



# Busan science high school

## 2023 Ocean ICT Festival

## 2023 BOIF

B  
05

QR 코드 영역  
QR 삽입 후  
테두리 삭제

Youtube 영상 QR

## 베이지 원리를 이용한 적조 발생확률 예측하기

Team Bagel

1110 백종윤 1115 이원혁

### 동기 및 목적

뉴스를 찾아보다 예전에 보도된 뉴스를 접하게 되었다. 남해안에 있는 양식장에서 적조에 의해 심각한 경제적 손실을 입었다는 내용이 담겼다. 적조에 의한 피해가 얼마나 심각한지 궁금하여 이에 대해 더 찾아보았고 적조로 인한 피해가 심각하다는 것을 알게 되었다. 이후 적조에 대해 알아보며 적조에 의한 피해를 줄일 수 있는 방법에 대해 고민하던 중 친구가 읽던 책에 베이지 원리에 관한 내용을 접하게 되었고 베이지 원리를 이용하여 적조 발생을 예측할 수 있지 않을까 생각하게 되었다. 그래서 우리는 미리 적조의 발생 위험에 대해 예측하여 미리 대응할 수 있도록 **베이지 원리**를 이용하여 **적조가 발생할 확률**을 구하는 프로그램을 제작하였다.

### 융합분야 및 이론적 배경

#### <정보과학-코딩>

컴퓨터 프로그래밍을 통해 자료를 수집, 가공, 분류한다.

우리는 코딩을 통해 입력 받은 정보를 분류한 후 이를 이용해 확률을 계산할 것이다.

#### <수학 - 베이지 원리>

베이지 원리는 사전 확률과 사후 확률 사이의 관계를 나타낸 정리로 다음의 식으로 나타낼 수 있다.

$$P(A|B)=P(B|A)P(A)/P(B)$$

(P(A): A의 사전확률, P(A|B): B가 주어질 때 A의 사후확률,

P(B|A): A가 주어질 때 B의 조건부 확률, P(B): B의 사전확률)

이 식을 확장하게 된다면  $P(A_i|B)=P(B|A_i)P(A_i)/\sum_i P(B|A_i)P(A_i)$ 로 나타낼 수 있다.

#### <생명과학-적조>

적조란 식물성플랑크톤의 대량번식에 의해 바다의 색이 적색계열로 바뀌는 현상으로 바다가 오염되면 적조가 발생해 생물의 입장을 차단함으로써 생태계 회복의 초석을 만든다. 그러나 적조가 발생하면 해양 관광업 및 수산업계 종사자들의 경제적 피해가 막심하다.

### 알고리즘

프로그램 시작

사전 자료 입력

베이지 원리로 적조발생  
확률 계산

확률과 범위를  
시각적으로 표현

예측 확률 출력

### 코딩에 사용한 모듈

**folium** : 파이썬에서 지도 시각화를 위해 사용하는 모듈. 여러 형태의 지도와 마커로 지역을 표기할 수 있다. 이 때문에 파이썬을 활용한 지리 관련 코딩에서 많이 활용된다. 출력되는 형태는 Google 지도와 비슷하다. (Goole Earth 아님.) 이번 코딩에서는 적조 발생 위험 범위를 시각화 하는 것에 사용되었다.

