



회귀 보고서 양식

머신 러닝을 이용한 예측 분석

1. 프로젝트 개요

1-1. 주제

전복 데이터 세트를 사용한 회귀 (24.05.01까지 대회 진행 중)

Regression with an Abalone Dataset

Playground Series - Season 4, Episode 4

[k https://www.kaggle.com/competitions/playground-series-s4e4/overview](https://www.kaggle.com/competitions/playground-series-s4e4/overview)

- 평가 지표 : **RMSLE**
- 4월 25일 오후 1시 44분 시점 상위 10% 지점: 213위

1-2. 주제 선정의 배경

- 해양수산부가 K-bluefood 세계화를 통한 수출 확대를 하고자 새로운 전략을 발표, 이에 따라 유망 품목으로 선정된 전복의 품질 경쟁력 확보를 위해 차별화된 양식 및 생산 방식, 유통 전략 필요
- 전복의 다양한 물리적 측정을 통한 전복 연령을 예측으로 전복의 품질 경쟁력 확보 및 생산량 최적화

1-3. 본 프로젝트의 활용 방안 제시

- 어업자, 양식업자의 전복 품질 관리 및 전복 생산량 최적화에 따른 수익 극대화
- 전복 양식업의 체계화된 시스템 구축 가능

- 해양 생태계 모니터링 및 인사이트 획득

2. 프로젝트 수행 절차 및 방법

2-1. 데이터 설명

- 총 8개의 피쳐 변수와 1개의 타겟 변수로 구성

	feature	데이터 타입	결측값	고유타값	max	min
0	id	int64	0	90615	90614	0
1	Sex	object	0	3	M	F
2	Length	float64	0	157	0.815	0.075
3	Diameter	float64	0	126	0.65	0.055
4	Height	float64	0	90	1.13	0.0
5	Whole weight	float64	0	3175	2.8255	0.002
6	Whole weight.1	float64	0	1799	1.488	0.001
7	Whole weight.2	float64	0	979	0.76	0.0005
8	Shell weight	float64	0	1129	1.005	0.0015
9	Rings	int64	0	28	29	1

- 피쳐 데이터

Feature		세부설명
sex	성별	<ul style="list-style-type: none"> • M: 수컷 / F: 암컷 / I:미성숙 개체 • 미성숙 개체는 성별을 명확하게 식별하기 어려운 경우가 존재
length	길이	
diameter	지름	
height	키	
whole weight	전체 중량	고기 무게 + 내장 무게 + 껍데기 무게

Feature		세부설명
whole weight.1 / Shucked weight	고기 무게	
whole weight.2 / Viscera weight	내장 무게	
shell weight / shell weight	껍데기 무게	

- 타겟 데이터

Feature		세부설명
Rings	전복 나이	전복의 바깥 껍데기에 그어진 줄의 개수 (1줄 당 1~.15년)

3. 데이터 전처리

3-1. 데이터 전처리 계획

- 모델링을 위해 고유값을 가진 ID 피쳐 제거
- 데이터 간 컬럼명 통일 및 병합
- 결측값 확인
- 이상치 확인
- 의미있는 피쳐 변수 생성
- 범주형 피쳐 인코딩 변환
- 타겟 변수 로그 변환

3-2. 데이터

▼ Data

Regression with an Abalone Dataset

[Regression with an Abalone Dataset.zip](#)

[abalone.zip](#)

```
train = pd.read_csv('/content/train.csv')
original = pd.read_csv('/content/abalone.csv')
test = pd.read_csv('/content/test.csv')
submission = pd.read_csv('content/sample_submission.csv')
```

```
train.shape = (90615, 10)
original.shape = (4177, 9)
test.shape = (60411, 9)
submission.shape = (60411, 2)
```

- 전처리 후 train / test data

- train.shape = (94792, 11)

	feature	데이터 타입	결측값	고유타	max	min	head1	head2	head3
0	Length	float64	0	157	0.8150	0.0750	0.5500	0.6300	0.1600
1	Diameter	float64	0	126	0.6500	0.0550	0.4300	0.4900	0.1100
2	Height	float64	0	90	1.1300	0.0000	0.1500	0.1450	0.0250
3	Whole weight	float64	0	3205	2.8255	0.0020	0.7715	1.1300	0.0210
4	Shucked weight	float64	0	1806	1.4880	0.0010	0.3285	0.4580	0.0055
5	Viscera weight	float64	0	983	0.7600	0.0005	0.1465	0.2765	0.0030
6	Shell weight	float64	0	1132	1.0050	0.0015	0.2400	0.3200	0.0050
7	Rings	int64	0	28	29.0000	1.0000	11.0000	11.0000	6.0000
8	F	int64	0	2	1.0000	0.0000	1.0000	1.0000	0.0000
9	I	int64	0	2	1.0000	0.0000	0.0000	0.0000	1.0000
10	M	int64	0	2	1.0000	0.0000	0.0000	0.0000	0.0000

- test.shape = (60411, 10)


	feature	데이터 타입	결측값	고유타값	max	min	head1	head2	head3
0	Length	float64	0	148	0.8000	0.0750	0.6450	0.5800	0.5600
1	Diameter	float64	0	130	0.6500	0.0550	0.4750	0.4600	0.4200
2	Height	float64	0	85	1.0950	0.0000	0.1550	0.1600	0.1400
3	Whole weight	float64	0	3037	2.8255	0.0020	1.2380	0.9830	0.8395
4	Shucked weight	float64	0	1747	1.4880	0.0010	0.6185	0.4785	0.3525
5	Viscera weight	float64	0	960	0.6415	0.0005	0.3125	0.2195	0.1845
6	Shell weight	float64	0	1089	1.0040	0.0015	0.3005	0.2750	0.2405
7	F	int64	0	2	1.0000	0.0000	0.0000	0.0000	0.0000
8	I	int64	0	2	1.0000	0.0000	0.0000	0.0000	0.0000
9	M	int64	0	2	1.0000	0.0000	1.0000	1.0000	1.0000

3-3. 데이터 수집 및 전처리

- 추가 data 확보

UCI Machine Learning Repository

Discover datasets around the world!

