

7조

파이널 프로젝트

캐글 경진 대회

김수지

박지은

변진영

이소희

이정수

CONTENTS

01 이진 분류 분석

Binary Classification with a Bank Churn Dataset

02 회귀 분석

Regression with an Abalone Dataset

01

CHAPTER

이진 분류 분석

분류 분석 목차

01 데이터 이해

02 데이터 전처리

03 데이터 분석

04 모델 평가

05 회고

개요

배경

디지털 전환으로 인한 업종 간 **경계 모호**, **경쟁 심화**
단순 금융상품 제조자로 전략

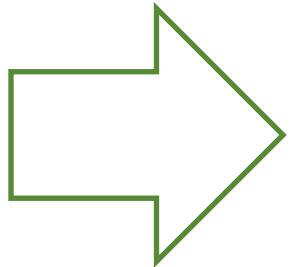
목적

예측 모델을 통한 **이탈 최소화**
이탈 발생 상황을 예측한 경쟁력 유지

데이터 및 영향도 예측

Feature

id	식별 번호
CustomerId	고객의 고유 식별 번호
Surname	고객의 성(씨)
Credit Score	고객의 신용 점수
Geography	고객이 거주하는 국가
Gender	고객의 성별
Age	고객의 나이
Tenure	고객이 은행을 이용한 기간
Balance	고객의 계좌 잔액
NumOfProducts	고객이 이용하는 은행 상품의 수
HasCrCard	신용카드 보유 여부
IsActiveMember	활성 회원 여부
EstimatedSalary	고객의 예상 연봉



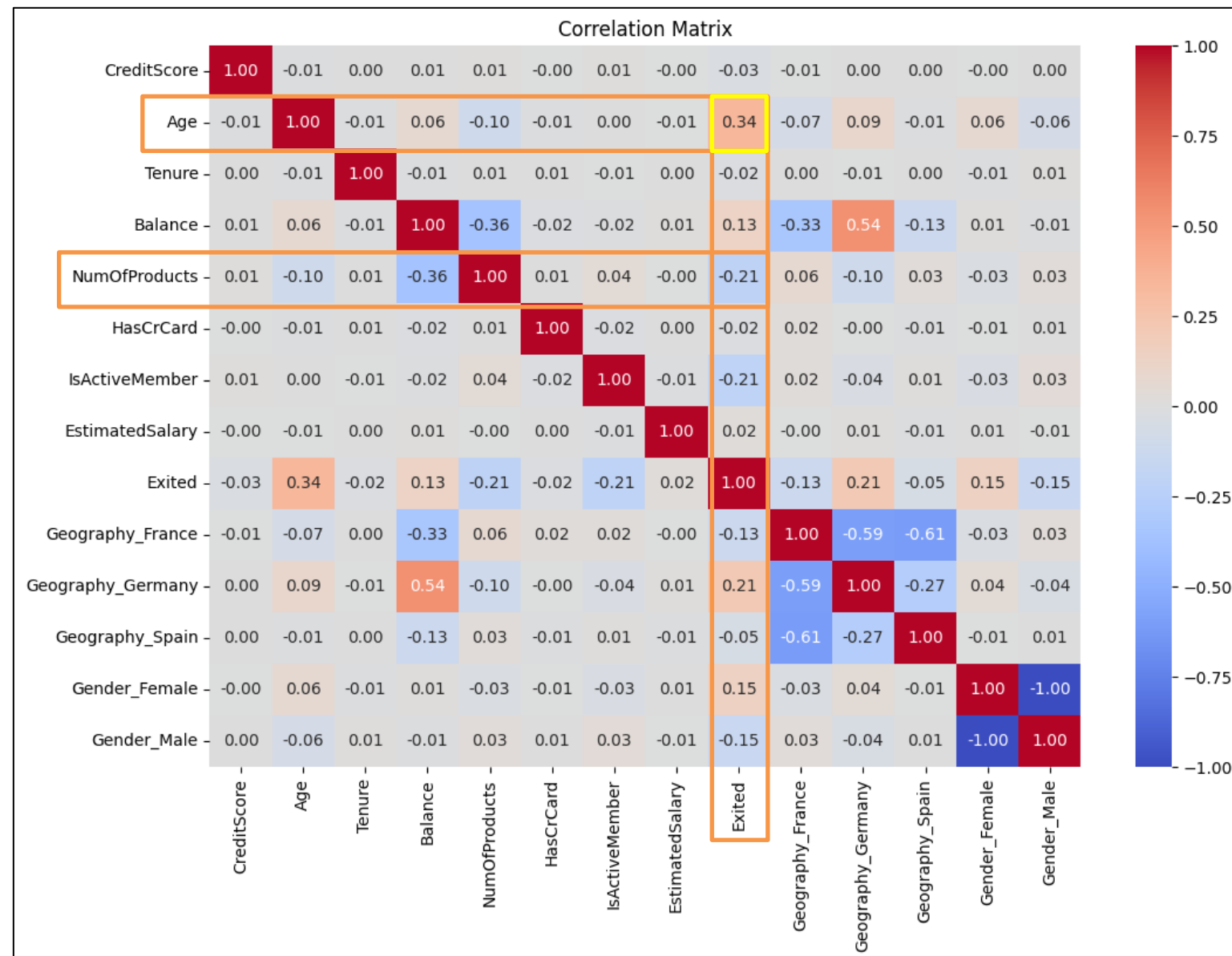
영향도 예측

Impact Feature

영향도 ↑	
Credit Score	고객의 신용 점수
Age	고객의 나이
Tenure	고객이 은행을 이용한 기간
NumOfProducts	고객이 이용하는 은행 상품의 수
HasCrCard	신용카드 보유 여부
EstimatedSalary	고객의 예상 연봉
영향도 ↓	
id	식별 번호
CustomerId	고객의 고유 식별 번호
Surname	고객의 성(씨)
Geography	고객이 거주하는 국가
Balance	고객의 계좌 잔액
IsActiveMember	활성 회원 여부

데이터 상관 관계

Heatmap



변수 간의 상관 계수

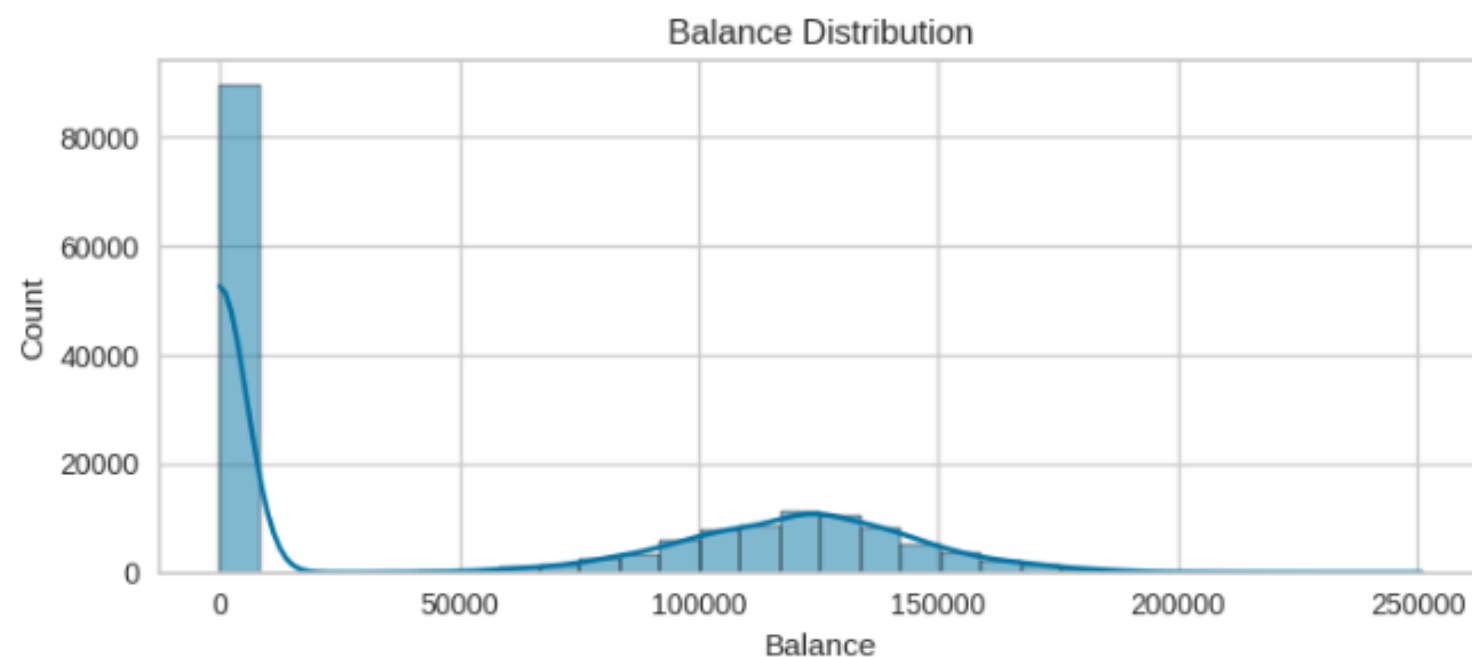
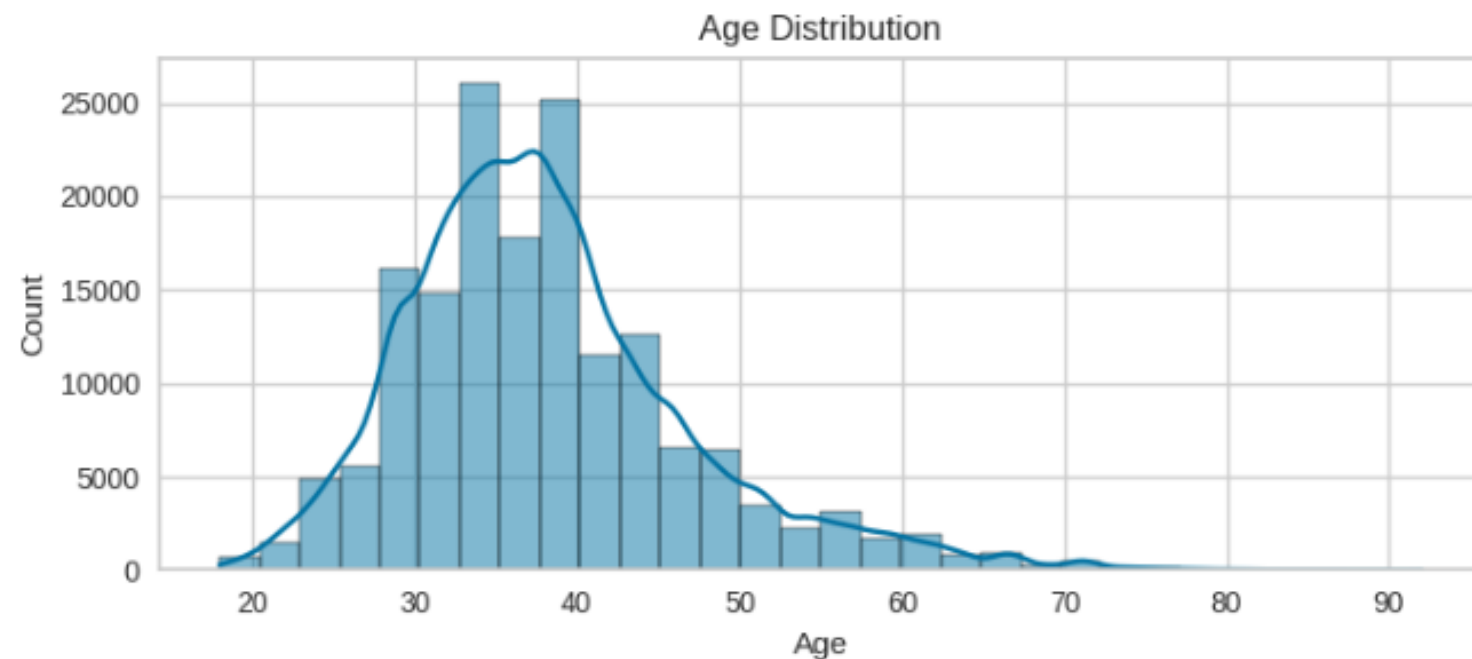
Exited <-> Age
: 0.34

