가입 고객 이탈 예측

1조: 최영민, 박찬규, 배윤관, 서장호

프로젝트 주제: 가입 고객 이탈 예측

사용 데이터: telecom churn (cell2cell)

데이터 출처: https://www.kaggle.com/datasets/jpacse/datasets-for-churn-telecom?

데이터 크기: Total records = 51047, Total columns = 58

- 데이터 소개:

캐글에서 제공하는 Telecom Churn (Cell2Cell) 데이터셋은 Duke University의 Teradata Center for Customer Relationship Management에서 제공한 고객 이탈 예측 데이터로

이 데이터셋은 텔레콤 산업의 고객 이탈(churn) 문제를 다루고 있으며, 고객 이탈을 예측하는 모델을 학습하고 평가하는 데 유용하게 사용될 수 있다.

Duke University의 Teradata Center for Customer Relationship Management에서 제공한 고객 이탈 예측 데이터라는 말은, 이 데이터셋이 특정 실제 텔레콤 회사의 데이터가 아니라 연구와 학습을 위한 시뮬레이션 데이터라는 의미로

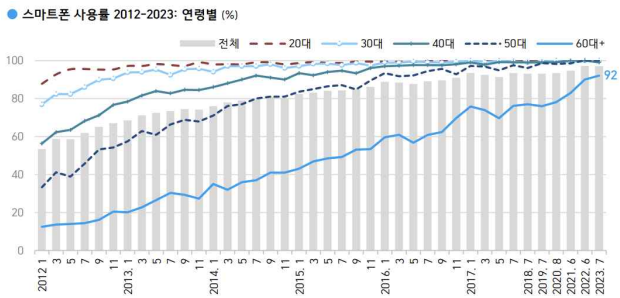
즉, 실제 텔레콤 회사의 데이터가 아니라, Duke University의 Teradata Center에서 가상의 고객 데이터를 생성하여 만든 데이터셋이라 뜻인데 이 데이터셋은 고객 이탈 예측 문제를 해결하기 위한 연구 목적이나 교육용으로 만들어졌으며, 특정 텔레콤 회사의 데이터를 반영하는 것이 아니라, 여러 텔레콤 산업에서 일어날 수 있는 이탈 패턴을 시뮬레이션한 가상 데이터라고 볼 수 있다

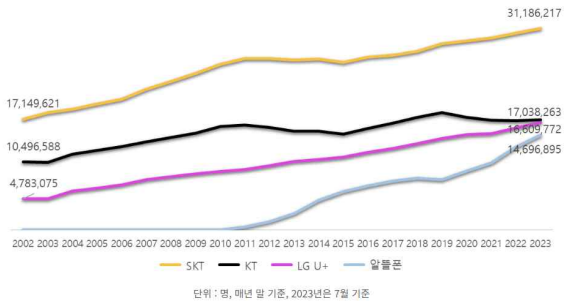
실제 국내 통신사 고객 데이터를 사용하였으면 좋겠지만, 이것이 사실상 불가능하기 때문에 캐글에서 공개된 cell2cell 데이터셋을 이용하였고, 해당 데이터를 가지고 분석한 결과 생각보다 유의미한 패턴과 의미를 발견할 수 있었다.

Duke University의 Teradata Center: 고객 관계 관리를 위한 데이터 분석과 CRM 관련 연구를 전문적으로 수행하는 학술 기관으로 기업들이 고객 데이터를 활용해 이탈 예측이나 마케팅 최적화 등의 문제를 해결할 수 있도록 돕는 연구를 진행하고 있다

- 데이터 선정 배경:

현재 국내 스마트폰 사용률은 60대 이상을 제외하면, 모든 연령대에서 98% 이상으로 상당히 높음 -> 국내 이동통신사업은 시장포화 상태





출처: https://www.gallup.co.kr/

출처: https://electronit.tistory.com/

이로 인해 국내 통신 3사의 고객 유지 및 유치를 위한 마케팅 비용은 연간 8조원 이상을 넘길 정도로 높음

이처럼 고객 하나하나가 소중한 통신사업에서 고객 이탈을 예측하고 이탈 고객의 행동을 해석하여 이를 방지하는 전략은 회사의 수익성을 크게 개선할 수 있음

특히 텔레콤 산업은 경쟁이 매우 치열하고 고객 이탈률이 높은 분야이기 때문에,

고객 이탈 예측 프로그램을 구축하는 것은 비즈니스에 매우 중요한 역할한다고 생각하여 이와 같은 데이터를 선정하게 되었다.

