

# 바다환경에 따른 어종별 어획 예측 모델

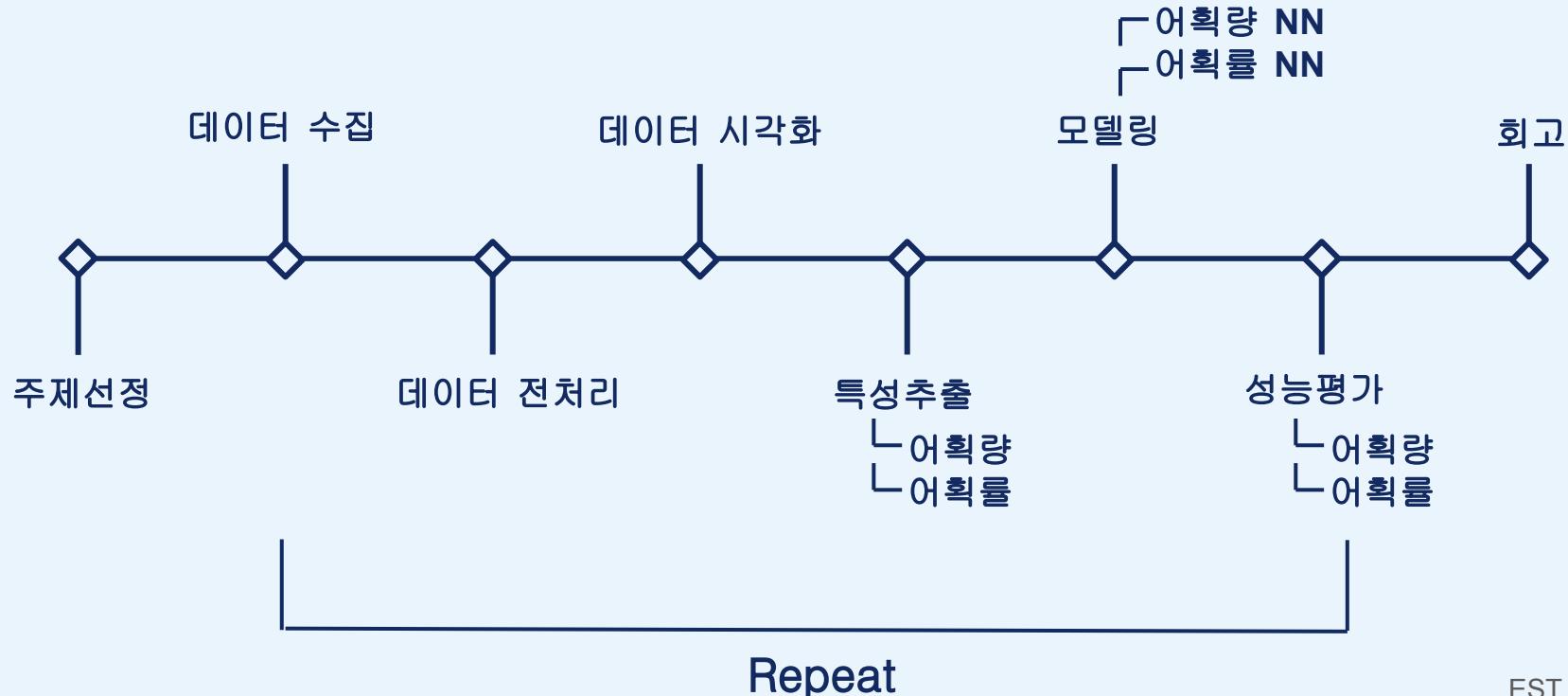
---

6조

TIME SERIES  
PROJECT

# Contents

---



Repeat

# WBS

번호	구분	4/28	4/29	4/30	5/1	5/2	5/3	5/4	5/5	5/6	5/7	5/8	5/9	5/10	5/11	5/12	5/13	5/14	5/15	5/16
		월	화	수	목	금	토	일	월	화	수	목	금	토	일	월	화	수	목	금
모델링																				
1-1	주제선정																			
1-2	전처리																			
1-3	모델링 협의																			
1-4	학습 / 결과피드백																			
1-5	완성																			
기획안공유 (오후2시)																				
2-1	초안																			
2-2	피드백																			
2-3	마무리																			
2-4	발표																			
중간 발표(오후 2시), 최종 발표 (오후 1시)																				
3-1	초안																			
3-2	피드백																			
3-3	마무리																			
3-4	발표																			

# 프로젝트 배경

기후(수온)변화에 따른 주요 어종 어획량 변화

□ 기후변화에 따른 표층수온 상승

○ 1850년대부터 경제 및 인구성장 등 다양한 원인으로 인해 지구 온난화가 심화되어 왔으며, 지난 100년(1918~2017년)동안 전 세계 평균기온은  $1.55^{\circ}\text{C}$ , 표층수온은  $0.62^{\circ}\text{C}$  상승

 뉴스펭귄 · [www.newspenguin.com](http://www.newspenguin.com) › 뉴스펭귄 › 기후 › 기후뉴스

열받은 바다, 올여름 '이상고수온' 발생 가능성 높아

 제주도 사랑, 그리고 환경 사랑 · 2025.01.22.

(해외환경뉴스) 2024년 해수면 온도와 더 깊은 수온, 사상 최고치 기록

- 주요 어종 어획량 증가
- 어장이동

기대효과

어획의 불확실성 감소

# 데이터 수집

## 수온 데이터

date	관측소	저층_중앙	중층_중앙	표층_중앙	저층_최고	중층_최고	표층_최고	저층_최저	중층_최저	표층_최저	저층_평균	중층_평균	표층_평균
2011-03-01	삼척(bsc8)	8.6	8.6	8.5	9.1	9	8.8	8.4	8.4	8.2	8.6	8.6	8.5
2011-03-02	삼척(bsc8)	8.6	8.9	8.7	9	9.1	9.3	7.3	8.5	8.6	8.5	8.8	8.8
2011-03-03	삼척(bsc8)	7.9	8.3	8.4	8.3	8.8	9.1	7.6	7.9	7.9	8	8.3	8.5
2011-03-04	삼척(bsc8)	8.45	8.7	8.2	8.7	9.05	8.3	8.25	8.4	8.2	8.5	8.7	8.2
2011-03-05	삼척(bsc8)	9	9.1	9.2	9.1	9.3	9.3	8.9	8.9	9	9	9.1	9.2
2011-03-06	삼척(bsc8)	9.2	9.7	9.8	9.5	9.7	9.9	8.9	9.7	9.7	9.2	9.7	9.8
2011-03-07	삼척(bsc8)	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6	9.6

(출처 : [해양수산환경관측시스템](#))

