمول ها بپردازم.

اگر شما هم از آن دسته انسان‌هایی هستید که مفاهیم را با کمک مثال یاد می‌گیرید و همچنین درک شهودی از مفاهیم برایتان مهم‌تر از اثبات قضایای ریاضی است، این کتاب برای شماست.

اگر در این زمینه تازه کار هم هستید همچنان این کتاب برای شما نگارش شده است، زیرا تئوری‌های ریاضی پشت مفاهیم بسیار چالش‌برانگیز است. در این کتاب می‌آموزید چگونه با کمک ابزاری به نام یادگیری ماشین، ایده‌هایی که دارید را پیاده‌سازی کنید و نتیجه آن را ببینید.

درواقع، هدف از نگارش این کتاب، تشریح تئوری‌های پیچیده که در پس پرده مفاهیم یادگیری ماشین وجود دارند نیست. بلکه ارائه شهودی مفاهیم در راستای درک نحوه کارکرد الگوریتم‌ها است. برای خود من، غرق شدن در ریاضیات محض و هزارتوی معادلات و فرمول‌ها، زیبایی چندانی ندارد، اما درک و مشاهده این مفاهیم در دنیای واقعی، رمزگشای زیبایی‌های آن است.

حال که بر جنبه‌های عملی تاکید داریم، نیازمند محیطی برای پیاده‌سازی هستیم. برای این مهم از زبان Python استفاده شده است. این زبان با توجه به ساختار خود و همچنین کتابخانه نیرومندی که برای آن موجود است، توانست به سرعت راه خود را به صدر جدول پرطرفدارها باز کند. به خصوص در میان متخصصین داده. در Python نیاز ندارید که با مفاهیم پیچیده‌تر برنامه‌نویسی درگیر شوید و قادر خواهید بود یک راست سراغ اصل مطلب بروید. در همین راستا، در کدها اصل ساده نویسی برای درک بهتر خواننده رعایت شده است. قدر مسلم راه‌های دیگری نیز برای پیاده‌سازی وجود دارند که گاهی در دل نمونه کدهای مشابه، جهت آشنایی خواننده، تفاوت‌هایی این چنینی گنجانده شده است.

کلام آخر؛ سفر در مسیر علوم داده، برای من سفری لذت بخش و زیبا بوده و هست. مسیر روبه‌رو، با چالش‌های بزرگ و کوچکی همراه خواهد بود، اما آنچه در پایان انتظار شما را می‌کشد، ارزشش را خواهد داشت.

موفق باشید

فصل صفر:

سرآغاز

۰-۱ مقدمه

با تولد کامپیوترها و افزایش سرعت محاسبات، بشر آرزوی ساخت ماشینی را کرد که بتواند تمام مسائل را حل کند. و این جرقه‌ای شد برای تولد هوش مصنوعی. ماشینی هوشمند که خود بتواند محیط خود را درک کند، از آن بیاموزد و به آن پاسخ دهد. ابتدا تلاش شد تا با تدوین الگوریتم‌هایی همه‌جانبه به این آرزو جامه عمل پوشانده شود، اما مشکل در اینجا بود که چنین الگوریتمی پیدا نشد و صورت مسئله به این تغییر کرد که چگونه ماشین می‌تواند بدون در دست داشتن الگوریتمی همه جانبه پاسخی برای پرسش‌ها بیابد.

پرسشی مهم در اینجا وجود دارد. ما از چه می‌آموزیم؟ از گذشته‌مان. آنچه در گذشته اتفاق افتاده، راهنمایی است برای تصمیمات آینده. در دنیای دیجیتال، گذشته، همان اطلاعاتی است که جمع‌آوری کرده‌ایم تا پیش از این، وظیفه بررسی و تحلیل داده‌ها در حوزه علمی آمار قرار داشت. نکته در اینجا بود که متدهای آماری ساده، به سختی پاسخگوی تحلیل‌های پیچیده بودند.

با آمار می‌توان مسائل را به همان شکلی که هستند ببینیم. بدین منظور، علم آمار، ابزارهای مختلفی را در اختیار شما می‌گذارد که از این میان می‌توان به جداول، نمودارها و تصویرسازی اشاره کرد تا بتوانیم دنیای پیرامون خود را توصیف کنیم. در این بین تشخیص آنچه که در حال اتفاق افتادن است یک مقوله است و توصیف چرایی آن بحثی دیگر.

این مفهوم در آمار، همبستگی (Correlation) نامیده می‌شود. با نگاه کردن به یک نمونه آماری به تنهایی، نمی‌توان اطلاعات زیادی کسب کرد زیرا تاثیر عوامل مختلف بر یکدیگر را نادیده گرفته‌اید. و این معنای همبستگی است: تغییر در یک چیز، چه تاثیری بر دیگر چیزها دارد. و این‌گونه می‌توانید فرآیندهایی را که در دنیای پیرامون روی می‌دهند را درک کنید.

پیدا کردن همبستگی بین موارد مختلف، از دید محاسباتی کاری بسیار پیچیده و زمان‌بر است. در عین حال برای پیدا کردن این همبستگی‌ها و اثبات کارکردشان، نیازمند حجم زیادی از داده‌ها خواهید بود که جمع‌آوری آن‌ها نیز کار ساده‌ای نیست. از سوی دیگر هرچه میزان مشاهدات ثبت شده شما، یا همان داده‌ها افزایش یابد، فرآیند محاسبه پیچیده‌تر می‌گردد!

مشکل محاسبات، با پیشرفت کامپیوترها حل شد و ظهور اینترنت و انقلاب داده‌ای، خط پایانی بود بر مشکل جمع‌آوری داده‌ها. و با کنار هم قرار گرفتن آمار و توانایی محاسباتی بالا، یادگیری ماشین به معنای واقعی خودنمایی کرد. الگوریتم‌های یادگیری ماشین، با پیدا کردن همبستگی بین داده‌ها و الگوهای موجود در مجموعه داده، قادر هستند تا داده‌های خام را به دانش تبدیل کرده و پیش‌بینی‌هایی از آینده داشته باشند.

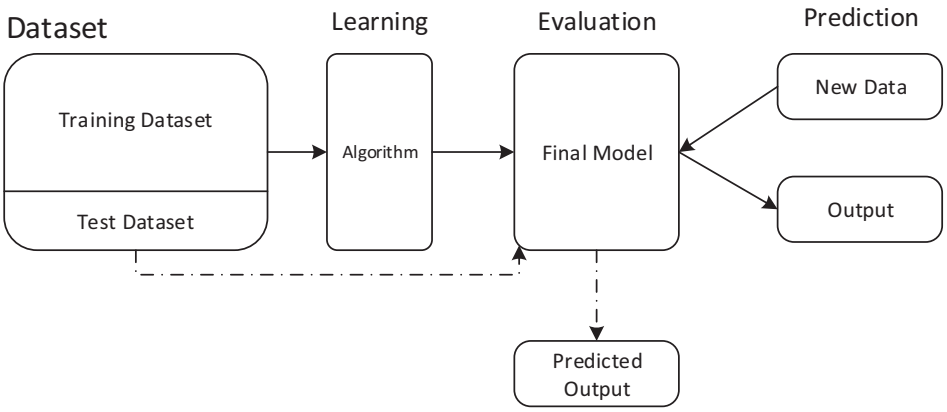
در این فصل، با مفاهیم ابتدایی یادگیری ماشین آشنا خواهید شد تا پایه و بنیان ادامه مسیر باشد. سپس محیط پیاده‌سازی را آماده خواهید کرد.

۲-۰ یادگیری ماشین چگونه کار می‌کند؟

به همراه یک کودک به یک میوه‌فروشی می‌روید تا نام میوه‌ها را به او بیاموزید. تمام میوه‌ها به صورت کاملا جدا از هم مرتب شده و حتی بالای هر قفسه نام میوه نیز نوشته شده است. به بیان دیگر، تمام میوه‌ها دارای برچسب (Label) نام هستند. این بدین معنی است که فردی از قبل نمونه‌های داده را دیده و دسته هر یک را مشخص کرده است. حتی اگر این‌گونه نیز نباشد، شما نام تمام میوه‌ها را می‌دانید و آن را در اختیار کودک قرار خواهید داد. حال می‌خواهید فرآیند آموزش را آغاز کنید و هدف شما این است که اگر در آینده کودک شما میوه جدیدی دید بتواند آن را به درستی تشخیص دهد. به او یک میوه می‌دهید و نام آن را به وی خواهید گفت. کودک تلاش می‌کند تا با کمک اطلاعاتی از قبیل شکل، رنگ، بو، نوع بافت و ... این ماهیت را درک کند.