- 주요 목표:

- 통신 고객 데이터(cell2cell)를 분석하여 서비스 사용량, 인구 통계 및 개인 정보를 포함한 정보를 바탕으로 이탈률을 예측

- 예측 모델에서 이탈률에 가장 큰 영향을 미치는 변수를 파악하여 추후에 이탈을 방지하도록 함

- data columns:

열 이름	설명	열 이름	설명
CustomerID	고객 ID	**Churn**	이탈 여부
MonthlyRevenue	월 지출 금액	**MonthlyMinutes**	한 달 통화 시간
TotalRecurringCharge	총 정기 청구 금액	**DirectorAssistedCalls**	상담원 도움 통화 횟수
OverageMinutes	초과 통화 시간	**RoamingCalls**	로밍 통화 횟수
PercChangeMinutes	통화 시간 변화율	**PercChangeRevenues**	수익 변화율
DroppedCalls	끊어진 통화 횟수	**BlockedCalls**	차단된 통화 횟수
UnansweredCalls	응답 없는 통화 횟수	**CustomerCareCalls**	고객 지원 센터 통화 횟수
ThreewayCalls	3자 통화 횟수	**ReceivedCalls**	받은 통화 횟수
OutboundCalls	발신 통화 횟수	**InboundCalls**	수신 통화 횟수
PeakCallsInOut	피크 시간 통화 횟수	**OffPeakCallsInOut**	비피크 시간 통화 횟수
DroppedBlockedCalls	끊어진/차단된 통화 횟수	**CallForwardingCalls**	착신 전환 통화 횟수

CallWaitingCalls	통화 대기 횟수	**MonthsInService**	서비스 사용 기간 (개월)
UniqueSubs	고유 가입자 수	**ActiveSubs**	활성 가입자 수
ServiceArea	서비스 지역	**Handsets**	보유 휴대전화 수
HandsetModels	보유 휴대전화 모델 수	**CurrentEquipmentDays**	현재 휴대전화 사용 기간 (일수)
AgeHH1	첫 번째 가구 구성원 나이	**AgeHH2**	두 번째 가구 구성원 나이
ChildrenInHH	가구 내 자녀 수	**HandsetRefurbished**	리퍼브 휴대전화 여부
HandsetWebCapable	휴대전화 웹 사용 가능 여부	**TruckOwner**	트럭 소유 여부
RVOwner	RV 소유 여부	**Homeownership**	자택 소유 여부
BuysViaMailOrder	우편 주문 여부	**RespondsToMailOffers**	우편 프로모션 응답 여부
OptOutMailings	우편 수신 거부 여부	**NonUSTravel**	미국 외 국가 여행 여부

OwnsComputer	컴퓨터 소유 여부	**HasCreditCard**	신용카드 소유 여부
RetentionCalls	유지 관련 전화 횟수	**RetentionOffersAccepted**	유지 제안 수락 횟수
NewCellphoneUser	새 휴대전화 사용자 여부	**NotNewCellphoneUser**	오래된 휴대전화 사용자 여부
ReferralsMadeBySubscriber	추천 인원 수	**IncomeGroup**	소득 그룹
OwnsMotorcycle	오토바이 소유 여부	**AdjustmentsToCreditRating**	신용 등급 조정 횟수
HandsetPrice	휴대전화 가격	**MadeCallToRetentionTeam**	유지 팀에 전화 여부
CreditRating	신용 등급	**PrizmCode**	생활 방식 코드
Occupation	직업	**MaritalStatus**	결혼 여부

EDA

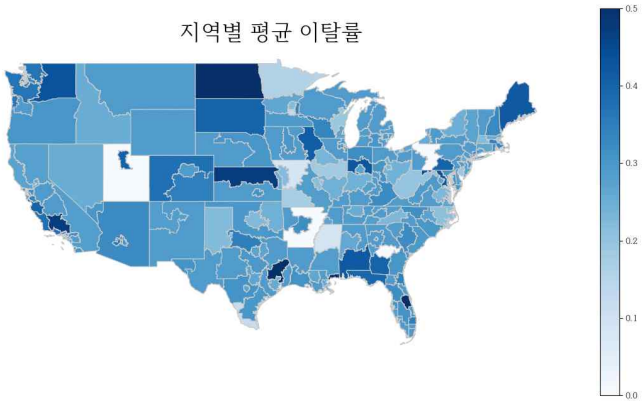
1. 결측 치 분포 :

Total records = 51047
Total columns = 58

	Total Missing	In Percent
AgeHH1	909	1.78
AgeHH2	909	1.78
PercChangeRevenues	367	0.72
PercChangeMinutes	367	0.72
DirectorAssistedCalls	156	0.31
TotalRecurringCharge	156	0.31
RoamingCalls	156	0.31
OverageMinutes	156	0.31
MonthlyRevenue	156	0.31
MonthlyMinutes	156	0.31
ServiceArea	24	0.05