 <https://archive.ics.uci.edu/dataset/1/abalone>

- 효율적인 모델링을 위해 고유값을 가진 ID 피쳐 제거 및 데이터 간 컬럼명 통일

```
[ ] 1 # id 삭제
    2 train = train.drop(['id'], axis = 1)
    3 train.columns = original.columns
```

```
[ ] 1 # id 삭제된 train 데이터 확인
    2 train.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	F	0.550	0.430	0.150	0.7715	0.3285	0.1465	0.2400	11
1	F	0.630	0.490	0.145	1.1300	0.4580	0.2765	0.3200	11
2	I	0.160	0.110	0.025	0.0210	0.0055	0.0030	0.0050	6
3	M	0.595	0.475	0.150	0.9145	0.3755	0.2055	0.2500	10
4	I	0.555	0.425	0.130	0.7820	0.3695	0.1600	0.1975	9

```
[ ] 1 # id 삭제된 train 데이터 shape 확인
    2 train.shape
```

```
(90615, 9)
```

```
[ ] 1 # submission id에 대입하기 위한 변수 test_id(test데이터에서 id데이터 삭제하기 전에 미리 저장)
2 test_id = test['id']
3 # test데이터에서 id 삭제
4 test = test.drop('id', axis = 1)
5 # test데이터의 컬럼들을 train데이터의 컬럼과 동일하게
6 test.columns = train.columns
7 test.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight
0	M	0.645	0.475	0.155	1.2380	0.6185	0.3125	0.3005
1	M	0.580	0.460	0.160	0.9830	0.4785	0.2195	0.2750
2	M	0.560	0.420	0.140	0.8395	0.3525	0.1845	0.2405
3	M	0.570	0.490	0.145	0.8740	0.3525	0.1865	0.2350
4	I	0.415	0.325	0.110	0.3580	0.1575	0.0670	0.1050



- 전처리 전

'id', 'Sex', 'Length', 'Diameter', 'Height', 'Whole weight', 'Whole weight.1', 'Whole weight.2', 'Shell weight', 'Rings'

-

전처리 후

'Sex', 'Length', 'Diameter', 'Height', 'Whole weight', '@ 'Shucked weight', 'Viscera weight', 'Shell weight', 'Rings'

- 기존 train data와 추가로 수집한 original data 병합

```
[ ] 1 # train데이터와 original(abalone)데이터 병합
2 train = pd.concat([train, original], axis = 0, ignore_index=True)
3 train.head()
```

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	F	0.550	0.430	0.150	0.7715	0.3285	0.1465	0.2400	11
1	F	0.630	0.490	0.145	1.1300	0.4580	0.2765	0.3200	11
2	I	0.160	0.110	0.025	0.0210	0.0055	0.0030	0.0050	6
3	M	0.595	0.475	0.150	0.9145	0.3755	0.2055	0.2500	10
4	I	0.555	0.425	0.130	0.7820	0.3695	0.1600	0.1975	9



- 전처리 전

train.shape =

(90615, 9)

original.shape = (4177, 9)

-

전처리 후

train.shape = (94792, 9)

- 결측값 확인

```
[ ] 1 # train 누락된 값 확인  
2 train.isna().sum()
```

```
Sex          0  
Length       0  
Diameter     0  
Height       0  
Whole weight 0  
Shucked weight 0  
Viscera weight 0  
Shell weight 0  
dtype: int64
```

```
[ ] 1 # test 누락된 값 확인  
2 test.isna().sum()
```

```
Sex          0  
Length       0  
Diameter     0  
Height       0  
Whole weight 0  
Shucked weight 0  
Viscera weight 0  
Shell weight 0  
dtype: int64
```



- train, test 둘 다 결측값이 없는 것을 확인

- 범주형 피처에 대한 onehot-encoding

```

1 encoder = OneHotEncoder(sparse_output = False, handle_unknown = 'ignore')
2
3 train = pd.concat([train.iloc[:,1:], pd.DataFrame(encoder.fit_transform(train[['Sex']]).astype('int'), columns = encoder.categories_[0]), axis = 1)
4 train.head()

```

	Length	Diameter	Height	Whole weight	Whole weight.1	Whole weight.2	Shell weight	F	I	M
0	0.550	0.430	0.150	0.7715	0.3285	0.1465	0.2400	1	0	0
1	0.630	0.490	0.145	1.1300	0.4580	0.2765	0.3200	1	0	0
2	0.160	0.110	0.025	0.0210	0.0055	0.0030	0.0050	0	1	0
3	0.595	0.475	0.150	0.9145	0.3755	0.2055	0.2500	0	0	1
4	0.555	0.425	0.130	0.7820	0.3695	0.1600	0.1975	0	1	0

steps: [Generate code with train](#) [View recommended plots](#)

```

1 test = pd.concat([test.iloc[:,1:], pd.DataFrame(encoder.transform(test[['Sex']]).astype('int'), columns = encoder.categories_[0]), axis = 1)
2 test.head()

```

	Length	Diameter	Height	Whole weight	Whole weight.1	Whole weight.2	Shell weight	F	I	M
0	0.645	0.475	0.155	1.2380	0.6185	0.3125	0.3005	0	0	1
1	0.580	0.460	0.160	0.9830	0.4785	0.2195	0.2750	0	0	1
2	0.560	0.420	0.140	0.8395	0.3525	0.1845	0.2405	0	0	1
3	0.570	0.490	0.145	0.8740	0.3525	0.1865	0.2350	0	0	1
4	0.415	0.325	0.110	0.3580	0.1575	0.0670	0.1050	0	1	0



- 후에 나올 EDA 결과 성별 피쳐 분포도가 다르게 나왔기 때문에 원핫 인코딩을 통해 범주형 데이터를 전처리, 성별 피쳐를 생성하고 그 값을 추가

- Target 피쳐에 대한 로그 변환

```

[ ] 1 y = train['Rings']
    2 y_log = np.log(1+y) # 값이 음수거나 0일때 로그변환하면 사용할 수 없어서 1을 더한다.