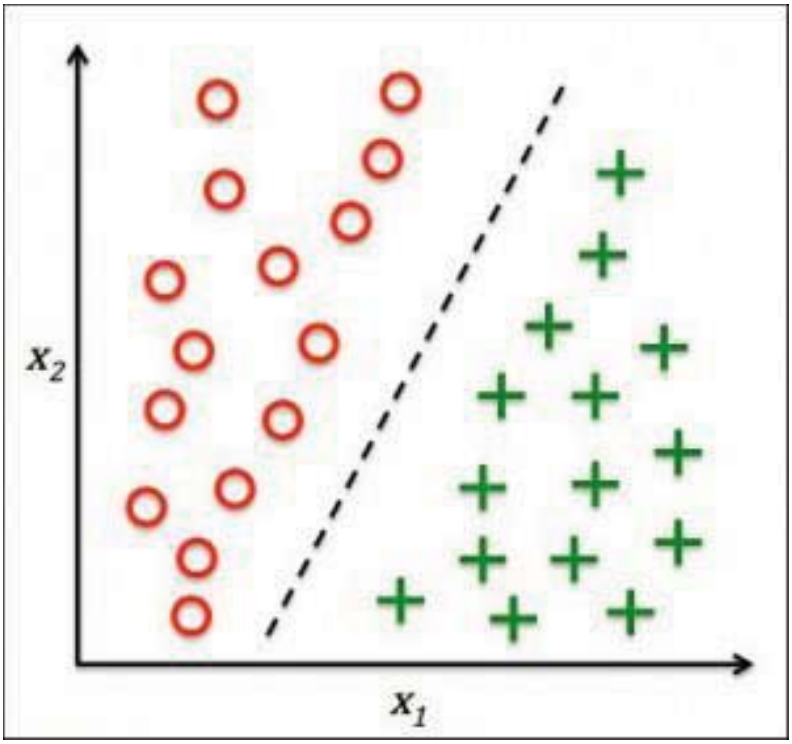
در این فرآیند، شما هم ورودی را در اختیار کودک می‌گذارید و هم خروجی مورد انتظار را به او می‌گویید. انتظار شما این است که پس از دیدن تعدادی از میوه‌ها، کودک بتواند میوه‌های جدیدی که تا به حال ندیده است را به درستی حدس بزند. پس کودک را در بوته آزمایش قرار می‌دهید و شما که از قبل به این مرحله نیز اندیشیده‌اید، تعدادی میوه را که از دید کودک پنهان کرده‌اید به او نشان داده و از او می‌خواهید که نام آن‌ها را بر زبان بیاورد.

فرآیندی که طی شد چیزی شبیه به شکل زیر است:

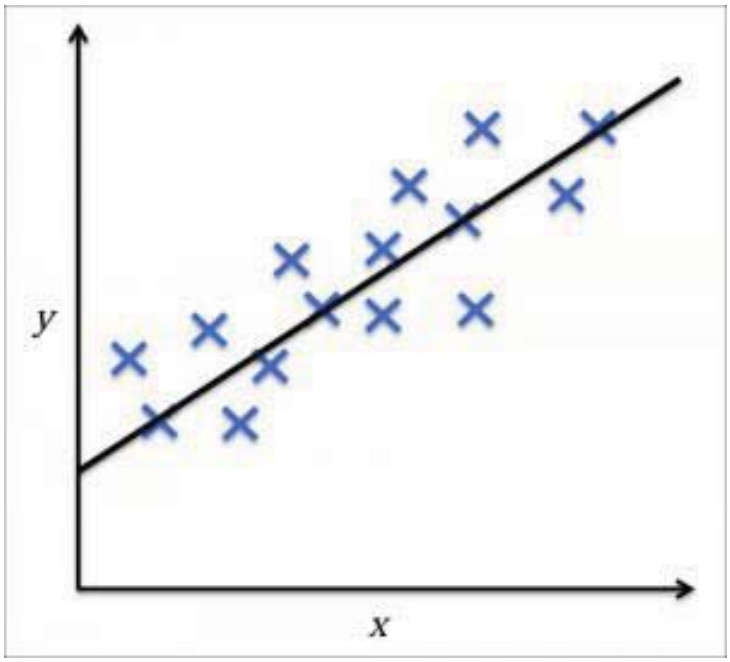


در سناریوی فوق، هر میوه یک نمونه داده است. این نمونه‌های داده تعدادی صفت خاصه دارند، مانند شکل، رنگ، بو، نوع بافت و ... نام هر میوه نیز، مقدار هدف و خروجی فرآیند است. میوه‌هایی که از ابتدا و با نام آن‌ها در اختیار کودک قرار داده‌اید، مجموعه داده آموزش است و میوه‌هایی که در آخر به عنوان امتحان به کودک دادید مجموعه آزمایش نام دارد.

تشخیص میزان یادگیری کودک در زمانی که او را امتحان می‌کردید نیز اعتبارسنجی یادگیری است و سرانجام به این دسته از فرآیندها در یادگیری ماشین، گروه‌بندی (Classification) گفته می‌شود.

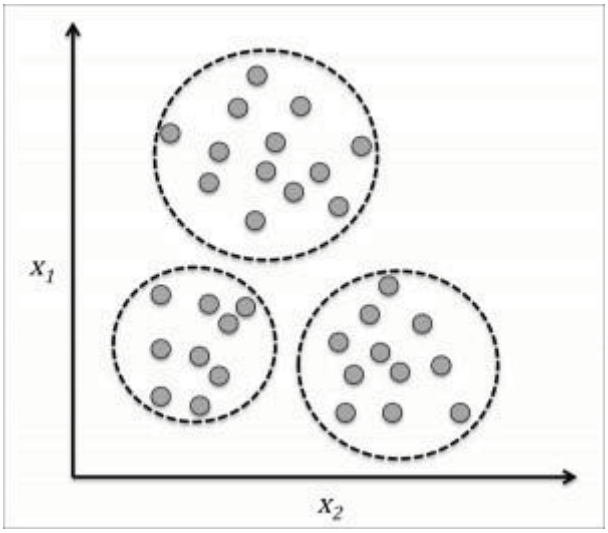


این همان چیزی است که در فصل دوم و سوم به آن خواهیم پرداخت. اما برخی از مواقع سوال به گونه ای است که خروجی نهایی مقادیر پیوسته است و نه گسسته. به عنوان مثال حدس زدن اینکه تغییرات قیمت در بورس چگونه خواهد بود. در این‌گونه موارد نیز خروجی مورد انتظار مشخص است و باز هم داده‌ها برچسب‌گذاری شده‌اند (اگر خروجی که از نمونه داده انتظار دارید مشخص شده باشد، داده‌ها دارای برچسب هستند). به این دسته از فرآیندها، Regression گفته می‌شود.



این دست فرآیندها را در فصل اول بررسی خواهیم کرد.

و سرانجام بعضی از مسائل صورت دیگری دارد. فرض کنید به یک کودک، مجموعه‌ای بزرگ از میوه‌ها را داده‌اید و از وی خواسته‌اید که آن‌ها را از هم جدا کند. از طرف دیگر، کودک نام میوه‌ها را نیز نمی‌داند. وی با توجه به ادراکی که از ظاهر میوه‌ها دارد، آن‌ها را در ظرف‌هایی که به وی داده‌اید خواهد چید. در این مسئله، داده‌های شما هیچ برچسبی نداشته است.



این دست از فرآیندها خوشه‌بندی نام داشته و در فصل چهارم بررسی خواهد شد.
به فرآیندهایی که در آن از داده‌های برچسب دار استفاده کردید، یادگیری با نظارت یا Supervised Learning گفته می‌شود و اگر داده‌ها بدون برچسب (Unlabeled) باشند یادگیری بدون ناظر یا Unsupervised Learning نام دارد.
حال که با نوع مسائل و برخی اصطلاحات آشنا شدید، به آماده سازی محیط کار می‌پردازیم.

۰-۳ آماده سازی محیط

رویکرد این کتاب، آموزش با کمک حل مثال است و نه بیان تئوری‌ها. از این رو آماده‌سازی یک محیط پیاده‌سازی برای ادامه کار، بسیار مهم می‌باشد. به همین دلیل در همین ابتدای کار، محیط پیاده‌سازی را آماده می‌کنیم.
در میان زبان‌های متعدد برنامه‌نویسی موجود در بازار، محبوب‌ترین زبان در میان متخصصین و دانشمندان علوم داده، Python نام دارد. این زبان بدون آنکه کاربر را درگیر پیچیدگی‌های برنامه‌نویسی کند، امکان تعریف مفاهیمی همچون لیست‌ها، دیکشنری‌ها، ماتریس‌ها و ... را در اختیار کاربر می‌گذارد. به این موارد طیف وسیعی از ابزارها، کتابخانه‌ها و امکانات متعدد را نیز اضافه کنید.
آخرین نسخه این زبان برنامه نویسی از طریق سایت www.python.org در دسترس می‌باشد. تمامی کدهای این کتاب بر روی نسخه 3.5.3 اجرا شده است. پس از نصب python، به کتابخانه‌های زیر نیز نیاز است:

- Numpy
- Scikit-learn
- Matplotlib
- Pandas

نحوه نصب کتابخانه‌های فوق بسیار ساده است. هنگامی که به اینترنت متصل هستید دستور زیر را در خط فرمان سیستم عامل اجرا کنید:

```
pip install <package_name>
```

و به جای <package_name> نام کتابخانه خود را وارد کنید. محیط پیاده‌سازی آماده شده است.
و قدم آخر؛ فرآیندهای یادگیری ماشین نیازمند مجموعه داده است. تمامی مجموعه داده‌های مورد استفاده در کتاب، گردآوری شده و از طریق آدرس https://github.com/soheil-kia/ml-book در دسترس خوانندگان قرار دارد. همچنین در راستای سهولت کار خوانندگان نیز، تمامی source code ها، از طریق آدرس فوق قابل حصول می‌باشد.

