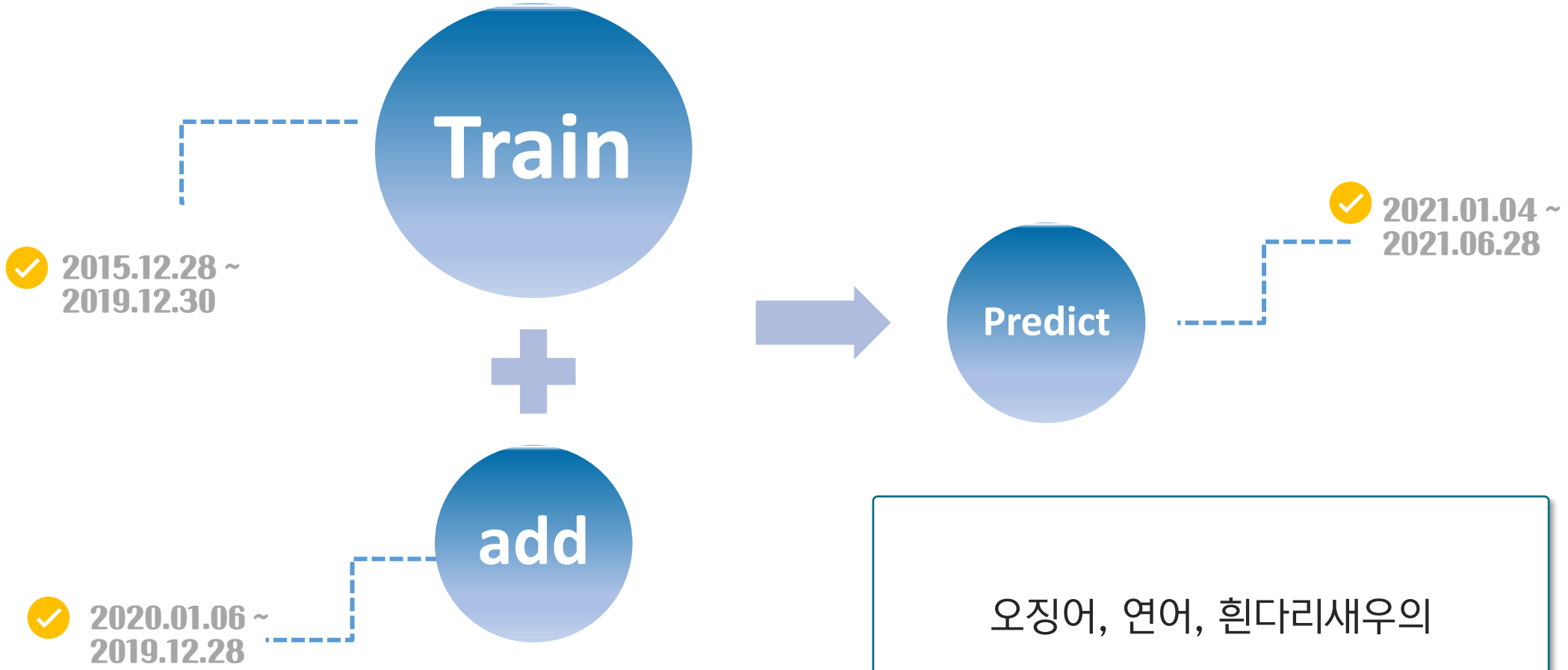
2021 BIG CONTEST CHAMPION LEAGUE

수산물 수입가격 예측 모델 구축

Team _ 코지모임 

☑ 조희승 (팀장) moohan132435@nate.com
☑ 김혜린 k1h2fls@naver.com

프로젝트 개요



오징어, 연어, 흰다리새우의

평균 단가($) 예측

1. 데이터 분석 배경

세계 해양수산업의 부가가치

2010년 약 1.5조 달러

2배

2030년 약 3조 달러

경제협력개발기구(OECD, 2016) “Ocean Economy in 2030”

1. 데이터 분석 배경

수산업의 특징

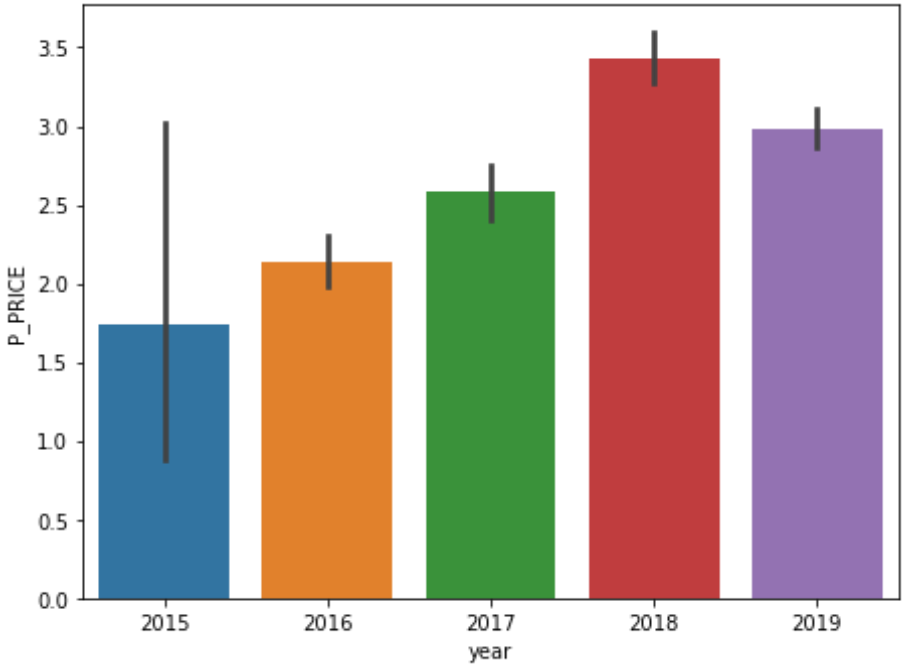
- 어획의 불확실성
- 수산물의 강부패성
- 수산자원 및 어장의 공유재산적 성격

