

Evaluation

ارزیابی مدل :

خطایی که در پیش بینی وجود دارد را می توان به روش های مختلفی عیب یابی کرد. انتخاب روش های زیر به صورت تصادفی نیست و معیارهای خاصی دارد.

۱. جمع آوری داده ها آموزشی بیشتر
۲. استفاده از مجموعه ویژگی کوچکتر
۳. امتحان کردن ویژگی های اضافی
۴. استفاده از ویژگی های چند جمله ای
۵. افزایش یا کاهش مقدار λ

در ارزیابی مدل ممکن است خطا کم باشد ولی مدل همچنان پیش بینی درستی به خاطر وجود مشکل Overfitting نکند.

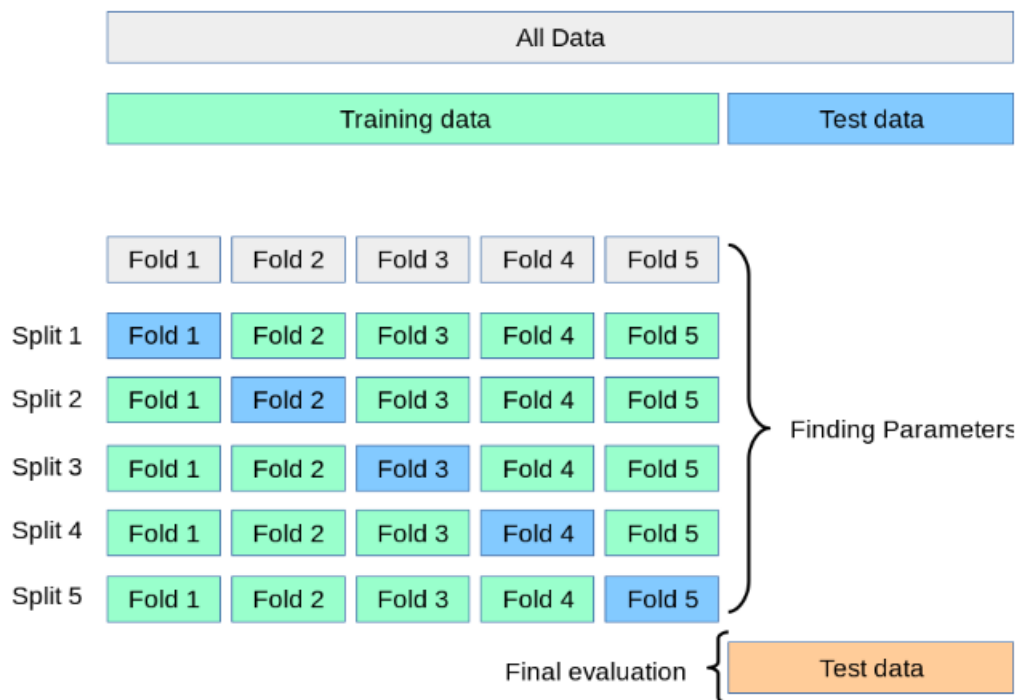
به همین خاطر داده ها را به دو دسته آموزش و تست تقسیم بندی می کنیم.

: Cross Validation

تکنیکی برای ارزیابی یک مدل یادگیری ماشین و آزمایش عملکرد آن است و به ما کمک می کند که کدام مدل را برای مدل سازی انتخاب نماییم. تا به اینجا با یکی از روش های اعتبارسنجی مدل آشنا شدیم که در آن روش داده ها را به داده های آموزش و تست تقسیم بندی می کردیم.

این روش دو مشکل بزرگ دارد. اول آن که عملکرد این روش بسیار بسته به آن است که تعداد مجموعه تست چه مقدار باشد. دوم آن که مدل بر روی تمام داده ها Train و Test نمی شود.

