فرآیند Regularization

دوره پایتون و یادگیری ماشین

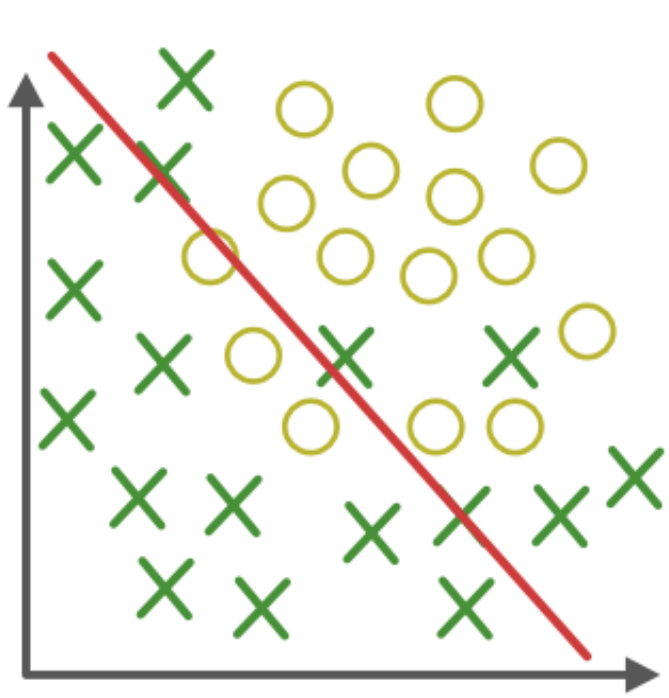
</Monolearn>

مشکلات مدل یادگیری ماشین

- Overfitting یا بیش برازش: زمانی اتفاق می افتد که مدل یادگیری ماشین، بر روی داده ها به خوبی آموزش ببیند (خطا بر روی داده های train کم و میزان دقت بالا) ولی بر روی داده های تست به خوبی عمل نکند (خطا بر روی داده های test زیاد و میزان دقت پایین).

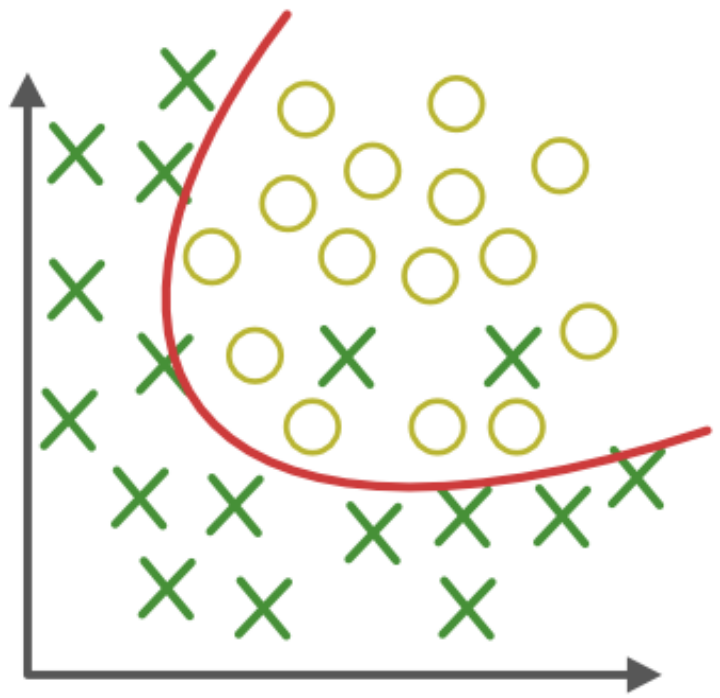
- Underfitting یا کم برازش: مدل بر روی داده های train به خوبی آموزش نبیند در نتیجه میزان خطا هم بر روی داده ها train و هم test مقدار بالایی خواهد بود و میزان دقت بر روی هر دو مجموعه داده، مقدار کمی خواهد بود. (مدل نه خوب آموزش می بیند و نه خوب عمل یا پیش بینی می کند.)

