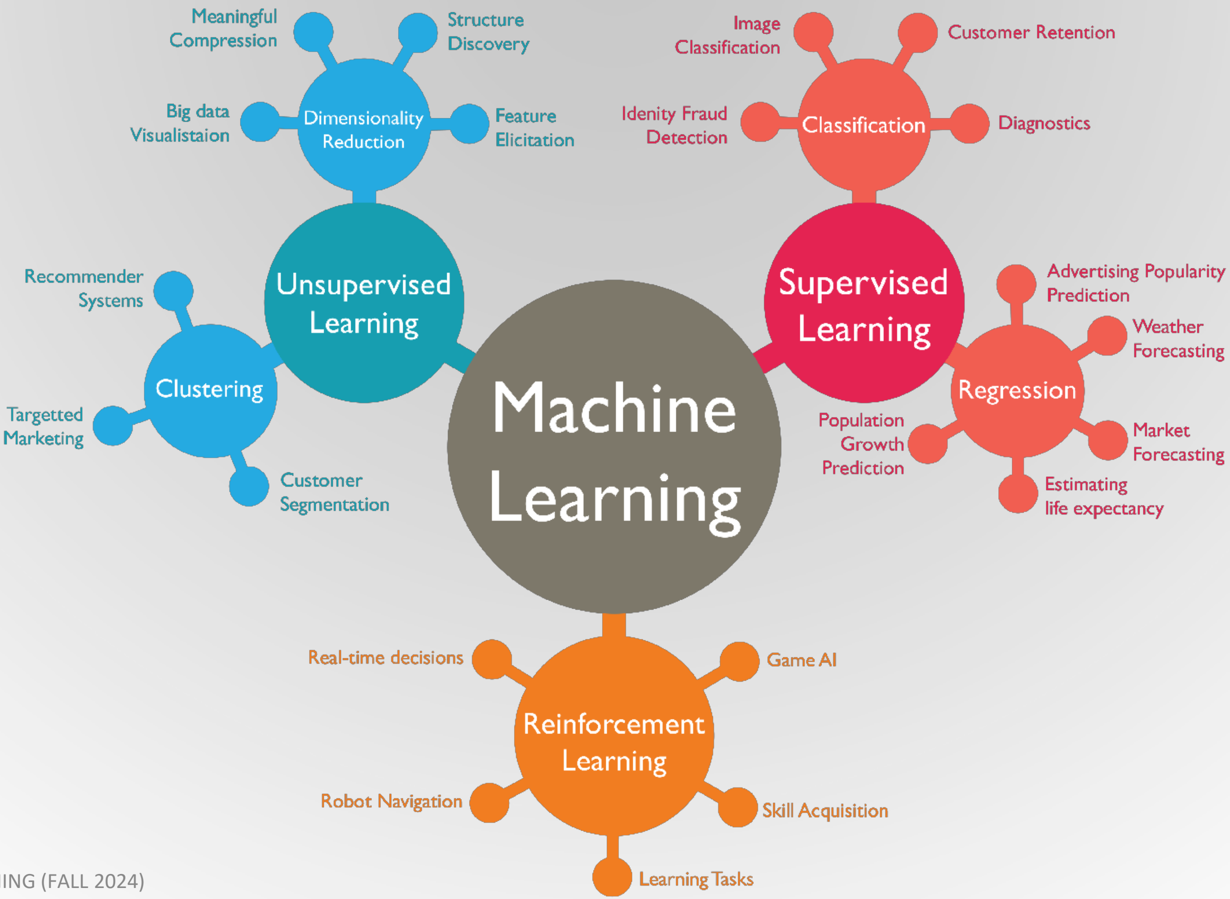
CSC 462: Machine Learning

3.1 Linear Regression

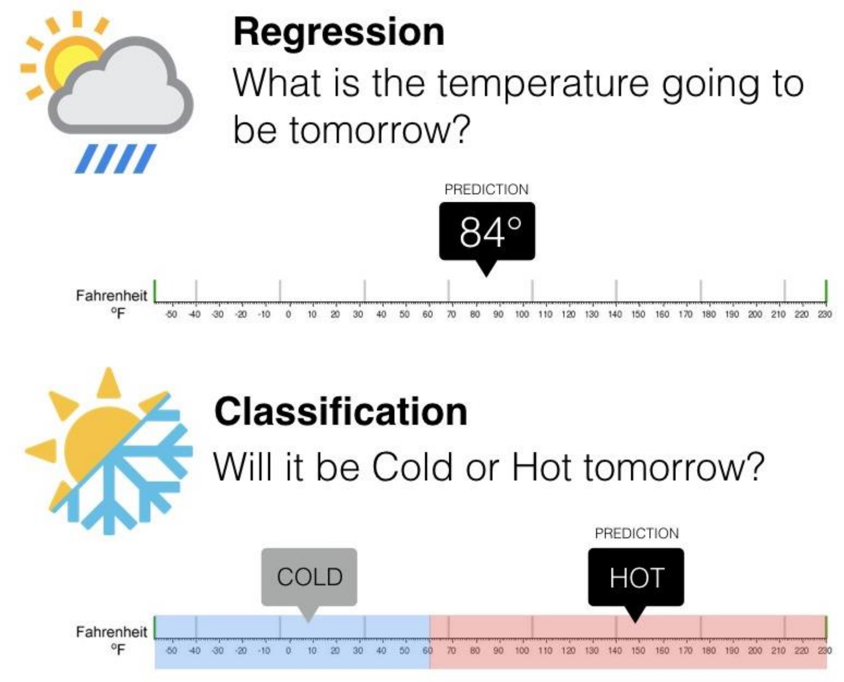
Machine Learning Approaches



Chapter 2

Classification vs. Regression

- **Classification** is a problem of automatically assigning a **label** to an **unlabeled example**.
- **Regression** is a problem of predicting a real-valued label (often called a *target*) given an unlabeled example.