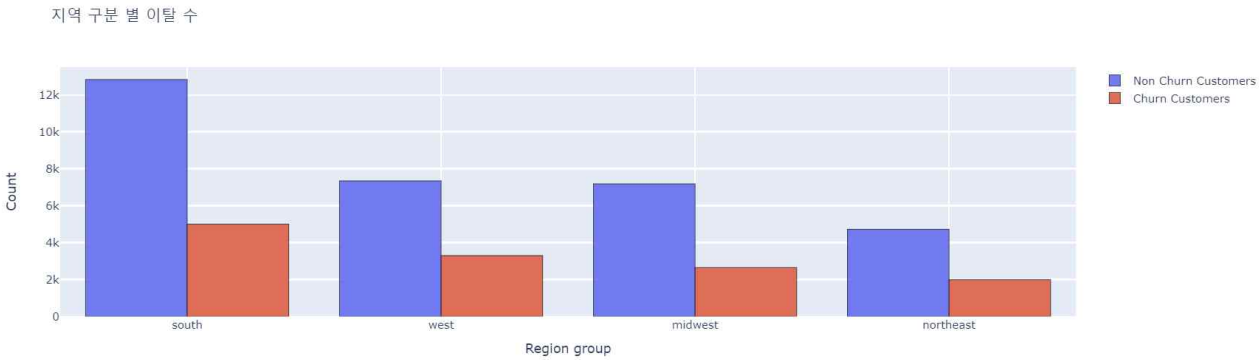
- 전체 51047 개의 데이터 샘플 중 58개의 열에서 각 열 당 2%이하의 결측 치를 보유하고 있다.

지역 별 이탈여부 시각화:

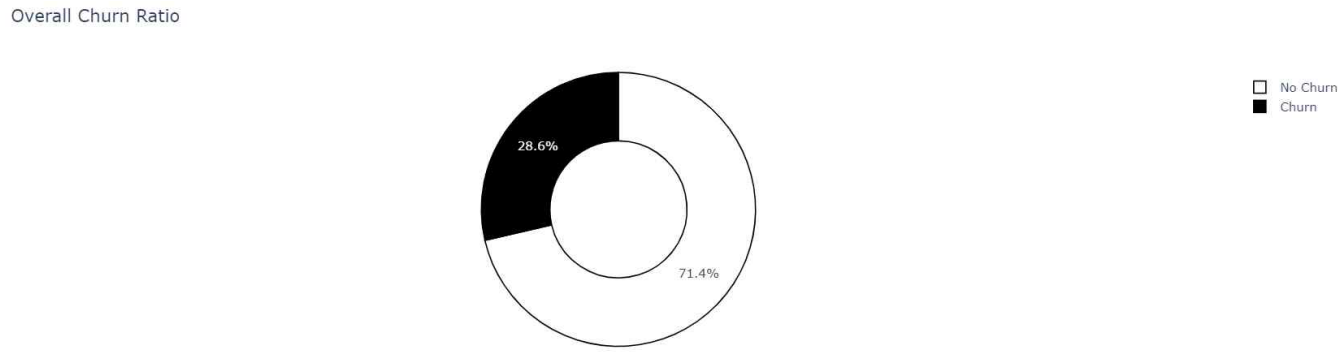


각 지역당 이탈 여부 분포가 상이한 것을 살펴볼 수 있다.

가설 : 지역 별 이탈 여부 분포가 상이하므로 지역 변수를 추가하였을 때, 예측 성능이 올라갈 것이다.



이탈률 분포:



전체 데이터 샘플 중 71.4%가 이탈하지 않은 데이터 샘플

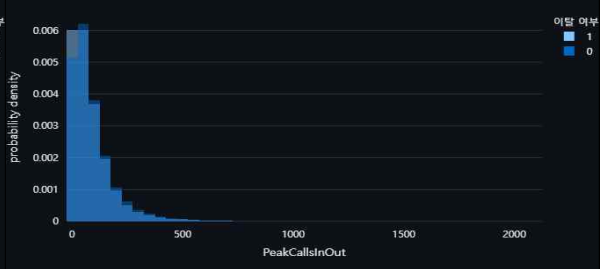
➔ 클래스 불균형이 매우 심한 데이터 셋임을 알 수 있다.
➔ 따라서, 학습 중 이를 해결하는 것이 중요한 사항이 될 것이다.

각 변수별 시각화:

- ➔ 각 변수 에서 이탈 여부에 따른 분포가 차이를 보이고 있지 않다.
- ➔ 각 변수가 이탈 여부를 예측하는 데 큰 설명성을 가지고 있다고 판단하기 힘들다.
- ➔ 따라서, feature importance를 기반으로 한 feature 선택을 통해 전처리를 우선 발전한다.