**numpy** : 파이썬에서 수치 데이터를 다루는 모듈. 주로 선형 대수 계산에서 사용된다. 넘파이는 일반 리스트보다 빠르고 메모리를 효율적으로 사용한다는 장점이 있다.

### 작성 코드

```
### 자료 입력받기

#받은자료
busan=[]

#받은 자료를 분류할 리스트
b_wd=[]
b_temp=[]
b_r_sea=[]
b_range=[]
zip_rng_wd_plus = []
r_r_sea_plus = []
data = []
data_1 = []

#자료 받아서 리스트 '부산'에 넣고 분류하기
n=int(input("자료 개수"))
print(f"자료입력 양식 : tt r ss \n (t:수온[°C], r:적조 발생 여부(0 or 1), s:발생반경[km])")
for i in range(1,n+1):
    x = input()
    busan.append(x)

for i in busan :
    b_temp.append(int(1[1:2]))

for i in busan :
    b_r_sea.append(int(1[3:4]))

for i in busan :
    b_range.append(int(1[5:7]))

#관찰 데이터 확률 자료
print(f"위에서 자료를 입력하신 순서대로 자료를 측정할 기기의 오차 확률을 입력해 주세요.(단 오차확률은 0과 1 사이)")
for i in range(1,n+1):
    y = str(round(float(input()),1))
    data.append(y)

for i in data:
    for j in j:
        data_1.append(str(k))
    data_1.append(' ')
del data_1[-1]

#평균 변경 구하기
R = round((sum(b_range)/(250/14)/n),1)

#적조 예측함수
import numpy as np
import matplotlib.pyplot as plt

def BUS_R_SEA(Observed_data, prior, Data_1):
    numerator = prior * sum([Data_1[i] for i in range(len(Observed_data)) if Observed_data[i] == 1])
    denominator = numerator * (1 - prior) + sum([Data_1[i] for i in range(len(Observed_data)) if Observed_data[i] == 0])
    posterior_prob = numerator / denominator
    return posterior_prob

#적조발생 데이터 가져오기(1: 적조발생, 0: 적조 미발생)
Observed_data = b_r_sea
print(Observed_data)

#사전확률 설정
prior = int(Observed_data.count(1))/len(Observed_data)
print(prior)

#관찰 데이터 확률
print(data_1)
n = len(data_1) + 2
for i in range(0,n):
    Data = data_1[0:n-1:2]
    print(Data)
    Data_ = list(map(int, Data))
    print(Data_)
    break
while 0 in Data_:
    Data_ = Data_.remove(0)
    print(Data_)
Data_ = np.array(Data_)
print(Data_==0.1)
Data_ = list(Data_*.1)
print(Data_==)

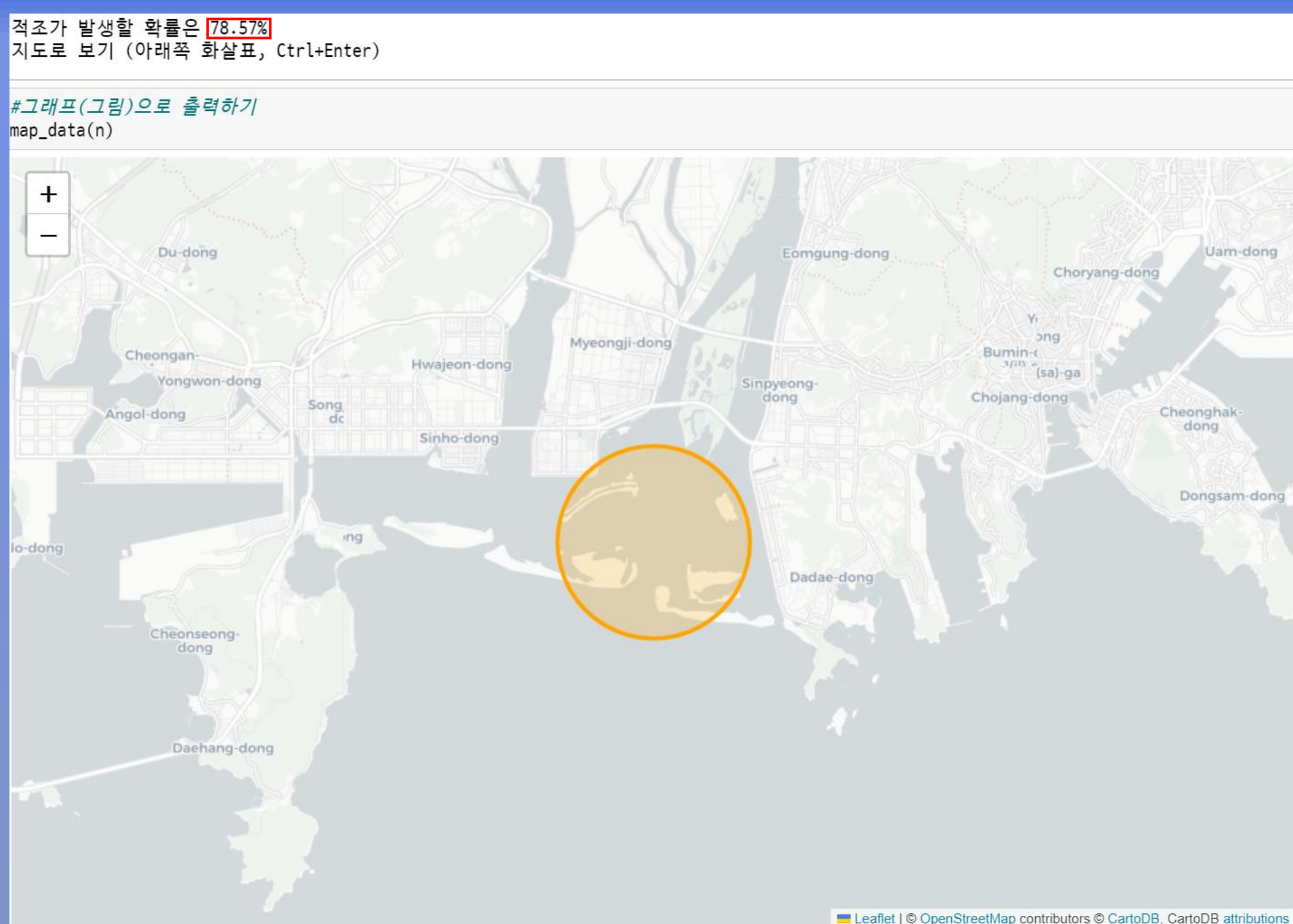
posterior = BUS_R_SEA(Observed_data, prior, Data_1)
print(f"사후 예측 확률: {posterior:.4f}")
print(round(posterior,4))

#지도로 출력하기
import folium as fm
#지도에 표시할 함수 정의
def map_data(n):
    if n == 1:
        m=fm.Map(location=[35.865389,128.926486], zoom_start=12, tiles="CartoDB positron")
        fm.CircleMarker(location=[35.865389,128.926486], radius=8, color='darkblue', popup='부산입력4', fill_color='darkblue').add_to(m)
        return m
    elif n == 2:
        m=fm.Map(location=[35.865389,128.926486], zoom_start=12, tiles="CartoDB positron")
        fm.CircleMarker(location=[35.865389,128.926486], radius=8, color='skyblue', popup='부산입력4', fill_color='skyblue').add_to(m)
        return m
    elif n == 3:
        m=fm.Map(location=[35.865389,128.926486], zoom_start=12, tiles="CartoDB positron")
        fm.CircleMarker(location=[35.865389,128.926486], radius=8, color='green', popup='부산입력4', fill_color='green').add_to(m)
        return m
    elif n == 4:
        m=fm.Map(location=[35.865389,128.926486], zoom_start=12, tiles="CartoDB positron")
        fm.CircleMarker(location=[35.865389,128.926486], radius=8, color='orange', popup='부산입력4', fill_color='orange').add_to(m)
        return m
    else:
        m=fm.Map(location=[35.865389,128.926486], zoom_start=12, tiles="CartoDB positron")
        fm.CircleMarker(location=[35.865389,128.926486], radius=8, color='red', popup='부산입력4', fill_color='red').add_to(m)
        return m

#결과값 출력하기
k = round(posterior,4)
num = k*100
if k >= 0 and k <= 0.2:
    n=1
    map_data(n)
    print(f"적조가 발생할 확률은 (k=100)%")
elif k > 0.2 and k <= 0.4:
    n=2
    map_data(n)
    print(f"적조가 발생할 확률은 (k=100)%")
elif k > 0.4 and k <= 0.6:
    n=3
    map_data(n)
    print(f"적조가 발생할 확률은 (k=100)%")
elif k > 0.6 and k <= 0.8:
    n=4
    map_data(n)
    print(f"적조가 발생할 확률은 (k=100)%")
else:
    n=5
    map_data(n)
    print(f"적조도 35.877, 경도 128.83에서 적조가 발생할 확률은 (num:.2f)%")
    print("지도로 보기 (아래쪽 화살표, Ctrl+Enter)")
```