```

```

[ ] 1 # 로그 변환 전
    2 y.head()

```

0	11
1	11
2	6
3	10
4	9

Name: Rings, dtype: int64

```

[ ] 1 # 로그 변환 후
    2 y_log.head()

```

0	2.484907
1	2.484907
2	1.945910
3	2.397895
4	2.302585

Name: Rings, dtype: float64



- 데이터의 분포를 정규 분포에 가깝게 만들고, 오차에 대한 예측 오류의 영향을 완화시키기 위해 로그 변환을 시행

- 타겟 값이 0이거나 음수인 경우 로그 변환시 -inf값이 발생하여 오류를 일으키기 때문에 +1

3-4. 활용 라이브러리 등 기술적 요소

- ✓ pandas → 모듈
- ✓ numpy
- ✓ matplotlib
- ✓ seaborn
- ✓ warnings
- ✓ random
- ✓ lightgbm
- ✓ catboost
- ✓ xgboost
- ✓ sklearn
- ✓ optuna

3-5. 프로젝트에서 분석한 내용

- ✓ 결측치 확인
- ✓ 데이터 타입 확인
- ✓ 이상치 확인
- ✓ 전체 수치 변수 시각화
- ✓ 병합된 데이터 shape 확인
- ✓ 병합된 데이터 타입 확인
- ✓ 수치 특징과 목표 변수 간의 상관관계 시각화

- ✓ 각 데이터들 분포도 시각화
- ✓ 성별 비율 시각화
- ✓ 피쳐 중요도 시각화

4. 기초 평가

4-1. 지표평가

• RMSLE(Root Mean Squared Log Error)

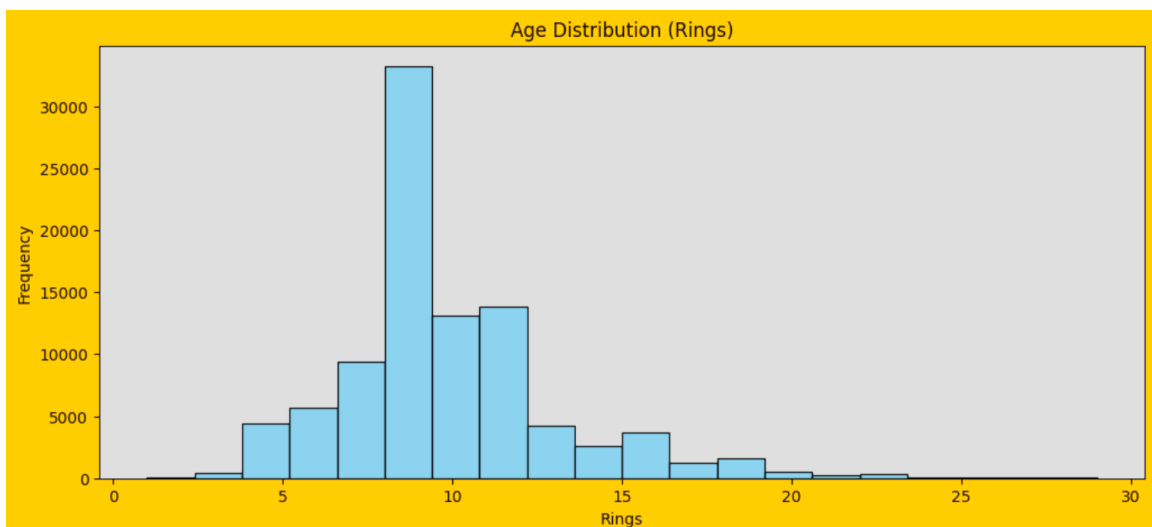
- 대상 변수가 다양한 값의 범위를 가지는 경우에 유용함. RMSLE는 예측 값의 로그와 실제 값의 로그 사이의 차이의 제곱의 평균의 제곱근으로 계산

$$RMSLE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\log(p_i + 1) - \log(a_i + 1))^2}$$

