4. 선형 회귀와 로지스틱 회귀

목 차

- 선형 회귀
- 로지스틱 회귀

선형 회귀 (Linear Regression)

날씨와 아이스크림

- 문제 상황
  - 여름철에 아이스크림 가게에서 일하는데, 날씨가 더울 때 아이스크림이 더 많이 팔리는 것을 알게 됨
  - 온도와 아이스크림 판매량 간의 관계를 이해하고, 내일의 온도를 기준으로 판매량을 예측

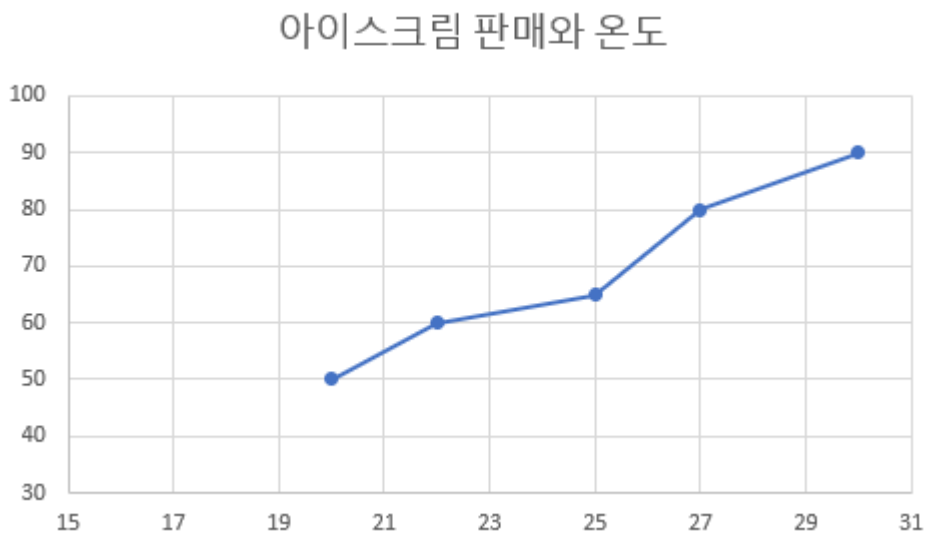
- 데이터

온도 (℃)	아이스크림 판매량 (개)
20	50
22	60
25	65
27	80
30	90

날씨와 아이스크림

- 목표
  - 온도에 따른 아이스크림 판매량을 예측할 수 있는 선형 방정식 찾기

- 1단계 : 데이터 시각화
  - 데이터를 그래프로 표현
  - 온도와 판매량 사이에 대략적인 직선관계가 있는 것을 확인



날씨와 아이스크림

- 2단계 : 선형 방정식 구하기
  - y = mx + b 형태의 방정식 구하기
    - y: 판매량
    - x: 온도
    - m: 기울기 (온도가 1도 증가할 때 판매량의 증가치)
    - b: y 절편 (온도가 0도일때의 판매량)

- 3단계 : 결과
  - y = 5x - 50 으로 가정
    - 온도가 1도 상승할 때마다 아이스크림이 5개씩 더 팔림
    - 온도가 0도일 때 아이스크림은 -50개 판매. (말이 안됨)

