



# Busan science high school

## 2023 Ocean ICT Festival

### 2023 BOIF

QR 코드 영역  
QR 삽입 후  
테두리 삭제

Youtube 영상 QR

## 선형회귀를 이용한 어획량 시각화

팀명: 김유이(Yui Kim)

2110김수훈 2114유지한 2115이영서

### 작품목적

우리의 작품은 해양과 기후 데이터와 경제, 사회, 환경 데이터까지 통합하여 어획량과 회귀 분석을 수행하여, 정확한 미래의 어획량을 지도에 표시하는 것이다. 다양한 분야에서 수산업의 어획량을 바탕으로 지도에 보기 쉽게 시각화하여 유용한 정보를 제공하는 것이 목적이다. 미래의 어획량을 예측함으로써 어업 산업의 미래를 파악하고자 작품 목적을 수립하였다. 이 작품을 토대로 어업 종사자들이 전략적으로 어업 계획을 취할 수 있는 정보를 제공하고 싶다.

### 정보과학과 융합분야

**지구과학:** 수온, 해수의 염도 등 해양의 관찰 데이터를 분석하여 미래의 어획량을 예측할 수 있다  
**사회:** GDP, 수입량과 수출량, 인구를 분석하고 예측하는 것을 통해 미래에 어업에 의한 경제적 이익을 특정하여 정부의 예산 확보와, 어업인 구제 활동에 이용할 수 있다.  
**환경:** 강수량, 녹조현상, 이산화탄소 농도, 기상이변 발생빈도와 같은 환경 데이터를 통하여 미래 어획량을 예측할 것이다.

### 알고리즘

- 한국해양수산 개발원에서 어획량 데이터와 E-나라지표에서 사회, 해양 데이터를 찾는다.
- 과거 몇 년의 동해, 서해, 남해의 어획량 데이터를 조사해서 평균적으로 달마다 어느 정도의 어업이 이루어졌는지를 우리나라의 지도에 색을 이용해 시각화한다.
- 선형회귀를 통하여 시도별 미래의 어업량을 도출하고, 지도에 색으로 시각화한다.
- 선형회귀분석을 통해 GDP, 수출입 등과 같은 사회 데이터와 어획량의 관계를 분석한다.
- 과거의 해양데이터, 환경데이터와 어획량을 회귀분석하여 미래의 어업량을 예측한다.
- 4, 5과정을 통해 예측한 시도별 미래의 어업량을 우리나라 지도에 색으로 시각화한다.

### 코드

#### (1) 선형회귀

```
from sklearn.linear_model import LinearRegression
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
```

```
filename='fishing_2.csv'
data=pd.read_csv(filename)
df=np.array(data.values)
data
```

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.pairplot(data[['years', 'fishing', 'GDP', 'trade', 'precipitation', 'Annual Mean', '5-yr smooth', 'CO2']])
plt.show()
```

1.선형회귀를 위한 모듈을 import하고, csv파일을 읽는다.  
pandas를 활용해 표로 정리하고,  
seaborn을 활용해 상관관계를 한눈에 나타낸다.

```
def loss(Y,pred_Y):
    MeanSQ=np.sum((Y-pred_Y)**2)
    Loss=MeanSQ/len(Y)
    return Loss
```

2. 어획량을 부산, 인천, 경기, 서울, 전남, 경북 등 시도별로 시계열 선형회귀분석하고 미래 2024년의 시도별 어획량을 딕셔너리에 저장한다. 전체 총 어획량을 회귀분석하여 산점도와 그래프로 나타낸다. 선형회귀의 loss(손실) 값과 평균제곱오차 값과 결정계수를 출력한다.

```
for i in range(8,18):
    col_name = data.columns[i]
    x_data['years']=int
    Y=data[col_name]
    lineFit=LinearRegression()
    lineFit.fit(X.values.reshape(-1,1),Y)
    beta_0=lineFit.coef_
    beta_1=lineFit.intercept_
    pred_Y=lineFit.predict(X.values.reshape(-1,1))
    if beta_0+2024+beta_1<0:
        l[col_name]=None
    else:
        l[col_name]=int(beta_0+2024+beta_1)
X=data['years'][:13]
Y=data['Ulsan'][:13]
lineFit=LinearRegression()
lineFit.fit(X.values.reshape(-1,1),Y)
beta_0=lineFit.coef_
beta_1=lineFit.intercept_
pred_Y=lineFit.predict(X.values.reshape(-1,1))
if beta_0+2024+beta_1<0:
    l['Ulsan']=None
else:
    l['Ulsan']=int(beta_0+2024+beta_1)
X=data['years']
Y=data['fishing']
lineFit=LinearRegression()
lineFit.fit(X.values.reshape(-1,1),Y)
beta_0=lineFit.coef_
beta_1=lineFit.intercept_
pred_Y=lineFit.predict(X.values.reshape(-1,1))
print('기울기:', lineFit.coef_)
print('절편:', lineFit.intercept_)
print('Loss:', 2*1 loss(Y,pred_Y))
```