- n : 관측치의 수
- p_i : 관측치 i 에 대한 예측값
- a_i : 관측치 i 에 대한 실제값
- \log : 자연 로그

4-2 . 시각화- 추이, 편차, 구성비율

▼ Target(Rings) 분포 확인





mean : 9.71

std : 3.18

min : 1.0

25% : 8.0

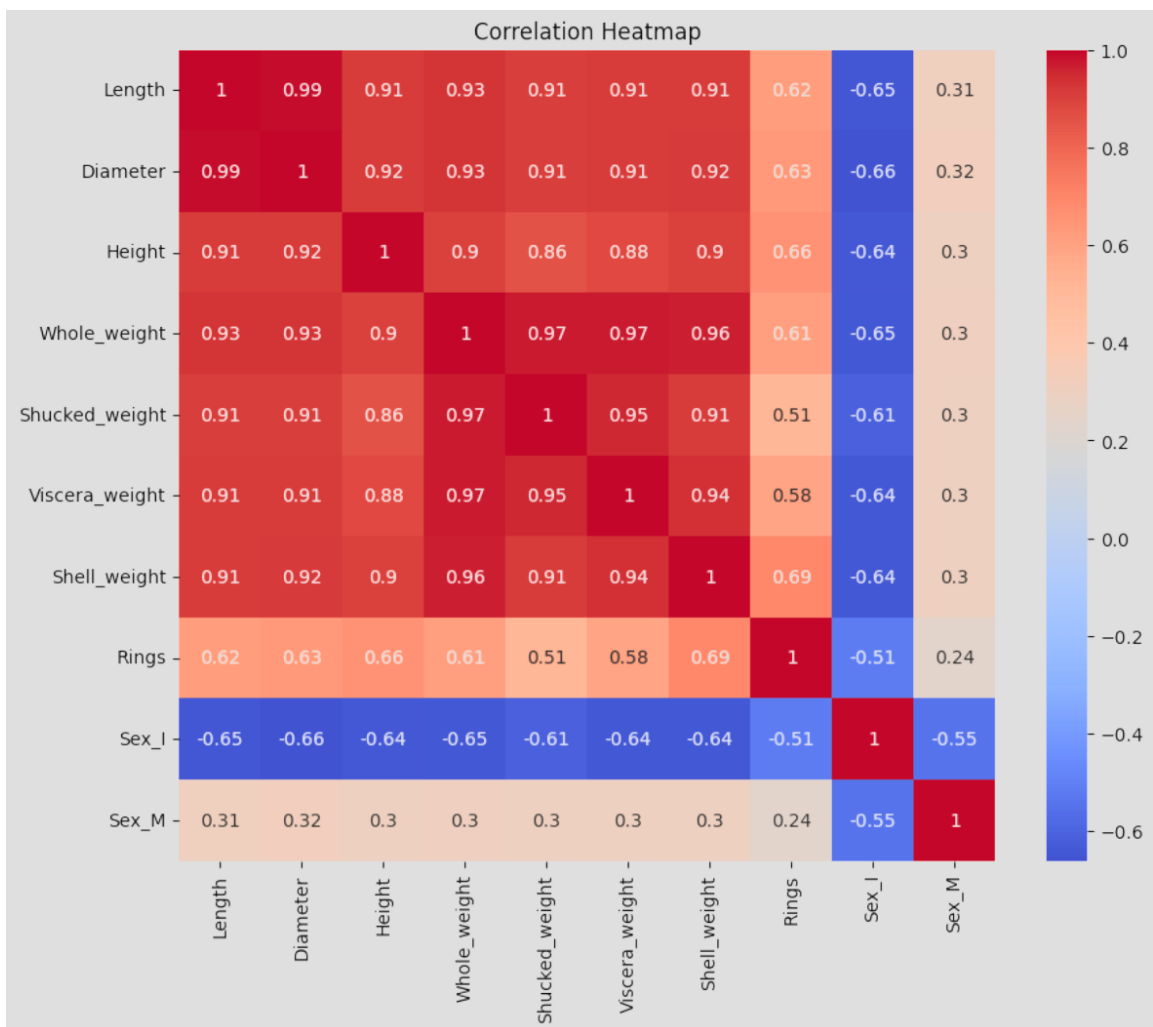
50% : 9.0

75% : 11.0

max : 29.0

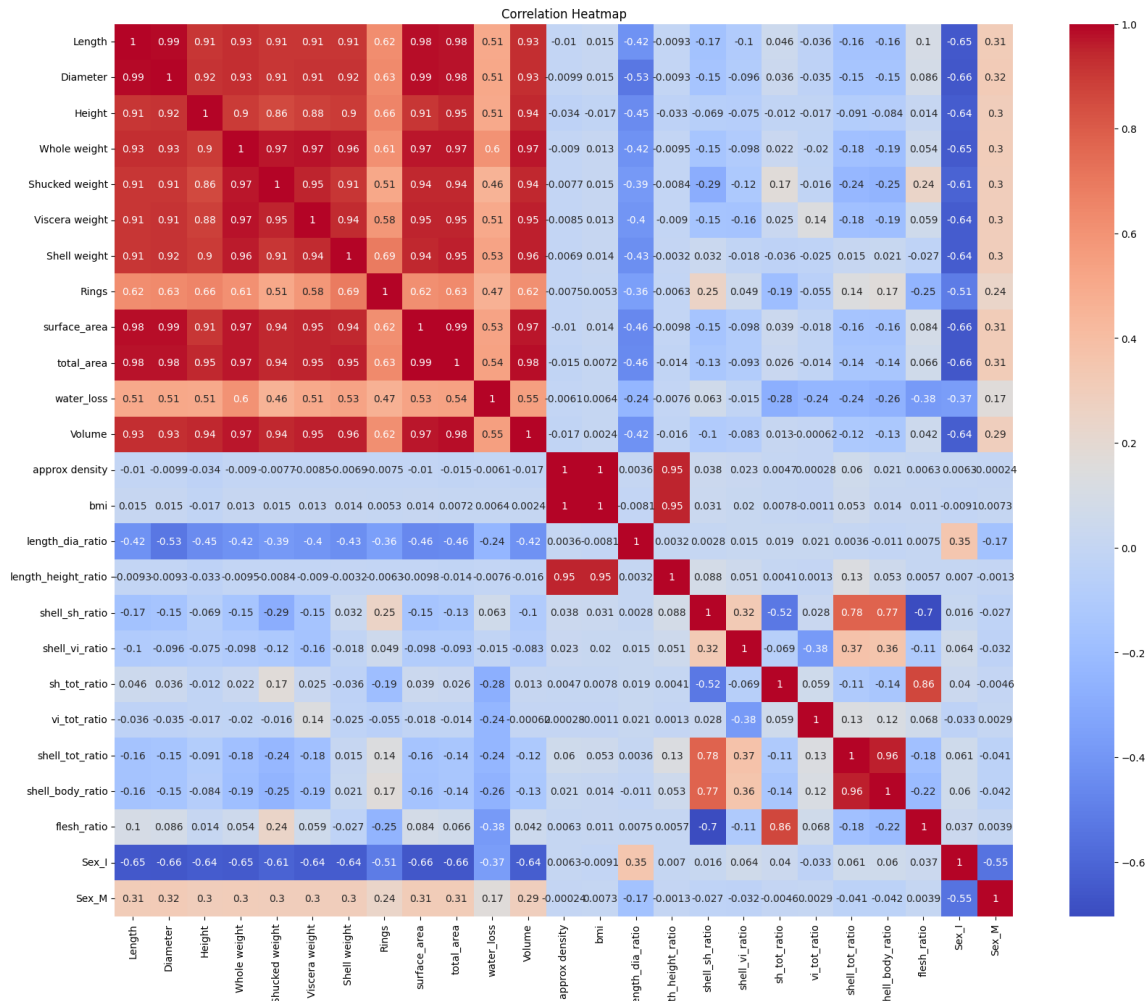
- 오른쪽으로 긴 꼬리를 형성

▼ 수치 피처와 타겟 변수 간의 상관 관계



- 각 피처들 간 상관 관계가 얼마나 높고 낮은지를 확인 결과, 기존 피처 유의미한 상관 관계를 확인

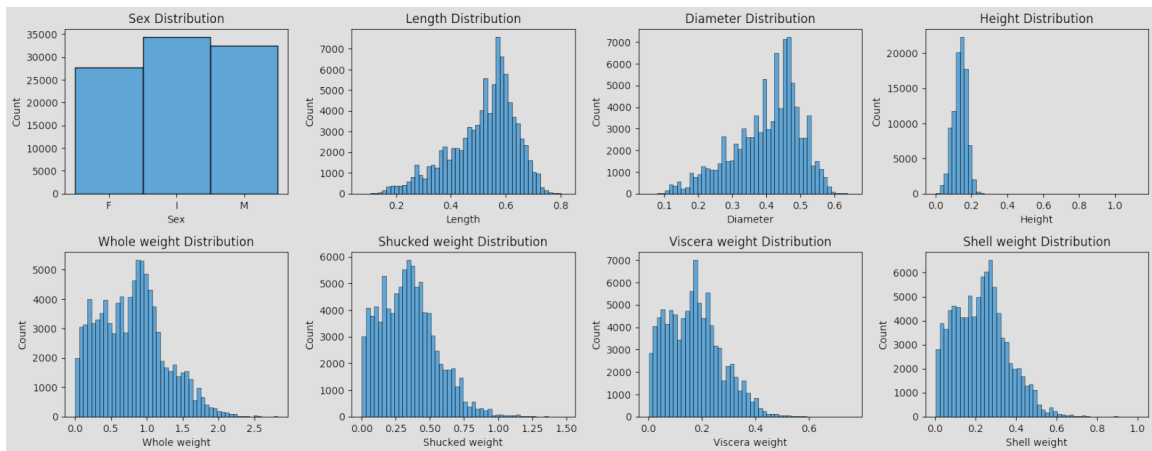
▼ 수치 피처와 타겟 변수 간의 상관 관계(뉴피처 생성)



