

# 머신러닝의 기초

# 프로그래밍이란

---

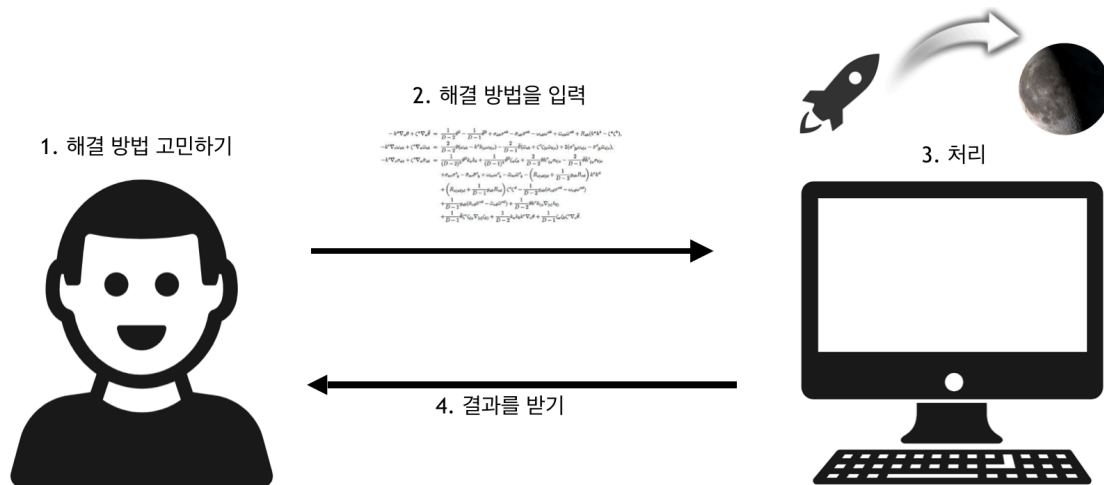
입력값( $X$ )에 대해 원하는 출력값( $Y$ )가 나오도록 컴퓨터 명령어를 구성하는 것



# 프로그래밍의 두가지 방법

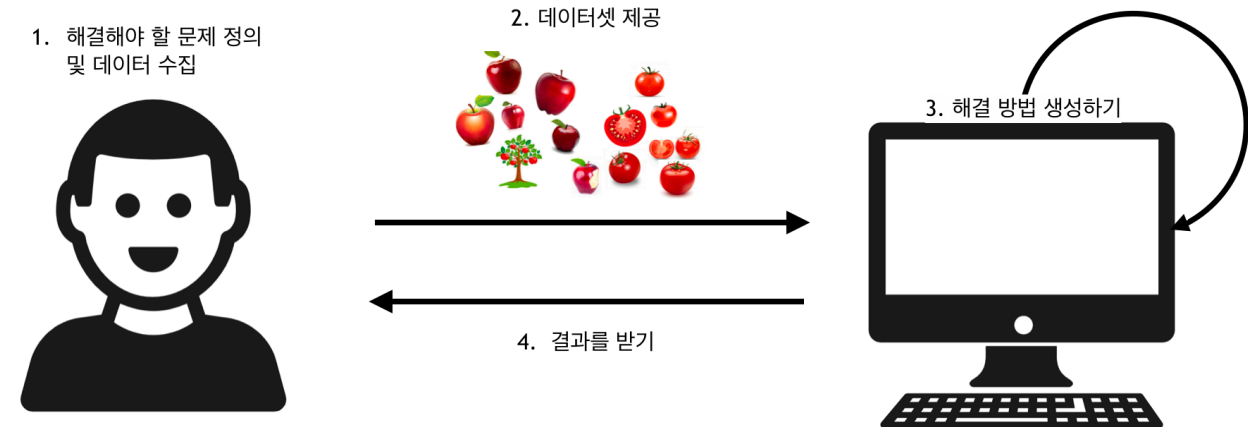
## 1. 명시적 프로그래밍

개발자가 입력조건과 프로그램 상태 조건에 따라 프로그램이 동작하는 방식을 직접 구현



## 2. 머신러닝 프로그래밍

프로그램이 데이터를 보고 스스로가 학습하여 동작방식을 결정하는 프로그래밍



# 머신러닝 프로그래밍이 적합한 문제

---

머신러닝이 적합한 문제 = 규칙이 모호한 문제

Q) 이미지 정보를 통해 사과를 분류하는 문제



• 색 기준



• 형태 기준



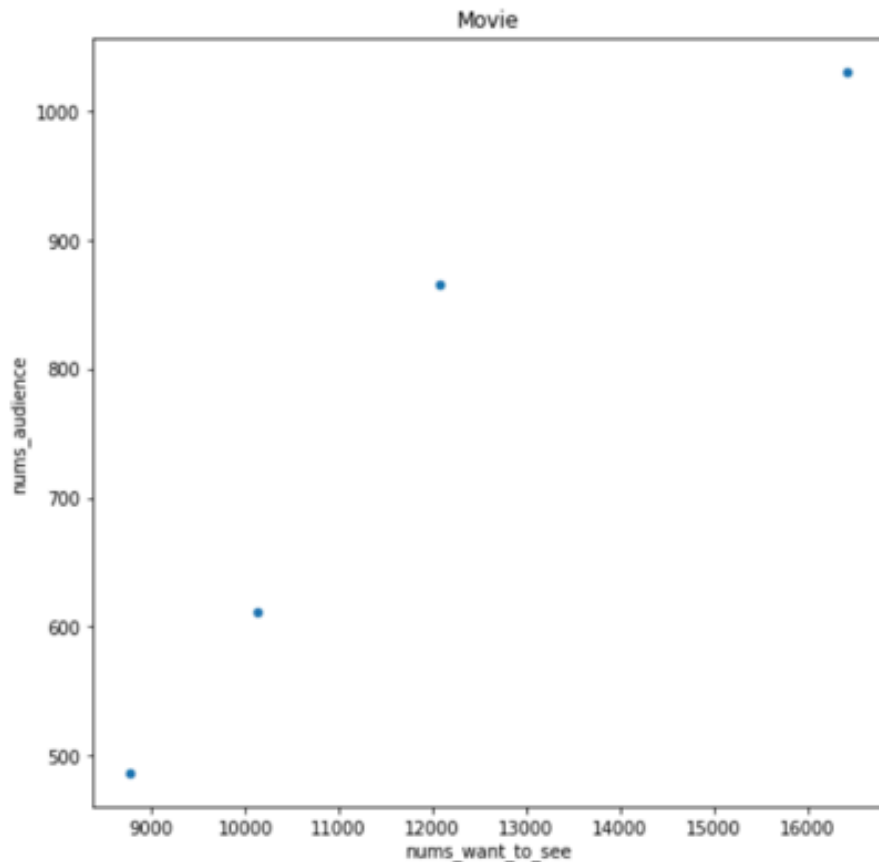
• 국소 특징 기준



# 머신러닝의 기초 : 선형회귀

## 문제

왓차에서 제공하는 "보고싶어요" 수와 관객 수 간의 관계를 바탕으로, 영화 "옥자"의 예상 관객 수 예측하기



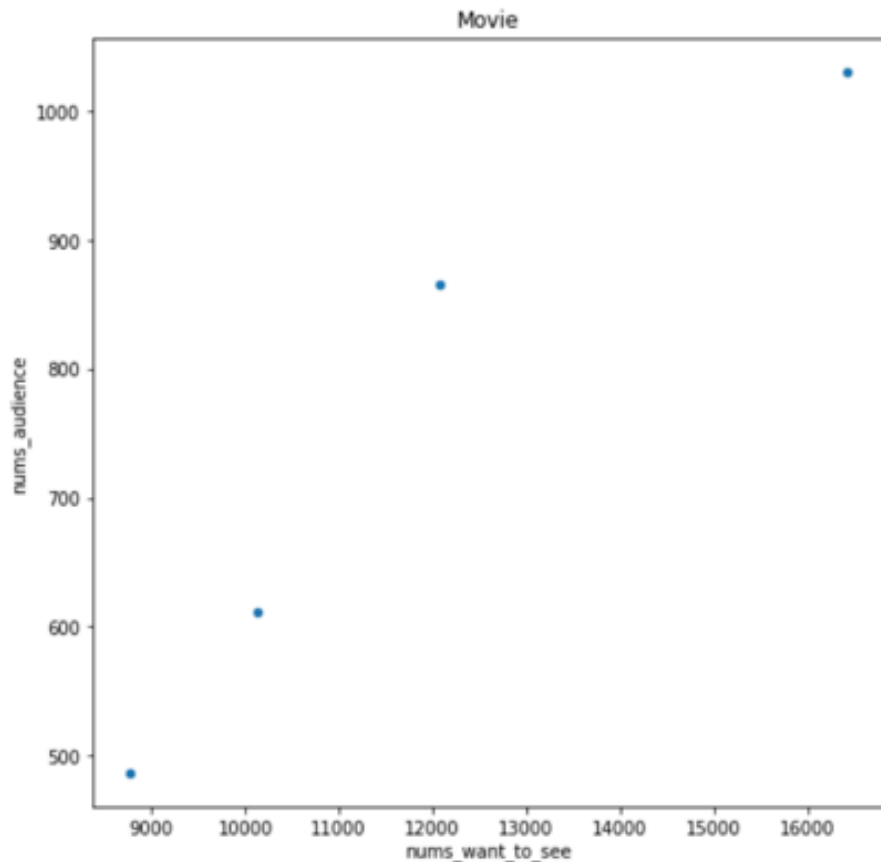
## 데이터

	보고싶어요 수(명)	총 관객 수 (만명)
마션	8759	487
킹스맨	10132	612
캡틴아메리카	12078	866
인터스텔라	16430	1030
옥자	12,008	?

# 머신러닝의 기초 : 선형회귀

## 문제

왓차에서 제공하는 "보고싶어요" 수와 관객 수 간의 관계를 바탕으로, 영화 "옥자"의 예상 관객 수 예측하기



## 데이터

	보고싶어요 수(명)	총 관객 수 (만명)
마션	8759	487
킹스맨	10132	612
캡틴아메리카	12078	866
인터스텔라	16430	1030
옥자	12,008	?

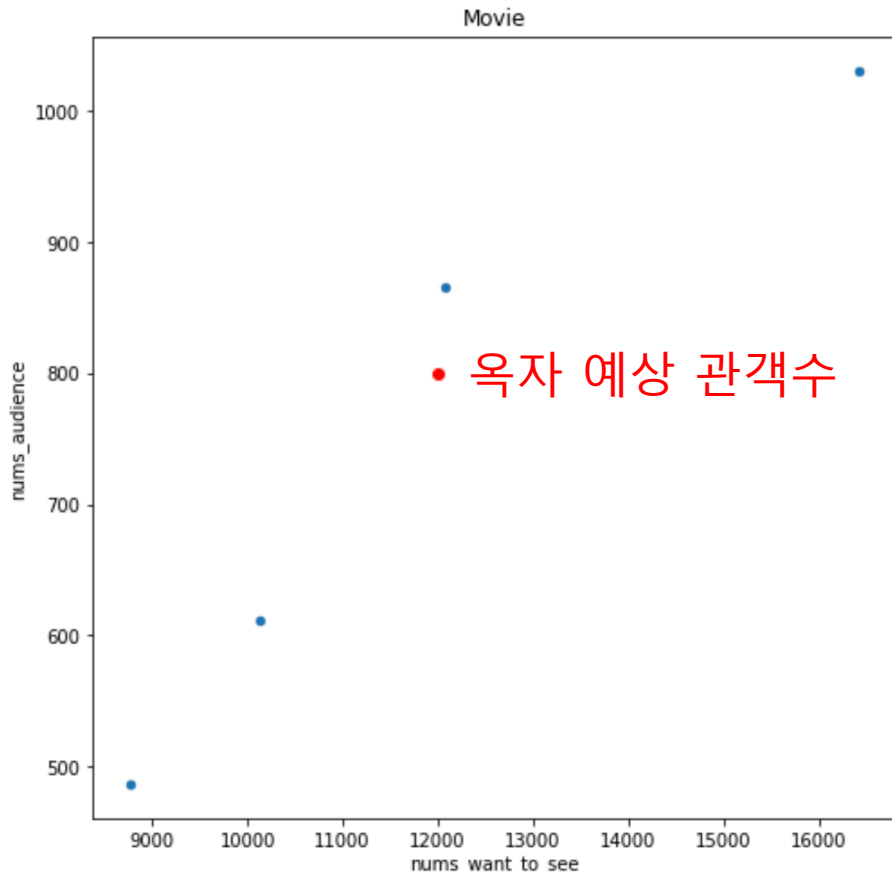
“보고싶어요” 수와 “총 관객” 수와의 관계를 파악



# 머신러닝의 기초 : 선형회귀

## 문제

왓차에서 제공하는 "보고싶어요" 수와 관객 수 간의 관계를 바탕으로, 영화 "옥자"의 예상 관객 수 예측하기



## 데이터

	보고싶어요 수(명)	총 관객 수 (만명)
마션	8759	487
킹스맨	10132	612
캡틴아메리카	12078	866
인터스텔라	16430	1030
옥자	12,008	?

