CPEN 355 – Lecture 3: Linear Regression

Mirza Sarwar, Ph.D.

Department of Electrical and Computer Engineering

University of British Columbia

Recap

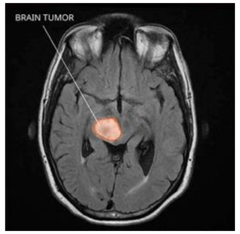
https://stats.libretexts.org/Bookshelves/Probability_Theory/Probability_Mathematical_Statistics_and_Stochastic_Processes_(Siegrist)/04%3A_Expected_Value/4.01%3A_Definitions_and_Basic_Properties

What is machine learning?

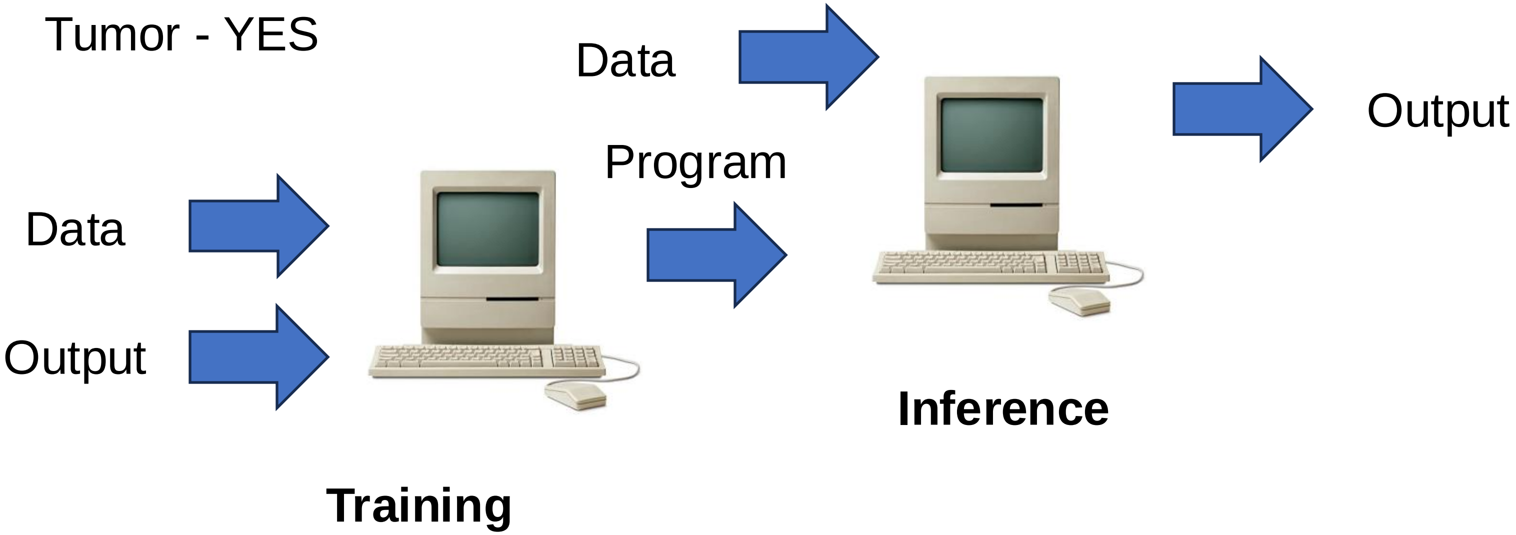
Formally (Mitchell 1997):

Algorithms that *improve* on some *task* with *experience*

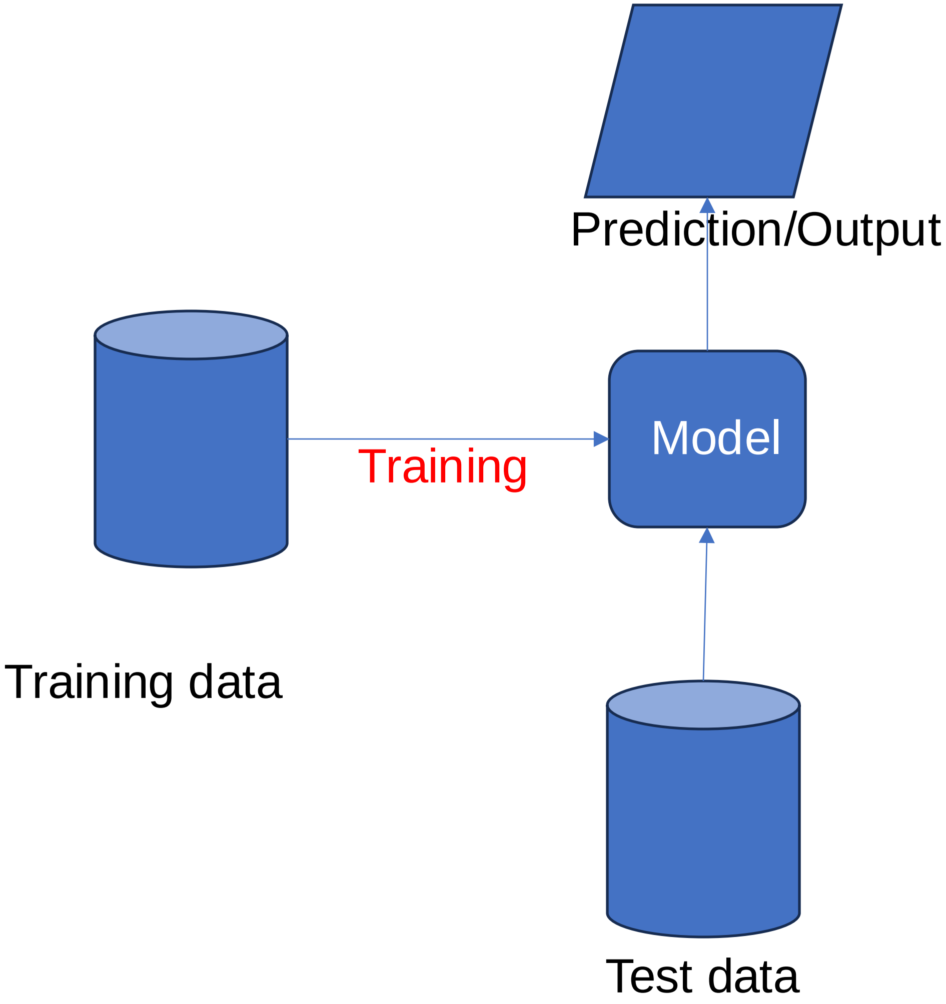
MRI image



Tumor - YES



Training and Testing

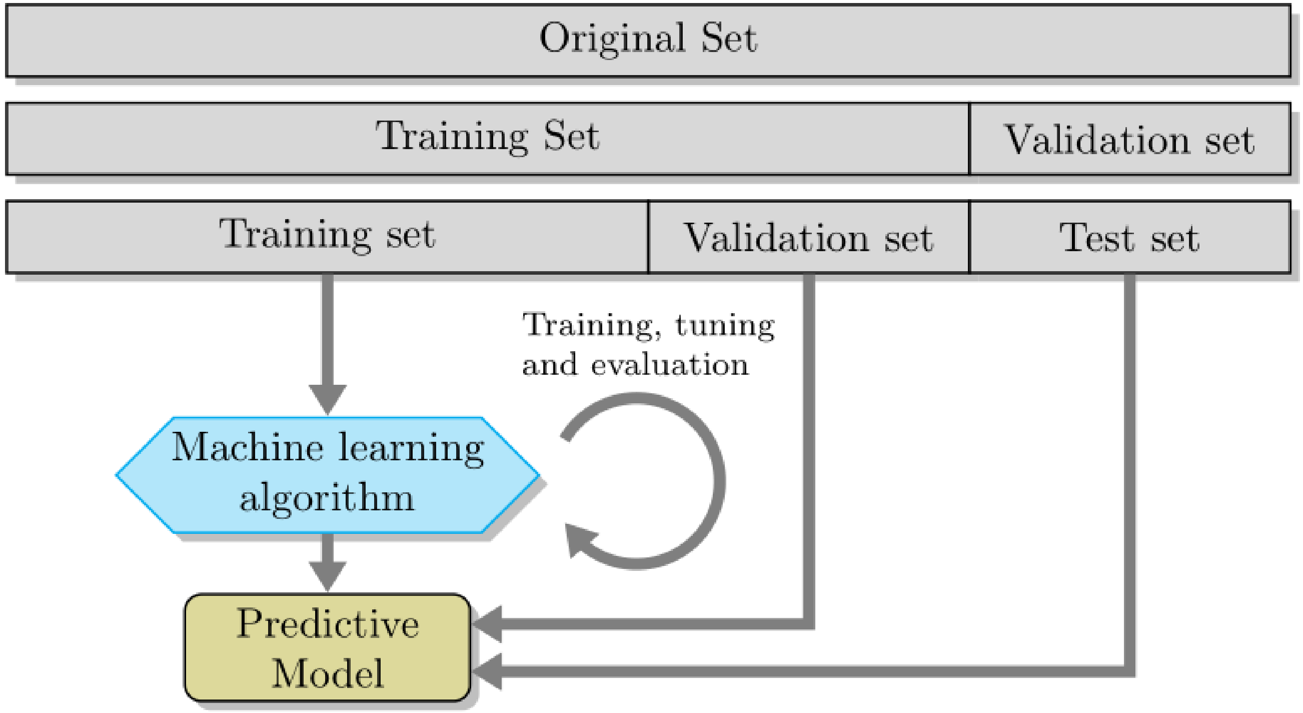


Loss function/objective function

Optimization

Data split

- Performing machine learning involves creating a model, which is trained on a training data set and then can process test data set,
- We sometimes also use a validation data set, which is a data set of examples used to tune the hyper-parameters of the model.



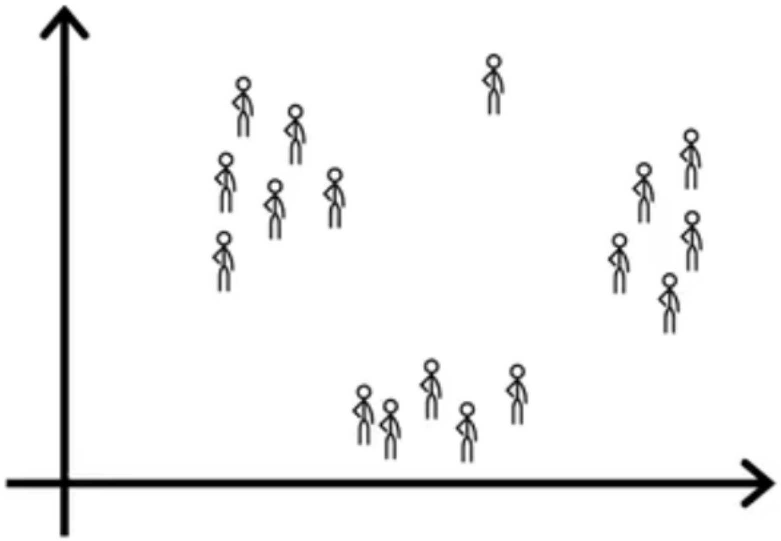
Supervised vs. Unsupervised Learning

- **Supervised learning**:
- Given a set of (x, y), learn to predict y using x.
- E.g., predict housing price based on its year, location, size, etc.