۰-۴ جمع بندی

در این فصل، با کلیات یادگیری ماشین آشنا شدیم تا خواننده تصویری کلی از آنچه در پیش دارد را در ذهن خود شکل دهد. هدف در این فرآیندها، دستیابی به ماشینی هوشمند است که قادر خواهد بود با توجه به تجربیاتی که تا به حال کسب شده، برای آینده پیش‌بینی و تصمیم‌سازی کند. بدین منظور، تجربیات گذشته در قالب مجموعه داده در اختیار ماشین قرار می‌گیرد و پس از آن با توجه به هدف مورد نظر، الگوریتم مناسب را انتخاب و اعمال می‌کنیم. در نتیجه این فرآیند، با توجه به ورودی‌ها و خروجی‌های مورد نظر، تابعی توسط ماشین حدس زده می‌شود که به آن مدل نیز اطلاق می‌گردد. در دسته دیگر از مسائل، ماشین دنبال ارتباطات پنهان در میان داده‌ها می‌گردد.

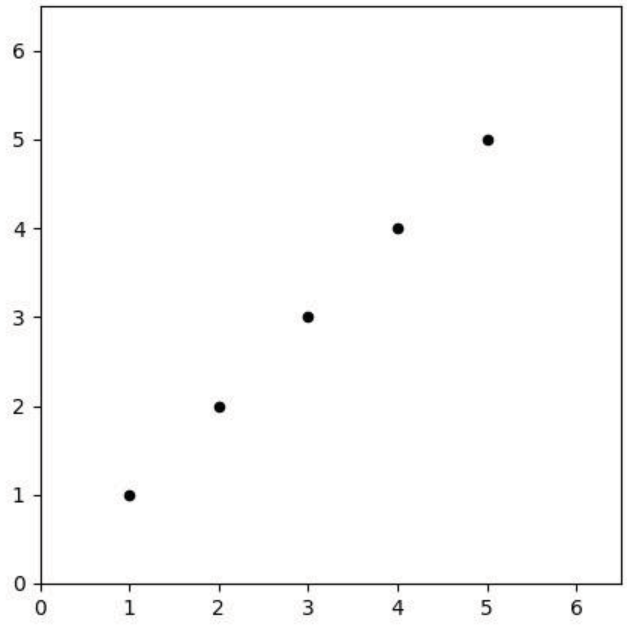
در فصول آتی به جزئیات آنچه به آن اشاره شد خواهیم پرداخت.

فصل اول

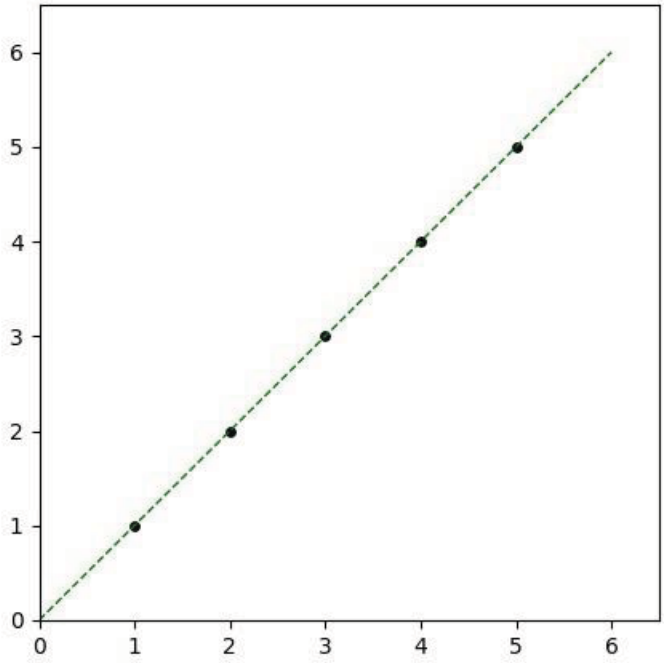
Regression

۱-۱ مقدمه

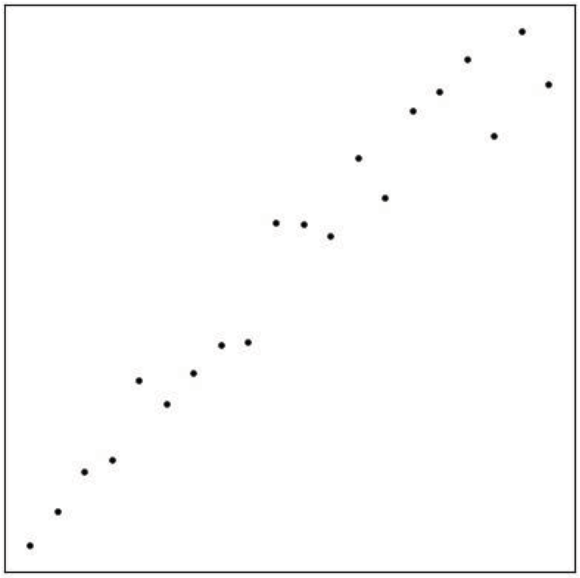
نمودار رو به رو را در نظر بگیرید. از شما خواسته شده است که حدس بزنید نقطه بعدی کجاست؟ احتمالا نقطه بعدی را در (۶،۶) خواهید گذاشت. اما چرا این نقطه را انتخاب می‌کنید؟



بگذارید نگاهی دقیق تر به این فرآیند انتخاب بیاندازیم. ذهن تلاش می کند تا ابتدا ارتباطی بین این نقاط پیدا کند و به این نتیجه می‌رسد که می‌تواند به کمک یک خط این نقاط را به هم وصل کند. این خط به مانند شکل زیر بوده و در زمان حدس زدن نقطه بعدی تلاش می‌شود این نقطه در امتداد خط فرضی انتخاب شود.



حال سعی کنیم تا سوال را اندکی پیچیده کنیم. شکل زیر را در نظر گرفته و نقطه بعدی را حدس بزنید.



پر واضح است که نمی‌توان خط راستی را پیدا کرد که از تمامی این نقاط عبور کند. پیدا کردن خطی که کمترین فاصله با این نقاط را داشته باشد نیز می‌تواند مطلوب باشد. اما چگونه می‌توان چنین خطی را یافت؟

Linear Regression ۲-۱

هر خط راست بر روی صفحه (نمودار دو بعدی) دارای معادله خطی به شکل $y = mx + b$ می‌باشد که در آن m شیب خط، b عرض از مبدا، x متغیر مستقل و $f(x)$ یا y متغیر وابسته نام دارند. از طرف دیگر ما مجموعه‌ای از (x, y) ها داریم که بیانگر نقاط موجود روی نمودار است. با جایگذاری تک‌تک مقادیر x در معادله خط مفروض، یک y محاسبه می‌شود که می‌توان آن‌را با y مطلوب در مجموعه مقایسه کرد. هدف در اینجا پیدا کردن یک $f(x)$ است که به ازای تک‌تک نقاط x، فاصله y از $f(x)$ مقداری کمینه باشد. یعنی کمینه کردن معادله زیر:

$$\sum_{i=1}^{n} |y_i - f(x_i)|$$

که در آن n تعداد نقاط است. نکته ای که باید به آن توجه داشت این است که هر چه تعداد نقاط بیشتر باشد، این مقدار نیز افزایش می‌یابد. از سوی دیگر، تعداد نقاط نشان‌دهنده میزان دانش و دانسته‌های ما است. هرچه دانش ما افزایش یابد، باید خطای ما کاهش یابد و نه بالعکس. بدیهی است که این رفتار، مطلوب ما نیست. پس باید به دنبال معیاری باشیم که وابستگی ما به تعداد نقاط را از میان بردارد. این معیار، میانگین است. میانگین فاصله فوق برابر است با:

$$\frac{1}{n}\sum_{i=1}^{n} |y_i - f(x_i)|$$

به مقدار فوق میانگین خطای مطلق یا Mean Absolute Error(M.A.E.) گفته می‌شود. هر چقدر این مقدار کاهش یابد، نشان‌دهنده این است که تخمین بهتری زده شده است. همچنین می توان از معیار دیگری به نام میانگین خطای مربعات یا Mean Squared Error(M.S.E) استفاده کرد که با مقدار MAE هم ارز است. به این دسته از توابع در یادگیری ماشین و همچنین در حالت کلی تر در مسائل بهینه سازی، Loss Function گفته می‌شود. برای هر دسته از مسائل، Loss Function خاص و متناسب تعریف و استفاده می‌شود که در جای خود به آن اشاره خواهد شد.

حال که هدف کمینه کردن Loss Function است، باید به دنبال روالی برای این کار گشت. راهکاری که عموما از آن بهره برده می‌شود، Gradient Descent نام دارد.