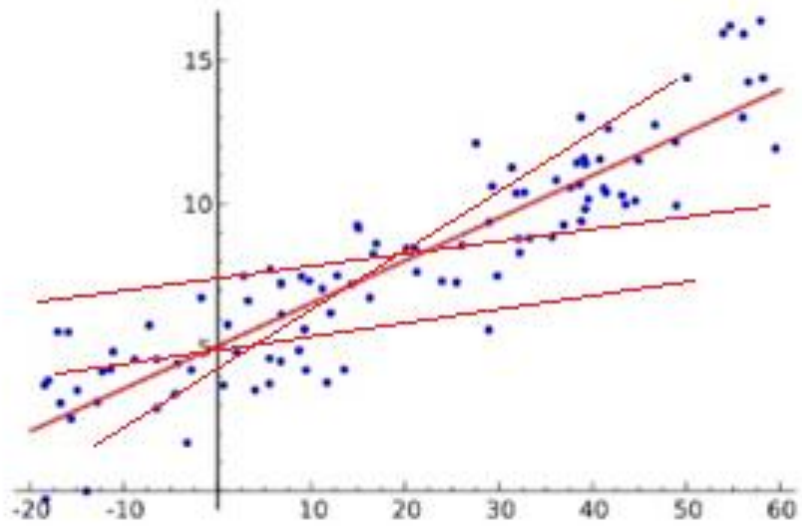
날씨와 아이스크림

- 4단계 : 예측
  - 내일 온도가 28도라 예상
  - y = 5 * 28 -50 = 140 - 50 = 90 (개) 팔릴 것으로 예상

- 선형 회귀
  - 데이터간의 관계를 이해하고 미래를 예측하는데 도움을 받을 수 있다

선형 회귀란?

- 선형 회귀 (Linear Regression)
  - 두 변수 사이의 관계를 분석하고 이를 직선 (선형 방정식) 으로 표현하는 통계 기법
  - 독립 변수와 종속 변수 간의 관계를 모델링하여 독립 변수의 값이 주어졌을 때 종속 변수를 예측



$$\hat{y} = w \times x + b$$

가중치 편향

선형 회귀의 기본 정의

- 독립 변수 (x) : 예측에 사용하는 변수, 설명 변수라고 함
- 종속 변수 (y) : 독립 변수에 의해 설명되거나 예측되는 변수
- 선형 방정식 : 일차 방정식 형태
  - y = mx + b
    - y는 종속 변수
    - x는 독립 변수
    - m 은 기울기(슬로프) : 독립변수가 1 단위 증가할 때 종속변수의 증가량
    - b는 y 절편으로 x가 0 일때 y 값