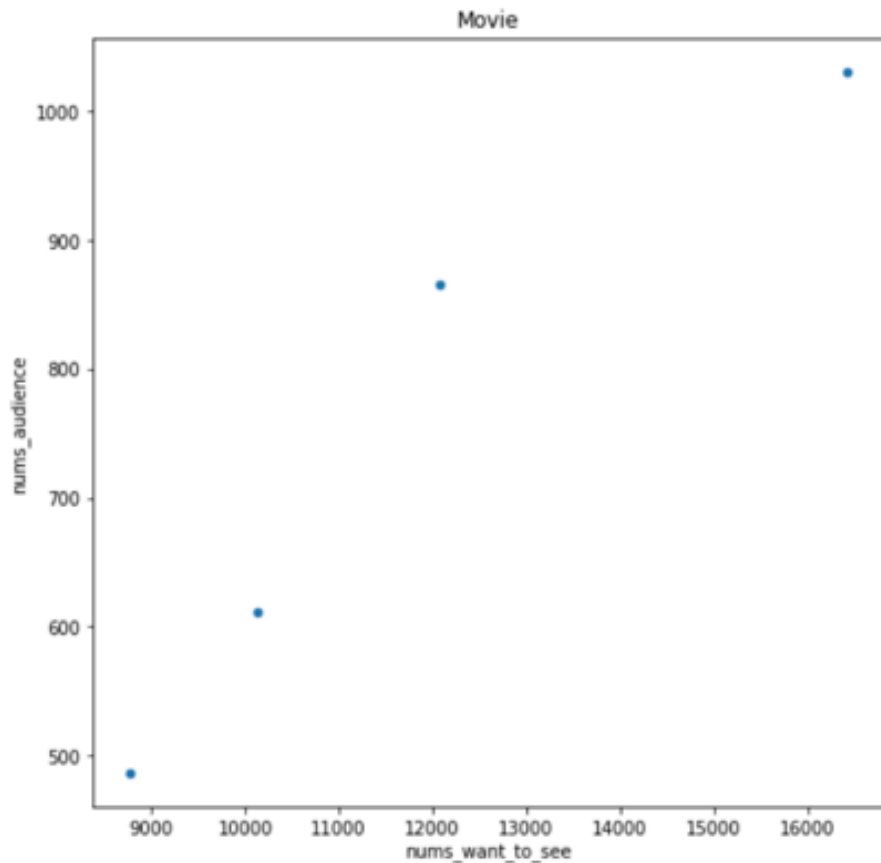
“보고싶어요” 수와 “총 관객” 수와의 관계를 파악



# 선형회귀 모델링

---

## (1) 입력값과 출력값 정의



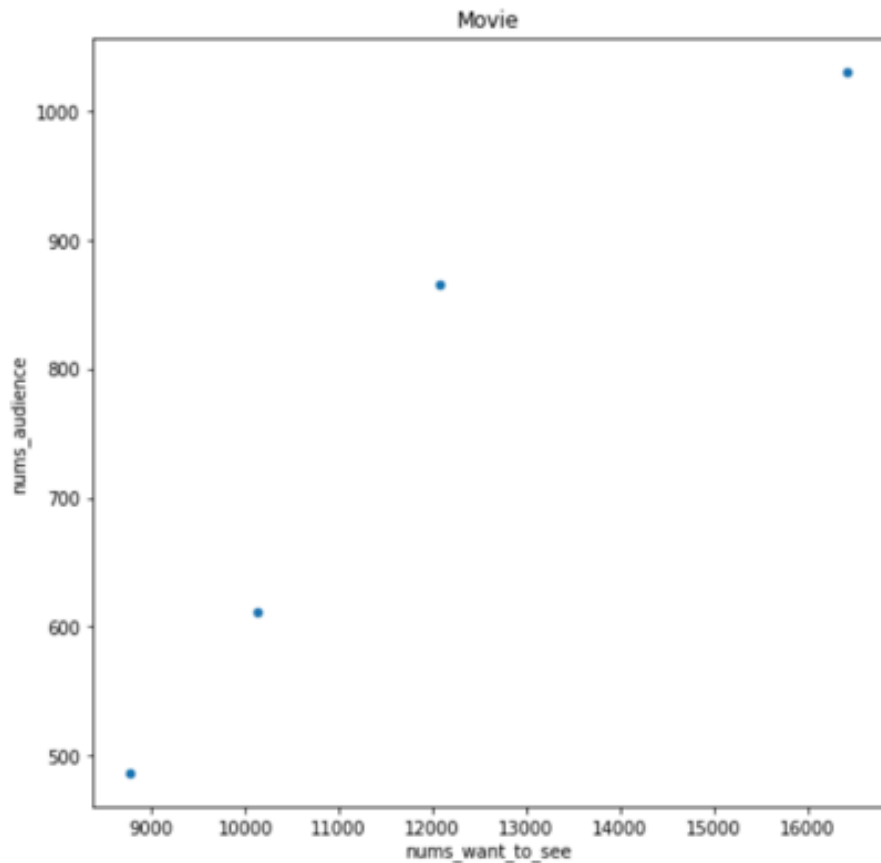


# 선형회귀 모델링

---

## (1) 입력값과 출력값 정의

- 입력값( $X$ ) : "보고싶어요" 수
- 출력값( $Y$ ) : 총 관객 수



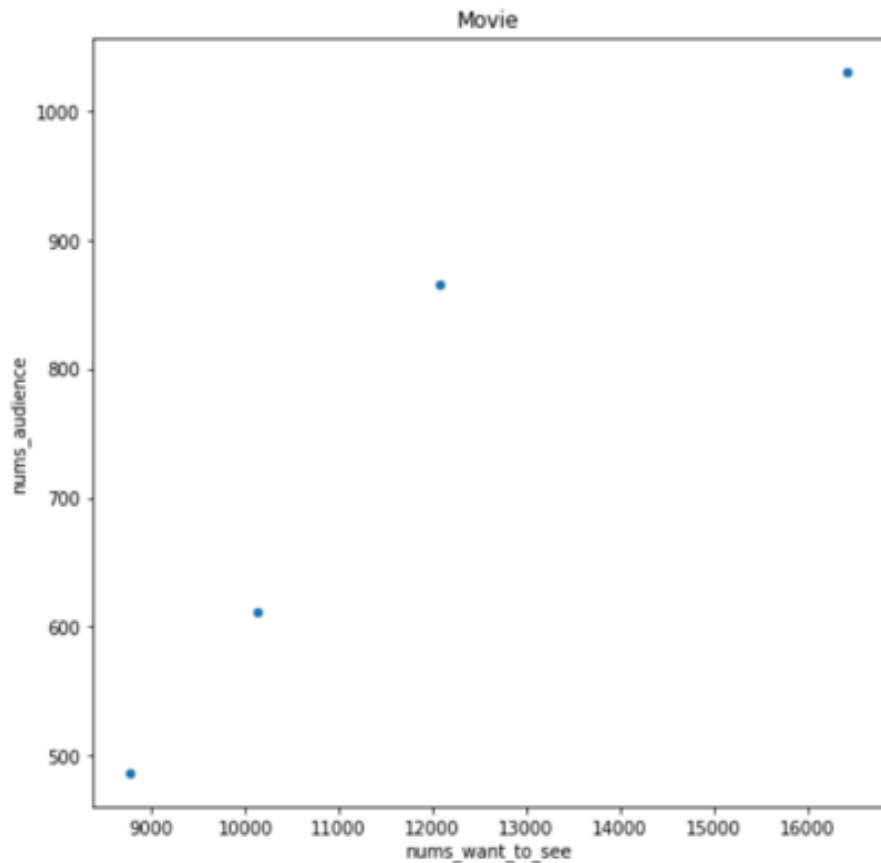
# 선형회귀 모델링

---

## (1) 입력값과 출력값 정의

- **입력값( $X$ )** : "보고싶어요" 수
- **출력값( $Y$ )** : 총 관객 수

## (2) 입력값과 출력값 관계 정의



# 선형회귀 모델링

## (1) 입력값과 출력값 정의

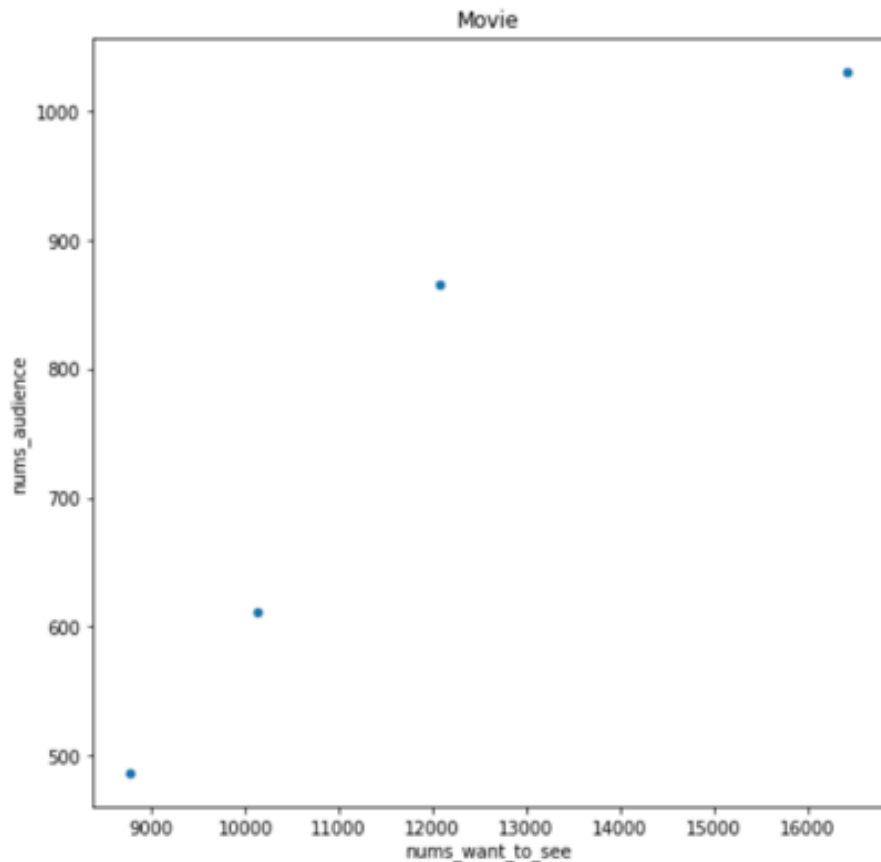
- 입력값( $X$ ) : "보고싶어요" 수
- 출력값( $Y$ ) : 총 관객 수