فرض کنید در کمرکش یک کوه ایستاده‌اید و می‌خواهید از آن پایین بیایید. مشکل اینجاست که به علت مه غلیظ حتی جلوی پای خود را نمی‌بینید. چه راهکاری را در پیش خواهید گرفت؟ یک

فصل دوم

Classification

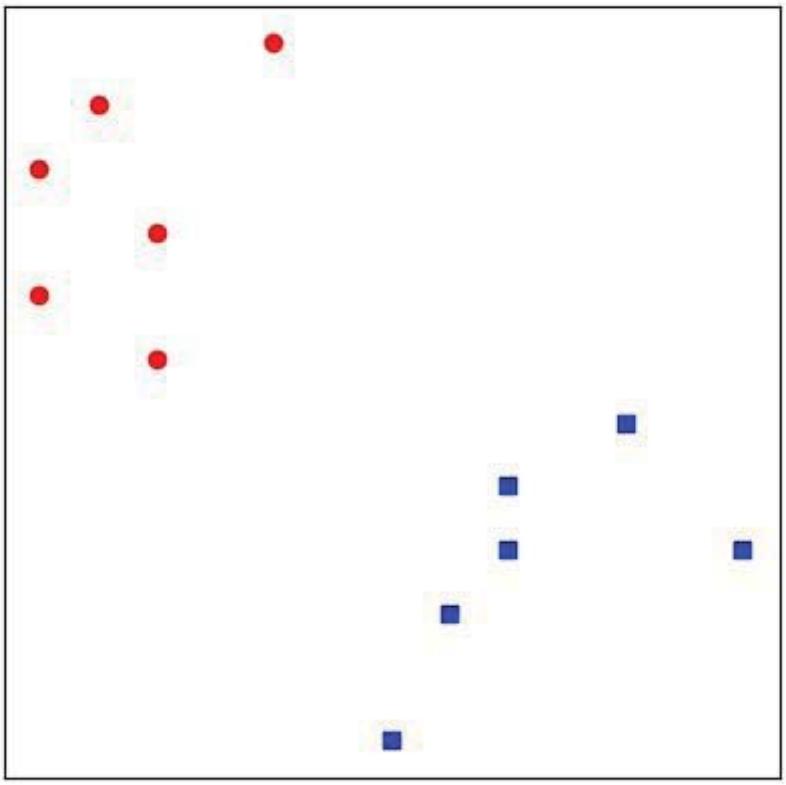
۲-۱ مقدمه

در فصل گذشته نگاهی داشتیم به مبحث Regression که تلاش داشت در نهایت مدلی ارائه کند که با کمترین خطا، از روی داده‌های در دست، خطی بهینه را ترسیم کند تا به کمک آن، برای داده‌های جدید نتیجه را تخمین بزند.

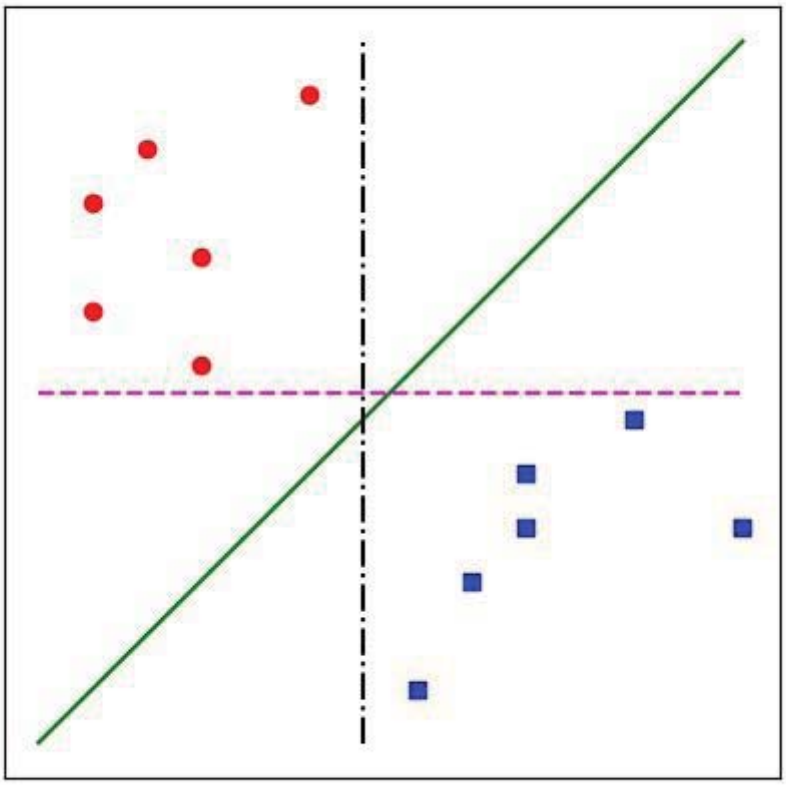
در این فصل کاربردی دیگر از فرآیند آموزش ماشین را مورد بررسی قرار می‌دهیم که هدف آن گروه‌بندی مجموعه داده‌ها خواهد بود. به عنوان مثال، تعدادی کافی سیب و پرتقال به عنوان ورودی به مدل داده می‌شود و سپس مدل تلاش می‌کند تا برای میوه بعدی متوجه شود که سیب است یا پرتقال. به این‌گونه مسائل، مسئله Classification یا گروه‌بندی اطلاق می‌شود.

در این‌گونه مسائل، به متغیر خروجی، برچسب یا Label گفته می‌شود. به عنوان مثال روز می‌تواند برچسب آفتابی یا ابری داشته باشد. مسائل گروه‌بندی می‌توانند علاوه بر مقادیر پیوسته، مقادیر گسسته را نیز به عنوان ورودی قبول کنند.

برای ایجاد یک تصویر ذهنی، فرض کنید شکل زیر به شما داده شده است.



اگر از شما خواسته شود که با یک خط دایره‌ها و مربع‌ها را از هم جدا کنید، این خط را چگونه انتخاب و ترسیم می‌کنید؟ تمامی خطوط ترسیم شده در شکل زیر پاسخی صحیح است.



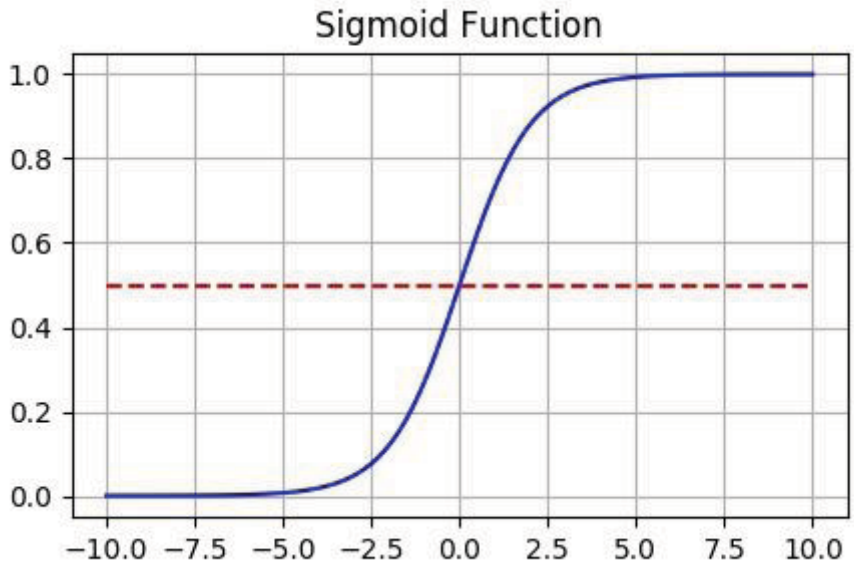
هدف این بخش از کتاب بررسی راهکارهای موجود برای حل چنین مسائلی است.

۲-۲ Logistic Regression یا Binary Classification

در برخی از مسائل، می‌توان با یک تغییر کوچک در مسئله، با کمک الگوریتم‌های Regression یک مسئله گروه‌بندی را حل کرد. به عنوان مثال، قیمت اجناس به صورت مقادیر پیوسته بیان می‌شود. اما می‌توان قیمت کالاها را با یک خط مرزی جدا کرده و به دو گروه گران و ارزان تقسیم کرد. به این فرآیند گسسته‌سازی یا Discretization گفته می‌شود.

حال که با یک تغییر توانستیم مسئله را از Regression به گروه‌بندی تبدیل کنیم، نیازمند یافتن تابعی هستیم که دامنه آن تمامی اعداد حقیقی را در بر گرفته و برد آن از بالا و پایین کران‌دار

باشد. کران دار بودن تابع سبب می‌شود تا با تغییر داده‌ها و کم و زیاد شدن مقادیر، عدد میانگین ثابت بماند. در اینصورت از مقدار میانگین می‌توان به عنوان معیاری برای جداسازی بهره برد. به عنوان مثال تابع زیر را در نظر بگیرید:



تابع فوق نگاشتی به بازه کران دار $(1\ 0)$ را به دست می‌دهد. در شکل فوق تمامی مقادیر بالاتر از خط چین در یک گروه و مقادیر پایین خط‌چین در گروهی دیگر قرار گرفته و به این شکل دسته‌بندی کامل می‌شود.

ضابطه تابع فوق، که به تابع Sigmoid معروف است، به شکل زیر می‌باشد:

$$f(x) = \frac{1}{1 + e^{-x}}$$

این تابع، همان تابعی است که در Logistic Regression مورد استفاده است.

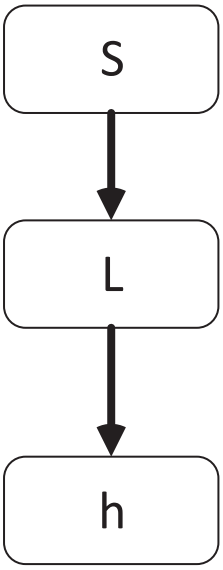
باکتری آلیسایکلوباسیلیوس، این توانایی را دارد که از فرآیند پاستوریزاسیون جان سالم به در ببرد. در مثال زیر از یک مجموعه داده استفاده شده که مقادیر pH، میزان ساکاروز موجود، دما و غلظت نیسین داده شده و در ستون آخر با مقادیر ۰ و ۱ نشان داده شده که در این شرایط آیا باکتری زنده مانده است یا نه؟ حال می‌خواهیم با استفاده از Logistic Regression تعیین کنیم در چه شرایطی باکتری زنده می‌ماند.

