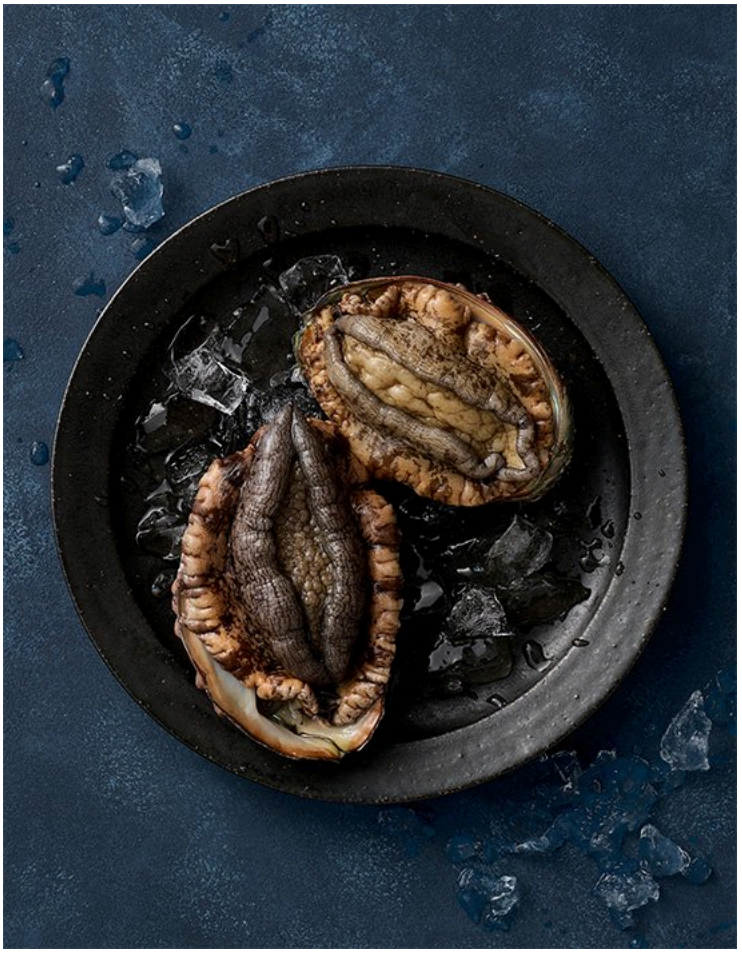
전복 연령 예측 회귀 분석

프로젝트 타입	팀 프로젝트
Tool	Python
날짜	@2024년 4월 11일 → 2024년 4월 26일



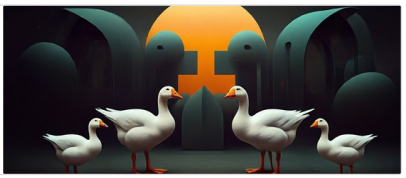
목차

Regression with an Abalone Dataset

: 전복 연령 예측

Regression with an Abalone Dataset
Playground Series - Season 4, Episode 4

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



💡 해당 보고서는 기존 프로젝트 진행 후 피드백 진행하여 작성

- 기존 프로젝트

  회귀 기획서

  회귀

- 피드백
  1. 상관관계 분석 결과 모델 활용
  2. 로그 변환 시 특정 값을 더하는 근거
  3. 딥러닝 시도(DNN)

. 데이터 설명

1_1. 데이터 크기

	행(row)	열(column)
Train	90615	10

	행(row)	열(column)
Test	60411	9
Submission	60411	2
Original	4177	10

1_2. 변수 설명

Feature	mean	Type	measure	Train NA	Test NA	Origin NA	Submiss
Id	아이디	int64	-	0	0	0	0
Sex	전복 성별	object	-	0	0	0	-
Length	전복 길이	float64	mm	0	0	0	-
Diameter	전복 둘레	float64	mm	0	0	0	-
Height	전복 높이	float64	mm	0	0	0	-
Whole_weight	전복 전체 무게	float64	grams	0	0	0	-
Shucked_weight	껍질 제외 무게	float64	grams	0	0	0	-
Viscra_weight	출혈 후 내장 무게	float64	grams	0	0	0	-
Shell_weight	건조 후 껍질 무게	float64	grams	0	0	0	-