## (2) 입력값과 출력값 관계 정의

보고싶어요 수와 총 관객 수 : 직선의 방정식 관계

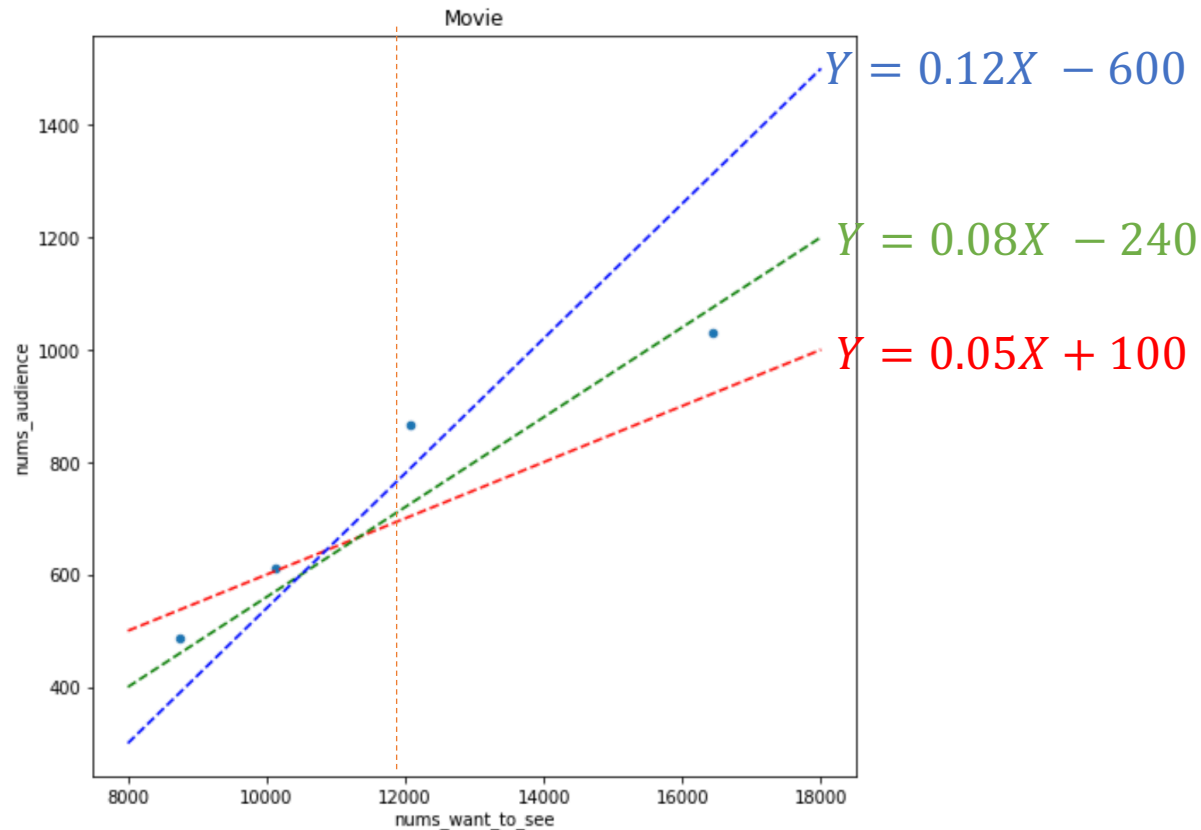
$$Y = \underline{a}X + \underline{b}$$

가중치 : 우리가 찾아야할 값



# 가중치 조합에 따른 예측값

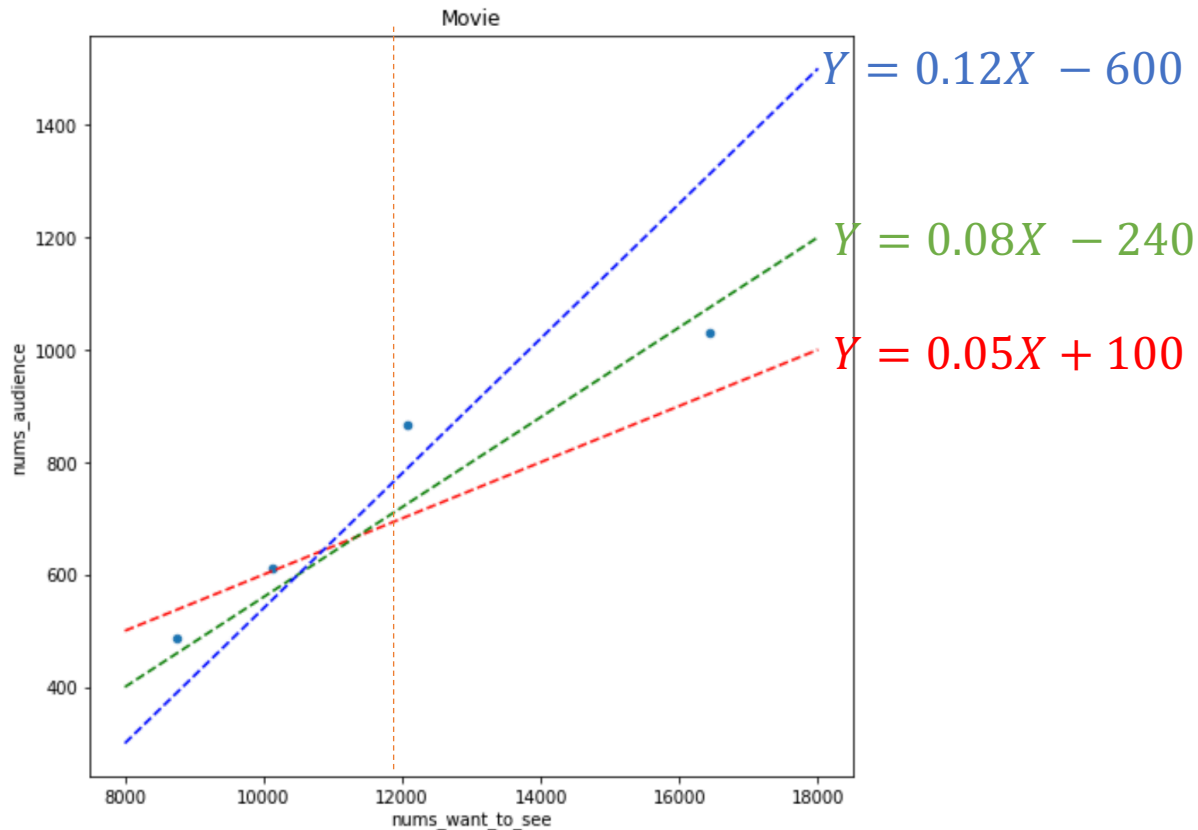
모델링에 따른 다양한 가능한 가중치의 조합이 존재



$$Y = aX + b$$

# 가중치 조합에 따른 예측값

모델링에 따른 다양한 가능한 가중치의 조합이 존재

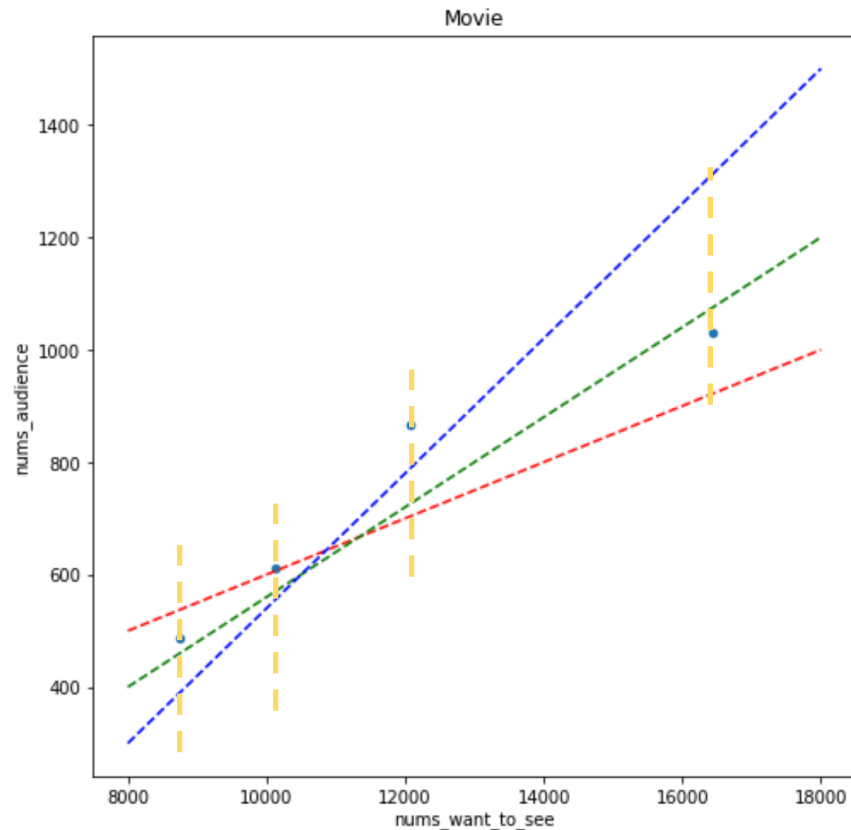


$$Y = aX + b$$

모델	예측값
파란색 예측선	840.96만명
초록색 예측선	720.64만명
빨간색 예측선	700.4만명

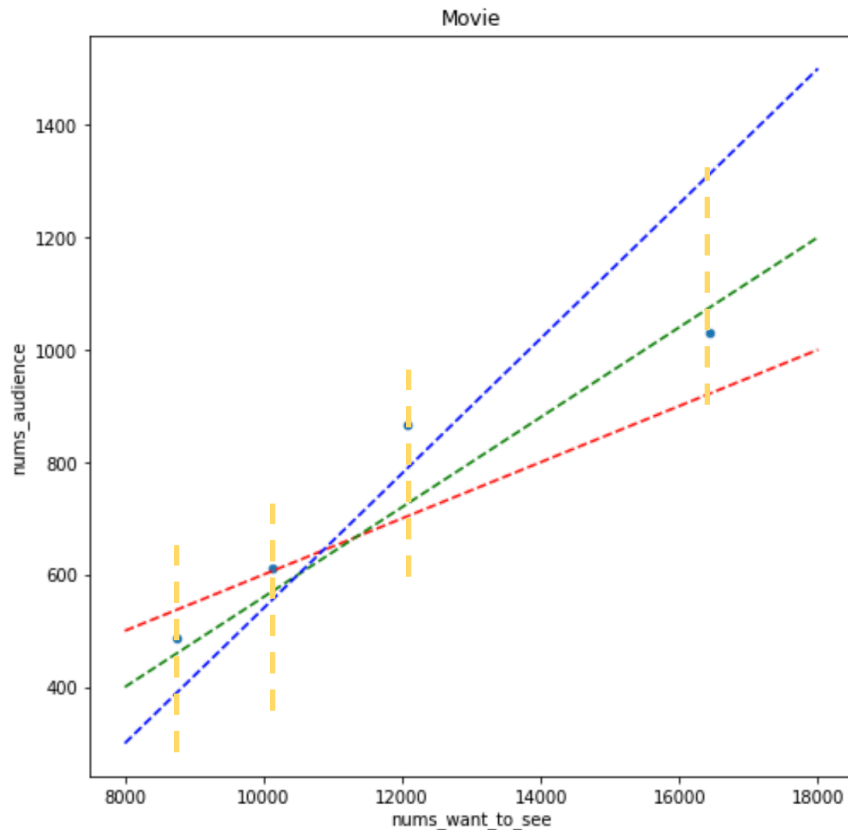
# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



# 최적 예측값의 기준 : 손실함수

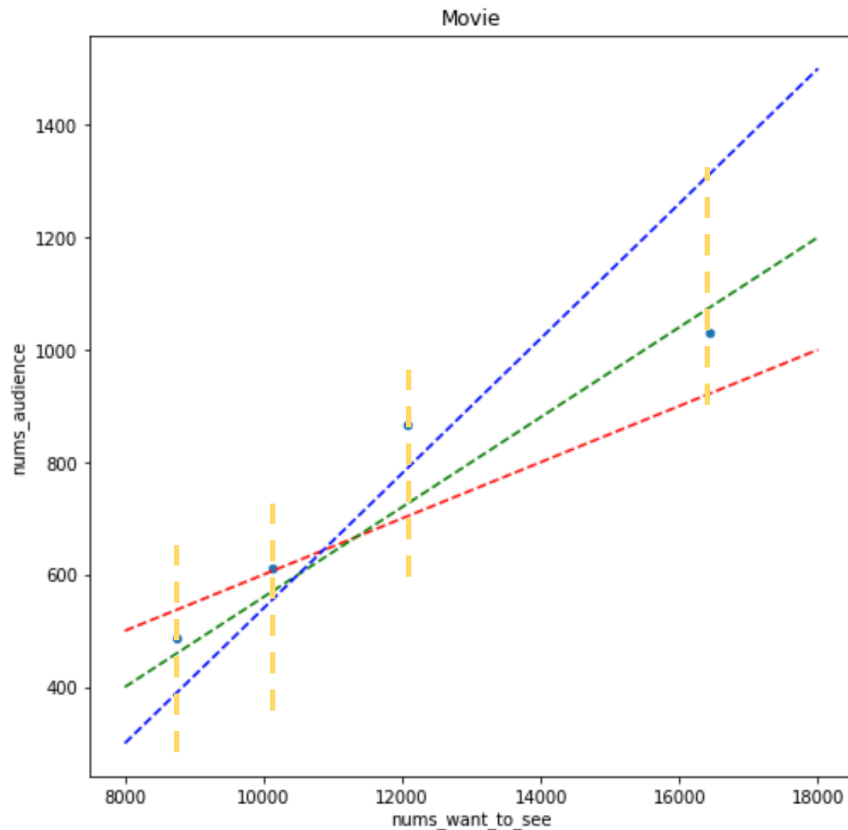
손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



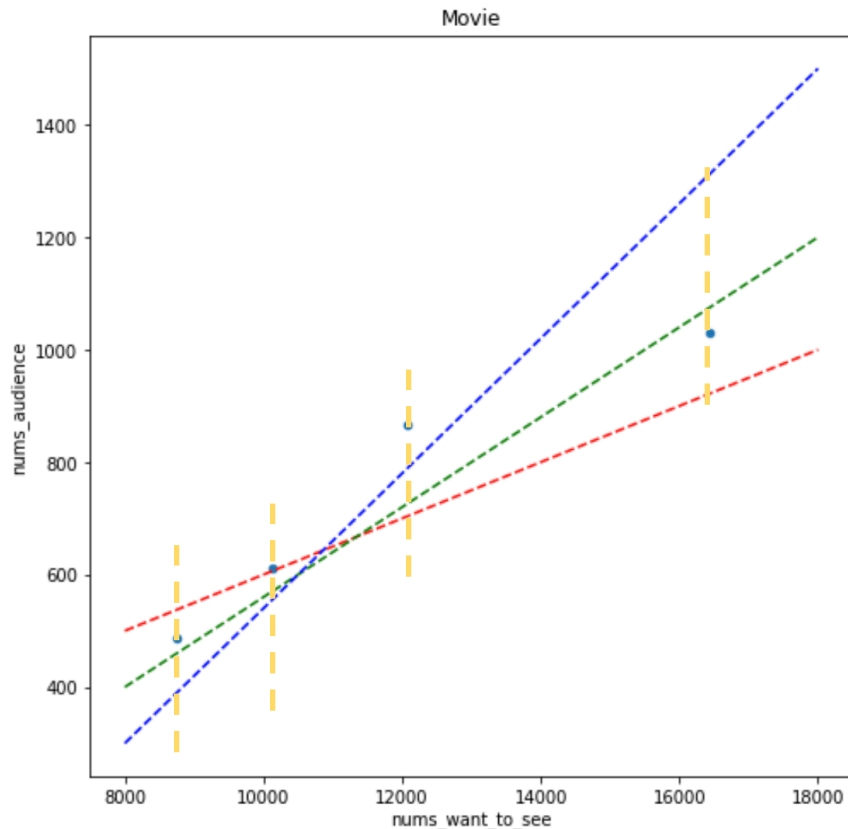
	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

**예측값과 정답값과의 차이**



# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



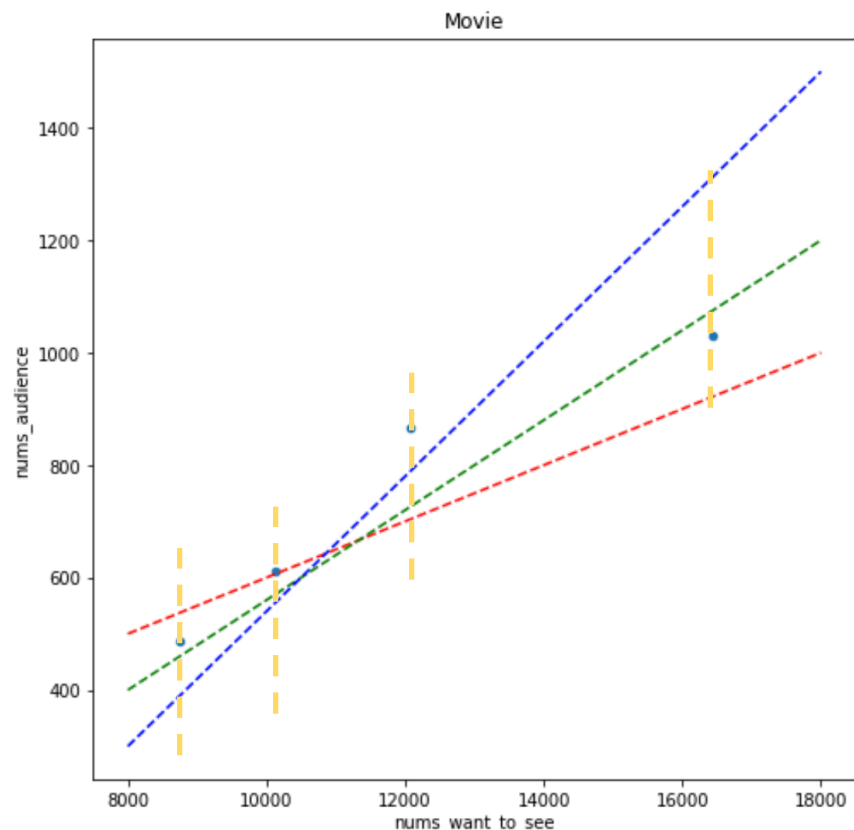
	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

**예측값과 정답값과의 차이**

$$Loss = \sum_{\text{모든 영화}} (\text{예측값} - \text{정답값})^2$$

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

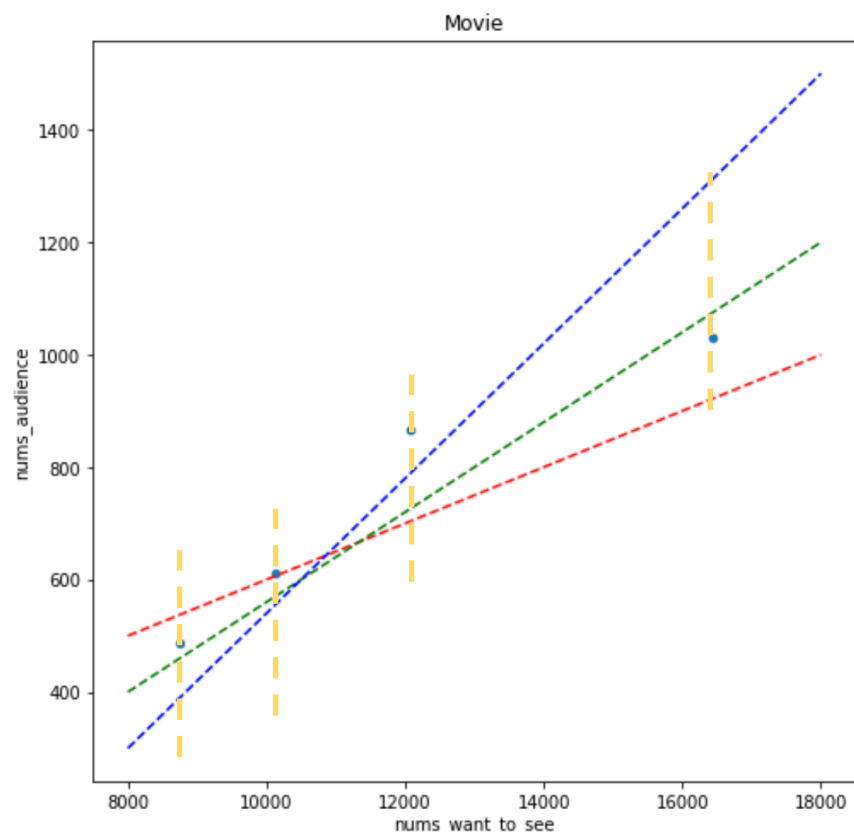
## 예측값과 정답값과의 차이

$$\text{붉은색} = (487 - 537.95)^2 + (612 - 606.6)^2 + (866 - 703.9)^2 + (1030 - 921.5)^2$$

$$\text{Loss} = \sum_{\text{모든 영화}} (\text{예측값} - \text{정답값})^2$$

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

## 예측값과 정답값과의 차이

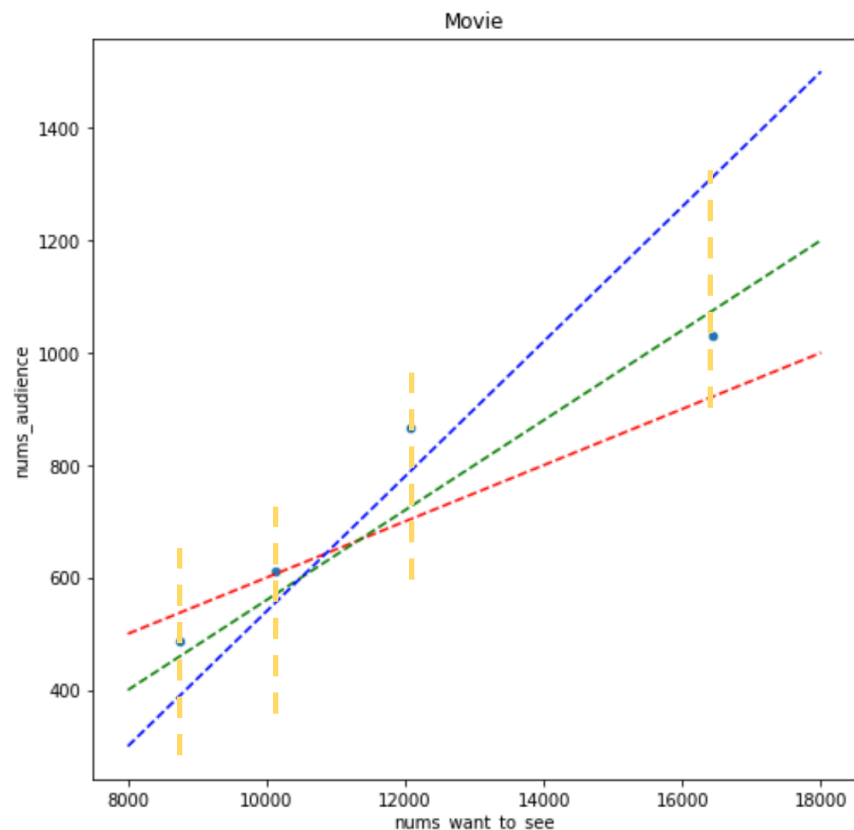
$$\text{붉은색} = (487 - 537.95)^2 + (612 - 606.6)^2 + (866 - 703.9)^2 + (1030 - 921.5)^2$$

$$\text{초록색} = (487 - 460.72)^2 + (612 - 570.56)^2 + (866 - 726.24)^2 + (1030 - 1074.4)^2$$

$$\text{Loss} = \sum_{\text{모든 영화}} (\text{예측값} - \text{정답값})^2$$

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

## 예측값과 정답값과의 차이

$$\text{붉은색} = (487 - 537.95)^2 + (612 - 606.6)^2 + (866 - 703.9)^2 + (1030 - 921.5)^2$$