Exited <-> Balance
: 0.13

Exited <-> NumOfProducts
: -0.21

Exited <-> IsActiveMember
: -0.21

데이터 전처리 - 공통 파생 변수 생성



Age :

전체 데이터 165034개 중

age의 60대 이상 -> 4016개

데이터의 크기 비율 조정위해 파생변수 생성 : **age_group**

20대 이하, 30대, 40대, 50대 이상

Balance :

전체 데이터 165034개 중

Balance = 0 인 값 -> 89648

데이터의 크기 비율 조정 위해 파생변수 생성 : **Balance_group**

Balance = 0 : 0

Balance != 0 : 1

파생 변수 생성

Generate

1. CreditScore By Age
나이대 별 신용점수

2. TenureByAge
연령 별 고객이 은행을 이용한 기간

3. Engagement
은행 상품의 수 + 신용카드 보유 여부 + 활성 회원 여부

4. CreditScore By (Balance or Balance_group)
잔고 별 신용점수

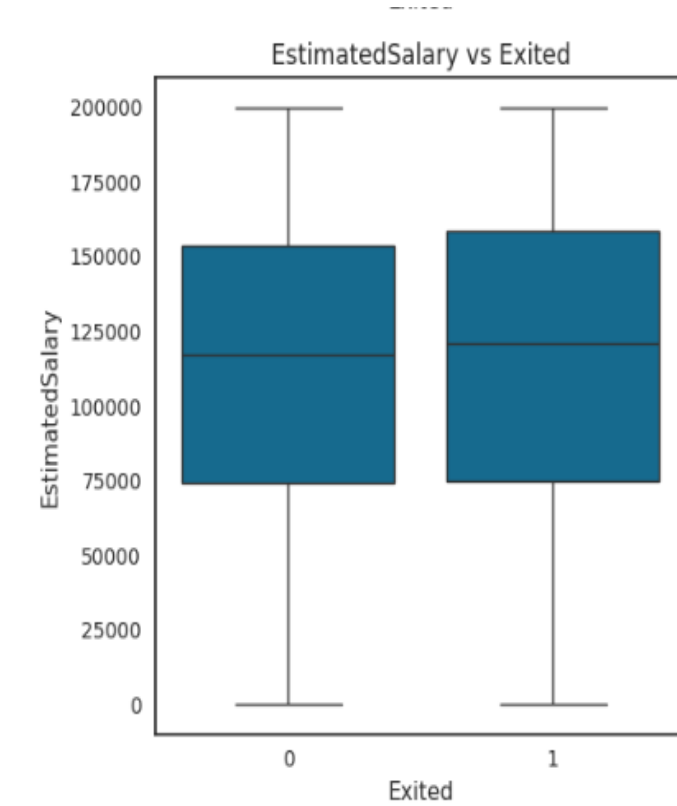
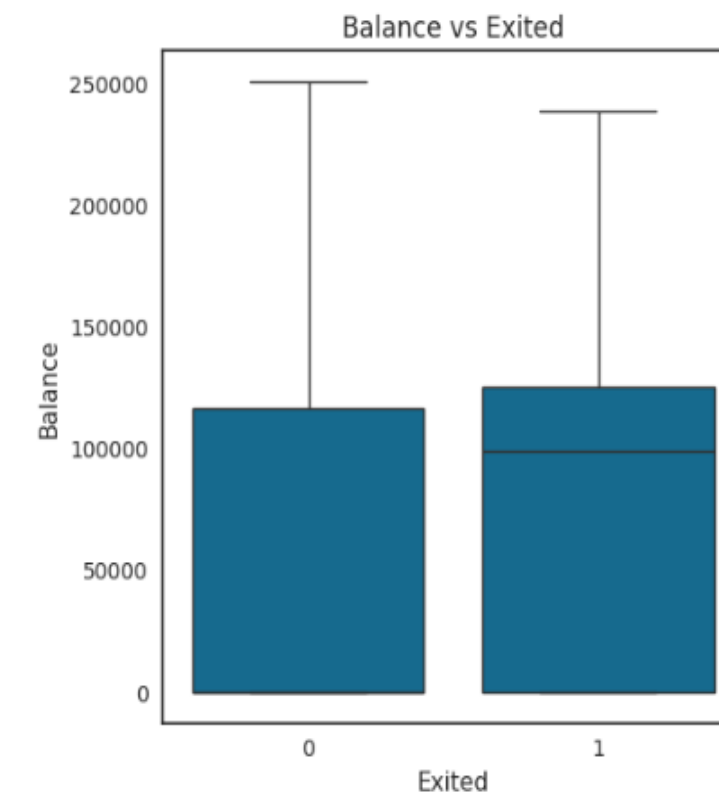
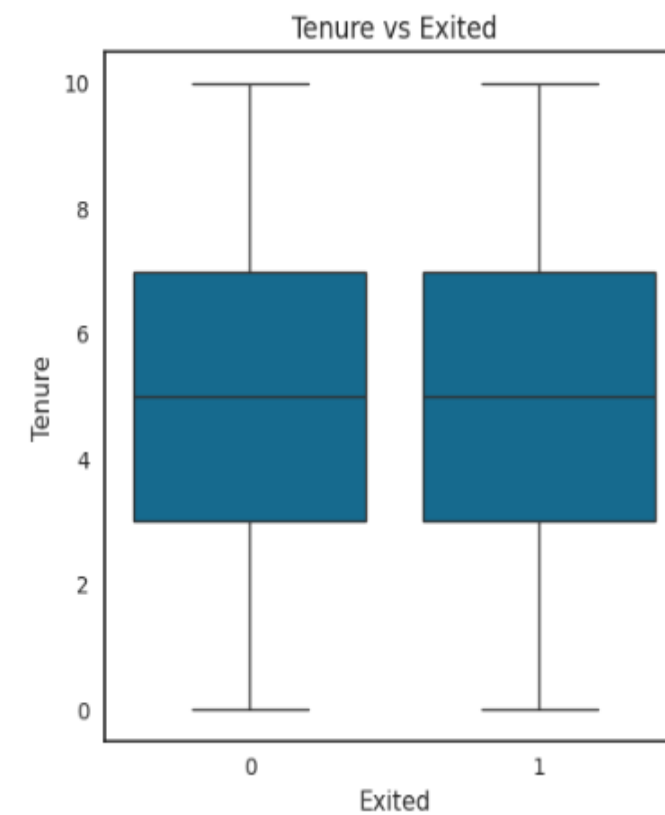
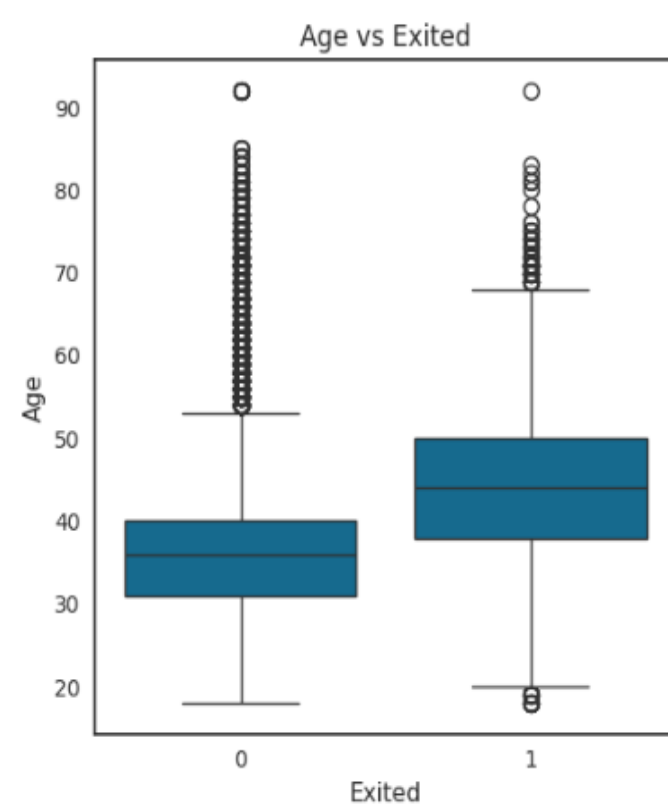
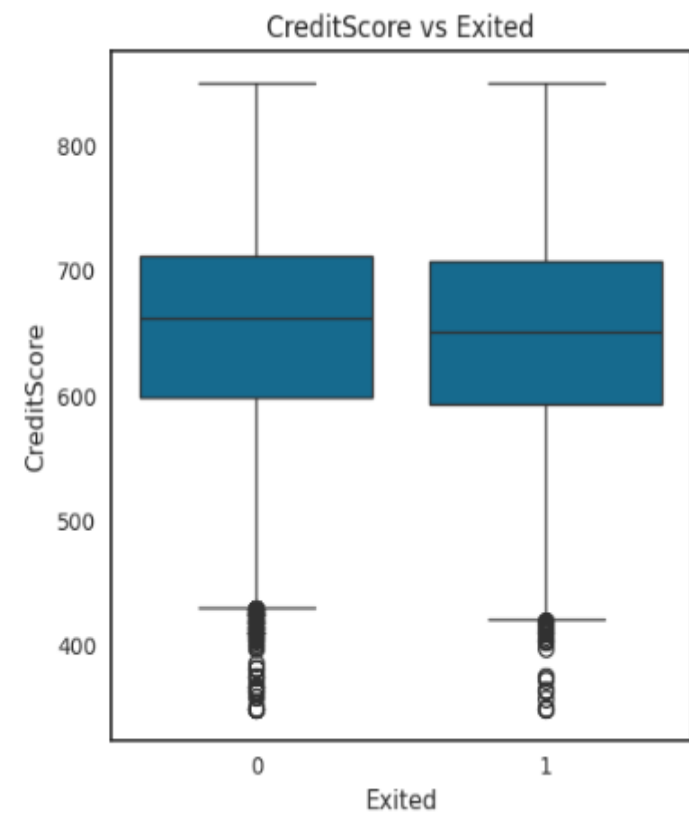
5. Balance By Age
연령 별 잔고

6. NumOfProducts By (Balance or Balance_group)
잔고 별 은행 상품의 수

전처리

- | | |
|-----------|---|
| 1. 변수제거 | Id, CustomerId, Surname |
| 2. 중복값 제거 | Id, CustomerId 제거 후 중복치 제거 |
| 3. 변수 변환 | HasCrCard, IsActiveMember,Balance, Age, EstimatedSalary |
| 4. 인코딩 | Gender, Geography |
| 5. 스케일링 | EDA 시각화 이상치를 통해 확인 |

전처리 - 스케일링 (EDA 시각화 분석)



Robust

- CreditScore, Age

MinMax

- Tenure, EstimatedSalary

StandardScaler

- Balance

AutoML을 통한 알고리즘 선택 (AUC 평가 기준)

Model		Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.864	0.887	0.548	0.741	0.630	0.549	0.558	7.680
catboost	CatBoost Classifier	0.863	0.886	0.540	0.744	0.626	0.545	0.555	32.350
gbc	Gradient Boosting Classifier	0.863	0.885	0.526	0.750	0.618	0.538	0.550	20.178
xgboost	Extreme Gradient Boosting	0.862	0.883	0.546	0.732	0.625	0.543	0.551	1.312
ada	Ada Boost Classifier	0.858	0.878	0.518	0.735	0.607	0.524	0.536	4.024

데이터 공통 전처리

+

공통 파생 변수

age_group, Balance_group

+

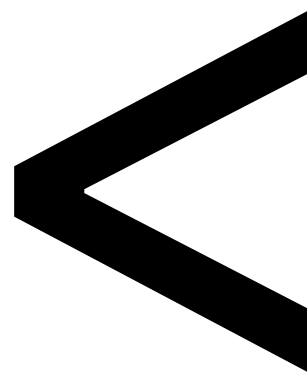
추가 파생 변수

CreditScoreByAge, TenureByAge, BalanceByAge Engagement
CreditScoreByBalance, NumOfProductsByBalance

