report

Rain Forecast Model

Amirreza Madadi

هدف: قصد ما از انجام این پروژه ارائه مدلی است که بر اساس تعداد مشخصی از ویژگی ها و مقادیر آنها، پیش‌بینی کند که آیا روز آینده بارش باران خواهیم داشت یا خیر.

برای ارائه مدلی کارآمد لازم است که آن را آموزش دهیم. مجموعه ای متشکل از 145460 روز در طی 10 سال با 23 ویژگی در اختیار داریم.

لزومی ندارد تمامی ویژگی هایی که برای هر روز ذخیره شده اند تاثیری بر روی بارش باران داشته باشند. حتی ممکن است این مجموعه برای اهداف دیگری تنظیم شده باشد.

بنابراین وظیفه ما به عنوان یک دانشمند داده این است که تحلیل های اولیه یا به اصطلاح pre processing را روی این مجموعه داده انجام دهیم.

در ادامه با نحوه انجام آن بیشتر آشنا خواهیم شد:

در مجموعه داده ستونی تحت عنوان تاریخ داریم که به صورت string ذخیره شده است. به عنوان مثال "2014/11/10". بنابراین اگر نیاز به استفاده از ویژگی تاریخ است، لازم است که آن را به صورت جداگانه به سال و ماه و روز تفکیک کنیم و خود Date را از ستون ها حذف می‌کنیم.

یک ویژگی داریم تحت عنوان "Unnamed: 0" که تاثیری در پیشبینی بارندگی ندارد. و احتمالا نقش Indexing را بازی می‌کند. بنابراین اقدام به حذف آن می‌کنیم.

ویژگی "Weather Station" نیز تاثیری بر روی پیشبینی بارش ندارد. بنابراین آن را نیز از مجموعه داده کنار می‌گذاریم.

به دنبال این هستیم که آیا می‌توانیم داده‌هایی را بیابیم که همبستگی شدیدی میان آنها وجود داشته باشد؟ زیرا که اگر چنین باشد، می‌توانیم از یکی از این ویژگی ها صرف نظر کنیم و از جمله مزیت های آن برای مدل می‌توان به جلوگیری از بروز بیش‌برازش، افزایش سرعت یادگیری و پیشبینی و ... اشاره کرد.

از نمودار heatmap برای چنین منظوری استفاده می‌کنیم. اگر n ویژگی داشته باشیم، نمودار heatmap یک نمودار n×n خواهد بود که دو به دو میزان همبستگی ویژگی ها را نمایش می‌دهد.

پس از بررسی نمودار، به این نتیجه رسیدیم که ویژگی های حداقل دما و حداکثر دما به ترتیب در ساعات ۹ صبح و ۳ بعد از ظهر از ایستگاه های مختلف گزارش می‌شوند. بنابراین می‌توان به این نتیجه رسید که این دو ویژگی عینا یکی هستند. پس از در نظر گرفتن حداقل و حداکثر دما در روز ها در یادگیری مدل صرف نظر می‌کنیم.

طبیعی به نظر می‌رسد که ماهی از سال که در آن قرار داریم، تاثیر زیادی بر روی پیش‌بینی بارش دارد. (تفاوت ماه‌های سرد سال با ماه‌های گرم) پس اجازه می‌دهیم این ویژگی در پیش‌بینی مدل نقش بازی کند.

پس از یک نگاه اجمالی به این مورد برخوردیم که در تعداد کثیری از ویژگی ها، با مقادیر زیادی از داده‌های گم‌شده روبه‌رو هستیم.

چنین رویکردی را در پیش خواهیم گرفت:

اگر تعداد داده های گمشده در یک ویژگی به خصوص، بیش از 50,000 داده بود، از آن ویژگی برای یادگیری مدل صرف نظر می‌کنیم.

اگر تعداد داده‌های گمشده در یک ویژگی به خصوص، بین 7,000 تا 50,000 بود، میانگین دیگر داده‌های در دسترس در آن ویژگی را، جایگزین مقادیر گم شده خواهیم کرد.