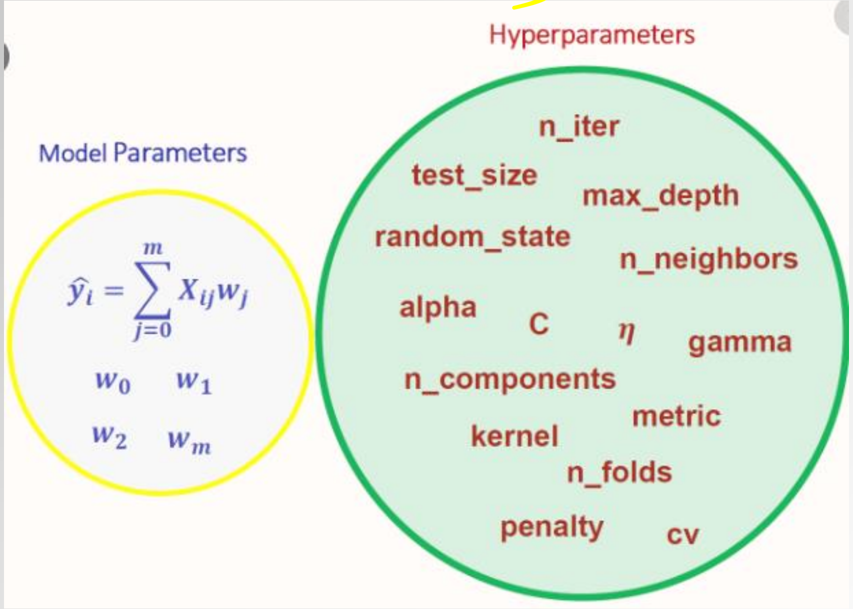


Parameters vs. Hyperparameters

setting before learning

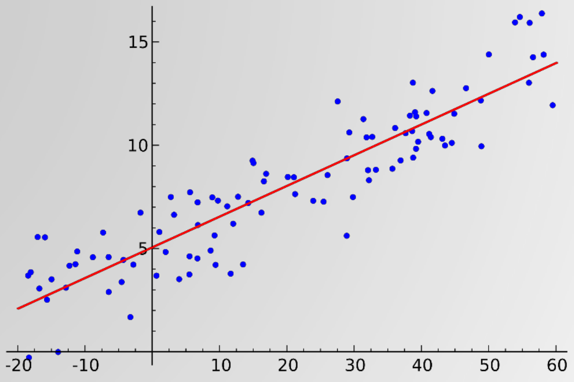
- A **hyperparameter** is a property of a learning algorithm, usually (but not always) having a numerical value.
- **Parameters** are variables that define the model learned by the learning algorithm. Parameters

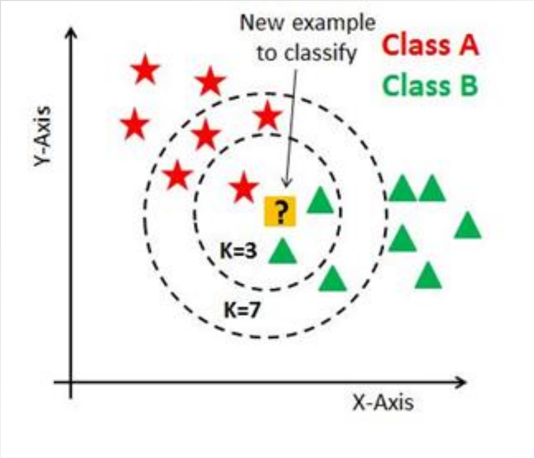
after



Model-Based vs. Instance-Based Learning

- Model-based learning algorithms use the training data to create a **model** that has **parameters** learned from the training data.
  - SVM
- Instance-based learning algorithms use the whole dataset as the model.
  - kNN



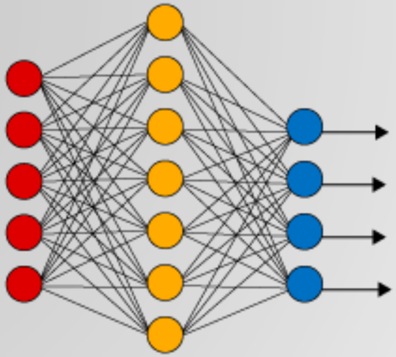


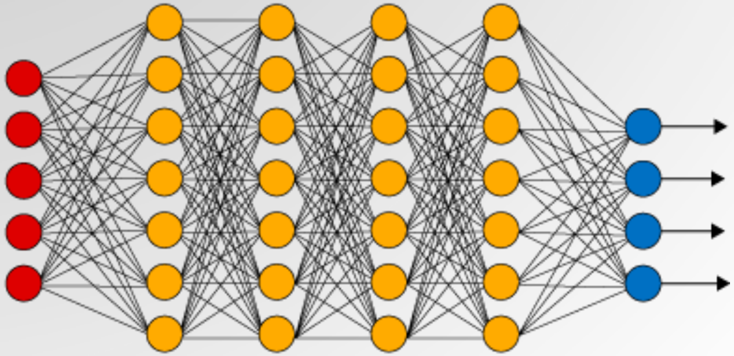
Shallow vs. Deep Learning

- A **shallow learning** algorithm learns the parameters of the model **directly** from the features of the training examples.