알고리즘에 영향을 주는 변화 찾기 - 파생 변수

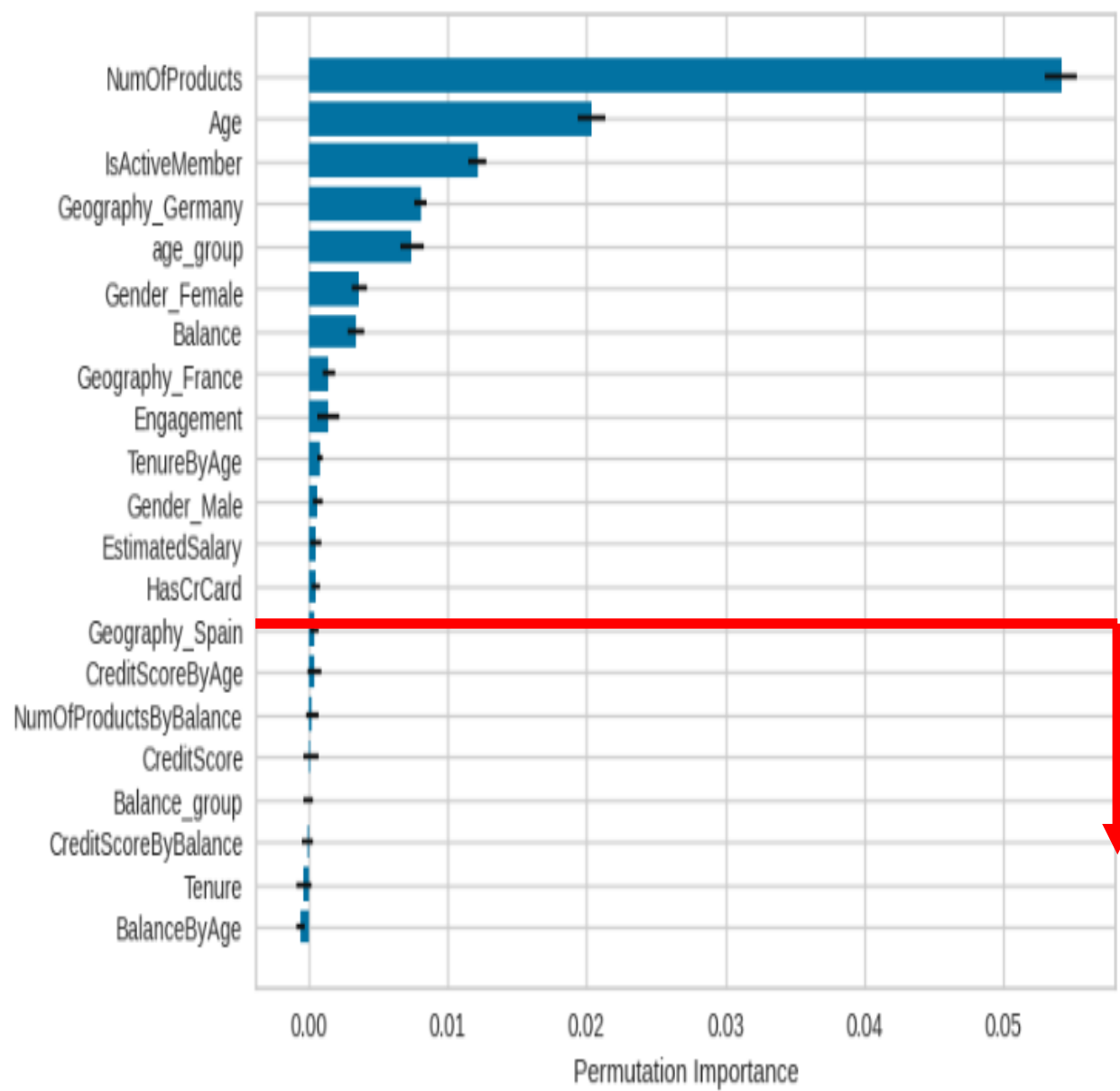
전처리

lightGBM	0.8884
CatBoost	0.8876
GradientBoosting	0.8878
XGBoost	0.8832
AdaBoost	0.8793

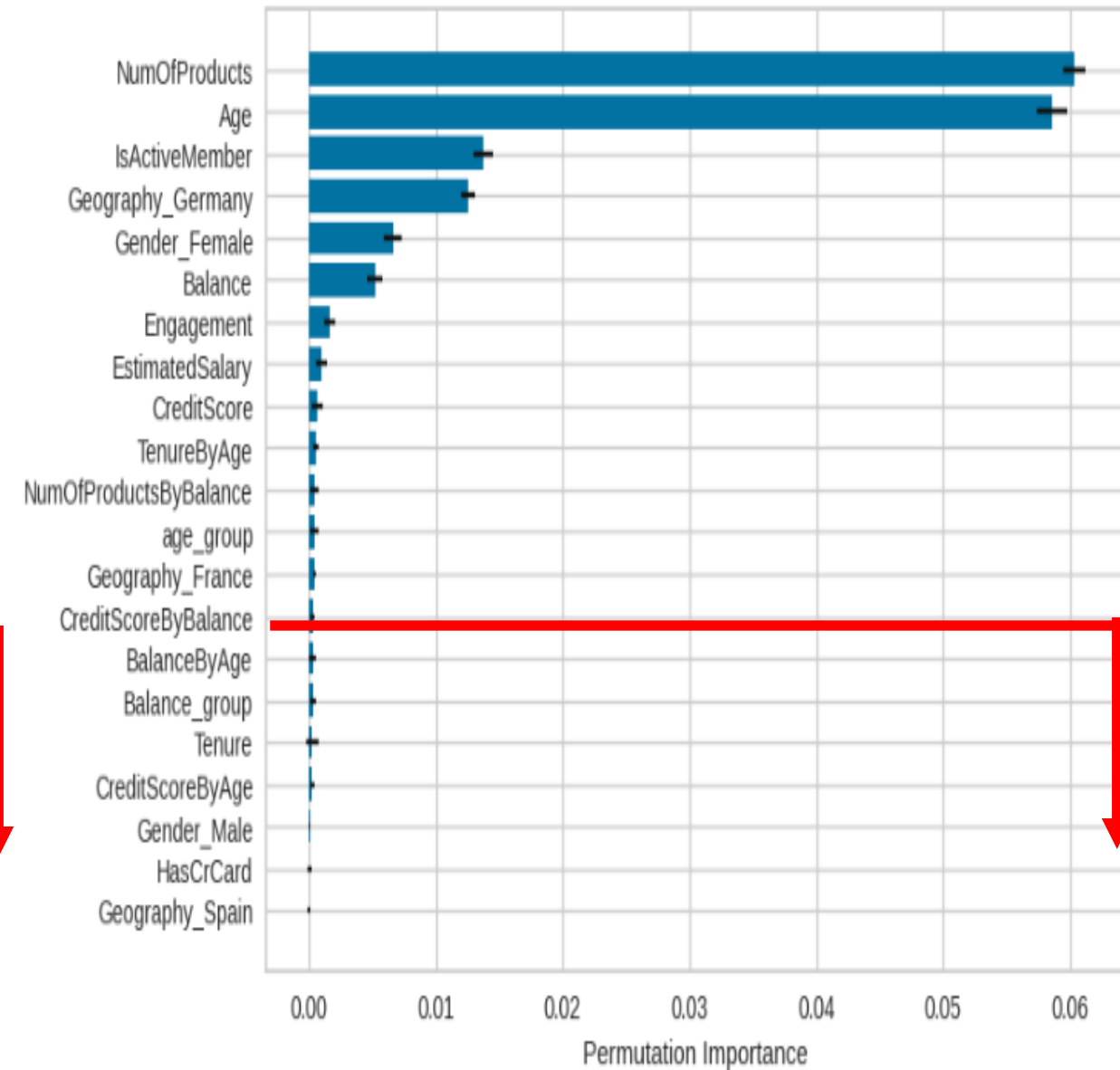


전처리 + 파생변수

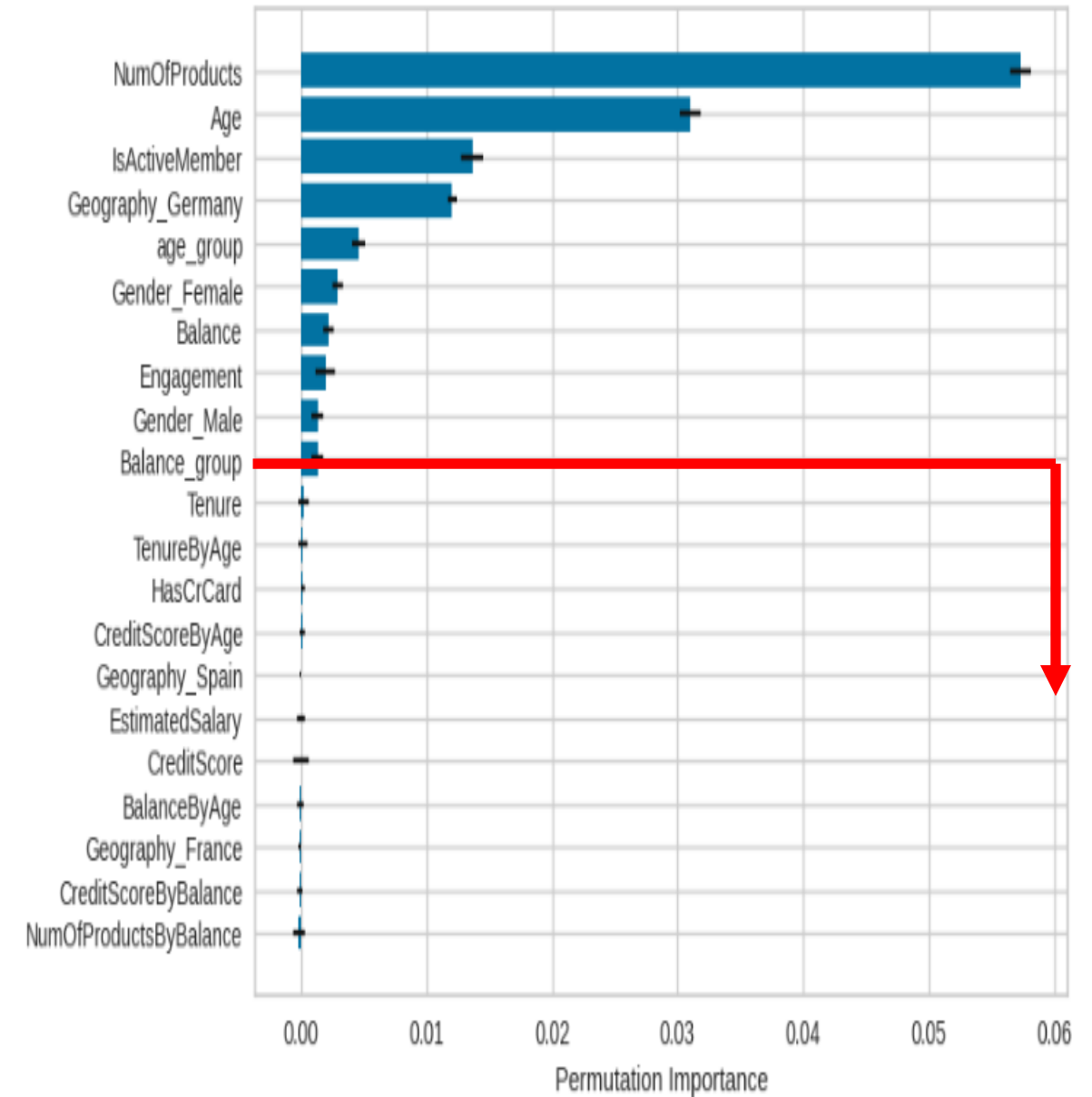
lightGBM	0.8918
CatBoost	0.8914
GradientBoosting	0.8905
XGBoost	0.8917
AdaBoost	0.8834