روش بهتر برای غلبه بر این مشکلات ، k-fold است. در این روش داده ها را به بخش هایی با نام fold تقسیم می کنیم. تعداد این بخش ها را می توانیم تعیین کنیم برای مثال اگر ۵ قرار دهیم یعنی آن که کل داده ها را به ۵ قسمت تقسیم کنیم و ۵ سری مدل را آموزش دهیم که در این ۵ سری آموزش برای مثال ۲۰ درصد داده ها را به عنوان تست و ۸۰ درصد را به عنوان آموزش در نظر می گیریم. در هر بار آموزش مدل یکی از این ۵ قسمت داده ها به عنوان تست خواهد بود و درنهایت همه داده ها یکبار به عنوان داده تست در نظر گرفته شده اند. این امر باعث می شود تا اثر شانس در فرآیند آموزش مدل کمتر شود و مدل به خوبی آموزش ببیند.



مزایای روش k-fold را بررسی کردیم. اما مشکل بعدی که به وجود می آید نشت (Leak) داده های تست به داده های آموزش است. به همین دلیل بخشی از داده ها را به عنوان داده های تست از ابتدا جدا می کنند تا هنگام ارزیابی نهایی از آن ها استفاده کنند و مابقی داده ها را به روش k-fold برای آموزش مدل و پیدا کردن بهترین پارامترها تقسیم بندی می کنند. به این امر Cross Validation می گویند.

این روش برای جلوگیری از Overfitting نیز کاربرد دارد.

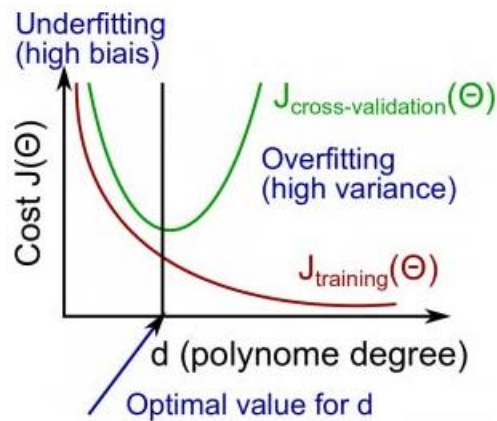
: The Bias/Variance Tradeoff 🌈

: Bias ❖

بایاس به دلیل فرضیات غلط رخ می دهد. برای مثال اگر مدل درجه دو را به صورت خطی در نظر بگیریم با بایاس بالا احتمالاً دچار Underfitting شده است.

❖ Variance :

واریانس به دلیل حساسیت بیش از حد مدل به تغییرات کوچک در داده های آموزشی رخ می دهد. مدلی با درجات زیادی از آزادی احتمالا دارای واریانس بالا بوده و دچار Overfitting شده است.



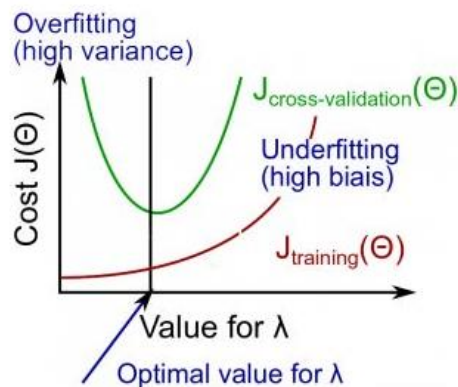
ما به دنبال تعادل بین بایاس و واریانس هستیم.

❖ اثر λ روی بایاس و واریانس :

به صورت کلی داریم :

λ بزرگ باعث بایاس زیاد می شود. همچنین هزینه آموزش و CV زیاد می شود.

λ کوچک باعث واریانس زیاد می شود. همچنین هزینه آموزش کم و هزینه CV زیاد می شود.