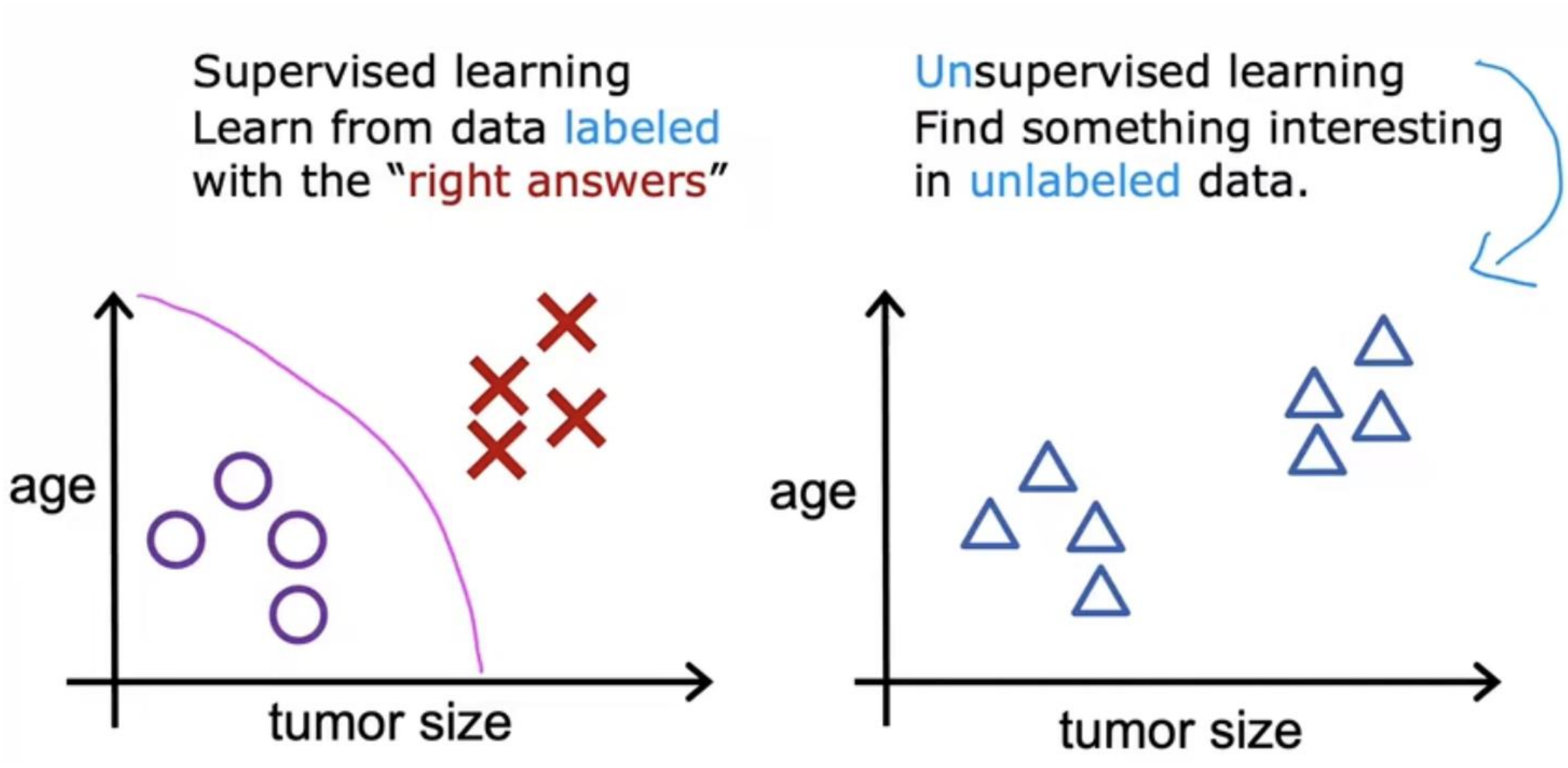
$$\underset{\text{input}}{x} \longrightarrow \underset{\text{output}}{y}$$

Supervised vs. Unsupervised Learning

- **Unsupervised learning**:
- Given a set of x, underlying structure or relationships of x.
- E.g., grouping customers, outlier detection, dimension reduction.



Supervised vs. Unsupervised Learning



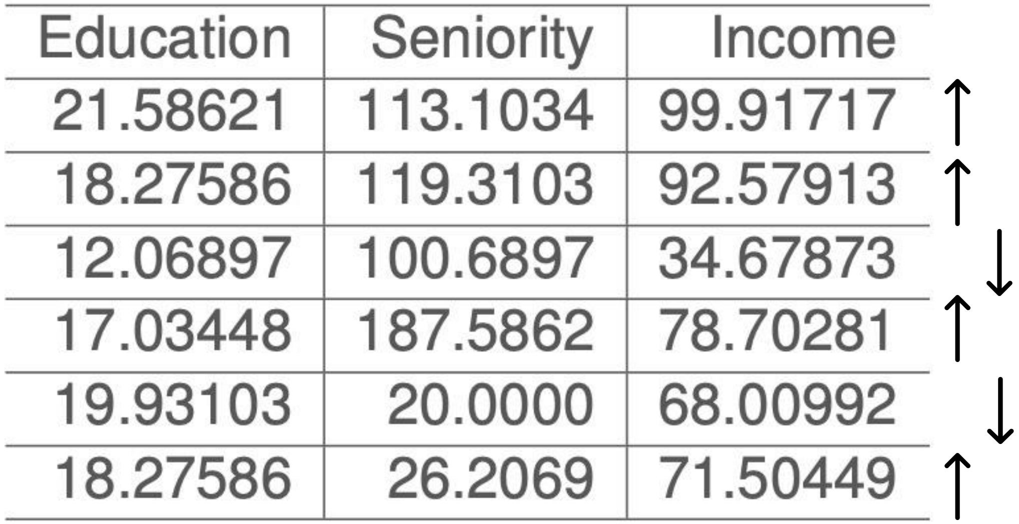
Supervised learning
Learn from data labeled
with the "right answers"

Unsupervised learning
Find something interesting
in unlabeled data.

Source: Andrew Ng

Classification vs. Regression

- The income data



X		$Y \in \mathbb{R}$	
Education	Seniority	Income	
21.58621	113.1034	99.91717	↑
18.27586	119.3103	92.57913	↑
12.06897	100.6897	34.67873	↓
17.03448	187.5862	78.70281	↑
19.93103	20.0000	68.00992	↓
18.27586	26.2069	71.50449	↑

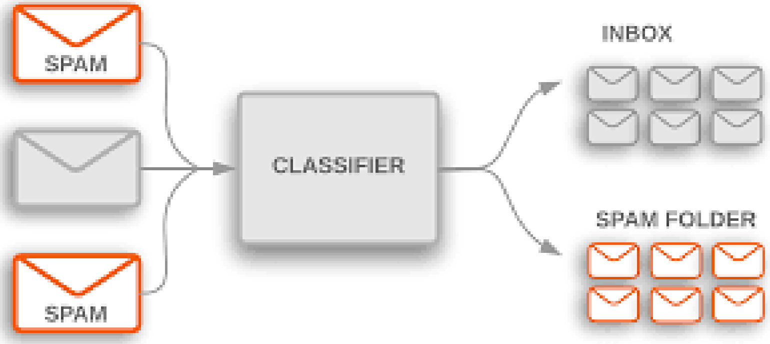
Regression: Model exact income based on other characteristics.

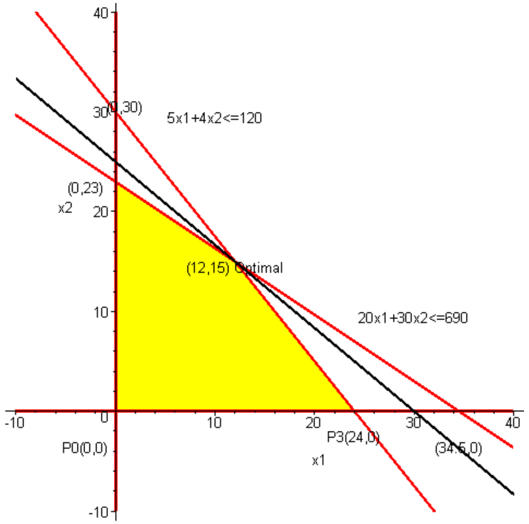
Classification: Model whether someone will earn above the 70 based on other characteristics.

Introduction to optimization

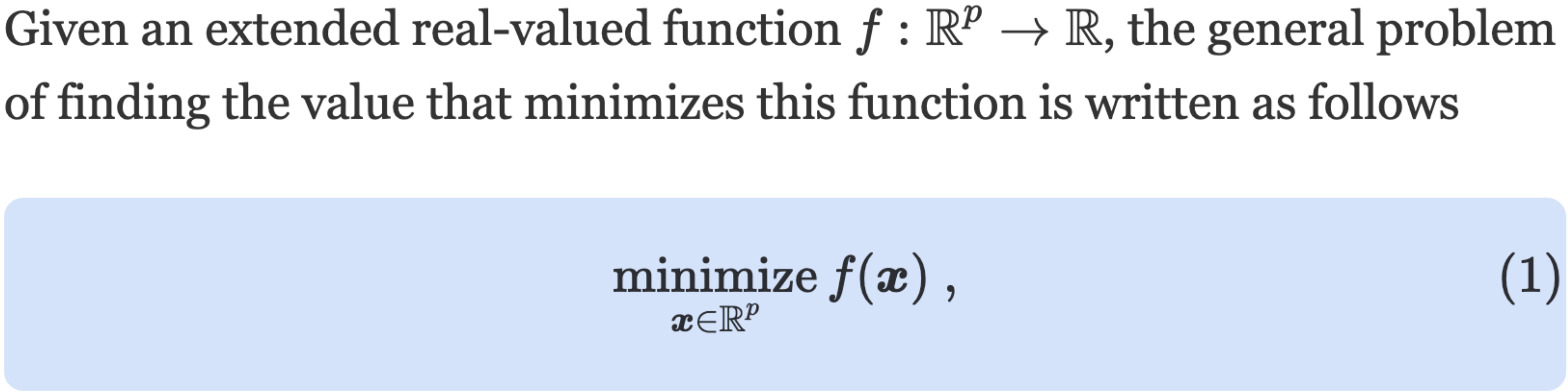
What is optimization

- **Optimization** is the branch of mathematics that aims to solve the problem of finding the elements that maximize or minimize a given function.
- Many problems in engineering and ML can be cast as optimization problem:
  - In a spam detection filter we might aim to find the system that minimizes the number of misclassified emails
  - When an engineer designs a pipe, we will seek for the design that minimizes cost while respecting some safety constraints.