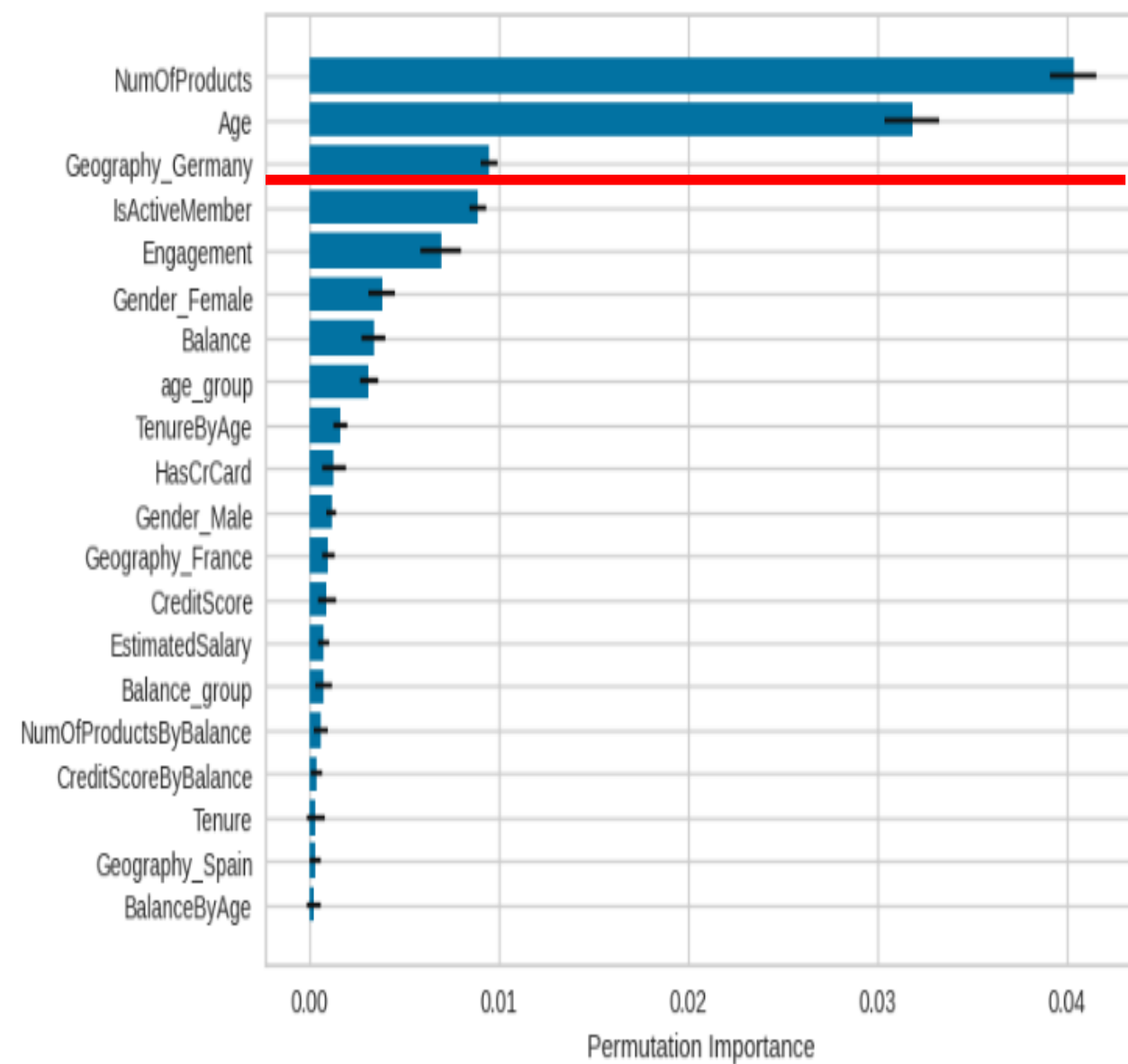
LightGBM



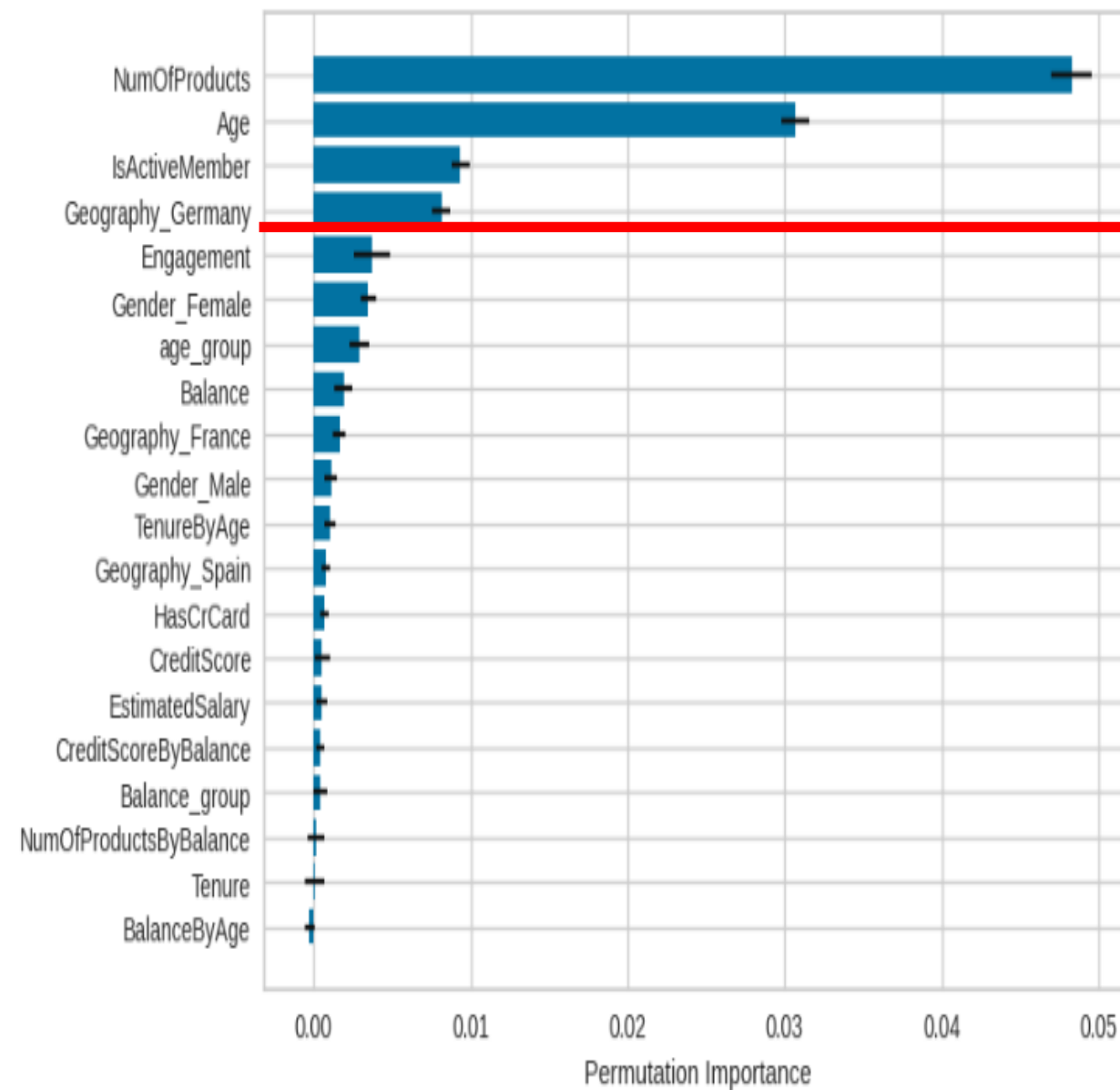
XGBoost



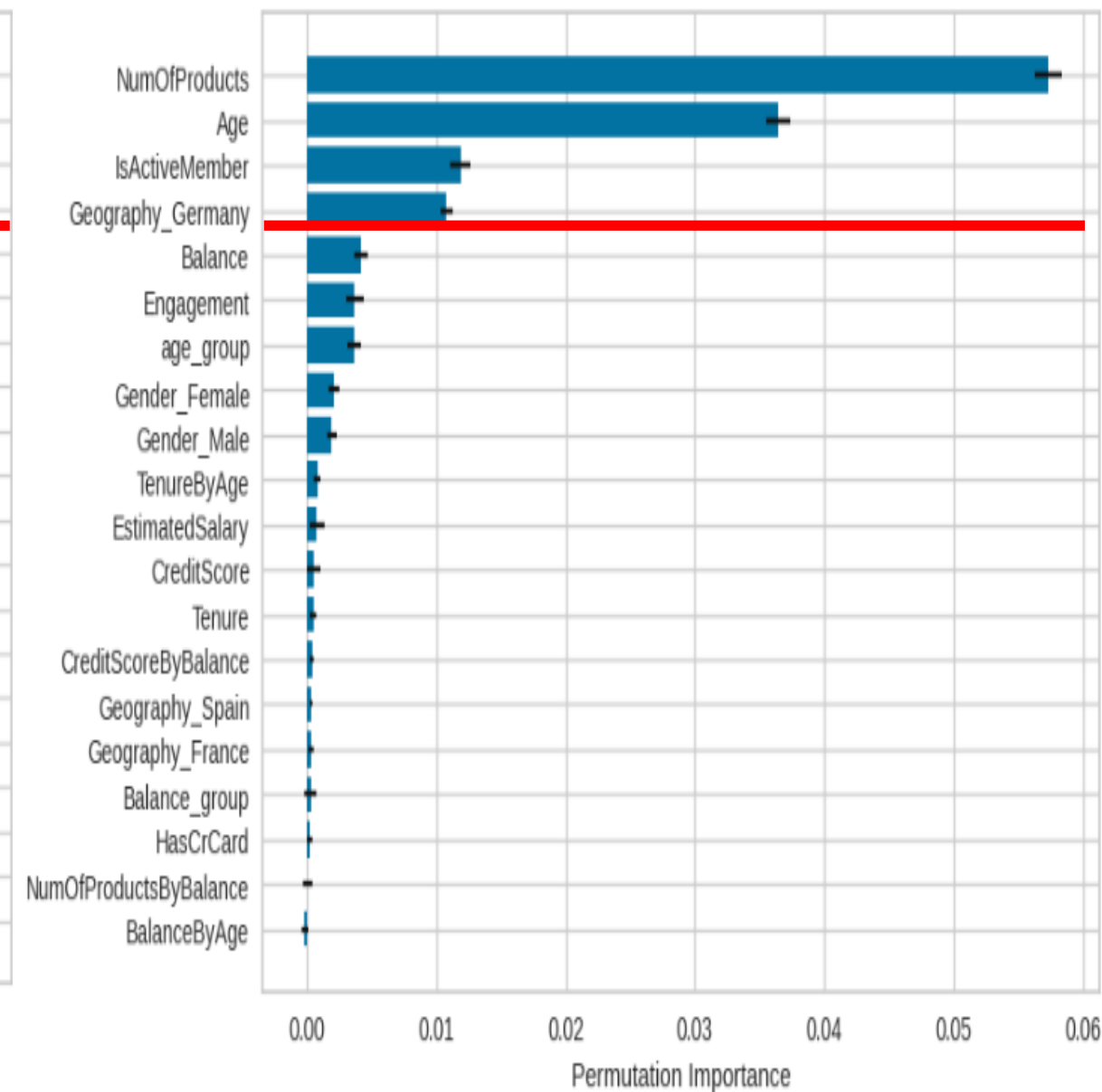
GradientBoost



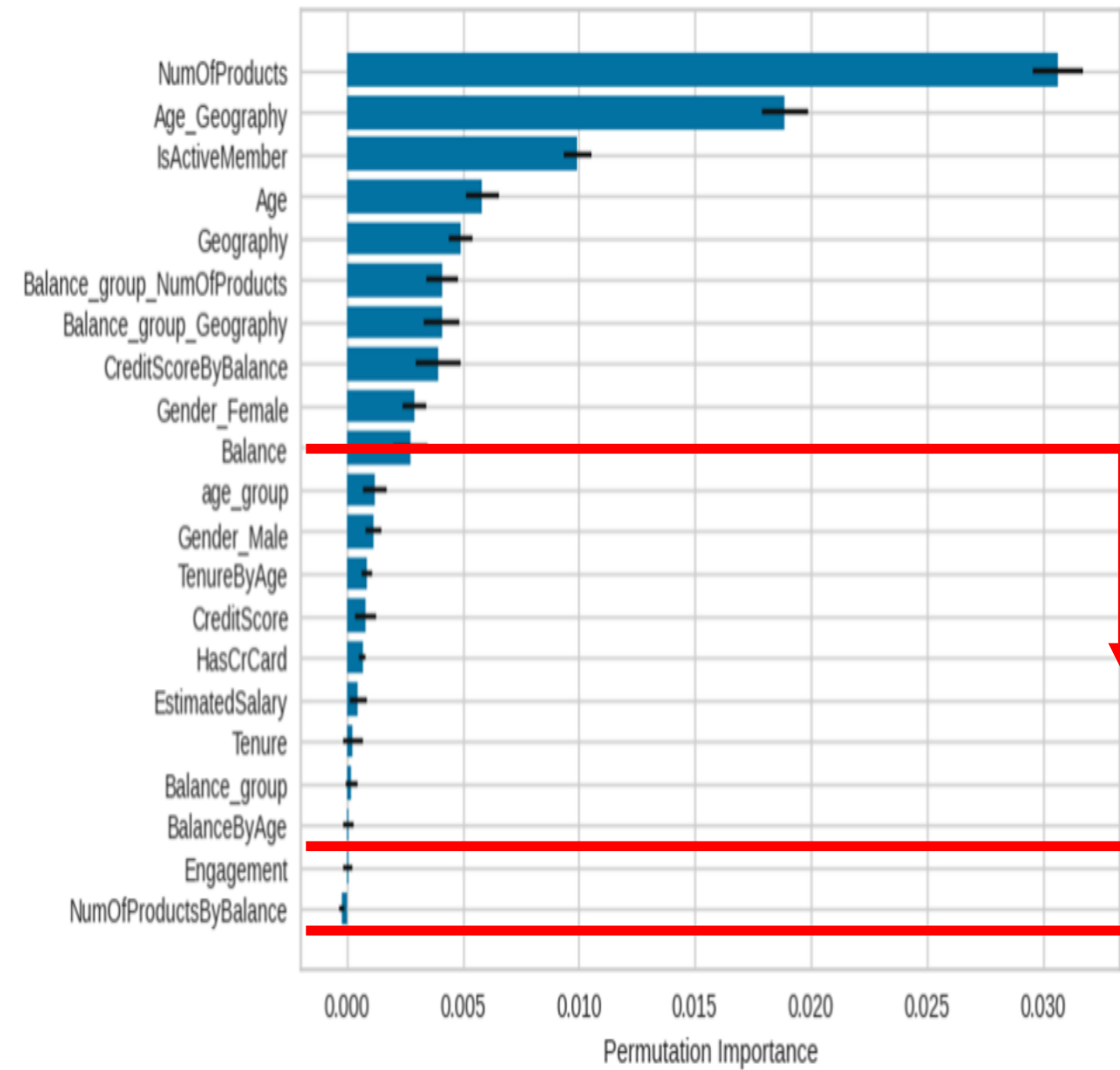
LightGBM



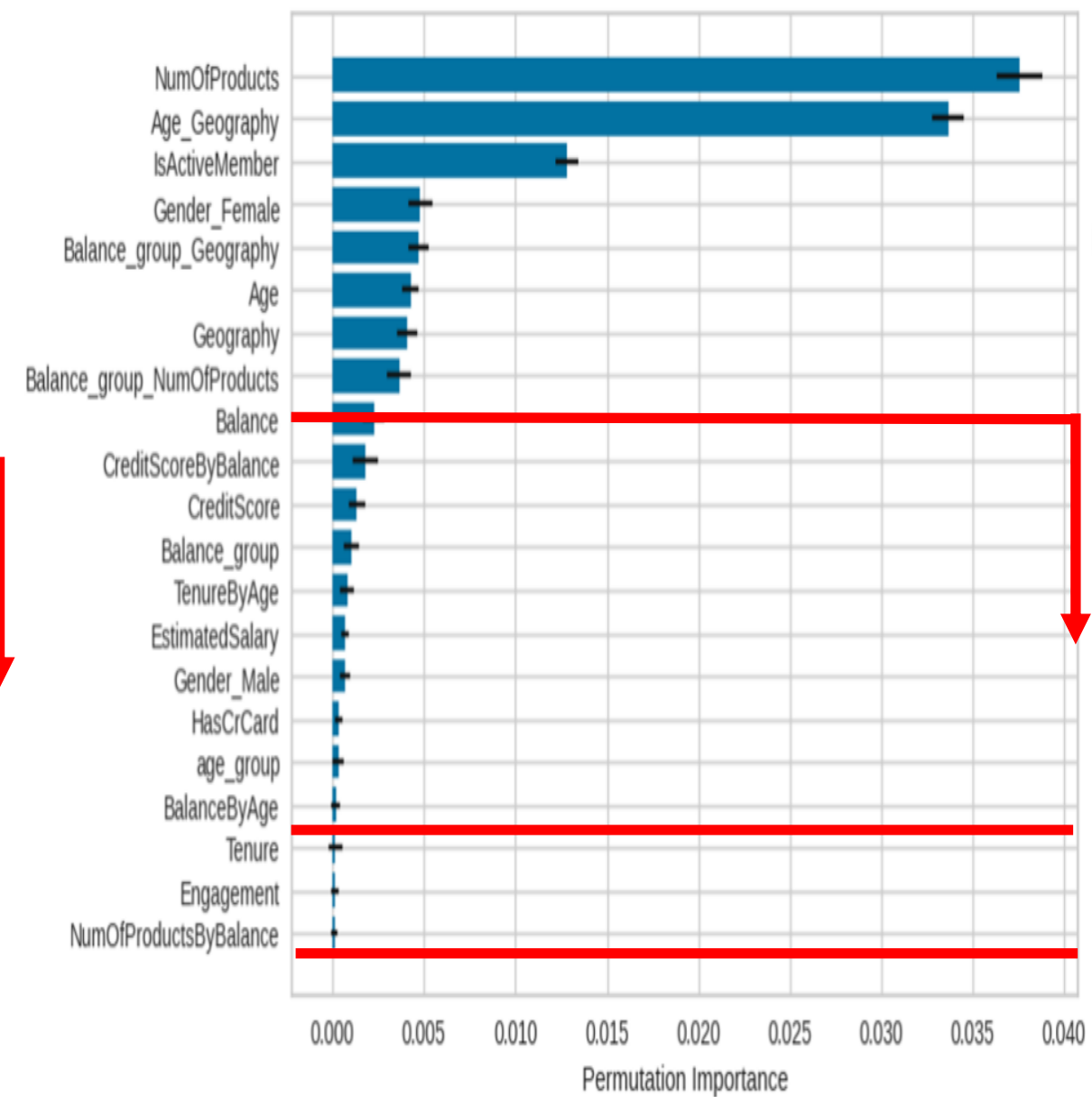
XGBoost



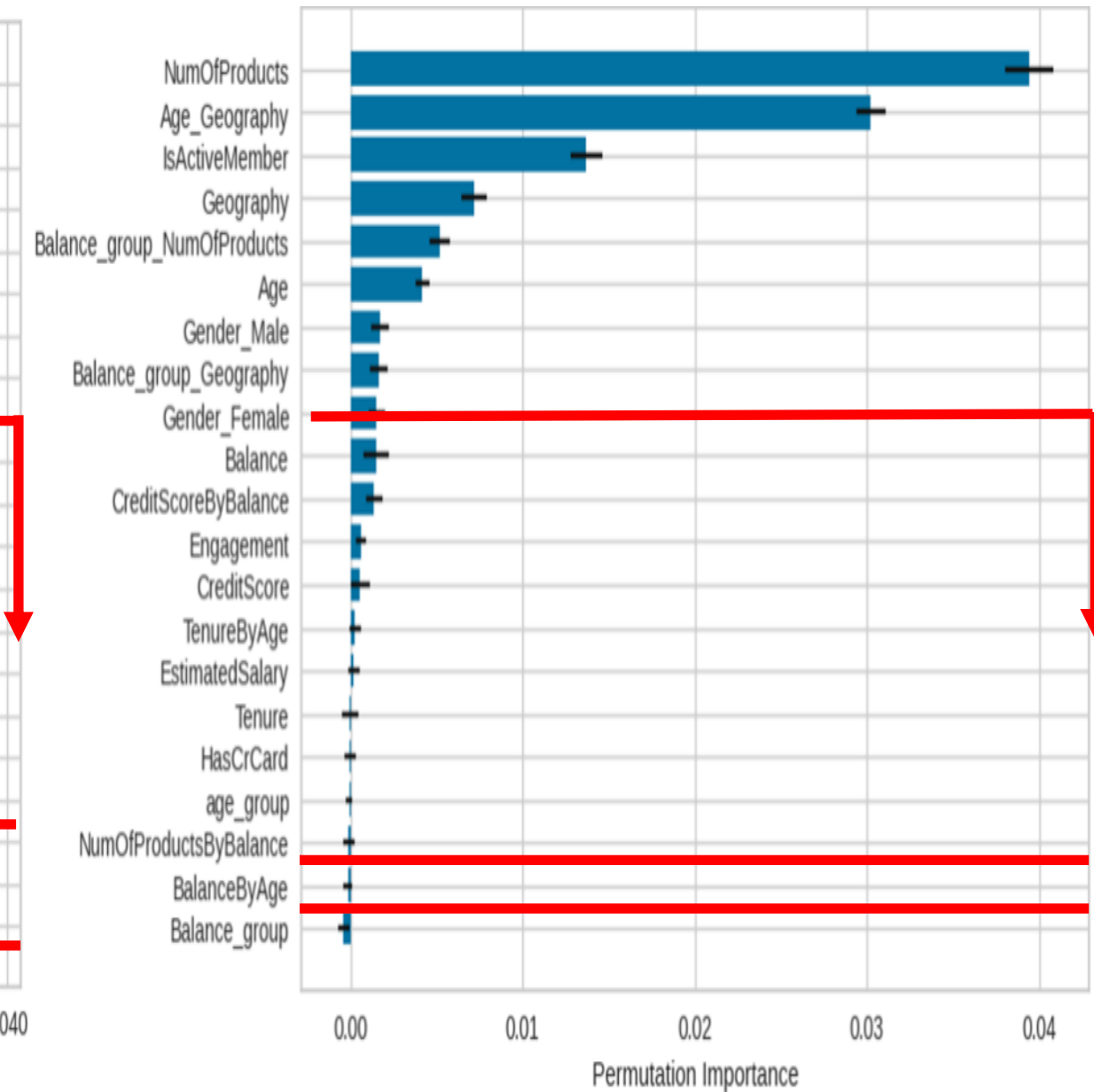
GradientBoost



LightGBM



XGBoost



CatBoost

모델 성능 향상을 위해 사용한 방법 - 파생변수 생성

방법 4

기본 공통 전처리
+
공통 파생변수(age_group + Balance_group)
+
모델 3에서 파생변수(BalanceByAge) 제거

lightGBM	0.8918(0.88773)
AdaBoost	0.8884
CatBoost	0.8915
XGBoost	0.8918
GradientBoost	0.8911

방법 5

기본 공통 전처리
+
공통 파생변수(age_group + Balance_group)
+
모델 3에서 파생변수(NumOfProductsByBalance) 제거

lightGBM	0.8918(0.88685)
AdaBoost	0.8848
CatBoost	0.8914
XGBoost	0.8916
GradientBoost	0.8915

