확률: 순열과 조합, 조건부 확률, 베이즈 정리

순열과 조합 (Permutation and Combination)

- 경우의 수를 계산하는 기본적인 방법입니다.
- **순열 (Permutation)**
  - 서로 다른 n개의 원소에서 r개를 **순서를 고려하여** 선택하거나 나열하는 경우의 수입니다.
  - e.g., 달리기 경주에서 1, 2, 3등 순서를 정하는 경우
  - 공식: $nPr = n! / (n-r)!$
- **조합 (Combination)**
  - 서로 다른 n개의 원소에서 r개를 **순서에 상관없이** 선택하는 경우의 수입니다.
  - e.g., 10명 중 3명의 대표를 뽑는 경우
  - 공식: $nCr = n! / (r! * (n-r)!)$

조건부 확률 (Conditional Probability)

- 어떤 사건 B가 일어났다는 조건 하에서, 다른 사건 A가 일어날 확률을 의미합니다.
- 기호로는 $P(A|B)$로 나타내며, "Probability of A given B"라고 읽습니다. 이는 정보가 추가되었을 때 확률이 어떻게 변하는지를 설명합니다.
- 공식: $P(A|B) = P(A ∩ B) / P(B)$

베이즈 정리 (Bayes' Theorem)

- 조건부 확률을 이용하여, 어떤 사건의 사전 확률(prior probability)과 추가적인 정보(evidence)를 통해 사후 확률(posterior probability)을 추론하는 방법입니다.
- 즉, $P(A|B)$를 알고 있을 때 $P(B|A)$를 계산할 수 있게 해줍니다. 스팸 메일 필터링, 의료 진단 등 많은 머신러닝 알고리즘의 기본 원리가 됩니다.
- 공식: $P(A|B) = [P(B|A) * P(A)] / P(B)$
  - $P(A|B)$: 사후 확률 (사건 B가 발생한 후의 사건 A의 확률)
  - $P(A)$: 사전 확률 (사건 A의 원래 확률)
  - $P(B|A)$: 가능도 (Likelihood) (사건 A가 발생했을 때 사건 B가 발생할 확률)
  - $P(B)$: 증거 (Evidence) (사건 B의 확률)

적용 가능한 상황

- **순열과 조합**: A/B 테스팅에서 가능한 테스트 조합의 수를 계산하거나, 샘플링 방법에서 가능한 표본의 경우의 수를 계산할 때.
- **조건부 확률**: 특정 고객 그룹(e.g., 20대 남성)이 특정 상품을 구매할 확률을 계산하는 등, 특정 조건 하에서의 사건 발생 가능성을 분석할 때.
- **베이즈 정리**:
  - **스팸 필터**: 'viagra'라는 단어가 포함된 메일(B)이 주어졌을 때, 이 메일이 스팸(A)일 확률 P(A|B)를 계산.
  - **의료 진단**: 어떤 사람이 특정 증상(B)을 보일 때, 그 사람이 실제로 특정 질병(A)을 가지고 있을 확률 P(A|B)를 계산.
  - **베이지안 추론**: 머신러닝 모델의 파라미터에 대한 믿음을 새로운 데이터를 통해 지속적으로 업데이트해 나갈 때.

순열과 조합

```
import math
from scipy.special import perm, comb
from itertools import permutations, combinations

# 예제: 10명의 선수 중 3명을 뽑아 금, 은, 동메달을 주는 경우의 수 (순열)
n, r = 10, 3
p_manual = math.factorial(n) / math.factorial(n - r)
p_scipy = perm(n, r)
p_iter = permutations(range(n), r)

print(f"Permutation ({n}P{r}): {p_manual}") # 720.0
print(f"Permutation (scipy): {p_scipy}") # 720
print(f"Permutation (iter): {len(list(p_iter))}") # 720

# 예제: 10명의 학생 중 3명의 대표를 뽑는 경우의 수 (조합)
c_manual = math.factorial(n) / (math.factorial(r) * math.factorial(n - r))
c_scipy = comb(n, r)
c_iter = combinations(range(n), r)
print(f"\nCombination ({n}C{r}): {c_manual}") # 120.0
print(f"Combination (scipy): {c_scipy}") # 120
print(f"Combination (iter): {len(list(c_iter))}") # 120
```

- `math`를 이용하여 직접 계산하여 순열과 조합을 도출해낼 수 있습니다.
- `scipy.special`의 `perm`과 `comb` 함수를 사용하면 순열과 조합을 간편하게 계산할 수 있습니다.
- `itertools`의 `permutations`과 `combinations`를 이용하면 각 경우의 수의 값들을 도출해낼 수 있습니다. (연산 시간이 오래 걸림)

베이즈 정리

- **주의사항**: 베이즈 정리를 적용하려면 사전 확률 $P(A)$와 가능도 $P(B|A)$를 정확히 아는 것이 중요합니다. 이 값들이 부정확하면 결과도 신뢰할 수 없습니다.