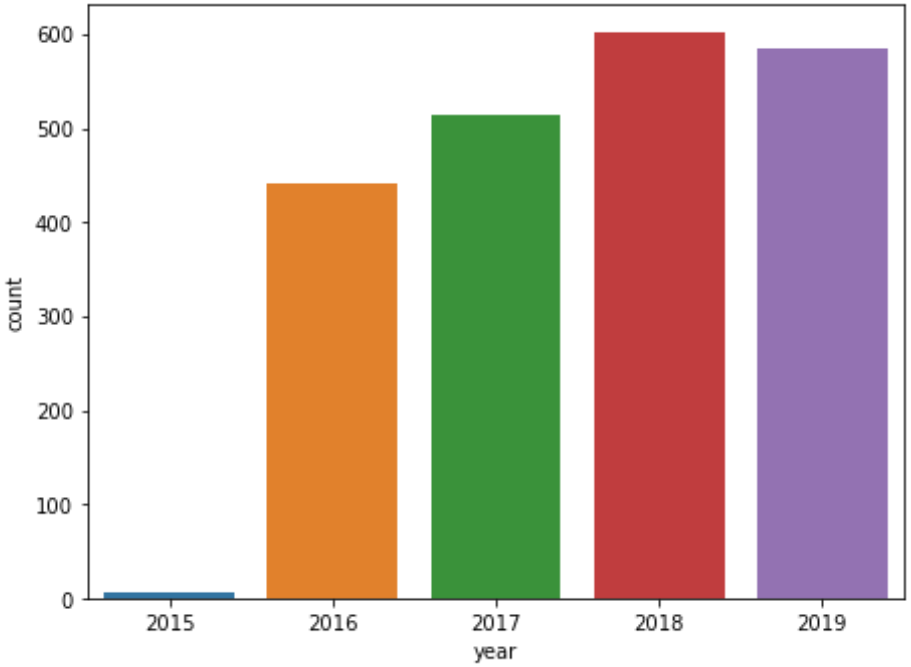
가격 불안정 초래

2. 데이터 탐색 - 오징어

수산물 가격분포와 생산량 확인

연도별 오징어의 가격 & 총 생산량



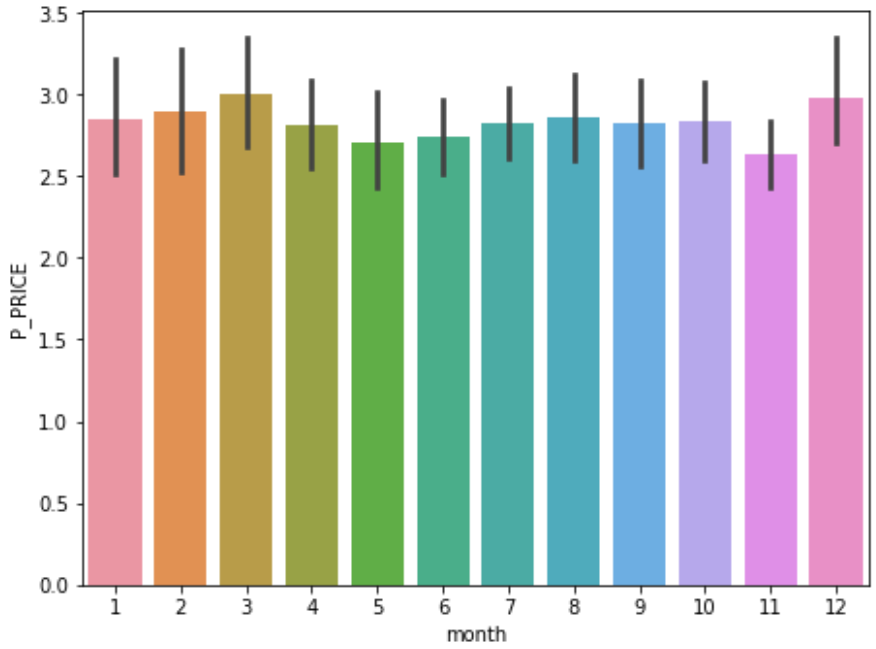


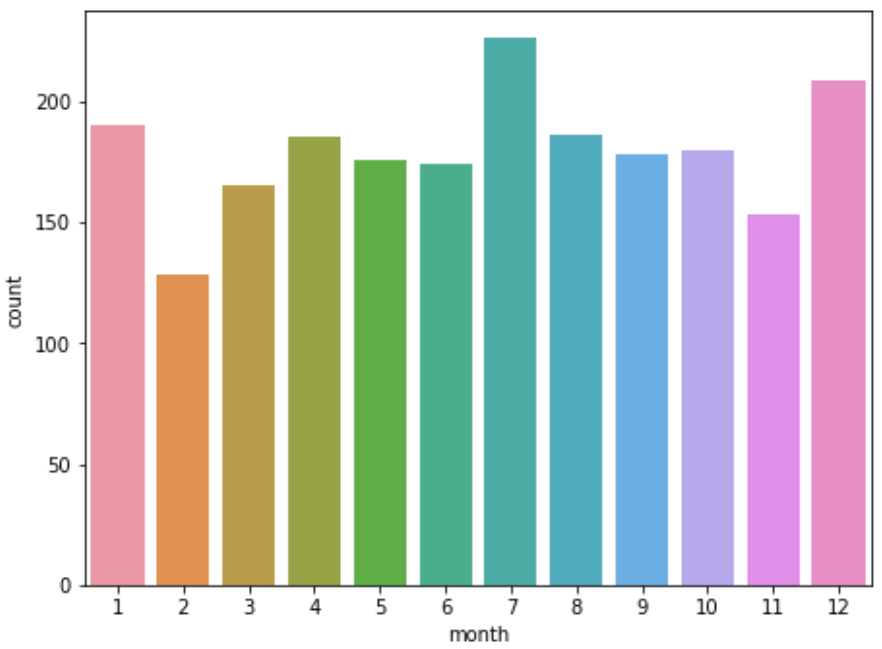
가격과 생산량 모두 **2018년**에 가장 높은 추세를 보임

2. 데이터 탐색

수산물 가격분포와 생산량 확인

월별 오징어의 가격 & 총 생산량



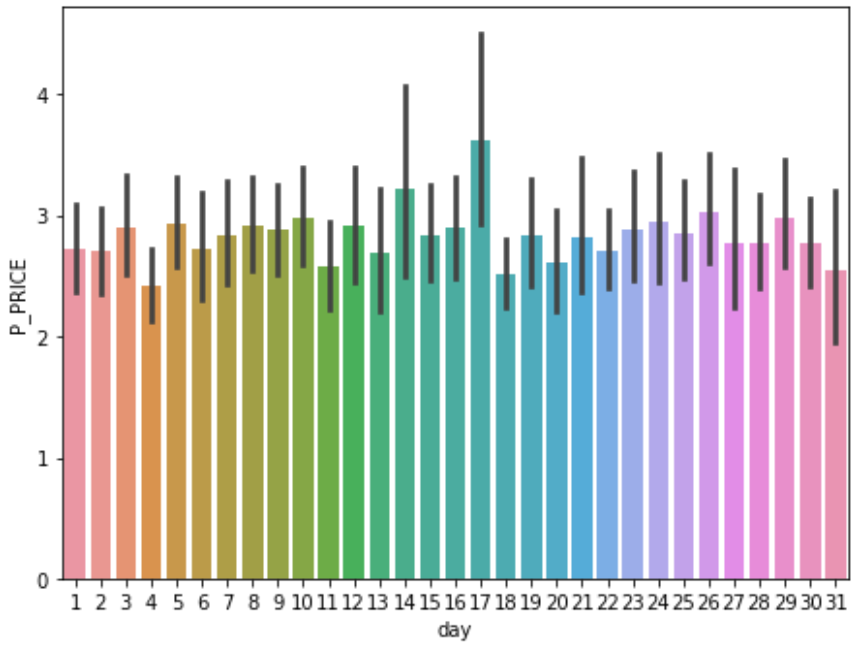


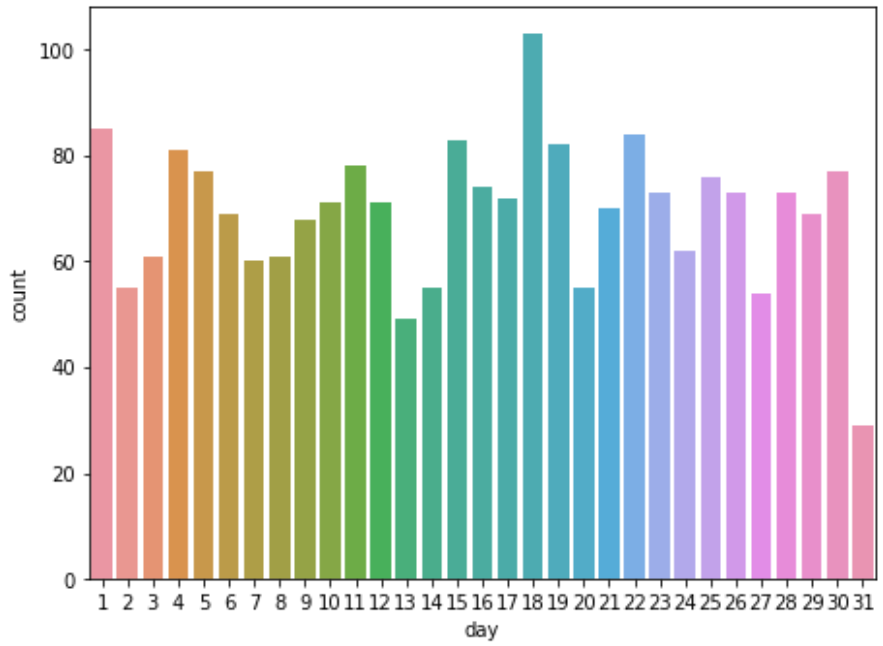
가격은 전체적으로 비슷한 수준이나
생산량은 2월에 가장 적고 7월에 가장 많음

2. 데이터 탐색

수산물 가격분포와 생산량 확인

일별 오징어의 가격 & 총 생산량



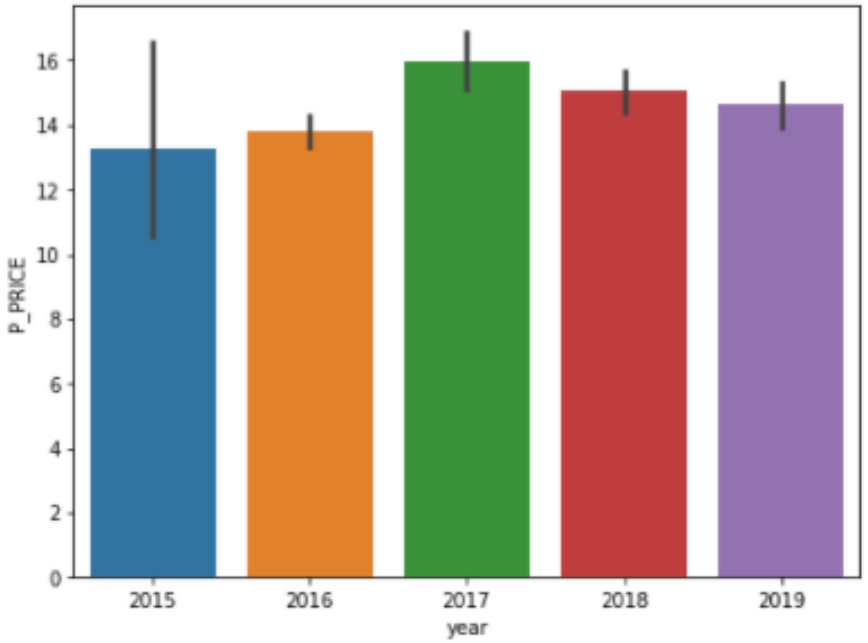


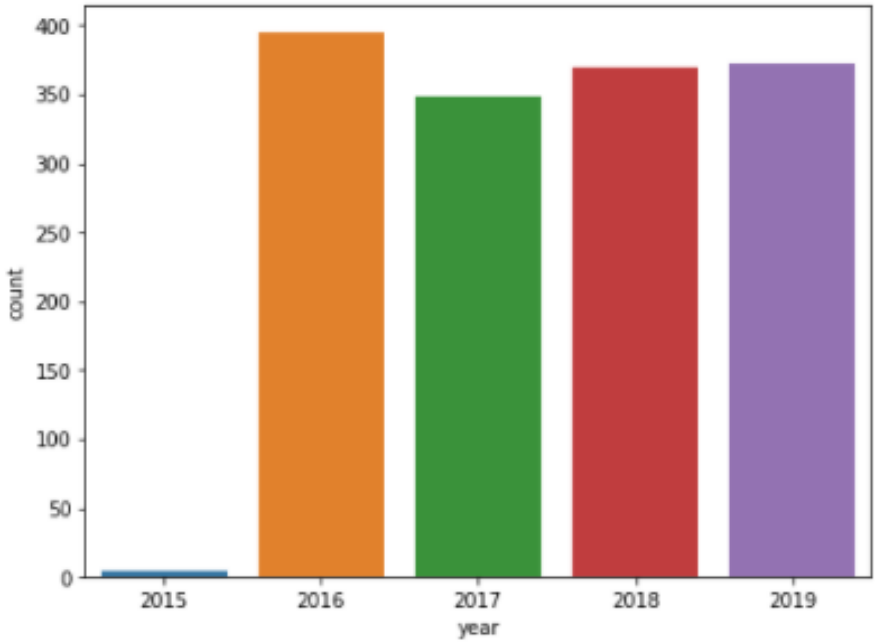
17일에 가격이 가장 높음
일별 총 생산량은 **상·하강 분포**를 보임

2. 데이터 탐색 - 연어

수산물 가격분포와 생산량 확인

연도별 연어의 가격 & 총 생산량



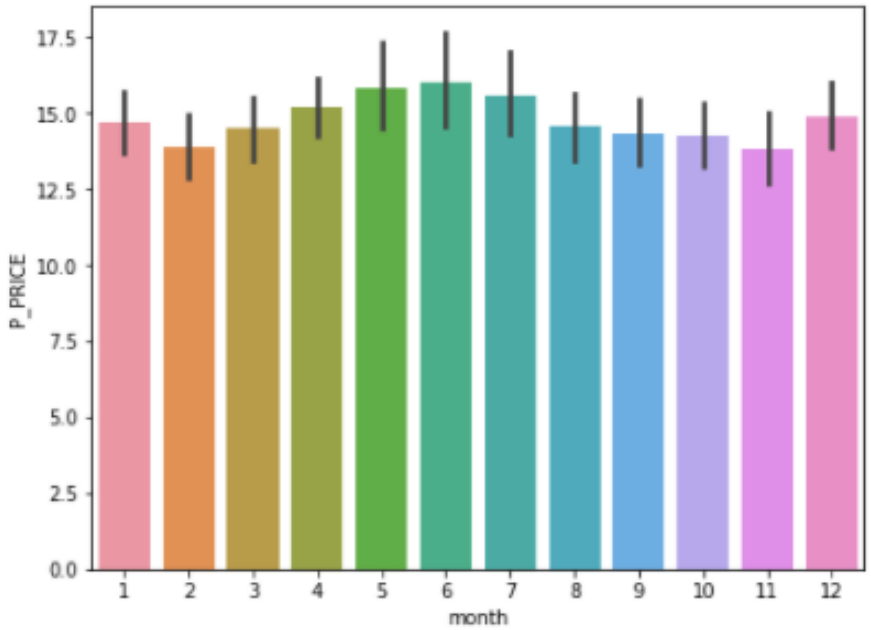


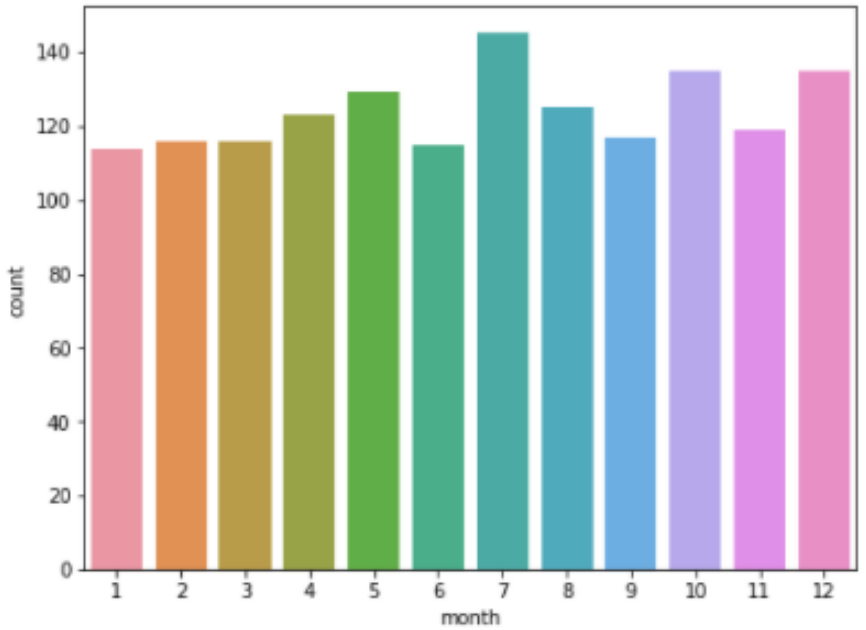
2015년은 12월 자료만 존재
2017년에 가격이 가장 높으며 2016년에 생산량이 가장 많음