اگر تعداد داده‌های گمشده در یک ویژگی به خصوص، زیر 7,000 بود، آن داده‌ها را از مجموعه داده حذف خواهیم کرد (دقت کنیم که گفته شد داده. نه ویژگی یا فیچر)

الگوریتم‌های یادگیری ماشین در دسترس، ویژگی ها categorical را نمی‌پذیرند. بنابرین لازم است که آنها را با روشی به داده های عددی پیوسته تبدیل کنیم.

دو راه پیش‌رو داریم:

One-Hot Encoding

Labeling method

اگر ویژگی ما از نوع اسمی (categorical) باشد و ترتیب نیز داشته باشند مقادیر آن، می‌توانیم از روش دوم استفاده کنیم

در صورتی که ویژگی های اسمی ما ترتیبی نباشند، باید حتما از روش اول استفاده کنیم.

از آنجایی که تمام داده‌های اسمی ما ترتیبی نیستند، پس تماما از روش one-hot encoding بهره می‌بریم.

با این کار تعداد ستون های ما به تعداد قابل توجه ای افزایش می‌یابد.

پس باید با روشی سعی کنیم مهم‌ترین ستون‌ها را به عنوان ویژگی های اصلی پیدا کنیم که جلوتر توضیح داده شده است.

به عنوان یک گام مهم در پیش‌پردازش، همه ویژگی ها را در یک scale استاندارد قرار می‌دهیم تا سرعت پردازش بالا برود و تحلیل و کار با نمودار های بصری ساده تر شود.

لیبل های هدف ما نیز متوازن نیست زیرا که تعداد برچسب های No سه برابر تعداد برچسب های Yes است.

برای متوازن کردن آنها، از متد Resampeling استفاده می‌کنیم که تشریح آن در فایل نوت‌بوک پروژه آمده است.

حال شروع به ساخت مدل decision tree می‌کنیم. این الگوریتم بر استفاده از معیار بهره اطلاعاتی و entorpy اقدام به انتخاب ویژگی های موجود می‌کند که بهترین تفکیک را میان داده‌ها انجام دهند.

پس با اجرای همین الگوریتم می‌توانیم مهم ترین ویژگی ها را نیز بیابیم.

به همین ترتیب مدل های KNN و SVM را نیز ساخته و آموزش می‌دهیم.

زمان اجرای الگوریتم SVM بالاست. مخصوصا در شرایطی که تعداد داده‌ها و ویژگی ها زیاد باشد. در چنین شرایطی دو راه پیش رو داریم که می‌توانیم به صورت هم زمان نیز از آنها استفاده کنیم:

یک: داده های کمتری را برای یادگیری به ماشین ارائه دهیم.

دو: سعی کنیم با روشی (PCA) یا feature selection ابعاد داده را کاهش دهیم

در هر دو صورت دقت پیش‌بینی مدل پایین خواهد آمد اما سرعت پیشبینی بالا و قابل دسترس تر خواهد شد.

در اصل باید یک نسبتی بین کاهش دقت و سرعت درنظر بگیریم.

در نهایت برای هر یک از آنها معیار های صحت سنجی را انجام می‌دهیم.

هر کدام که معیار هایی داشتند که به یک نزدیکتر بودند، به این معنی است که پیش‌بینی های موفقتری روی داده های آزمایشی انجام داده‌اند و می‌توانند برای انتخاب به عنوان مدل نهایی شانس بالاتری داشته باشند.

با توجه به توضیحات بالا:

معیار های متفاوتی برای سنجش مدل های مختلف وجود دارد. اما به صورت کلی اگر فقط بخواهیم بر مبنای معیار f1-score تصمیم گیری کنیم، الگوریتم KNN مدل بهتری را برای پیشبینی بارش باران ارائه داده است (نمره نزدیک به یک)