```
X=df[:,0]
Y=df[:,1]
x_train, x_test, y_train, y_test=train_test_split(X.reshape(-1,1),Y,test_size=0.3,random_state=5)
pred_Y_train=lineFit.predict(x_train)
mse_train=mean_squared_error(y_train,pred_Y_train)
print("MSE on train data: %f"%mse_train)
pred_Y_test=lineFit.predict(x_test)
mse_test=mean_squared_error(y_test,pred_Y_test)
print("MSE on test data: %f"%mse_test)
r2_train = r2_score(y_train, pred_Y_train)
r2_test = r2_score(y_test, pred_Y_test)
print("train의 결정계수: %f"%r2_train)
print("test의 결정계수: %f"%r2_test)
```

```
X=data[['2022'][:16]]
Y=data[['fishing'][:16]]
lineFit=LinearRegression()
lineFit.fit(X.values.reshape(-1,1),Y)
beta_0=lineFit.coef_
beta_1=lineFit.intercept_
pred_Y=lineFit.predict(X.values.reshape(-1,1))
print('기울기:', lineFit.coef_)
print('절편:', lineFit.intercept_)
print('Loss:', loss(Y,pred_Y))
plt.plot(X,Y,'*')
plt.plot(X,pred_Y)
plt.show()
X=df[['2022'][:16]]
Y=df[['fishing'][:16]]
x_train, x_test, y_train, y_test=train_test_split(X.reshape(-1,1),Y,test_size=0.3,random_state=5)
pred_Y_train=lineFit.predict(x_train)
mse_train=mean_squared_error(y_train,pred_Y_train)
print("MSE on train data: %f"%mse_train)
pred_Y_test=lineFit.predict(x_test)
mse_test=mean_squared_error(y_test,pred_Y_test)
print("MSE on test data: %f"%mse_test)
r2_train = r2_score(y_train, pred_Y_train)
r2_test = r2_score(y_test, pred_Y_test)
print("train의 결정계수: %f"%r2_train)
print("test의 결정계수: %f"%r2_test)
```

3. GDP, 수출량, 강수량, CO2농도 등 다양한 사회, 환경, 해양 데이터를 차례로 선형회귀한다.

#### (2) 지도 시각화