- 새로 생성한 피처의 경우 기존에 비해 아주 미약한 상관 관계를 갖고 있었기 때문에 생성한 피처를 제외하고 기존 피처만으로 분석을 진행

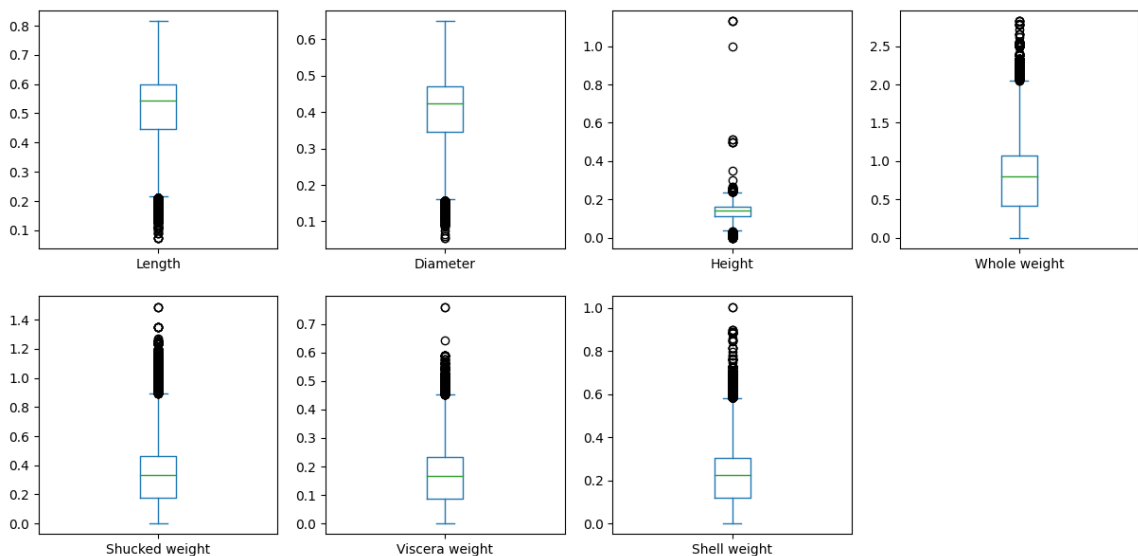
- 새로 생성한 피처 중 상관 관계가 높게 관측된 면적과 부피 데이터는 모델링 결과 오히려 측정 수치가 악화되는 현상이 발견되어 제외

▼ 각 데이터 분포



- 각 데이터들의 분포도를 확인했을 때, 모델 학습에 중요한 전복 무게 관련 된 데이터들은 골고루 분포되어 있는 것을 확인

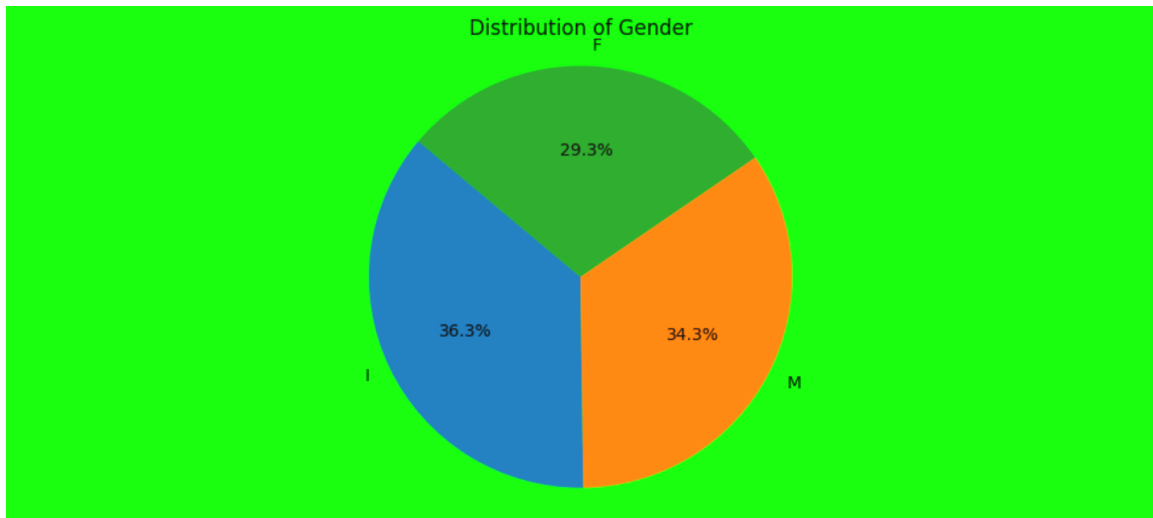
▼ 각 데이터의 이상치



- 각 데이터들의 이상치를 확인했을 때, height(키)를 제외한 데이터들은 모두 넓게 퍼짐
 - 피쳐 데이터에 IQR 기준 이상치가 확인되었으나, test 데이터의 경우에도 이상치 확인
 - 이상치를 제거하고 진행하나 모델의 성능에 따라 이상치 제거 보류