Notation

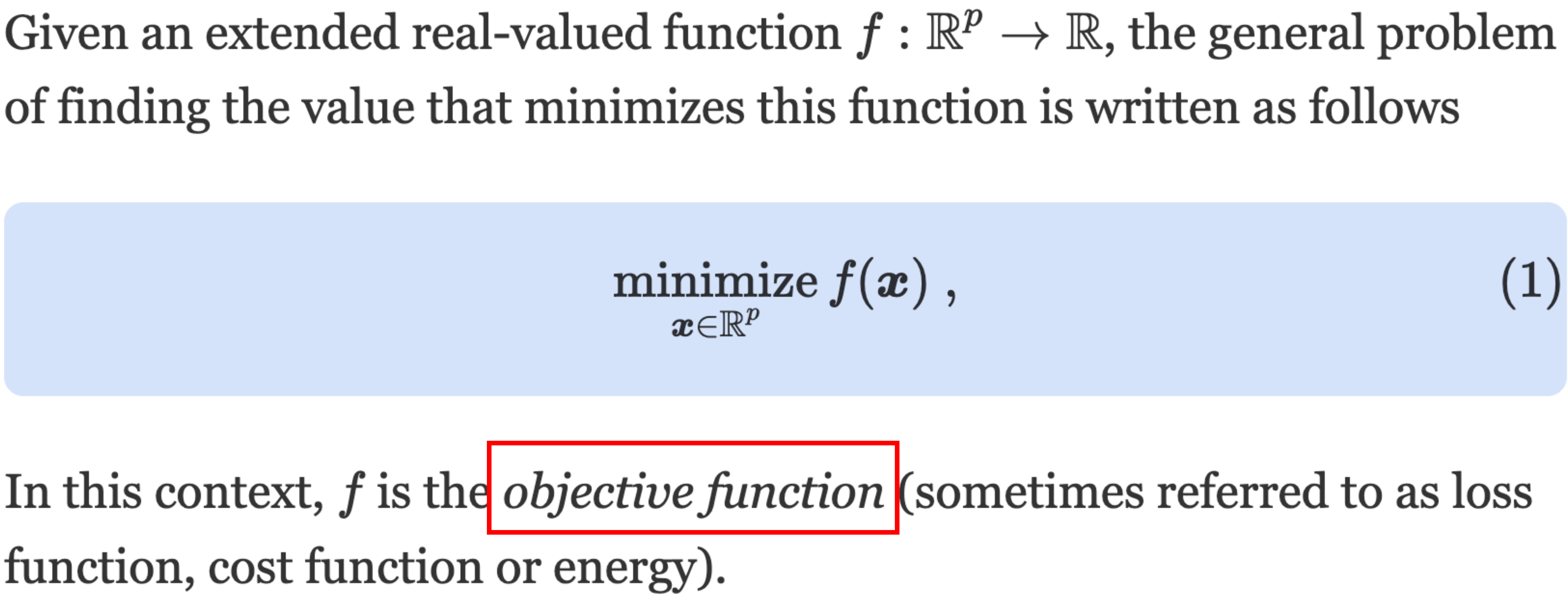


Given an extended real-valued function $f : \mathbb{R}^p \to \mathbb{R}$, the general problem of finding the value that minimizes this function is written as follows

$$\underset{\boldsymbol{x} \in \mathbb{R}^p}{\text{minimize}}\, f(\boldsymbol{x}) \;, \tag{1}$$

https://fa.bianp.net/teaching/2018/eecs227at/introduction.html

Notation



Given an extended real-valued function $f : \mathbb{R}^p \to \mathbb{R}$, the general problem of finding the value that minimizes this function is written as follows

$$\operatorname*{minimize}_{\boldsymbol{x} \in \mathbb{R}^p} f(\boldsymbol{x}) \ , \qquad (1)$$

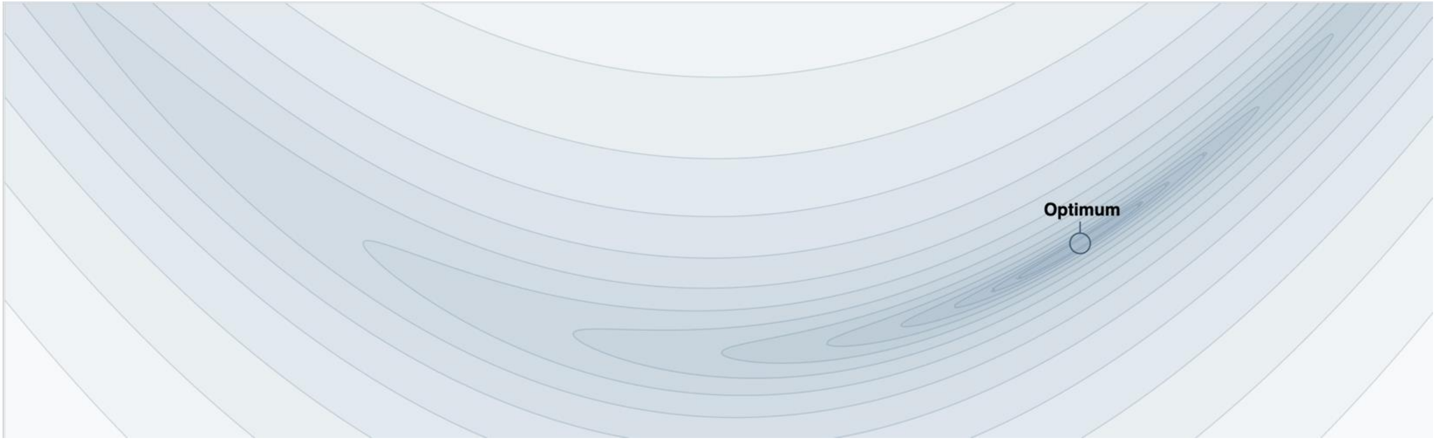
In this context, f is the *objective function* (sometimes referred to as loss function, cost function or energy).

The rules of the game

Consider the following 2-dimensional optimization problem, with

$$\underset{x \in \mathbb{R}^2}{\text{minimize}} \underbrace{(1 - x_1)^2 + 100(x_2 - x_1^2)^2}_{=f(x)} .$$

Since the *domain* of the objective function is a 2-dimensional space, we can visualize this objective function as an image in 2-D, where the color (darker=lower value) encodes the value of the objective function.

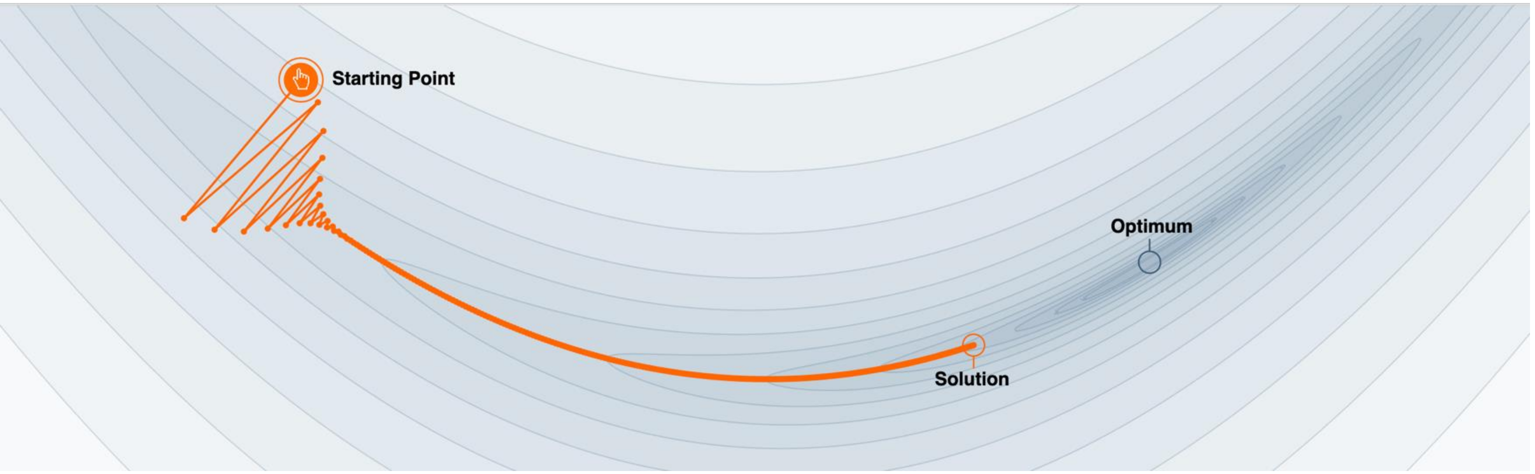


Finding the optimal

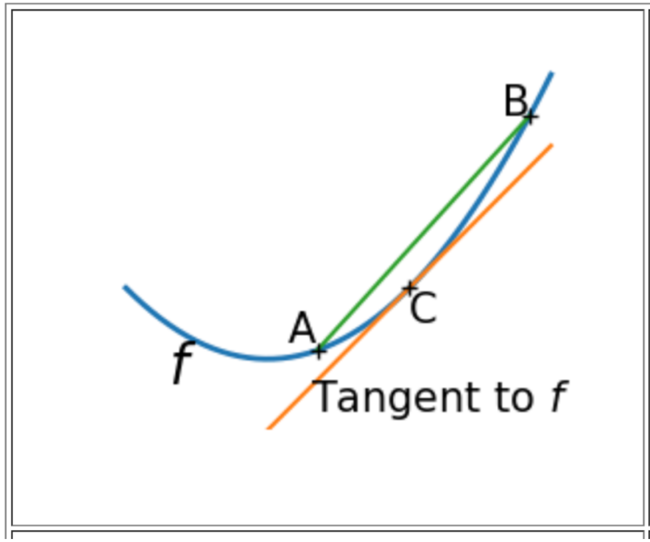
- Naïve solution:

  Grid Search

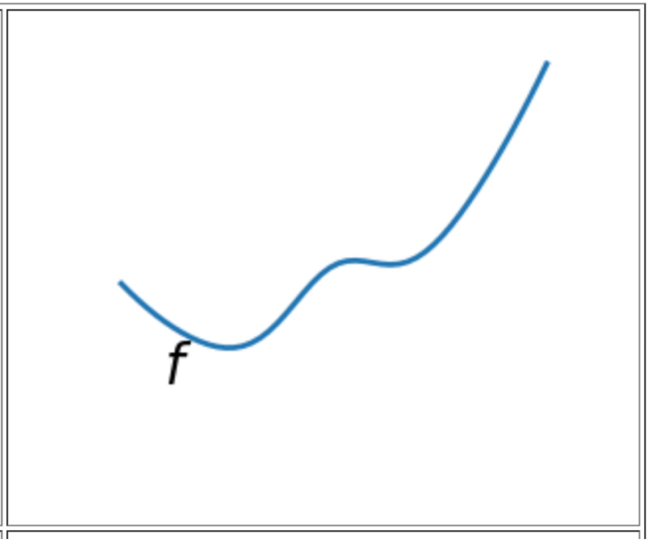
- A more efficient approach:start from an initial guess and iteratively refine the initial guess



Convex vs non-convex



Convex

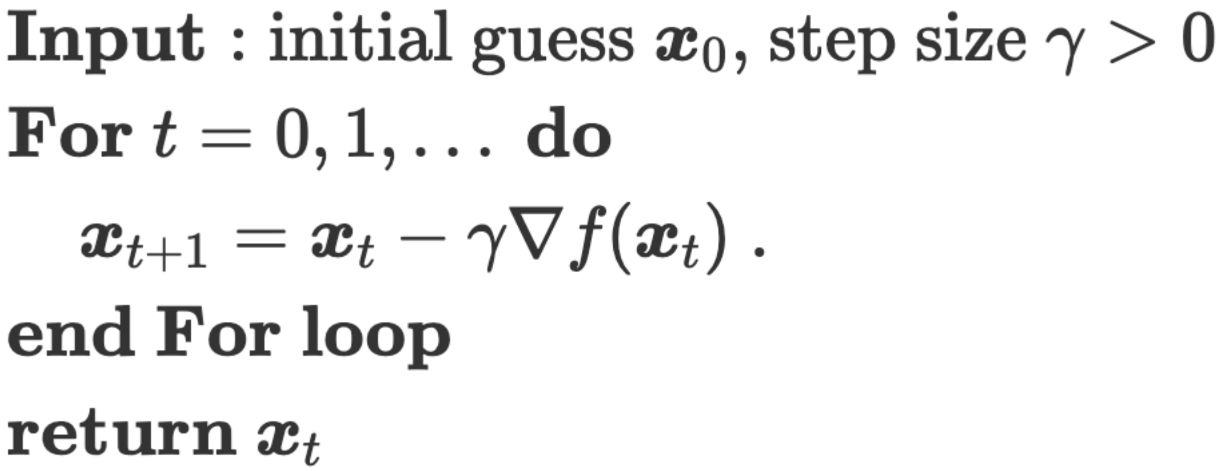


Non-convex

Optimizing convex functions is typically easier than optimizing non-convex functions.