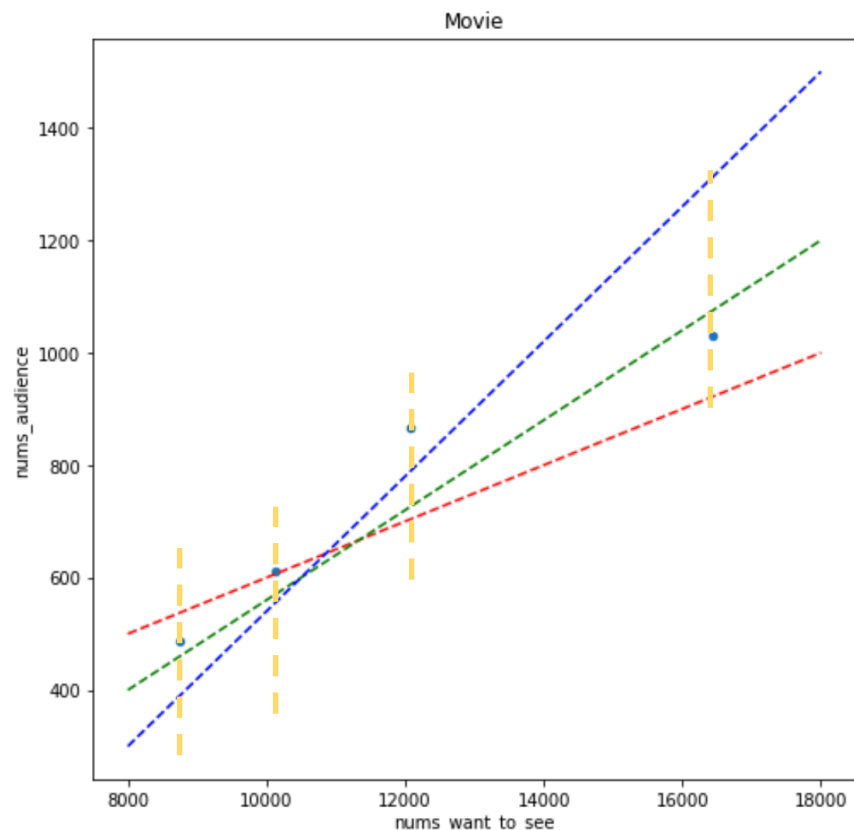
$$\text{초록색} = (487 - 460.72)^2 + (612 - 570.56)^2 + (866 - 726.24)^2 + (1030 - 1074.4)^2$$

$$\text{파란색} = (487 - 451.08)^2 + (612 - 615.84)^2 + (866 - 849.36)^2 + (1030 - 1371.6)^2$$

$$\text{Loss} = \sum_{\text{모든 영화}} (\text{예측값} - \text{정답값})^2$$

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

## 예측값과 정답값과의 차이

$$\text{붉은색} = (487 - 537.95)^2 + (612 - 606.6)^2 + (866 - 703.9)^2 + (1030 - 921.5)^2$$

$$\text{초록색} = (487 - 460.72)^2 + (612 - 570.56)^2 + (866 - 726.24)^2 + (1030 - 1074.4)^2$$

$$\text{파란색} = (487 - 451.08)^2 + (612 - 615.84)^2 + (866 - 849.36)^2 + (1030 - 1371.6)^2$$

$$\text{붉은색} = 10168.54$$

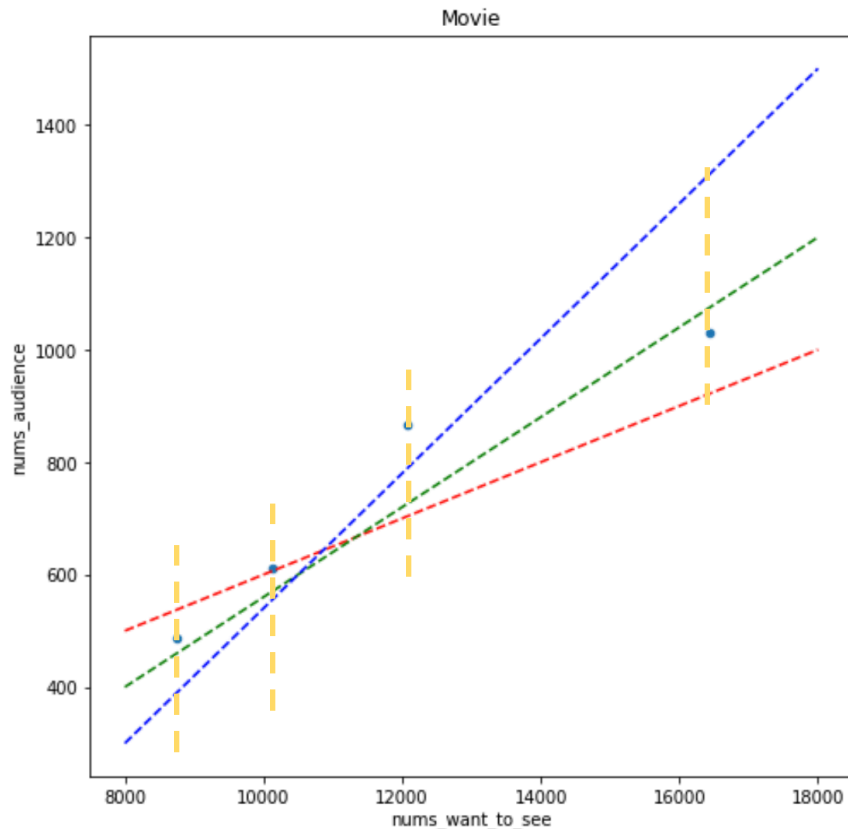
$$\text{초록색} = 5978.03$$

$$\text{파란색} = 29568.11$$

손실함수가 가장 적은  
예측 함수가 가장 좋은 함수

# 최적 예측값의 기준 : 손실함수

손실함수란 **예측값과 정답값과의 차이**를 계산하는 함수



	총 관객수	붉은색 예측	초록색 예측	파란색 예측
마션	487	537.95	460.72	451.08
킹스맨	612	606.6	570.56	615.84
캡틴아메리카	866	703.9	726.24	849.36
인터스텔라	1030	921.5	1074.4	1371.6

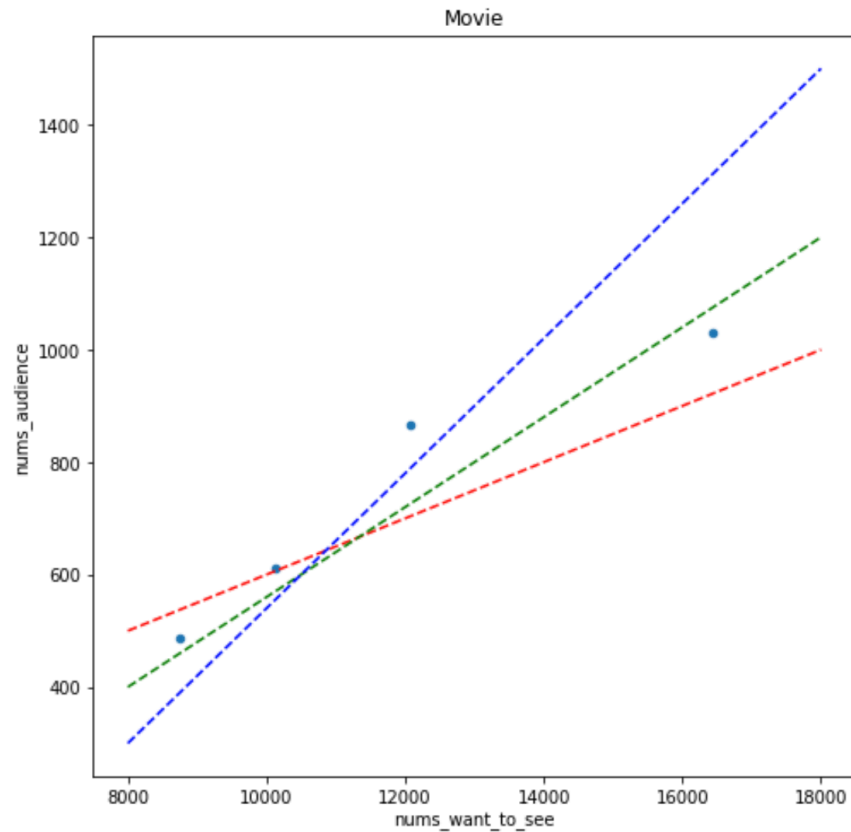
예측값과 정답값과의 차이

$$Loss = \sum_{\text{모든 영화}} (\text{예측값} - \text{정답값})^2$$

손실함수가 가장 적은 예측 함수가 가장 좋은 함수

# 최적의 가중치를 찾는 방법

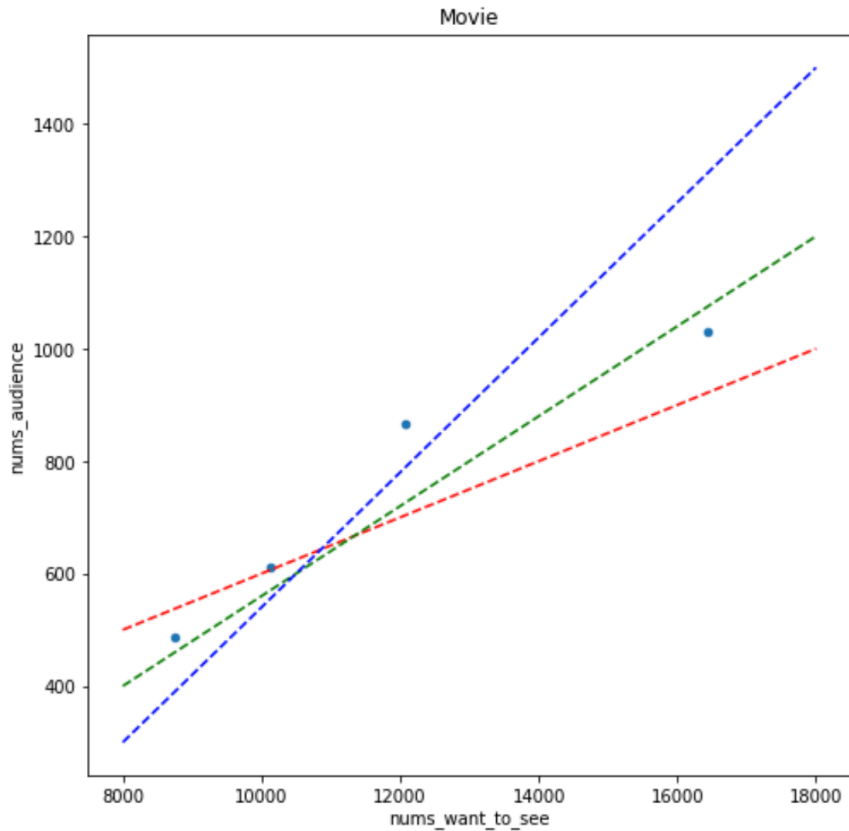
예측값과 정답값의 차이를 결정짓는 요인 : **가중치**



$$Y = aX + b$$

# 최적의 가중치를 찾는 방법

예측값과 정답값의 차이를 결정짓는 요인 : **가중치**



$$Y = aX + b$$

우리가 알고 싶은 것 :