InboundCalls에 따른 이탈 여부 분포

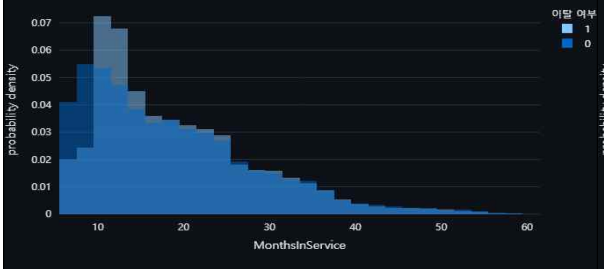
PeakCallsInOut에 따른 이탈 여부 분포



CallForwardingCalls에 따른 이탈 여부 분포

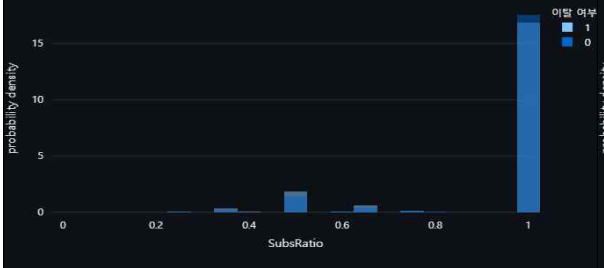
CallWaitingCalls에 따른 이탈 여부 분포

MonthsInService에 따른 이탈 여부 분포



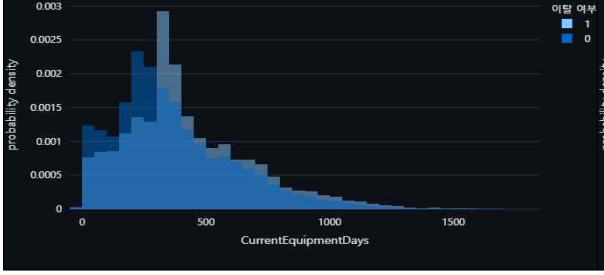
UniqueSubs에 따른 이탈 여부 분포

SubsRatio에 따른 이탈 여부 분포



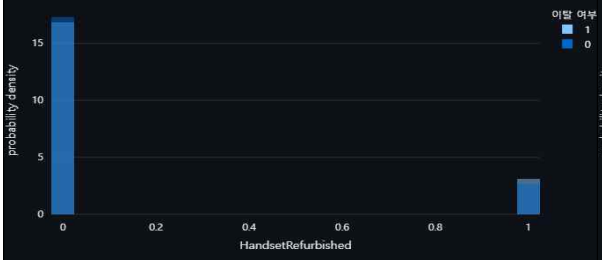
Handsets에 따른 이탈 여부 분포

CurrentEquipmentDays에 따른 이탈 여부 분포



ChildrenInHH에 따른 이탈 여부 분포

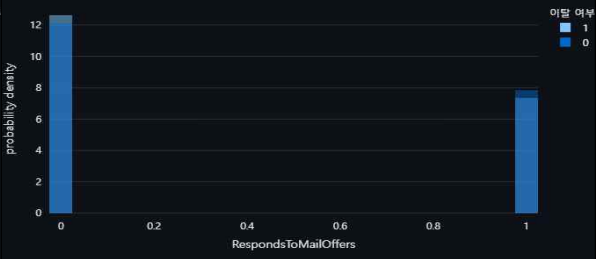
HandsetRefurbished에 따른 이탈 여부 분포



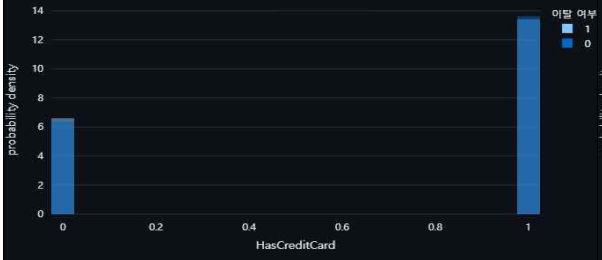
HandsetWebCapable에 따른 이탈 여부 분포

BuysViaMailOrder에 따른 이탈 여부 분포

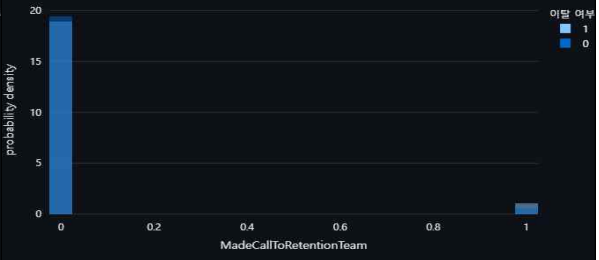
RespondsToMailOffers에 따른 이탈 여부 분포



HasCreditCard에 따른 이탈 여부 분포