فصل سوم

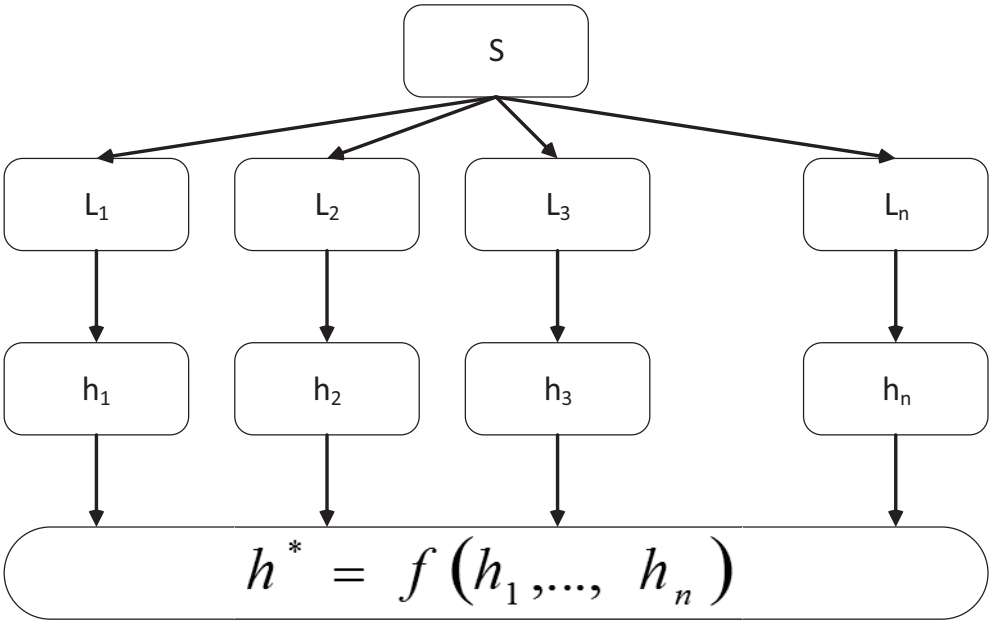
Ensembling

۳-۱ مقدمه

تا بدین جا، راهکارهایی متعدد برای حل مسائل مختلف مورد بررسی قرار گرفته که در هر یک الگوریتمی خاص به عنوان راه حل مورد استفاده بوده است. در تمامی این روش‌ها مجموعه داده اولیه به عنوان ورودی به مدل داده می‌شود، سپس مدل تلاش می‌کند تا ارتباطی بین مقادیر داده ای یافته و با کمک آن، برای ورودی‌های جدید، خروجی را محاسبه کند. چیزی همانند شکل زیر:



که در آن S مجموعه داده، L نشانگر Learner و h نتیجه نهایی مدل (hypothesis) است.
قدر مسلم، پاسخ یک مدل به تنهایی نمی‌تواند جوابی دقیق باشد. اما آیا راهکاری وجود دارد که با کمک آن، چندین مدل را با هم ادغام کرد و به نتیجه‌ای بهینه‌تر از نتیجه یک مدل دست یافت (چیزی شبیه به شکل زیر)؟ پاسخ دادن به این سوال، فصل پیش رو را رقم می‌زند.

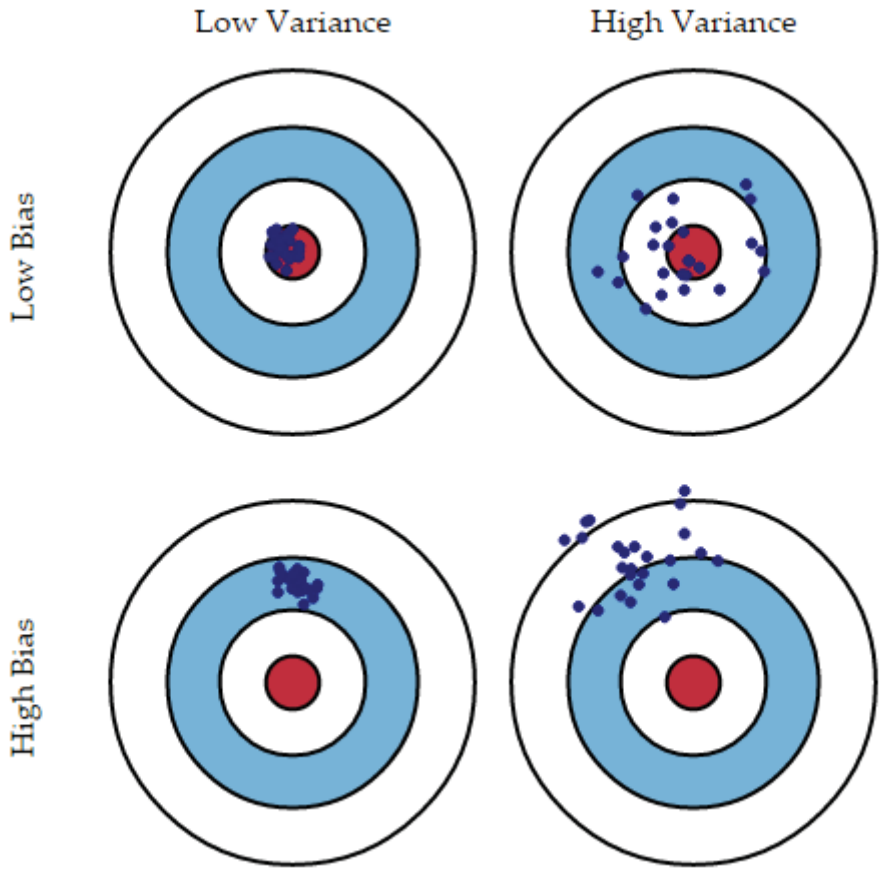


در این فصل تلاش داریم تا چگونگی روش‌های ادغام چند Learner با یکدیگر را بررسی کنیم و ببینیم هریک به چه طریق عمل می‌کند. این متدولوژی، Ensemble نام گرفته است (Ensemble در لغت به معنی مجموعه‌ای است که به صورت یک شی واحد دیده می‌شود و نه اجزای جدا از هم).

برای درک بهتر، به مثال زیر توجه کنید. فرض کنید می‌خواهید در یک شرکت سرمایه‌گذاری کنید اما از نتیجه کار اطمینان ندارید. از این رو از سه نفر متخصص می‌خواهید که به شما مشاوره بدهند (نظر این مشاوران مستقل از هم هستند). اگر میزان دقت هر یک از مشاوران تنها ۷۵٪ باشد و شما به حرف یکی از آن‌ها بسنده کنید، میزان ریسک شما بالا خواهد بود (۲۵٪)، اما اگر هر سه مشاور به شما یک نظر واحد را بدهند، آن‌گاه اعتماد زیادی نسبت به سرمایه‌گذاری خود پیدا خواهید کرد.

پیش از آنکه به بررسی راهکارهای موجود بپردازیم، بهتر است ابتدا به مرور مفهوم Bias و Variance پرداخته شود که به درک راحت‌تر مباحث پیش رو خواهد انجامید.

Bias به میزان اختلاف مقدار مورد حدس توسط مدل با مقدار واقعی اطلاق می‌گردد. همچنین در زبان ساده، Variance معیاری است برای نشان دادن میزان پراکندگی در داده‌ها. هنگامی که از چند Predictor استفاده می‌کنیم یکی از حالات زیر اتفاق می‌افتد:



در هر قسمت شکل بالا، دایره میانی نشانگر مقدار مورد نظر و هرکدام از نقاط نشانه مقدار مورد حدس توسط مدل‌هایی است که در Ensemble استفاده شده است. در شکل بالا سمت چپ، مقادیر مورد حدس مطابق با مقدار واقعی است. این حالت را می‌توان حالتی مطلوب در نظر گرفت. در شکل پایین سمت چپ، مقادیر مورد حدس به یکدیگر نزدیک است اما با مقدار واقعی فاصله دارد. در این حالت پراکندگی مقادیر مورد حدس کم، اما Bias زیاد است. شکل بالا سمت راست، مقادیر مورد حدس حول مقدار واقعی قرار گرفته، اما نسبت به یکدیگر پراکنده است (Low Bias-High Variance) و در نهایت در شکل پایین سمت راست هر دو مقدار Bias و Variance زیاد است.

هدف در Ensemble Learning کاهش مقدار Bias و Variance با راهکارهایی است که به آن خواهیم پرداخت.

۲-۳ Ensemble خود را بسازید (Stacking)

همان‌گونه که پیش از این مطرح شد، ایده شکل دهنده Ensemble، ترکیب چندین مدل با یکدیگر که نتایج آن‌ها می‌تواند چندان دقیق نباشد و ایجاد یک مدل جدید است که خطای کمتری نسبت به مدل‌های اولیه دارد می‌باشد. از این رو به گروه‌بندهای اولیه، Weak Learners نیز می‌گویند. آنچه که در اینجا نیازمند تشریح است، نحوه ترکیب Weak Learner ها با یکدیگر است.