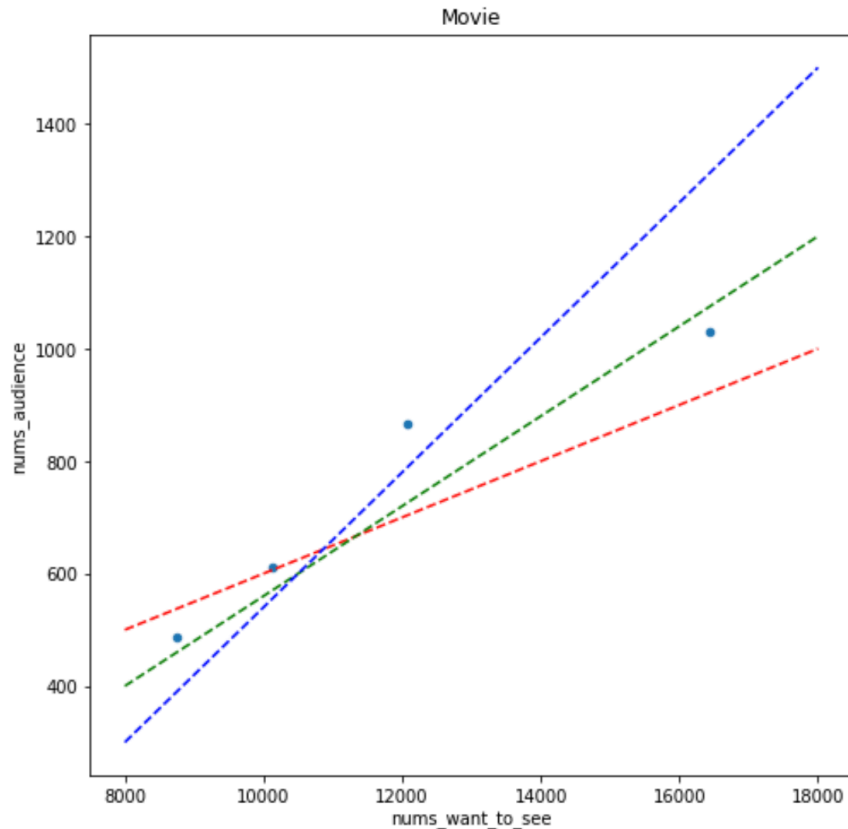
손실함수의 값이 가장 작게 만드는  $a, b$ 의 쌍

$$\underset{a,b}{\text{minimize}} \text{ Loss}(a,b)$$



# 최적의 가중치를 찾는 방법

예측값과 정답값의 차이를 결정짓는 요인 : **가중치**

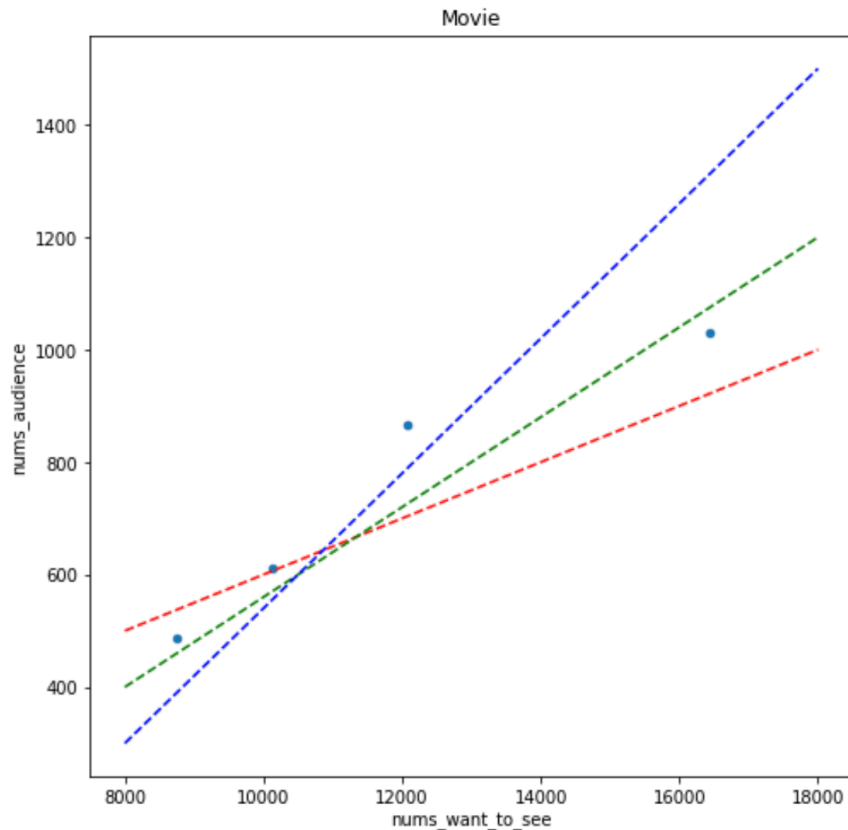


$$Y = aX + b$$

가중치의 조합 별로 손실함수를 각각 계산하자  
-> Grid Search

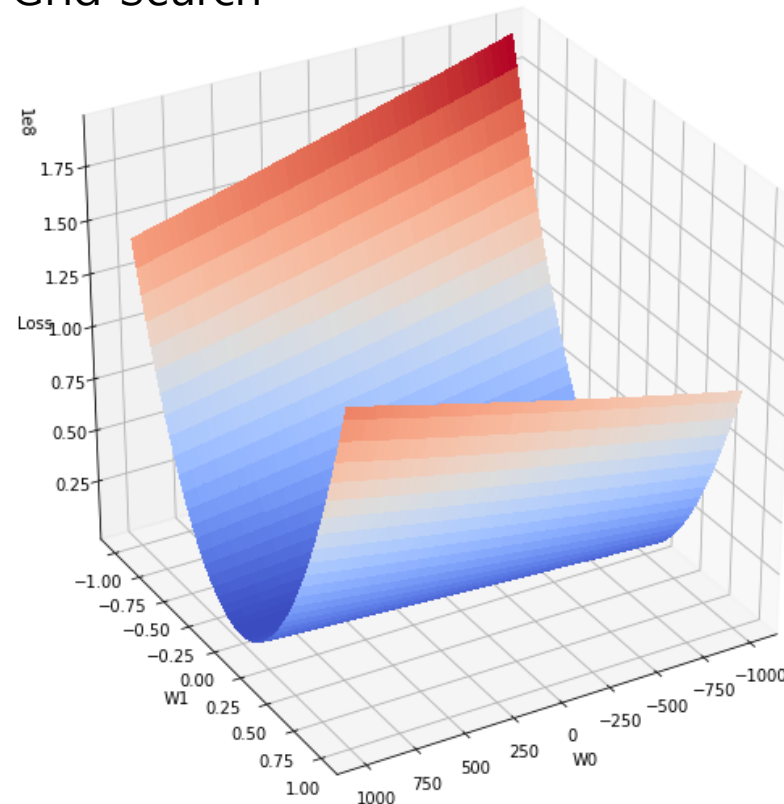
# 최적의 가중치를 찾는 방법 (1): Grid Search

예측값과 정답값의 차이를 결정짓는 요인 : **가중치**



$$Y = \mathbf{a}X + \mathbf{b}$$

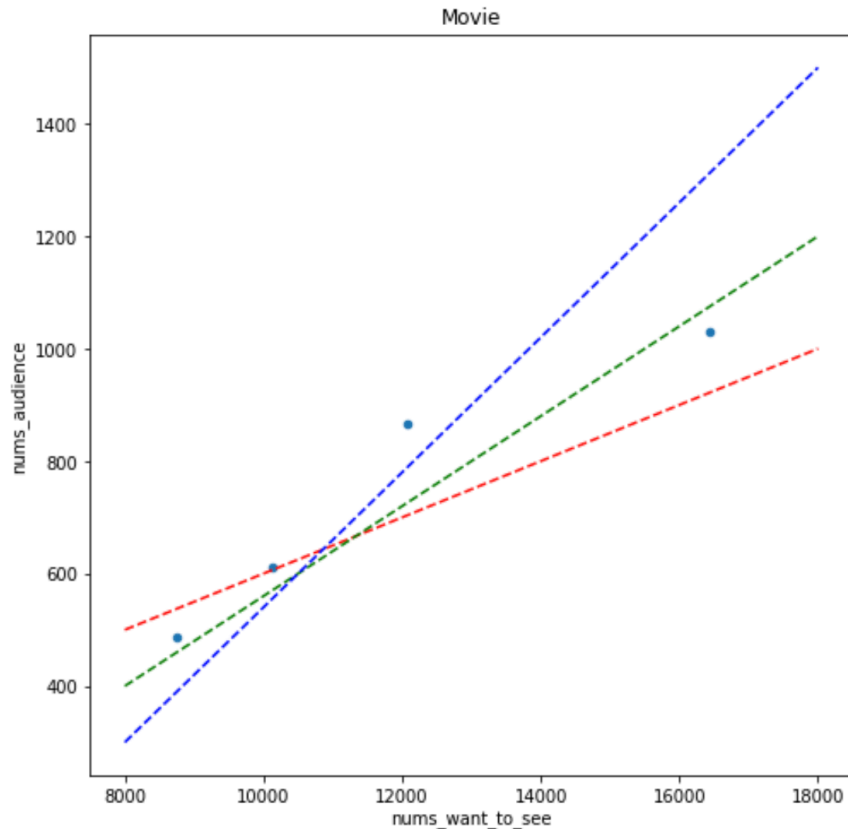
가중치의 조합 별로 손실함수를 각각 계산하자  
-> Grid Search



a : (-1,1) 사이 1000가지 수  
b : (-1000,1000) 사이 1000가지 수  
----  
<- 총 100만번의 연산

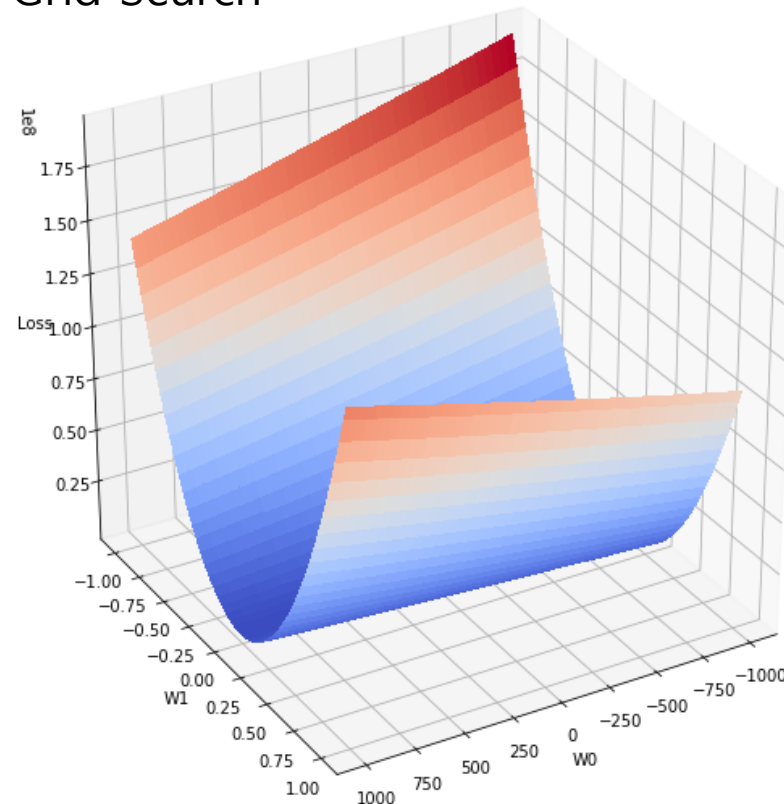
# 최적의 가중치를 찾는 방법 (1): Grid Search

예측값과 정답값의 차이를 결정짓는 요인 : **가중치**



$$Y = \mathbf{a}X + \mathbf{b}$$

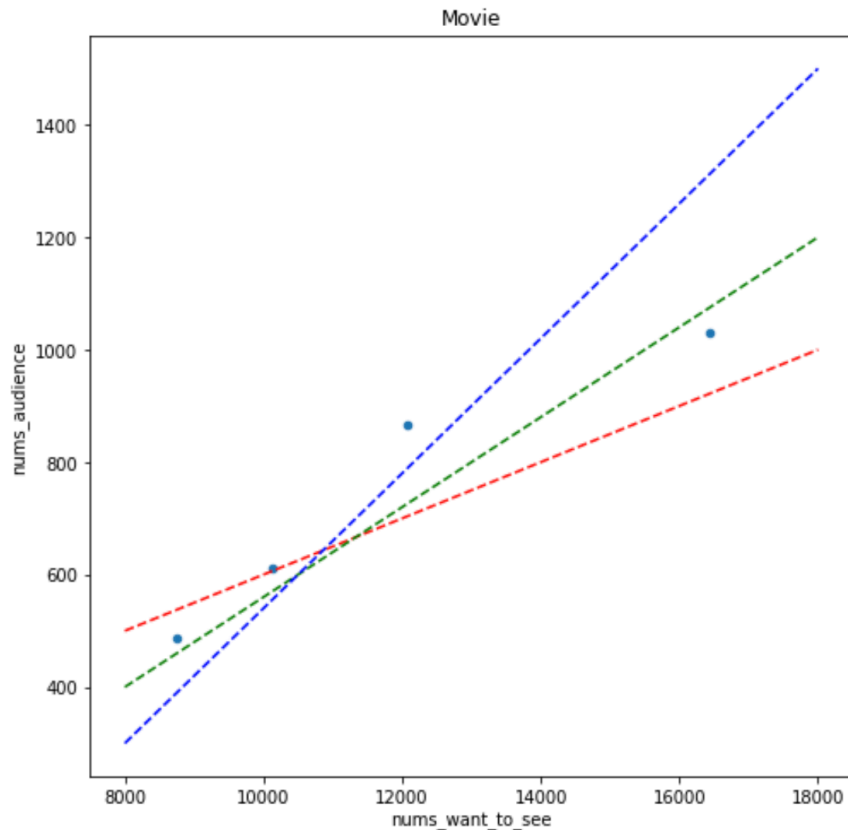
가중치의 조합 별로 손실함수를 각각 계산하자  
-> Grid Search