خطأ

- In **deep learning**, most model parameters are learned not directly from the features of the training examples, but **from the outputs of the preceding layers**.





Shallow vs. Deep Learning

Shallow Learning	Deep Learning
• Linear Regression • Logistic Regression • Decision Tree Learning • Support Vector Machine • k-Nearest Neighbors • ….	• Neural Networks

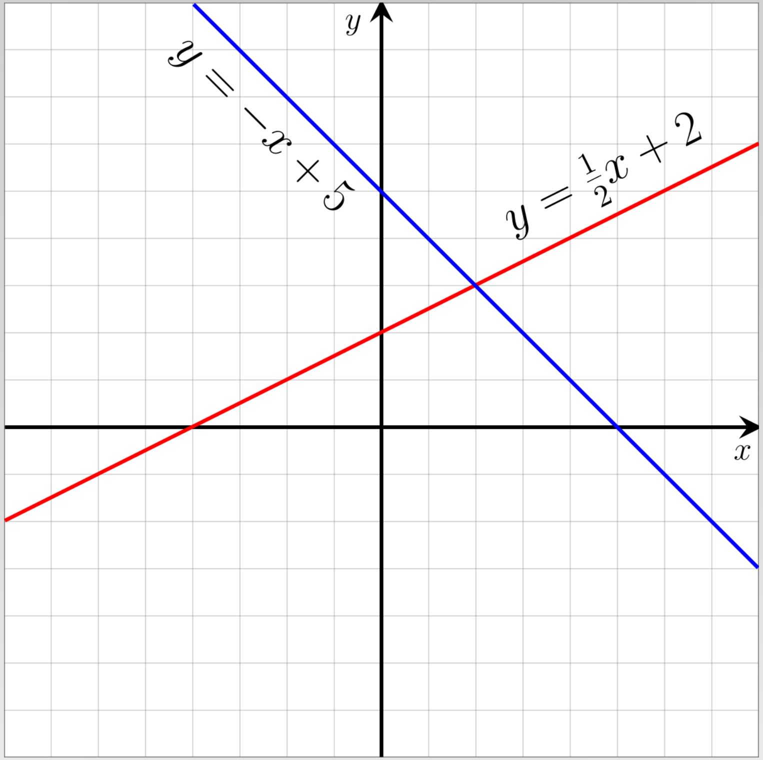
Linear Regression

Linear Equation

Coeffs

b =

W =



slop

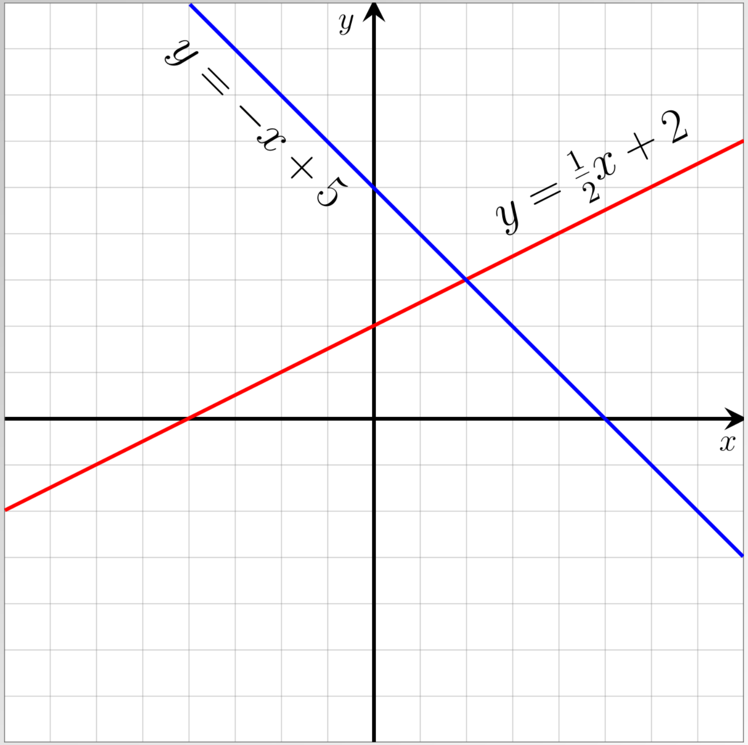
$y = wx + b$

Linear Regression

- The main idea of Linear regression is to fit a straight line through the data, where it is as closer as possible from every data point.
- In case we have **one feature *x***, the equation will be as follows:

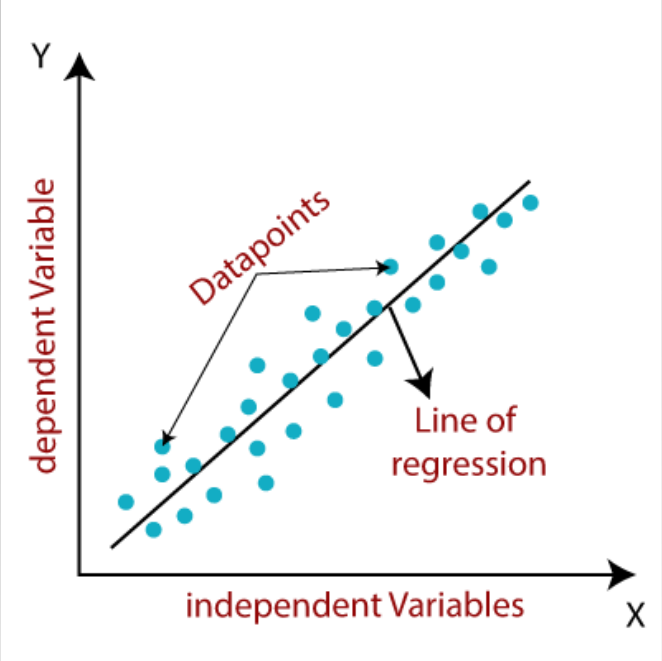
$$y = wx + b$$

The notation sometimes can be different.
For example:
$y = b_1 x + b_0$
is another notation for
$y = wx + b$



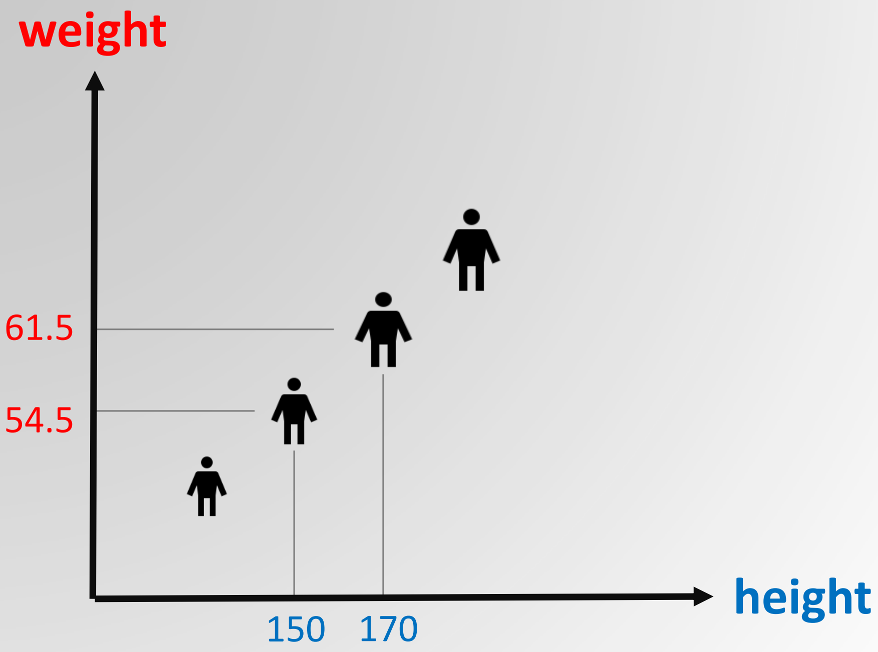
Linear Regression

- Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (x) variables, hence called as linear regression.



Human Wieght Prediction Example

- Let's say we want to predict someone's weight based on his height.
  - Feature (x) : height (cm)
  - predicted variable (y) : weight (kg)
- We can see that the more your height increase the more weight we will gain.



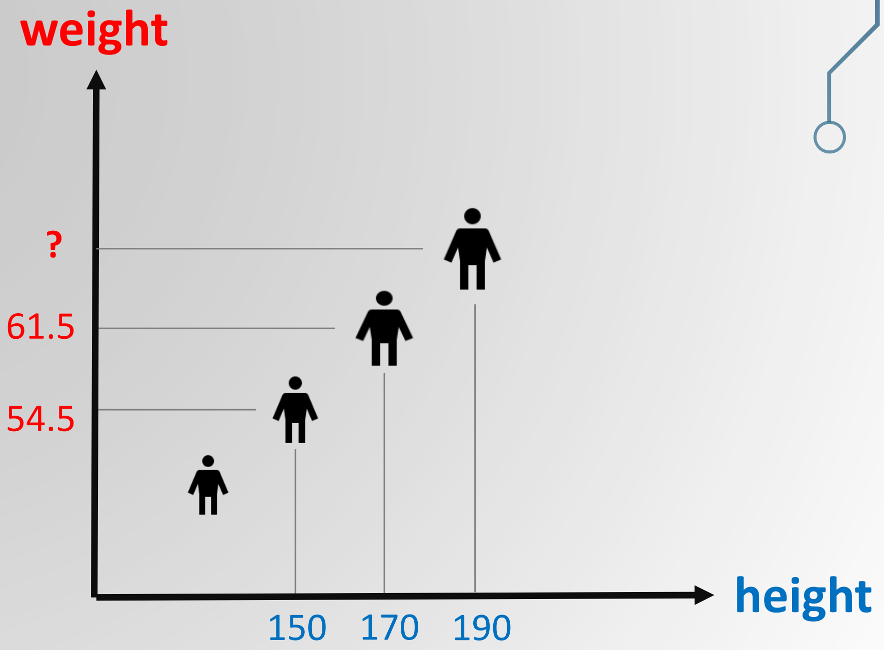
Human Wieght Prediction Example

- Now let's give some attention to the previous equation: $y = wx + b$
  - w is called weight
  - b is called bias or intercept