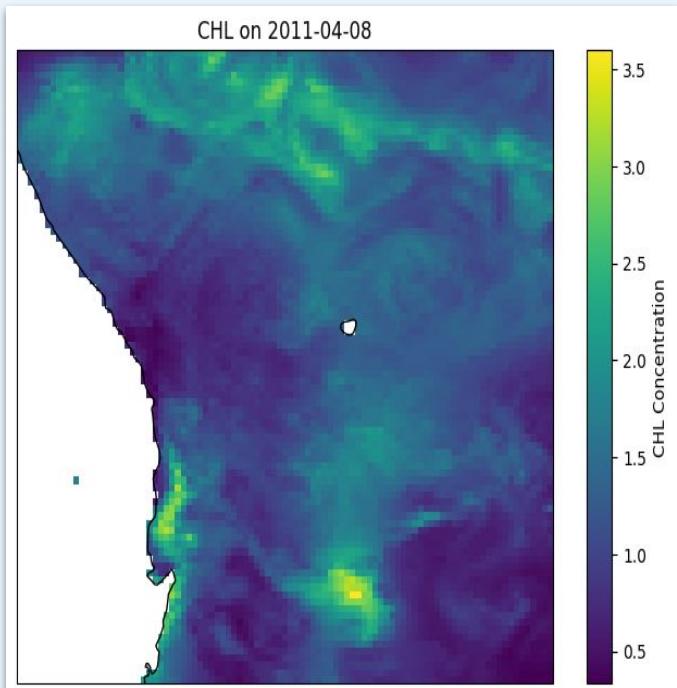
## 기상 데이터

avgTa	minTa	minTaHrm	maxTa	maxTaHrm	maxInsWs	maxInsWs'	maxInsWs'	maxWs	maxWsWd	maxWsHrr	avgWs	hr24SumR	maxWd	avgTd	minRhm
1.5	-0.5	2354	3.1	1245	9.4	340	1318	6.1	340	1425	2.2	1937	340	-0.6	76
0.4	-2.7	2356	4	1418	12	320	1242	6.8	290	1242	3.2	2723	320	-9.5	25
-1.6	-4.9	634	2.2	1148	10.3	360	1519	6.4	20	1246	2.6	2256	200	-11.1	28
0.6	-3.9	1	3.8	1141	8.2	250	242	5.8	20	1223	2.6	2254	200	-10.4	26
4.8	-0.7	25	8.4	1216	6.6	180	445	4.3	110	1109	2.2	1923	110	-6.9	26
6.2	1.8	221	10.2	1418	6.9	110	1514	5	110	1517	2	1740	290	-5.6	30
3.3	0.2	2134	6.5	4	11.3	320	702	7.8	360	1116	4	3442	360	-5.9	38

minRhmH	avgRhm	avgPv	avgPa	maxPs	maxPsHrm	minPs	minPsHrm	avgPs	ssDur	sumSsHr	avgTs	minTg
1511	86.1	5.9	1008.5	1016	0	1011.6	1417	1013.4	11.4	0.2	0.5	-3
1411	49.4	3	1012.5	1020.7	2150	1014.4	319	1017.5	11.4	9.5	1.7	-4.8
1109	49.4	2.6	1015.2	1021.3	821	1018.6	1444	1020.2	11.4	10.6	2.5	-7.9
1103	43.9	2.8	1017.1	1023.1	938	1020.3	44	1022.1	11.5	10.8	2.9	-6.6
1030	42.8	3.7	1014.9	1022.2	0	1017	1751	1019.8	11.5	5.9	2.6	-4.4
1247	42.8	4.1	1010.2	1018.6	0	1012.6	2246	1015.1	11.5	3	3.5	-3.5
2328	51	4	1007.8	1014.4	2331	1010.7	1507	1012.7	11.6	9.8	5.2	-3.2

(출처 : [기상자료 개방포털](#))

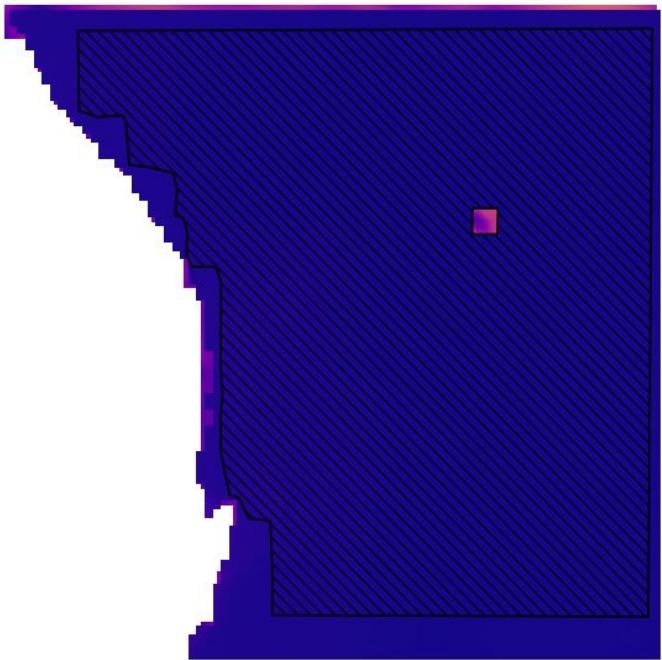
## 해상 이미지 데이터



(출처 : [코페르니쿠스 해양 환경 모니터링 서비스\(CMEMS\)](#))

# 데이터 수집

## 해상이미지 데이터



(출처: [코페르니쿠스 해양 환경 모니터링 서비스\(CMEMS\)](#))

## 해상이미지 데이터 (빗줄친부분)

time	염분_50m	염분_100m	염분_200m	염분_300m	염분_400m	염분_500m	해저수온_50m	해저수온_100m	해저수온_200m	해저수온_300m
2011-01-01	33.94462746	33.94937436	34.05825861	34.01897871	34.04684231	34.06458853	13.57913748	12.96342405	6.983299792	1.476057754
2011-01-02	33.95560548	33.95678852	34.05585499	34.01876437	34.04629331	34.0645726	13.52555688	12.89550593	6.98237208	1.476556785
2011-01-03	33.96481027	33.9635935	34.05400385	34.01840804	34.04578259	34.06456552	13.46298044	12.82522491	6.954859204	1.47565446
염분_400m	염분_500m	해저수온_50m	해저수온_100m	해저수온_200m	해저수온_300m	해저수온_400m	해저수온_500m			
34.04684231	34.06458853	13.57913748	12.96342405	6.983299792	1.476057754	0.911934204	0.680976657			
34.04629331	34.0645726	13.52555688	12.89550593	6.98237208	1.476556785	0.915629362	0.683508772			
34.04578259	34.06456552	13.46298044	12.82522491	6.954859204	1.47565446	0.919101026	0.6861488			