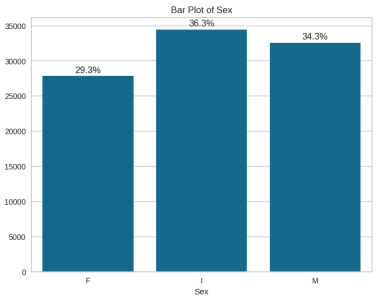
Target	mean	Type	Train NA	Test NA	Origin NA	Submission NA
Rings	전복 연령	int64	0	-	0	0

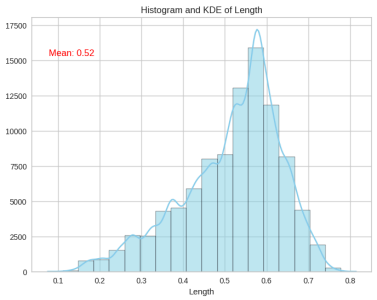
* 자세한 전복 데이터의 설명은 링크 참조

. 데이터 EDA 시각화

: train, test 분포 동일

2_1. 단일 변수 분석(Histogram,BarPlot)

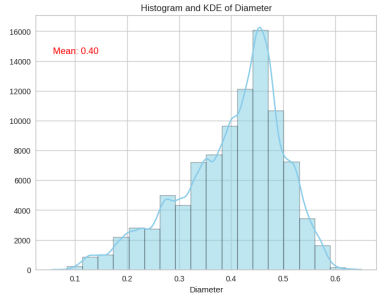


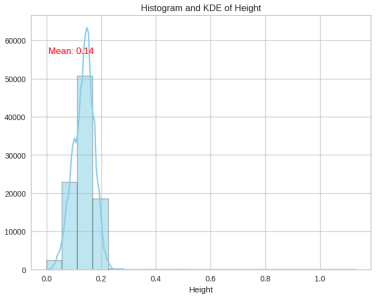


- **Sex (전복 성별)**
  - 분포
    - F(29.3%), I(36.3%), M(34.3%)로 구성되어 있음.
    - I(Intermediate)가 가장 높은 비율을 차지하며, F와 M은 상대적으로 유사한 분포를 보임.
    - 데이터의 성별 분포가 특정 그룹에 치우치지 않고 고르게 나타남.

- **Length (전복의 가장 긴 부분 측정)**
  - 평균
    - 0.52
  - 분포
    - 대체로 정규분포 형태
    - 0.5~0.6 사이에 데이터가 집중
    - 극단값(0.1 이하, 0.7 이상)은 상대적으로 적음
  - 값

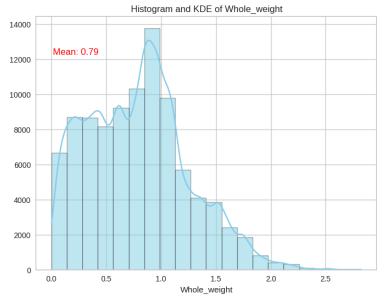
- 최소 : 0.075
- 최대 : 0.815

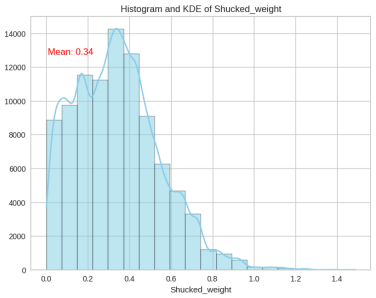




- **Diameter (전복 직경)**
  - 평균
    - 0.40
  - 분포
    - 대체로 정규분포 형태
    - 0.4~0.5에 데이터가 집중
  - 값
    - 최소 : 0.055
    - 최대 : 0.65