Convex functions have the nice property that the gradient minimizes only at a global optimum (single optimum)

The Gradient Descent Algorithm

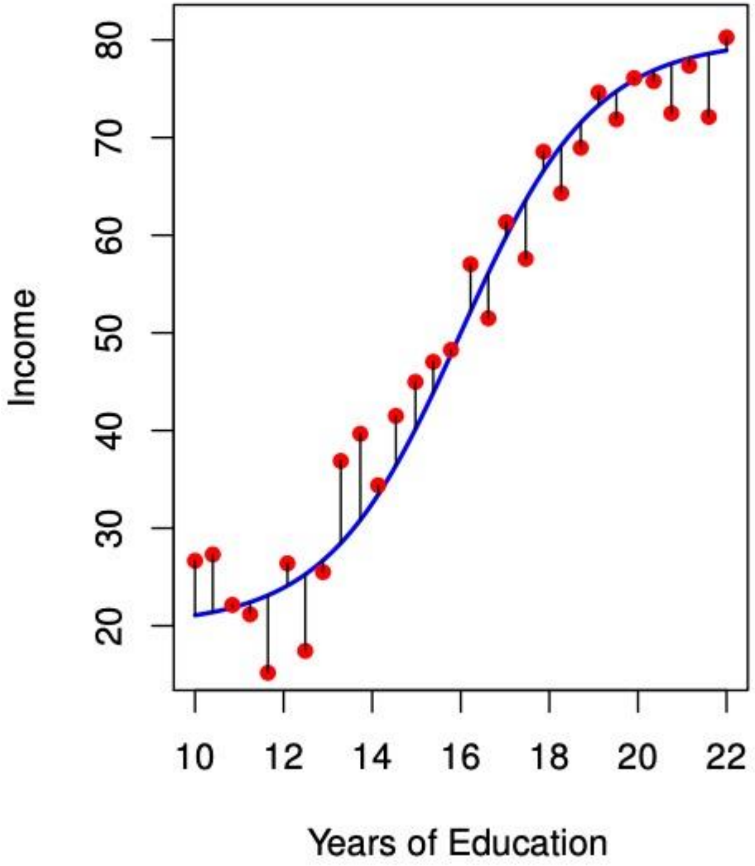


```
Input : initial guess $\boldsymbol{x}_0$, step size $\gamma > 0$
For $t = 0, 1, \ldots$ do
    $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \gamma \nabla f(\boldsymbol{x}_t)$ .
end For loop
return $\boldsymbol{x}_t$
```

https://engineering.purdue.edu/ChanGroup/ECE595/files/Lecture05_descent.pdf

Linear Regression

Regression Example



Quantitative response Y

Predictors $X = (X_1, \ldots, X_p)$

Assume the relationship can be expressed by:

$$Y = f(X) + \epsilon,$$

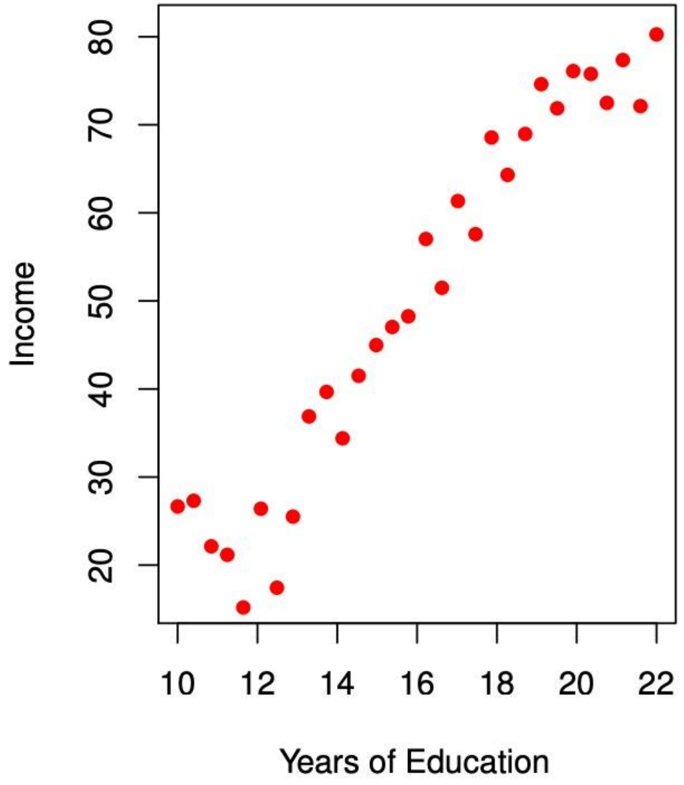
where f is a fixed, unknown function and ϵ is error term.

https://www2.isye.gatech.edu/~yxie77/isye2028/lecture11.pdf

https://www.cs.toronto.edu/~rgrosse/courses/csc421_2019/readings/L02a%20Linear%20Regression.pdf

Regression Example

Back to regression with $p = 1$:



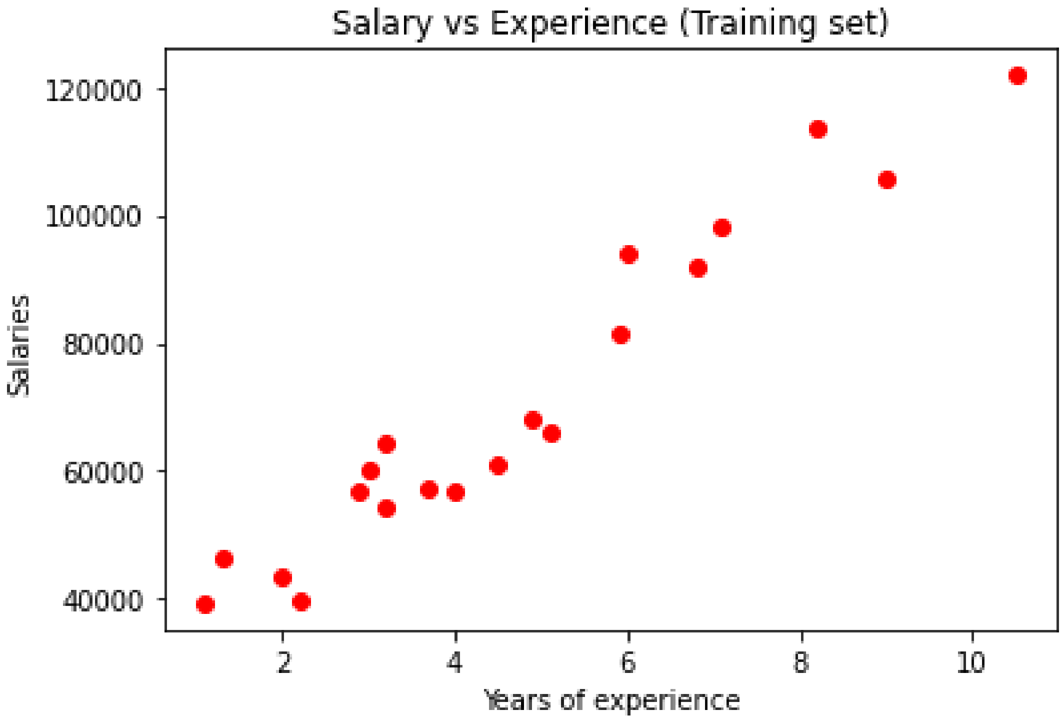
$$Y = f(X) + \epsilon$$

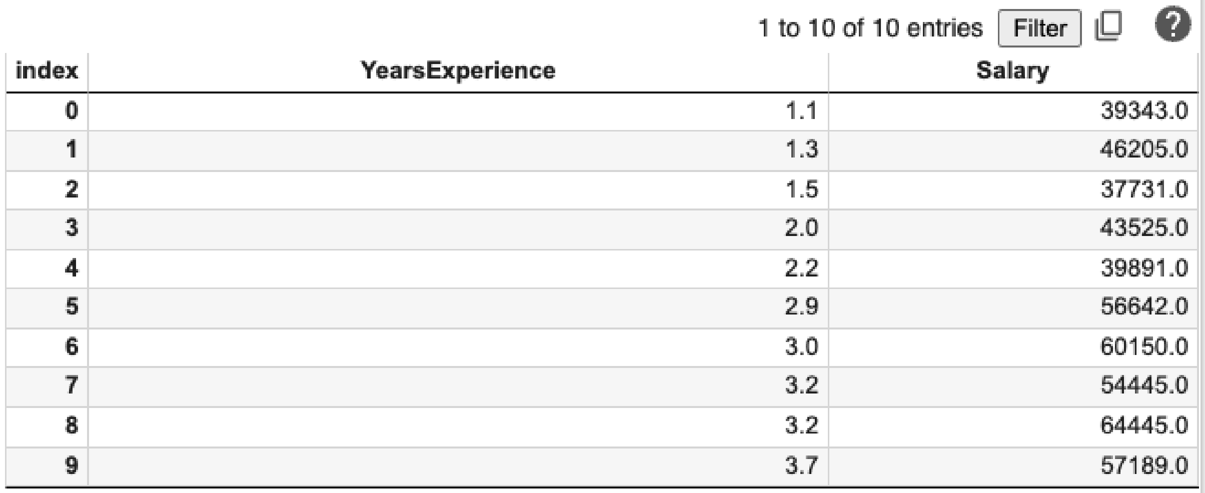
Modeling:

Use a procedure to get $\widehat{f}$. Derive estimates $\widehat{Y} = \widehat{f}(X)$.

Example of Linear Regression

- Data: Salary Prediction



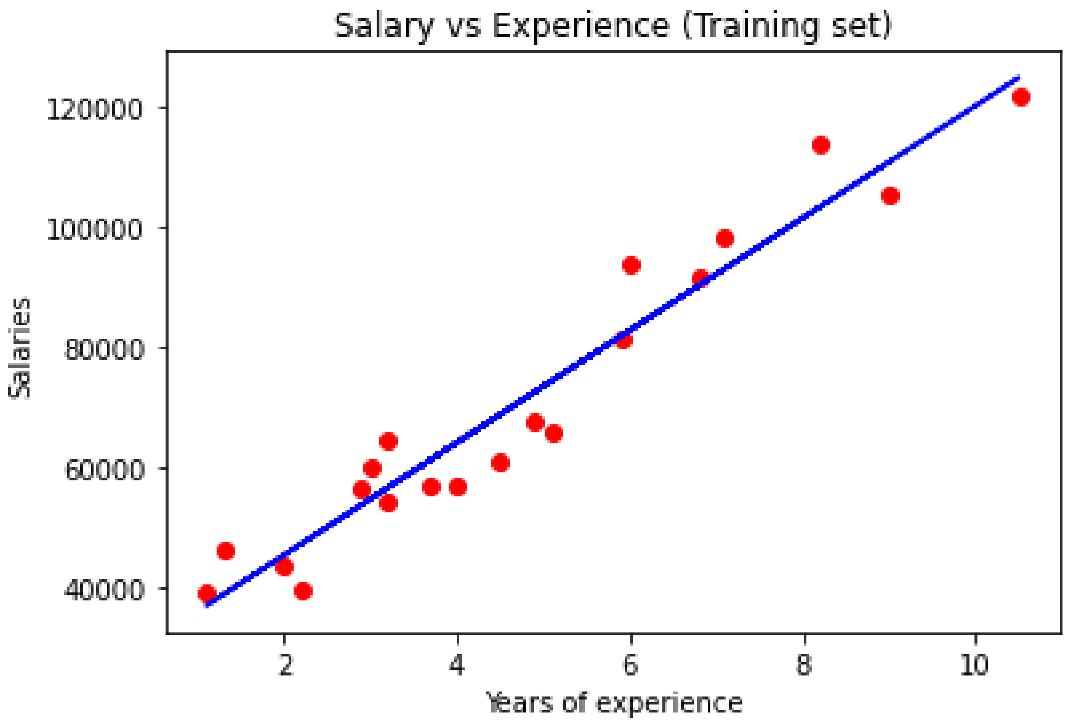


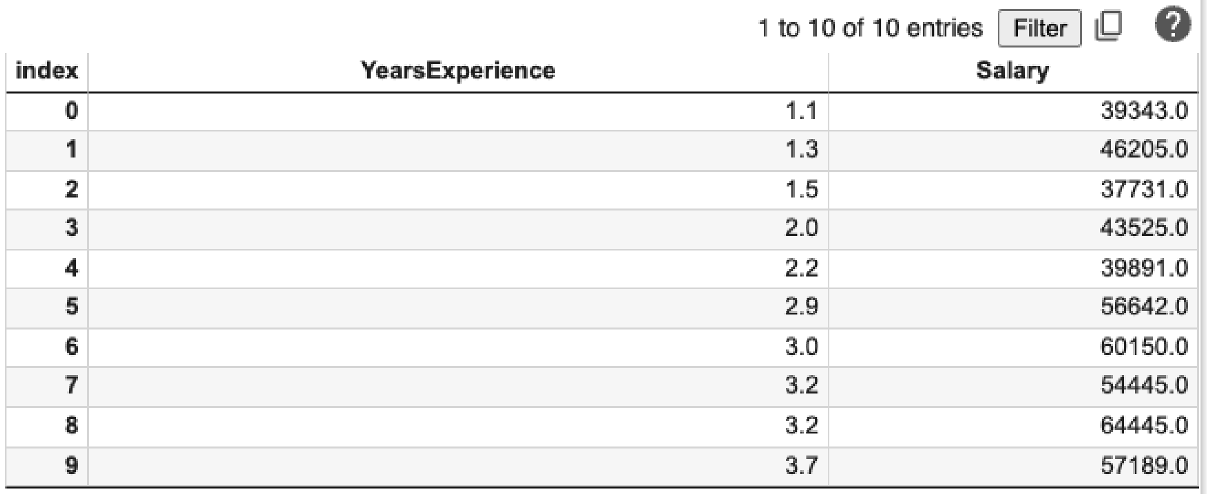
index	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0