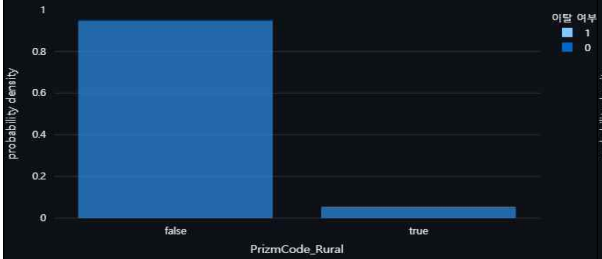
MadeCallToRetentionTeam에 따른 이탈 여부 분포



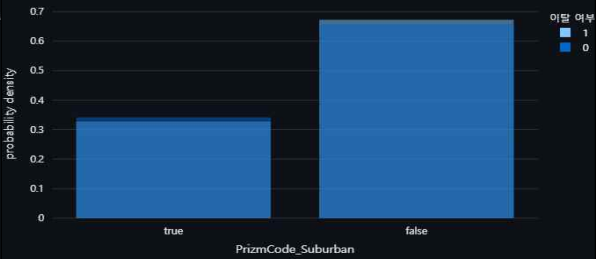
CreditRating에 따른 이탈 여부 분포

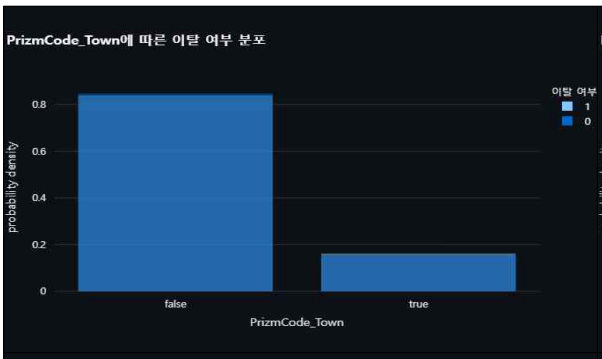
PrizmCode_Other에 따른 이탈 여부 분포

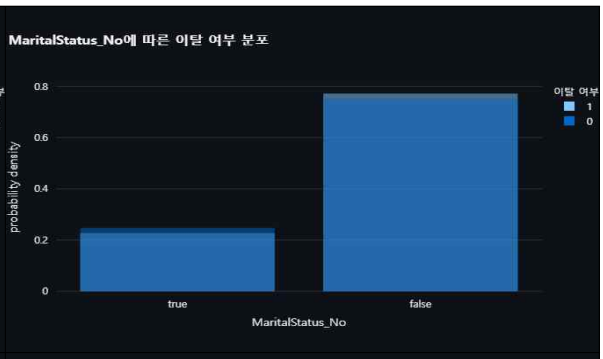
PrizmCode_Rural에 따른 이탈 여부 분포

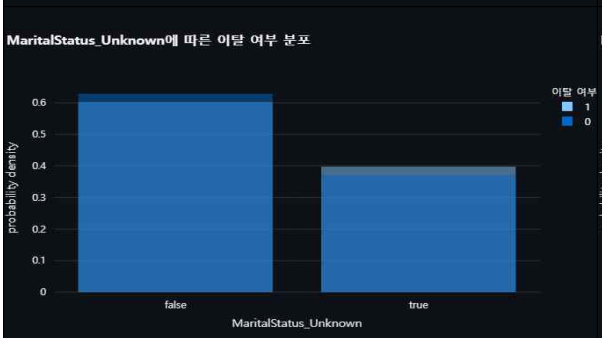


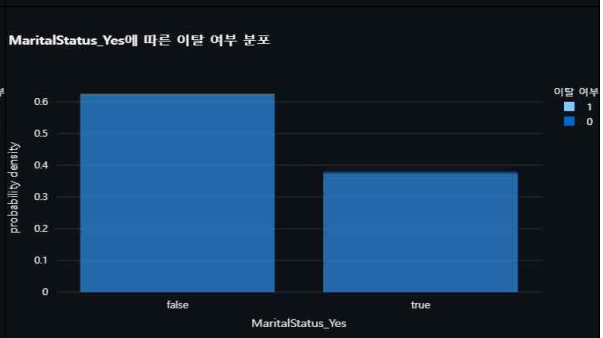
PrizmCode_Suburban에 따른 이탈 여부 분포

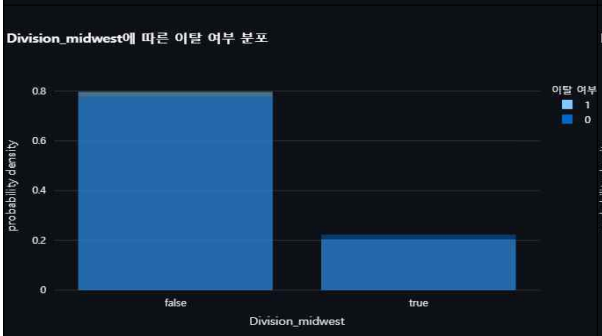


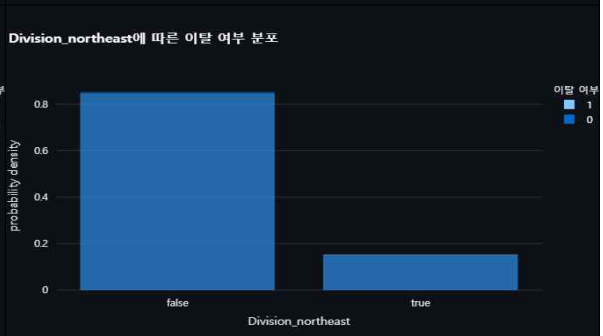

PrizmCode_Town에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.2
0.4
0.6
0.8
false
true
PrizmCode_Town

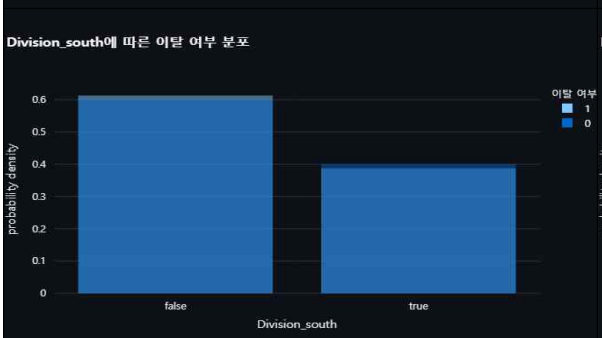

MaritalStatus_No에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.2
