تمرین سری 2

بخش 2:

1.رگرسیون لسو و ریج نوعی از رگرسیون هستند که از متد shrinkage استفاده میکند.
Shrinkage هم زمانی اتفاق میفتد که مقادیر داده ها به سمت یک نقطه مرکزی مانند میانگین جمع میشوند.

در رگرسیون اگر تعداد متغیرها زیاد باشد ممکن است overfitting رخ دهد و اگر تعداد آنها کم باشد ممکن است underfitting رخ دهد؛ یکی از راه های مقابله با این استفاده از رگرسیون ridge است.
رگرسیون ridge سعی میکند ضرایبی را پیدا کند که به بهترین صورت به معادله بخورند.او سعی میکند به جای RSS ، معادله زیر را مینیمم کند که قسمت سمت چپ آن همان RSS است و سمت راست آن پنالتی shrinkage است و زمانی کوچک میشود که B1... Bp نزدیک صفر باشند. λ هم پاراکتر تنظیم است برای کنترل میزان تاثیر این دو بر روی تخمین ضرایب رگرسیون است.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2,$$

رگرسیون ridge یک عیب دارد آن هم اینکه تمام p پیش بینی را در مدل نهایی در نظر میگیرد.

رگرسیون Lasso رگرسیونی است که این عیب را ندارد و جایگزین خوبی است.
لسو از پنالتیL1 به جایL2 استفاده میکند. در اینجا نیز لسو مانند ریج سعی میکند ضرایب را به سمت صفر ببرد اما لسو برخلاف ریج میتواند ضرایب را دقیقا برابر ضفر کند هنگامی که ضریب لامبدا خیلی بزرگ است.

$$\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| = \mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|.$$

در جاهایی که تعداد متغیرها کم است لسو بهتر عمل میکند. از cross validation میتوان برای تخمین اینکه کدام روش روی مجموعه داده ی حاضر بهتر عمل میکند استفاده کرد.

منبع : کتاب + جزوهو

2.یک راه حل این است که نموداری از ضرایب برای مقادیر مختلف λ بکشیم و به دنبال کوچکترین مقدار λ هستیم در جایی که مقادیر مختلف ضرایب تثبیت شوند. کمترین مقدار λ را میخواهیم چون باعث کمترین مقدار بایاس میشود.

راه حل دیگر استفاده از k fold cross validation است به ازای مقادیر مختلف λ . خطای CV را در هر مرحله حساب میکنیم و جایی که خطای کمتری داشته باشد λ مورد نظر ماست.

منبع :

https://www.real-statistics.com/multiple-regression/ridge-and-lasso-regression/estimating-ridge-regression-lambda

3. افزایش تعداد فولدها باعث کاهش بایاس و افزایش واریانس میشود.
اگر با تعداد داده کمی کار داریم میتوانیم تعداد فولدها را افزایش دهیم تا با داده بیشتری را train کنیم و بایاس و خطا را کاهش دهیم.

منبع : https://sebastianraschka.com/faq/docs/number-of-kfolds.html

4. LOVCV برای تعداد داده های کم مناسبت است. یک نوعی از k fold است که k تعداد مثالهای داخل دیتاست است. یعنی مدل روی تمام داده ها غیر از یکی train میشود و سپس مقدار آن نقطه ی جامانده را تخمین میزند.

https://www.cs.cmu.edu/~schneide/tut5/node42.html

https://machinelearningmastery.com/loocv-for-evaluating-machine-/learning-algorithms

5. در bootstrapping از یک مجموعه داده با جایگذاری نمونه برداری میکنیم. این کار اجازه میدهد که یک داده بیش از یکبار ظاهر شود و از یک دیتاست نمونه های فراوانی بدست آوریم. سایز نمونه و تعداد تکرار را باید تعیین کنیم. تعداد تکرار باید به قدری بزرگ باشد که بتوانیم آمار معنی داری بدست آوریم.
با افزایش تعداد نمونه های بوت استرپ واریانس کاهش میابد.
k-fold هم از نظر بایاس و هم و واریانس بهتر است.

https://machinelearningmastery.com/a-gentle-introduction-to-the-/bootstrap-method

6. 5 times 2 fold cross validation

Cross validation 2 fold است که داده ها به 2 دسته تقسیم میشوند سپس محاسبه میشوند مانند اما تفاوتش در این است که اینکار 5 بار تکرار میشود

N times k fold CV