2. 데이터 탐색

수산물 가격분포와 생산량 확인

월별 연어의 가격 & 총 생산량



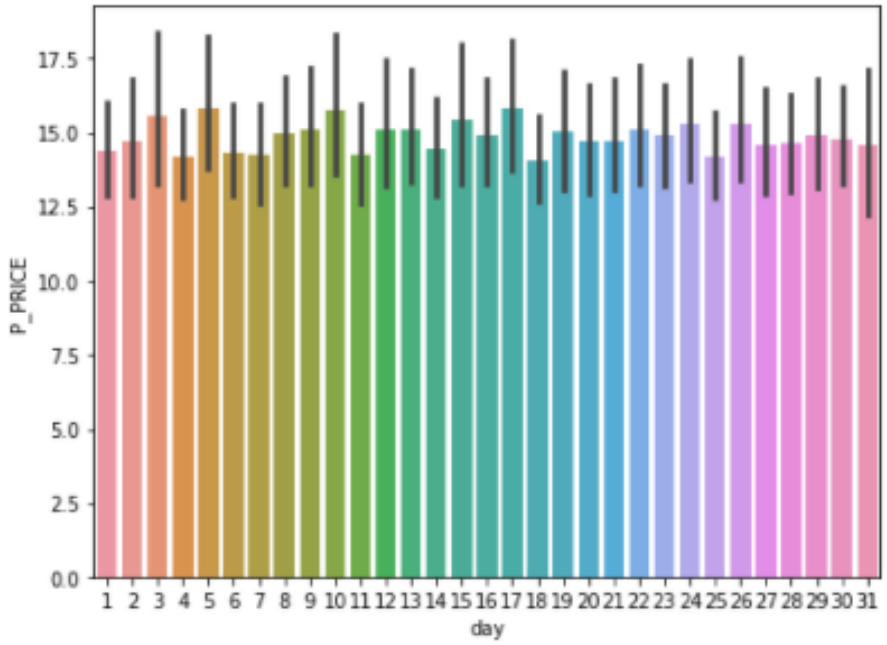


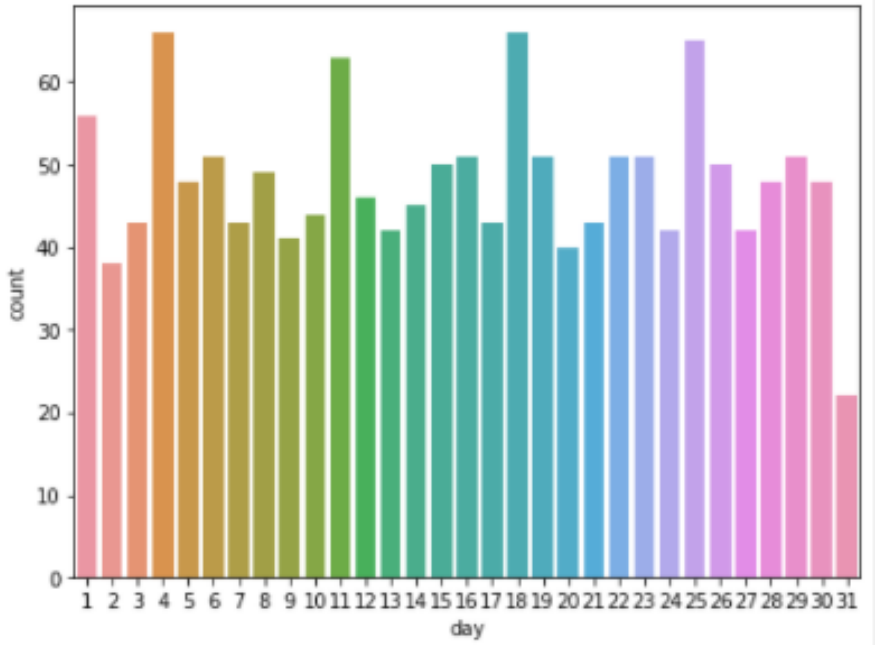
가격이 분기별로 상・하강을 보임
생산량은 7월에 가장 많음

2. 데이터 탐색

수산물 가격분포와 생산량 확인

일별 연어의 가격 & 총 생산량



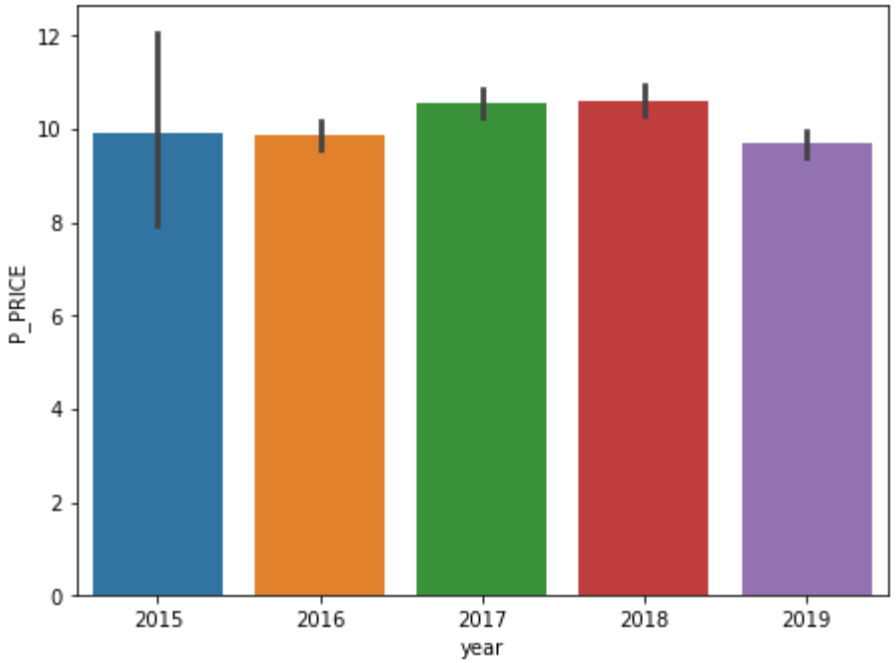


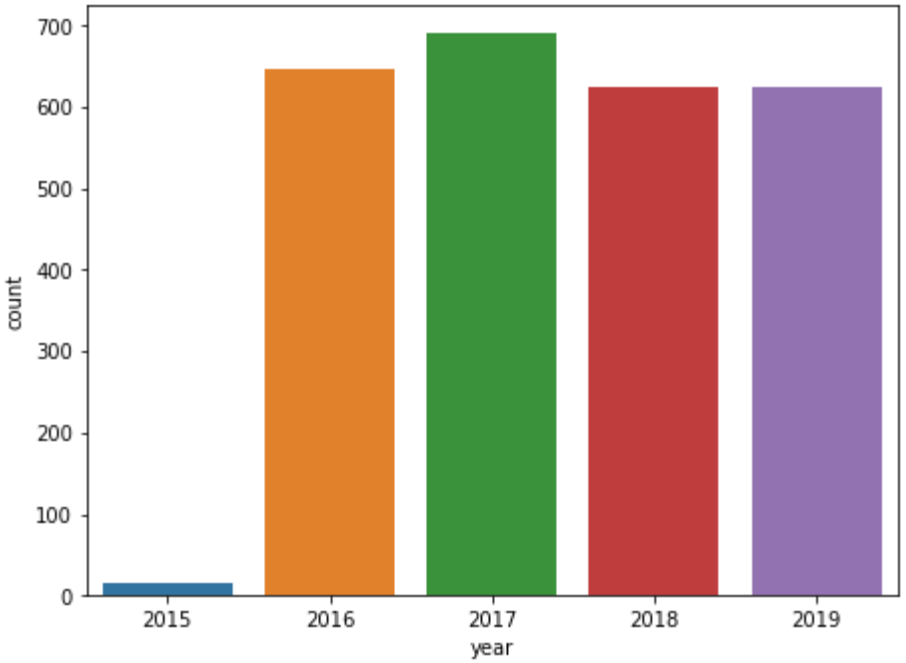
일별 가격과 생산량 모두 **상·하강 분포**를 보임

2. 데이터 탐색 - 흰다리새우

수산물 가격분포와 생산량 확인

연도별 흰다리새우의 가격 & 총 생산량



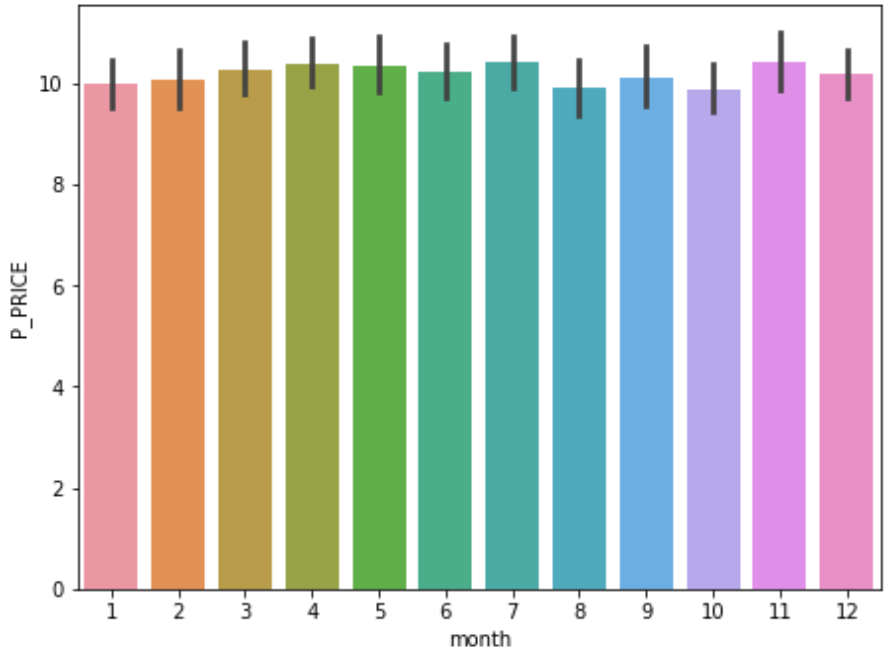


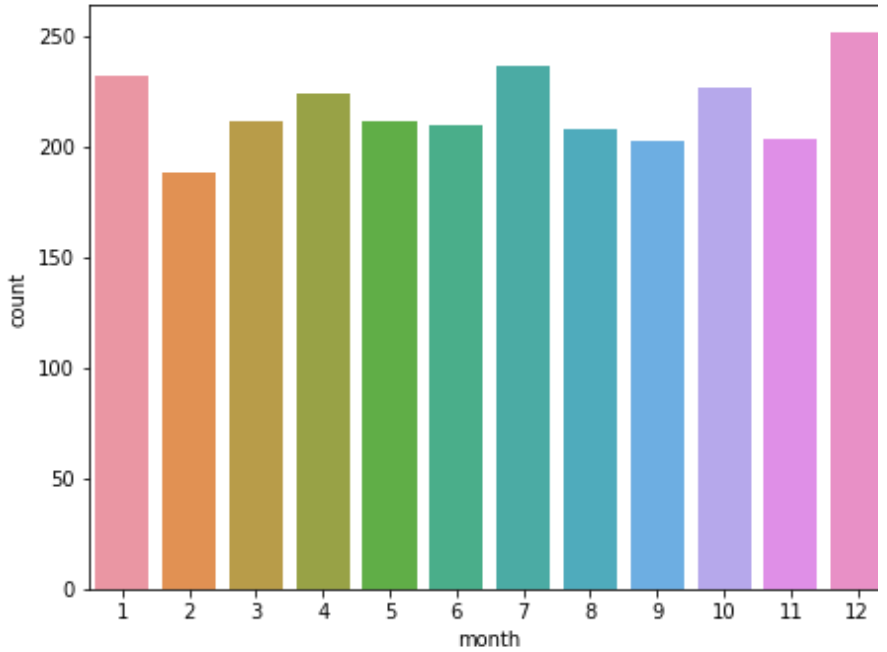
가격과 생산량 모두 전체적으로 비슷한 수준을 보임

2. 데이터 탐색

수산물 가격분포와 생산량 확인

월별 흰다리 새우의 가격 & 총 생산량



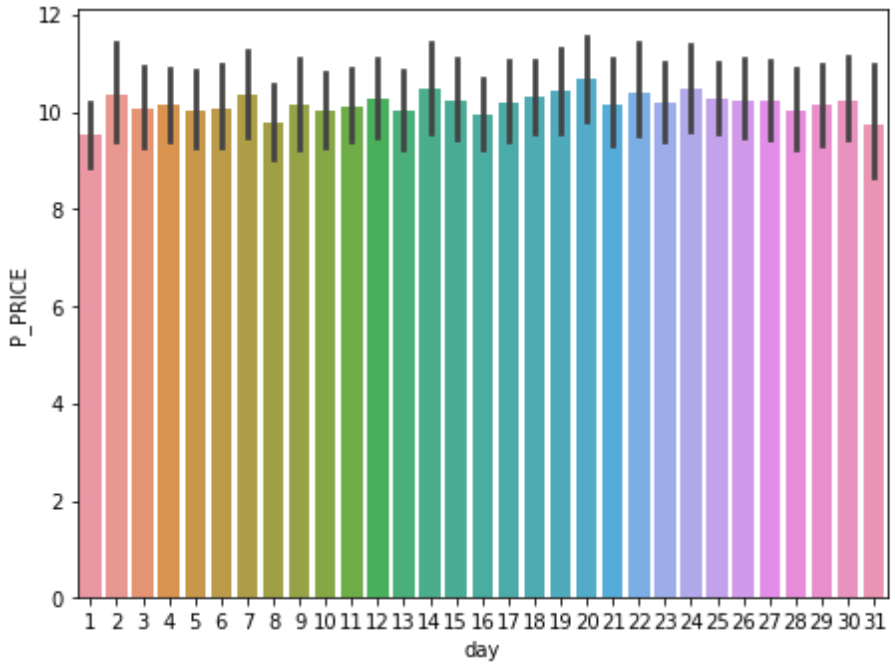


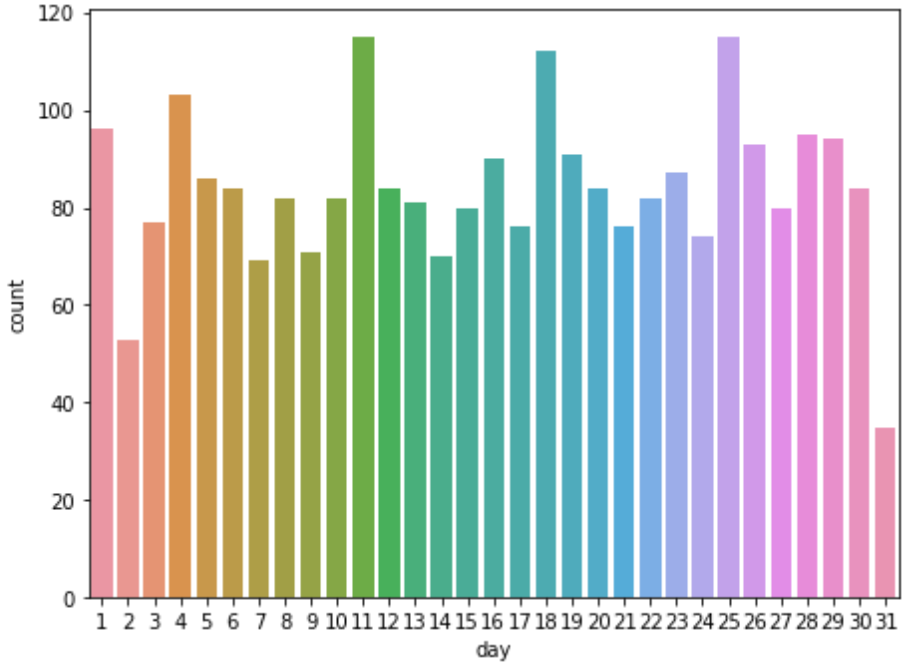
가격과 생산량 모두 전체적으로 비슷한 수준을 보임

2. 데이터 탐색

수산물 가격분포와 생산량 확인

일별 흰다리 새우의 가격 & 총 생산량





일별 가격의 차이는 미미하나
일별 총 생산량은 일별 총 생산량은 **상·하강 분포**를 보임

2. 데이터 검증

품목별 분류 작업

각 품목별 'REG_DATE'로 주차별 평균 가격으로 작업을 진행함

```
# 시계열분석 모델용 Data Preparation
def prepare_df_arima(df, subject):
    # 품목별 저장
    df = train.loc[train['P_NAME'] == subject]
    arima_df = df.groupby('REG_DATE').mean()

    return arima_df
```

```
prepare_df_arima(train, '연어').head()
```

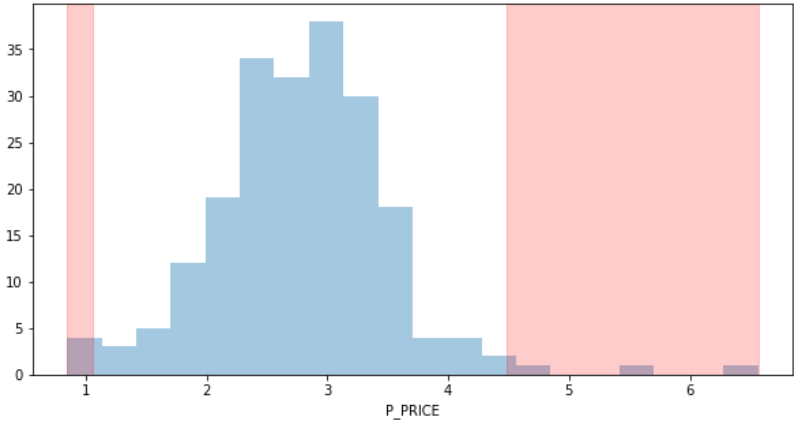
REG_DATE	P_PRICE
2015-12-28	13.287212
2016-01-04	12.331994
2016-01-11	12.876513

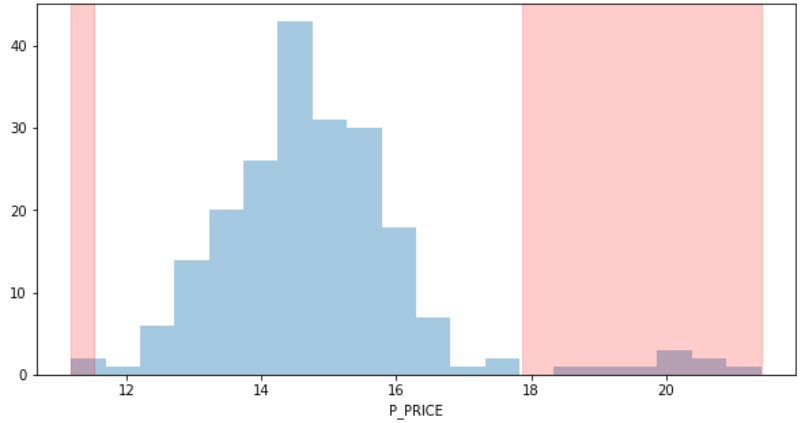
2. 데이터 검증 - 이상치

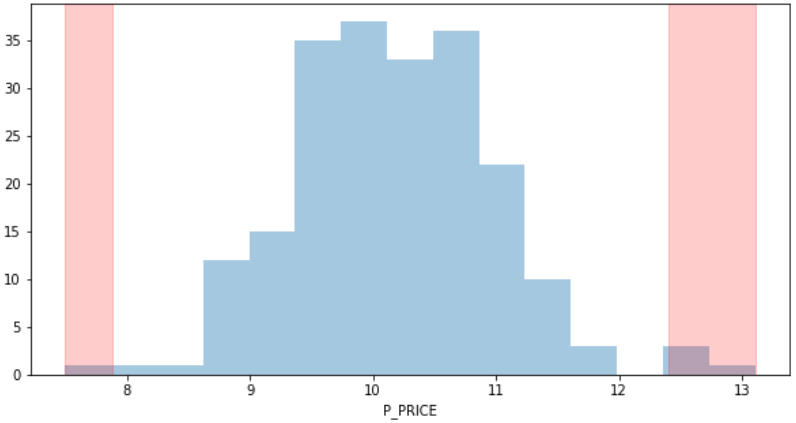
이상치 파악

품목별 이상치를 파악하여 처리 고민

오징어, 연어, 흰다리새우 순







빨간박스안에 품목별 빨간 박스안에 이상치가 보이긴 하나,
소비자물가지수(e나라지표), 어획량 변동에 따라 가격 변동이 있을 것이라 판단함

- 따라서 **이상치를 제거하지 않기**로 결정

2. 데이터 검증

데이터 Stationary 검증 ①

ADF test (통계적 판단)

