BITAmin B-Day 2022-1

신용카드 이탈 고객 예측

BITAmin 8기 복습 프로젝트

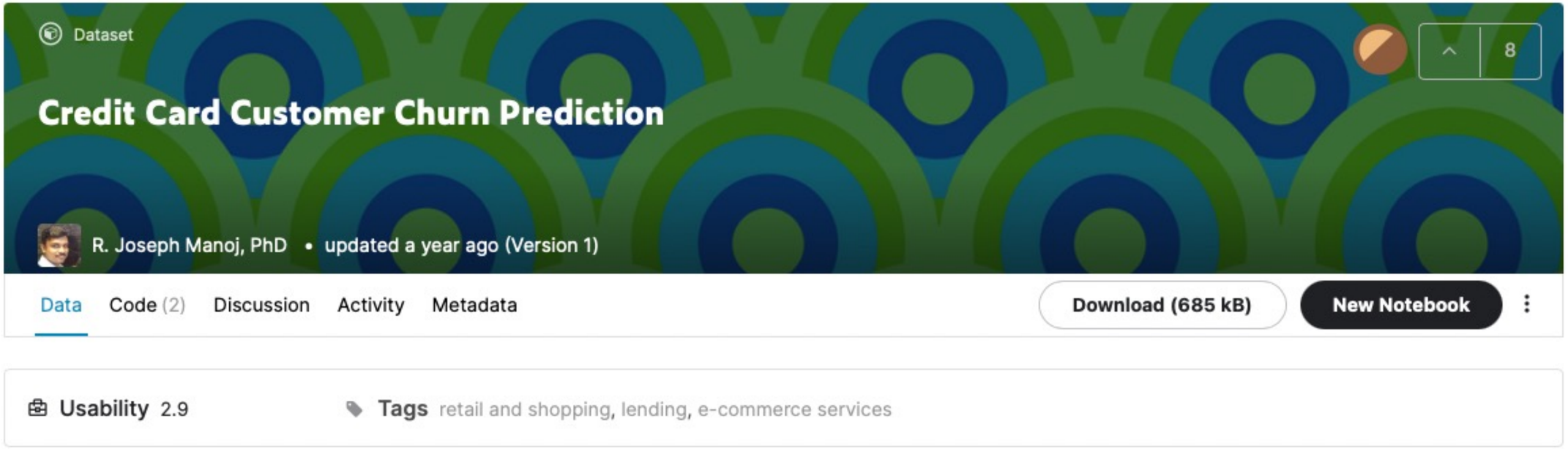
곽아람 이수현 이선호

목차

01. 프로젝트 소개

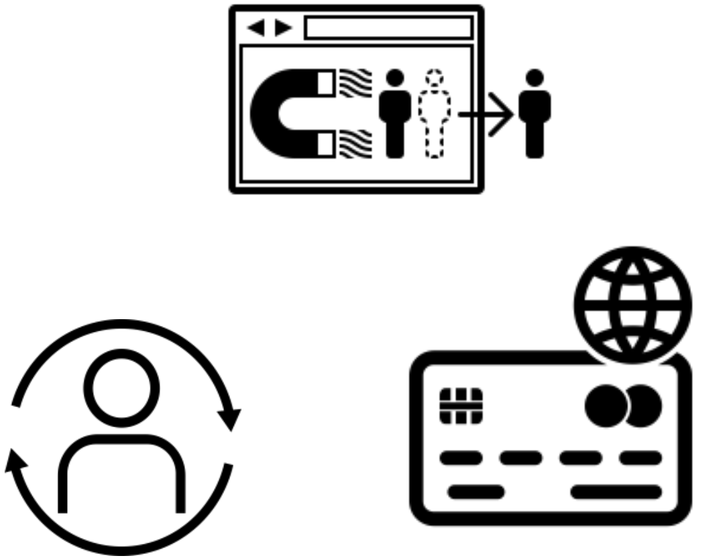
01. 프로젝트 소개

https://www.kaggle.com/rjmanoj/credit-card-customer-churn-prediction



01. 프로젝트 소개

신용카드 사용 고객의
데이터를 바탕으로
고객의 이탈 가능성 여부를 예측



01. 프로젝트 소개

Exited 컬럼: Target
Exited 외의 컬럼: Feature

RowNumber	행 번호	Tenure	상환 기간
CustomerId	고객 번호	Balance	잔고
Surname	성	NumOfProducts	상품 수
CreditScore	신용점수	HasCrCard	신용카드 소지 유무
Geography	거주 국가	IsActiveMember	활성 고객 여부
Gender	성별	EstimatedSalary	추정 연봉
Age	나이	Exited	이탈 유무