▼ 성별 비율

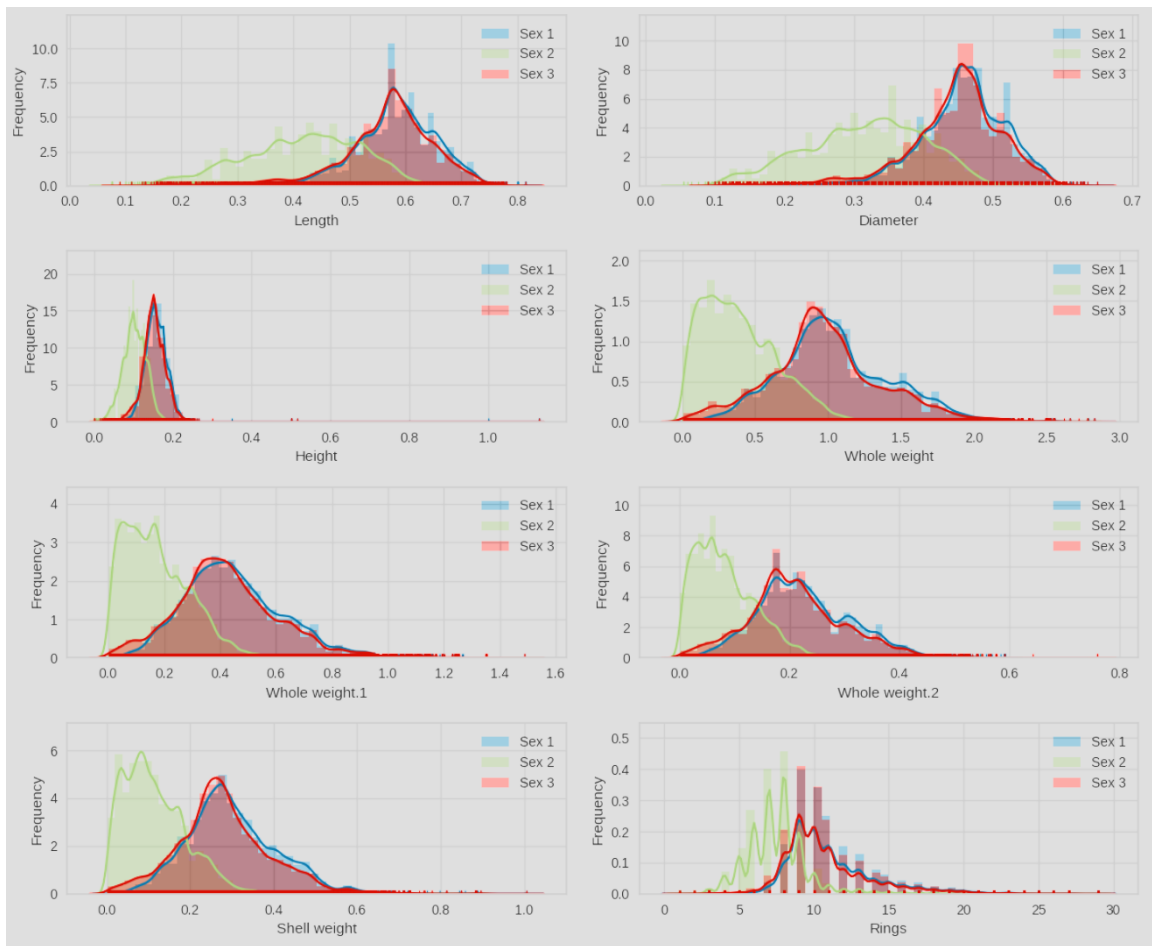
- Sex
 - F: Female, I: immature, M: male