문제 상황: 어떤 질병 A의 발병률($P(A)$)은 `0.1%`이다. 이 질병을 진단하는 시약(B)은 실제 질병이 있는 사람(A)을 `99%`의 확률로 양성(B)으로 진단($P(B|A)$)하고, 질병이 없는 사람(A')을 `2%`의 확률로 잘못하여 양성(B)으로 진단($P(B|A')$)한다. 어떤 사람이 이 시약으로 양성 판정을 받았을 때, 이 사람이 실제로 질병 A를 가지고 있을 확률 $P(A|B)$는 얼마일까?

```
# 사전 확률 (Prior)
p_a = 0.001  # P(A): 실제로 병이 있을 확률
p_not_a = 1 - p_a # P(A'): 실제로 병이 없을 확률

# 가능도 (Likelihood)
p_b_given_a = 0.99 # P(B|A): 병이 있을 때 양성으로 진단할 확률 (민감도)
p_b_given_not_a = 0.02 # P(B|A'): 병이 없을 때 양성으로 진단할 확률 (위양성률)

# 증거 (Evidence) 계산: P(B) = P(B|A)*P(A) + P(B|A')*P(A')
# 전체 양성 판정 확률 = (실제 병이 있으면서 양성일 확률) + (실제 병이 없으면서 양성일 확
```

```
률)
p_b = (p_b_given_a * p_a) + (p_b_given_not_a * p_not_a)

# 베이즈 정리를 이용한 사후 확률 (Posterior) 계산
# P(A|B) = [P(B|A) * P(A)] / P(B)
p_a_given_b = (p_b_given_a * p_a) / p_b

print(f"P(A) = {p_a:.3f} (사전 확률)")
print(f"P(B) = {p_b:.5f} (전체 양성 판정 확률)")
print(f"P(A|B) = {p_a_given_b:.3f} (사후 확률)")
```

- **결과 해석**:

  ```
  P(A) = 0.001 (사전 확률)
  P(B) = 0.02097 (전체 양성 판정 확률)
  P(A|B) = 0.047 (사후 확률)
  ```

  결과적으로, 양성 판정을 받았음에도 불구하고 실제로 병을 가지고 있을 확률은 약 4.7%에 불과합니다. 이는 질병 자체가 매우 희귀하기 때문입니다(낮은 사전 확률). 이처럼 베이즈 정리는 우리의 직관과 다른 결과를 보여주며, 확률적 추론의 중요성을 일깨워줍니다.

장단점 및 대안

개념	장점	단점	관련 개념
순열/조합	경우의 수를 체계적으로 계산하는 기본 도구.	실제 세계의 복잡한 확률 문제에 직접 적용하기에는 너무 단순할 수 있음.	**몬테카를로 시뮬레이션**: 복잡한 확률 문제를 무작위 시뮬레이션을 통해 근사적으로 계산하는 방법.
조건부 확률	새로운 정보가 확률에 미치는 영향을 정량화할 수 있음.	두 사건의 교집합 확률 $P(A ∩ B)$를 알아야 계산 가능.	**독립 사건 (Independent Events)**: 한 사건의 발생이 다른 사건의 발생 확률에 영향을 주지 않는 경우. $P(A\|B) = P(A)$. 이 경우 $P(A ∩ B) = P(A) * P(B)$로 간단히 계산됨.
베이즈 정리	사전 지식과 새로운 증거를 결합하여 합리적인 추론을 가능하게 함. 정보가 업데이트됨에 따라 믿음을 갱신하는 과정을 수학적으로 모델링.	사전 확률 $P(A)$와 가능도 $P(B\|A)$를 정확히 추정하기 어려울 수 있음. 이 값들이 주관적이거나 부정확하면 결과의 신뢰도가 떨어짐.	**나이브 베이즈 분류기 (Naive Bayes Classifier)**: 베이즈 정리를 이용한 간단하고 강력한 분류 알고리즘. 모든 특성(feature)이 서로 독립적이라고 '순진하게(naively)' 가정함. **베이지안 네트워크 (Bayesian Network)**: 변수들 간의 조건부 의존성을 그래프 형태로 모델링하여 복잡한 확률적 관계를 추론.