مشکلات مدل یادگیری ماشین

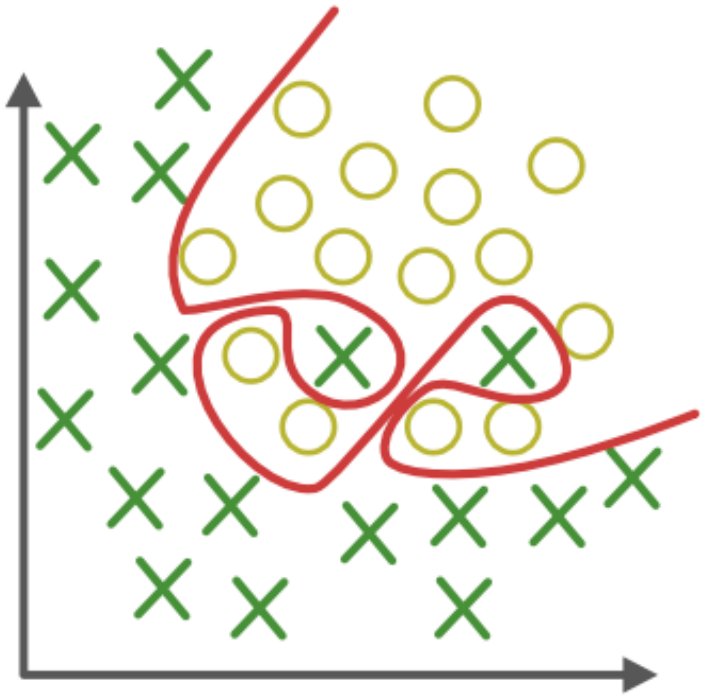


Under-fitting

(too simple to explain the variance)



Appropirate-fitting



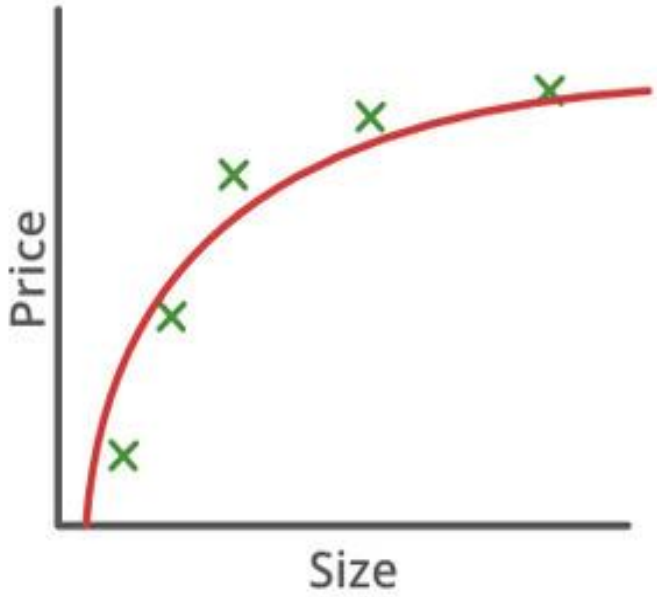
Over-fitting

(forcefitting--too good to be true)