선형 회귀

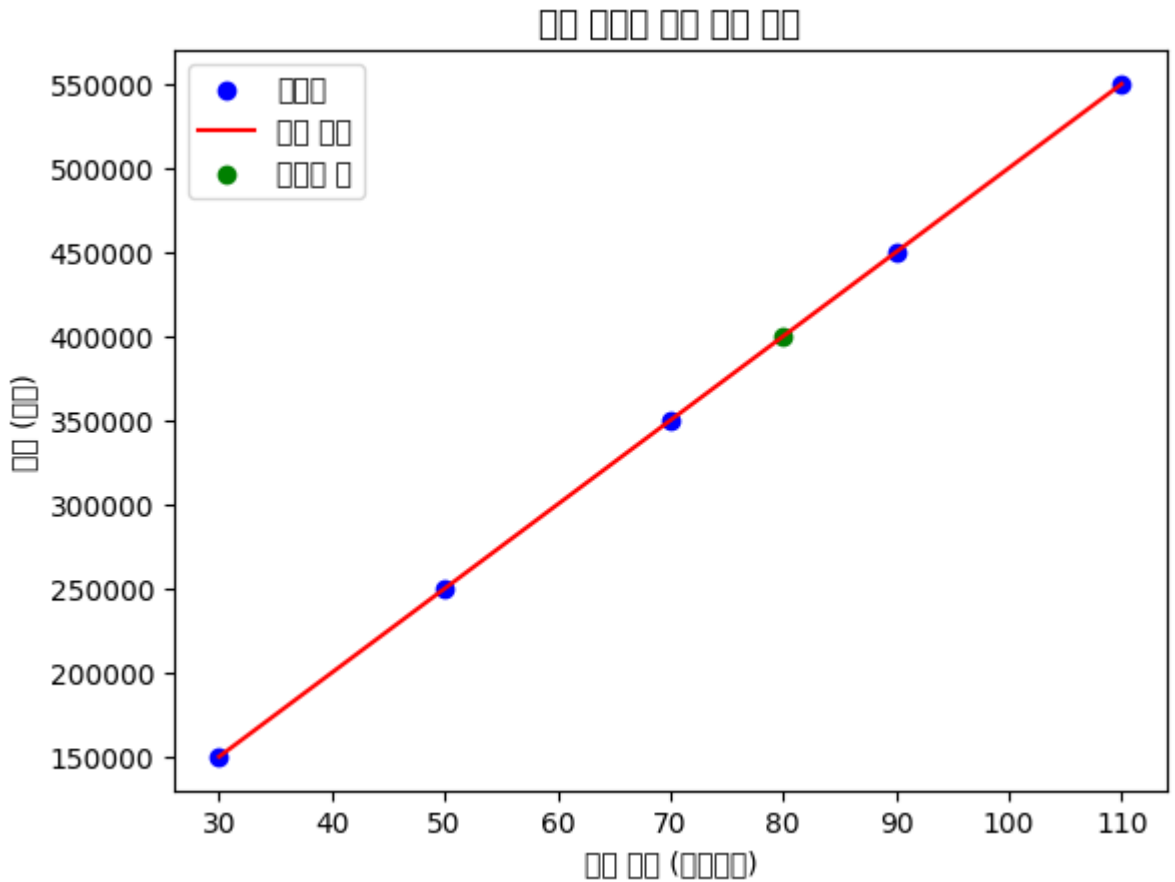
- 목적
  - 주어진 데이터를 가장 잘 설명하는 직선을 찾는 것

- 예시
  - 부동산 가격 예측
    - 주택의 크기와 가격 사이의 관계를 분석
  - 마케팅 효과 분석
    - 광고비와 제품 판매량 사이의 관계를 분석
  - 학생 성적 예측
    - 학생의 공부 시간과 시험 성적 간의 관계를 분석

선형 회귀 코드 사례

```python
import numpy as np
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# 1. 데이터 준비
X = np.array([[30], [50], [70], [90], [110]])  # 주택 크기 (평방미터)
y = np.array([150000, 250000, 350000, 450000, 550000])  # 주택 가격 (달러)

# 2. 선형 회귀 모델 생성
model = LinearRegression()

# 3. 모델 학습
model.fit(X, y)

# 4. 예측
new_house_size = np.array([[80]])  # 새로운 주택 크기 (평방미터)
predicted_price = model.predict(new_house_size)

print(f"예측된 80평방미터 주택의 가격: ${predicted_price[0]:,.2f}")

# 5. 데이터 시각화
plt.scatter(X, y, color='blue', label='데이터')
plt.plot(X, model.predict(X), color='red', label='선형 회귀')
plt.scatter(new_house_size, predicted_price, color='green', label='예측된 값')
plt.xlabel('주택 크기 (평방미터)')
plt.ylabel('가격 (달러)')
plt.title('주택 크기에 따른 가격 예측')
plt.legend()
plt.show()
```

```
예측된 80평방미터 주택의 가격: $400,000.00
```



로지스틱 회귀

이메일 스팸 분류

- 로지스틱 회귀
  - 종속 변수가 범주형 (예 : 예/아니오, 참/거짓, 승리/패배) 일 때 사용됨
  - 출력값이 0과 1 사이이 확률로 변환
  - 결과적으로 특정 사건이 발생할 확률 예측이 가능

- 문제 상황
  - 받은 이메일 중에서 스팸 메일을 거르고자 함
  - 이메일의 다양한 특징을 바탕으로 스팸/햄 여부를 예측

이메일 스팸 분류

- 데이터
  - 스팸여부 판단에 사용되는 특징
    - 이메일 본문에 “할인 " 이라는 단어가 포함되어 있는지 여부
    - 발신자가 우리 이메일 주소록에 있는지 여부

