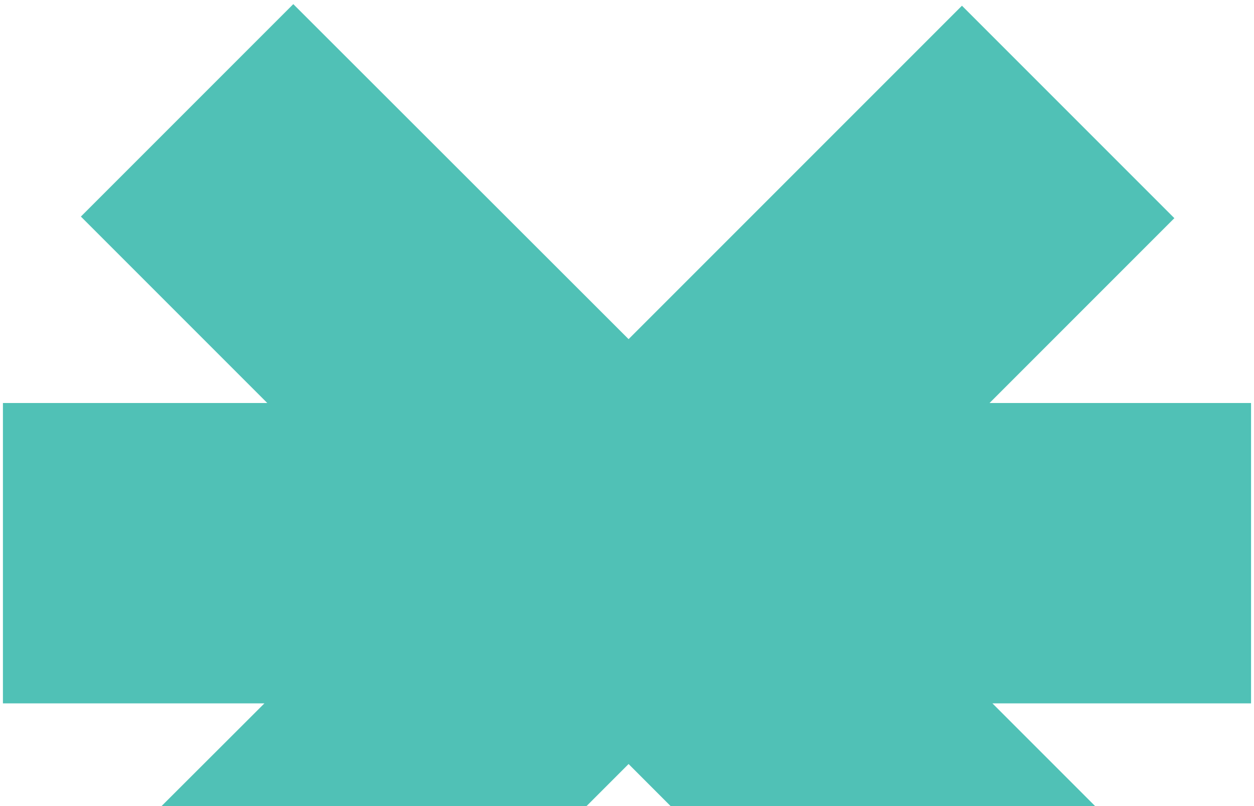
MÓDULO 2
Aprendizagem supervisionada
Especialização em Machine Learning

4

Regularização

New Technology School

Tokio.



4 Regularização

Sumário

New Technology School

Tokio.

4.1 Desvio-padrão e variância

No momento de avaliar um modelo já treinado, podem apresentar-se três situações:

- **Subajuste:** o modelo não possui a precisão suficiente, pois não pôde ajustar corretamente a curva, segundo os exemplos usados.
- **Sobreajuste:** o modelo apresenta uma precisão excelente, um ajuste quase perfeito aos exemplos, mas parece um modelo irreal, que não funcionará corretamente perante novos casos.
- **Ajuste correto:** o modelo conta com uma precisão bastante boa e é capaz de ajustar a curva suficientemente bem para realizar novas predições.