Dataset download link: https://github.com/content-anu/dataset-simple-linear

Example of Linear Regression

- Data: Salary Prediction





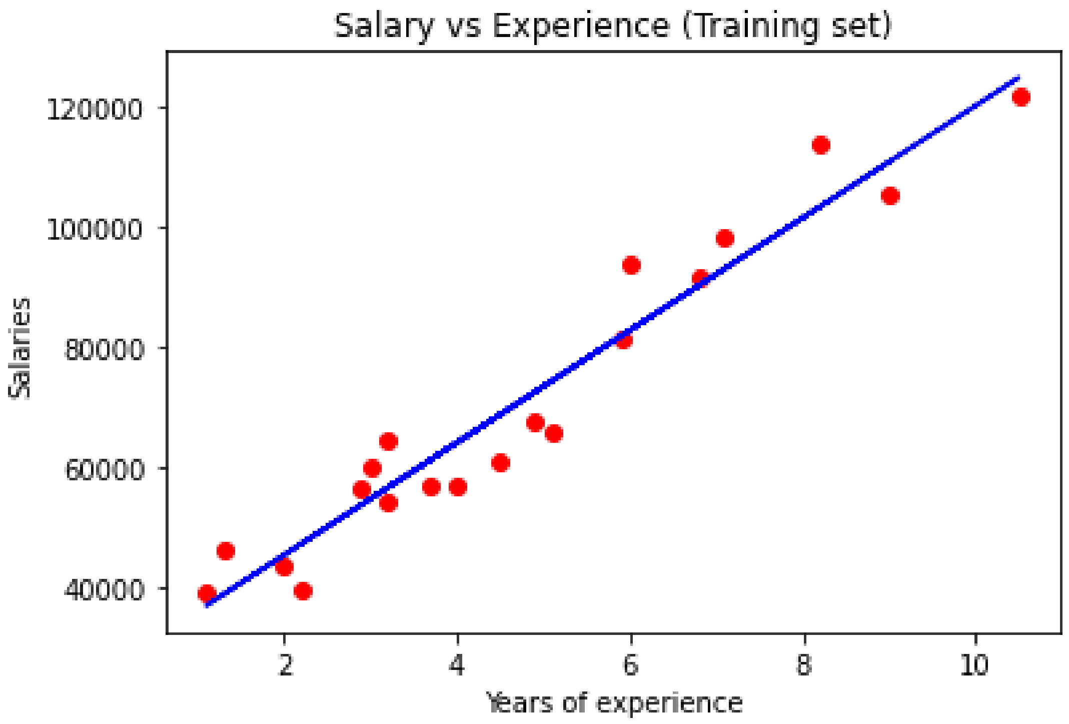
index	YearsExperience	Salary
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0

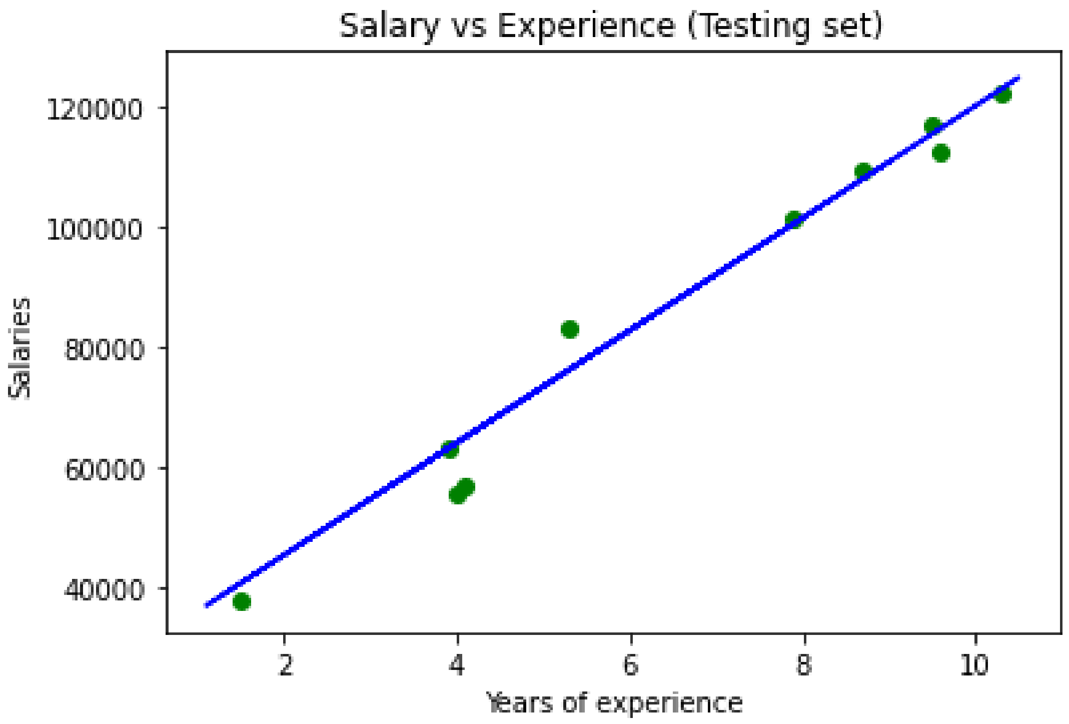
Dataset download link: https://github.com/content-anu/dataset-simple-linear

Example of Linear Regression

- Data: Salary Prediction

Training (red) Fitting function (blue) Testing (green)





Dataset download link: https://github.com/content-anu/dataset-simple-linear

Linear Regression

- **Formulation for n-dimensional feature space**

What if the income is also related to other features (factors), e.g., city, title ?

We denote *n* variables $x_1, x_2, \dots, x_n$ to represent n features

$$\hat{y} = f_{\{\theta_0,\theta_1,\dots,\theta_n\}}(x_1, x_2, \dots, x_n) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$$

- **Goal**: Find the best $\{\theta_0, \theta_1, \dots, \theta_n\}$ to predict y given x.

Linear Regression

- **Formulation for n-dimensional feature space**

Write the multivariate linear function in matrix form

$$\hat{y} = f_{\Theta}(X) = X^T\Theta$$

where

$$\Theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{pmatrix}, \text{ and } X = \begin{pmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

- We have n vairables, $\Theta \in \mathbb{R}^{n+1}, X \in \mathbb{R}^{n+1}$, and $y \in \mathbb{R}$.
- Θ is the parameters that should be **learned** from training data.

Linear Regression

- How about representing a linear regression problem with m samples?

Suppose we have

$$X^{(1)} = \begin{pmatrix} 1 \\ x_1^{(1)} \\ \vdots \\ x_n^{(1)} \end{pmatrix}, X^{(2)} = \begin{pmatrix} 1 \\ x_1^{(2)} \\ \vdots \\ x_n^{(2)} \end{pmatrix}, \dots, X^{(\mathrm{m})} = \begin{pmatrix} 1 \\ x_1^{(m)} \\ \vdots \\ x_n^{(m)} \end{pmatrix},$$

with respective labels $y^{(1)}, y^{(2)}, \dots, y^{(m)}$.

$$\hat{y} = f_\Theta(X) = X^T\Theta$$

We denote $\mathbf{X} \in \mathbb{R}^{m \times (n+1)}$ as the data matrix, of which each row represents a sample, each column represents a feature.

$$\mathbf{X} = \begin{pmatrix} {X^{(1)}}^T \\ \vdots \\ {X^{(m)}}^T \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

Linear Regression

- Matrix Representation of Linear Function

The linear function with n features and m samples is written as

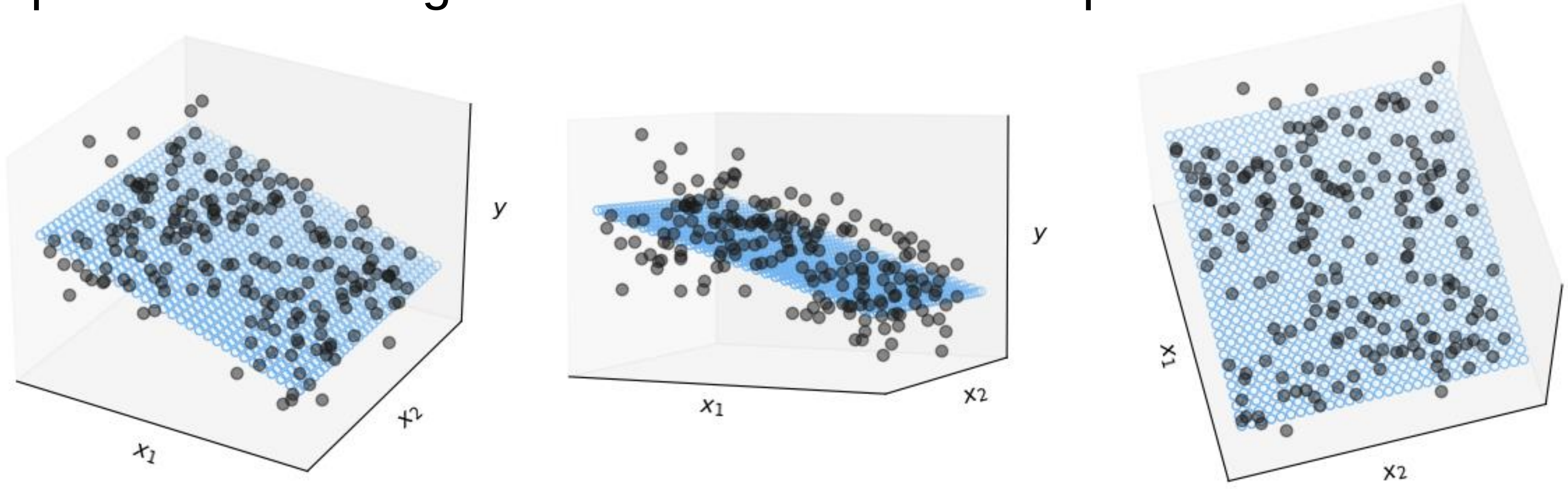
$$\hat{Y} = f_{\Theta}(\mathbf{X}) = \mathbf{X}\Theta,$$

where $\hat{Y} = \left(\hat{y}^{(1)}, \dots, \hat{y}^{(m)}\right)^T, \Theta = (\theta_0, \theta_1, \dots, \theta_n)^T$, and

$$\mathbf{X} = \begin{pmatrix} {X^{(1)}}^T \\ \vdots \\ {X^{(m)}}^T \end{pmatrix} = \begin{pmatrix} 1 & x_1^{(1)} & \cdots & x_n^{(1)} \\ 1 & \vdots & \ddots & \vdots \\ 1 & x_1^{(m)} & \cdots & x_n^{(m)} \end{pmatrix}$$

Geometry of linear regression

- Data points $\{\left(x_1^{(1)}, \dots, x_n^{(1)}, y^{(1)}\right), \dots, \left(x_1^{(1)}, \dots, x_n^{(1)}, y^{(1)}\right)\}$ form a (n+1)-dimensional space.
- The ”fitting lines” for (n+1)-dimensional feature space are n-dimensional hyperplanes.
- Examples for linear regression on 2-dim feature space:



Optimization Problem Setting

- **Problem:** How to find the best parameter Θ^*

- Assume that all data points are from the same distribution, once $\varTheta^*$ ”perfectly” fits the training data, it should be the optimal to fit all the data from the same distribution.