(출처: [코페르니쿠스 해양 환경 모니터링 서비스\(CMEMS\)](#))

## 해상기후 데이터

평균 풍속(m/s)	평균 기압(hPa)	평균 상대습도(%)	평균 기온(°C)	평균 수온(°C)	평균 최대 파고	평균 유의 파고	평균 파주기(sec)
5.2	1012.8	83	5.3	10.1	2.6	1.6	3.7
8.8	1016.7	64	2.6	10	3.4	2	3.5
4.7	1019.7	53	2	9.7	2.5	1.5	3.8

(출처: 기상자료 개방포털)

## 어획률 데이터

가자미류	청어	새우	대구	도루묵	기타어류	총어획량
73.4	0.4	7	2.5	6.3	10.4	107127
62.7	7.6	6.8	4.1	3.2	15.6	37872
48.5	1.6	3.4	2.4	37.8	6.3	79934

(출처: 국립수산과학원)

EST Soft

# 데이터 전처리

## 데이터 조합 케이스 결측치

### CASE1

```
ocean_data_1 = pd.read_csv('data/surface_Tm+Weather.csv')
ocean_data_1.isnull().sum()
✓
date          0
관총소        0
저출_중앙 수온(°C)  0
중출_중앙 수온(°C)  0
표출_중앙 수온(°C)  0
저출 최저 수온(°C)  0
중출 최고 수온(°C)  0
표출_최고 수온(°C)  0
저출_최저 수온(°C)  0
중출_최저 수온(°C)  0
표출_평균 수온(°C)  0
중출_평균 수온(°C)  0
표출_평균 수온(°C)  0
avgTa         0
minTa         0
minTaHrmr    0
maxTa         0
maxTaHrmr    0
maxInsWS     0
maxInsWSHrd  0
maxInsWSHrmr 0
maxWs         0
maxWsHrd    0
maxWsHrmr   0
...
ssDur         0
sumSsHr       0
avgTs         0
minTg         0
```

### CASE2 & CASE3

```
ocean_data_2 = pd.read_csv('data/surface_Tm+weather+cnn.csv')
ocean_data_2.isnull().sum()
✓
Date          0
관총소        0
저출_중앙 수온(°C)  0
중출_중앙 수온(°C)  0
표출_중앙 수온(°C)  0
저출 최저 수온(°C)  0
중출 최고 수온(°C)  0
표출_최고 수온(°C)  0
저출_최저 수온(°C)  0
중출_최저 수온(°C)  0
표출_평균 수온(°C)  0
중출_평균 수온(°C)  0
표출_평균 수온(°C)  0
...
cnn_feat_2043  0
cnn_feat_2044  0
cnn_feat_2045  0
cnn_feat_2046  0
cnn_feat_2047  0
Length: 2091, dtype: int64
```

### CASE4

```
ocean_data_4 = pd.read_csv('data/ocean_env.csv')
ocean_data_4.isnull().sum()
✓
time          0
염분_50m      0
염분_100m     0
염분_200m     0
염분_300m     0
...
규산염_200m    0
규산염_300m    0
규산염_400m    0
규산염_500m    0
mean_광학산 감쇠 계수  0
Length: 64, dtype: int64
```

### CASE5 & CASE6

```
ocean_data_5 = pd.read_csv('data/Ocean_env+Ocean_weather.csv')
ocean_data_5.isnull().sum()
✓
time          0
염분_50m      0
염분_100m     0
염분_200m     0
염분_300m     0
...
평균 기온(°C)  0
평균 수온(°C)  0
평균 최대 파고(m)  0
평균 유의 파고(m)  0
평균 파주기(sec)  0
Length: 66, dtype: int64
```

< 바다온도+기상 >

< 바다온도+기상+CNN+(Target) >

< 바다성분 >

< 바다성분+기상+(Target) >

EST Soft

# 데이터 전처리

## 어획률/어획량 결측치 처리

```
rate.isnull().sum()
```

✓

Date	0
가자미류	41
청어	41
새우	41
대구	41
도루묵	41
기타어류	41
총어획량	41
<b>dtype:</b>	<b>int64</b>

해당 결측치는 금어기.

금어기 여부를 파생 변수  
생성 및 0으로 결측치 채움

```
rate['금지여부'] = rate['가자미류'].isna().astype(int)  
rate.fillna(0,inplace=True)
```

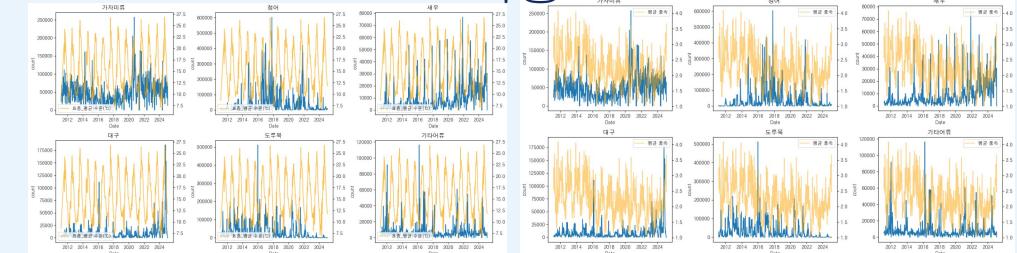
```
rate.isnull().sum()
```

✓

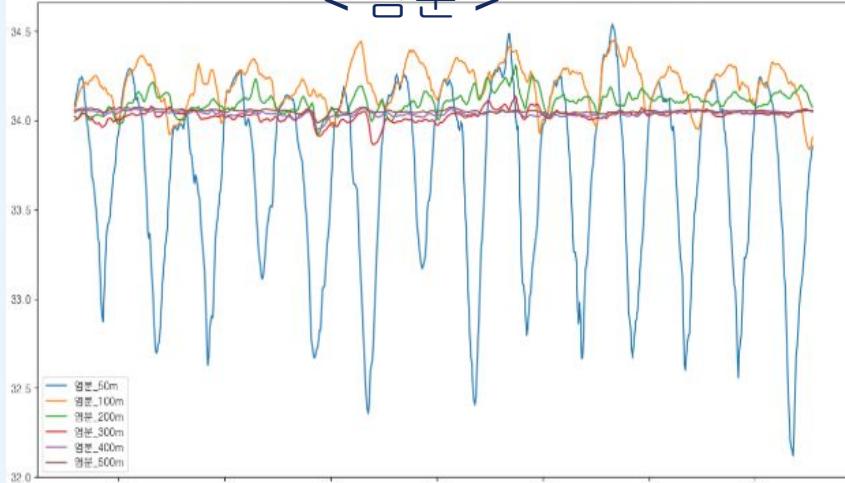
Date	0
가자미류	0
청어	0
새우	0
대구	0
도루묵	0
기타어류	0
총어획량	0
금지여부	0
<b>dtype:</b>	<b>int64</b>