در مسائل گروه‌بندی، یکی از پرکاربردترین و در نتیجه محبوب‌ترین روش‌های ساخت Ensemble، استفاده از Majority Voting و یا رأی اکثریت است. به زبان ساده، در این روش، گروهی را انتخاب می‌کنیم که بیش از نیمی از رأی‌ها را به دست آورد. کاربرد رأی اکثریت برای تنها دو گروه است، اما به سادگی می‌توان آن را به چند گروه نیز تعمیم داد. در این حالت، که Plurality Voting یا رأی جمعی نام دارد، گروهی به عنوان جواب انتخاب می‌شود که بیشترین رأی را دارد و نه لزوما بیش از نیمی از آرا را. در کتابی که در دست دارید، این دو مفهوم به جای یکدیگر به کار می‌روند.

برای درک بهتر این مفاهیم، به مثال زیر توجه کنید. در این مثال، مسئله ۲-۴ را که با کمک SVM حل شده بود، بار دیگر به کمک Ensemble حل می‌کنیم. ابتدا Weak Learner ها را با مدل‌های SVM، Logistic Regression و درخت تصمیم می‌سازیم. سپس با استفاده از رأی اکثریت، با توجه به خروجی سه مدل یاد شده نتیجه نهایی را انتخاب می‌کنیم.

فصل چهارم

Clustering

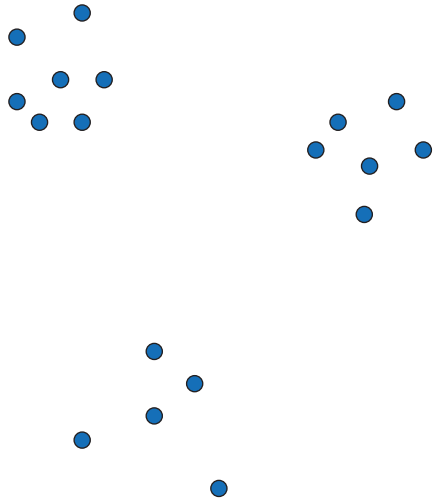
۴-۱ مقدمه

حتما برای شما هم اتفاق افتاده است که هنگامی که در سایت‌های مختلف در حال جستجو برای یک فیلم یا موسیقی هستید، همزمان با جستجوی شما، پیشنهاداتی از طرف سایت مزبور مبنی بر اینکه اگر این فیلم یا موسیقی را می‌پسندید، این موارد را هم ببینید، دریافت کرده‌اید. نکته جالب این است که معمولا این پیشنهادات را می‌توان مثبت هم ارزیابی کرد و به آن‌ها توجه داشت. چه‌چیز باعث می‌شود که یک سایت با توجه به جستجوی شما، از میان انبوه فیلم‌های خود، مجموعه‌ای از دیگر فیلم‌ها را به شما پیشنهاد دهد که اتفاقا پیشنهادهای مناسبی نیز هستند؟ در این فصل قصد داریم تا با دسته‌ای از الگوریتم‌ها آشنا شویم که رفتار فوق را دارند.

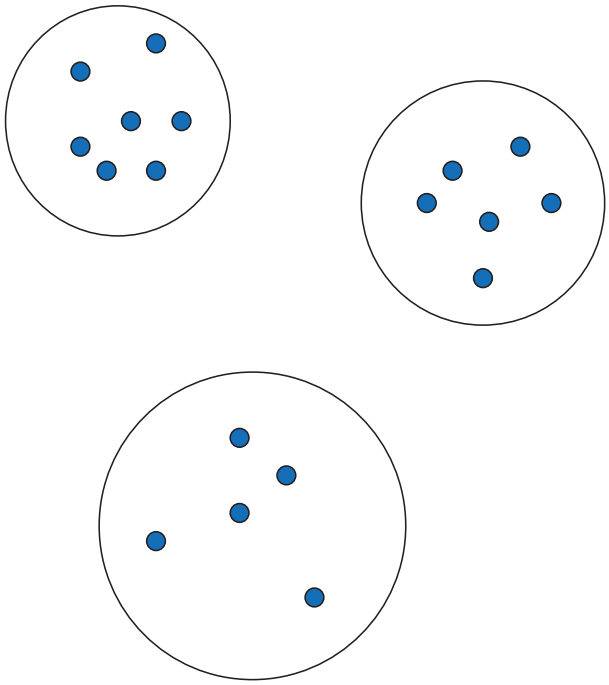
الگوریتم‌هایی که در این فصل مورد بحث قرار می‌گیرند، هدف خوشه‌بندی داده‌ها را دنبال می‌کنند. الگوریتم‌های خوشه‌بندی داده‌ها در دسته الگوریتم‌های بدون ناظر (Unsupervised) قرار می‌گیرد. هدف این دست از الگوریتم‌ها، قرار دادن داده‌های مشابه در گروه‌های یکسان است. تفاوت مهم این الگوریتم‌ها با الگوریتم‌های گروه‌بندی در این است که الگوریتم‌های گروه‌بندی در دسته الگوریتم‌های با نظارت (Supervised) قرار می‌گیرد.

در الگوریتم‌های با نظارت، هر نمونه داده، یک برچسب دارد که نشان می‌دهد این نمونه داده، در مجموعه هدف، چه مقداری را می‌پذیرد. به عنوان مثال در گروه‌بندی می‌دانیم هر نمونه داده جزو دسته سیب‌ها است یا پرتقال‌ها. اما در خوشه‌بندی این‌گونه نیست. بلکه تعدادی نمونه داده داریم و می‌خواهیم بدانیم آیا داده‌ها با یکدیگر تشکیل یک خوشه را می‌دهند یا خیر و اگر تشکیل خوشه می‌دهند، کدام نمونه داده متعلق به کدام خوشه است. حتی در زمان شروع کار، نمی‌دانیم که این خوشه‌ها چه هستند، بلکه به دنبال یافتن ارتباطاتی ناشناخته در داده‌ها هستیم. در واقع می‌خواهیم بدانیم کدام نمونه‌های داده بیشتر به یکدیگر مربوط و شبیه هستند.

برای درک بهتر موضوع به شکل زیر توجه کنید.



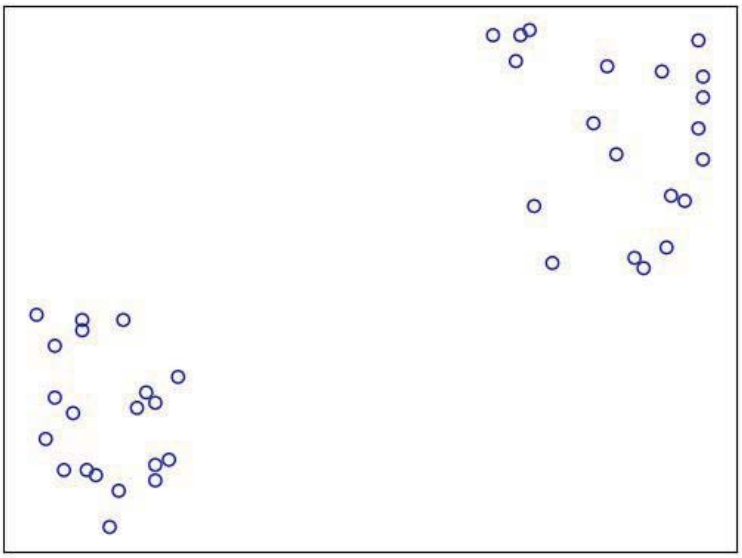
در شکل فوق، نمونه‌های داده دارای برچسب نیستند و همگی فقط داده هستند. اما هنگامی که از شما خواسته شود تا این داده‌ها را از همدیگر جدا کرده و در گروه‌های متفاوت قرار دهید، این گروه‌ها را به شکل زیر تشخیص خواهید داد.



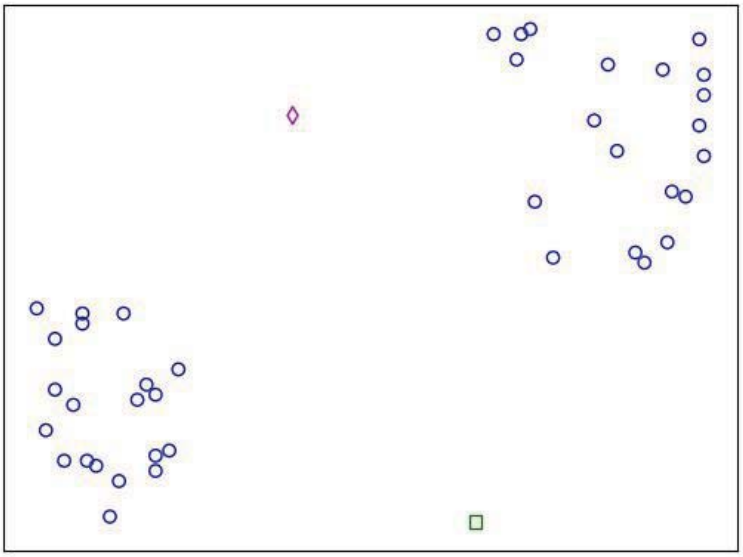
هدف الگوریتم‌های این فصل، یافتن راهکاری برای انجام فرآیندهایی است که در نهایت به خوشه‌بندی داده‌ها می‌انجامد.