a : (-1,1) 사이 1000가지 수  
b : (-1000,1000) 사이 1000가지 수  
----  
<- 총 100만번의 연산

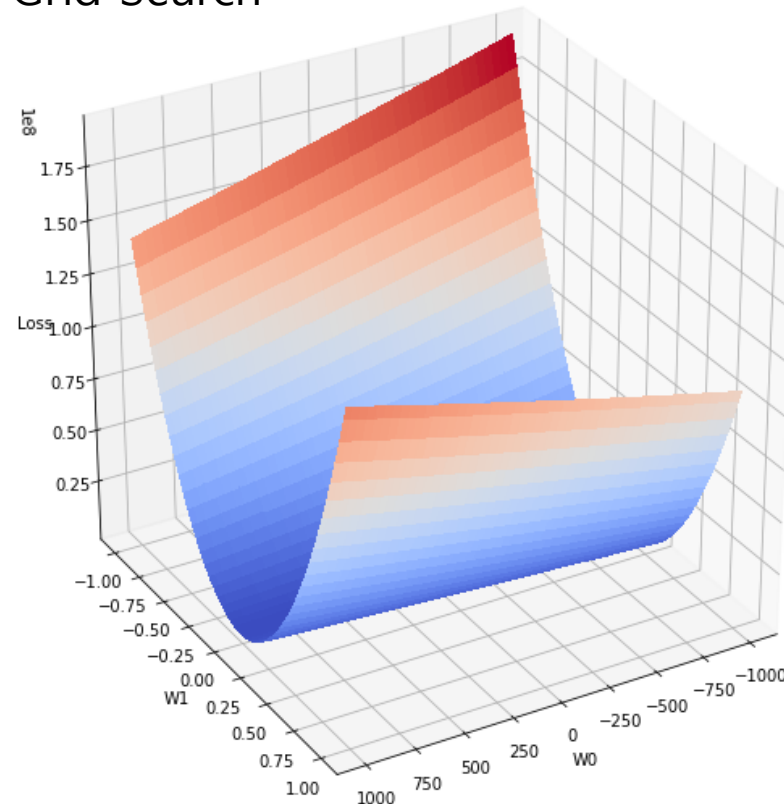
# 최적의 가중치를 찾는 방법 (1): Grid Search

예측값과 정답값의 차이를 결정짓는 요인 : **가중치**



$$Y = \mathbf{a}X + b$$

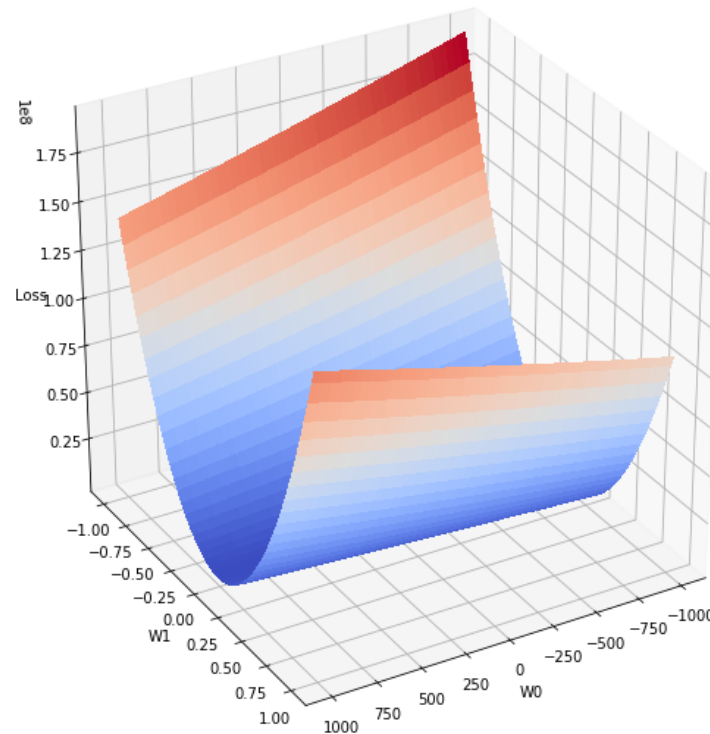
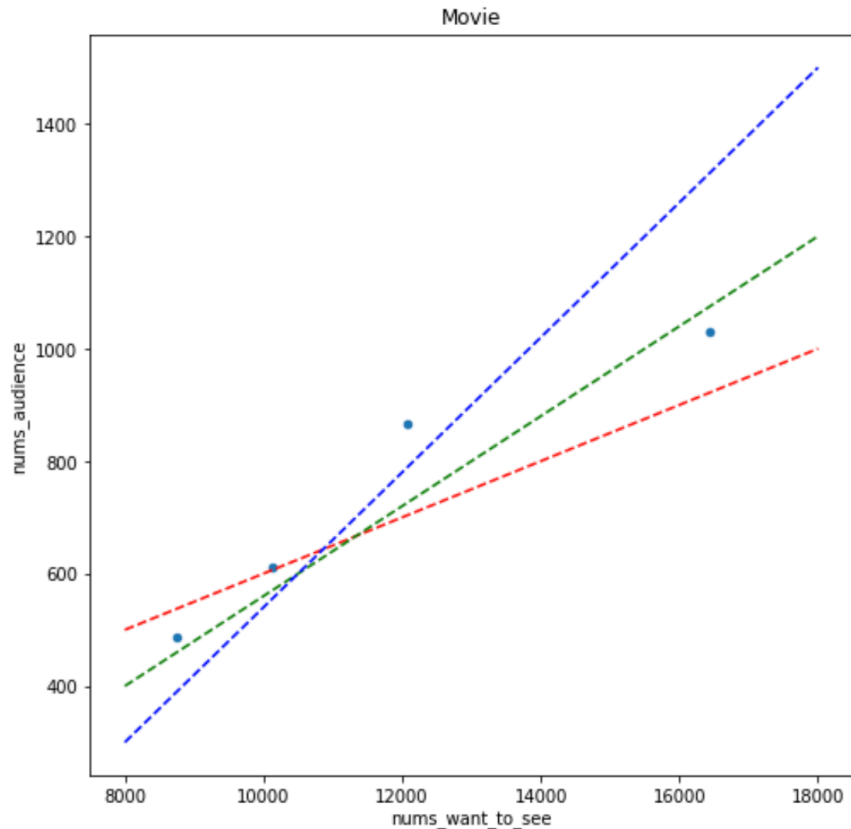
가중치의 조합 별로 손실함수를 각각 계산하자  
-> Grid Search



가중치의 조합이 많아지면?  
-> 연산량이 지나치게 많아짐

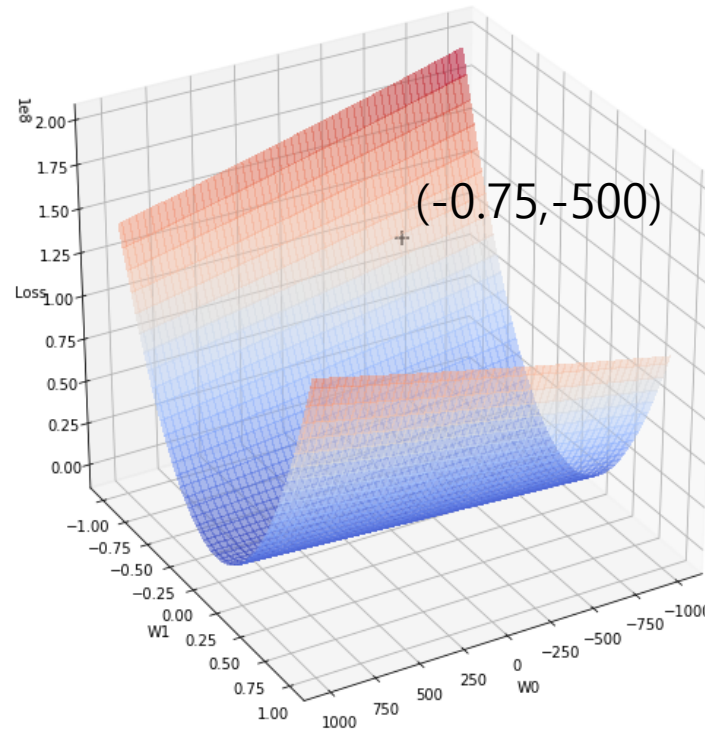
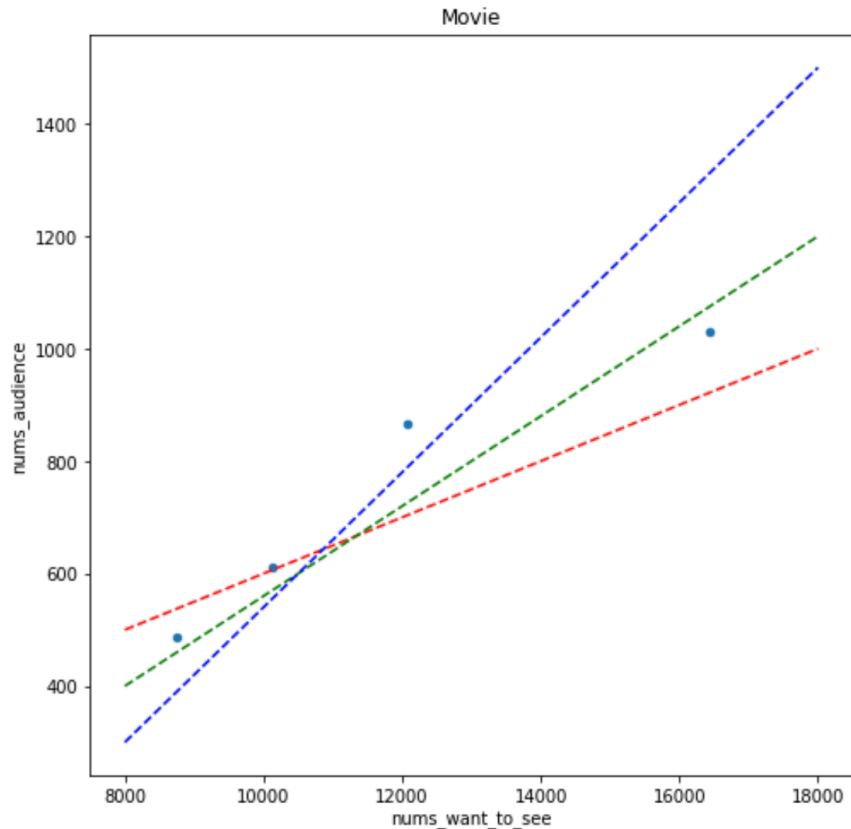
# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, **일부 조합만 탐색해서** 찾자



# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자

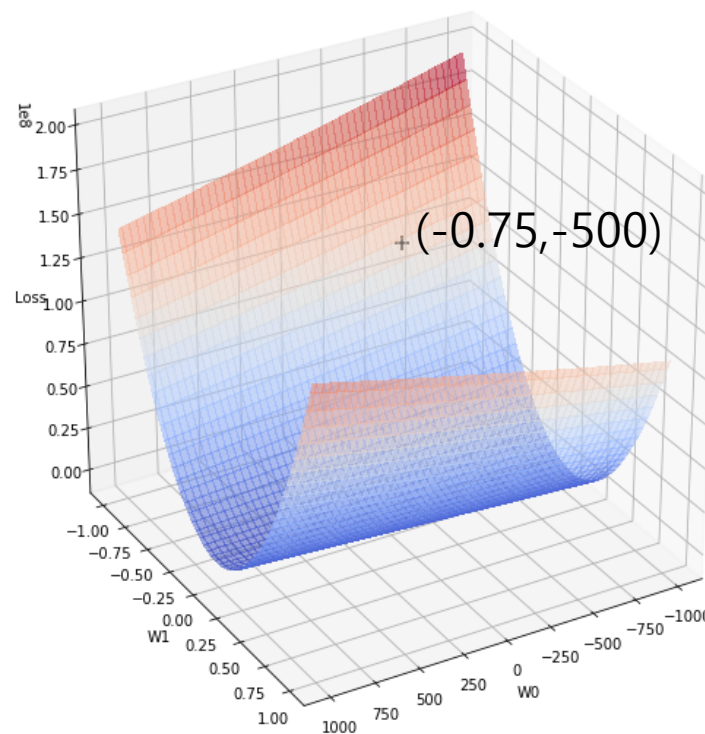
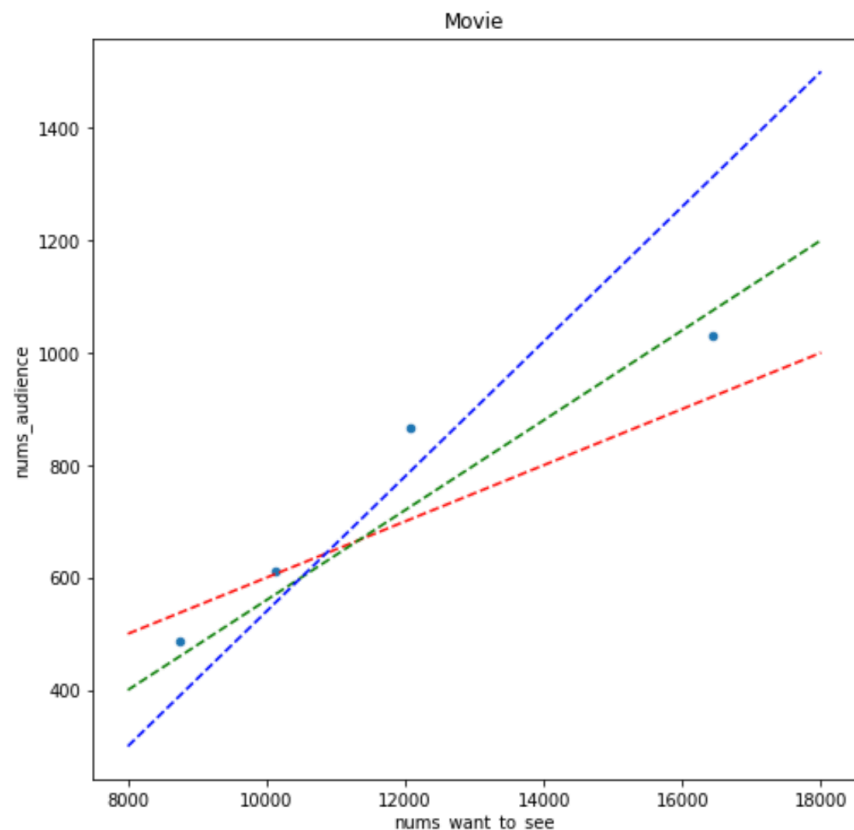


1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

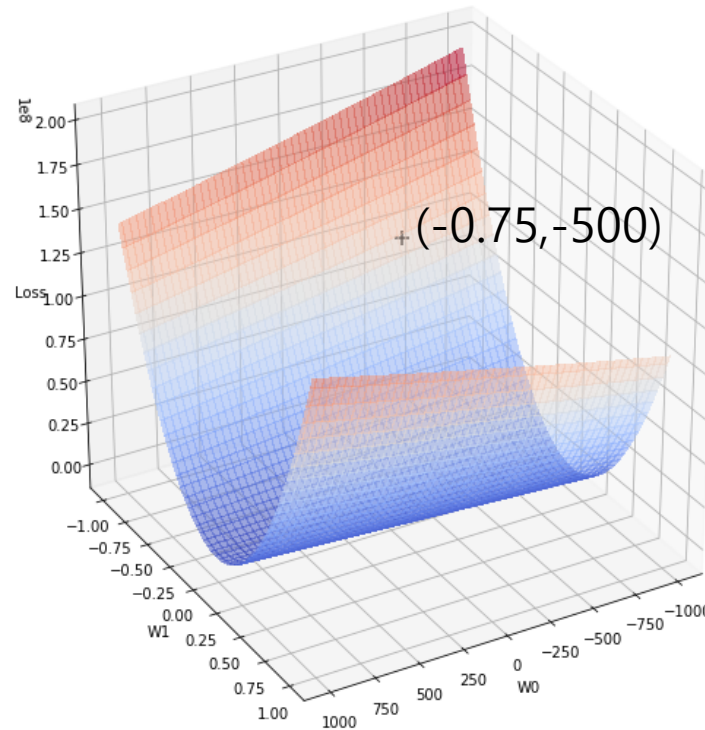
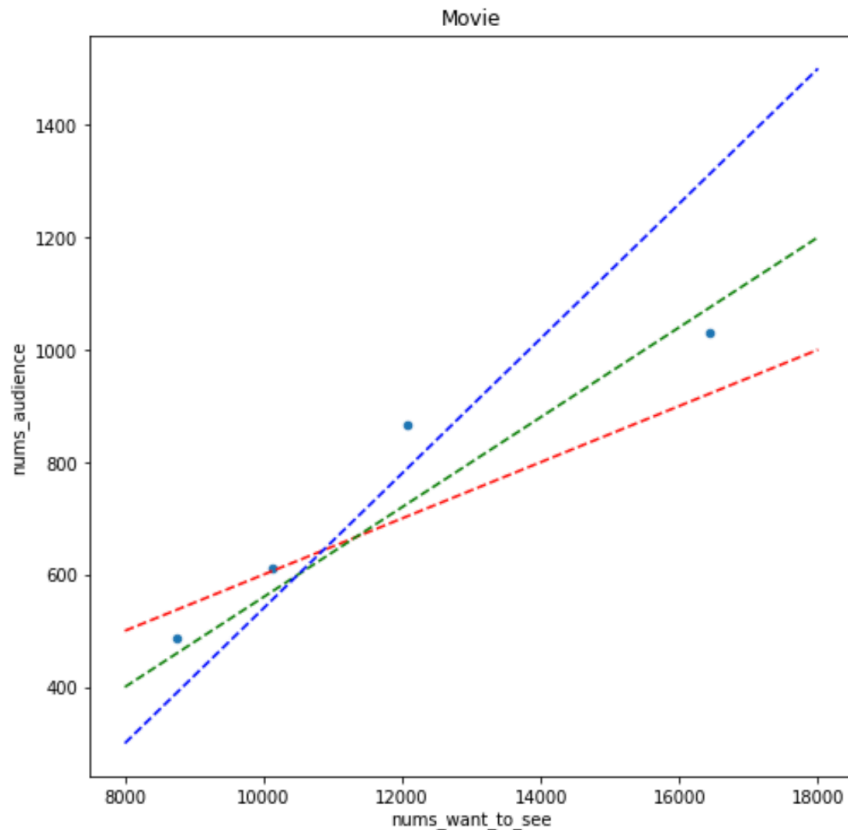
우리가 알 수 있는 것