```python
# 정상성 검정 adf
from statsmodels.tsa.stattools import adfuller
def ad_test(dataset):
    dftest = adfuller(dataset, autolag = 'AIC')
    print("1. ADF : ",dftest[0])
    print("2. P-Value : ", dftest[1])
    print("3. Num Of Lags : ", dftest[2])
    print("4. Num Of Observations Used For ADF Regression:",      dftest[3])
    print("5. Critical Values :")
    for key, val in dftest[4].items():
        print("\t",key, ": ", val)
```

오징어
P-Value : 0.1777

연어
P-Value : 0.0108

흰다리새우
P-Value : 0.0027

p- value가 0.05보다 작으면
귀무가설(단위근 존재) 기각

연어(0.01<0.05), 흰다리새우(0.002<0.05) 기각
오징어(0.177 > 0.05) 채택
연어, 흰다리 새우는 정상성, 오징어는 **비정상성**을 보임

2. 데이터 검증

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단) – 코드 및 방향

```python
def acf_pacf(data):
    fig, ax = plt.subplots(1,2, figsize=(10,5))
    fig.suptitle('Raw Data')
    sm.graphics.tsa.plot_acf(data.values.squeeze(), lags=40, ax=ax[0])
    sm.graphics.tsa.plot_pacf(data.values.squeeze(), lags=40, ax=ax[1])
```

```python
# 차분 진행 후 acf, pacf plot
def diff_acf_pacf(data):
    diff_data = data.copy()
    diff_data = diff_data.diff()
    diff_data = diff_data.dropna()

    fig, ax = plt.subplots(1,2, figsize=(10,5))
    fig.suptitle('Differenced Data')
    sm.graphics.tsa.plot_acf(diff_data.values.squeeze(), lags=40, ax=ax[0])
    sm.graphics.tsa.plot_pacf(diff_data.values.squeeze(), lags=40, ax=ax[1]);
```

: Raw Data와 1차 차분 Data의
ACF, PACF plot을 비교함

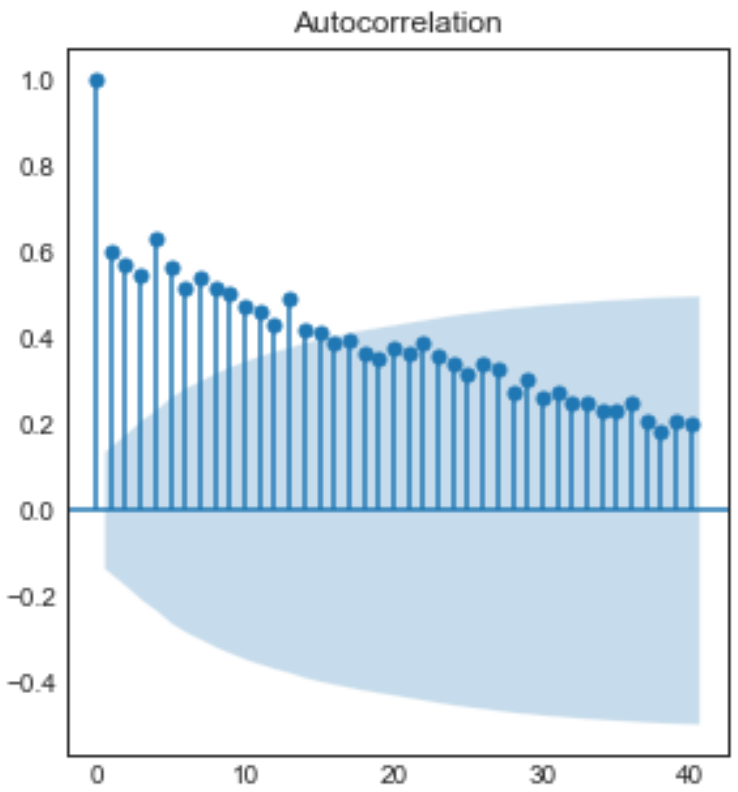
: 연어, 오징어, 흰다리 새우 모두 1차 차분 후
plot이 **정상적인** 형태를 보임

