

Q 1.

$$(1) \det \begin{bmatrix} a^1 & b^0 & c^1 \\ d^0 & e^2 & f^3 \\ g^0 & h^1 & i^3 \end{bmatrix}$$

$$\det(A) = a(ei - fh) - b(di - fg) + c(dh - eg) \\ = \underline{\underline{6}}$$

3x3 행렬의 determinant는 부피(volume)를 의미한다.

즉, 해당 행렬은 3차원 공간에서의 변환이며, 이 변환은 양의 부피를 가진다.

이 행렬은 양의 스케일 변환을 표현하며, 기하학적으로 부피를 확대하는 변환이라고 해석할 수 있다.

(2)

(1) 고유값(λ) 계산

$$\det(A - \lambda I) = 0 \\ = (1 - \lambda)(2 - \lambda)(3 - \lambda)$$

$$\therefore \lambda = 1, 2, 3$$

(2) 고유벡터(v) 계산

$$(A - \lambda I)v = 0$$

$$\bullet \lambda = 1 \text{ 일 때, } v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \bullet \lambda = 2 \text{ 일 때, } v_2 = \begin{bmatrix} -1 \\ 3 \\ 0 \end{bmatrix} \bullet \lambda = 3 \text{ 일 때, } v_3 = \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}$$

• 고유값(λ):

선형 변환을 통해 해당 변환에 대해 스케일링되는 크기를 나타냄.

주어진 행렬 A의 고유값은 1, 2, 3으로, 각각 x, y, z 축 방향으로의 스케일링 요소를 의미

• 고유벡터(v):

해당 선형 변환에서 크기만 변화시키고, 방향은 변화시키지 않는 방향을 나타냄.

주어진 행렬 A의 고유벡터는 v_1, v_2, v_3 으로, 각각 고유값에 대응하는 변환에 대해 방향이 변하지 않는 벡터를 의미

Q 2.

(1) 고유값 (λ) 구하기

$$\lambda_1 = 0, \lambda_2 = 1$$

(2) 고유벡터 (v) 구하기

$$\lambda_1 = 0 \text{ 인 경우, } v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\lambda_2 = 1 \text{ 인 경우, } v_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

(3) P 만들기

$$P = \begin{bmatrix} v_1 & v_2 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$$

(변환 행렬)

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

(4) 완성

$$P^{-1}BP = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

\underline{D}

Q 3.

- $P(\text{Spam})$: 메일 시스템에 도착하는 메일 중 스팸 메일의 비율 = 0.20
- $P(\text{'당첨'} | \text{Spam})$: 스팸 메일 중에서 '당첨'이라는 단어를 포함하는 비율 = 0.50
- $P(\text{'당첨'} | \text{Not Spam})$: 스팸이 아닌 정상 메일 중에서 '당첨'이라는 단어를 포함하는 비율 = 0.01

$$\begin{aligned} \bullet P(\text{'당첨'}) &= P(\text{'당첨'} | \text{Spam}) * P(\text{Spam}) \oplus P(\text{'당첨'} | \text{Not Spam}) * P(\text{Not Spam}) \\ &= (0.5 \times 0.2) \oplus [0.01 \times (1 - 0.2)] \quad \text{↳ } 1 - P(\text{Spam}) \\ &= 0.11 \end{aligned}$$

$$\begin{aligned} \bullet P(\text{Spam} | \text{'당첨'}) &= \frac{P(\text{'당첨'} | \text{Spam}) * P(\text{Spam})}{P(\text{'당첨'})} \\ &= \frac{(0.5 \times 0.2)}{0.11} \\ &= 0.9091 \end{aligned}$$

\therefore '당첨'이라는 단어를 포함하는 메일이 스팸일 확률
= 90.91%

Q 4.

(2)

- Hit : 9번
- Out : 6번
- 스윙 결과의 확률 분포
 $P(\text{Hit}) = 0.6$
 $P(\text{Out}) = 0.4$

- 엔트로피 계산

$$\begin{aligned}\text{Entropy} &= -P(\text{Hit}) * \log_2(P(\text{Hit})) - P(\text{Out}) * \log_2(P(\text{Out})) \\ &= -0.6 * \log_2(0.6) - 0.4 * \log_2(0.4) \\ &\approx 0.44218 + 0.52877 \\ &\approx 0.97095\end{aligned}$$

(3)

- 첫번째 팀의 스윙 결과 확률 분포
 $P(\text{Hit}) = 0.6$
 $P(\text{Out}) = 0.4$
- 두 번째 팀의 스윙 결과 확률 분포
 $Q(\text{Hit}) = 0.2$
 $Q(\text{Out}) = 0.8$

- KL Divergence

$$\begin{aligned} KL(P \parallel Q) &= P(\text{Hit}) * \log_2\left(\frac{P(\text{Hit})}{Q(\text{Hit})}\right) + P(\text{Out}) * \log_2\left(\frac{P(\text{Out})}{Q(\text{Out})}\right) \\ &= 0.6 * \log_2\left(\frac{0.6}{0.2}\right) + 0.4 * \log_2\left(\frac{0.4}{0.8}\right) \\ &\approx 0.6 * (1.585) + 0.4 * (-1) \\ &\approx 0.551 \end{aligned}$$

Q 5.

Logistic Regression 은 입력 변수와 목표 변수 사이의 관계를 모델링하는 이진 분류 알고리즘이다.

이를 학습하기 위해 Convex optimization 알고리즘을 사용하며, 그 과정은 다음과 같다.

1. 데이터 준비
2. 파라미터 초기화 : 주로 랜덤값 혹은 0 으로 초기화
3. 손실함수 정의 : 주로 cross-entropy 손실함수 사용
4. Gradient 계산 : Gradient는 손실함수를 각 파라미터로 미분한 값으로,
현재 파라미터 위치에서 가장 빠르게 손실함수를 감소시키는 방향을 나타낸다.
5. 파라미터 업데이트 : Learning rate를 사용하여 파라미터 업데이트의 크기를 조절한다.
6. 수렴 확인 : 업데이트된 파라미터로 다시 손실함수를 계산하고 Gradient를 구한다.
이 과정을 반복하여 최적값에 수렴하는지 확인
7. 최종 모델 평가 : 최적화가 완료되면 최종 모델을 평가