❖ اثر اندازه Train set بر بایاس و واریانس :

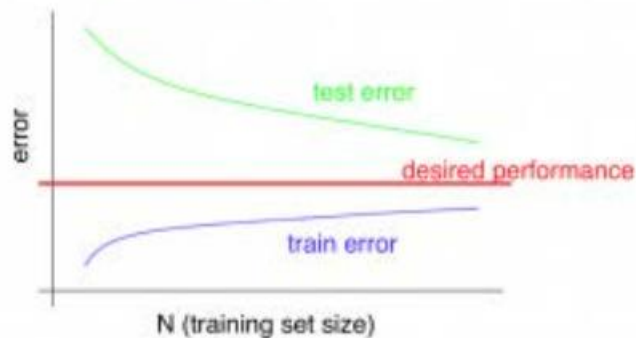
More on Bias vs. Variance

Typical learning curve for high bias (at fixed model complexity):



More on Bias vs. Variance

Typical learning curve for high variance (at fixed model complexity):



نکته ! با افزایش اندازه Train set احتمالاً مشکل واریانس زیاد حل می شود اما افزایش اندازه Train set به تنهایی مشکل بایاس زیاد را حل نخواهد کرد.

می توان به صورت کلی برای حل مشکل بایاس و واریانس اقدامات زیر را انجام داد :

- جمع آوری نمونه آموزشی بیشتر برای اصلاح واریانس زیاد
- استفاده از مجموعه کوچکتری از ویژگی ها برای اصلاح واریانس زیاد
- اضافه کردن ویژگی برای اصلاح بایاس زیاد
- اضافه کردن ویژگی های چندجمله ای برای اصلاح بایاس زیاد
- کاهش λ برای اصلاح بایاس زیاد

- افزایش λ برای اصلاح واریانس زیاد

✚ معیارهای ارزیابی :

در مسائل طبقه بندی گاهی کلاس های مختلف به صورت متوازن نیستند. یعنی از یک کلاس نمونه های خیلی بیشتری وجود دارد. این امر باعث می شود تا معیارهای ارزیابی مختلفی برای بررسی دقت عملکرد مدل استفاده شود.

❖ Confusion Matrix :

در مسائل طبقه بندی برای بررسی میزان خطا معیارهای متفاوتی وجود دارد. برای مثال تصور کنید می خواهیم سرطان داشتن افراد را طبقه بندی و پیش بینی کنیم. اگر ۱ را به عنوان Label برای افرادی که سرطان دارند در نظر بگیریم و صفر را برای افرادی که سرطان ندارند ، داریم :

پیش بینی شده: ۱، حقیقی: ۱ \Leftarrow True positive
 پیش بینی شده: ۰، حقیقی: ۰ \Leftarrow True negative
 پیش بینی شده: ۰، حقیقی: ۱ \Leftarrow False negative
 پیش بینی شده: ۱، حقیقی: ۰ \Leftarrow False positive

داریم :

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

❖ Precision :

Precision بیان می کند از تمام بیمارانی که پیش بینی برای آن ها مقدار ۱ بوده است ، چه بخشی از آن ها واقعا مبتلا به سرطان هستند.

$$\frac{\text{True Positives}}{\text{Total number of predicted positives}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False positives}}$$

❖ Recall :

Recall بیان می کند از تمام بیمارانی که واقعا مبتلا به سرطان هستند ، چه بخشی از آن ها توسط مدل درست تشخیص داده شده اند.

$$\frac{\text{True Positives}}{\text{Total number of actual positives}} = \frac{\text{True Positives}}{\text{True Positives} + \text{False negatives}}$$

نکته ! هدف ما این است که هر دو معیار مقادیر زیادی داشته باشند. یعنی دقت عملکرد مدل بالا باشد.

دقت عملکرد از رابطه زیر بدست می آید :

$$Accuracy = \frac{\text{true positive} + \text{true negative}}{\text{total population}}$$

❖ F1_Score :

F1_Score رابطه ای بین Precision و Recall درست می کند یا به عبارت دیگر بین این دو معیار توازن ایجاد می کند.

$$FScore = 2 \frac{PR}{P + R}$$