- Suppose we the weight and bias are as follows:

$$y = 0.35\,x + 2$$

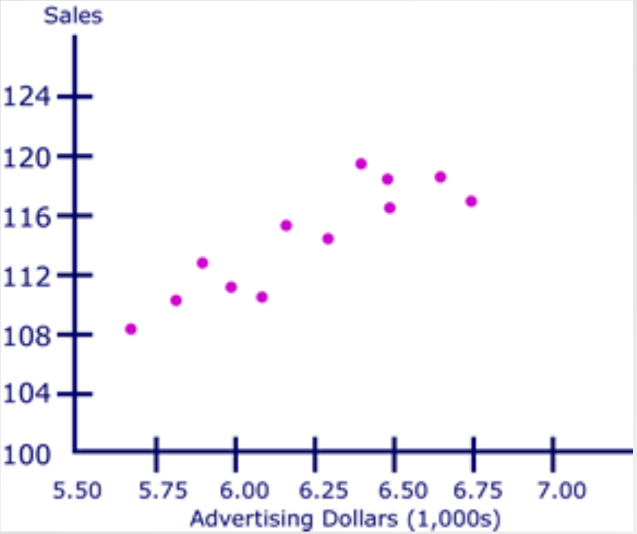
- Now we can use them to predict the human weight based on the height
  - Example: $0.35 \times 190 + 2 = 68.5$



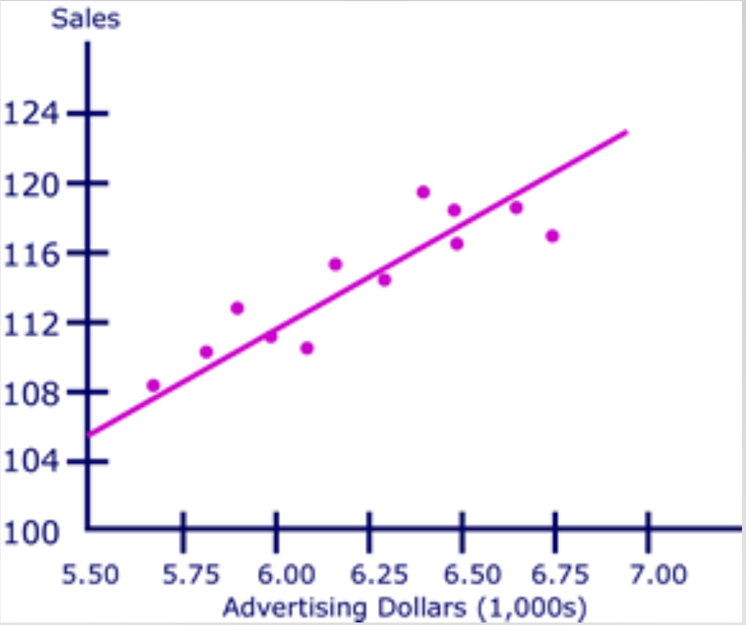
Height	Weight
150	54.5
170	61.5
190	?

Example: Dollars Spent (Monthly) for Advertisement and the Sales Recorded

Month	Sales (in 1000s)	Advertising Dollars (1000s)
January	100	5.5
February	110	5.8
March	112	6
April	115	5.9
May	117	6.2
June	116	6.3
July	118	6.5
August	120	6.6
September	121	6.4
October	120	6.5
November	117	6.7
December	123	6.8