مشکلات مدل یادگیری ماشین

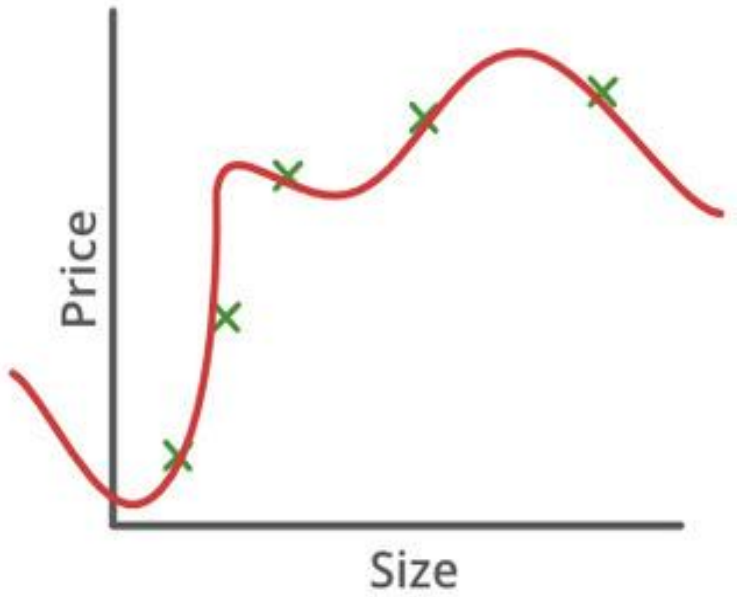
$\theta_0 + \theta_1 x$

High bais (underfit)



$\theta_0 + \theta_1 x + \theta_2 x^2$

High bais (underfit)



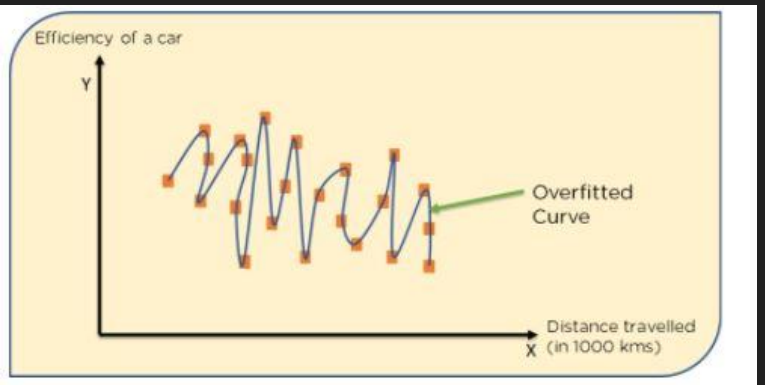
$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

High variance (overfit)

Overfitting یا بیش برازش

- زمانی اتفاق می افتد که مدل یادگیری ماشین بر روی داده های آموزشی بیش از حد آموزش ببیند و در نتیجه در تست با مشکل مواجه شود و نتواند خوب پیش بینی کند.
- در این مورد، مدل یادگیری ماشین، نویز موجود در داده ها را نیز یاد می گیرد.

- چه مواقعی اتفاق می افتد؟
  - مدل بیش از حد پیچیده است و ویژگی های همخط (Collinear) را در بر می گیرد که واریانس داده ها را افزایش می دهد.
  - تعداد ویژگی های داده ها بیشتر یا برابر با تعداد داده هاست (تعداد ویژگی ها بالاست)
  - حجم داده ها کم یا بسیار کم است
  - داده پیش پرادزش نشده، تمیز نیست، نویز دارد



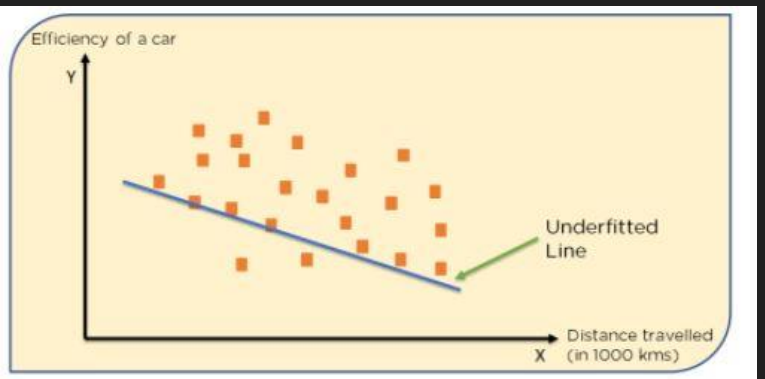
روش های مقابله با Overfitting

- اعمال روش K-fold cross validation
- افزایش داده ها (جمع آوری داده های بیشتر و همچنین، ایجاد داده های ساختگی)
- انتخاب ویژگی ها به طور موثر
- نظم دهی یا Regularization (L۱ و L۲) و (همچنین استفاده از تکنیک Drop Out)

- حدف لایه ها از مدل (از شبکه های عصبی در دیپ لرنینگ)
- توقف زود هنگام عمل یادگیری (در شبکه های عصبی در دیپ لرنینگ)