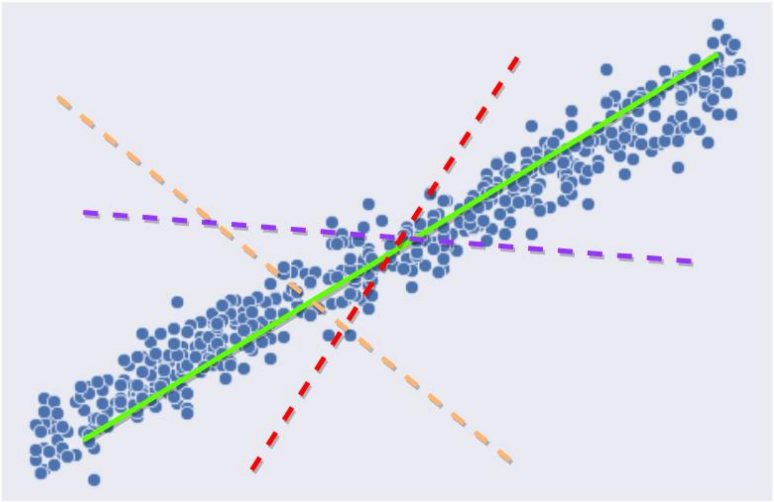
- Thus, we first focus on training data:

$$\{\left(x_1^{(1)}, \dots, x_n^{(1)}, y^{(1)}\right), \dots, \left(x_1^{(m)}, \dots, x_n^{(m)}, y^{(m)}\right)\}$$

Optimization Problem Setting

- **Problem:** What does a good fit mean?

- Intuitively, we need to measure the distance between the predictions $\hat{Y}$ and the true label of training data Y.



Objective Function

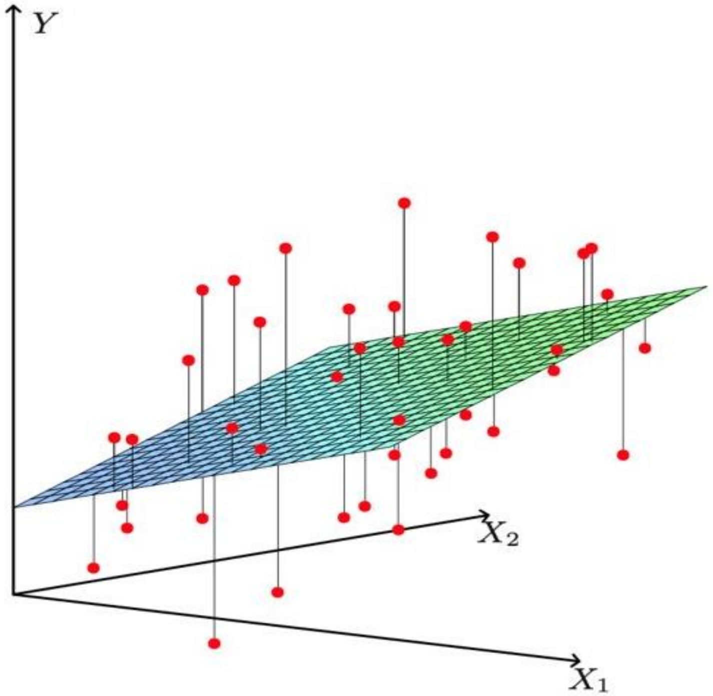
- We denote $J_\Theta(Y,\hat{Y})$as the objective function (a.k.a. cost or loss function) to measure the distance between Y and $\hat{Y} = f_\Theta(\mathbf{X})$.
- Goal: find the optimal Θ^* that minimizes $J_\Theta(Y,\hat{Y})$
- Example: Residual Sum of Squares (RSS):

$$J_\Theta\left(Y,\hat{Y}\right) = \sum_{i=1}^{m}\left(f_\Theta\left(X^{(i)}\right) - y^{(i)}\right)^2 = \left\|\hat{Y} - Y\right\|_2^2$$

Minimize RSS using ordinary least squared (OLS) method.

Geometry of RSS

- For each sample, “residual” means the difference between the estimated value (the plane) and the corresponding training label (red points).



Minimizing RSS

- Minimizing the convex objective function == find the Θ^* that minimize the loss function $J(\Theta)$.
- Θ^* is the minimal iff

$$J'(\Theta^*) = 0, J''(\Theta^*) > 0$$

Minimizing RSS

- Analytic solution:

$$J(\Theta) = \|f_{\Theta}(\mathbf{X}) - Y\|_2^2 = (\mathbf{X}\Theta - Y)^T(\mathbf{X}\Theta - Y)$$
$$= \Theta^T\mathbf{X}^T\mathbf{X}\Theta - Y^T\mathbf{X}\Theta - \Theta^T\mathbf{X}^T Y - Y^T Y$$

Minimizing RSS

- Analytic solution:

$$J(\Theta) = \|f_\Theta(\mathbf{X}) - Y\|_2^2 = (\mathbf{X}\Theta - Y)^T(\mathbf{X}\Theta - Y)$$
$$= \Theta^T\mathbf{X}^T\mathbf{X}\Theta - Y^T\mathbf{X}\Theta - \Theta^T\mathbf{X}^TY - Y^TY$$

- First derivative:

$$\frac{\partial J(\Theta)}{\partial\Theta} = 2\mathbf{X}^T\mathbf{X}\Theta\text{-}\ \mathbf{X}^TY\text{-}\ \mathbf{X}^TY$$
$$= 2\mathbf{X}^T(\mathbf{X}\Theta - Y) = 0$$

Minimizing RSS

- Analytic solution:

$$J(\Theta) = \|f_\Theta(\mathbf{X}) - Y\|_2^2 = (\mathbf{X}\Theta - Y)^T(\mathbf{X}\Theta - Y)$$

$$= \Theta^T\mathbf{X}^T\mathbf{X}\Theta - Y^T\mathbf{X}\Theta - \Theta^T\mathbf{X}^TY - Y^TY$$

- Second derivative:

$$\frac{\partial^2 J(\Theta)}{\partial \Theta^2} = 2\mathbf{X}^T\mathbf{X} > 0 \textit{ is for true}$$

Minimizing RSS

- First derivative:

$$\frac{\partial J(\Theta)}{\partial \Theta} = 2\mathbf{X}^T\mathbf{X}\Theta\text{-}\ \mathbf{X}^T Y\text{-}\ \mathbf{X}^T Y$$

$$= 2\mathbf{X}^T(\mathbf{X}\Theta - Y) = 0$$

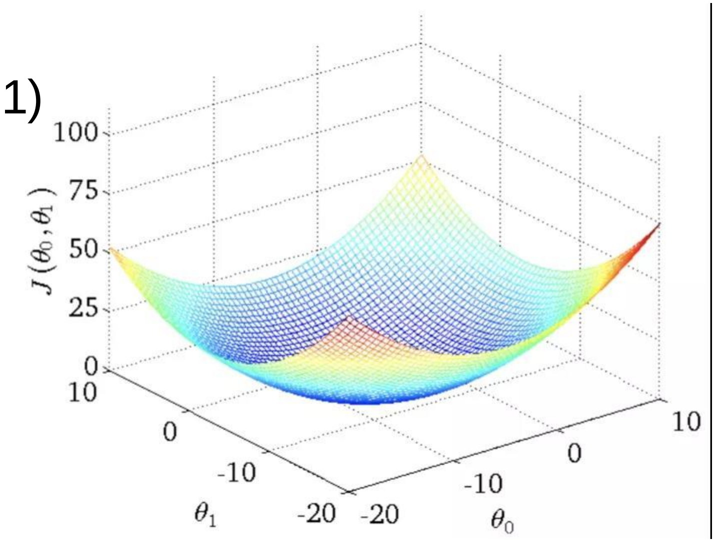
Optimal: $\Theta^* = (\mathbf{X}^T\mathbf{X})^{-1}\ \mathbf{X}^T Y$

*See https://en.wikipedia.org/wiki/Matrix_calculus#Scalar-by-matrix_identities for details on computing scalar-matrix derivatives.

Understanding the analytic solution

- Find the optimal solution of the convex objective function: $J(\Theta)$
- $J(\Theta^{*})$ is the global minimum iff $J'(\Theta^{*}) = 0$ and $J''(\Theta^{*})$>0

(X^T X)^(-1)



Questions on the analytic solution

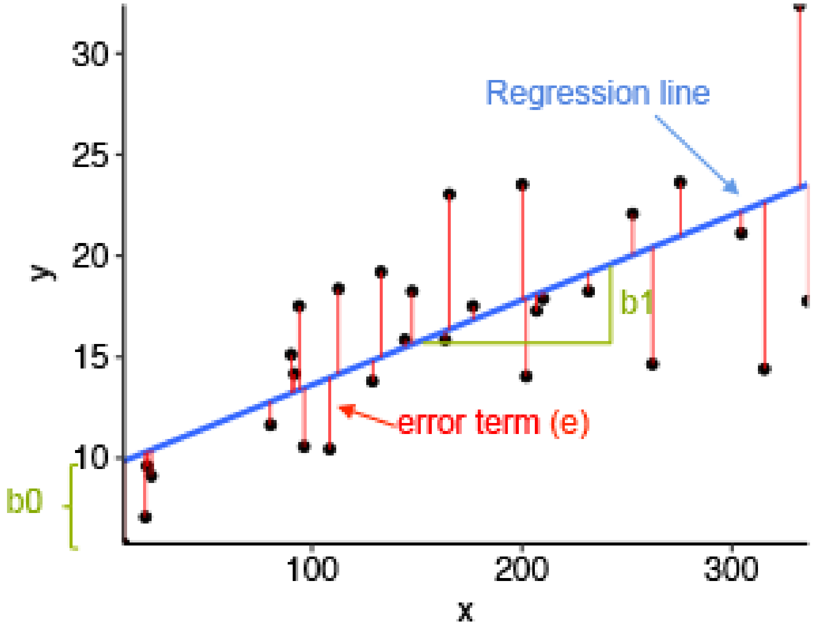
Given that $\Theta^* = (\mathbf{X}^T\mathbf{X})^{-1}\ \mathbf{X}^T Y$

- What if $(\mathbf{X}^T\mathbf{X})^{\mathbf{-1}}$ does not has the exact form, i.e., $(\mathbf{X}^T\mathbf{X})$ is not invertable?
- What are the conditions to ensure $(\mathbf{X}^T\mathbf{X})$'s invertibility?

$\mathbf{X}^T\mathbf{X}$ is invertible $\Leftrightarrow$ $\mathbf{X}$ has linearly independent columns

Metrics to evaluate regression models

- Regression aims to predict numeric values
- Metrics for regression involve calculating an error score to summarize the predictive skill of a model.