SUBAJUSTE

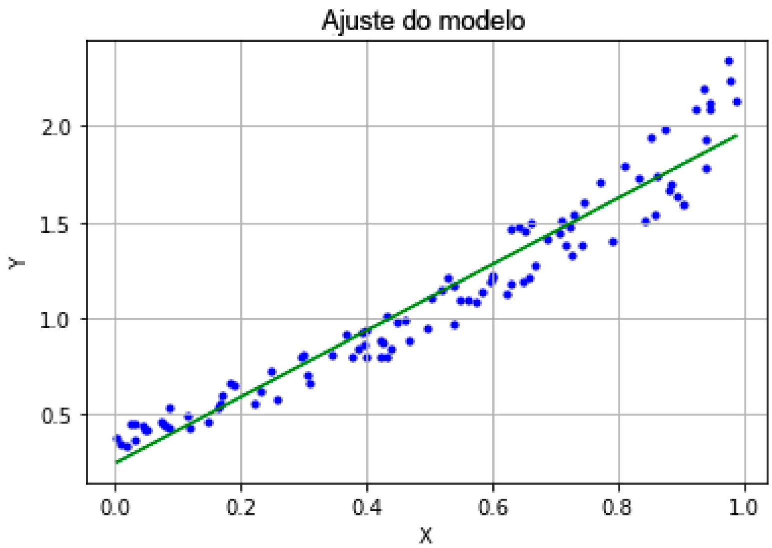


Figura 12

SOBREAJUSTE

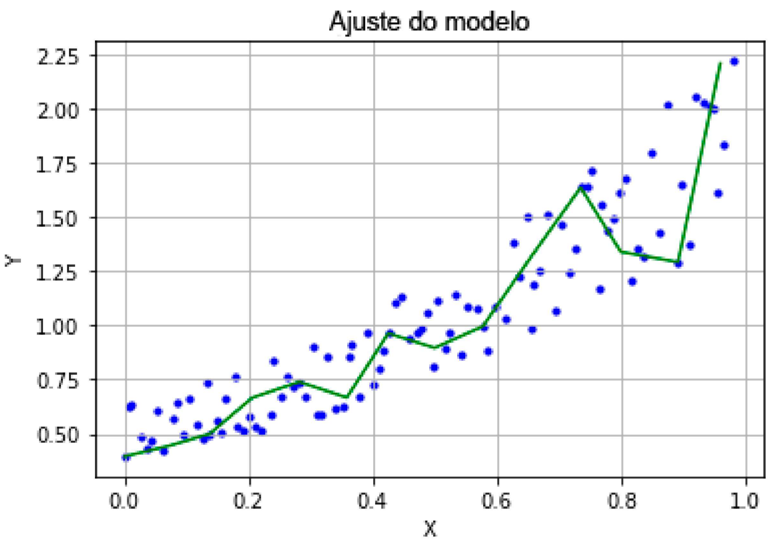


Figura 13

AJUSTE CORRETO

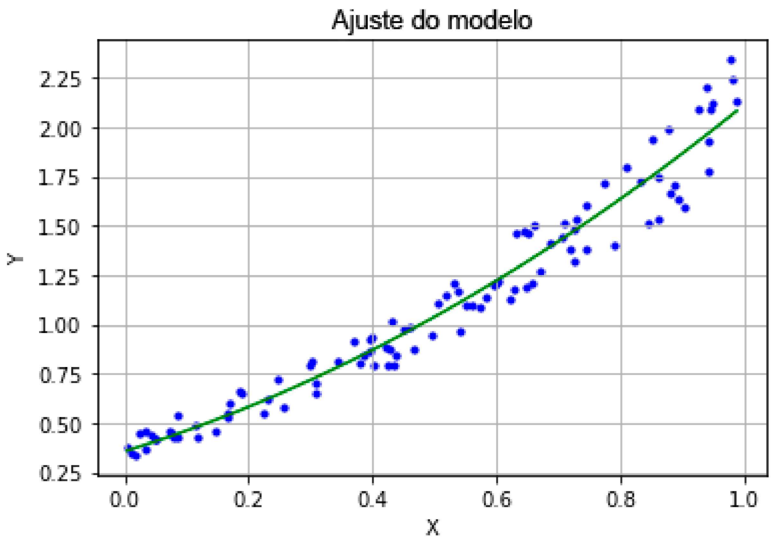


Figura 14

É especialmente importante a questão do sobreajuste, uma vez que nem sempre o melhor ajuste ou precisão possíveis são os desejados para o nosso modelo.