이메일 본문에 "할인" 포함	발신자가 주소록에 있음	이메일이 스팸인지 여부(타겟)
1	0	1
0	1	0
1	0	1
1	1	0
0	0	0

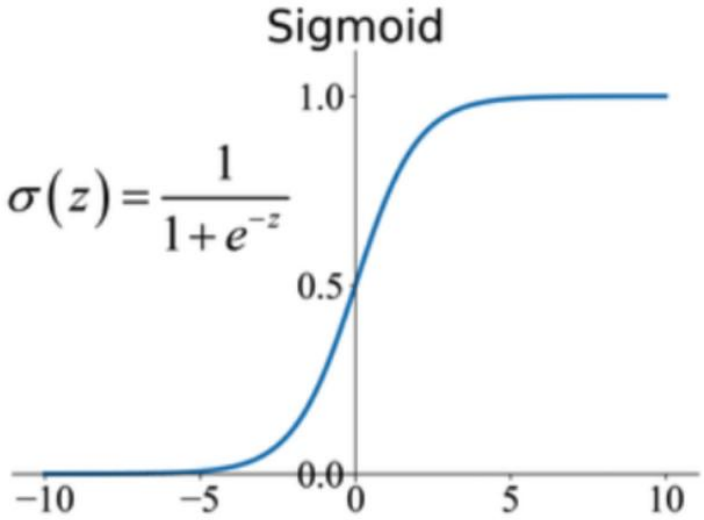
- 목표
  - 새로운 이메일이 스팸일 확률을 예측

로지스틱 회귀의 기본 개념

- 로지스틱 함수 (시그모이드 함수)
  - 선형 회귀 결과를 확률로 제한

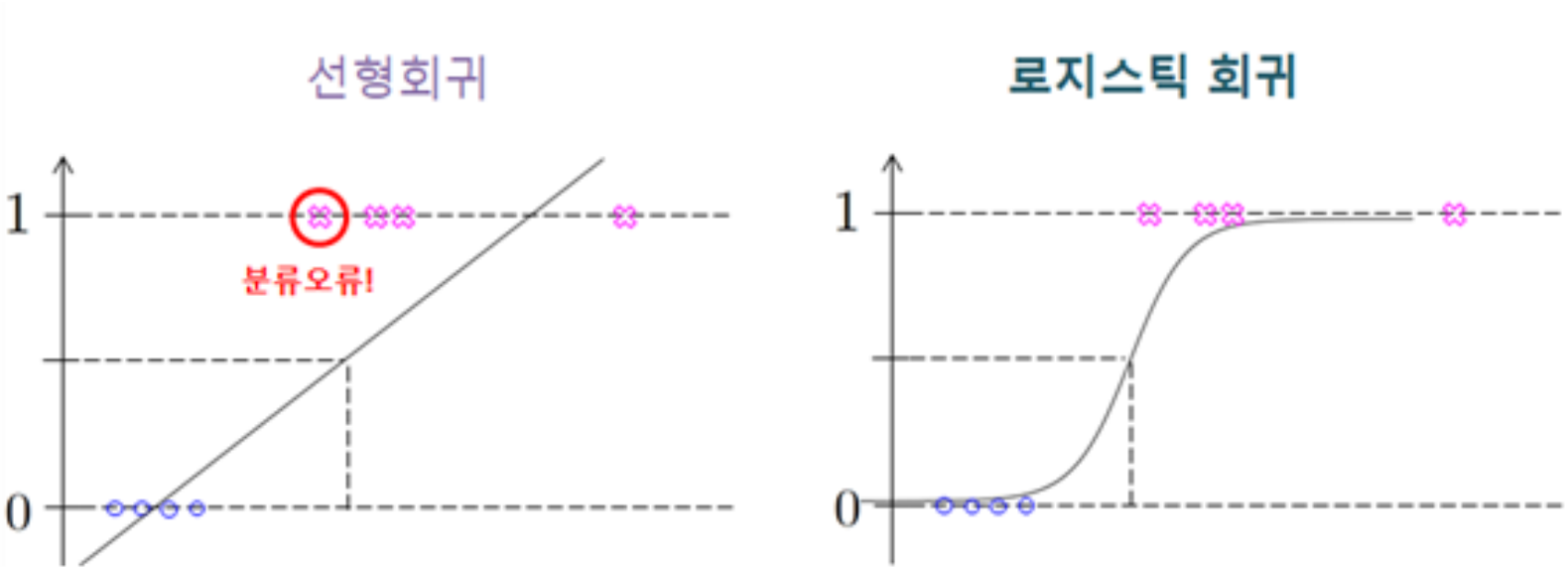
$$P(y=1|x) = \frac{1}{1+e^{-(mx+b)}}$$

    - P(y=1|x): 주어진 입력 x에 대해 y=1일 확률, 특정 사건이 발생할 확률
    - m, b: 모델의 파라미터로 기울기와 절편에 해당
    - E: 자연상수 (약 2.718)
    - mx + b: 독립 변수와 모델 파라미터를 포함한 선형 식



로지스틱 회귀의 목적

- 로지스틱 회귀의 주된 목적
  - 주어진 입력 변수(독립 변수)들이 특정 사건(종속 변수) 발생에 영향을 미치는지를 모델링
  - 해당 사건이 발생할 확률을 예측