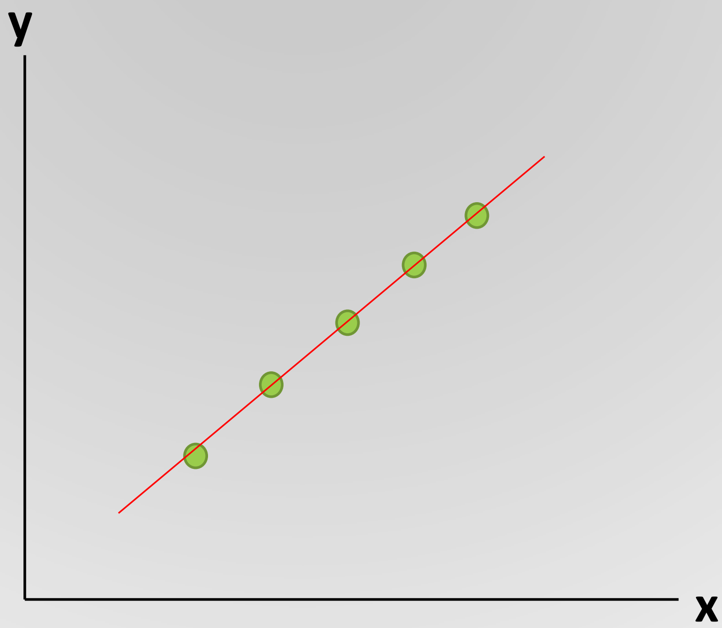


Advertising Dollars and Sales Linear Relationship



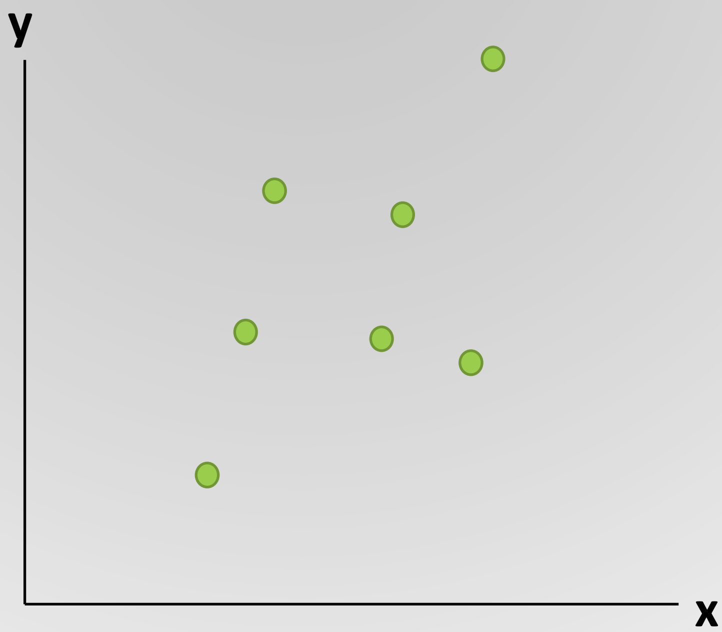
There is a positive linear relationship between advertising dollars and sales.

If we have the following points, how can we fit a line?



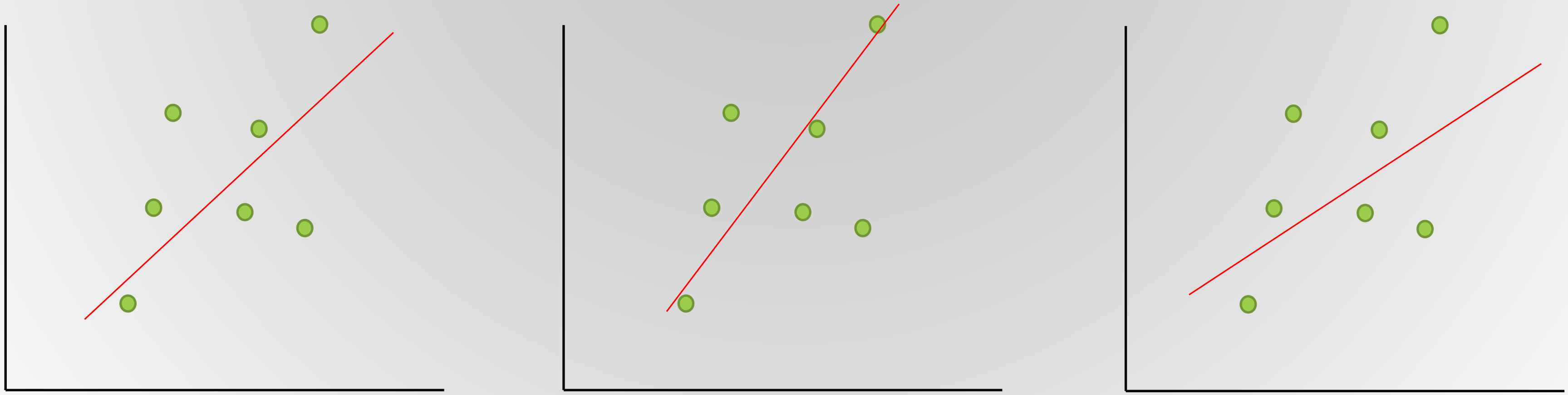
$$y = wx + b$$

What about this one, how to fit the line?



$$y = wx + b$$

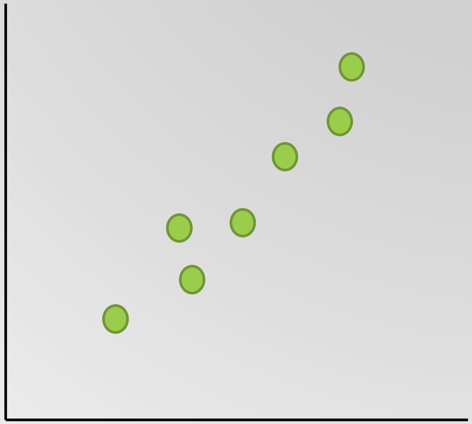
Which one?



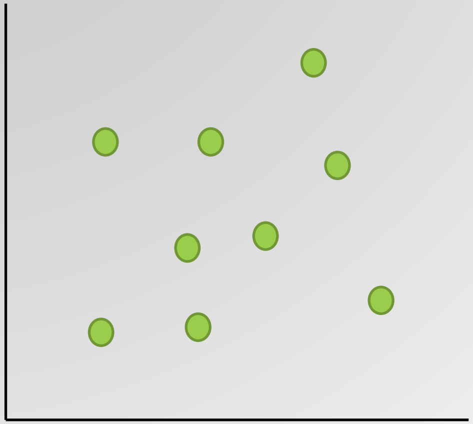
$$y = wx + b$$

Noisy data

It becomes **harder** to determine how to fit the data when we have more data points or when the data points are **noisy**



Normal



Noisy

Finding the best fit line

- When working with linear regression, our main goal is to find the **best fit line** that means the **error** between predicted values and actual values should be **minimized**.
  - The best fit line will have the least error.