02. 전처리

02. 전처리

삭제한 Column
RowNumber: Index와 같은 역할
CustomerId, Surname: 고객 이탈 예측에 필요한 정보 X

	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
0	619	France	Female	42	2	0.00	1	1	1	101348.88	1
1	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
2	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
3	699	France	Female	39	1	0.00	2	0	0	93826.63	0
4	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...
9995	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9996	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9997	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9998	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
9999	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

02. 전처리

```
CreditScore          0
Geography            0
Gender               0
Age                  0
Tenure               0
Balance              0
NumOfProducts        0
HasCrCard            0
IsActiveMember       0
EstimatedSalary      0
Exited               0
dtype: int64
```

결측값 확인

모든 컬럼에 관해 결측값은 없는 것으로 나타남

02. 전처리

범주형 변수 Label Mapping

	Geography	Gender
0	France	Female
1	Spain	Female
2	France	Female
3	France	Female
4	Spain	Female
...	...	...
9995	France	Male
9996	France	Male
9997	France	Female
9998	Germany	Male
9999	France	Female

[Geography]

France → 0
Germany → 1
Spain → 2

[Gender]

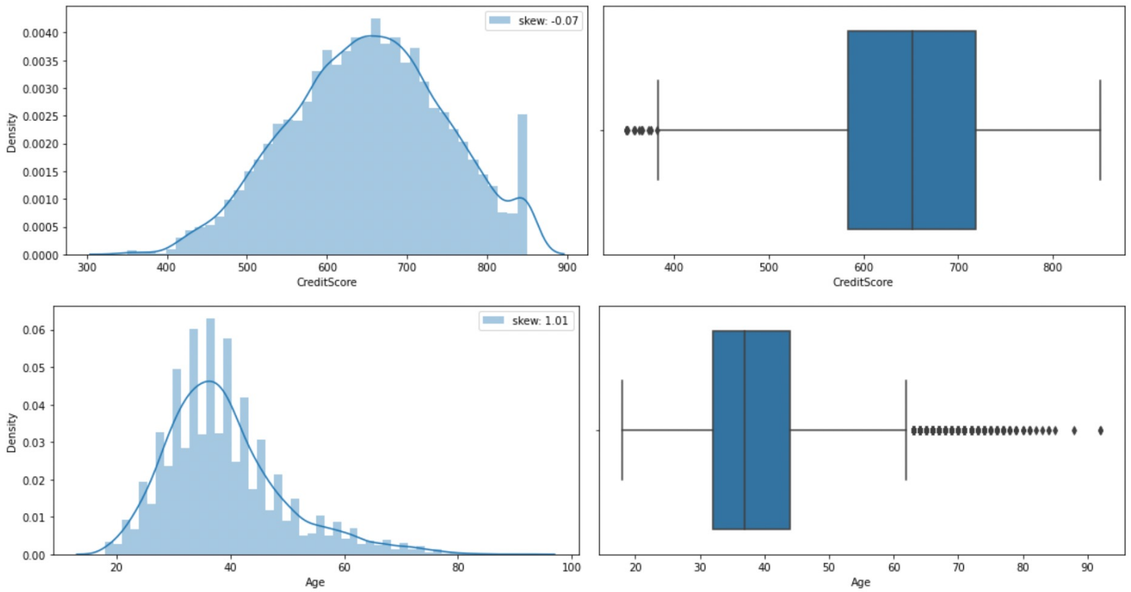
Female → 0
Male → 1

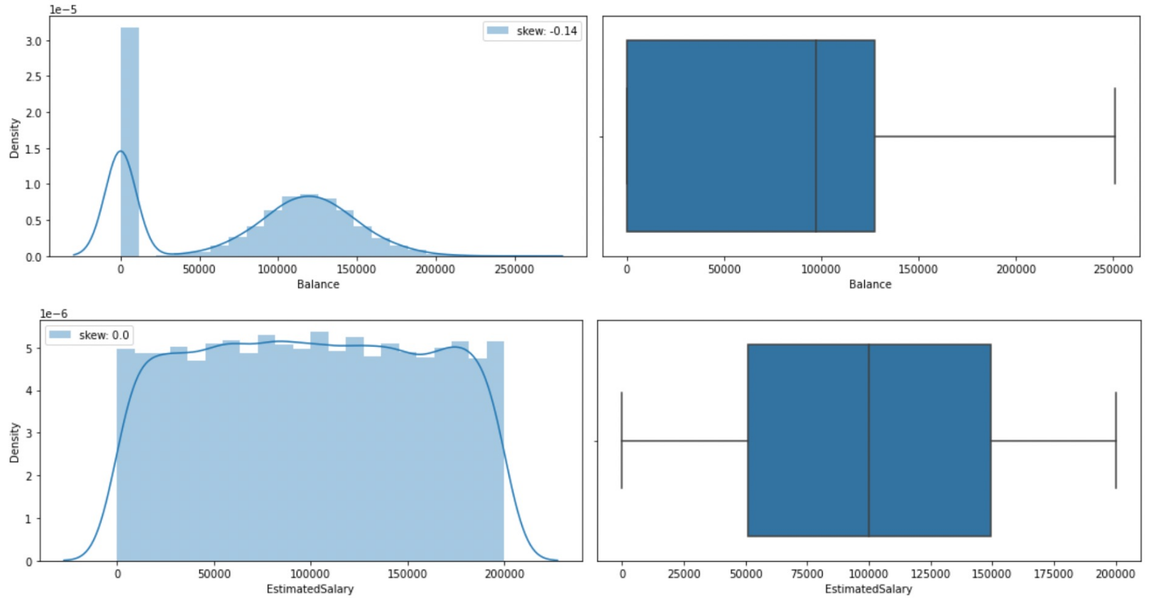
	Geography	Gender
0	0	0
1	2	0
2	0	0
3	0	0
4	2	0
...	...	...
9995	0	1
9996	0	1
9997	0	0
9998	1	1
9999	0	0

02. 전처리

이상치 확인

CreditScore, Age, Balance, EstimatedSalary





02. 전처리

StandardScaler로 정규화 진행
CreditScore, Age, Balance, EstimatedSalary

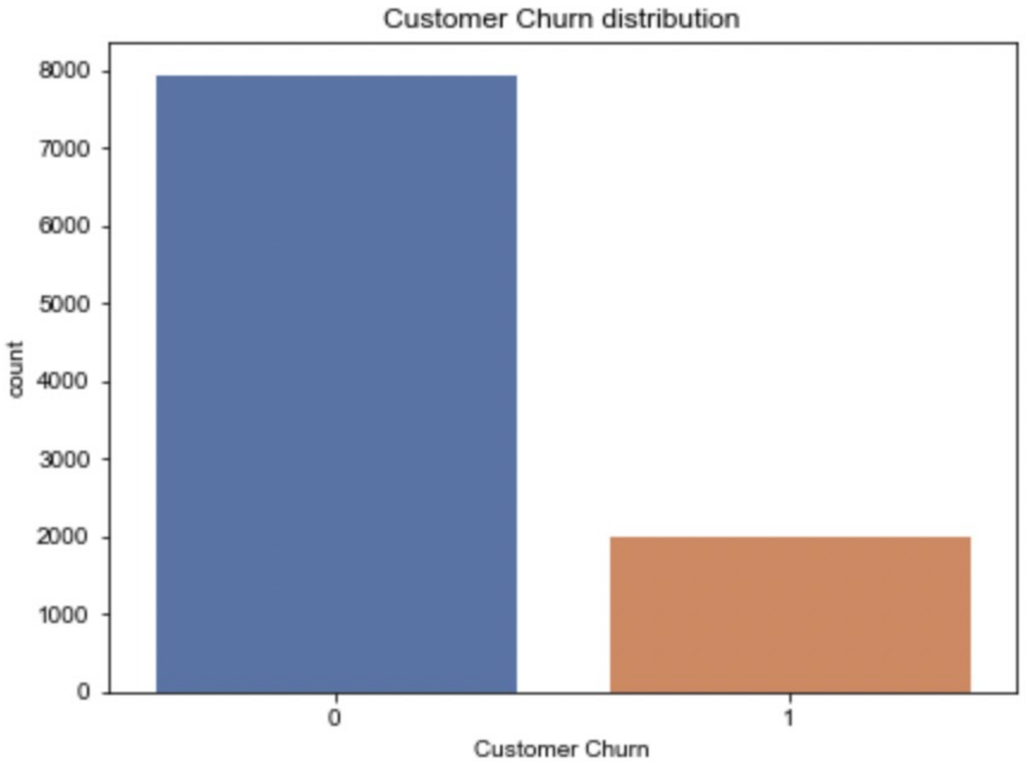
	CreditScore	Age	Balance	EstimatedSalary
0	619	42	0.00	101348.88
1	608	41	83807.86	112542.58
2	502	42	159660.80	113931.57
3	699	39	0.00	93826.63
4	850	43	125510.82	79084.10
...	...	...	...	...
9995	771	39	0.00	96270.64
9996	516	35	57369.61	101699.77
9997	709	36	0.00	42085.58
9998	772	42	75075.31	92888.52
9999	792	28	130142.79	38190.78