# 데이터 시각화

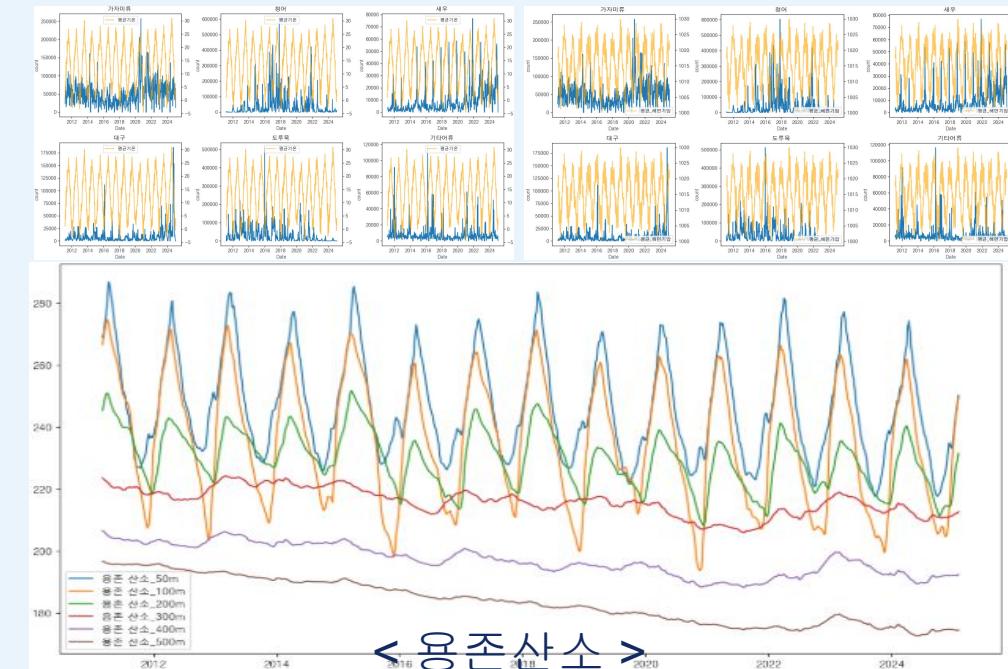
< 어종 >



< 영분 >



< 용존산소 >



< 해저수온 >

# 특성추출

---

Total

CNN

PCA

Stepwise

# 모델링

CNN



```
Cnn = Sequential()
Cnn.add(Conv1D(16,2, activation='relu', input_shape=(X_train.shape[1],X_train.shape[2])))
Cnn.add(MaxPooling1D(2))
Cnn.add(Flatten())
Cnn.add(Dense(64,activation = 'relu'))
Cnn.add(Dense(32,activation = 'relu'))
Cnn.add(Dense(y_train.shape[1]))
Cnn.compile(optimizer='adam', loss='mse')
```



	특성추출	이회랑		이회률	
		MSE	MAE	MSE	MAE
Case1	ALL	202,202,501.10	11,224.87	145.90	9.08
	CNN	117,752,468.48	8,053.59	145.90	8.77
	PCA	154,650,535.69	9,316.55	89.00	6.89
	Stepwise	140,502,596.04	9,250.80	107.57	7.93
Case2	ALL	235,052,861.55	11,875.49	179.80	10.58
	CNN	158,132,890.78	9,110.33	108.16	7.40
	PCA	108,797,088.63	8,202.19	144.97	8.81
	Stepwise	141,997,068.71	9,004.95	121.04	8.28
Case3	ALL	252,567,044.34	12,772.77	219.49	11.62
	CNN	181,437,547.31	8,837.32	130.08	8.48
	PCA	119,846,398.55	8,683.17	127.42	8.21
	Stepwise	143,315,065.57	9,105.12	124.22	8.54
Case4	ALL	152,831,793.81	9,563.66	161.99	9.88
	CNN	99,172,094.38	7,551.86	91.39	6.94
	PCA	100,080,994.04	8,081.44	91.29	7.29
	Stepwise	249,696,292.70	11,736.55	202.78	11.13
Case5	ALL	210,654,485.26	11,525.29	95.24	7.11
	CNN	155,964,669.28	8,776.36	105.46	7.23
	PCA	122,820,782.03	8,541.50	115.46	7.76
	Stepwise	268,537,681.03	12,664.60	148.58	8.95
Case6	ALL	169,115,045.59	10,259.85	207.72	11.83
	CNN	129,928,392.12	8,139.80	116.36	8.23
	PCA	107,989,280.97	8,132.09	126.12	8.45
	Stepwise	112,485,128.51	8,183.26	226.10	12.22

# 모델링

RNN



```
Rnn = Sequential()
Rnn.add(SimpleRNN(units=16,activation='tanh', input_shape=(X_train.shape[1],X_train.shape[2])))
Rnn.add(Dense(y_train.shape[1]))
Rnn.compile(optimizer='adam', loss='mse')
```



	특성추출	어획량		어획률	
		MSE	MAE	MSE	MAE
Case1	ALL	146383340.64	9351.47	94.05	7.10
	CNN	131138274.51	8208.71	85.58	6.05
	PCA	110814310.37	7794.88	61.07	5.59
	Stepwise	69522036.39	6508.78	70.64	6.37
Case2	ALL	235274484.63	12234.65	196.20	10.98
	CNN	102932088.92	7387.83	89.11	6.41
	PCA	112151170.90	7928.87	61.80	5.61
	Stepwise	100732687.53	7706.94	70.16	5.94
Case3	ALL	209572612.43	11240.81	185.96	10.34
	CNN	146624745.73	8348.74	73.97	6.34
	PCA	85459823.23	7209.39	61.71	5.54
	Stepwise	132391353.10	9153.32	62.14	5.55
Case4	ALL	101334616.91	7704.53	196.35	9.09
	CNN	89777854.31	7157.76	57.30	5.44
	PCA	95451414.93	7343.86	76.58	6.26
	Stepwise	99986487.82	8020.50	290.79	14.25
Case5	ALL	194763080.94	11176.21	74.62	6.32
	CNN	72228693.25	6608.43	57.45	5.60
	PCA	84800777.25	7070.55	73.70	6.11
	Stepwise	153064745.63	9825.17	124.00	7.45
Case6	ALL	178278551.87	10784.71	150.96	8.91
	CNN	96599293.36	7557.79	80.69	6.48
	PCA	77011497.56	6974.92	74.58	6.15
	Stepwise	106097652.53	7605.09	143.51	8.70