모델 성능 평가 : 각 모델 보팅 성능 비교

알고리즘	LGB	ADA	CAT	XGB	GBC	보팅 (각 방법 중 Score Top3) - 캐글 기준
방법 1	0.8918	0.8834	0.8913	0.8917	0.8915	0.88793 (LGB, XGB, GBC)
방법 2	0.8918	0.8832	0.8916	0.8919	0.8919	0.88799 (LGB, XGB, GBC)
방법 3	0.8917	0.8883	0.8914	0.8916	0.8908	0.88823(LGB, CAT, XGB)
방법 4	0.8918	0.8884	0.8915	0.8918	0.8911	0.88776 (LGB, XGB, GBC)
방법 5	0.8918	0.8848	0.8914	0.8916	0.8915	0.88788 (LGB, XGB, GBC)

은행 고객 이탈 분석 결론

고객 이탈에 영향 있을 것 같은 요인 예상

Age (고객의 나이)

은행이 고객 이탈을 방지하기 위해서
Gender (고객의 성별)
NumOfProducts (고객이 이용하는 은행 상품의 수)

CreditScore (고객의 신용 점수)

연령별 적용가능한 마케팅 계획 수립
HasCrCard (신용카드 보유 여부)
EstimatedSalary (고객의 예상 연봉)

고객 이탈에 영향 없을 것 같은 요인 예상

Id (식별 번호)

연령별 다양한 은행 상품 수립
CustomerId (고객 식별 번호)

Surname (고객의 성(씨))

Geography (고객이 거주하는 국가)