1. 해당 가중치 조합의 손실함수 값

	보고싶어요 수	총 관객 수 (만명)	예측
마션	8759	487	-7069.25
킹스맨	10132	612	-8099.0
캡틴아메리카	12078	866	-9558.5
인터스텔라	16430	1030	-12822.5

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

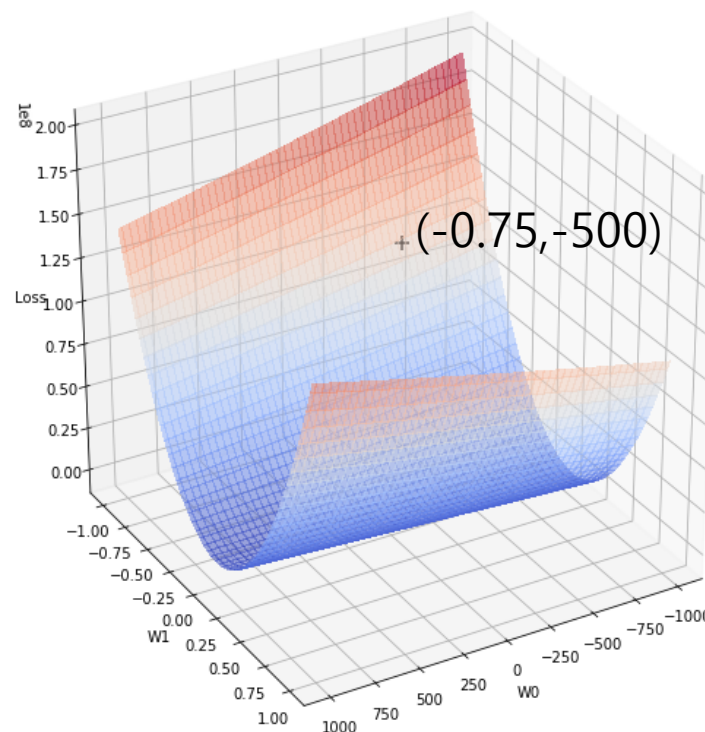
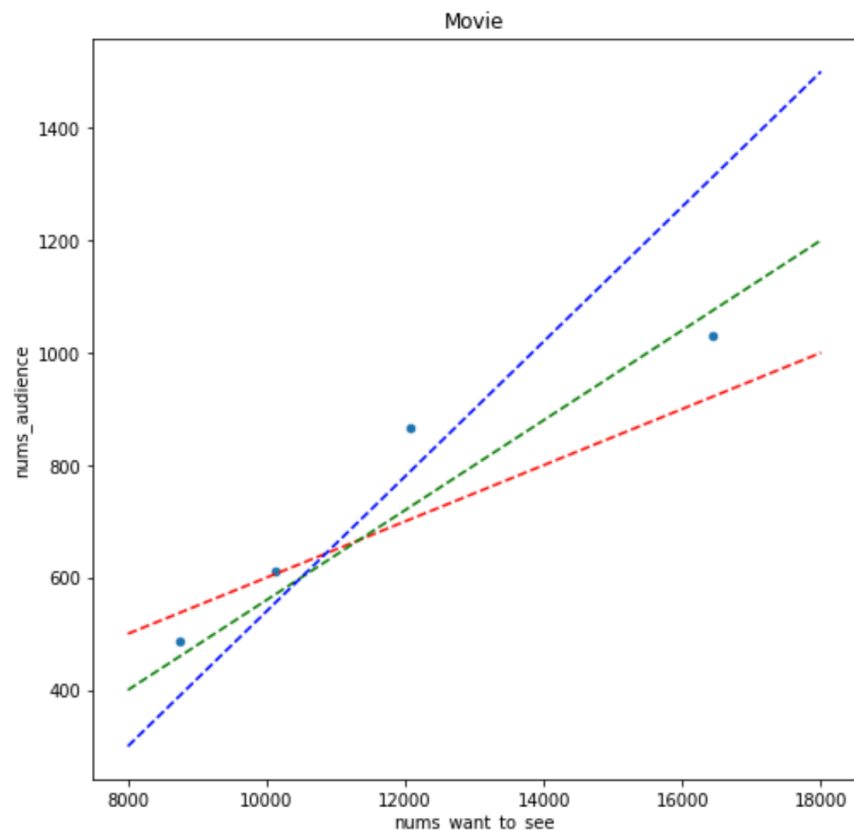
1. 해당 가중치 조합의 손실함수 값

$$Loss = (\text{예측값} - \text{정답값})^2$$



# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, **일부 조합만 탐색**해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

1. 해당 가중치 조합의 손실함수 값

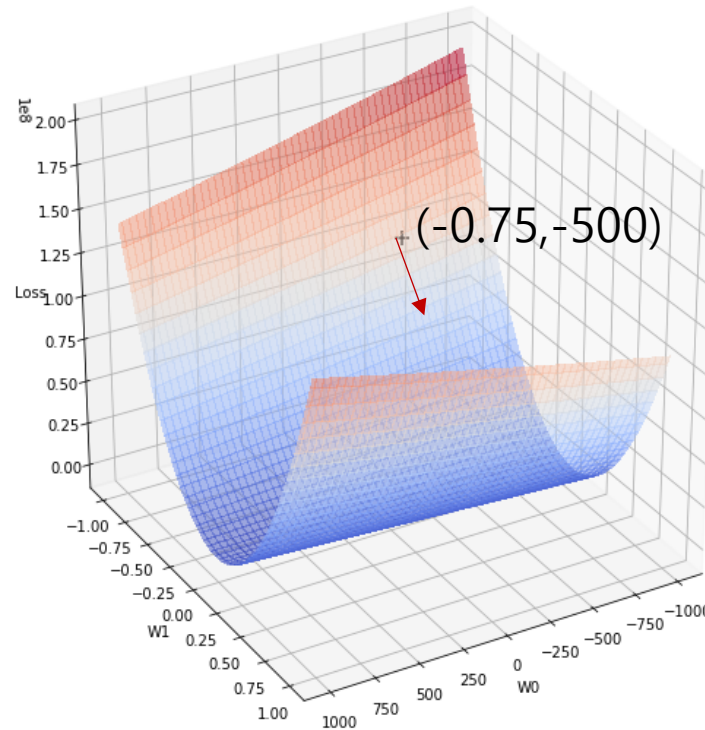
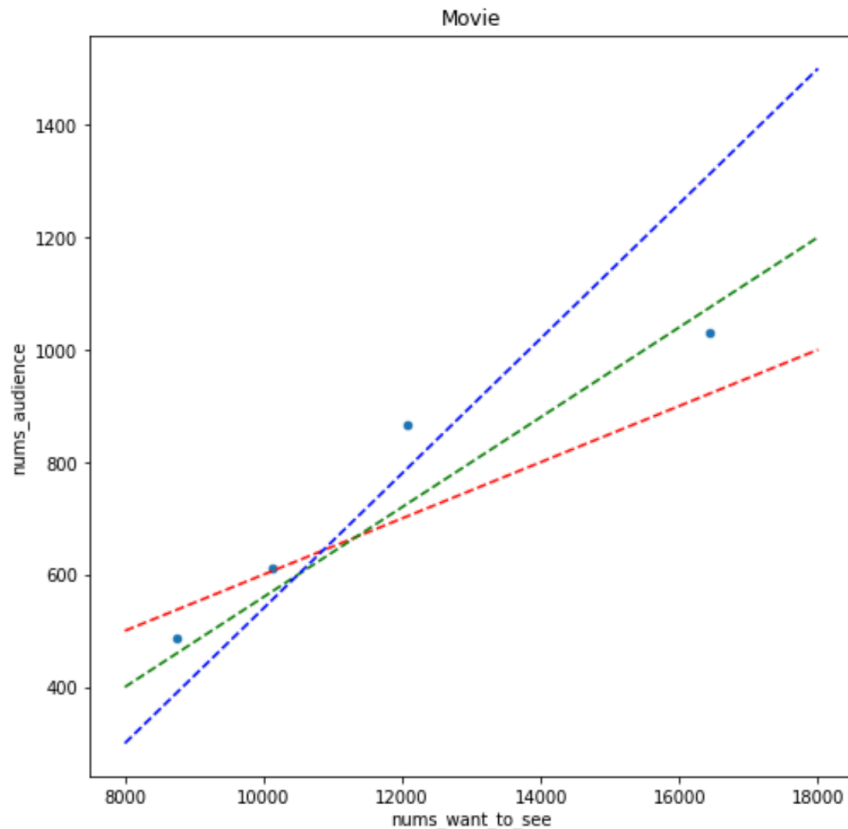
$$Loss = (\text{예측값} - \text{정답값})^2$$

핵심 아이디어

손실함수가 줄어드는 방향으로  
탐색하자!

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

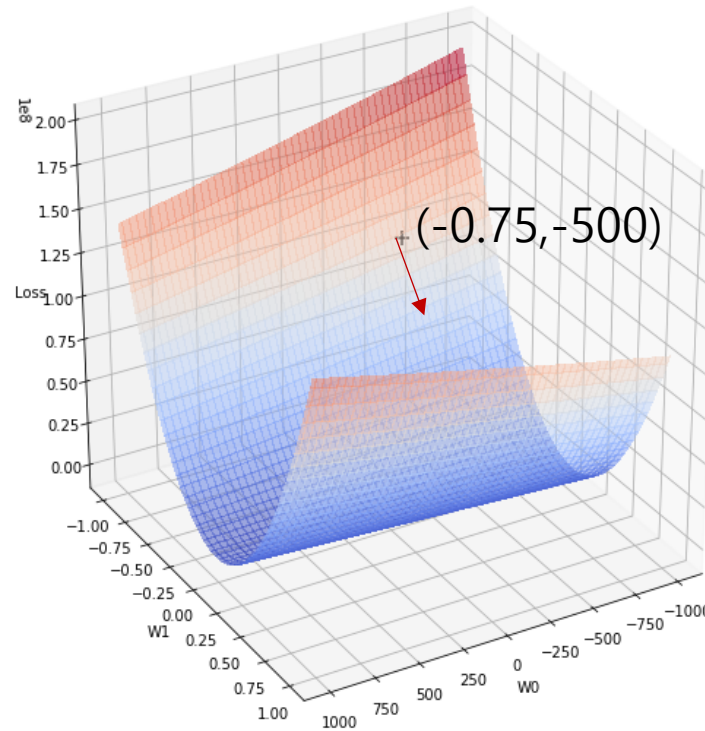
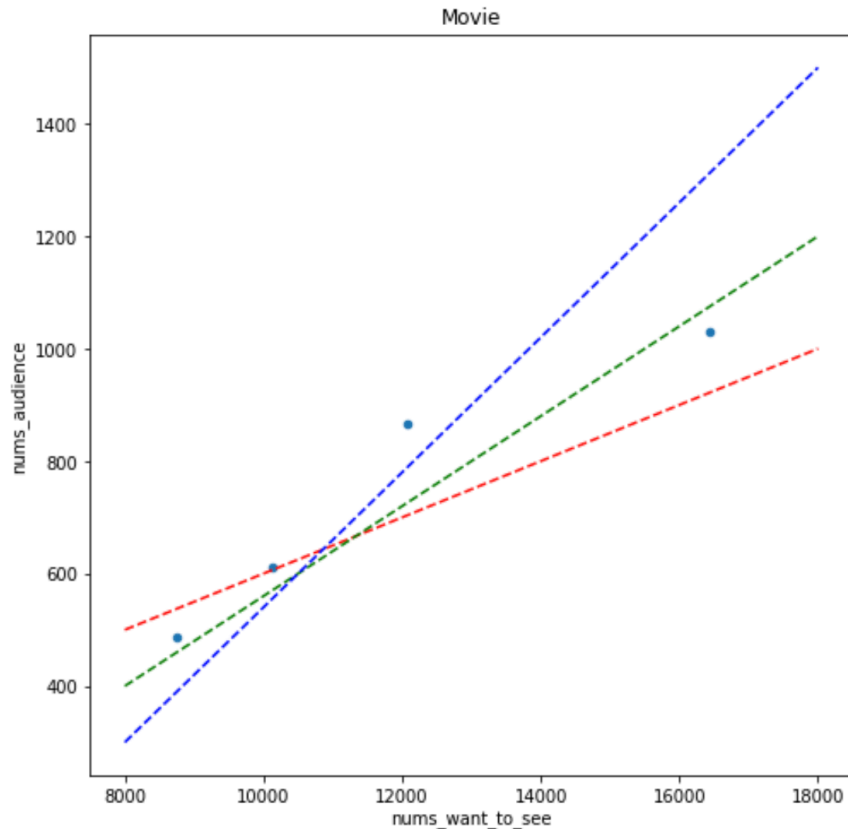
1. 해당 가중치 조합의 손실함수 값

$$Loss = (\text{예측값} - \text{정답값})^2$$

2. 해당 가중치 조합에서의 기울기

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

1. 해당 가중치 조합의 손실함수 값

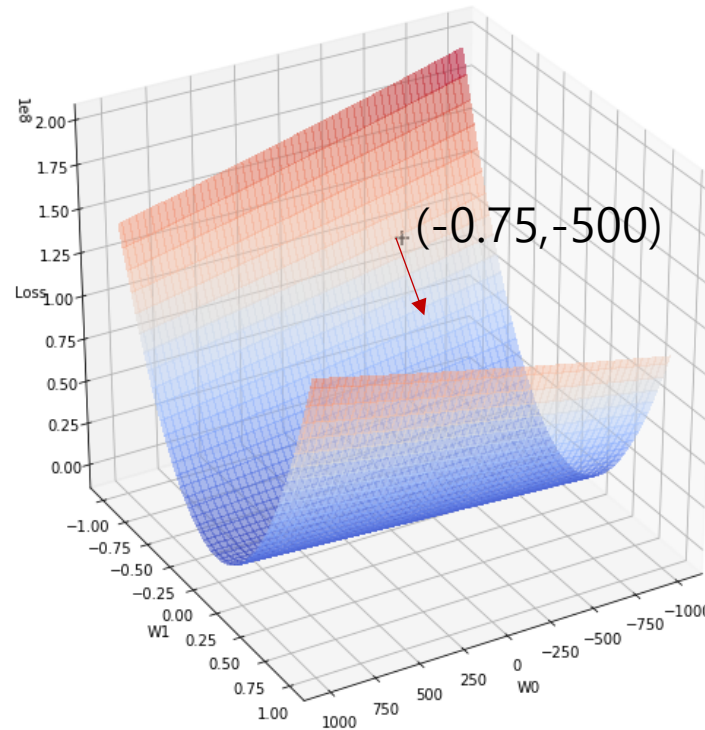
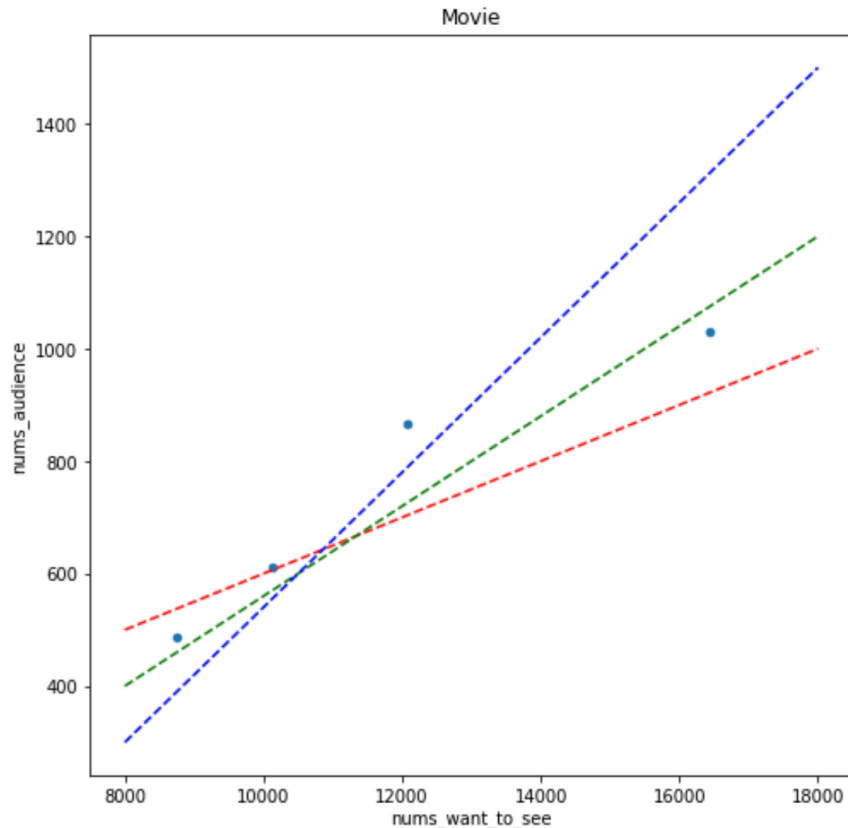
$$Loss = (\text{예측값} - \text{정답값})^2$$