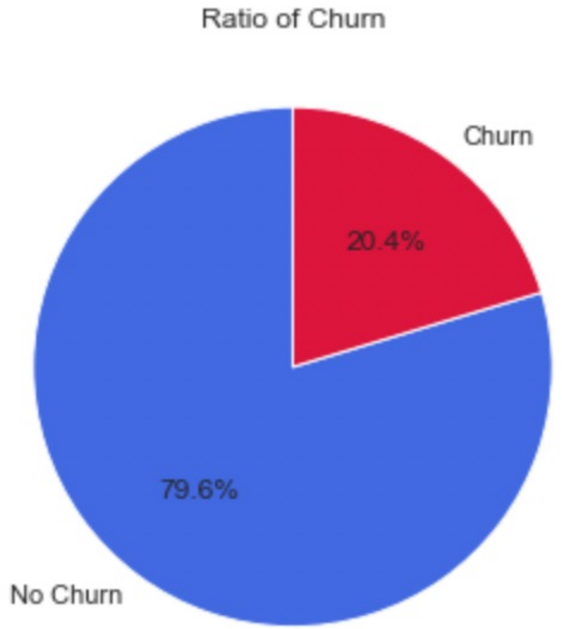
	CreditScore	Age	Balance	EstimatedSalary
0	-0.326221	0.293517	-1.225848	0.021886
1	-0.440036	0.198164	0.117350	0.216534
2	-1.536794	0.293517	1.333053	0.240687
3	0.501521	0.007457	-1.225848	-0.108918
4	2.063884	0.388871	0.785728	-0.365276
...	...	...	...	...
9995	1.246488	0.007457	-1.225848	-0.066419
9996	-1.391939	-0.373958	-0.306379	0.027988
9997	0.604988	-0.278604	-1.225848	-1.008643
9998	1.256835	0.293517	-0.022608	-0.125231
9999	1.463771	-1.041433	0.859965	-1.076370

02. 전처리

Target Column 불균형 문제 해소

이탈하지 않은 고객이 이탈 고객보다 약 4배 많은 불균형 데이터



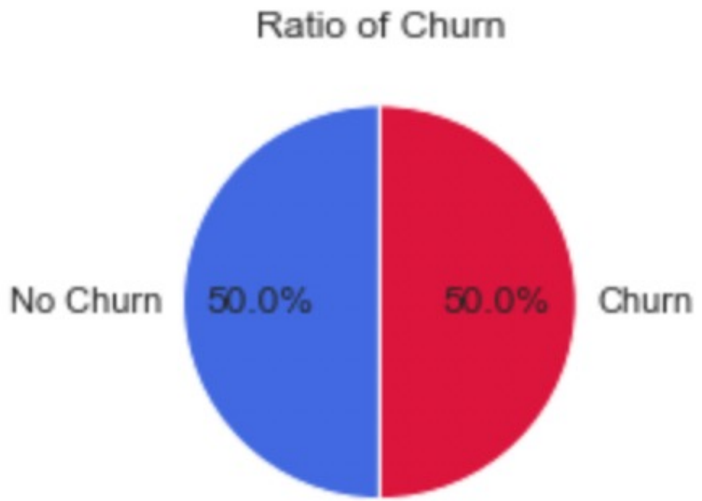


균형적으로 만들어주는 Sampling 과정이 필요!

02. 전처리

Target Column 불균형 문제 해소
이탈한 고객 데이터를 Oversampling

```
label "0(이탈X)"의 개수 : 7963
label "1(이탈O)"의 개수 : 7963
```



SMOTE를 사용해서 Oversampling 하여 이탈한 고객과 이탈하지 않은 고객의 수를 맞춰준다!