고객 참여와 주인의식 고취
Balance (고객의 계좌 잔액)

IsActiveMember (활성회원여부)

이번 은행 고객 이탈 분석의 한계점 & 회고

1. 중복값 처리에 대한 방향성
2. SVD (특이값 분해) 처리 역량

02

CHAPTER

회귀 분석

회귀 분석 목차

01 대회 선정

02 데이터 소개 및 EDA

03 데이터 전처리

04 데이터 분석 결과

05 회귀 모델 평가 및
제출 결과

06 인사이트 제안

전복 데이터 회귀 분석 대회



[REGRESSION WITH AN ABALONE DATASET]

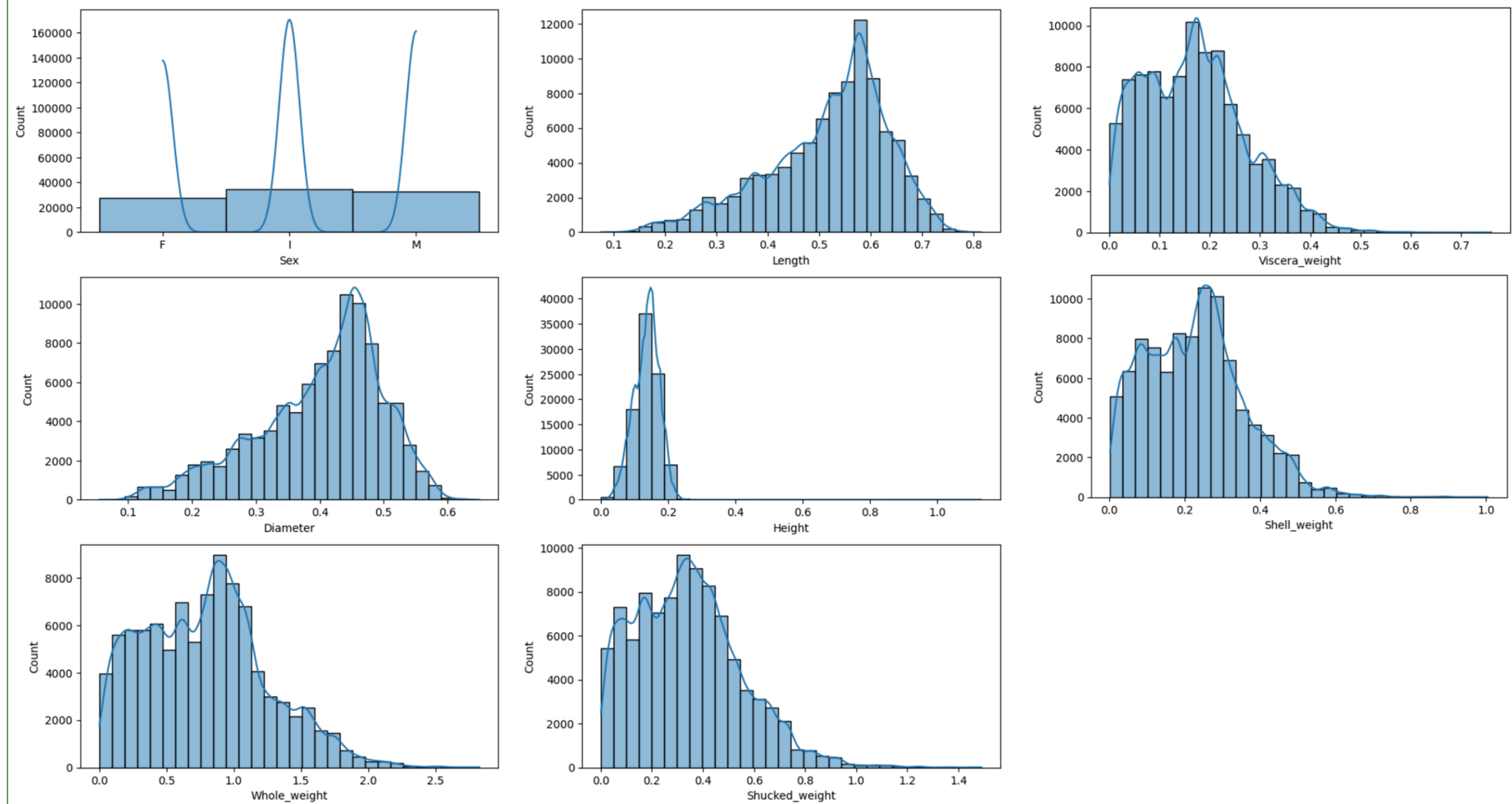
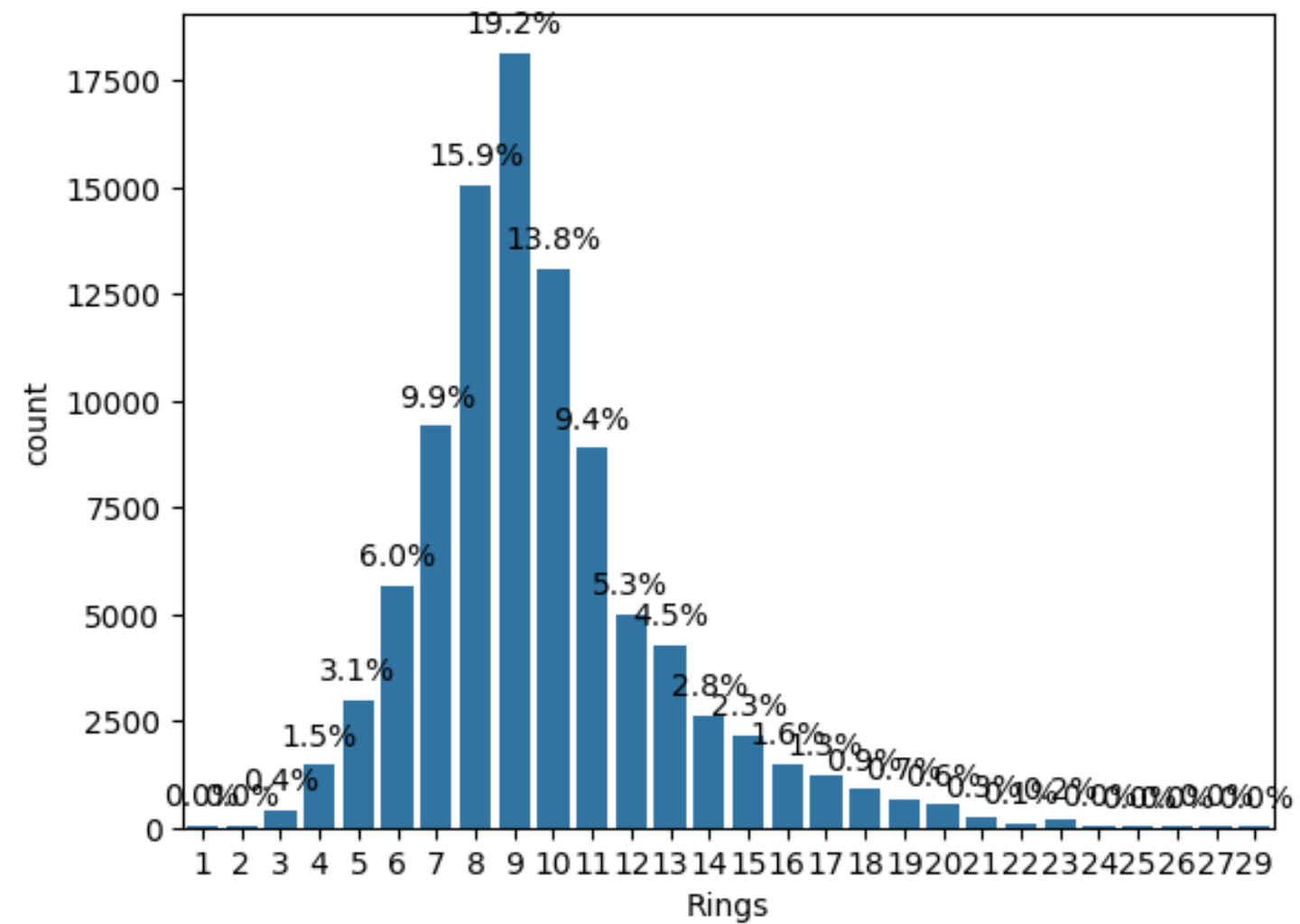
- 대회 목표: 전복의 연령 예측
 - 다양한 물리적 측정을 통해 얻은 전복 데이터
- 대회 유형: 회귀
- 대회 평가 유형: Root Mean Squared Logarithmic Error (RMSLE)
- 대회 기간: 2024.04.01 ~ 2024.05.01 (현재 진행 중)

데이터 소개

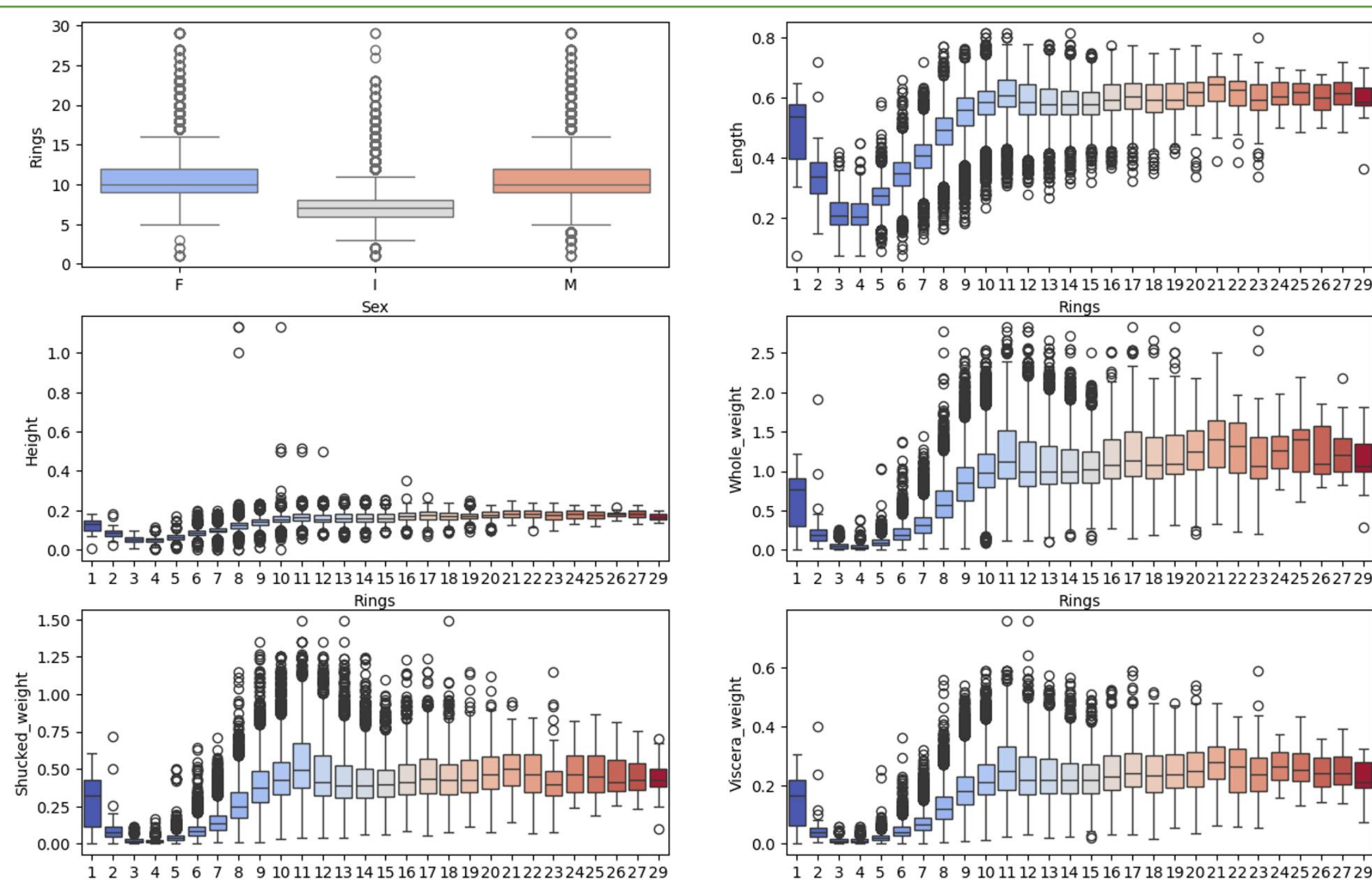
대회에서 제공하는 전복의 다양한 물리적 측정 데이터

[FEATURE DATA, TARGET DATA]	
▪ ID	전복의 샘플 아이디
▪ SEX	전복의 성별
▪ LENGTH	전복의 길이
▪ DIAMETER	전복의 지름
▪ HEIGHT	전복의 높이
▪ WHOLE WEIGHT	전복의 전체 무게
▪ WHOLE WEIGHT.1	= SHUCKED WEIGHT 껍질 벗긴 전복의 출혈 후 무게
▪ WHOLE WEIGHT.2	= VISCERA WEIGHT 전복의 내장 무게
▪ SHELL WEIGHT	물기를 제거한 전복의 껍질 무게
▪ RINGS	전복의 나이를 나타내는 고리 수