- **Cost function**[1] is used to estimate the values of the coefficient (*w* & *b*) for the best fit line.
- For Linear Regression, we can use the **Mean Squared Error (MSE)** cost function
  - The average of squared error occurred between the predicted values and actual values:

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}$$

y is the actual value
$\hat{y}$ is the predicted value
n is the number of data points

[1] The term cost is often used as synonymous with loss. However, some authors make a clear difference between the two. For them, the cost function measures the model's error on a group of objects, whereas the loss function deals with a single data instance.

Regression Evaluation

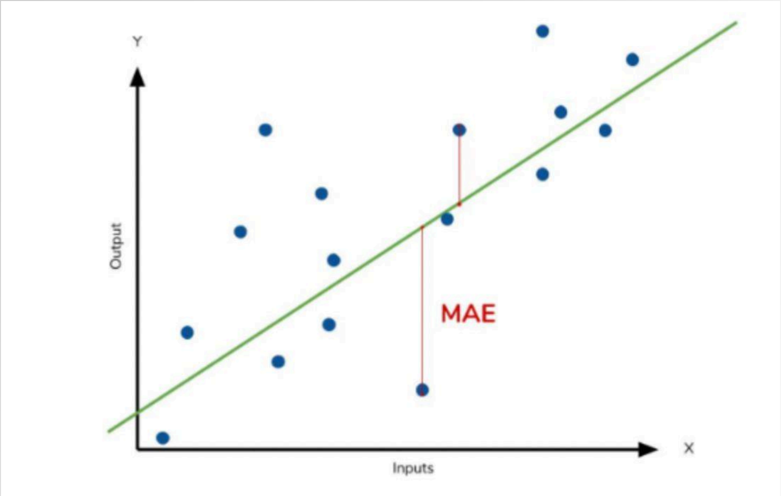
- The most common metrics for evaluating regression learning problem predictions are:
  - Mean Absolute Error (MAE)
  - Mean Squared Error (MSE)
  - Root Mean Squared Error (RMSE)
  - Mean Absolute Percentage Error (MAPE) *
  - R2 *

Day	Actual Temp	Predicted Temp	Error	Absolute Error	Squared Error
1	20	22	-2	2	4
2	19	17	2	2	4
3	18	21	-3	3	9
4	19	18	1	1	1
5	18	18	0	0	0
6	20	18	2	2	4
7	21	21	0	0	0
8	19	18	1	1	1
9	20	23	-3	3	9
10	21	19	2	2	4
Total			**0**	**16**	**36**
Average			**0**	**1.6**	**3.6**

* We will study it later

Mean Absolute Error (MAE)

- The Mean Absolute Error (MAE) is the average of the absolute differences **between predictions and actual values**
- It gives an idea of **how wrong the predictions were**
- The measure gives an idea of the magnitude of the error
  - But no idea of the direction (e.g., over or under predicting)



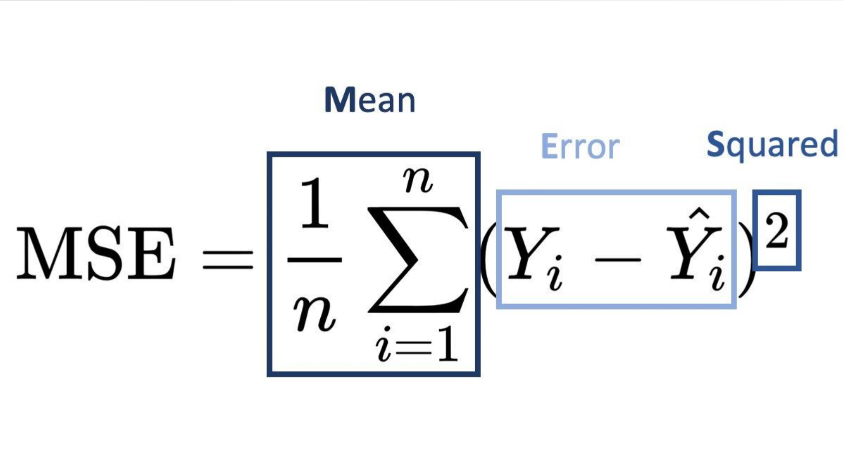
$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

- y is the actual value
- $\hat{y}$ is the predicted value
- n is the number of data points

Mean Squared Error (MSE)

- If the MSE of the model on the test data is substantially higher than the MSE obtained on the training data, this is a sign of overfitting.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$



- y is the actual value
- $\hat{y}$ is the predicted value
- n is the number of data points

Root Mean Squared Error (RMSE)