```
import folium
import matplotlib.pyplot as plt
from folium.plugins import MarkerCluster
print('1985년부터 2021년까지 어획량을 확인해보세요')
print('2024년의 예측 어획량을 확인해보세요')
year=int(input('please write the year(1985~2021,2024):'))
y=year-1985
center_coords=[36.5,127.5]
m=folium.Map(location=center_coords, zoom_start=7)
marker_cluster=MarkerCluster().add_to(m)
title.html = f'<h3 align="center" style="font-size:20px;"><b>(year)년 시도별 어획량</b></h3>'
m.get_root().html.add_child(folium.Element(title.html))
fishing_data=[('부산',df[:,1][y],35.1786,129.0756),('인천',df[:,2][y],37.4562567,126.750262),('울산',df[:,3][y],35.5383773,129.3113596)]
a=[]
for i in range(len(fishing_data)):
    a.append(df[:,1][y])
max_amount=max([amount for amount in a])
for city, fishing_amount, latitude, longitude in fishing_data:
    marker_cluster.add_child(folium.Marker(location=(latitude, longitude), popup=f'fishing_amount>50000 else 'orange' if fishing_amount>10000 else 'yellow' if fishing_amount>10000 else 'green',
    normalizer=folium.CircleMarker(location=(latitude, longitude), radius=marker_size, color=marker_color, fill=True, fill_color=marker_color, fill_opacity=0.5)
    output_filename = f'({year})년도 어획량 지도.html'
    m.save(output_filename)
```

각 시도별 어획량 파일을 연다.  
pandas를 이용하여 표로 한눈에 보기 쉽게 나타낸다.  
그 뒤, folium을 활용하여 각 시도별 위도, 경도와 어획량을 묶고, 지도에 마커로 표시한다.  
마커는 어획량에 따라 100000Mt 이상은 빨간색, 50000Mt 이상은 주황색, 10000Mt 이상은 노란색, 그 이하는 녹색으로 설정하였고, 마커의 크기도 어획량에 비례한다.  
초기에 연도를 입력받으면 그 연도의 시도별 어획량 데이터를 가지고 html형식으로 저장되며, 지도를 확대, 축소하여 마커를 통해 어획량을 확인할 수 있게 하였다.

### 결과

어획량과 GDP, 수출량, 강수량, 평균온도, CO2농도가 1850년부터 2021년까지 나타내어졌다.



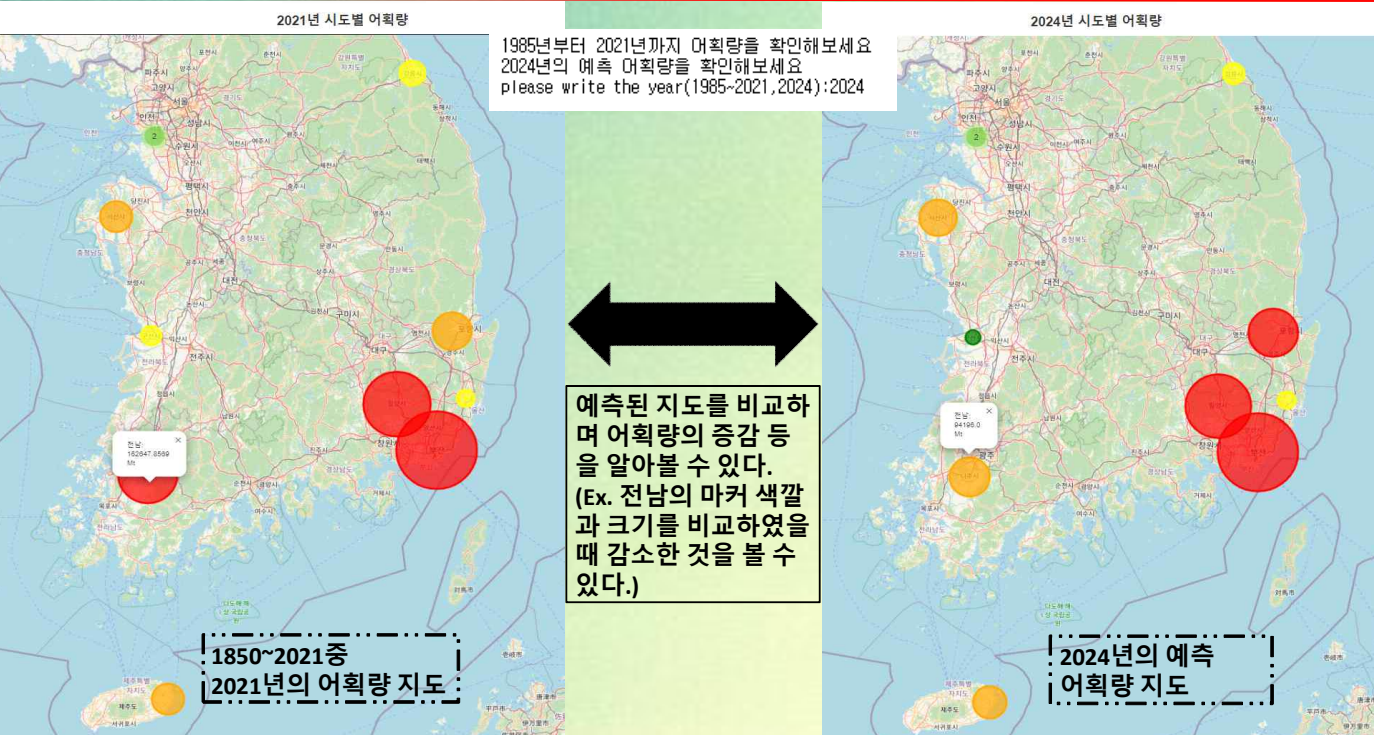
어획량 선형회귀 결과

years	Busan	Incheon	Ulsan	Gyeonggi	Gangwon	Chungnam	Jeonbuk	Jeonnam	Gyeongbuk	Gyeongnam	Jeju
0	1985	1985	1985	1985	1985	1985	1985	1985	1985	1985	1985
1	1986	1986	1986	1986	1986	1986	1986	1986	1986	1986	1986
2	1987	1987	1987	1987	1987	1987	1987	1987	1987	1987	1987
3	1988	1988	1988	1988	1988	1988	1988	1988	1988	1988	1988
4	1989	1989	1989	1989	1989	1989	1989	1989	1989	1989	1989
5	1990	1990	1990	1990	1990	1990	1990	1990	1990	1990	1990
6	1991	1991	1991	1991	1991	1991	1991	1991	1991	1991	1991
7	1992	1992	1992	1992	1992	1992	1992	1992	1992	1992	1992
8	1993	1993	1993	1993	1993	1993	1993	1993	1993	1993	1993