K-Means ۲-۴

می‌توان از K-Means به عنوان شناخته شده‌ترین الگوریتم خوشه‌بندی نام برد. برای درک بهتر خوانندگان از نحوه عملکرد این الگوریتم، با یک مثال به تشریح K-Means می‌پردازیم. همچنین درک این الگوریتم کمک شایانی در درک ساده‌تر و بهتر دیگر الگوریتم ها خواهد داد.
برای شکل زیر می‌خواهیم فرآیند خوشه‌بندی به روش K-Means را اجرا کنیم.

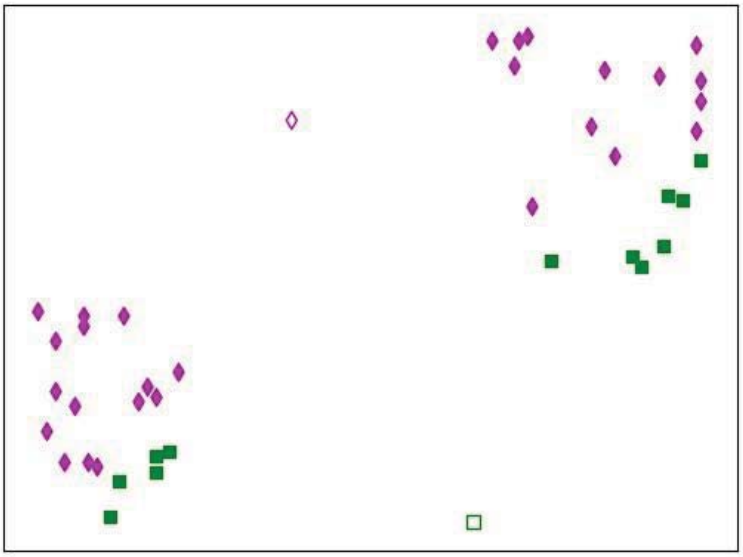


این الگوریتم برای شروع، نیازمند معرفی تعداد خوشه‌ها می‌باشد (به نحوه تعیین خوشه‌ها در ادامه اشاره خواهد شد). پر واضح است که در شکل بالا دو خوشه داریم. برای شروع، به تعداد خوشه‌ها، نقاطی در فضای داده به صورت تصادفی انتخاب می‌گردد. این نقاط نشان‌دهنده مرکز هر یک از خوشه‌ها خواهد بود.



در شکل بالا دو خوشه مربع و لوزی خواهیم داشت. مربع تو خالی نشان‌دهنده مرکز یک خوشه و لوزی تو خالی بیانگر مرکز خوشه دیگر است. توجه داشته باشید که در این مرحله، نقاط همچنان عضو هیچ خوشه‌ای نیستند.

در مرحله دوم، فاصله هر نمونه داده با هریک از مراکز خوشه‌ها محاسبه می‌شود و هر نمونه داده به هر کدام از این مراکز نزدیک‌تر باشد، متعلق به آن خوشه است.



در مرحله سوم نقطه میانگین هر خوشه از روی عناصر آن محاسبه شده و آن نقطه به عنوان مرکز جدید خوشه قرار داده می‌شود. به بیان دیگر، در این مرحله مرکز خوشه‌ها را جابه‌جا خواهیم کرد.

فصل پنجم

Data Preprocessing

۵-۱ مقدمه

تا بدینجا با روش‌های مختلف مطرح در آموزش ماشین آشنا شده‌ایم. در این روش‌ها، با توجه به مسئله پیش‌رو، مدل و الگوریتم مورد نظر را انتخاب کرده و بر روی مجموعه داده اعمال می‌کنیم تا به نتیجه و خروجی دلخواه دست پیدا کنیم. بخشی که تا اینجا به آن نپرداخته‌ایم، مجموعه داده است. اهمیت مجموعه داده به حدی است که تاثیر آن در رسیدن به نتیجه مطلوب، می‌تواند شانه به شانه انتخاب مدل بساید.

حجم داده‌ها در عصر دیجیتال با سرعتی دهشتناک در حال افزایش است. به همین دلیل نام این حوزه Big Data می‌باشد. متاسفانه استخراج داده‌های با معنی از این بین، همچون یافتن سوزن در انبار کاه است. در حال حاضر، بیشتر زمان متخصصین داده صرف تبدیل داده‌های بد به داده‌های قابل استفاده می‌شود.

داده‌های دنیای واقعی در بسیاری از مواقع ناقص، متناقض و دارای مشکل است. وقوع این امر بدین سبب است که فرآیند جمع‌آوری داده‌ها وابسته به عوامل متعددی می‌باشد که هریک به طور جداگانه می‌توانند تولید خطا کنند. همچون ابزارهای اندازه‌گیری، شرایط محیطی آزمایش، خطاهای انسانی و ... با این اوصاف توجه، تدوین و تبیین راهکارهایی جهت آماده‌سازی داده‌ها راهگشا خواهد بود.

با توجه به ماهیت بروز مشکل در مجموعه داده و تنوع این مشکلات، پرداختن به تمامی اتفاقات و امکانات امری غیرممکن می‌نماید. لذا در ادامه تلاش شده است که خوانندگان با مشکلاتی که عمومی‌تر بوده آشنا شده و همچنین با یادگیری راهکارهایی که بیشتر مورد استفاده است، به زاویه دیدی دست پیدا کنند تا بتوانند در آینده به کمک خلاقیت خود مسائل پیش رو را حل و فصل کنند.

۵-۲ بررسی مجموعه داده

هنگامی که یک مجموعه داده را در اختیار دارید تا فرآیندهای یادگیری ماشین را بر روی آن اجرا کنید، در گام نخست لازم است تا مجموعه داده بررسی شود. این بررسی از هر جهت می‌تواند کمک کند تا با شناخت داده‌ها، خطاهای موجود در آن را رفع کرده و در نتیجه از به خطا رفتن

الگوریتم‌هایی که بر روی داده‌ها اعمال می‌شود پیشگیری کرد. با یک مثال به تشریح موارد فوق می‌پردازیم.

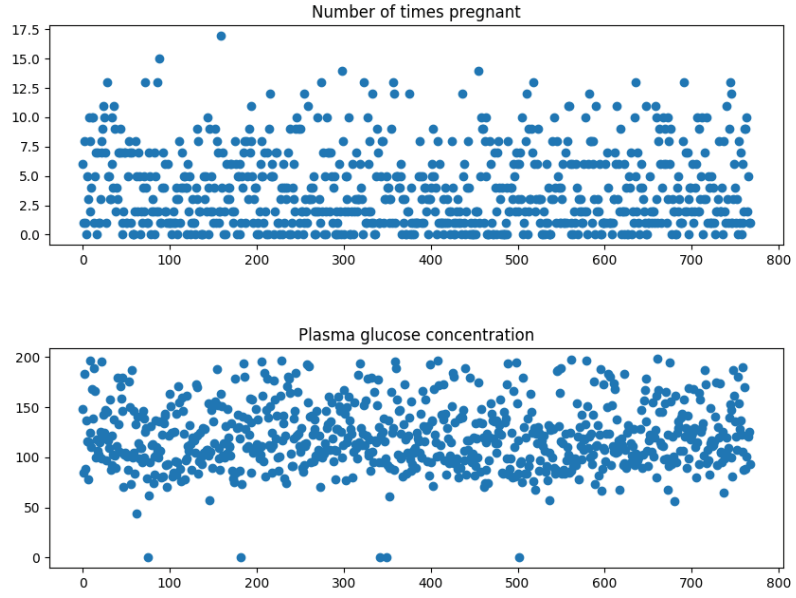
مجموعه داده ای که در ادامه از آن استفاده می‌شود، حاوی اطلاعات اشخاصی است که در نهایت نشان می‌دهد این بیماران آیا دچار دیابت می‌باشند یا نه. ابتدا به بررسی مجموعه داده می‌پردازیم.

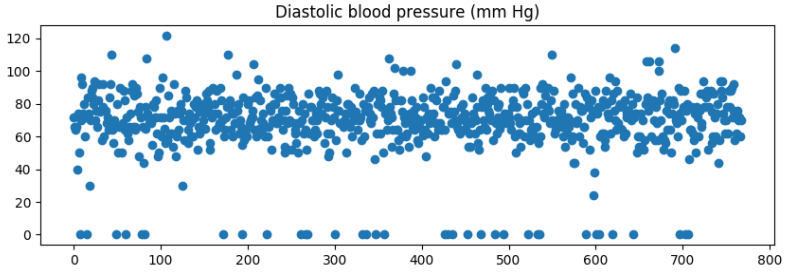
در توضیحات مجموعه داده موارد زیر به عنوان توضیح هر صفت خاصه آمده است:

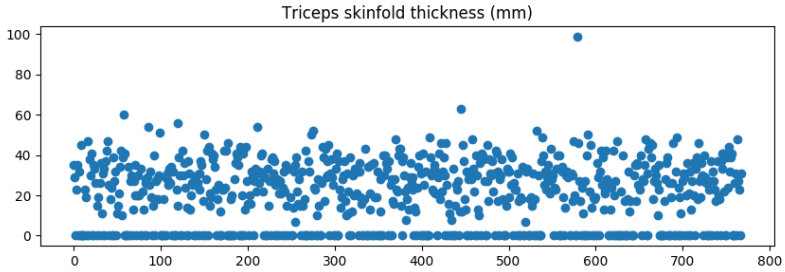
- تعداد دفعات بارداری
- غلظت گلوکز
- فشار خون
- ضخامت پوست بر حسب میلی‌متر
- شاخص BMI
- تابع دیابت Pedigree (که به کمک یک تابع میزان تاثیر محیط و توارث را در تمایل به دیابت نشان می‌دهد)
- سن

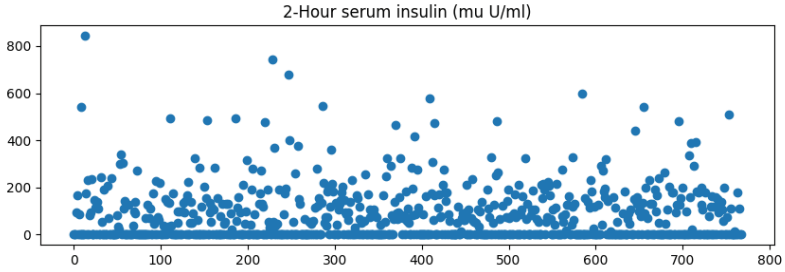
ستون آخر نیز نتیجه نمونه‌گیری را نشان می‌دهد که عدد صفر بیانگر عدم ابتلا به دیابت در بازه پنج ساله و عدد یک نشانگر ابتلا به دیابت است.