- 전복들의 성별 비율을 확인했을 때, 대체로 골고루 분포되어 있는 것을 확인

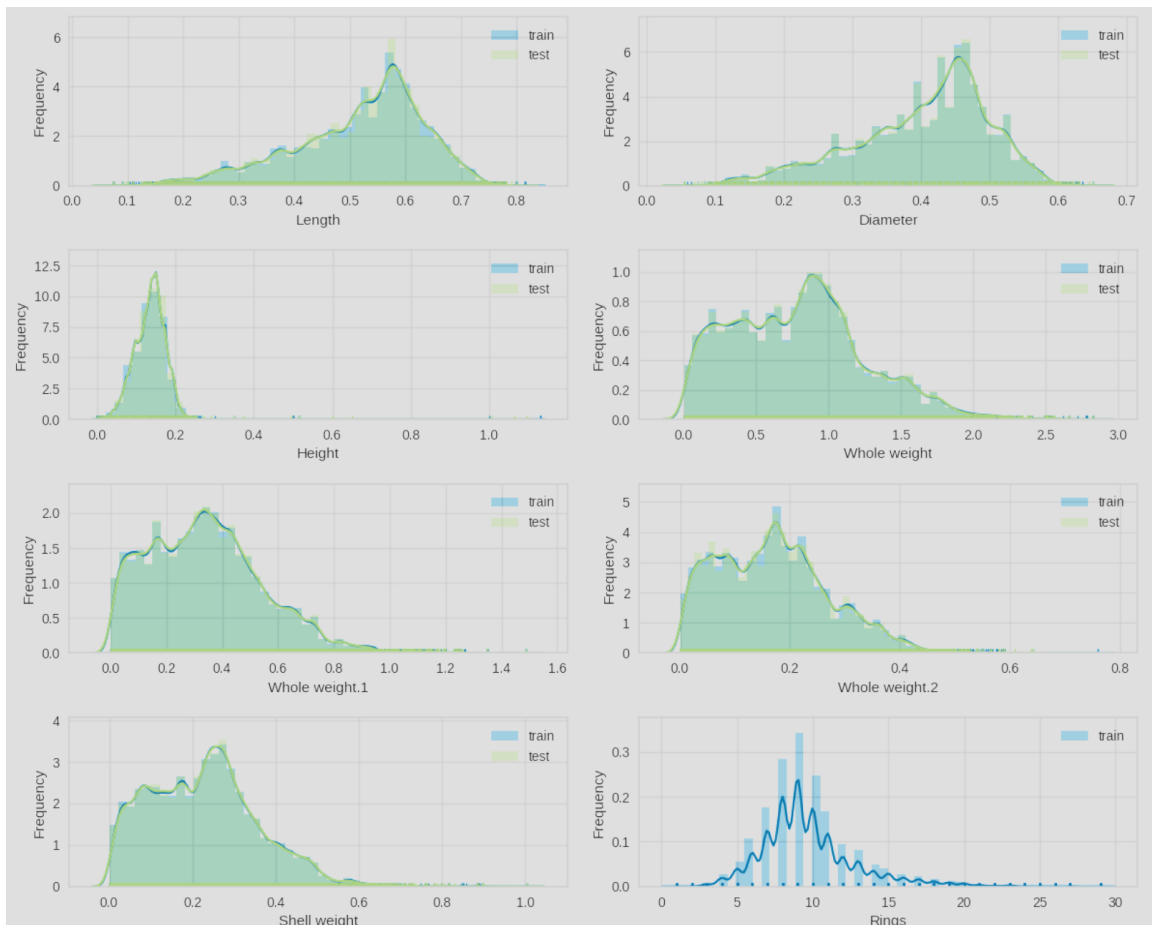
▼ 성별 피쳐 데이터 분포

- Sex
 - 1: Female, 2: immature, 3: male



- 성별 피쳐 데이터 분포를 확인한 결과 Female과 Male 전복은 대체로 비슷한 양상을 보이는 것을 확인
- Immature의 경우 분포가 위의 둘과 다르게 형성
- 모델 학습시 Female과 Male은 함께 학습하되, Immature의 경우는 분리시켜 학습하는 방향을 고려

▼ train과 test 분포도 비교 확인



- train과 test 분포도가 비슷한 것을 확인

5. 피쳐 엔지니어링

5-1 피쳐 엔지니어링

1. 추가 데이터 병합 및 이를 위한 ID피쳐 제거

```
train.drop(columns=["id"], inplace=True)
test.drop(columns=["id"], inplace=True)

train=pd.concat([train,original],axis='rows')
train.reset_index(inplace=True,drop=True)
```

2. 로그 변환


```
y = train['Rings']
y_log = np.log(1+y)
```

- train['Rings']의 값이 음수거나 0일 때 로그 변환 시 오류 발생 → +1


3. 성별 인코딩


```
1 encoder = OneHotEncoder(sparse_output = False, handle_unknown = 'ignore')
2
3 train = pd.concat([
4     train.iloc[:,1:],
5     pd.DataFrame(encoder.fit_transform(train[['Sex']]).astype('int'),
6                   columns = encoder.categories_[0])
7 ],
8                 axis = 1
9 )
10
11 test = pd.concat([
12     test.iloc[:,1:],
13     pd.DataFrame(encoder.transform(test[['Sex']]).astype('int'),
14                   columns = encoder.categories_[0])
15 ],
16                 axis = 1
17 )
```


- M / F / I 로 구분되어 있는 범주형 피처인 "Sex" 피처를 원핫인코딩 처리


6. 모델 학습

6-1. 프로젝트에 사용했던 방법들






 (0.14700)FM_I top3model(cat, lgbm, rf)

 (0.14708)FM_I top4model(cat,lgbm,xgb,rf)












 (0.14843)FM_I top4model_feature(피처 추가 ...

 (0.14958)top6model_(피처 추가 및 이상값 제거)







- 다양한 피처 추가 및 이상값 제거
- pycaret 사용 top_model 추출 및 blend_models 형성

 (0.14564) submission_voting_fold10(파라미터 수정1)
 (0.14565) submission_voting_fold10(파라미터 수정3)
 (0.14568) submission_voting_fold10(파라미터 수정2)
 (0.14576) submission_voting_fold5(원본 파라미터)
 (0.14606) submission_voting_feature(remains, volume)

- 앙상블은 Voting으로 고정
- 모델은 XGBoost, CatBoost, LGBM 3개 사용
- 뉴피쳐 추가하여 학습 → 성능 저하
- 하이퍼 파라미터를 여러 번 수정하며 학습

 (0.14559) submission_voting_fold11(최종)
 (0.14559) submission_voting_fold13(최종)
 (0.14559) submission_voting_fold14(최종)
 (0.14560) submission_voting_fold12(최종)
 (0.14563) submission_voting_fold15(최종)
 (0.14564) submission_voting_fold10(최종)
 (0.14565) submission_voting_fold9(최종)
 (0.14567) submission_voting_fold7(최종)
 (0.14567) submission_voting_fold8(최종)
 (0.14569) submission_voting_fold6(최종)
 (0.14576) submission_voting_fold5(최종)

- 앙상블 : Voting
- 모델 : XGBoost, CatBoost, LGBM
- Fold 횟수를 5~15까지 각각 적용하며 학습
- **0.1455점대의 점수가 나온 fold(11, 13, 14)를 최종적으로 사용**

 (0.14558) voting_fold10_top3(가중치_1, 1, 1)
 (0.14559) voting_fold10_top5(가중치_1, 1, 1, 1, 1)
 (0.14565) voting_fold10_top4(가중치_10,10,1,1)_ensemble
 (0.14565) voting_fold10_top5(가중치 모두 1로 동일)_ensemble
 (0.14566) voting_fold10_top4(가중치_10, 1, 1, 1)_ensemble
 (0.14566) voting_fold10_top4(가중치_10, 10, 10, 1)_ensemble
 (0.14566) voting_fold10_top7(가중치_10,9,8,7,6,5,4)_ensemble
 (0.14567) voting_fold10_top3(가중치_10,1,1)_ensemble


- 앙상블 : Voting
- 모델 : XGBoost, CatBoost, LGBM
- 각 Fold(5~15)데이터들을 사용하여 앙상블 진행할 때, 가중치를 여러 번 수정하며 학습