03. 모델링

03. 모델링

지난 학기 배웠던 대다수의 분류 모델 적용 시도

LR, LDA, QDA, KNN, SVM, NB, DT, RF, GBM, Adaboost, XGB, LGBM

```
models = []
models.append(('LR', LogisticRegression(max_iter =5000)))  # 로지스틱 분류기
models.append(('LDA', LinearDiscriminantAnalysis()))  # LDA 모델
models.append(('QDA', QuadraticDiscriminantAnalysis()))  # QDA 모델
models.append(('KNN', KNeighborsClassifier()))  # KNN 모델
models.append(('SVM', SVC(gamma='auto')))  # SVM 모델
models.append(('NB', GaussianNB()))  # 가우시안 나이브 베이즈 모델
models.append(('DT', DecisionTreeClassifier()))  # 의사결정나무 모델
models.append(('RF', RandomForestClassifier()))  # 랜덤포레스트 모델
models.append(('GBM', GradientBoostingClassifier()))  # GBM 모델
models.append(('Adaboost', AdaBoostClassifier()))  # AdaBoost 모델
models.append(('XGB', XGBClassifier(objective='binary:logistic', eval_metric='logloss')))  # XGB 모델
models.append(('Light_GBM', LGBMClassifier(boost_from_average=False)))  # Light_GBM 모델
```

사용한 Metric: F1-Score, AUC

03. 모델링

기본적으로 세 가지 유형으로 학습 진행
+ Hyperparameter Tuning 일부 모델 적용

SMOTE

단순히 앞의 전처리에서
SMOTE한 결과로
모델 학습

SMOTE + PCA

SMOTE한 결과에
PCA(주성분 분석)
적용하여 학습

SMOTE + FS

SMOTE한 결과에
일부 Feature만
골라서 학습

03. 모델링

SMOTE

모델 평가 결과

f1-score auc	LR	LDA	QDA	KNN	SVM	NB	DT	RF	GBM	Ada	XGB	LGBM
SMOTE	0.734 0.734	0.734 0.734	0.763 0.766	0.854 0.843	0.808 0.810	0.758 0.762	0.817 0.815	0.879 0.879	0.834 0.836	0.805 0.808	0.895 0.898	0.884 0.887

KNN, RF, GBM, XGB, LGBM의 결과에 주목

03. 모델링

SMOTE + PCA

	explained_variance_ratio
1	0.593561
2	0.669509
3	0.740926
4	0.811980
5	0.881420
6	0.929415
7	0.950237
8	0.968233
9	0.985288
10	1.000000

PCA 결과

```
from sklearn.decomposition import PCA

pca = PCA()
pca.fit(X)
cumsum = np.cumsum(pca.explained_variance_ratio_)
dim = np.argmax(cumsum >= 0.95) + 1
print('PCA 주성분 수 : %s차원' % (dim))
```

95% 이상의 설명력을 지니는 PCA 주성분 수: 7

03. 모델링

SMOTE + PCA

모델 평가 결과

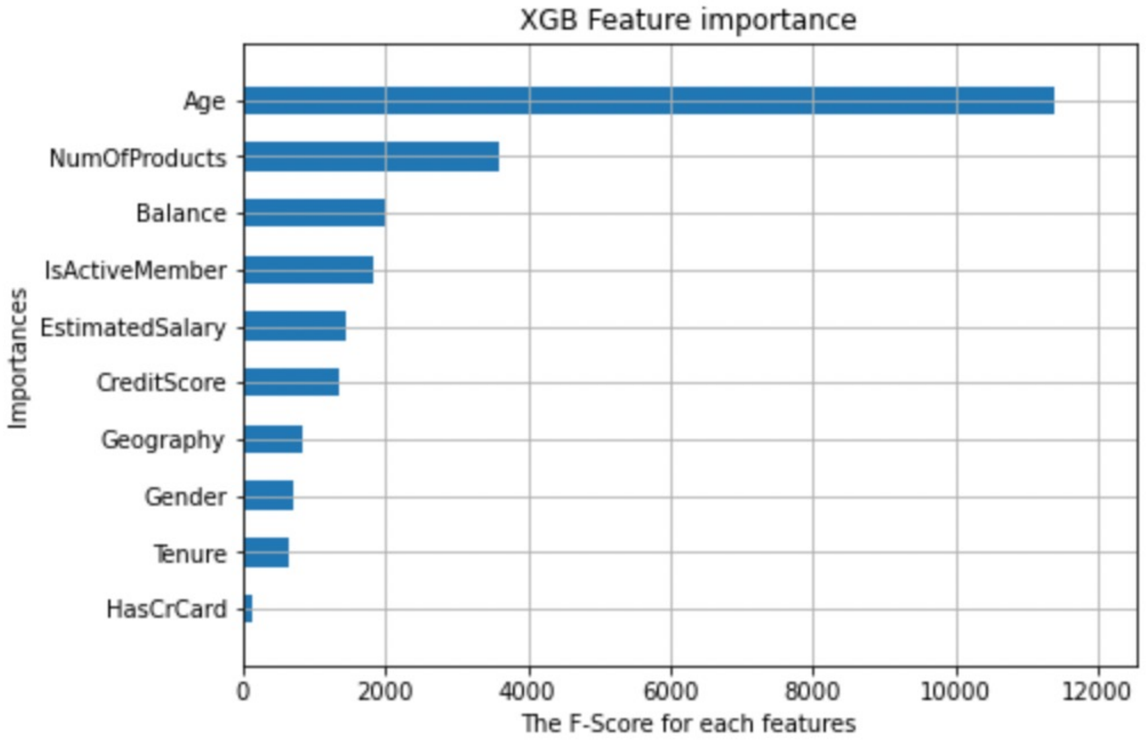
f1-score auc	LR	LDA	QDA	KNN	SVM	NB	DT	RF	GBM	Ada	XGB	LGBM
SMOTE + PCA	0.683 0.693	0.682 0.692	0.706 0.713	0.839 0.827	0.784 0.784	0.706 0.715	0.793 0.790	0.867 0.867	0.816 0.816	0.772 0.775	0.875 0.877	0.857 0.859