2. 해당 가중치 조합에서의 기울기

가중치가 변했을 때 손실함수가  
줄어드는 방향

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

1. 해당 가중치 조합의 손실함수 값

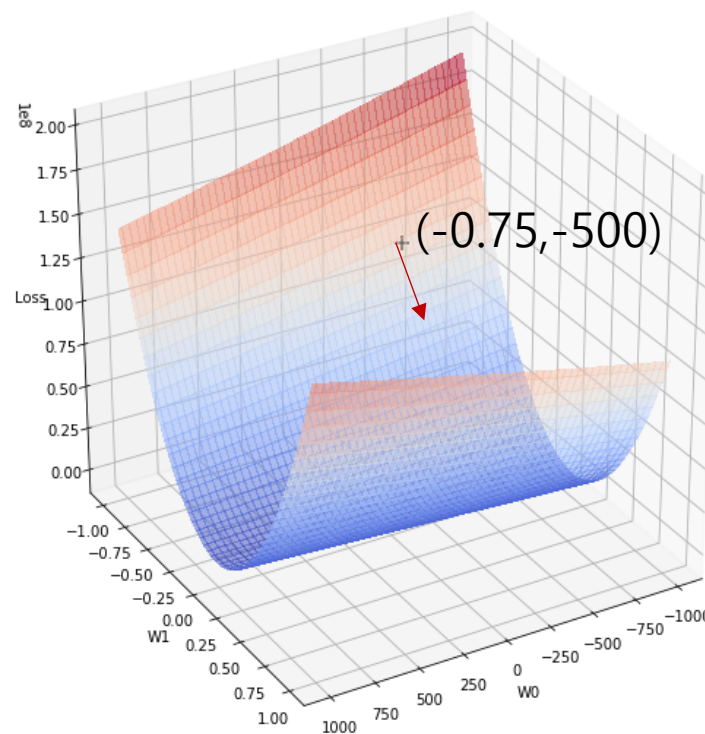
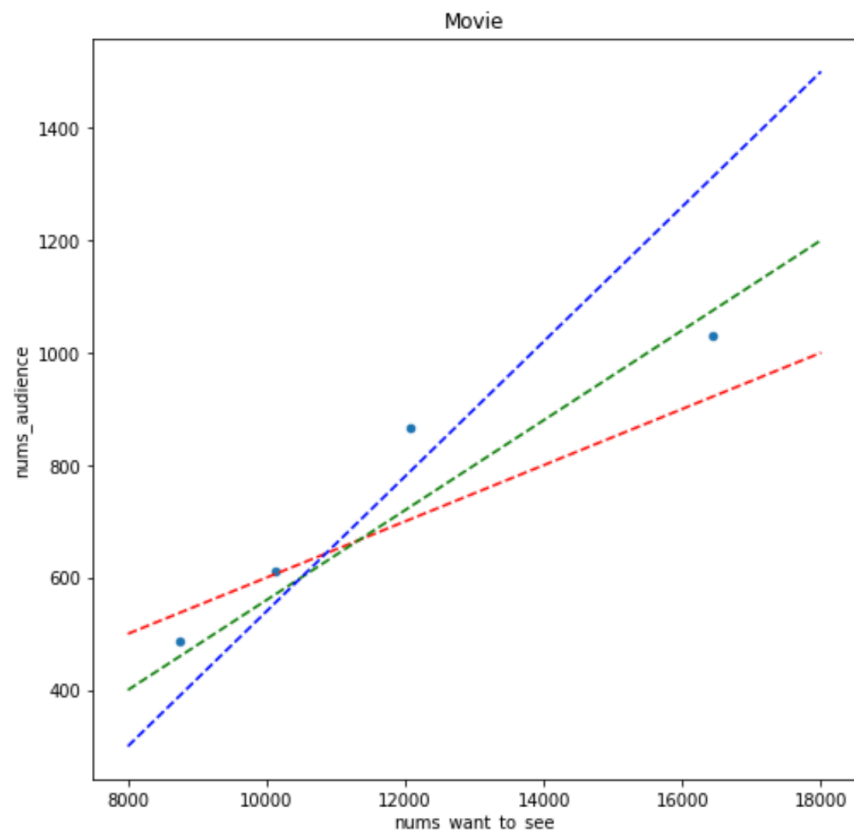
$$Loss = (\text{예측값} - \text{정답값})^2$$

2. 해당 가중치 조합에서의 기울기

손실함수(Loss)로 유도할 수 있음

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



1. 무작위로 가중치 조합 중에서 하나 선택

$$Y = aX + b$$

우리가 알 수 있는 것

1. 해당 가중치 조합의 손실함수 값

$$Loss = (\text{예측값} - \text{정답값})^2$$

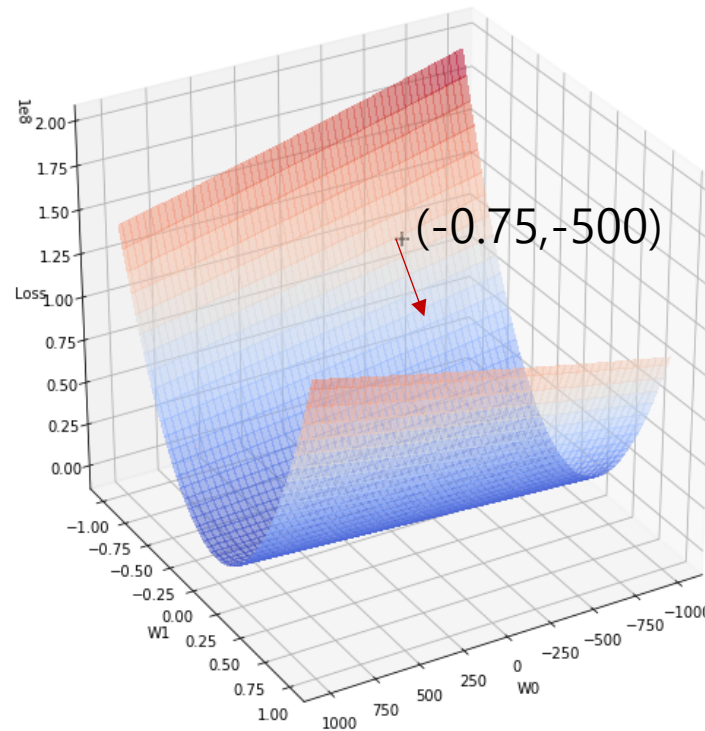
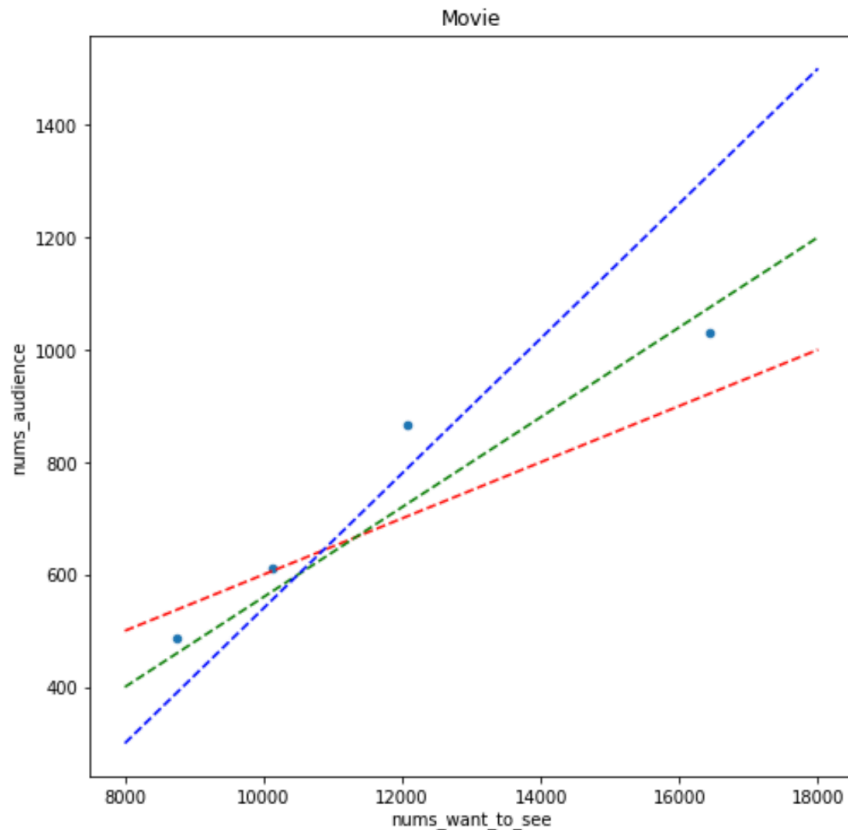
2. 해당 가중치 조합에서의 기울기

$$\frac{\partial Loss}{\partial a} : a \text{의 변화량에 따른 손실함수의 변화량}$$

$$\frac{\partial Loss}{\partial b} : b \text{의 변화량에 따른 손실함수의 변화량}$$

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

모든 조합을 탐색하지 말고, 일부 조합만 탐색해서 찾자



2. 손실함수가 줄어드는 방향으로  
가중치 갱신

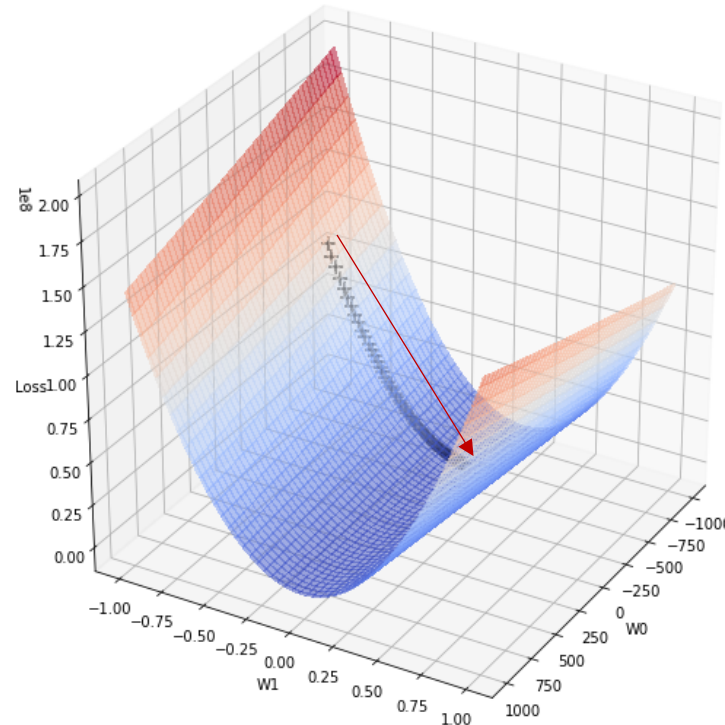
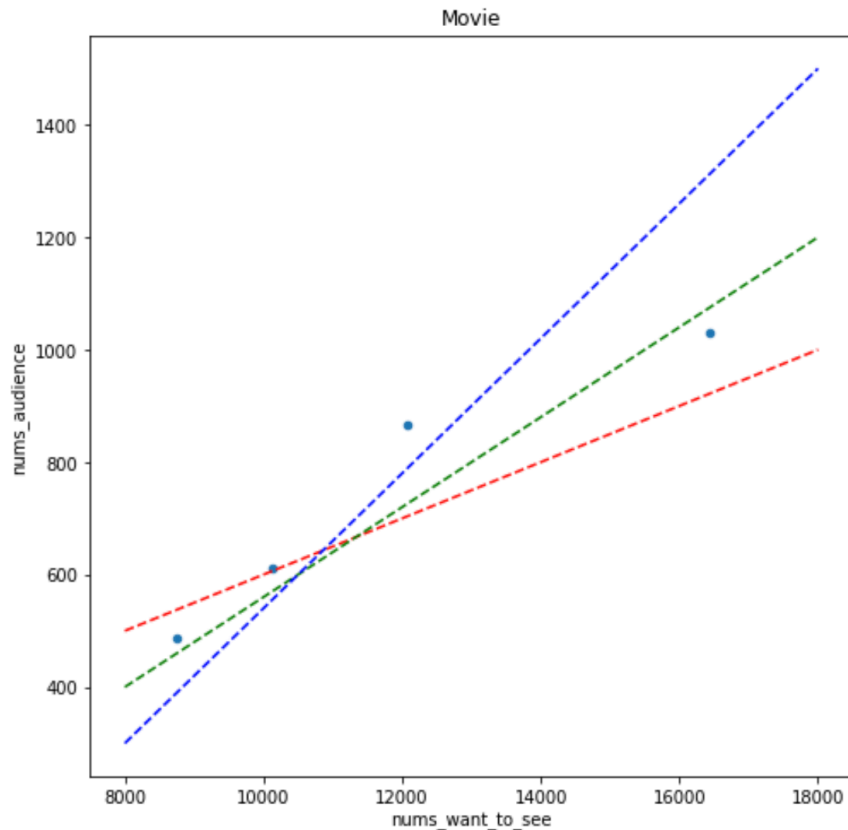
$$Y = aX + b$$

$$a := a - \alpha \frac{\partial \text{Loss}}{\partial a}$$

$$b := b - \alpha \frac{\partial \text{Loss}}{\partial b}$$

# 최적의 가중치를 찾는 방법 (2): Gradient Descent

경사하강법 : 손실함수가 줄어드는 방향으로 가중치 조합을 갱신하는 과정



3. 손실함수의 값이 충분히 작아질 때까지 반복

$$Y = aX + b$$

$$a := a - \alpha \frac{\partial \text{Loss}}{\partial a}$$

$$b := b - \alpha \frac{\partial \text{Loss}}{\partial b}$$