# 모델링

## LSTM



```
lstm = Sequential()  
lstm.add(LSTM(64, activation='tanh', input_shape=(X_train.shape[1],X_train.shape[2]),return_sequences = True))  
lstm.add(LSTM(32,activation='tanh',return_sequences=False))  
lstm.add(Dense(y_train.shape[1]))  
lstm.compile(optimizer='adam', loss='mse')
```

	특성추출	어획량		어획률	
		MSE	MAE	MSE	MAE
Case1	ALL	139996949.56	9487.18	71.75	6.49
	CNN	82383456.60	6446.18	55.74	5.55
	PCA	130048153.85	8249.90	119.93	6.33
	Stepwise	113614803.78	8416.49	62.82	6.06
Case2	ALL	229055329.63	11966.81	205.81	10.79
	CNN	70768677.66	6306.39	66.30	5.61
	PCA	128447169.77	8565.18	140.44	6.57
	Stepwise	91839099.06	7188.17	85.84	6.72
Case3	ALL	261507320.87	13065.28	162.22	9.24
	CNN	129628228.31	8060.92	57.06	5.41
	PCA	129309214.51	8511.56	66.00	5.82
	Stepwise	85762513.53	7183.40	76.61	6.81
Case4	ALL	91070986.84	7311.44	101.94	7.13
	CNN	69806018.25	6521.90	52.75	5.26
	PCA	76920860.21	6919.79	95.45	6.24
	Stepwise	99751142.76	7761.40	182.50	10.47
Case5	ALL	136216404.22	8891.70	189.30	8.21
	CNN	121844688.25	7495.04	83.98	5.88
	PCA	78612342.68	7103.93	63.97	5.73
	Stepwise	129218003.37	8711.13	165.16	8.68
Case6	ALL	116041996.73	8747.53	177.67	10.71
	CNN	114393894.12	7440.14	71.58	6.16
	PCA	89834962.27	7277.39	57.31	5.39
	Stepwise	162785372.88	9705.28	151.07	9.84

# 모델링

FNN



```
Fnn = Sequential()  
Fnn.add(Dense(128, activation='relu', input_dim=X_train_fnn.shape[1]))  
Fnn.add(Dense(64, activation = 'relu'))  
Fnn.add(Dense(32, activation = 'relu'))  
Fnn.add(Dense(y_train.shape[1]))  
Fnn.compile(optimizer='adam', loss='mse')
```