SMOTE만 적용했을 때보다 전반적으로 좋지 않은 결과

03. 모델링

SMOTE + FS

F-Score를 통해 중요도가 높은 Feature 고르기



Age, NumOfProducts, Balance, IsActiveMember

03. 모델링

SMOTE + FS

모델 평가 결과

f1-score auc	LR	LDA	QDA	KNN	SVM	NB	DT	RF	GBM	Ada	XGB	LGBM
SMOTE + FS	0.689 0.694	0.685 0.691	0.688 0.706	0.807 0.805	0.753 0.760	0.704 0.716	0.819 0.823	0.841 0.844	0.789 0.793	0.753 0.759	0.859 0.860	0.846 0.848

SMOTE만 적용했을 때보다 전반적으로 좋지 않은 결과

03. 모델링

SMOTE + Hyperparameter Tuning

Random Forest

```
parameters_rf = {
     'n_estimators': [100,200,500],
     'max_depth' : [None,5,10],
    'min_samples_leaf' : [1,2,3],
    'min_samples_split' : [2,5,7]
}
```

GBM

```
parameters_gbm = {
    'random_state':[10],
     'n_estimators': [50,100,200,500],
     'learning_rate': [0.01,0.1,1]
}
```

KNN

```
parameters_knn = { 'n_neighbors' : [6],
                   'weights' : ['uniform','distance'],
                   'metric' : ['minkowski','euclidean','manhattan'],
                   'p':[1,2],
                   'leaf_size':[20,30,40]}
```

XGBoost

```
parameters_xgb = {
     'eval_metric':['logloss'],
    'booster':['gbtree'],
    'eta':[0.01,0.1,0.3],
    'nrounds':[1000,2000],
    'max_depth':[4,6,8],
    'objective':['binary:logistic'],
    'gamma':[0,2,4]
}
```

LGBM

```
parameters_lgbm = {
    'random_state':[10],
     'objective':['binary'],
    'metric':['binary_logloss'],
    'learning_rate':[0.01, 0.1],
    'max_depth':[-1,1,3,5],
    'num_iterations':[100,1000,2000]
}
```