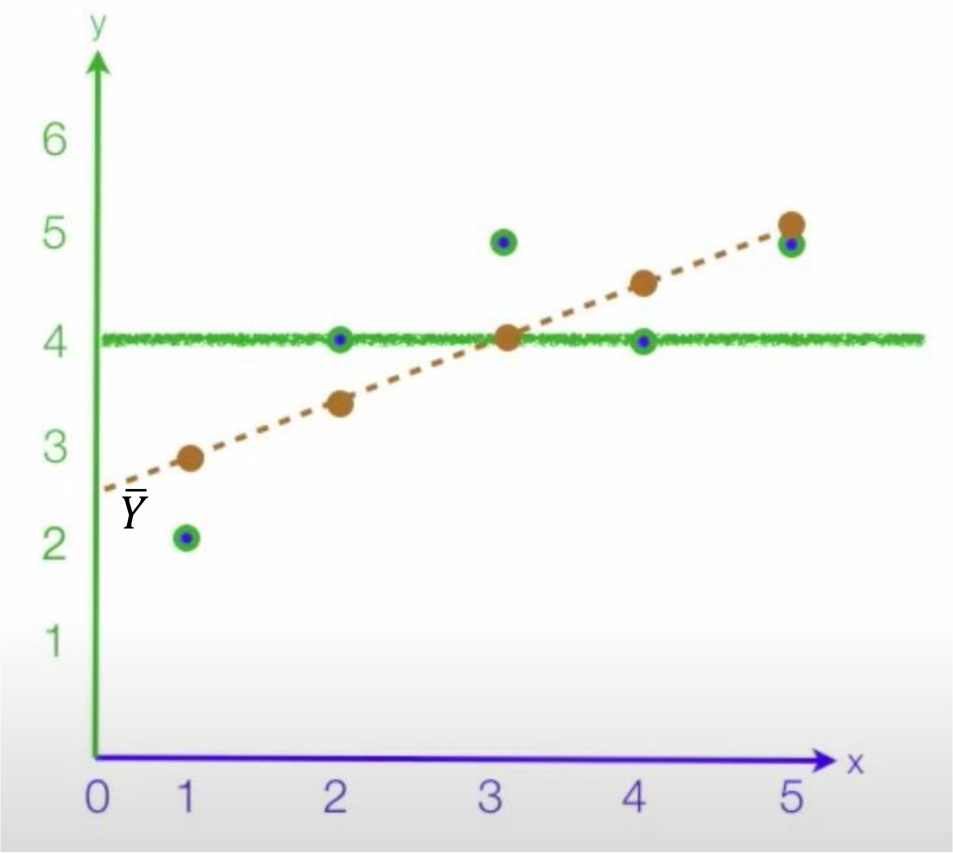


Metrics to evaluate regression models

Denote total samples as N, true label as Y, prediction as $\hat{Y}$.

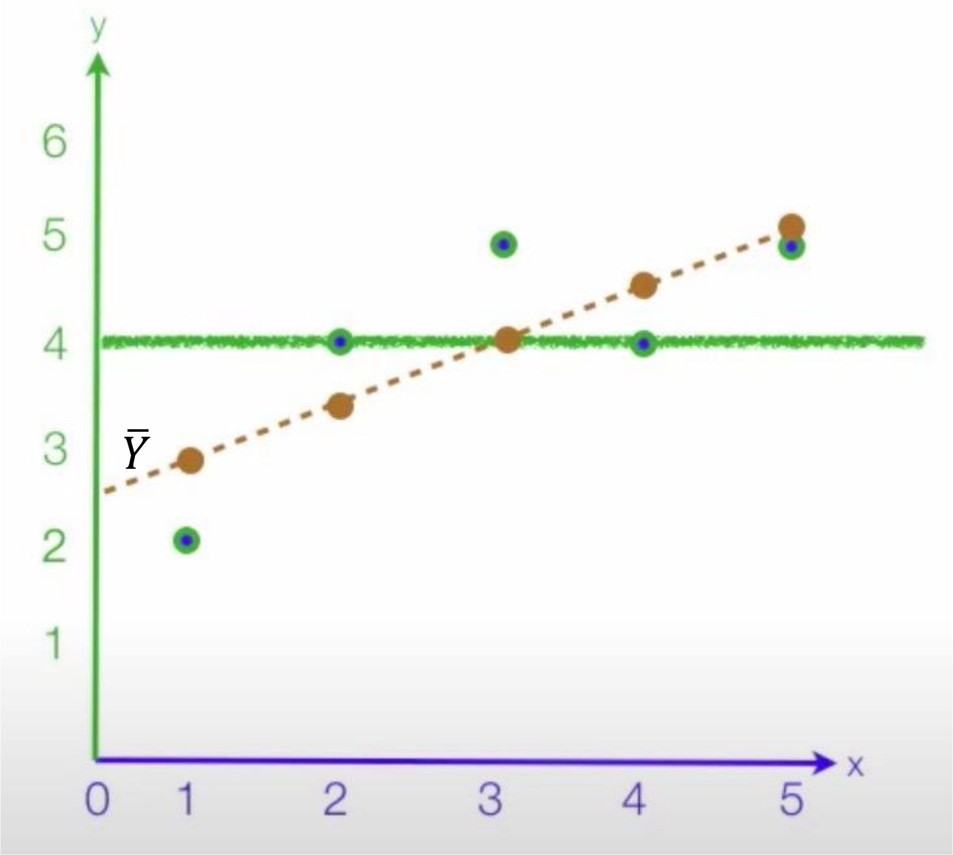
- Mean Absolute Error (MAE): $\mathrm{MAE} = \frac{1}{N}\sum_{n}^{N}\left|Y_n - \hat{Y}_n\right|$
- Mean Square Error (MSE): $\mathrm{M}S\mathrm{E} = \frac{1}{N}\sum_{n}^{N}\left(Y_n - \hat{Y}_n\right)^2$
- Root Mean Square Error (RMSE) : $\mathrm{RM}S\mathrm{E} = \sqrt{\mathrm{MSE}}$
- R-square: $R^2 = 1 - \frac{\sum(Y - \hat{Y}\ )^2}{\sum(Y - \bar{Y})^2}$

An example of R-square



		Total Variance		Residual Square		
X	Y	$Y-\bar{Y}$	$(Y-\bar{Y})^2$	$\hat{Y}$	$Y-\hat{Y}$	$(Y-\hat{Y})^2$
1	2			2.8		
2	4			3.4		
3	5			4		
4	4			4.6		
5	5			5.2		

An example of R-square

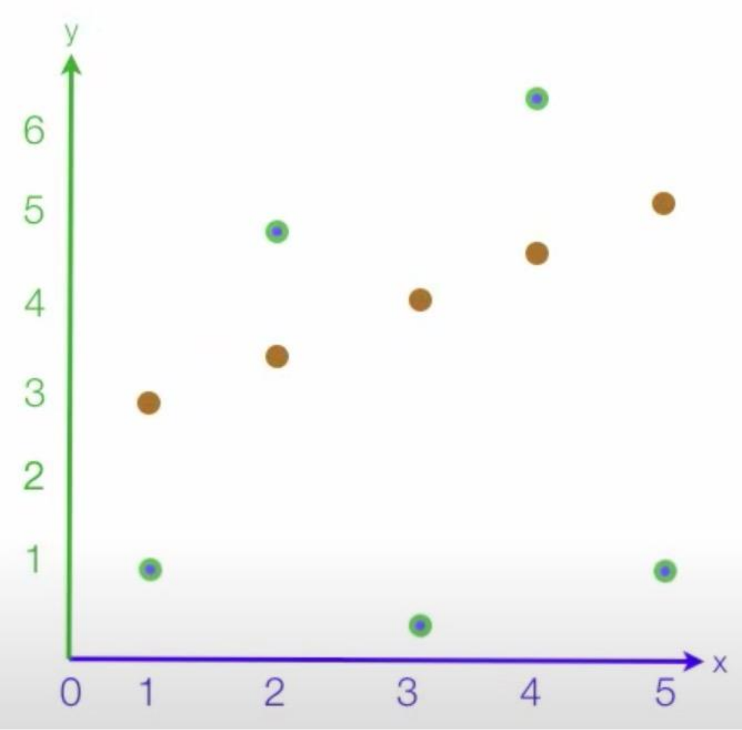


			Total Variance			Residual Square
X	Y	$Y-\bar{Y}$	$(Y-\bar{Y})^2$	$\hat{Y}$	$Y-\hat{Y}$	$\left(Y-\hat{Y}\right)^2$
1	2	-2	4	2.8	-0.8	0.64
2	4	0	0	3.4	.6	0.36
3	5	1	1	4	1	1
4	4	0	0	4.6	-0.6	0.36
5	5	1	1	5.2	-0.2	0.04

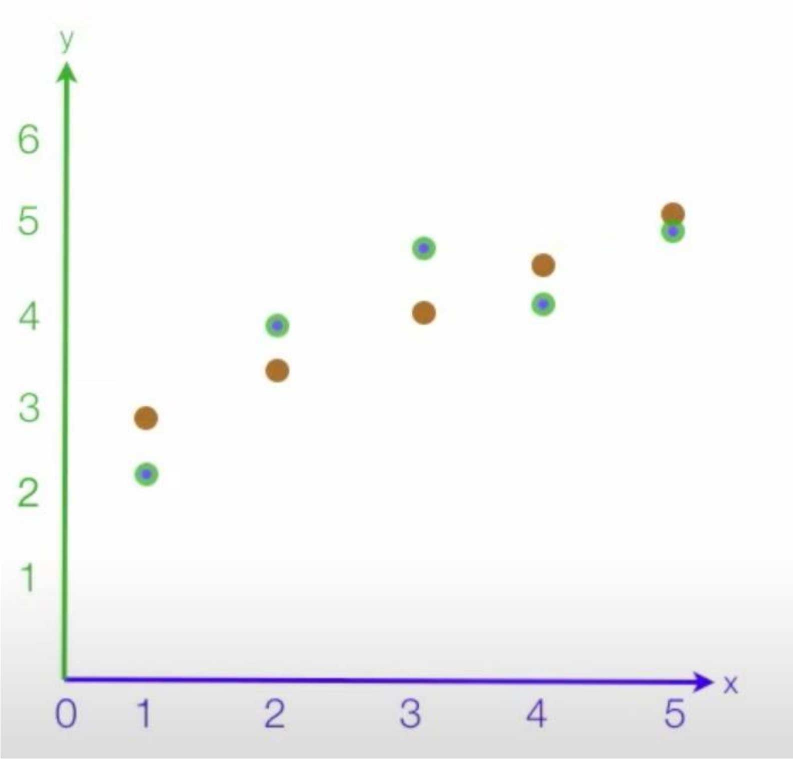
$$R^2 = 1 - \frac{\sum\left(Y - \hat{Y}\right)^2}{\sum(Y-\bar{Y})^2} = 1 - \frac{2.4}{6} = 0.6$$