타겟 데이터, 피쳐 데이터 바 플랏으로 분포 확인

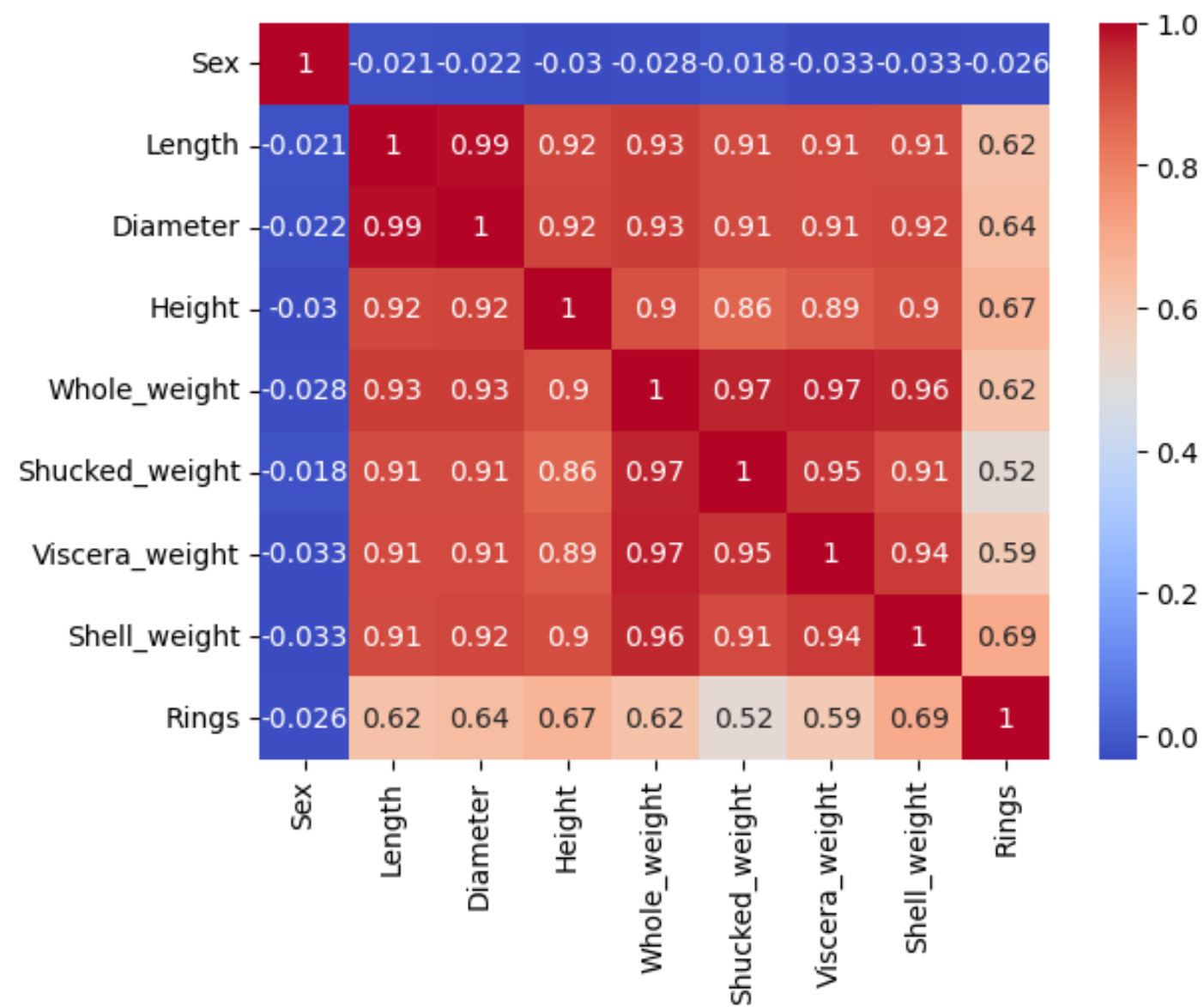


피쳐 데이터 박스 플랏으로 이상치 확인



피쳐 데이터에서 **이상치**가 다수 존재하므로
파생변수 생성으로 인한 성능 향상을 기대하기 **어려움**

피처 데이터와 타겟 데이터 히트맵으로 상관 관계 확인




성별을 제외한 피처들은 서로 매우 강한 양의 상관 관계

성별을 제외한 피처들은 타겟 데이터와 강한 양의 상관 관계

성별은 다른 피처, 타겟 데이터와 매우 약한 음의 상관 관계

전처리에 앞서, 훈련 데이터 양을 늘리기 위하여 원본 데이터를 획득 후 훈련 데이터와 합침




Abalone

Donated on 11/30/1995

Predict the age of abalone from physical measurements

Dataset Characteristics	Subject Area	Associated Tasks
Tabular	Biology	Classification, Regression
Feature Type	# Instances	# Features
Categorical, Integer, Real	4177	8

DOWNLOAD

 **IMPORT IN PYTHON**

CITE

57 citations
174391 views

Keywords

ecology

Creators

- Warwick Nash
- Tracy Sellers
- Simon Talbot
- Andrew Cawthorn
- Wes Ford

DOI

10.24432/C55C7W

Dataset Information

Additional Information

Predicting the age of abalone from physical measurements. The age of abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope -- a boring and time-consuming task. Other measurements, which are easier to obtain, are used to predict the age. Further information, such as weather patterns and location (hence food availability) may be required to solve the problem.

From the original data examples with missing values were removed (the majority having the predicted value missing), and the ranges of the continuous values have been scaled for use with an ANN (by dividing by 200).

SHOW LESS ^

Has Missing Values?

No

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

불필요한 ID 변수 제거 후, 피처 데이터 이름 변경

id	Sex	Length	Diameter	Height	Whole weight	Whole weight.1	Whole weight.2	Shell weight	Rings
0	F	0.55	0.43	0.150	0.7715	0.3285	0.1465	0.24	11
1	F	0.63	0.49	0.145	1.1300	0.4580	0.2765	0.32	11

id	Sex	Length	Diameter	Height	Whole weight	Whole weight.1	Whole weight.2	Shell weight
0	M	0.645	0.475	0.155	1.238	0.6185	0.3125	0.3005
1	M	0.580	0.460	0.160	0.983	0.4785	0.2195	0.2750

id	Rings
0	10
1	10

Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings
F	0.55	0.43	0.150	0.7715	0.3285	0.1465	0.24	11
F	0.63	0.49	0.145	1.1300	0.4580	0.2765	0.32	11
M	0.645	0.475	0.155	1.238	0.6185	0.3125	0.3005	
M	0.580	0.460	0.160	0.983	0.4785	0.2195	0.2750	

id	Rings
0	10
1	10

I, M, F로 이루어진 Sex에서 IsAdult 파생 변수 생성 Object 타입인 Sex를 Label Encoding을 통해 0, 1, 2로 변경

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	Rings	IsAdult
0	0	0.550	0.430	0.150	0.7715	0.3285	0.1465	0.2400	11	1
1	0	0.630	0.490	0.145	1.1300	0.4580	0.2765	0.3200	11	1
2	1	0.160	0.110	0.025	0.0210	0.0055	0.0030	0.0050	6	0
3	2	0.595	0.475	0.150	0.9145	0.3755	0.2055	0.2500	10	1
4	1	0.555	0.425	0.130	0.7820	0.3695	0.1600	0.1975	9	0

Sex: F = 0, I = 1, M = 2로 인코딩

IsAdult: F, M은 성체 전복, I는 유체 전복으로 판단하여
성체 전복은 1, 유체 전복은 0으로 값을 매김

```
Sex      IsAdult
I      34435      1      60357
M      32555      0      34435
F      27802
Name: count, dtype: int64
Name: count, dtype: int64
```

I, M, F로 구성된 Sex에서 IsAdult 파생변수를 생성

성체는 유체의 약 2배 가량 존재함

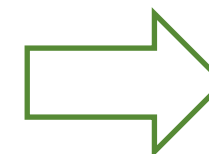
Water, Volume, Density 파생 변수 생성

	Sex	Length	Diameter	Height	Whole_weight	Shucked_weight	Viscera_weight	Shell_weight	IsAdult	Water	Volume	Density	Rings
0	0	0.550	0.430	0.150	0.7715	0.3285	0.1465	0.2400	1	0.6830	0.035475	9.260042	11
1	0	0.630	0.490	0.145	1.1300	0.4580	0.2765	0.3200	1	0.9920	0.044761	10.232007	11
2	1	0.160	0.110	0.025	0.0210	0.0055	0.0030	0.0050	0	0.0205	0.000440	12.500000	6
3	2	0.595	0.475	0.150	0.9145	0.3755	0.2055	0.2500	1	0.7890	0.042394	8.857438	10
4	1	0.555	0.425	0.130	0.7820	0.3695	0.1600	0.1975	0	0.6100	0.030664	12.050059	9

- Water(물과 피의 양) : 전복의 전체 무게 - (껍질 벗긴 무게 + 껍질 무게)
- Volume(부피) : 전복의 길이 * 지름 * 높이
- Density(실수율, 밀도) : 전복의 껍질 벗긴 무게 / Volume

```
1 zero_count = (train.Height == 0).sum()
2 print("0의 개수:", zero_count)
```

0의 개수: 8



무한대 발생

```
# infinity 값 nan으로 대체
X_train3.replace([np.inf, -np.inf], np.nan, inplace=True)
X_valid3.replace([np.inf, -np.inf], np.nan, inplace=True)

# NaN 값을 해당 열의 중앙값으로 대체
X_train3.fillna(X_train3.median(), inplace=True)
X_valid3.fillna(X_valid3.median(), inplace=True)
```

피쳐 데이터 로그 변환

```
1 # 로그 변환
2 train["Sex"] = np.log(0.00001 + train["Sex"])
3 train["Length"] = np.log(0.00001 + train["Length"])
4 train["Diameter"] = np.log(0.00001 + train["Diameter"])
5 train["Height"] = np.log(0.00001 + train["Height"])
6 train["Whole_weight"] = np.log(0.00001 + train["Whole_weight"])
7 train["Shucked_weight"] = np.log(0.00001 + train["Shucked_weight"])
8 train["Viscera_weight"] = np.log(0.00001 + train["Viscera_weight"])
9 train["Shell_weight"] = np.log(0.00001 + train["Shell_weight"])
10
11
12 test["Sex"] = np.log(0.00001 + test["Sex"])
13 test["Length"] = np.log(0.00001 + test["Length"])
14 test["Diameter"] = np.log(0.00001 + test["Diameter"])
15 test["Height"] = np.log(0.00001 + test["Height"])
16 test["Whole_weight"] = np.log(0.00001 + test["Whole_weight"])
17 test["Shucked_weight"] = np.log(0.00001 + test["Shucked_weight"])
18 test["Viscera_weight"] = np.log(0.00001 + test["Viscera_weight"])
19 test["Shell_weight"] = np.log(0.00001 + test["Shell_weight"])
```

- 히스트 플랏으로 확인한 바와 같이,
피쳐 데이터의 분포는 고르지 않음
- 대회 평가 유형 RSMLE
 - 로그 변환으로 인한 성능 향상을 기대
- 입력 값이 0 이하인 경우 로그 변환 적용 불가능
 - 0.00001을 합한 값으로 로그 변환 수행