Underfitting یا کم برازش

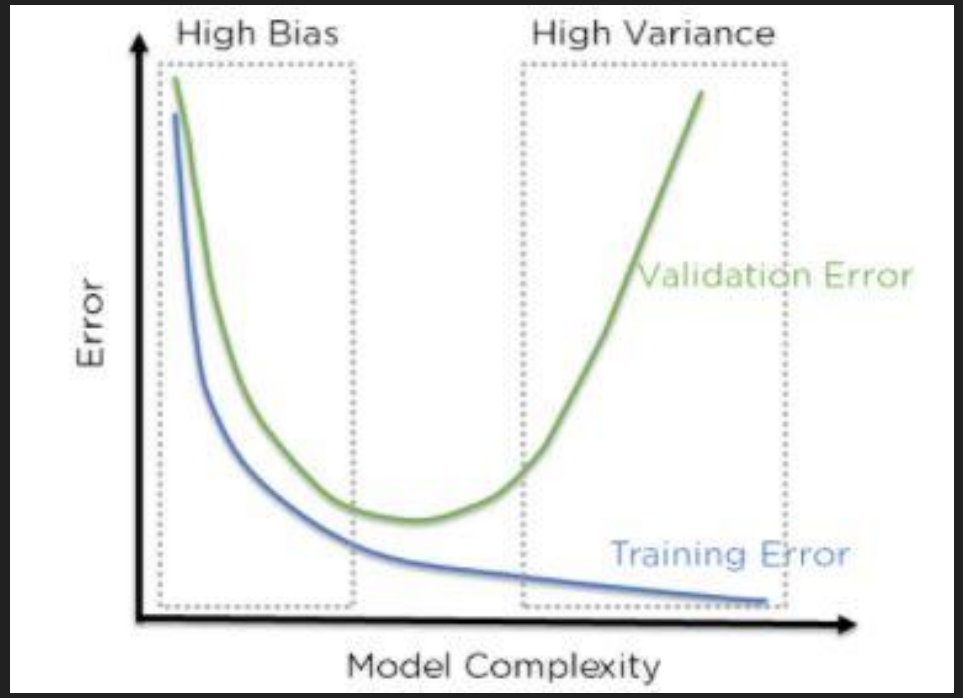
- زمانی اتفاق می افتد که مدل یادگیری ماشین به اندازه ی کافی پیچیده نباشد که بتواند روابط معنا دار میان ویژگی ها و متغیر هدف را به خوبی یاد بگیرد.
- در این حالت مقدار Bias بالا و واریانس پایینی داریم.
- مدلی که دارای Underfitting باشد، به خوبی آموزش نمی بیند و هم بر روی داده های train و هم test به خوبی عمل نمی کند.



روش های مقابله با Underfitting

- کاهش تنظیم (Decreasing Regularization): با کاهش میزان Regularization پیچیدگی و تنوع مدل بیشتر می شود و امکان آموزش بهتر فراهم می شود.
- افزایش زمان آموزش بر روی داده
- انتخاب ویژگی ها به طور موثر