Comparing R-square

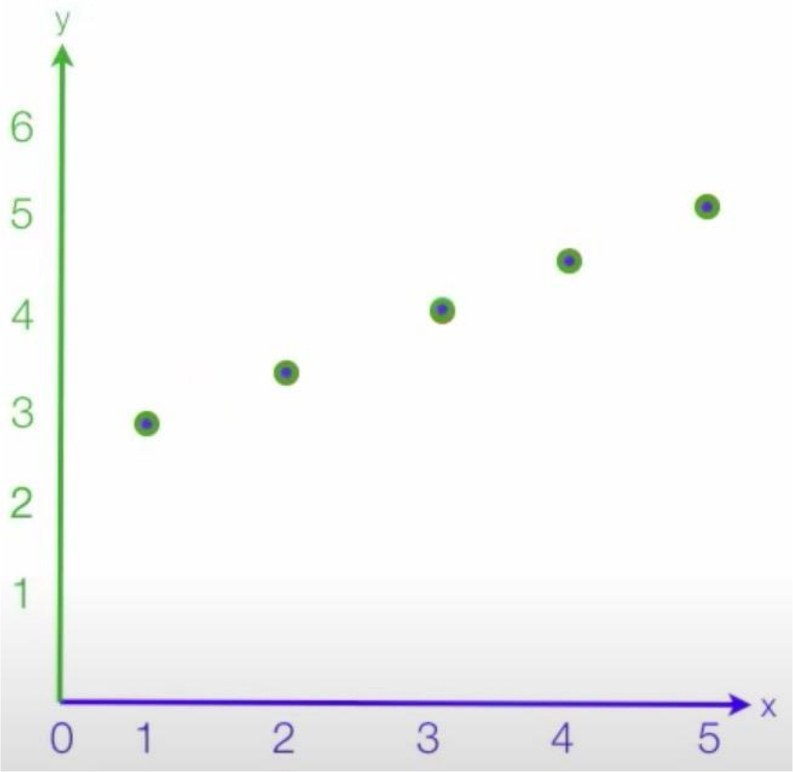
$R^2 = 0.02$



$R^2 = 0.6$

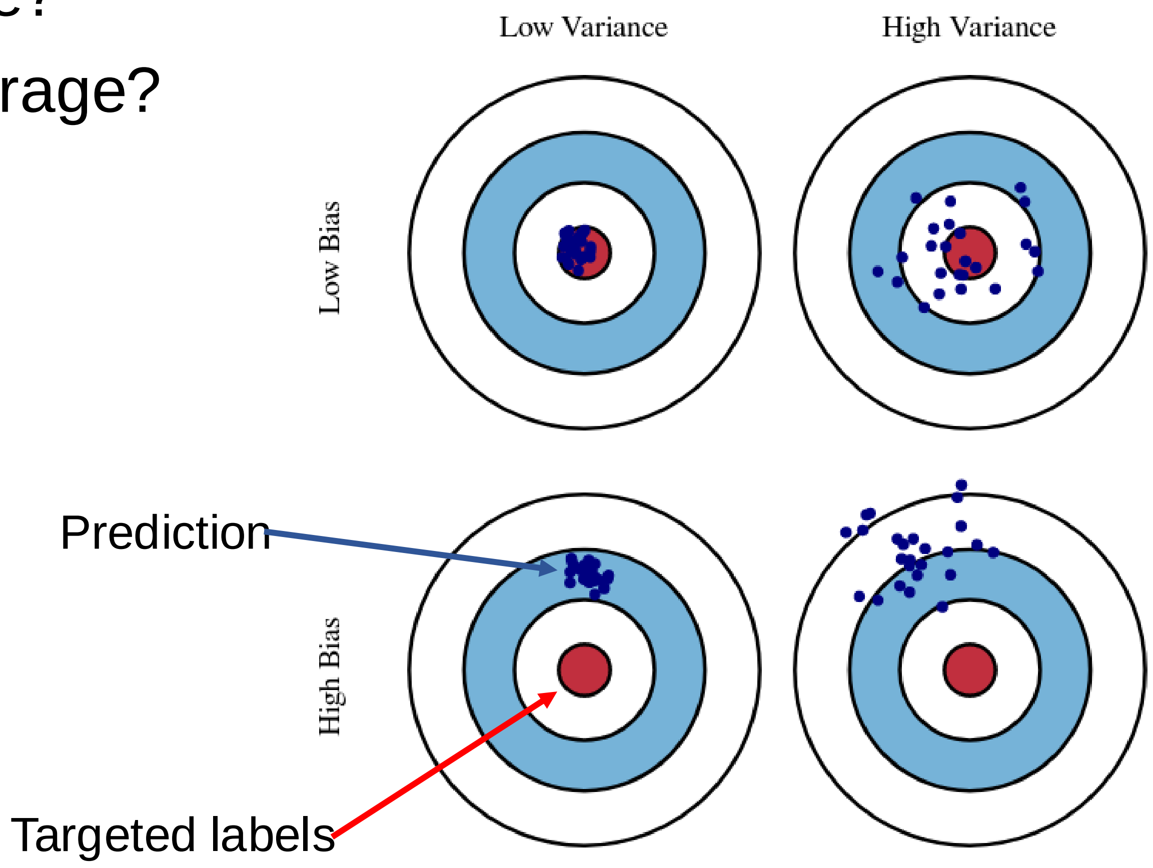


$R^2 = 0.9$



Bias and variance

- Bias: How much are we off—on average?
- Variance: How variable are we—on average?



Expected prediction error (Risk)

- The relation of input and output is modeled by the function f.
- Due to the noise from observation, $y = f(X) + \epsilon$, where $\mathbb{E}(\epsilon) = 0$ and $\mathrm{Var}(\epsilon) = \sigma^2$.
- For any fixed input X and its label y, the expected prediction error (EPE) on X is

$$\mathrm{EPE}(X) = \mathbb{E}\left[\left(y - \hat{f}(X)\right)^2\right] = \mathrm{Bias}\left(\hat{f}(X)\right)^2 + \mathrm{Var}\left(\hat{f}(X)\right) + \sigma^2$$

where

$$\mathrm{Bias}\left(\hat{f}(X)\right) = f(X) - \mathbb{E}[\hat{f}(X)]$$

$$\mathrm{Var}\left(f(X)\right) = \mathbb{E}[\left(\hat{f}(X) - \mathbb{E}\left[\hat{f}(X)\right]\right)^2]$$

See https://en.wikipedia.org/wiki/Bias-variance_tradeoff#Derivation for the detailed derivation.

Derivations

EPE = Bias^2 + Variance

$$\mathbb{E}\left[\left(y - \hat{f}(X)\right)^2\right] = \mathrm{Bias}\left(\hat{f}(X)\right)^2 + \mathrm{Var}\left(\hat{f}(X)\right)$$

$$\mathbb{E}\left[\left(y - \hat{f}(X)\right)^2\right] = \left(y - \mathbb{E}\hat{f}(X)\right)^2 + \mathbb{E}\left(\hat{f} - \mathbb{E}\hat{f}\right)^2$$

Some derivations

- Proof:

$$\mathbb{E}\left[\left(f - \hat{f}\right)^2\right] = \mathbb{E}\left(f - \mathbb{E}\hat{f} + \mathbb{E}\hat{f} - \hat{f}\right)^2$$

$$= \mathbb{E}\left(f - \mathbb{E}\hat{f}\right)^2 - 2\mathbb{E}\left\{\left(f - \mathbb{E}\hat{f}\right)\left(\hat{f} - \mathbb{E}\hat{f}\right)\right\} + \mathbb{E}\left(\hat{f} - \mathbb{E}\hat{f}\right)^2$$

$$= \mathbb{E}\left(f - \mathbb{E}\hat{f}\right)^2 - 2\left(f - \mathbb{E}\hat{f}\right)\mathbb{E}\left(\hat{f} - \mathbb{E}\hat{f}\right) + \mathbb{E}\left(\hat{f} - \mathbb{E}\hat{f}\right)^2$$

$$= \mathbb{E}\left(f - \mathbb{E}\hat{f}\right)^2 + \mathbb{E}\left(\hat{f} - \mathbb{E}\hat{f}\right)^2$$

= Bias^2 + Variance

Bias-variance trade-off

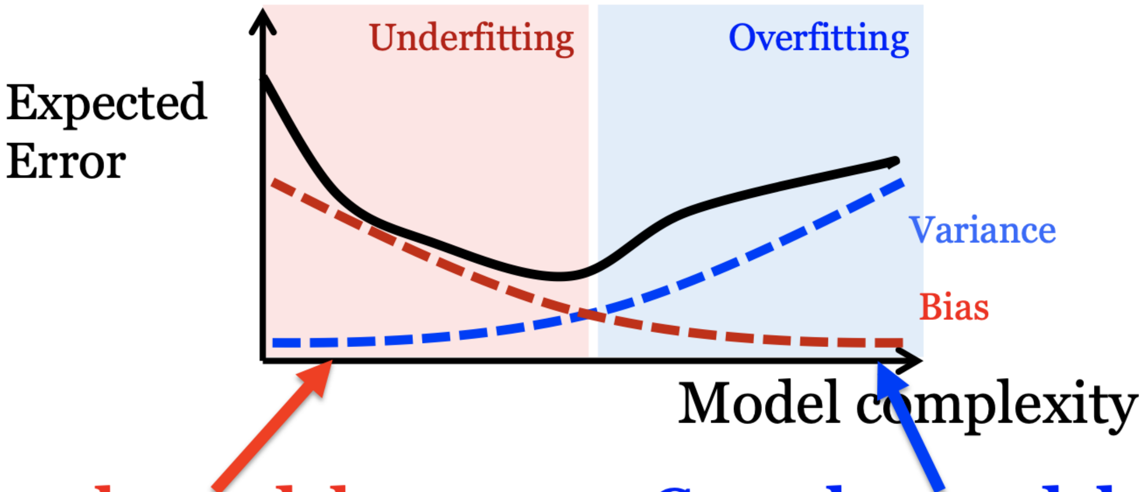
- However, in general, low variance will cause high bias, while low bias will result in high variance.

- Think about we repeat the training process on randomly sampled data for many times.
    1. For each training, if the model perfectly fits the training data, the prediction bias is very low but the variance will be very high since the model will vary significantly among different training data.
    2. If the model is constant among different training data, the prediction variance is zero but definitely the prediction bias is very high.

- Thus, we need to make a trade-off between minimizing bias and minimizing variance.

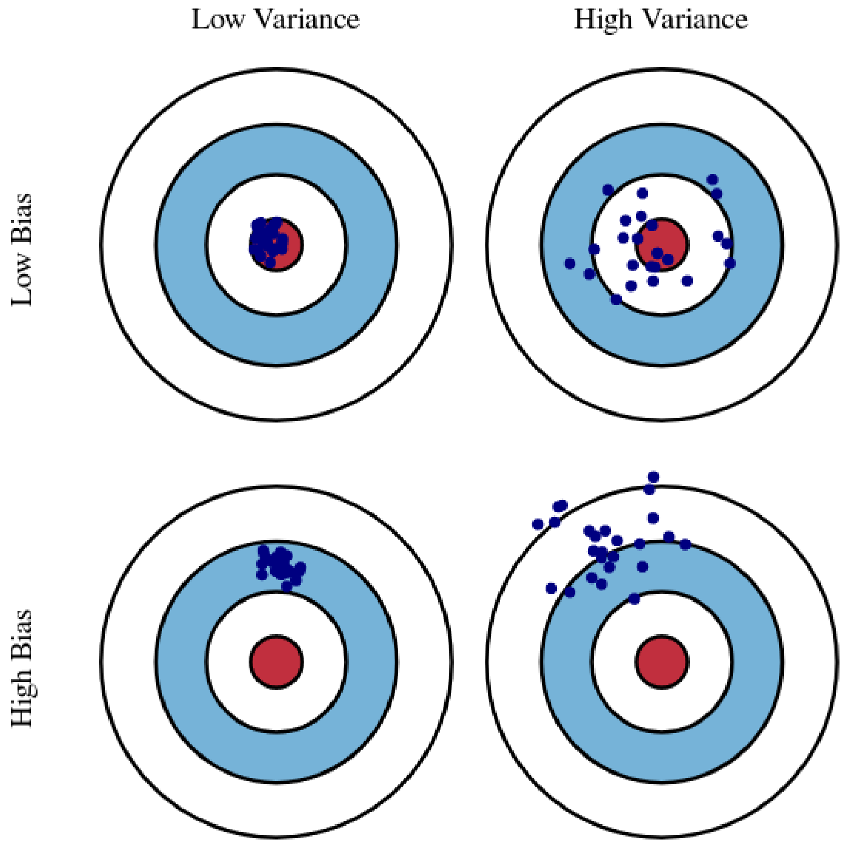
Bias-variance trade-off

Underfitting and Overfitting



Simple models:
High bias and low variance

Complex models:
High variance and low bias



https://courses.grainger.illinois.edu/cs446/sp2015/Slides/Lecture05.pdf