데이터 분리 후, 정규화 진행

```
7 # 데이터 분리
8 X_train, X_valid, y_train, y_valid = train_test_split(train_final_feature, train_final_target,
9                                                       test_size=0.1,
10                                                      stratify=train_final_target,
11                                                      random_state=42)
12
13 X_train.shape, X_valid.shape, y_train.shape, y_valid.shape
```

((85312, 8), (9480, 8), (85312,), (9480,))

`train_test_split` 함수를 이용하여 기존 훈련+원본데이터를
훈련 데이터와 검증 데이터로 9 : 1 분할하여 이용

```
1 # 스케일링
2 from sklearn.preprocessing import StandardScaler
3 scaler = StandardScaler()
4 X_train_scaled = scaler.fit_transform(X_train)
5 X_train_scaled_df = pd.DataFrame(X_train_scaled, columns=X_train.columns)
6 X_val_scaled = scaler.transform(X_valid)
```

```
1 X = pd.concat([X_train_scaled_df, y_train.reset_index(drop=True)], axis=1)
2 X.shape
```

(85312, 9)

```
1 test_scaled = scaler.transform(test)
```

훈련, 검증, 테스트 데이터 모두 정규화 스케일링

회귀 레포트 분석을 통하여
변수 간 관계, 예측 및 추정, 이상치 및 다중 공선성 등 파악

OLS Regression Results						
Dep. Variable:	Rings	R-squared:	0.602			
Model:	OLS	Adj. R-squared:	0.602			
Method:	Least Squares	F-statistic:	1.590e+04			
Date:	Wed, 24 Apr 2024	Prob (F-statistic):	0.00			
Time:	07:07:38	Log-Likelihood:	-2.0051e+05			
No. Observations:	94792	AIC:	4.010e+05			
Df Residuals:	94782	BIC:	4.011e+05			
Df Model:	9					
Covariance Type:	nonrobust					
	coef	std err	t	P> t	[0.025	0.975]
Intercept	1.258e+11	1.36e+12	0.092	0.926	-2.54e+12	2.79e+12
Sex[T.I]	-1.258e+11	1.36e+12	-0.092	0.926	-2.79e+12	2.54e+12
Sex[T.M]	-0.0119	0.016	-0.721	0.471	-0.044	0.020
Length	-1.9489	0.392	-4.975	0.000	-2.717	-1.181
Diameter	8.2616	0.486	16.985	0.000	7.308	9.215
Height	20.1040	0.463	43.456	0.000	19.197	21.011
Whole_weight	3.9256	0.119	32.957	0.000	3.692	4.159
Shucked_weight	-15.8939	0.148	-107.715	0.000	-16.183	-15.605
Viscera_weight	-6.9887	0.286	-24.442	0.000	-7.549	-6.428
Shell_weight	20.2519	0.213	94.916	0.000	19.834	20.670
IsAdult	-1.258e+11	1.36e+12	-0.092	0.926	-2.79e+12	2.54e+12
Omnibus:	30763.600	Durbin-Watson:	1.971			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	192333.943			
Skew:	1.420	Prob(JB):	0.00			
Kurtosis:	9.375	Cond. No.	6.42e+14			

결정계수

0.602으로 회귀 모형은 데이터를 60.2%로 설명

왜도 Skew

1.402으로 오른쪽으로 긴 꼬리

다중 공선성 VIF

6.42e+14으로 높은 다중 공선성

AutoML 결과 RMSLE 기준 TOP 4 모델 사용

Model		MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	1.241	3.385	1.840	0.666	0.150	0.126	13.324
lightgbm	Light Gradient Boosting Machine	1.249	3.439	1.854	0.661	0.151	0.126	1.292
xgboost	Extreme Gradient Boosting	1.257	3.491	1.868	0.656	0.152	0.127	0.504
rf	Random Forest Regressor	1.284	3.559	1.886	0.649	0.155	0.131	35.632
gbr	Gradient Boosting Regressor	1.280	3.583	1.893	0.647	0.153	0.129	7.624

기본 전처리 TOP 5 모델

Model		MAE	MSE	RMSE	R2	RMSLE	MAPE	TT (Sec)
catboost	CatBoost Regressor	1.239	3.365	1.834	0.665	0.150	0.126	14.686
lightgbm	Light Gradient Boosting Machine	1.245	3.407	1.846	0.661	0.151	0.126	1.764
xgboost	Extreme Gradient Boosting	1.255	3.471	1.863	0.655	0.152	0.127	0.562
rf	Random Forest Regressor	1.280	3.525	1.877	0.649	0.154	0.131	34.102
gbr	Gradient Boosting Regressor	1.274	3.531	1.879	0.649	0.153	0.129	8.188

기본 전처리 + IsAdult 파생 변수
TOP 5 모델

catboost, lightgbm, xgboost, gbr
모델을 통하여 학습 진행

베이지안 최적화를 통하여 기본 전처리에서의
다양한 모델 학습 진행

기본 전처리

cat	0.14864
lgbm	0.14824
xgb	0.14871
gbr	0.14855
cat +lgbm +gbr	0.14708
cat + lgbm + xgb	0.14645
TOP 4	0.14808

기본 전처리 + 모든 변수 로그
변환

cat	0.14873
lgbm	0.14808
xgb	0.14871
gbr	0.14852
cat +lgbm +gbr	0.14669
cat + lgbm + xgb	0.14679
TOP 4	0.14668

대회 평가 지표가 RMSLE인 것을
고려하여
로그 변환으로 인한 성능 향상 가능성 고려

Lgbm 기준
피처 중요도

Sex : 114
Length : 472
Diameter : 417
Height : 309
Whole_weight : 548
Shucked_weight : 612
Viscera_weight : 529
Shell_weight : 752

기본 전처리 + Sex 변수 제외

cat	0.15012
lgbm	0.14988
xgb	0.14997
gbr	0.14980
cat +lgbm +gbr	0.14949
cat + lgbm + xgb	0.14930
TOP 4	0.14920

Sex의 피처 중요도가 낮게 나와서
해당 변수를 제외했을 때의
모델 성능 향상을 고려

기본 전처리에 다양한 파생변수를 추가해보며 모델 성능을 확인해보았다.

IsAdult 파생변수 추가

cat	0.14734
lgbm	0.14724
xgb	0.14758
gbr	0.14699
cat + lgbm + gbr	0.14656
cat + lgbm + xgb	0.14656
TOP 4	0.14642

F/M (성체 전복) -> 1
I (새끼 전복) -> 0

IsAdult, water 파생변수 추가

cat	0.14854
lgbm	0.14861
xgb	0.14855
gbr	0.14896
cat + lgbm + gbr	0.14731
cat + lgbm + xgb	0.14830
TOP 4	0.14761

Water = Whole_weight -
Shucked_Weight - Shell_weight

IsAdult,
Volume, Density 파생변수 추가

cat	0.14915
lgbm	0.14850
xgb	0.14919
gbr	0.14928
cat + lgbm + gbr	0.14778
cat + lgbm + xgb	0.14829
TOP 4	0.14806

Volume = Length * Diameter * Height
Density = Shucked_Weight/Volume

IsAdult, water,
Volume, Density 파생변수 추가

cat	0.14916
lgbm	0.14884
xgb	0.14932
gbr	0.14915
cat + lgbm + gbr	0.14769
cat + lgbm + xgb	0.14819
TOP 4	0.14793

파생변수를 생성한 모델의 성능이 더 낮게 나온 것을 볼 수 있었다.

베이지안 최적화를 통하여 기본 전처리에서의
다양한 모델 학습 진행

기본 전처리

cat	0.14864
lgbm	0.14824
xgb	0.14871
gbr	0.14855
cat + lgbm + gbr	0.14708
cat + lgbm + xgb	0.14645
TOP 4	0.14808

기본 전처리 + 모든 변수 로그
변환

cat	0.14873
lgbm	0.14824
xgb	0.14871
gbr	0.14855
cat + lgbm + gbr	0.14669
cat + lgbm + xgb	0.14679
TOP 4	0.14668

대회 평가 지표가 RMSLE인 것을
고려하여
로그 변환으로 인한 성능 향상 가능성 고려

Lgbm 기준
피쳐 중요도

Sex : 114
Length : 172
Volume : 417
Height : 309
Whole_weight : 518
Shelled_weight : 611
Viscera_weight : 529
Shell_weight : 752

기본 전처리 + Sex 변수 제외

cat	0.15012
lgbm	0.14988
xgb	0.14997
gbr	0.14980
cat + lgbm + gbr	0.14949
cat + lgbm + xgb	0.14930
TOP 4	0.14920

Sex의 피쳐 중요도가 낮게 나와서
해당 변수를 제외했을 때의
모델 성능 향상을 고려

성별에 관련된 Sex, IsAdult 변수는
성능에 큰 영향을 미치고 있으며,
로그 변환 데이터가 좋은 성능을 보이고 있음

캐글에서 구한 origin 추가 데이터를 활용하여
앞에서 성능이 좋았던 모델을 보안

기본 전처리		기본 전처리 + 로그 변환		기본 전처리 + IsAdult 변수 추가		기본 전처리 + IsAdult 변수 추가 + 로그 변환	
cat	0.14902	cat	0.14902	cat	0.14909	cat	0.14909
lgbm	0.14874	lgbm	0.14912	lgbm	0.14881	lgbm	0.14873
xgb	0.14920	xgb	0.14920	xgb	0.14943	xgb	0.14943
gbr	0.14858	gbr	0.14859	gbr	0.14852	gbr	1.48501
cat +lgbm +gbr	0.14907	cat +lgbm +gbr	0.14631	cat +lgbm +gbr	0.14648	cat +lgbm +gbr	0.14632
cat + lgbm + xgb	0.14931	cat + lgbm + xgb	0.14684	cat + lgbm + xgb	0.14671	cat + lgbm + xgb	0.14656
TOP 4	0.14897	TOP 4	0.14643	TOP 4	0.14643	TOP 4	0.14633

로그 변환을 하지 않은 데이터는 top4에서,
로그 변환을 한 데이터는 cat+lgbm+gbr에서 좋은 성능을 보임

마지막으로, **그리드 서치**를 통한 모델링을 진행하여 성능을 개선시킴

기본 전처리 로그 변환 (3차)

cat + lgbm + gbr	0.14632
cat + lgbm + xgb	0.14650
TOP 4	0.14637

기본 전처리 로그 변환 (6차)

cat + lgbm + gbr	0.14667
cat + lgbm + xgb	0.14643
TOP 4	0.14650

기본 전처리 + IsAdult 변수 추가
+ 로그 변환 (3차)

cat + lgbm + gbr	0.14650
cat + lgbm + xgb	0.14675
TOP 4	0.14672

기본 전처리 + IsAdult 변수 추가
+ 로그 변환 (6차)

cat + lgbm + gbr	0.14663
cat + lgbm + xgb	0.14690
TOP 4	0.14677

6차 그리드 서치까지 진행한 모델이 좋은 성능을 낼 것이라고 예상했지만,
3차까지 진행한 모델의 성능이 더 좋게 나왔기에 과적합을 의심

내용을 종합해본 결과, 가장 성능이 좋았던 Top3를 최종 모델로 선정

*4월 25일 PM 3:40 Public Score기준

1st

최종 스코어

0.14631

370등/2138 (상위 17.3%)

전처리

기본 전처리 + origin 데이터 추가
+ 변수 로그 변환

모델링

베이지안 최적화를 통한
앙상블
(cat+lgbm+gbr)

2nd

최종 스코어

0.14632

372등/2138 (상위 17.4%)

전처리

기본 전처리 + origin 데이터 추가
+ IsAdult 변수 추가 + 변수 로그 변환

모델링

베이지안 최적화를 통한
앙상블
(cat+lgbm+gbr)

2nd

최종 스코어

0.14632

373등/2138 (상위 17.4%)

전처리

기본 전처리 + origin 데이터 추가
+ 변수 로그 변환

모델링

이분법적 그리드 서치
3차 진행 앙상블
(cat+lgbm+gbr)



- 그리드 서치에서 과적합이 발생하여 성능이 오히려 저하되는 문제가 발생하였기 때문에
L1, L2 규제를 통하여 모델 성능 향상을 예상 가능
- 대부분의 상위권 사람들이 Neural Network를 사용하여 모델링을 진행하였기 때문에
추후 DNN과 같은 모델로 만들어본다면 모델 성능 향상을 예상 가능

Q&A

감사합니다