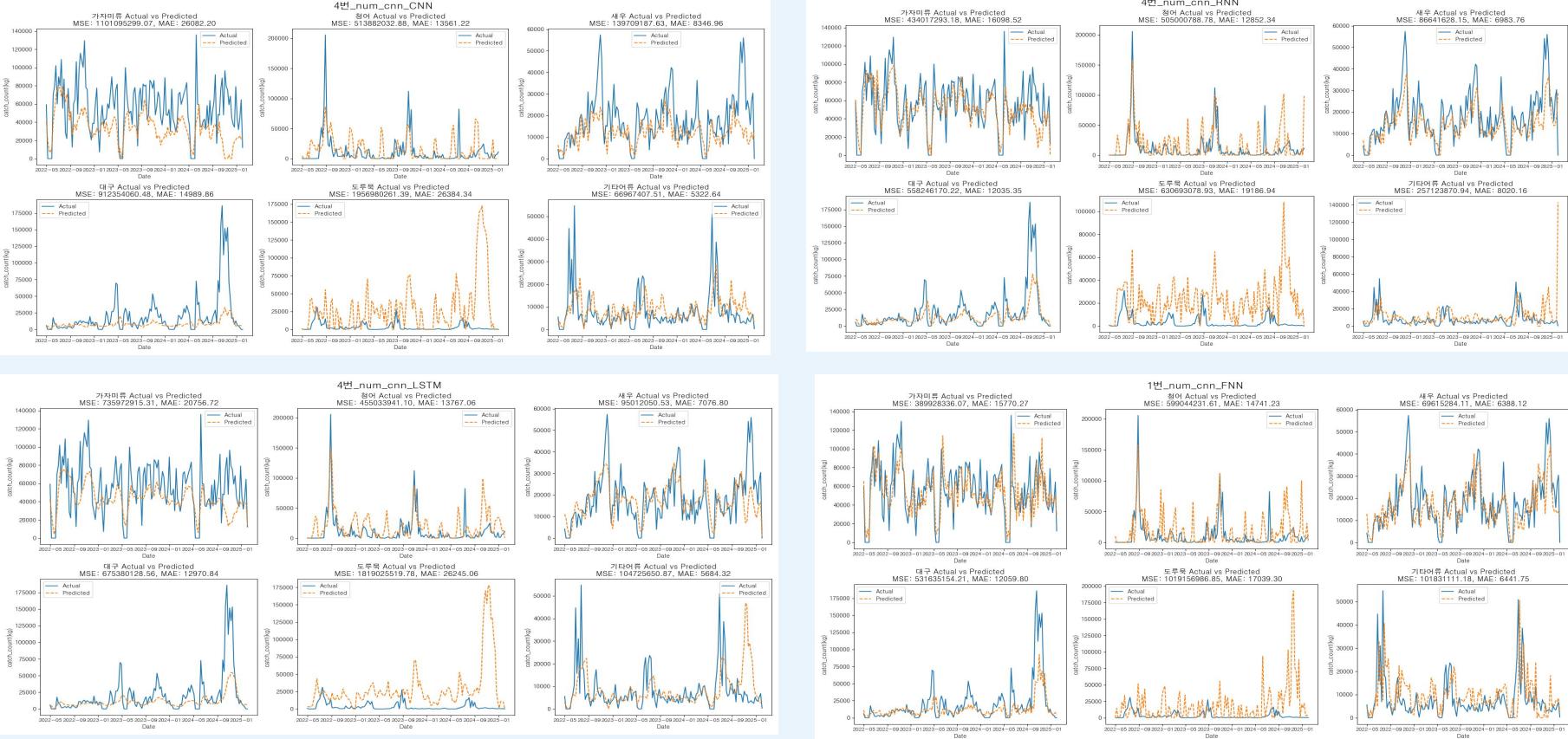


	특성추출	어획량		어획률	
		MSE	MAE	MSE	MAE
Case1	ALL	111191927.11	8362.12	101.47	7.99
	CNN	83571775.79	7093.29	65.77	5.66
	PCA	98431118.29	7689.61	59.89	5.61
	Stepwise	91653701.94	7483.79	67.26	6.01
Case2	ALL	229859609.32	12114.59	193.88	11.33
	CNN	101150217.42	7744.71	66.05	5.79
	PCA	80773656.49	7204.60	61.82	5.67
	Stepwise	91007422.54	7526.70	71.38	6.07
Case3	ALL	247100529.82	12062.01	201.04	11.47
	CNN	97699880.43	7483.35	64.06	5.67
	PCA	89121849.63	7435.71	61.83	5.74
	Stepwise	91548659.79	7528.83	78.13	6.60
Case4	ALL	93719233.28	7516.98	103.67	7.88
	CNN	84752905.58	7076.85	76.06	6.24
	PCA	90550864.23	7509.97	60.88	5.58
	Stepwise	123028781.30	8622.27	102.73	7.72
Case5	ALL	123606821.29	8601.29	103.99	7.93
	CNN	91928631.62	7355.06	83.49	6.67
	PCA	91427420.74	7525.73	73.00	6.22
	Stepwise	204720804.79	11163.74	147.76	9.49
Case6	ALL	126508978.20	8833.26	195.17	11.19
	CNN	96872652.65	7450.76	64.14	5.69
	PCA	93758237.31	7588.37	67.85	5.98
	Stepwise	148106006.10	9864.16	104.89	7.76

# 모델링

특성추출	CNN	모델	MSE	MAE	어획률	모델	MSE	MAE
		CNN	99,172,094	7,551.86		CNN	91.39	6.94
		RNN	89,777,854	7,157.76		RNN	57.30	5.44
		FNN	84,752,905	7,076.85		FNN	76.06	6.24
		LSTM	<b>69,806,018</b>	<b>6,521.90</b>		LSTM	<b>52.75</b>	<b>5.26</b>
Step	FNN	CNN	249,696,292	11,736.55	어획률	CNN	202.78	11.13
		RNN	99,986,487	8,020.50		RNN	290.79	14.25
		FNN	123,028,781	8,622.27		FNN	<b>102.73</b>	<b>7.72</b>
		LSTM	<b>99,751,142</b>	<b>7,761.40</b>		LSTM	182.50	10.47
		CNN	100,080,994	8,081.44		CNN	91.29	7.29
Total		RNN	95,451,414	7,343.86	어획률	RNN	76.58	6.26
		FNN	90,550,864	7,509.97		FNN	<b>60.88</b>	<b>5.58</b>
		LSTM	<b>76,920,860</b>	<b>6,919.79</b>		LSTM	95.45	6.24
		CNN	152,831,793	9,563.66		CNN	161.99	9.88
		RNN	101,334,616	7,704.53		RNN	196.35	9.09
		FNN	93,719,233	7,516.98	어획률	FNN	103.67	7.88
		LSTM	<b>91,070,986</b>	<b>7,311.44</b>		LSTM	<b>101.94</b>	<b>7.13</b>

# 데이터 선택



# 파이프라인

모델링-시각화 파이프라인 초기화

Args:  
data: 입력데이터프레임(ocean\_1\_rate,ocean\_1\_num...ocean\_6\_num)  
data\_info: 데이터프레임 이름(ocean\_1\_rate)  
target\_range: '축이어助长' 열을 타겟 설정 여부  
in\_n: 단계선택법에서 전진선택법의 일계값  
in\_out: 단계선택법에서 축진제거법의 일계값  
components: int or float  
PCA를 통한 특성 선택을 위한 차원 수  
int 선택 시 pca 차원수 상위 components개  
float 선택 시 분산의 components% 되도록 자동 생성  
count\_n: int  
선택된 특성의 개수 기준 절  
step: 슬라이딩 윈도우 실행 시 슬라이딩 개수  
types: 'rate' or 'num' 기본값은 'rate'  
선행방법은 데이터의 마지막 변수가 rate 0이면 'rate'  
그렇지 않으면 'num'  
is\_test: True or False 기본값은 False

steps : int 기본값 52  
(단기 예측 시 예측할 데이터 개수  
ex) 5

...  
self.data = data  
self.data\_info = data\_info  
self.target\_range = target\_range  
num = -2 if target\_range == 'False' else -1  
self.target = rate.columns[1:num].to\_list()  
self.in\_n = in\_n  
self.in\_out = in\_out  
self.components = components  
self.count\_n = count\_n  
self.step = step  
self.types = types  
self.setup = False  
self.is\_test = is\_test  
self.steps = steps  
  
self.cnn\_f = None  
self.step\_f = None  
self.pca\_f = None  
self.total\_f = None  
self.time\_d = None

```
_feature_select_extract(self):  
    ...  
    total_feature 함수를 사용하여 특성 선별 및 특성 추출 <- 도메인 선택 방법도 추가 예정  
  
이 함수는 cnn, pca를 통한 특징 추출, 단계선택법을 통한 축장 선별, 전체 데이터 및 시간 데이터를 생성합니다.  
  
시간 데이터는 이후 시각화에 사용이 됩니다.  
...  
  
print("특징 선별 추출 중 시작합니다.")  
start_time = time.time()  
self.cnn_f, self.step_f, self.pca_f, self.total_f, self.time_d = total_feature(data = self.data,  
target_data = self.target,  
in_n = self.in_n,  
in_out = self.in_out,  
components = self.components,  
count_n = self.count_n)  
  
end_time = time.time()  
total_time = round(end_time - start_time)  
  
print(f"총 소요시간 {total_time}초 입니다. \n특징 선별 추출 완료되었습니다..")
```

## 특징 추출 코드

```
def _New_data(self):  
    ...  
    # 원 데이터를 단계별로 예측 이후 환경데이터를 통하여 원래 예측  
    ...  
  
    self.data = New_data(self.data, self.step, self.steps, self.target)
```