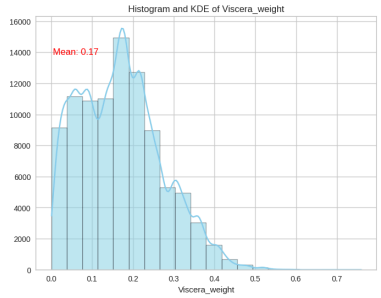
- **Height(전복 높이)**
  - **평균**
    - 0.14
  - **분포**
    - 0.1 근처에 데이터가 매우 집중
    - 데이터 대부분이 0.0 - 0.2 사이 분포
  - 값
    - 최소 : 0.004
    - 최대 : 1.13



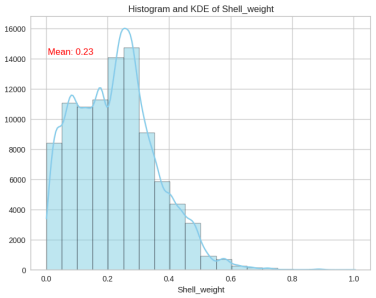


- **Whole_weight(전복 전체 무게)**
  - 평균
    - 0.79
  - 분포
    - 오른쪽 꼬리가 긴 분포
    - 0.5~1.0 구간에 대부분의 데이터가 밀집
    - 전체 무게가 약 1.0인 개체가 가장 많음
  - 값
    - 최소 : 0.002
    - 최대 : 2.8255

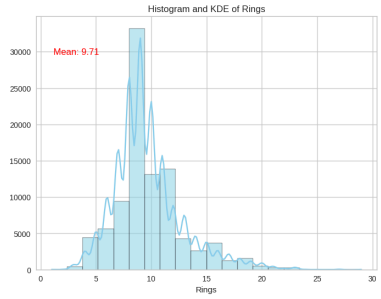
- **Shucked_weight(껍질 분리 무게)**
  - **평균**
    - 0.34
  - **분포**
    - 오른쪽 꼬리가 긴 분포
    - 0.2~0.4 구간에 대부분의 데이터가 밀집
    - 껍질 분리 무게 약 0.4인 개체가 가장 많음
  - 값
    - 최소 : 0.001
    - 최대 : 1.488



- **Viscera_weight(전복 내장 무게)**
  - 평균
    - 0.17
  - 분포
    - 오른쪽 꼬리가 긴 분포
    - 0.1~0.3 구간에 대부분의 데이터가 밀집
    - 내장 무게 약 0.2인 개체가 가장 많음
  - 값
    - 최소 : 0.0005
    - 최대 : 0.76

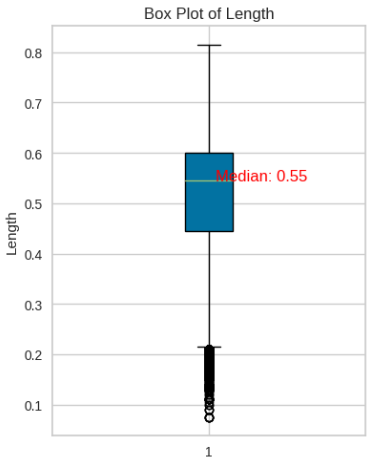


- **Shell_weight(전복 껍질 무게)**
  - 평균
    - 0.23
  - 분포
    - 오른쪽 꼬리가 긴 분포
    - 0.2 구간에 대부분의 데이터가 밀집
    - 껍질 무게 약 0.3인 개체가 가장 많음
  - 값
    - 최소 : 0.0015
    - 최대 : 1.005

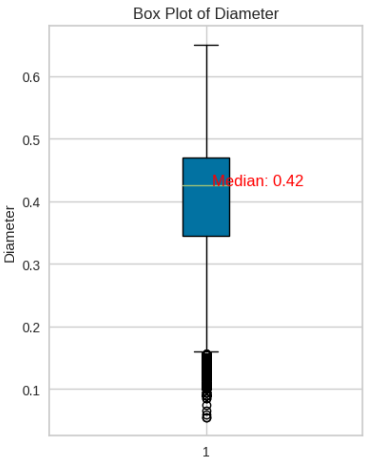


- **Rings(전복 연령)_Target**
  - 평균
    - 9.71
  - 분포
    - 오른쪽 꼬리가 긴 분포
    - 8~11 구간에 대부분의 데이터가 밀집
    - 고리가 약 10개인 개체가 가장 많음
  - 값
    - 최소: 1
    - 최대 : 29