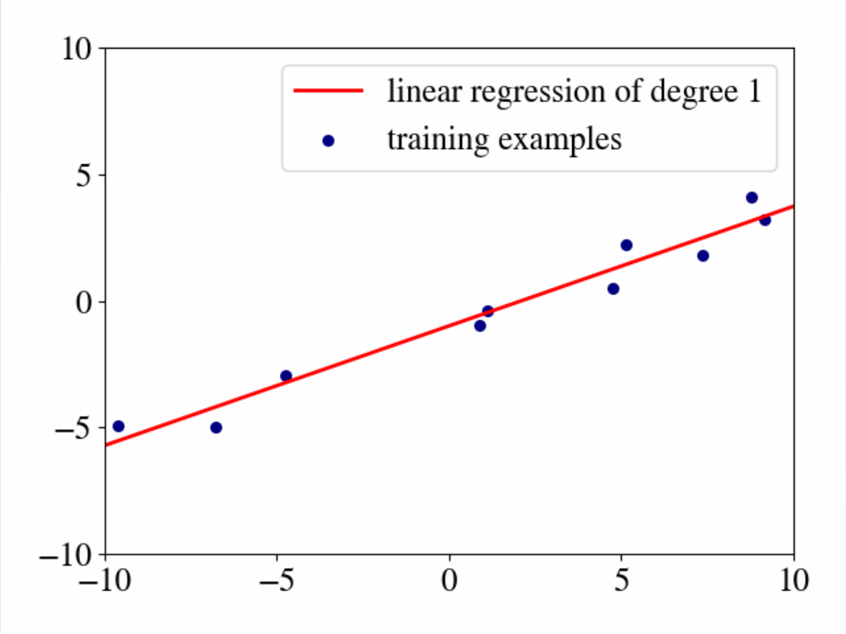
- The Root Mean Squared Error (RMSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of the error
- Since the errors are squared before they are averaged, the RMSE gives a relatively **high weight to large errors**
  - This means the RMSE should be more useful when large errors are particularly undesirable

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

- y is the actual value
- $\hat{y}$ is the predicted value
- n is the number of data points

Multiple Linear Regression

- Height alone is not enough to predict someone's weight.
- What if we want to use other features like age, gender and lifestyle ?
- We can use the same equation with:



$$y = wx + b$$ (Simple Linear Regression)

$$y = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$$ (Multiple Linear Regression)

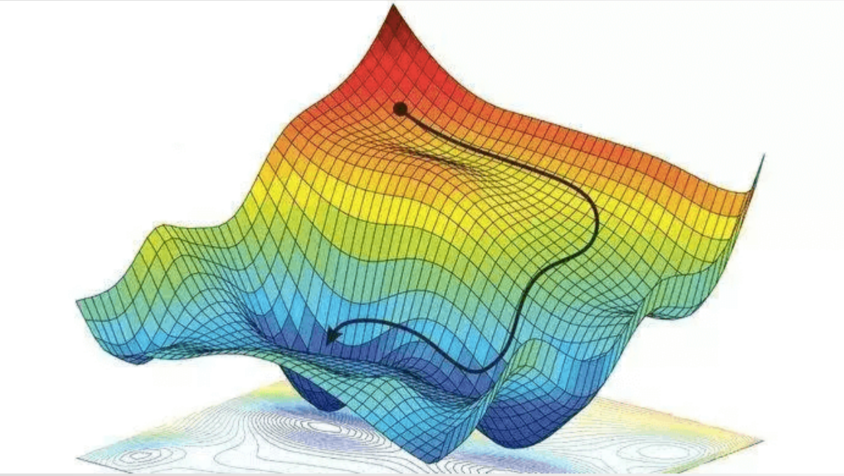
How to determine the values of weight and bias?

- Weight and bias are learnable, that means they keep changing until we find some values that we believe they are the best for the solution

$$y = 2x + 3$$
$$y = 1.5x + 5$$
$$y = 0.9x + 1$$
$$\vdots$$
$$y = 0.35x + 2$$

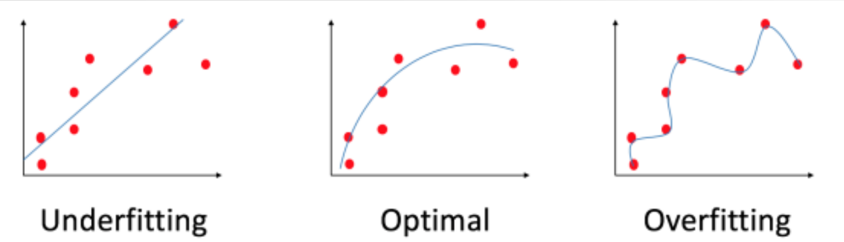
Gradient Descent

- Gradient descent is a method of updating w and b to reduce (minimize) the cost function (for example, MSE).
  - Gradient descent minimizes the MSE by calculating the gradient of the cost function.
- A regression model uses gradient descent to update the coefficients of the line by reducing the cost function.
- It is done by a random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.



Overfitting & Underfitting

- Underfitting
  - Poor performance on the training data and poor generalization to other data.
- Overfitting
  - Good performance on the training data, poor generalization to other data.



Advantages & Disadvantages

Advantages

- Can be regularized to avoid overfitting
- Small number of hyperparameters
- Easy to understand and explain

Disadvantages

- Prone to overfitting with many features are present.
- May not work well when the hypothesis function is non-linear.
- Input data need to be scaled and there are a range of ways to do this.