## 예측 데이터 코드

## 초기화 코드

```
_NN_Visual(self,data_type_name="cnn"):  
    ...  
    특성을 이용하여 각각의 MNIST들을 이용하여 단순 소음과 다른 소음을 찾을 및 찾는 평가  
    ...  
  
    data_type = None  
    if data_type_name == 'cnn':  
        data_type = self.cnn_f  
    elif data_type_name == 'step':  
        data_type = self.step_f  
    elif data_type_name == 'pca':  
        data_type = self.pca_f  
    elif data_type_name == 'total':  
        data_type = self.total_f  
    else:  
        print(f"data_type_name이 잘못된 인자입니다. In 'cnn', 'step', 'pca', 'total' 중 하나를 사용하세요.")  
        return  
  
    if data_type is None:  
        print("특징 데이터가 추출되지 않았습니다. 먼저 feature_selection()을 호출하고 해당 특징이 생성되는지  
        확인해주세요.")  
        return  
  
    plot_title = f'(self.data_info)[data_type_name]'  
    print(f"\n\nNN 모델을 시작합니다.")  
    start_time = time.time()  
  
    cnn=cnn_plot(data = data_type,  
                step = self.step,  
                target_name = self.target,  
                plot_name = plot_title,  
                types = self.types,  
                time_data = self.time_d,  
                target_range = self.target_range,  
                is_test=self.is_test,steps=self.steps)  
  
    end_time = time.time()  
    total_time = round(end_time - start_time)  
  
    print(f"\n\n{total_time}초 소요시간 {total_time}초입니다. \nNN 모델을 시작한 원인!")  
  
    print(f"\n\nML 모델을 시작합니다.")  
    start_time = time.time()  
  
    rnn = Rnn_plot(data = data_type,  
                  step = self.step,  
                  target_name = self.target,  
                  plot_name = plot_title,  
                  types = self.types,  
                  time_data = self.time_d,  
                  target_range = self.target_range,  
                  is_test=self.is_test,steps=self.steps)  
  
    end_time = time.time()  
    total_time = round(end_time - start_time)  
  
    print(f"\n\n{total_time}초 소요시간 {total_time}초입니다. \nML 모델을 시작한 원인!")  
  
    print(f"\n\nNN,ML 모델을 시작합니다.")
```

```
NN_run(self,data_type_name="cnn"):  
    if not self.setup:  
        fold_name = f'{self.types}_plot'  
        os.makedirs(fold_name,exist_ok=True)  
        self._feature_select_extract()  
        self.setup = True  
    else:  
        print("특징 추출 이미 진행하였습니다.")  
    result_df = self._NN_Visual(data_type_name)  
    return result_df
```

```
def ML_run(self,data_type_name="cnn"):  
    if not self.setup:  
        fold_name = f'{self.types}_plot'  
        os.makedirs(fold_name,exist_ok=True)  
        self._feature_select_extract()  
        self.setup = True  
    else:  
        print("특징 추출 이미 진행하였습니다.")  
    result_df = self._ML_Visual(data_type_name)  
    return result_df
```

```
#####
#      is_test = True인 경우 아래 함수 사용 가능      #
#####
```

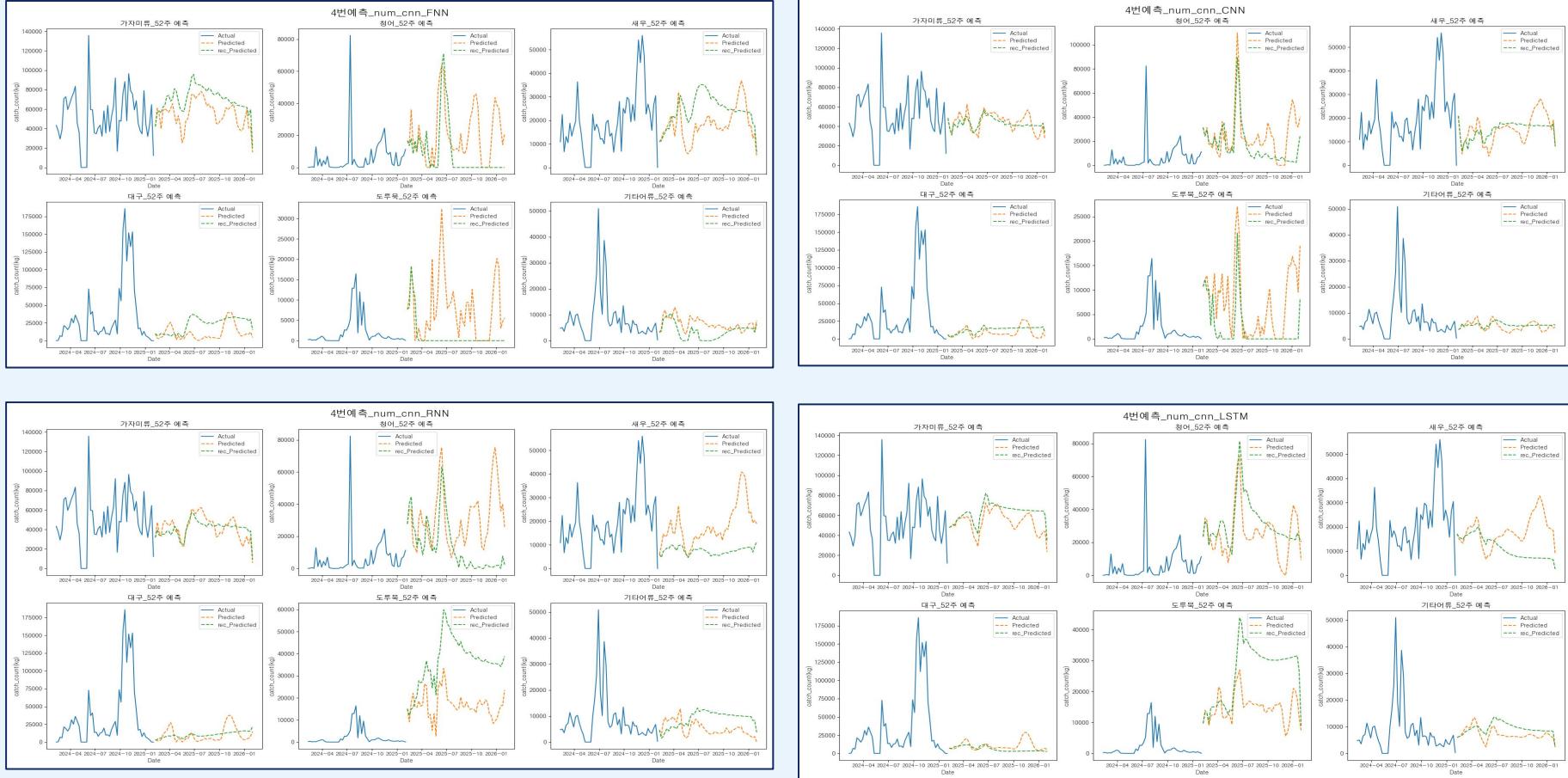
```
def predict_run(self,data_type_name="cnn"):  
    if not self.setup:  
        fold_name = f'{self.types}_plot'  
        os.makedirs(fold_name,exist_ok=True)  
        self._New_data()  
        self._feature_select_extract()  
        self.setup = True  
    else:  
        print("특징 추출 이미 진행하였습니다.")  
    NN_df = self._NN_Visual(data_type_name)  
    ML_df = self._ML_Visual(data_type_name)  
    return NN_df,ML_df
```