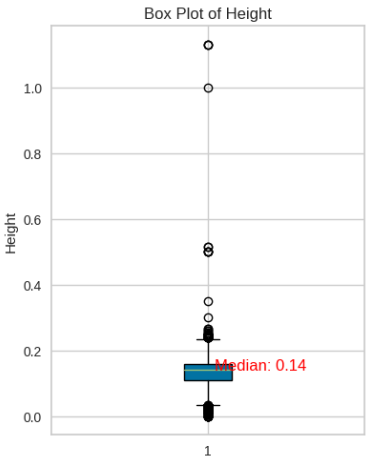
2_2. 단일 변수 분석(BoxPlot)



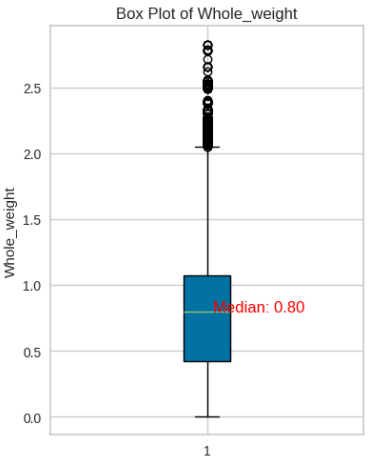
- **Length (전복의 가장 긴 부분 측정)**
  - 중앙값
    - 0.55
  - 이상치
    - 존재
    - 약 0.2 이하의 길이를 가진 전복
  - 분포
    - 대체로 대칭



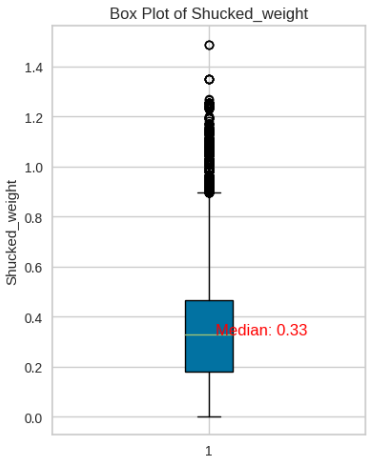
- **Diameter (전복 직경)**
  - 중앙값
    - 0.42
  - 이상치
    - 존재
    - 약 0.2 이하의 지름을 가진 전복
  - 분포
    - 대체로 대칭



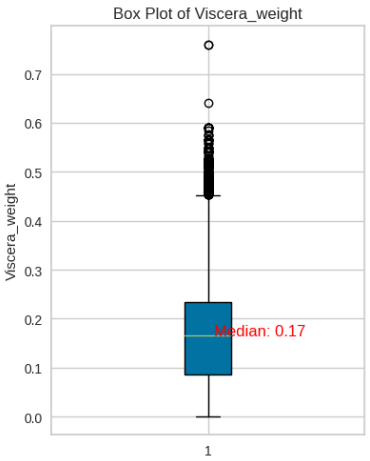
- **Height(전복 높이)**
  - 중앙값
    - 0.14
  - 이상치
    - 존재
    - 0과 근사치, 0.2 이상의 높이를 가진 전복
  - 분포
    - 왼쪽 꼬리가 긴 분포



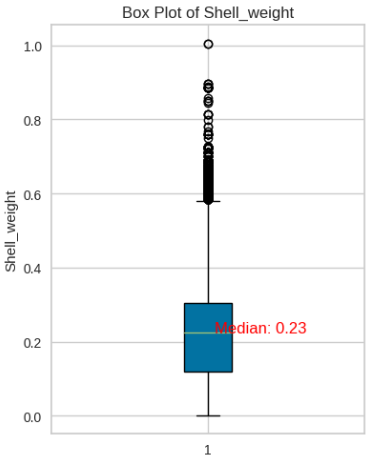
- **Whole_weight(전복 전체 무게)**
  - 중앙값
    - 0.80
  - 이상치
    - 존재
    - 약 2.0 이상의 전체 무게를 가진 전복
  - 분포
    - 오른쪽 꼬리가 긴 분포