1. 적조발생에 관한 사전 자료와 사전 확률, 적조발생 정확도를 입력 받아 리스트로 분류한다.
2. 입력 받은 자료를 이용해 베이지 원리를 계산하여 사후 예측 값을 출력한다.
3. 출력된 사후 예측 값을 백분율로 고친 후 시각화를 곁들여 화면에 출력한다.

### 실제 출력 화면



### 기대 효과 및 보완할 점

베이지 원리를 사용하기 위해서는 정확한 확률 계산을 위해 빅데이터를 수집하여 처리하는 것이 필요하다. 그러나 그만큼 자료를 구할 수 없어 겨우 몇 개의 자료만으로 계산하다 보니 아주 큰 오차가 발생할 수 밖에 없었다. 또 시간이 부족해 위험 지역 표시가 임의의 지역을 표시하게 되어 아쉬웠다. 이런 부분을 보완하면 정확한 적조발생확률을 예측하여 적조로 인한 피해를 막을 수 있을 것이다.

### 느낀점

학교 수업 시간에는 그날 배운 것들로 그날그날 문제를 푼다는 느낌을 강하게 받았고 문제를 해결하는데 급급하였다. 그러나 이번 OCEAN ICT를 통해서 내가 직접 코딩해보고 오랜 기간동안 여러 시행착오와 오류 등을 경험하고 수정해 나가다 보니 오류를 해결하고 결과물을 보았을 때 평소와는 비교도 되지 않을 만큼의 만족감과 희열감, 쾌감, 뿌듯함 등등의 감정이 휘몰아치는 듯한 기분이 들었다.