9	1994	1994	1994	1994	1994	1994	1994	1994	1994	1994	1994
10	1995	1995	1995	1995	1995	1995	1995	1995	1995	1995	1995
11	1996	1996	1996	1996	1996	1996	1996	1996	1996	1996	1996
12	1997	1997	1997	1997	1997	1997	1997	1997	1997	1997	1997
13	1998	1998	1998	1998	1998	1998	1998	1998	1998	1998	1998
14	1999	1999	1999	1999	1999	1999	1999	1999	1999	1999	1999
15	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000	2000
16	2001	2001	2001	2001	2001	2001	2001	2001	2001	2001	2001
17	2002	2002	2002	2002	2002	2002	2002	2002	2002	2002	2002
18	2003	2003	2003	2003	2003	2003	2003	2003	2003	2003	2003
19	2004	2004	2004	2004	2004	2004	2004	2004	2004	2004	2004
20	2005	2005	2005	2005	2005	2005	2005	2005	2005	2005	2005
21	2006	2006	2006	2006	2006	2006	2006	2006	2006	2006	2006
22	2007	2007	2007	2007	2007	2007	2007	2007	2007	2007	2007
23	2008	2008	2008	2008	2008	2008	2008	2008	2008	2008	2008
24	2009	2009	2009	2009	2009	2009	2009	2009	2009	2009	2009
25	2010	2010	2010	2010	2010	2010	2010	2010	2010	2010	2010
26	2011	2011	2011	2011	2011	2011	2011	2011	2011	2011	2011
27	2012	2012	2012	2012	2012	2012	2012	2012	2012	2012	2012
28	2013	2013	2013	2013	2013	2013	2013	2013	2013	2013	2013
29	2014	2014	2014	2014	2014	2014	2014	2014	2014	2014	2014
30	2015	2015	2015	2015	2015	2015	2015	2015	2015	2015	2015
31	2016	2016	2016	2016	2016	2016	2016	2016	2016	2016	2016
32	2017	2017	2017	2017	2017	2017	2017	2017	2017	2017	2017
33	2018	2018	2018	2018	2018	2018	2018	2018	2018	2018	2018
34	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019	2019
35	2020	2020	2020	2020	2020	2020	2020	2020	2020	2020	2020
36	2021	2021	2021	2021	2021	2021	2021	2021	2021	2021	2021
37	2022	2022	2022	2022	2022	2022	2022	2022	2022	2022	2022
38	2023	2023	2023	2023	2023	2023	2023	2023	2023	2023	2023
39	2024	2024	2024	2024	2024	2024	2024	2024	2024	2024	2024

부산, 울산, 경기, 전남, 전북, 경남, 경북, 제주 등 각 시도별 어획량이 1850년부터 2021년까지 나타내어져 있고, 선형회귀를 통한 2024년의 예측 어획량 또한 나타내어져 있다.

수출량과 어획량 선형회귀 결과

CO2농도와 어획량 선형회귀 결과



예측된 지도를 비교하며 어획량의 증감 등을 알아볼 수 있다. (Ex. 전남의 마커 색깔과 크기를 비교하였을 때 감소한 것을 볼 수 있다.)

1850~2021중 2021년의 어획량 지도

2024년의 예측 어획량 지도

### 결론 및 기대효과

**결론:** 어획량과 GDP는 낮은 양의 상관관계를 가졌지만, 수출량과 이산화탄소 농도는 큰 양의 상관관계를 가지는 것을 알게되었다.  
그리고 제주, 충남, 경북은 2024년도의 어획량이 증가할 것이라고 예측하였다.  
**기대효과:** 미래의 어업량을 토대로 어업 종사자에게 유용한 정보를 제공할 수 있다.