› 이후 ARIMA model에 차분 고려

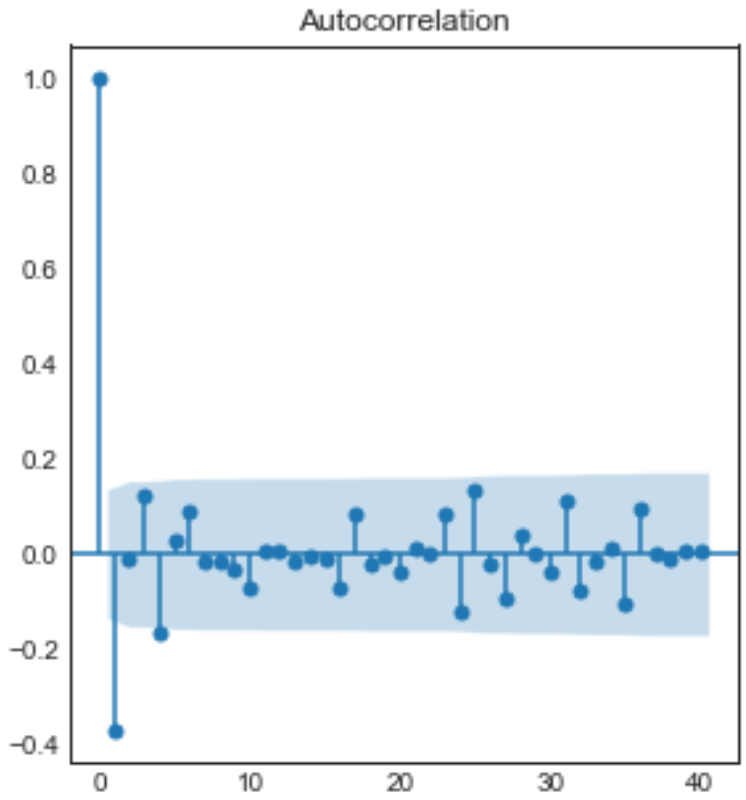
2. 데이터 검증 - 오징어

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



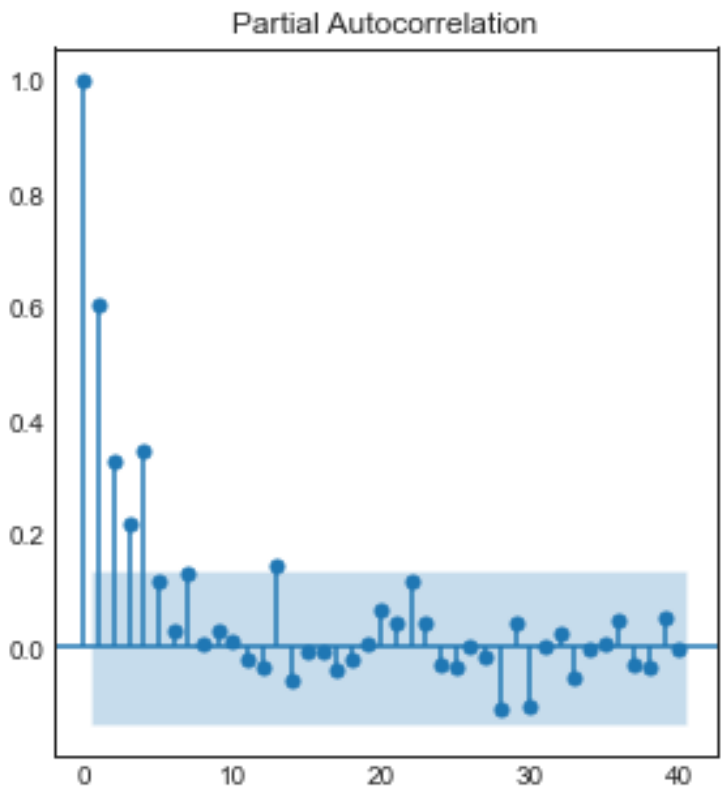
1차 차분



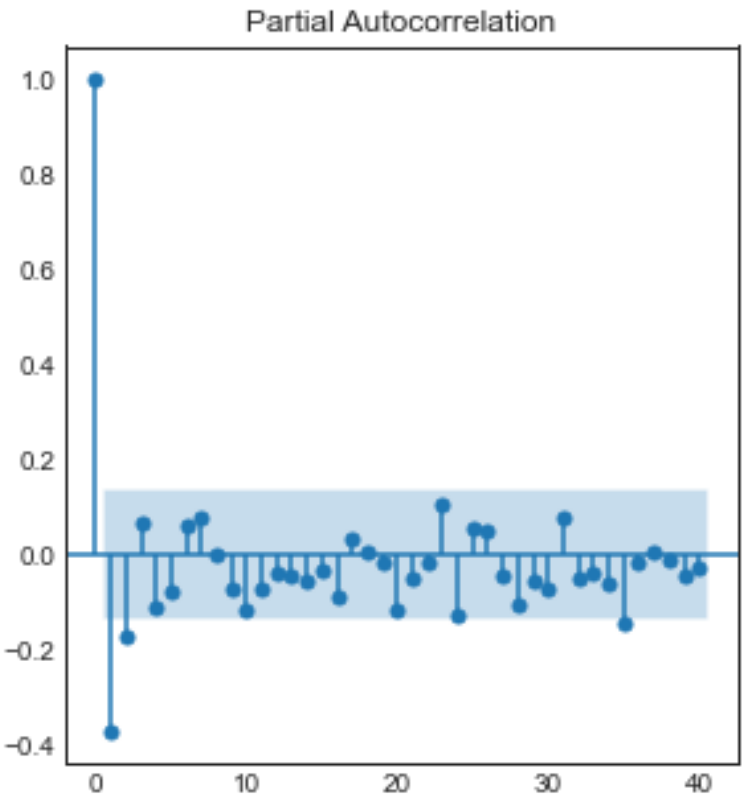
2. 데이터 검증

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



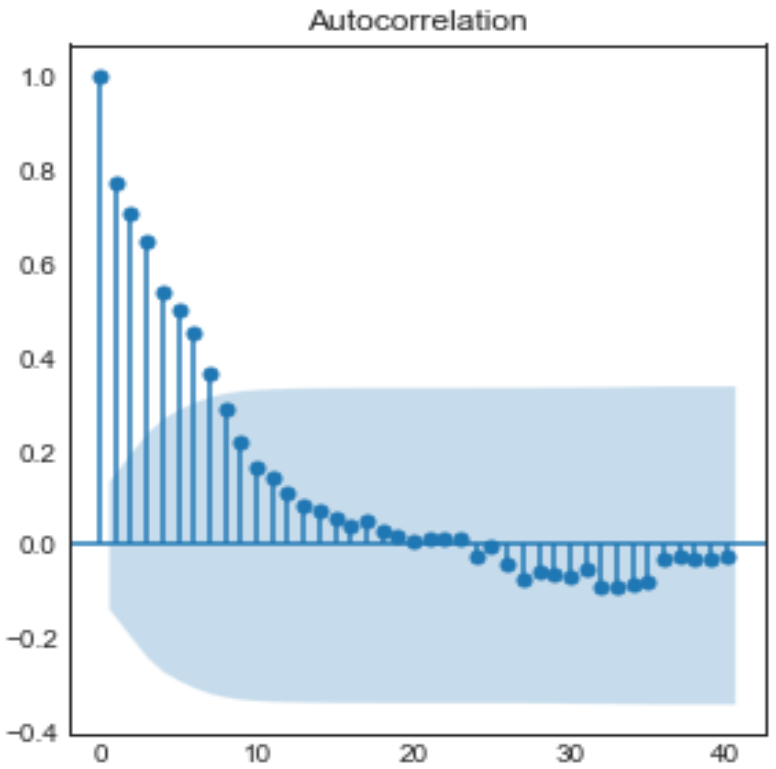
1차 차분



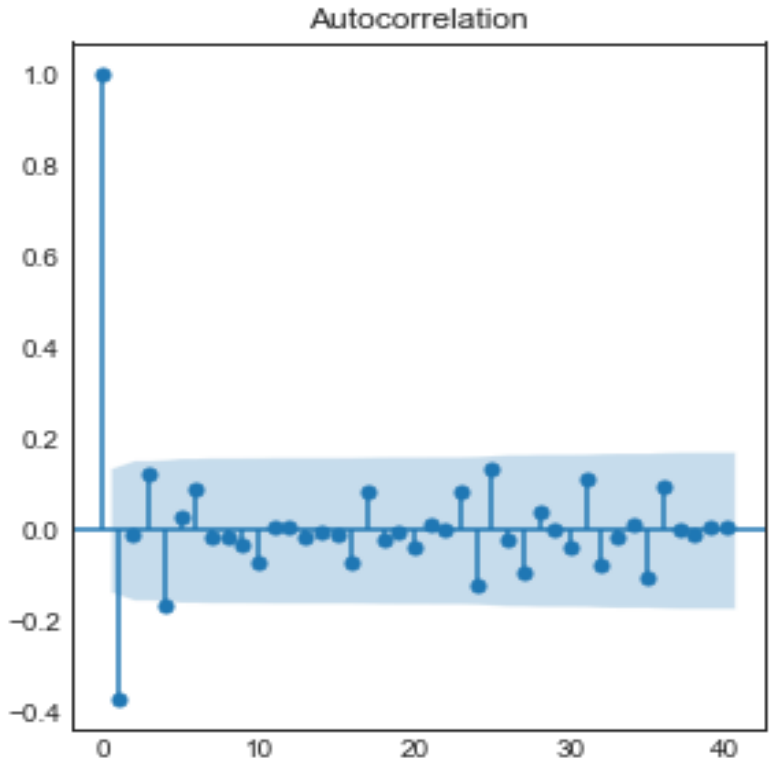
2. 데이터 검증 - 연어

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



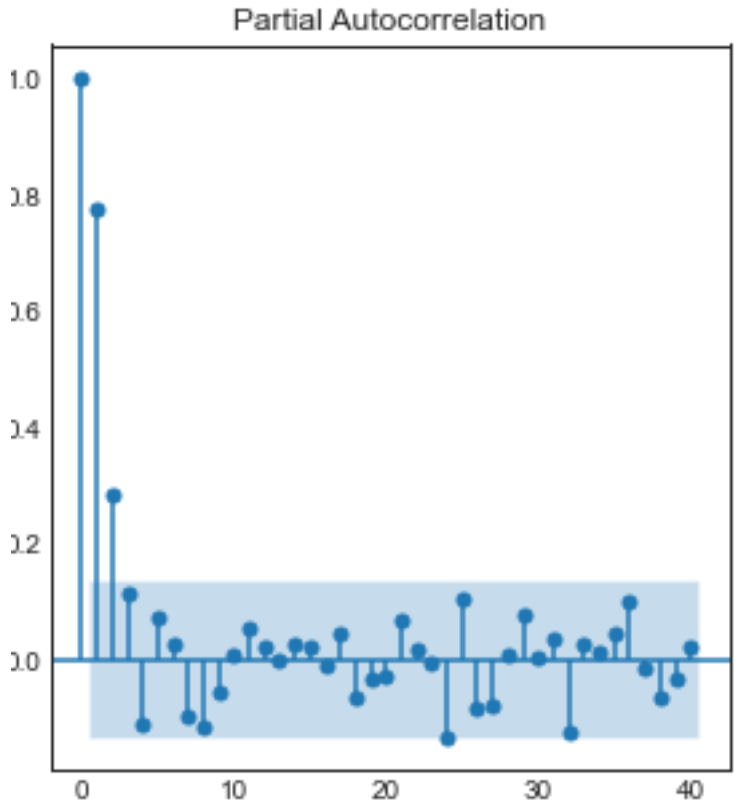
1차 차분



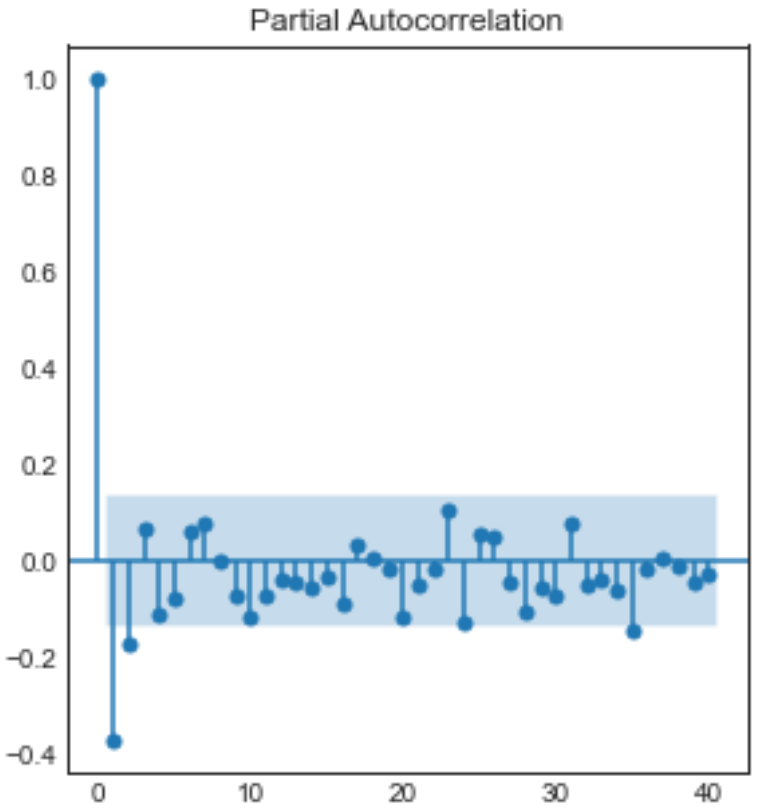
2. 데이터 검증

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



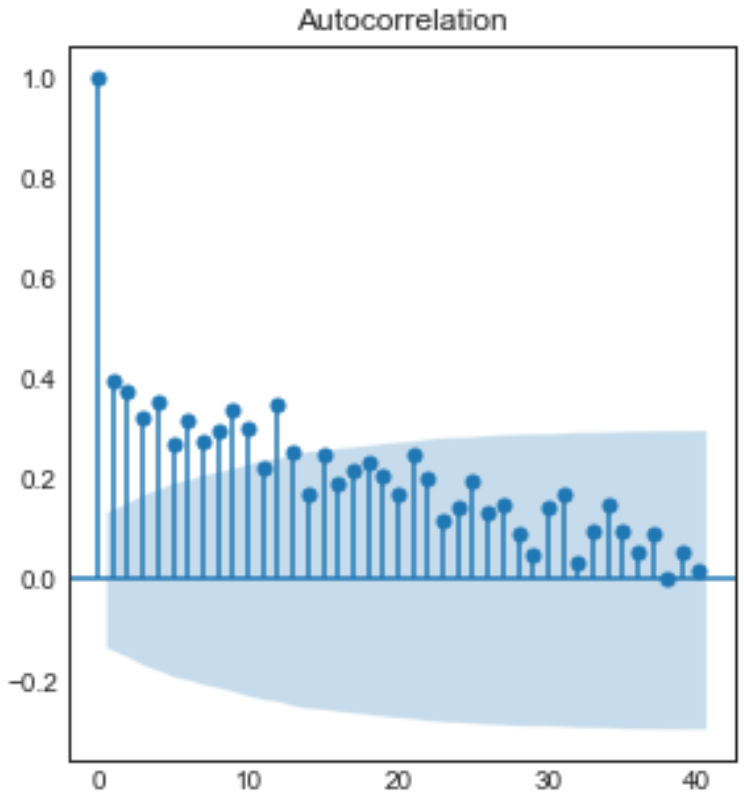
1차 차분



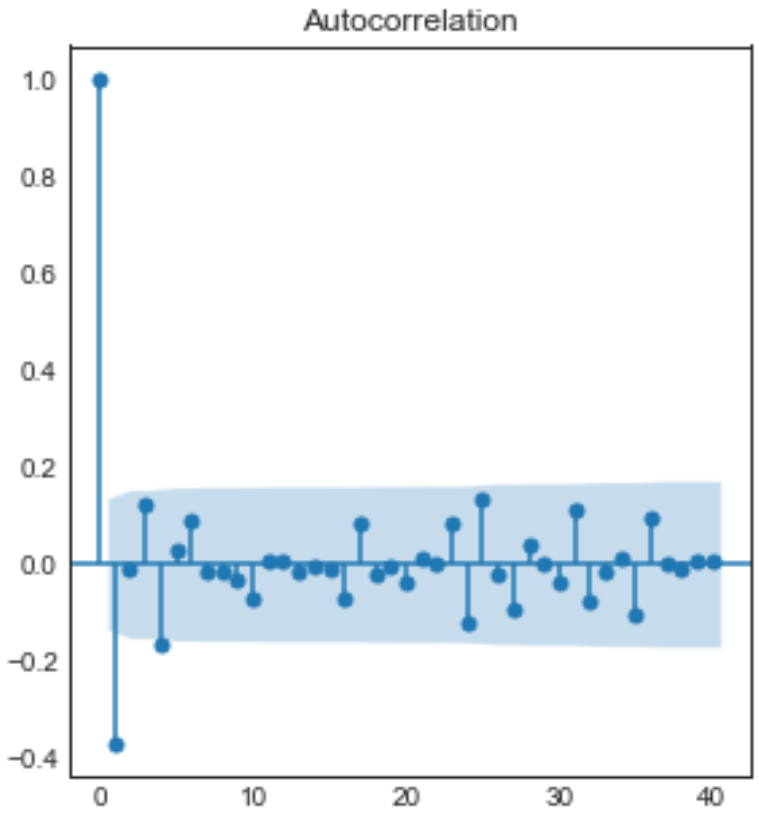
2. 데이터 검증 – 흰다리새우

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



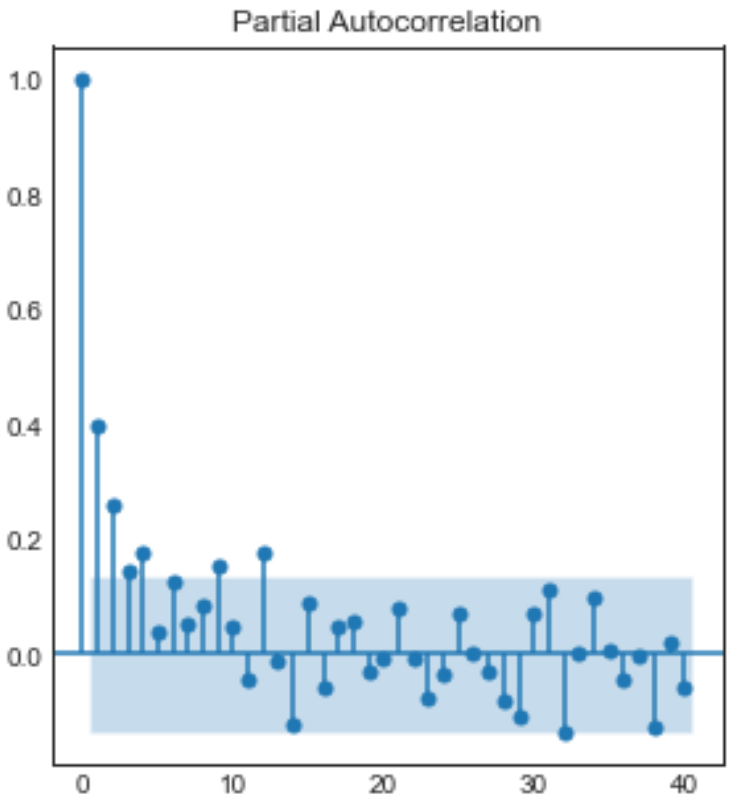
1차 차분



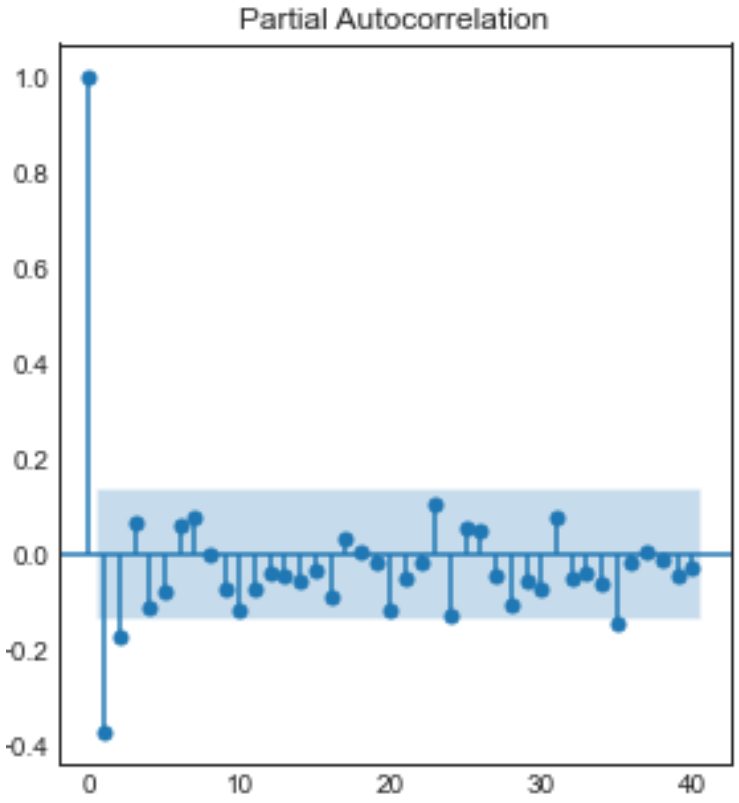
2. 데이터 검증

데이터 Stationary 검증 ②

ACF, PACF (정성적 판단)



1차 차분



3. 분석 세부 내용 – 시계열 모델

※ Arima Model

〈Modeling 과정〉

1) 데이터정상성 체크(In EDA) : adf, acf, pacf
2) 차분
3) validation sample 지정
4) AR, MA 선택(min AIC select)
5) 모델 구축(forecast)
6) 모델 검정

```
def ARIMA_model_2021(df):
    import warnings

    from statsmodels.tsa.arima.model import ARIMA
    from statsmodels.tools.sm_exceptions import ConvergenceWarning

    warnings.simplefilter('ignore', ConvergenceWarning)
    # Ignore convergence warning

    p = [1, 2, 4, 6, 8]
    d = q = range(0, 2)
    params_arima = list(it.product(p,d,q))

    combs = {}
    aics = []

    for i, param in enumerate(params_arima):
        try:
            m = ARIMA(df,
                      order=param,
                      enforce_invertibility=False,
                      enforce_stationarity=False)
            m_fit = m.fit()
            combs.update({m_fit.aic : param})
            aics.append(m_fit.aic)

        except: continue

    m_arima_best_aic_idx = min(aics)
    m_arima = ARIMA(df,
                    order=combs[m_arima_best_aic_idx],
                    enforce_invertibility=False,
                    enforce_stationarity=False)
    m_arima_fit = m_arima.fit()

    return m_arima_fit.forecast(26)
```