0.4
0.6
0.8
true
false
MaritalStatus_No

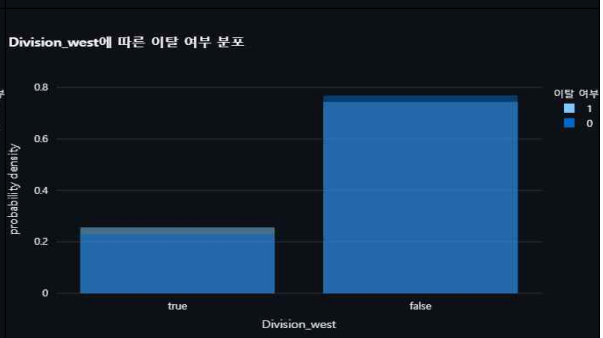

MaritalStatus_Unknown에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.1
0.2
0.3
0.4
0.5
0.6
false
true
MaritalStatus_Unknown

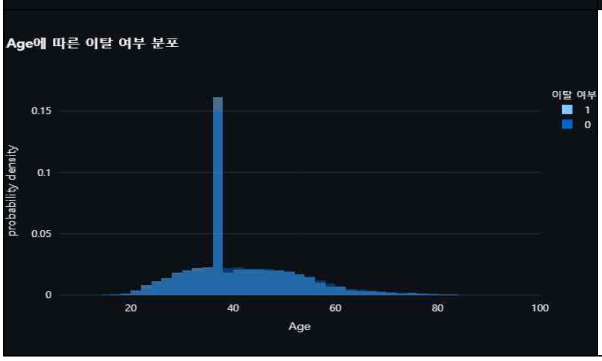

MaritalStatus_Yes에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.1
0.2
0.3
0.4
0.5
0.6
false
true
MaritalStatus_Yes


Division_midwest에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.2
0.4
0.6
0.8
false
true
Division_midwest


Division_northeast에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.2
0.4
0.6
0.8
false
true
Division_northeast


Division_south에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.1
0.2
0.3
0.4
0.5
0.6
false
true
Division_south