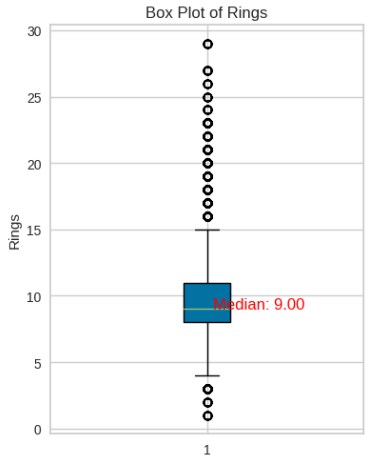
- **Shucked_weight(껍질 분리 무게)**
  - 중앙값
    - 0.33
  - 이상치
    - 존재
    - 약 0.8 이상의 껍질 분리 무게를 가진 전복
  - 분포
    - 오른쪽 꼬리가 긴 분포



- **Viscera_weight(전복 내장 무게)**
  - 중앙값
    - 0.17
  - 이상치
    - 존재
    - 약 0.4 이상의 내장 무게를 가진 전복
  - 분포
    - 오른쪽 꼬리가 긴 분포



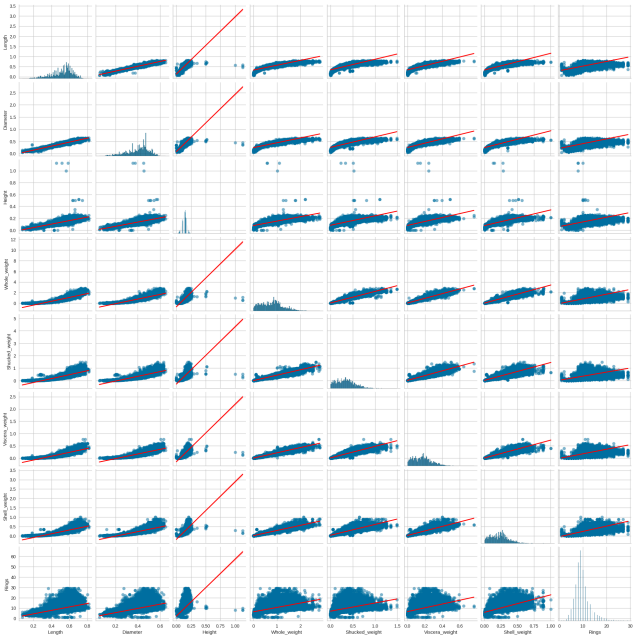
- **Shell_weight(전복 껍질 무게)**
  - 중앙값
    - 0.23
  - 이상치
    - 존재
    - 약 0.6 이상의 껍질 무게를 가진 전복
  - 분포



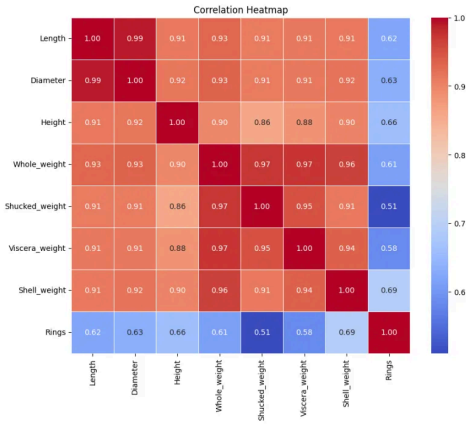
- **Rings(전복 연령)_Target**
  - 중앙값
    - 9.00
  - 이상치
    - 존재
    - 약 5 이하, 15 이상의 고리 개수를 가진 전복
  - 분포

- 오른쪽 꼬리가 긴 분포
- 대체로 대칭

2_3. 변수 간 관계 분석(ScatterPlot, Heatmap)