AIC가 가장 낮은 모델에 예측

3. 분석 세부 내용

※ LGBM, SVR, Random Forest

<Modeling 과정>

1) Data selection 및 Labeling
2) REG_DATE를 제외한 2020년, 2021년의 Meta-Data 예측(by Prophet model)
3) 회귀모델 구축 (LGBM, RF, SVR)
4) Train, Validation set 비율 지정(0.85:0.15)
5) 주요 파라미터 지정
6) GridSearCV ➔ 최적 파라미터 탐색
7) 최종 model fit

```
# 파라미터 지정

lgb_params_grid = {
    'num_leaves': [10,20,30,40],
    'min_data_in_leaf': [10,100, 500, 1000],
    'lambda_l1': [0, 1, 2.0],
    'lambda_l2': [0, 1]
}

svr_params_grid = {
    'kernel':['rbf','poly'],
    'degree':[2,3,4,5,6,7],
    'epsilon':[0.1,0.2,1,10,20,30]
}


rf_params_grid = {
    'n_estimators': [10, 100, 1000],
    'max_depth': [8,10,12, 14],
    'min_samples_leaf': [8,12,18],
    'min_samples_split': [8,16,20]
}
```

3. 분석 세부 내용 – 회귀 분석 모델

Train set
(2015~2019)

- REG_DATE
- P_TYPE
- CTRY_1
- CTRY_2
- P_PURPOSE
- CATEGORY_1
- CATEGORY_2
- P_NAME
- P_IMPORT_TYPE
- P_PRICE

def prepare_df_meta
def meta_predict

PROPHET

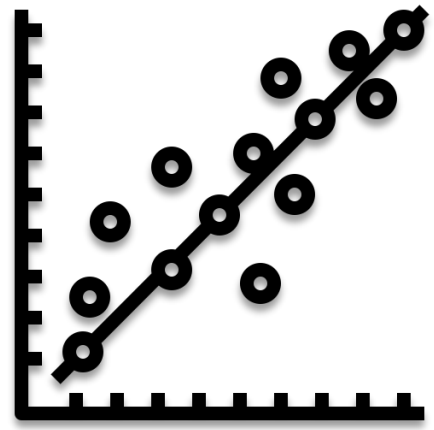
Meta-Data 예측

Meta set ➔ Test set
(2020, 2021)

- REG_DATE
- CTRY_1
- CTRY_2
- P_PURPOSE
- P_IMPORT_TYPE
- P_PRICE

def prepare_df_modeling
def Regression_model

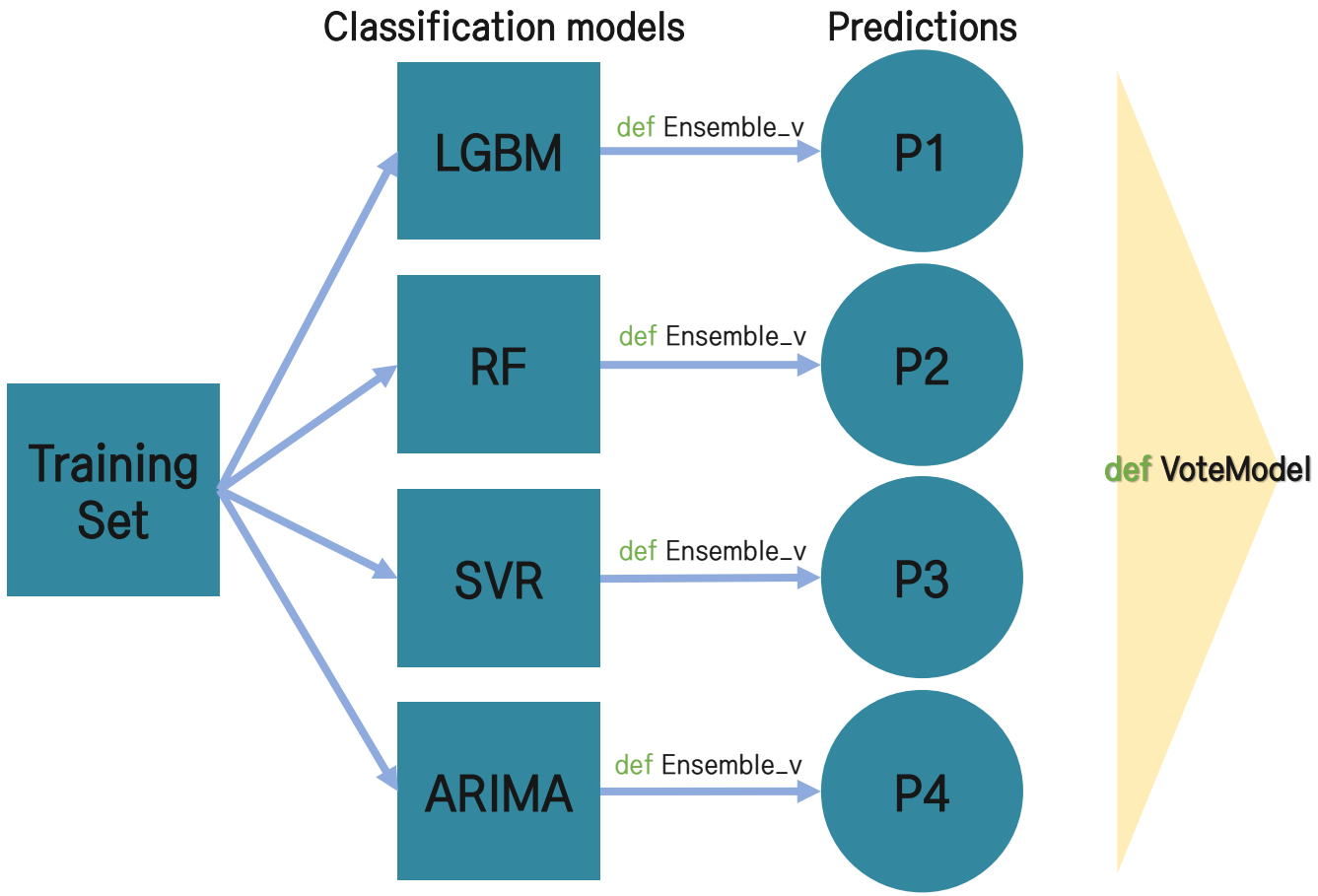
주요 파라미터 지정 후
최적 파라미터 탐색



LGBM, RF, SVR

4. 모델 구축 및 훈련

최종 모델 선정 알고리즘- Voting



Description

2020년 Predict Price 中

자율평가데이터 P_PRICE와

difference 절대값 中 최소값인 모델 선택

(예시 – 오징어 기준)
2020-01-06 예측값(P1-P4) 중 자율평가데이터와
가장 유사한 값을 나타낸 모델은 LGBM
➔ 2021-01-04 예측모델 LGBM 선정

(Final Price)
2021-01-04
LGBM's PRICE

4. 모델 구축 및 훈련

최종 모델 선정
- Final RMSE

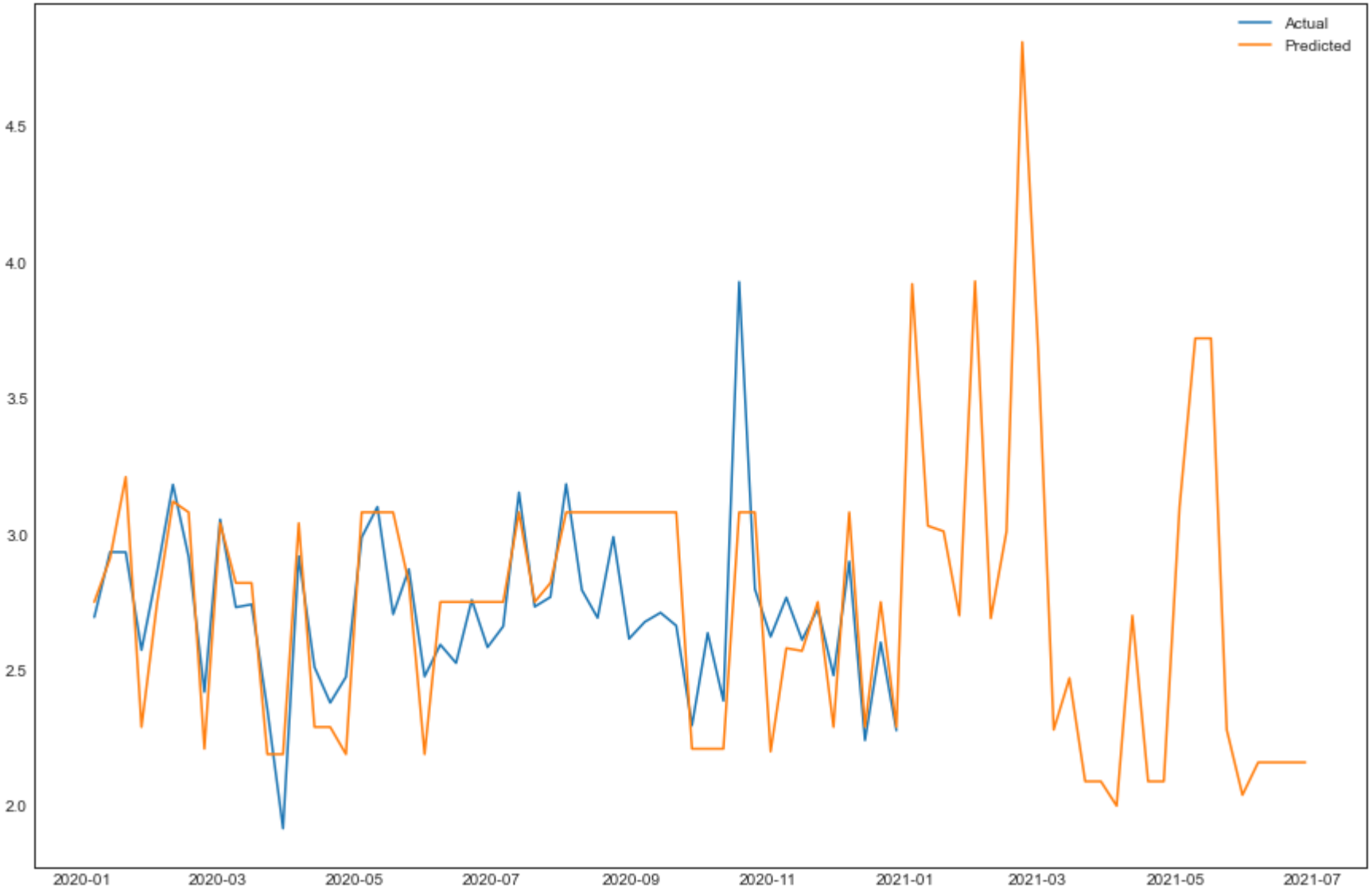
Voting

2020년 Predicted Price vs 자율평가데이터 Price 기준

RMSE	오징어	연어	흰다리새우
LGBM	0.84024	2.84086	1.39337
RF	0.88520	2.84303	1.39588
SVR	1.03100	5.86141	2.84416
ARIMA	0.47365	1.87783	0.77140
Ensemble	**0.245678**	**0.873577**	**0.596125**