## NN 코드

## 실행 코드

# 최종예측

## 어획량

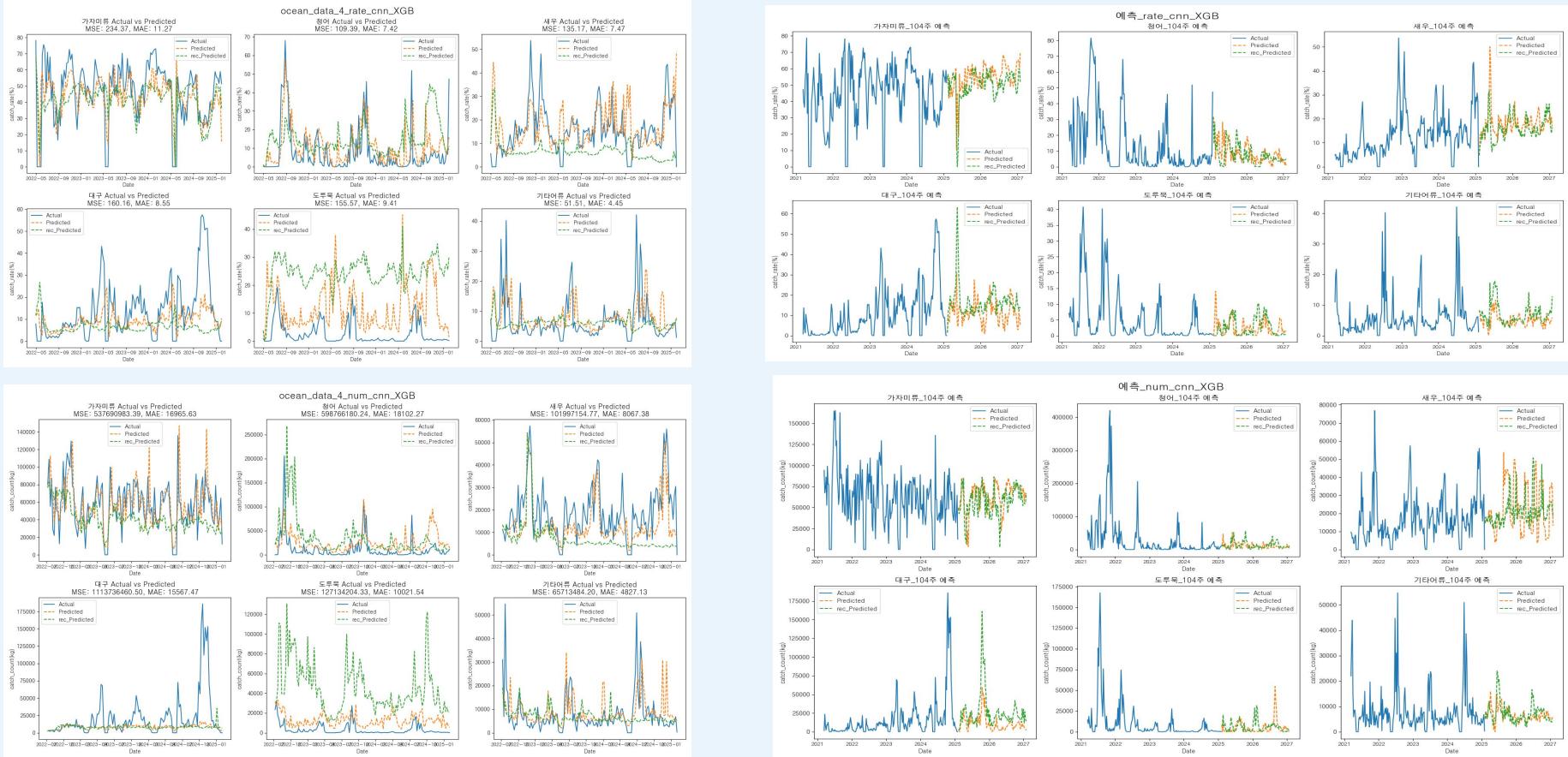


# 최종예측

## 어획률



# 추가 진행 (ML)



## 모델 성능의 한계점 분석

- 예상치 못한 어획량  
변동

연합뉴스 · www.yna.co.kr > 연합뉴스 > 최신뉴스  
그 많던 울산 가자미 어디에...기상악화·고수온에 어획량 '뚝'  
2024.03.01. (울산=연합뉴스) 장지현 기자 = "다음 세대에서는 가자미 보기가 어렵지 않을까 싶습니다."

뉴스투데이 이준호  
씨 마른 도루묵..7년 사이 어획량 12분의 1로  
2016년 강원도의 도루묵 어획량은 7,497톤.  
하지만 바다의 고수온 현상과 과도한 남획으로 지난 해는 610톤, 7년 새 어획량이 12분의 1로 줄었습니다.

[네트워크R]생선 대구 어획량 3배 증가  
2025-03-17 | 김건업 | 754  
[ CG : 올해 2월 강원도의 대구 어획량은 158톤. 금어기를 포함해 조업일이 두 배가량 늘자, 어획량이 지난해의 3.4배로 늘어났습니다.]

매일경제  
국산 청어 과메기 뜬다...어획량 31.4% 확 늘어  
5일 통계청이 조사한 어업생산동향에 따르면 올해 상반기 청어 어획량은 2만831t으로 지난해 같은 기간의 1만5천849t보다 31.4% 급증했다.  
이는 최근 6년 사이 가장 많은 어획량이다.  
2017-11-05 08:06:17

## 모델 성능의 한계점 분석

- 인위적인 개체수 조절

2020.06.05

## 국내 최초 참가자미 대량 인공 부화…이달 방류

2016년 시험 연구에 착수한 지 5년 만으로, 이번에 생산된 참가자미는 이달 중 강릉시 연안 해역에 방류

강원도, 겨울 어종 '도루묵·뚝지' 치어 방류

KBS

네이버 뉴스 • 2025.04.03.



뉴스피크

수과원, 어린 도루묵 20만 마리 방류...모두 바다로 이동

# 모델 성능의 한계점 분석

## - 데이터 부족 가능성?

“Neural networks tend to overfit when the number of parameters exceeds the number of samples — especially in small datasets.” — Zhang et al., Understanding Deep Learning Requires Rethinking Generalization, ICLR 2017 → 여기서 말하는 샘플은 X\_train 개수 현재 X\_train 개수:582개

Layer (type)	Output Shape	Param #
simple_rnn_220 (SimpleRNN)	(None, 16)	624
dense_500 (Dense)	(None, 6)	102

Layer (type)	Output Shape	Param #
conv1d_319 (Conv1D)	(None, 3, 16)	720
dropout_24 (Dropout)	(None, 3, 16)	0
max_pooling1d_19 (MaxPooling1D)	(None, 1, 16)	0
flatten_19 (Flatten)	(None, 16)	0
dense_499 (Dense)	(None, 6)	102

# Q & A