- 모든 변수가 양의 상관 관계
- Rings
  - Length
    - 상대적으로 큰 분산
  - Diameter
    - 상대적으로 큰 분산
  - Height
    - 밀집된 분포를 보이나, 직접적인 상관관계
- Height는 다른 모든 변수와 데이터 포인트가 밀집되어있음
- 논리적으로 연관이 있는 변수가 상관 관계 높은 경향
  - length, diameter
  - All Weight (Whole, Shucked, Viscera, Shell)
- Rings는 Height 제외한 피처데이터가 상대적으로 분산 되어있음



→ 모두 강한 양의 상관관계를 가짐

- **Length**
  - 길이가 길 수록 직경, 높이, 무게가 높은 수치를 나타냄 (피처 간의 수치가 비례함)
  - 길이가 길 수록 전복의 고리의 수가 많음
- **Diameter**
  - 직경이 클 수록 길이, 높이, 무게가 높은 수치를 나타냄 (피처 간의 수치가 비례함)
  - 직경이 클 수록 전복의 고리의 수가 많음
- **Height**
  - 높이가 높을 수록 길이, 직경, 무게가 높은 수치를 나타냄 (피처 간의 수치가 비례함)
  - 높이가 높을 수록 전복의 고리의 수가 많음
- **All Weight (Whole, Shucked, Viscera, Shell)**
  - 무게가 클 수록 길이, 직경, 높이가 높은 수치를 나타냄 (피처 간의 수치가 비례함)
  - 무게가 클 수록 전복의 고리의 수가 많음

. 전처리

3_1. 변수 제거

- id : 아이디

3_2. 이상치

: 논리적으로 존재할 수 없는 데이터 제거

설명	총 개수	내장 무게 > 전체 무게	껍질 제외 무게 > 전체 무게	껍질 무게 > 전체 무게	전복 특성 변수 = 0
train	63	2	45	8	8
test	51	2	38	9	2

- train과 test에 존재하는 이상치의 개수가 거의 유사
  - train에 존재하는 이상치 제거

3_3. 로그 변환

: 히스토그램 그래프를 통해 로그 변환 판단

- All Weight
  - Whole_weight
  - Shucked_weight
  - Viscera_weight
  - Shell_weight

3_4. Scailing

- Robust Scailing
  - 이상치의 영향을 최소화 하기 위해 사용
  - Sex와 Rings(target)를 제외한 모든 변수

3_5. Encoding

- dummy 변수 생성
  - Sex_I
  - Sex_F
  - Sex_M

. 분석 모델링

4_1. 데이터 분리

- train = 95%
- test = 5%

4_2. 사용 모델 결정

- AutoML : 모델 개발 작업 자동화

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	1.247	3.404	1.845	0.664	0.150	0.126	14.738
lightgbm	Light Gradient Boosting Machine	1.255	3.456	1.859	0.659	0.151	0.127	5.286
xgboost	Extreme Gradient Boosting	1.260	3.502	1.871	0.655	0.152	0.127	0.712
gbr	Gradient Boosting Regressor	1.287	3.604	1.898	0.645	0.154	0.130	9.610
rf	Random Forest Regressor	1.287	3.572	1.890	0.648	0.155	0.132	45.678

  - **CatBoost**
  - **LightGBM**
  - **XGBoost**

- **GBR**
- RandomForest
  - 파이널 프로젝트 당시 RF의 RMSLE(0.1506)가 다른 알고리즘에 비해 높게 나타났으며, 소요 시간이 길어 비효율적인 알고리즘으로 판단

4_3. 하이퍼 파라미터 최적화(단일 모델 score)

- **Optuna + StratifiedKFold : AWS에서 제공하는 모델 별 하이퍼 파라미터 목록 사용**
  - **CatBoost Hyper Parameters**

```
Best RMSLE: 0.1495001562545041
Best hyperparameters:
  max_depth: 7
  subsample: 0.9865994295496124
  min_data_in_leaf: 43
  learning_rate: 0.10626171034559712
  n_estimators: 485
  l2_leaf_reg: 3
  random_strength: 0.8817285905269581
  rsm: 0.8420506888504059
  bagging_temperature: 2
  max_leaves: 127
  sampling_frequency: PerTree
  max_bin: 485
  thread_count: 2
```