03. 모델링

SMOTE + Hyperparameter Tuning

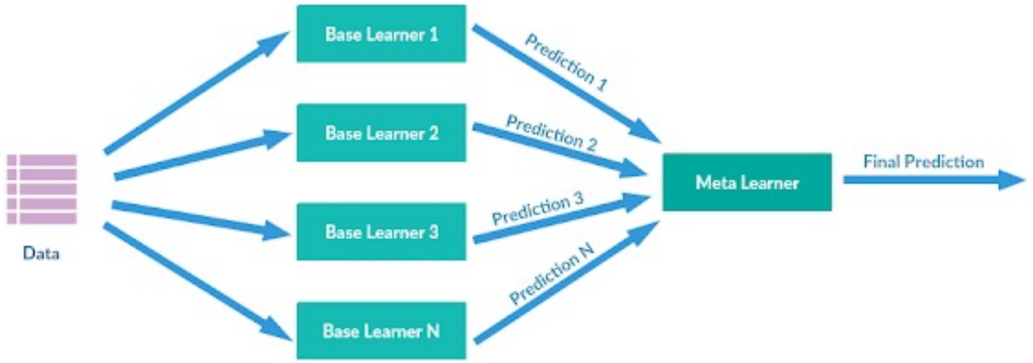
모델 평가 결과

f1-score auc	LR	LDA	QDA	KNN	SVM	NB	DT	RF	GBM	Ada	XGB	LGBM
SMOTE	0.734 0.734	0.734 0.734	0.763 0.766	0.854 **(0.871)** 0.843 **(0.861)**	0.808 0.810	0.758 0.762	0.817 0.815	0.879 **(0.881)** 0.879 **(0.881)**	0.834 **(0.890)** 0.836 **(0.892)**	0.805 0.808	0.895 0.898	0.884 **(0.899)** 0.887 **(0.901)**

SMOTE만 적용했을 때보다 지표의 향상을 볼 수 있다!

03. 모델링

SMOTE + Stacking



Stacking이란?

앙상블 기법 중의 하나

개별적인 모델들이 학습하고 예측한 데이터를 쌓아서
다른 학습 데이터셋으로 사용해서 학습하는 방법

03. 모델링

SMOTE + Stacking

```
pred_knn = clf_knn.predict(X_test)
pred_rf = clf_rf.predict(X_test)
pred_gbm = clf_gbm.predict(X_test)
pred_lgbm = clf_lgbm.predict(X_test)
```

```
print('KNN AUC: {0:.3f}'.format(roc_auc_score(y_test, pred_knn)))
print('Random Forest AUC: {0:.3f}'.format(roc_auc_score(y_test, pred_rf)))
print('GBM AUC: {0:.3f}'.format(roc_auc_score(y_test, pred_gbm)))
print('LGBM AUC: {0:.3f}'.format(roc_auc_score(y_test, pred_lgbm)))
```

```
KNN AUC: 0.861
Random Forest AUC: 0.885
GBM AUC: 0.892
LGBM AUC: 0.901
```

```
pred = np.array([pred_knn, pred_rf, pred_gbm, pred_lgbm])

pred = np.transpose(pred)
```

```
clf_xgb.fit(pred, y_test)
final = clf_xgb.predict(pred)

print('최종 메타 모델의 AUC: {0:.4f}'.format(roc_auc_score(y_test , final)))
```