Imaginemos, por exemplo, um modelo que não extrai qualquer informação sobre os exemplos e as suas características, mas conta simplesmente com uma regra para recordar cada valor de *Y* em função de cada valor de *X* que "viu" no seu treino. Logo, segue um código semelhante a "*if* x == '42': y = 3,14":

- Teria uma boa precisão como modelo? Sim, de 100%, é capaz de memorizar os valores para cada exemplo.
- Serviria de algo na vida real? O que aconteceria quando fossemos usá-lo para realizar predições sobre valores diferentes àqueles que tinha treinado e que não tinha visto anteriormente? Não seria uma modelo possível de usar, pois devolveria valores aleatórios.

O sobreajuste ou *overfitting* é produzido quando o nosso modelo se ajusta demasiado aos dados sobre os quais treinou. Logo, pode não generalizar bem, perante novos dados, quando tiver de realizar predições: por vezes, seguirá uma curva demasiado complexa ou até irreal, de forma que, se a avaliarmos, podemos detetar que provavelmente os valores, na vida real, não vão seguir uma curva com tantos altos e baixos.

Nesse caso, necessitamos de treinar o nosso modelo para que consiga uma boa precisão sobre os dados de treino, mas que não confie cegamente neles e possa generalizar para novos dados no futuro.

Para esse efeito, temos duas opções:

- Reduzir o número de características usadas no modelo. Talvez não contemos com o número suficiente de dados para treinar um modelo de tal complexidade, pelo que podemos reduzi-lo a menos características, cujos pesos devemos otimizar.
- "Regularizar" o modelo ou penalizar, em parte, o ajuste das características para as suavizar ou para que adotem valores menos extremos.

Uma recomendação sobre a relação entre o número mínimo de exemplos e o número de características que podemos modelar pode ser $m \cong n\ (x10^2)$.

4.2 Regularização

A regularização consiste em penalizar os valores extremos de Θ suavizando-os, ou seja, evitar que adote valores demasiado extremos, que o modelo sobreajuste a reta da equação.

Embora isto signifique, em parte, conseguir que um modelo possua menor precisão, o que nos ajudará a encontrar um ponto intermédio entre sobreajustar o modelo, aos valores sobre os que se treina, e obter um modelo que possa generalizar bem perante novos valores.

Para isso, adicionaremos um fator de regularização à função de custo, segundo um parâmetro λ, que poderemos modificar para conseguir mais ou menos regularização:

$$J_\theta = \frac{1}{2m} [\sum\limits_{i=0}^{m} (h_\theta(x^i)-y^i)^2 + \lambda \sum\limits_{j=1}^{n} \theta_j^2]$$

Figura 15

Código Latex:

J_\theta = \frac{1}{2m} [\sum\limits_{i=0}^{m} (h_\theta(x^i)-y^i)^2 + \lambda \sum\limits_{j=1}^{n} \

4.3 Função de custo regularizada

Ao regularizar a função de custo, também regularizamos as atualizações dos parâmetros de Θ, segundo as suas derivadas parciais do custo total:

$$
\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_0^i \\
\theta_j := \theta_j - \alpha [\frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_j^i + \frac{\lambda}{m} \theta_j] \\
\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_j^i \\
j \in [1, n]
$$

Figura 16

Código Latex:

```
\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_0^i \\

\theta_j := \theta_j - \alpha [\frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_j^i + \frac{\lambda}{m} \theta_j] \\

\theta_j := \theta_j (1 - \alpha \frac{\lambda}{m}) - \alpha \frac{1}{m} \sum_{i=0}^{m}(h_\theta (x^i) - y^i) x_j^i \\

j \in [1, n]
```

Devemos recordar que não regularizamos θ_0, já que corresponde a $x_0 = 1$ ou ao nosso parâmetro de *bias* ou *intercept*.

Contudo, tal como no rácio de aprendizagem α, estamos a adicionar outro hiperparâmetro ao nosso modelo: como escolhemos o parâmetro de regularização (λ) idóneo?