  - **LightGBM Hyper Parameters**

```
Best RMSLE: 0.14958315313216322
Best hyperparameters:
  num_boost_round: 549
  max_depth: 11
  subsample: 0.271963602330989
  min_data_in_leaf: 83
  learning_rate: 0.056233765958497474
  n_estimators: 714
  l2_leaf_reg: 1
  feature_fraction: 0.8111960329373981
  bagging_fraction: 0.9449319850477282
  num_leaves: 48
  max_bin: 490
  thread_count: 2
  bagging_freq: 6
  max_delta_step: 8.845779023076112
  lambda_l1: 4.938299731440447
  lambda_l2: 4.572127810578763
  min_gain_to_split: 0.875054487071554
  feature_fraction_bynode: 0.9000396254058078
  num_threads: 11
  tree_learner: data
  tweedie_variance_power: 1.131534682185736
```

  - **XGBoost Hyper Parameters**

```
Best RMSLE: 0.1504762607666011
Best hyperparameters:
  max_depth: 10
  learning_rate: 0.0855578084718704
  n_estimators: 250
  subsample: 0.539502650596425
  num_round: 485
  alpha: 8.391484056276457
  gamma: 3.2237185871415996
  lambda: 8.582605846188228
  lambda_bias: 6.569575706287025
  max_leaves: 124
  min_child_weight: 4.784627032949729
  base_score: 5.29698073664351
  colsample_bylevel: 0.6278608058509046
  colsample_bynode: 0.9261929464846934
  colsample_bytree: 0.5963754416589548
  csv_weights: 0.011035973853488007
  eta: 0.5417543810547697
  tweedie_variance_power: 1.550169566483108
```

- **GradientBoostingRegressor Hyper Parameters**

```
Best RMSLE: 0.1514334035141312
Best hyperparameters:
  learning_rate: 0.18641287847558408
  n_estimators: 386
  subsample: 0.3639755898480478
  max_depth: 4
  min_samples_split: 7
  min_samples_leaf: 2
  min_weight_fraction_leaf: 0.006660890611244898
  min_impurity_decrease: 0.4802279957700628
  alpha: 0.20948280132044253
  max_leaf_nodes: 6
  validation_fraction: 0.5364580858808121
  max_features: 0.8665681638771795
```

4_4. Voting 모델링

- Voting 사용
  - Final Project 당시 분석 : 기본 전처리 + 로그 변환

모델	RMSLE
cat	0.14902
lgbm	0.14912
xgb	0.14920
gbr	0.14859
cat +lgbm +gbr	**0.14631**
cat + lgbm + xgb	**0.14684**
TOP 4	**0.14643**

  - 단일 모델들 보다 Voting을 진행한 모델들이 RMSLE 값이 현저히 낮음

- **Colab 성능**
  - 성능이 안 좋은 개별 모델이 추가될 수록 Voting 모델 점수 하락
  - 최고 성능
    - **교차 검증 X + Voting(CatBoost+LightGBM) : 0.1461**

모델	교차검증 X	K-Fold	Stratified K-Fold
Cat + LGB + XGB + GBR	**0.1466** ✅	**0.1492** ✅	0.1491
Cat + LGB + XGB	**0.1463** ✅	**0.1490** ✅	**0.1489** ✅
Cat + LGB	**0.1461** ✅	**0.1489** ✅	**0.1487** ✅
Cat + XGB	0.1467	0.1494	0.1492
Cat + GBR	0.1470	0.1495	**0.1490** ✅
LGB + XGB + GBR	0.1469	0.1496	0.1495

- **Kaggle 성능 <Private>**
  - 최고 성능
    - **Stratified K-Fold + Voting(CatBoost+LightGBM) : 0.14660**

