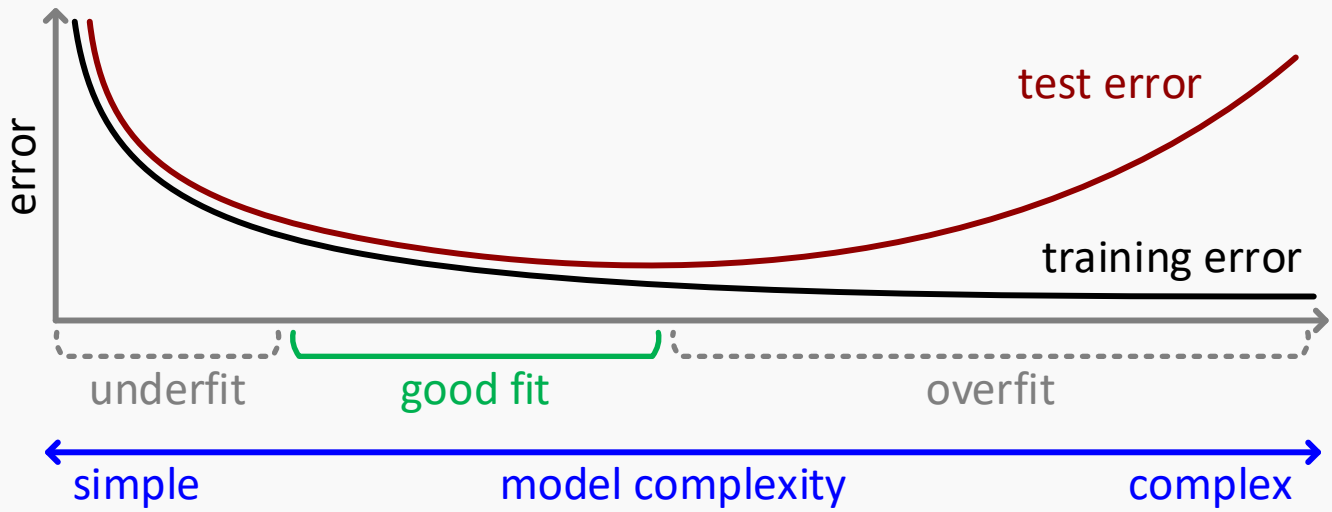
statistical learning theory 📈 for supervised learning

Ockham's Razor states the best models are simple models that fit the data well.

William of Ockham, English fryer and philosopher (1287-1347) said that among hypotheses that predict equally well, choose the one with the fewest assumptions.

The key to understanding Statistical Learning Theory:



Therefore, the goal is obtain a **balance of accuracy and simplicity**.

Most common machine learning methods choose f to **minimize training error and complexity**.

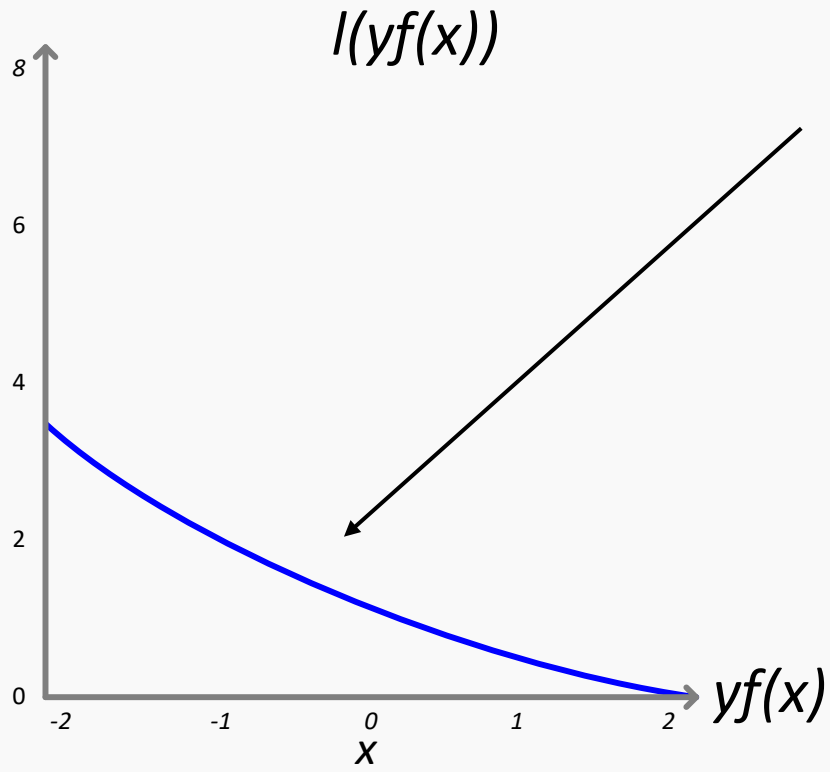
The ultimate goal is to mitigate what is known as the **"curse of dimensionality"**

> The curse of dimensionality is the tendency to overfit a dataset when many features are available, but not enough data to compensate for predictive purpose. Therefore, the amount of data available would need to increase exponentially to prevent models overfitting.

Simplicity is measured in multiple ways and referred often to as **regularization**:

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i f(x_i)) + \text{Regulatization}(f)$$

logistic regression ⬍ for machine learning



$$\min_{\text{models } f} \log\left(1 + e^{-yf(x)}\right)$$

$$f(x_i) = \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} = \sum_{j=1}^{p} \beta_j x_{ij}$$

$$\min_{\beta_1, \beta_2, \ldots, \beta_p} \frac{1}{n} \sum_{i=1}^{n} \left(1 + e^{-y \sum_{j=1}^{p} \beta_j x_{ij}}\right)$$

Constructing Models with R

- Wide range of models available in R
- R models are defined with expressive formula language
- A formula is a language within a language
- A formula is ubiquitous across most R model types
- Labels and features can be both numeric or categorical (factor)

R Formula Language

```
lm(x~y, data = df) #The Linear Relationship between x and y
lm(x~y+z, data = df) #Adding another feature
lm(x~., data = df) # “.” indicates the usage of all features
lm(x~. - z, data = df) #To use all features except z
lm(x~0 + y + z, data = df) #The “0” drops the intercept term
lm(x~y + I(y^2) + z + I(z^2), data = df) #Utilizes function I(x)
lm(x~0 + y + z + y:z, data = df) #Adds an Interaction Term y:z
```