5. 결과 분석 - 오징어

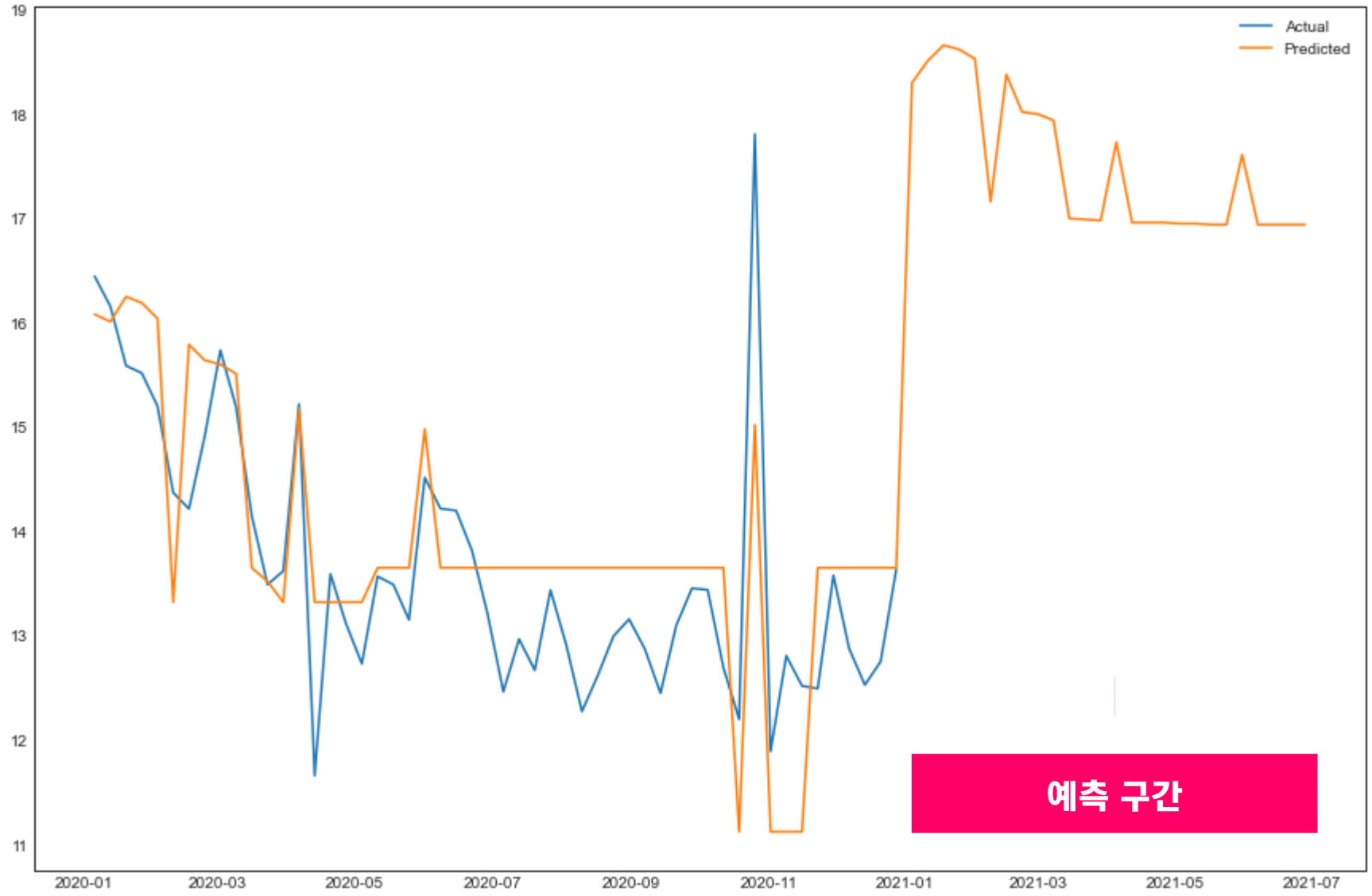
예측 구간



5. 결과 분석 - 오징어

	PREDICT_MODEL	FINAL_VALUE
REG_DATE		
2021-01-04	ARIMA	4.69
2021-01-11	ARIMA	2.91
2021-01-18	ARIMA	3.21
2021-01-25	ARIMA	2.19
2021-02-01	ARIMA	4.69
2021-02-08	SVR	3.12
2021-02-15	ARIMA	3.08
2021-02-22	ARIMA	5.28
2021-03-01	ARIMA	3.04
2021-03-08	ARIMA	1.76
2021-03-15	SVR	2.01
2021-03-22	SVR	1.94
2021-03-29	SVR	1.94
2021-04-05	ARIMA	1.75
2021-04-12	SVR	2.19
2021-04-19	SVR	1.94
2021-04-26	SVR	1.94
2021-05-03	SVR	3.08
2021-05-10	SVR	3.08
2021-05-17	SVR	3.08
2021-05-24	SVR	1.76
2021-05-31	ARIMA	1.85
2021-06-07	SVR	2.11
2021-06-14	SVR	2.11
2021-06-21	SVR	2.11
2021-06-28	SVR	2.11

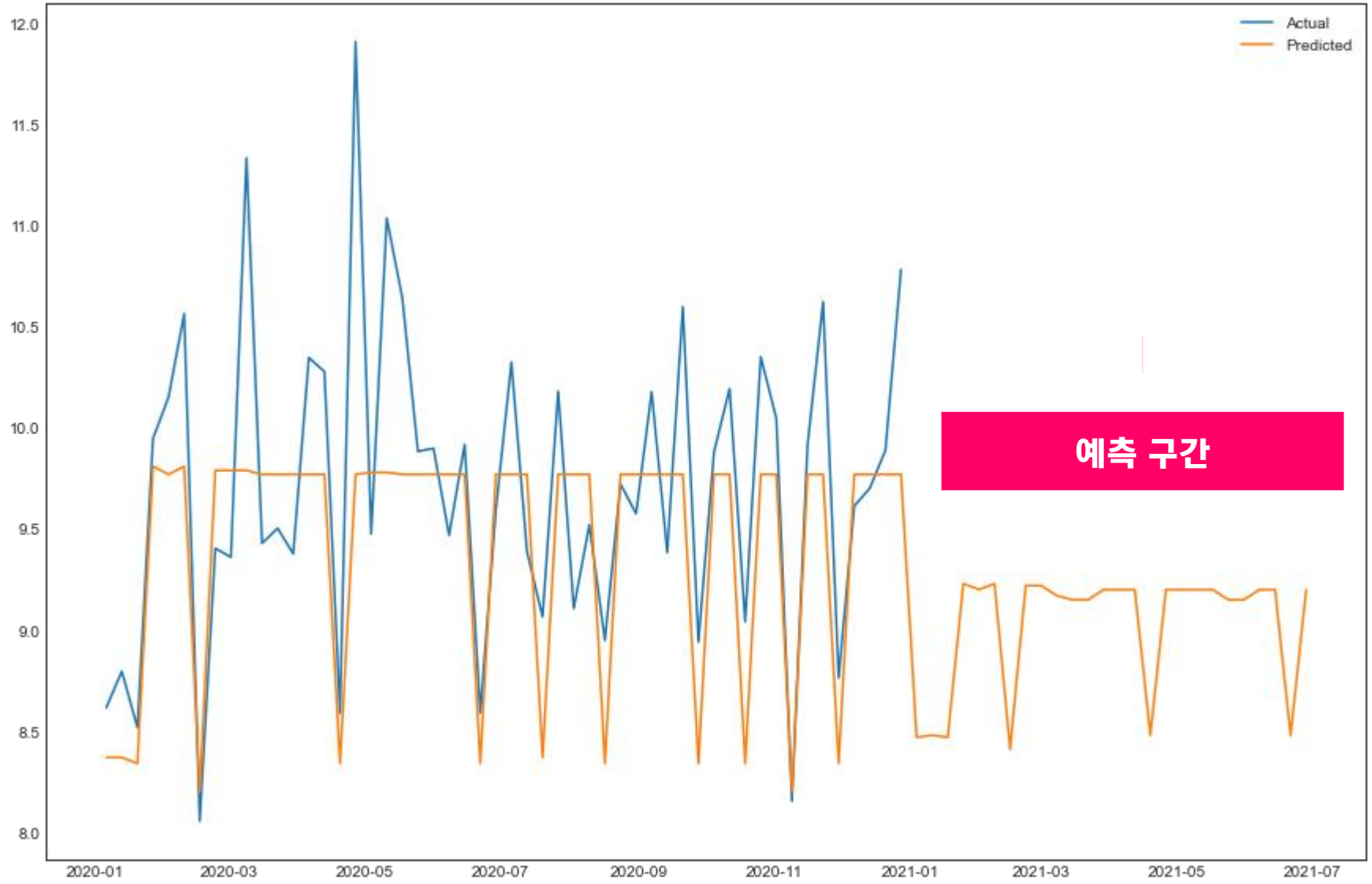
5. 결과 분석 - 연어


Actual
Predicted
19
18
17
16
15
14
13
12
11
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
예측 구간

5. 결과 분석 - 연어

	PREDICT_MODEL	FINAL_VALUE
REG_DATE		
2021-01-04	ARIMA	16.08
2021-01-11	ARIMA	16.01
2021-01-18	ARIMA	16.25
2021-01-25	ARIMA	16.19
2021-02-01	ARIMA	16.04
2021-02-08	SVR	13.32
2021-02-15	ARIMA	15.79
2021-02-22	ARIMA	15.64
2021-03-01	ARIMA	15.60
2021-03-08	ARIMA	15.51
2021-03-15	SVR	13.65
2021-03-22	SVR	13.65
2021-03-29	SVR	13.65
2021-04-05	ARIMA	15.17
2021-04-12	SVR	13.65
2021-04-19	SVR	13.65
2021-04-26	SVR	13.65
2021-05-03	SVR	13.65
2021-05-10	SVR	13.65
2021-05-17	SVR	13.65
2021-05-24	SVR	13.65
2021-05-31	ARIMA	14.98
2021-06-07	SVR	13.65
2021-06-14	SVR	13.65
2021-06-21	SVR	13.65
2021-06-28	SVR	13.65

5. 결과 분석 - 흰다리새우


Actual
Predicted
12.0
11.5
11.0
10.5
10.0
9.5
9.0
8.5
8.0
2020-01
2020-03
2020-05
2020-07
2020-09
2020-11
2021-01
2021-03
2021-05
2021-07
예측 구간

5. 결과 분석 - 흰다리새우

	PREDICT_MODEL	FINAL_VALUE
REG_DATE		
2021-01-04	LGBM	8.34
2021-01-11	LGBM	8.34
2021-01-18	LGBM	8.34
2021-01-25	ARIMA	9.81
2021-02-01	ARIMA	9.77
2021-02-08	ARIMA	9.81
2021-02-15	SVR	8.20
2021-02-22	ARIMA	9.79
2021-03-01	ARIMA	9.79
2021-03-08	ARIMA	9.79
2021-03-15	ARIMA	9.77
2021-03-22	ARIMA	9.77
2021-03-29	ARIMA	9.77
2021-04-05	ARIMA	9.77
2021-04-12	ARIMA	9.77
2021-04-19	LGBM	8.34
2021-04-26	ARIMA	9.77
2021-05-03	ARIMA	9.78
2021-05-10	ARIMA	9.78
2021-05-17	ARIMA	9.77
2021-05-24	ARIMA	9.77
2021-05-31	ARIMA	9.77
2021-06-07	ARIMA	9.77
2021-06-14	ARIMA	9.77
2021-06-21	LGBM	8.34
2021-06-28	ARIMA	9.77

6. 결론 및 시사점

기대효과

정책 당국측면

양식어가의 소득안정화 방안을 모색

가격안정화사업과 같은 정책 효과에 대한 평가

6. 결론 및 시사점

기대효과

생산 어업 측면

유통 및 가공업자 측면

일정한 어업경영 계획 › 가격변동 및 시장의 변화에 대응

2021 BIG CONTEST CHAMPION LEAGUE

감사합니다

Team _ 코지모임

☑ 조희승 (팀장) moohan132435@nate.com
☑ 김혜린 k1h2fls@naver.com