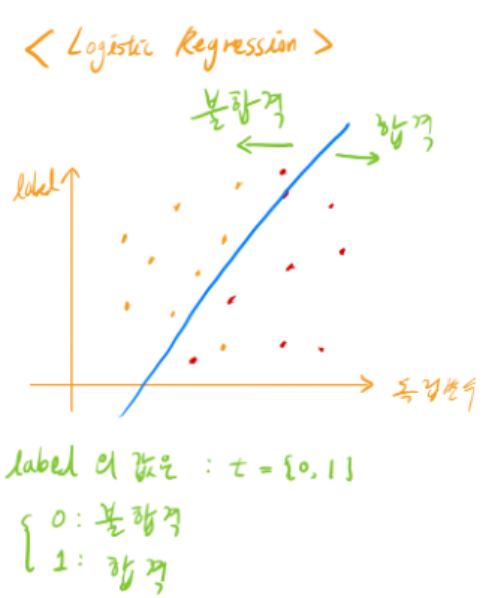


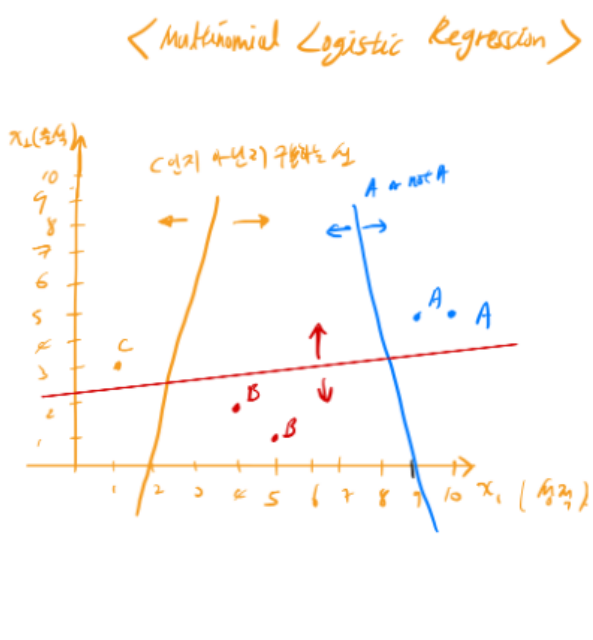
로지스틱 회귀의 과정

- 데이터 수집
  - 예측하려는 결과(종속 변수)와 그에 영향을 미칠 수 있는 특징(독립 변수)을 포함한 데이터 수집
- 데이터 전처리
  - 로지스틱 회귀에 적합한 형태로 전처리
  - 범주형 데이터 -> 숫자형 데이터 (원-핫 인코딩)
- 모델 학습
  - 로지스틱 회귀 모델을 데이터에 피팅
  - 최적의 기울기와 절편 발견
- 예측
  - 새로운 데이터에 대해 예측을 수행
  - 입력된 변수값에 따라 특정 사건이 발생할 확률을 반환
- 모델 평가
  - 혼동 행렬(confusion matrix), 정확도(accuracy), 정밀도(precision), 재현율(recall), F1 점수 등 사용

로지스틱 회귀의 종류

- 이진 로지스틱 회귀 (Binary Logistic Regression)
  - 결과 변수가 두 개의 범주일 때 사용
  - 질병에 걸림(1) vs 걸리지 않음 (0)
- 다항 로지스틱 회귀 (Multinomial Logistic Regression)
  - 결과 변수가 세 개 이상의 범주일 때 사용
  - 질병의 중증도 (경증, 중증, 위중)
- 순서형 로지스틱 회귀 (Ordical Logistic Regression)
  - 결과 변수가 순서가 있는 세 개 이상의 범주일 때 사용
  - 고객 만족도 (매우 불만족, 불만족, 만족, 매우 만족)





로지스틱 회귀의 가정

- 독립 변수와 종속 변수 간의 선형 관계
  - 독립 변수의 선형 조합으로 종속 변수를 예측
- 독립 변수 간의 다중공선성 없음
  - 독립 변수들이 상관관계가 거의 없음
- 관찰치의 독립성
  - 각 관측치가 서로 독립적

로지스틱 회귀의 실제 응용

- 의료
  - 환자의 특정 질병 발생 확률 예측
- 마케팅
  - 고객의 구매 가능성을 예측하고 타겟 광고 개발
- 금융
  - 고객의 대출 상환 가능성을 평가
- 사기 탐지
  - 트랜잭션이 사기일 가능성을 예측하여 사기 거래를 탐지

로지스틱 회귀 코드 사례

```python
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

# 가상 데이터 생성 (공부 시간, 수업 참석 횟수)
X = np.array([
    [10, 2],
    [15, 3],
    [10, 1],