<결론>

- 5~15fold까지 각각 제출하였을 때, 점수가 가장 높았던 top3(fold(11, 13, 14))를 사용하여
최종 학습 진행(가중치는 모두 1로 설정)

7. 머신 러닝 결과

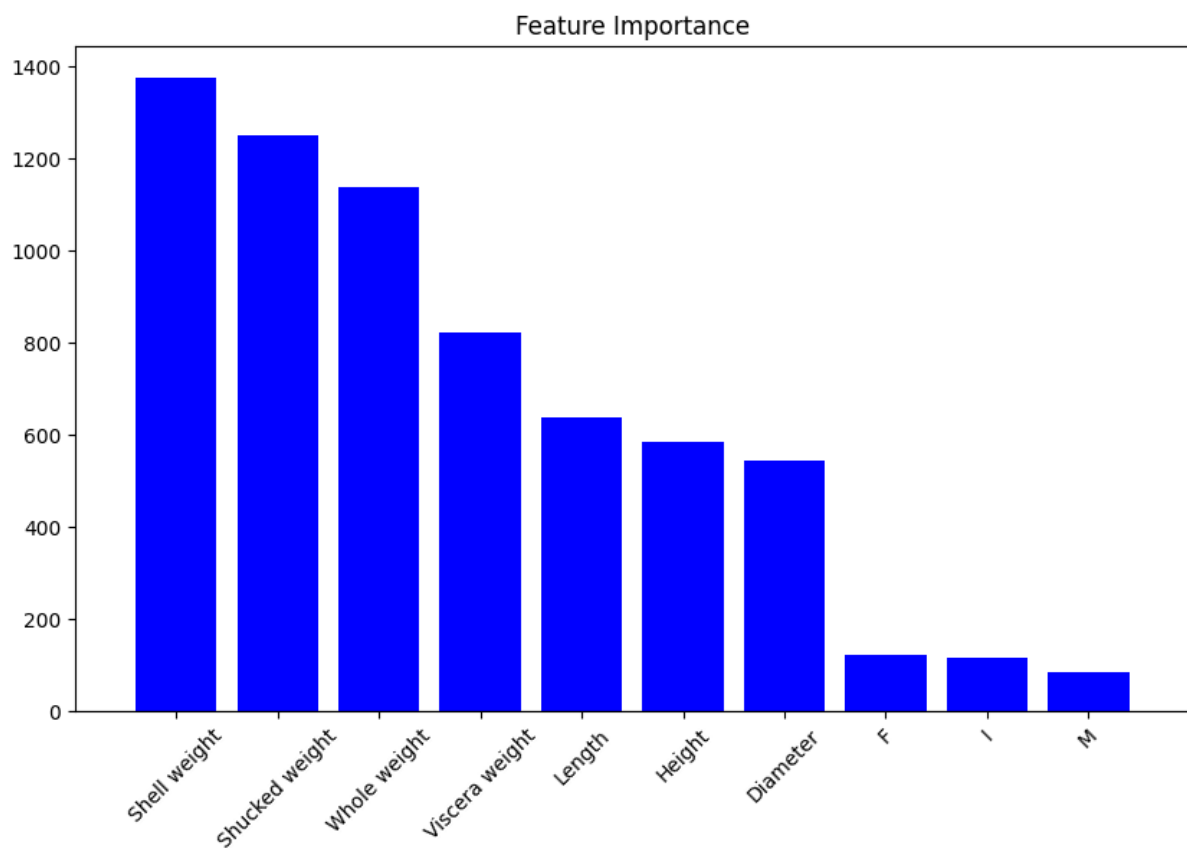
7-1. 결과 분석

	vote_fold_top3(_1 1 1).csv 완료 · 5시간 전	0.14558
---	--	----------------

199	하디 가루디		0.14557	삼	7일
200	전사		0.14557	11	5d
201	파벨 니콜라이체프		0.14557	1	12일
202	아흐메드 아불크헤어		0.14557	1	10일
203	크리스베밥		0.14557	53	1일
204	곽재우		0.14558	16	1일
최고의 출품작입니다! 가장 최근 제출하신 점수는 0.14558로, 이전 점수인 0.14559보다 향상된 수치입니다. 잘 했어!			이것을 트윗하세요		
205	시밤 싱		0.14558	1	14일
206	데이터투		0.14558	5	8일
207	조자성		0.14559	10	1일
208	미즈틱		0.14559	17	1일
209	GMROH637		0.14561	6	11일

- 4월 25일 오후 1시 44분 기준 0.14558로 204등으로 상위 10%(213등)안에 진입

7-2. 피쳐 중요도 결과



- 무게 관련 피쳐 중요도가 높고, 그 뒤로 길이 관련 피쳐들이 뒤따르는 것을 확인

8. 프로젝트 회고 및 개선점

8-1. 피드백

제출 전에는 이곳이 공백입니다.

발표 후 QnA 시간에 나온 질문과 피드백을 모두 작성해 주세요.

듣는 즉시 바로 작성하면 빠뜨리지 않고 모두 적을 수 있을 거예요! 이때 개선점으로 넘어가도 좋을 반영할 부분을 발견했다면 최종 제출 전에 그 부분 위주로 정리하는 것도 좋아요. 그리고 발표 시간에 적극적으로 질문과 피드백을 주고 받으면 서로의 성장에 무척 도움이 되겠죠?

8-2. 회고

8-3. 개선점

8-4. 추후 개선 계획

개선점에 대한 회고 이후, 가능하다면 실제 액션 계획도 세워보세요. 포트폴리오에서 '개선 시도/경험'은 아주 긍정적인 요소로 작용한답니다.

8. 부록

8-1. 참고자료

<https://www.kaggle.com/code/arunklenin/ps4e4-abalone-age-prediction-regression/notebook#4.1-New-Features>

<https://www.kaggle.com/code/bunny11/voting-classifier/notebook>

<https://www.kaggle.com/code/satyaprakashshukl/cb-regression-analysis/notebook>

8-2. 출처

<https://archive.ics.uci.edu/dataset/1/abalone>

<https://www.kaggle.com/competitions/playground-series-s4e4>