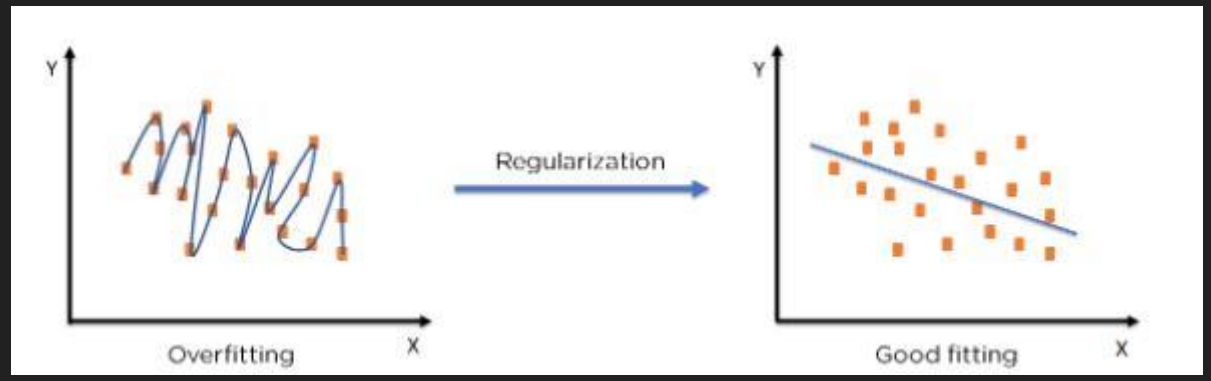
بایاس و واریانس

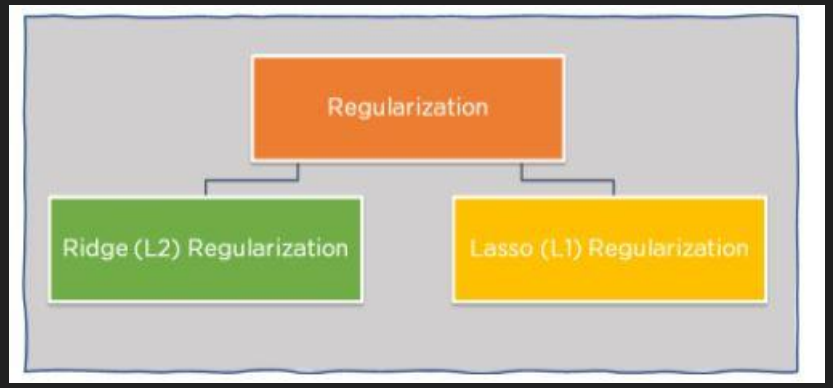


- بایاس زمانی اتفاق می افتد که مدل یاگیری ماشین، قابلیت انعطاف پذیری کمی برای یادگیری داشته باشد. چنین مدلی، توجه زیادی به داده ها می کند و قوانین را بیش از حد ساده سازی می کند. در نتیجه، میزان خطا بر روی داده های train و prediction افزایش می یابد.
- واریانس میزان حساسیت مدل یادگیری ماشین بر روی داده ها را مشخص می کند. یک مدل یادگیری ماشین با واریانس بالا، توجه زیادی به داده های آموزشی می کند و نمی تواند به خوبی قوانین را کلی سازی یا Generalize کند در نتیجه خطای یادگیری بر روی prediction بالا است.(خطای یادگیری در این حالت بر روی train کم است. چون داده های train را خیلی خوب یاد می گیرد یا بهتر بگوییم، بیش از اندازه خوب یاد می گیرد.)
- پس میزان بایاس و واریانس باید به درستی انتخاب شوند که در کنار یکدگیر به فرآیند یادگیری کمک کنند.

نظم دهی یا Regularization

- این روش باعث اعمال محدودیتی هایی بر روی مدل یادگیری ماشین می شود تا از Overfitting یا Underfitting مدل جلوگیری کند.
- در نظم دهی نوع L۱ یا L۲ می توانیم برای تابع زیان یا همان Loss function یک مجازات در نظر بگیریم تا ضرایب Coefficient ها را به سمت صفر سوق دهد. نظم دهی نوع L۲ اجازه می دهد وزن ها به سمت صفر پیش بروند اما صفر نشوند. در حالیکه در L۱ وزن ها به صفر می رسد.





L1 Regularization یا Lasso Regression یا نُرم ۱

- تصور کنید الگوریتمی نیاز داریم تا رتبه کنکور فردی را بر اساس سوابق او پیشبینی کند. طبیعی است که همه ویژگی های در سابقه یک فرد تاثیر یکسانی در کسب رتبه ندارند. مثلاً معدل نمرات فرد در پیشبینی بیشتر به کار می آید تا سوابق فعالیت های داوطلبانه و خارج از درس، و یا میانگین رتبه ها در منطقه آن فرد هم موثرتر از شاخص فیزیکی BMI اوست. پس با کمک نُرم L1 در حین آموزش به ویژگی های کمتر موثر وزن بسیار بسیار کوچکی (نزدیک به صفر) اعمال می شود چرا که تاثیر آن ها کمتر است. به طور مثال وزن مربوط به میانگین رتبه ها در منطقه به سمت عددی غیر صفر میل می کند، اما وزن شاخص BMI فرد مدام کوچکتر می شود تا به صفر برسد.
- قسمت قرمز رنگ، مقدار پنالتی یا جریمه را مشخص می کند. در این قسمت، مجموع قدر مطلق همه ی وزن ها در یک مقدار ثابت (لاندا) ضرب می شود. با تغییر لاندا، الگوریتم سعی دارد تا میزان تابع هزینه را کمینه کند.