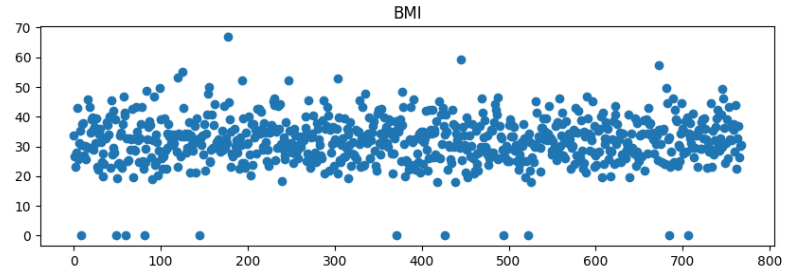
با توجه به اینکه نوع داده‌ها عددی است، گام نخست در اینجا تصویر سازی از داده‌ها می‌باشد. هر کدام از صفات خاصه را به صورت جداگانه تصویر کرده و به بررسی آن می‌پردازیم.

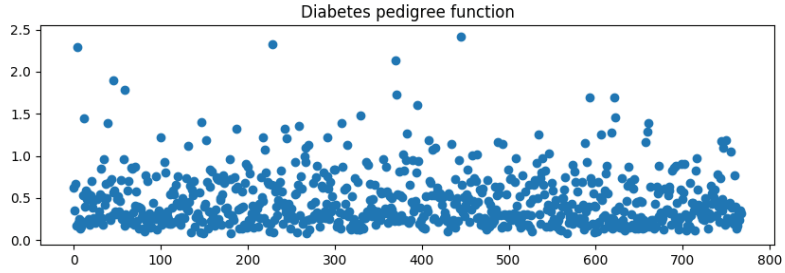


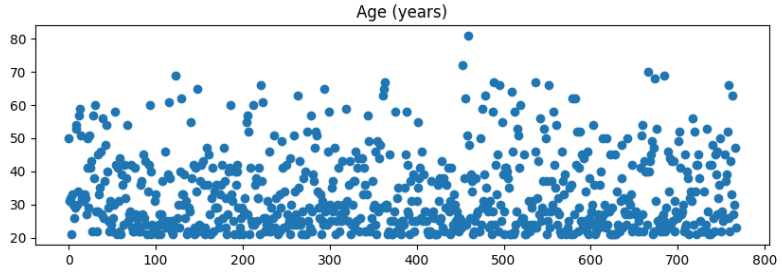

Diastolic blood pressure (mm Hg)
0 20 40 60 80 100 120
0 100 200 300 400 500 600 700 800


Triceps skinfold thickness (mm)
0 20 40 60 80 100
0 100 200 300 400 500 600 700 800


2-Hour serum insulin (mu U/ml)
0 200 400 600 800
0 100 200 300 400 500 600 700 800


BMI
0 10 20 30 40 50 60 70
0 100 200 300 400 500 600 700 800


Diabetes pedigree function
0.0 0.5 1.0 1.5 2.0 2.5
0 100 200 300 400 500 600 700 800



همچنین علاوه بر تصویر سازی داده‌ها، آن ها را خوانده و اطلاعات آماری اولیه را نیز استخراج می‌کنیم:

```
import numpy as np
import pandas as pd

data = pd.read_csv('pima.txt', header=None)
print(data.describe())
```

```
                0           1           2           3  \
count  768.000000  768.000000  768.000000  768.000000
mean     3.845052  120.894531   69.105469   20.536458
std      3.369578   31.972618   19.355807   15.952218
min      0.000000    0.000000    0.000000    0.000000
25%      1.000000   99.000000   62.000000    0.000000
50%      3.000000  117.000000   72.000000   23.000000
75%      6.000000  140.250000   80.000000   32.000000
max     17.000000  199.000000  122.000000   99.000000

                4           5           6           7  \
count  768.000000  768.000000  768.000000  768.000000
mean    79.799479   31.992578    0.471876   33.240885
std    115.244002    7.884160    0.331329   11.760232
min      0.000000    0.000000    0.078000   21.000000
25%      0.000000   27.300000    0.243750   24.000000
50%     30.500000   32.000000    0.372500   29.000000
75%    127.250000   36.600000    0.626250   41.000000
max    846.000000   67.100000    2.420000   81.000000

                8
count  768.000000
mean     0.348958
std      0.476951
min      0.000000
25%      0.000000
50%      0.000000
75%      1.000000
max      1.000000
```

کلام آخر

آنچه گذشت، تنها نقطه آغازی است برای ورود به این دنیای بیکران. همان‌گونه که از ابتدای کتاب نیز به آن اشاره شد، هدف از نگارش این کتاب آشنایی خوانندگان با فرآیندهای یادگیری ماشین، بدون درگیری با پیچیدگی‌های الگوریتمی آن است. خوانندگان علاقه‌مند می‌توانند در کتب تخصصی‌تر به دنبال تئوری‌های پشت این الگوریتم‌ها باشند. در این کتاب همواره تکیه بر جنبه عملی این مفاهیم بوده است تا خواننده بتواند آنچه را آموخته در حوزه‌های مختلف به کار ببندد.

آنچه که سبب دلگرمی خواهد بود، راهنمایی‌های خوانندگان برای بهبود این کتاب، در ویرایش‌های بعدی است. از این رو پست الکترونیکی ml.soheil.Kia@gmail.com شنونده‌ی راهنمایی‌های دوستان و پاسخگوی پرسش‌های ایشان خواهد بود.

امیدوارم آنچه خوانده‌اید برایتان مفید واقع شود و از آن لذت برده باشید.

۲۱ شهریور ۱۳۹۷

کیا

منابع

- Raschka, S. et. al. *Python Machine Learning 2nd. Ed.* Packt Publishing Ltd. 2017
- Géron, A. *Hands-On Machine Learning with Scikit-Learn and TensorFlow.* O'Reilly Media, Inc. 2017
- VanderPlas, J. *Python Data Science Handbook*. O'Reilly Media, Inc. 2017
- Chollet, F. *Deep Learning with Python*. Manning Publications Company 2018
- UCI Machine Learning Repository. *https://archive.ics.uci.edu/ml/index.php*
- Wikipedia. *https://en.wikipedia.org/*
- Machine Learning Mastery. *https://machinelearningmastery.com/*
- Analytics Vidhya. *https://www.analyticsvidhya.com/*
- Toward Data Science. *https://towardsdatascience.com/*
- Ng, A. et. al. CS229: Machine Learning. *http://cs229.stanford.edu/*
- Aguiar, E. et. al. Data Mining CSE 40647/60647. *https://www3.nd.edu/~rjohns15/cse40647.sp14/www/home.php*
- Rai, P. Probabilistic Machine Learning CS772A. *https://www.cse.iitk.ac.in/users/piyush/courses/pml_fall17/pml_fall17.html*
- Rai, P. Bayesian Machine Learning CS698S. *https://www.cse.iitk.ac.in/users/piyush/courses/bml_winter17/bayesian_ml.html*
- Rai, P. Machine Learning CS771A. *https://www.cse.iitk.ac.in/users/piyush/courses/ml_autumn16/ML.html*
- RJ, D. The Mean Shift Clustering Algorithm. *http://efavdb.com/mean-shift/*
- Powell, V. Principal Component Analysis. *http://setosa.io/ev/principal-component-analysis/*
- Analytics Vidhya Content Team. Practical Guide to Principal Component Analysis (PCA) in R & Python. *https://www.analyticsvidhya.com/blog/2016/03/practical-guide-principal-component-analysis-python/*
- Brownlee, J. How to Handle Missing Data with Python. *https://machinelearningmastery.com/handle-missing-data-python/*
- Sarkar, D. Understanding Feature Engineering (Part 1)-Continuous Numeric Data. *https://towardsdatascience.com/understanding-feature-engineering-part-1-continuous-numeric-data-da4e47099a7b*
- Sarkar, D. Understanding Feature Engineering (Part 2)-Categorical Data. *https://towardsdatascience.com/understanding-feature-engineering-part-2-categorical-data-f54324193e63*