    [25, 5],
    [30, 5],
    [35, 6],
    [20, 3],
    [20, 4],
    [5,  1],
    [15, 2]
])

# 결과: 1 = 합격, 0 = 불합격
y = np.array([0, 0, 0, 1, 1, 1, 1, 1, 0, 0])

```

```python
# 데이터를 훈련용과 테스트용으로 나눕니다.
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

```

```python
# 로지스틱 회귀 모델 생성 및 학습
model = LogisticRegression()
model.fit(X_train, y_train)

```

LogisticRegression
LogisticRegression()

로지스틱 회귀 코드 사례

```python
# 테스트 데이터를 사용하여 예측
y_pred = model.predict(X_test)

# 모델 평가
print(f"정확도: {accuracy_score(y_test, y_pred):.2f}")
print("혼동 행렬:")
print(confusion_matrix(y_test, y_pred))
print("분류 보고서:")
print(classification_report(y_test, y_pred))

```

```
정확도: 1.00
혼동 행렬:
[[2 0]
 [0 1]]
분류 보고서:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00         2
           1       1.00      1.00      1.00         1

    accuracy                           1.00         3
   macro avg       1.00      1.00      1.00         3
weighted avg       1.00      1.00      1.00         3
```

```python
# 새로운 데이터에 대한 합격 여부 예측
new_data = np.array([[18, 3], [25, 4], [8, 2]])
predictions = model.predict(new_data)

print(f"새로운 데이터 {new_data}에 대한 예측 결과: {predictions}")

```

```
새로운 데이터 [[18  3]
 [25  4]
 [ 8  2]]에 대한 예측 결과: [1 1 0]
```