Cost function:

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} |\theta_i|$$

L۲ Regularization یا Ridge Regression یا نُرم ۲

- در Ridge، تابع هزینه با توجه به مجموع مجذور وزن ها بدست می آید.
- در Ridge، مانند Lasso، سعی الگوریتم بر کوچک کردن وزن ها هست، اما بر خلاف Lasso، در این حالت وزن ها را صفر نمی کند.
- هنگامی که داده های Outlier در دیتاست موجود باشد، این روش به خوبی عمل نمی کند چرا که در نقاط Outlier خطای پیش بینی مدل بسیار زیاد است و با داشتن جمله ی پنالتی مانند زیر، وزن ها کوچکتر خواهد شد.
- مدل یادگیری ماشینی که از Ridge استفاده می کند، زمانی بهتر عمل می کند که تمامی ویژگی های داده ورودی روی پیش بینی هدف یا خروجی تاثیر گذار بوده و همچنین، وزن های داخل مدل به طور مساوی مقداردهی اولیه شده باشند.

Cost function:

$$L(x,y) \equiv \sum_{i=1}^{n} (y_i - h_\theta(x_i))^2 + \lambda \sum_{i=1}^{n} \theta_i^2$$

مقایسه L1 و L2

- نُرم L1
  - مجموع قدر مطلق وزن ها را به هزینه/خطا اضافه می کند.
  - نُرم L1 یک مدل خلوت تر ایجاد می کند.
  - ممکن است براساس اثردهی زیرمجموعه های متفاوت از ویژگی ها، مدل با نُرم L1 چند یادگیری متفاوت ایجاد کند.
  - با نُرم L1خاصیت feature selectionحاصل می شود.
  - نُرم L1در مواجهه با نقاط outlier در داده های ورودی، عملکرد بهتری و مقاوم‌تری دارد.
  - نُرم L1 می تواند مدلی ساده و قابل تفسیر ایجاد کند اما همین سبب می شود تا الگوریتم قادر نباشد الگوهای پیچیده را یاد بگیرد.

- نُرم L2
  - مجموع مجذور وزن ها را به هزینه/خطا اضافه می کند.
  - نُرم L2 یک مدل خلوت ایجاد نمی کند.
  - نُرم L2 تنها یک یادگیری/راه حل در الگوریتم یادگیری ماشین ایجاد می کند و براساس استفاده از زیرمجموعه های متفاوت از ویژگی ها عمل نمی کند.
  - با نُرم L2خاصیت feature selection حاصل نمی شود.
  - نُرم L2 در مواجهه با نقاط outlier در داده های ورودی، عملکرد خوبی ندارد.
  - زمانی که مقدار هدف پیش‌بینی تابعی از همه ویژگی ها داده ورودی باشد، نُرم L2یادگیری بهتری را سبب می شود.
  - با نُرم L2 بر خلاف نُرم L1، می توان الگوهای پیچیده را در داده ورودی یاد گرفت.

ترکیب دو روش قبل

- آیا بهتر نیست هر دو را با همدیگر استفاده کنیم تا ضعف های یکدیگر را پوشش دهند؟ آیا منجر به نتیجه بهتر خواهد شد؟ جواب مثبت است. به الگوریتم Elastic net regularization که ترکیبی از هر دو مورد L1 و L2 regularization را استفاده می کند نگاهی بیاندازید!

پایان

با تشکر از توجه تان، اوقات خوشی را برایتان آرزومندم.