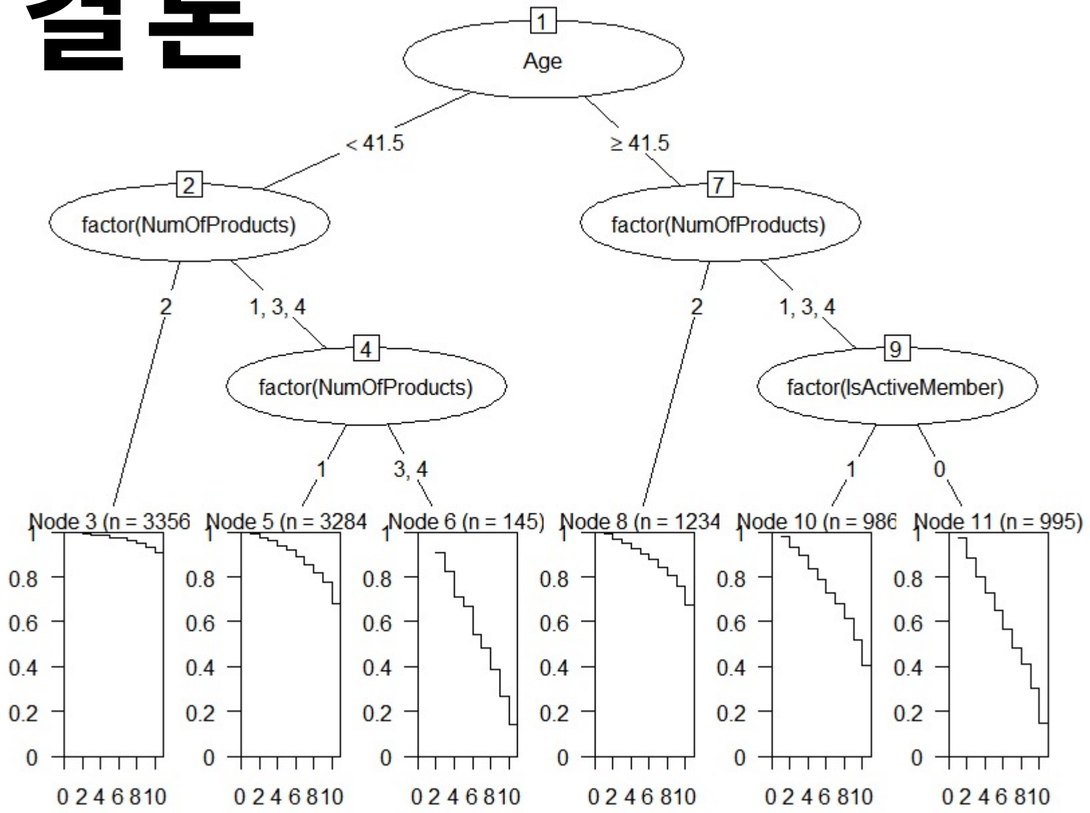
```
최종 메타 모델의 AUC: 0.9108
```

KNN + RF + GBM + LGBM ⇒ XGB(final)

최종 AUC: 0.910

04. 해석 및 결론

04. 해석 및 결론

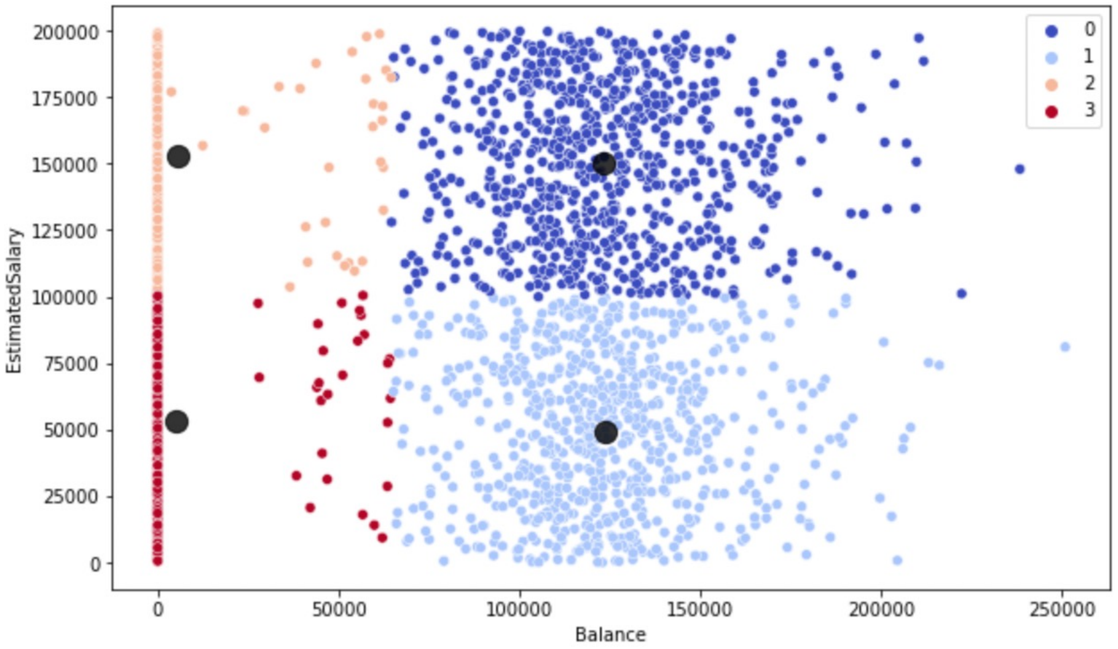


어떤 사람들이 이탈 가능성이 높을까?

Age(나이)가 41.5세보다 작고
NumOfProducts(보유 상품 수)가 3 또는 4일 때

Age(나이)가 41.5세 이상이고
NumOfProducts(보유 상품 수)가 2가 아니면서
IsActiveMember(활성 사용자 유무)가 0일 때

04. 해석 및 결론



이탈 고객의 특징은 어떨까?

이탈 고객 군집 1: 계좌 잔액 ⇩, 추정 연봉 ⇧

이탈 고객 군집 2: 계좌 잔액 ⇧, 추정 연봉 ⇩

이탈 고객 군집 3: 계좌 잔액 ⇧, 추정 연봉 ⇧

이탈 고객 군집 2: 계좌 잔액 ⇩, 추정 연봉 ⇩

THANK YOU

곽아람 이수현 이선호