모델	교차검증 X	K-Fold	Stratified K-Fold
Cat + LGB + XGB + GBR	0.14690 ✅	0.14699 ✅	
Cat + LGB + XGB	0.14675 ✅	0.14682 ✅	0.14662 ✅
Cat + LGB	0.14674 ✅	0.14680 ✅	**0.14660** ✅✅
Cat + XGB			
Cat + GBR			0.14678 ✅
LGB + XGB + GBR			

- **kaggle 성능 <public>**
  - 최고 성능
    - **K-Fold + Voting(CatBoost+LightGBM) : 0.14710**

모델	교차검증 X	K-Fold	Stratified K-Fold
Cat + LGB + XGB + GBR	0.14739 ✅	0.14757 ✅	
Cat + LGB + XGB	0.14725 ✅	0.14731 ✅	0.14728 ✅
Cat + LGB	0.14711 ✅	**0.14710** ✅✅	0.14718 ✅
Cat + XGB			
Cat + GBR			0.14741 ✅
LGB + XGB + GBR			

. Deep Learning

: Optuna와 같이 사용

5_1. MLP

1. **epoch=100 + optuna 활용**

- **Optuna**
  - 50 trial

    - **hyper parameter**
        - *dropout_rate :* 0.0 ~ 0.5
        - *optimizer_name :* Adam, RMSprop, SGD
        - *learning_rate :* 1e-5 ~ 1e-2
- **Dense**
    - 128
        - activation : Relu
    - 64
        - activation : Relu
- ***early_stopping_rounds***
    - patience=4
- **hyper parmeter**
    - *dropout_rate* : 0.1925116505889767
    - *optimizer_name* : Adam
    - *learning_rate* : 0.000685280087375688
- **Validation RMSLE**
    - 0.15210

2. **epoch=10, hyper parameter 고정**

- **hyper parameter**
    - *dropout_rate* : 0.1925116505889767
    - *optimizer_name* : Adam
    - *learning_rate* : 0.000685280087375688
- **Dense**
    - 128
        - activation : Relu
    - 64
        - activation : Relu
- **Validation RMSLE**
    - 0.15077

5_2. DNN

1. **epoch=100, optuna 활용**

- **Optuna**
    - 50 trial
    - **hyper parameter**
        - *dropout_rate :* 0.0 ~ 0.5
        - *optimizer_name :* Adam, RMSprop, SGD
        - *learning_rate :* 1e-5 ~ 1e-2
- **Dense**

    - 256
        - activation : Relu
    - 128
        - activation : Relu
    - 64
        - activation : Relu
    - 32
        - activation : Relu
    - 16
        - activation : Relu
- ***early_stopping_rounds***
    - patience=4
- **hyper parameter**
    - *dropout_rate* : 0.0714098671768656
    - *optimizer_name* : adam
    - *learning_rate* : 0.00011810487365492639
- **Validation RMSLE**
    - 0.15933

2. **epoch=10, optuna 활용**

- **Optuna**
    - 50 trial
    - **hyper parameter**
        - *dropout_rate :* 0.0 ~ 0.5
        - *optimizer_name :* Adam, RMSprop, SGD
        - *learning_rate :* 1e-5 ~ 1e-2
- **Dense**
    - 256
        - activation : Relu
    - 128
        - activation : Relu
    - 64
        - activation : Relu
    - 32
        - activation : Relu
    - 16
        - activation : Relu
- ***early_stopping_rounds***
    - patience=4
- **hyper parameter**

- *dropout_rate* : 0.11223936580456312
- *optimizer_name* : adam
- *learning_rate* : 0.0003178594873943924

- **Validation RMSLE**
  - 0.14992

5_3. 딥러닝 모델 정리

모델	epoch=100 + optuna (MLP)	epoch=10, hyper parameter(epoch=1000 사용 (MLP)	epoch=100 + optuna (DNN)	epoch=10 + optuna (DNN)
Colab	0.15210	0.15077	0.15933	0.14992
Kaggle private		0.15183		**0.15054** ✅
kaggle public		0.15343		0.15138