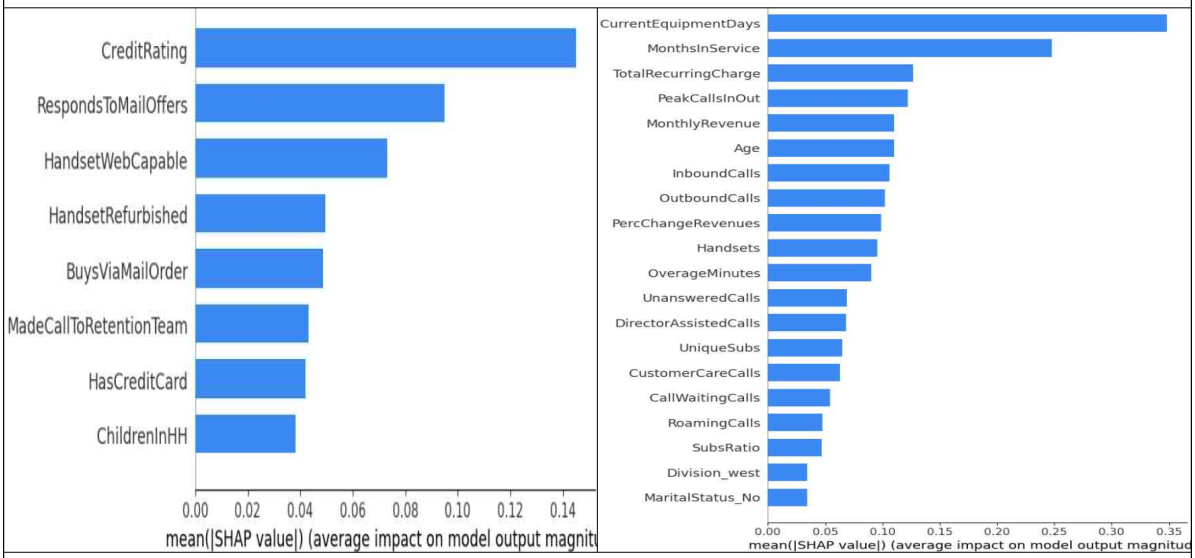

Division_west에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.2
0.4
0.6
0.8
true
false
Division_west


Age에 따른 이탈 여부 분포
이탈 여부
1
0
probability density
0
0.05
0.1
0.15
20
40
60
80
100
Age

피쳐 중요도:

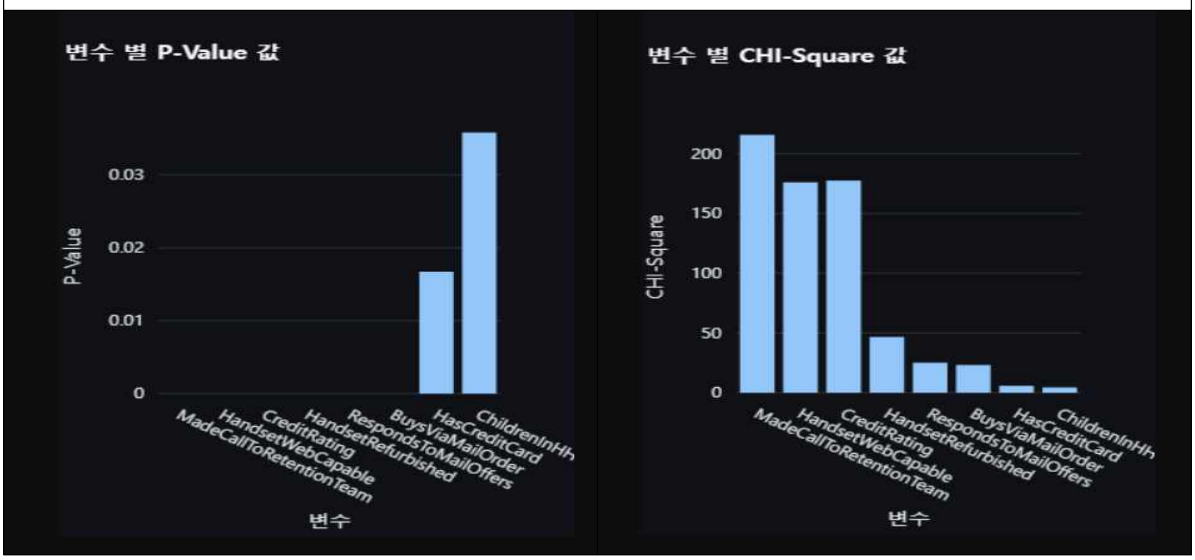
SHAP Value

모델의 예측에 각 변수들이 미치는 영향을 설명하기 위해 **SHAP Value**분석

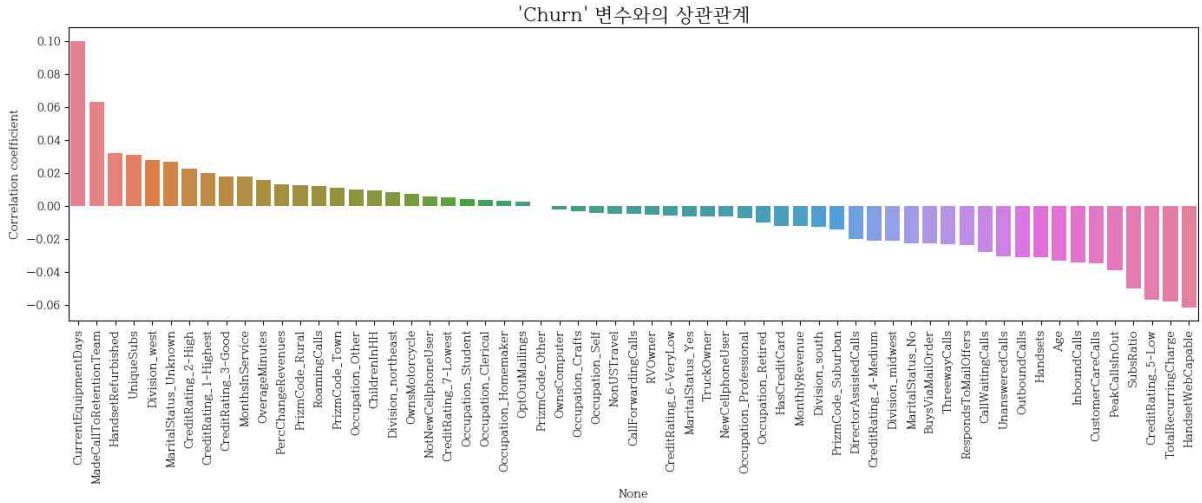


P-Value(차이가 우연히 발생 할 가능성) CHI-Square(차이)

변수와 타겟 변수 사이의 통계적 유의성을 평가하기 위해 **P-Value**를 계산



Target 변수와의 상관관계:



타켓 변수인 churn과 각 변수와의 상관관계 크기에 따라 정렬한 막대그래프

- **상관관계가 높은 변수**를 중점적으로 분석하고, 모델에 반영

데이터 전처리 과정

베이스 모델: random forest

1. Baseline : 가장 간단한 전처리
   - 결측치 **단순 제거**방식으로 처리
   - 범주형 변수에 대해 **Label Encoding**을 적용
   - 데이터에서 불필요한 변수로 판단된 **ServiceArea**를 제거

	precision	recall	f1-score	support
0	0.73	0.98	0.83	8877
1	0.60	0.07	0.13	3561
accuracy			0.72	12438
macro avg	0.66	0.53	0.48	12438
weighted avg	0.69	0.72	0.63	12438

2. Improve 1
   : **KNN Imputer**를 사용하여 결측치를 보완했으며, k 값을 다양하게 설정해 성능을 비교
   a) 결측치 KNN Imputer (k=10)

	precision	recall	f1-score	support
0	0.73	0.97	0.83	9084
1	0.57	0.09	0.15	3678
accuracy			0.72	12762
macro avg	0.65	0.53	0.49	12762
weighted avg	0.68	0.72	0.64	12762

b) 결측치 KNN Imputer (k=7)

	precision	recall	f1-score	support
0	0.72	0.97	0.83	9084
1	0.56	0.08	0.14	3678
accuracy			0.72	12762
macro avg	0.64	0.53	0.49	12762
weighted avg	0.68	0.72	0.63	12762

c) 결측치 KNN Imputer (k=5)

	precision	recall	f1-score	support
0	0.72	0.97	0.83	9084
1	0.57	0.09	0.15	3678
accuracy			0.72	12762
macro avg	0.65	0.53	0.49	12762
weighted avg	0.68	0.72	0.63	12762

➔ Knnimputer (k=10) 선정하기로 결정

3. Improve 2

데이터 불균형 처리를 추가

a) Train data undersampling (sklearn utils resampling)

	precision	recall	f1-score	support
0	0.79	0.60	0.68	9084
1	0.38	0.60	0.47	3678
accuracy			0.60	12762
macro avg	0.58	0.60	0.57	12762
weighted avg	0.67	0.60	0.62	12762

b) Train data oversampling (SMOTE)

	precision	recall	f1-score	support
0	0.73	0.90	0.81	9084
1	0.41	0.17	0.24	3678
accuracy			0.69	12762
macro avg	0.57	0.54	0.52	12762
weighted avg	0.64	0.69	0.64	12762

➔ 오버샘플링, 언더샘플링 모두 한번씩 적용해보는 걸로 결정

4. Improve 3
범주형 변수 인코딩 (Ordinal encoding)
a) Train data undersampling (sklearn utils resampling)

	precision	recall	f1-score	support
0.0	0.79	0.59	0.68	9084
1.0	0.38	0.62	0.47	3678
accuracy			0.60	12762
macro avg	0.59	0.61	0.58	12762
weighted avg	0.68	0.60	0.62	12762

범주형 변수 인코딩 Ordinal encoding
b) Train data oversampling (SMOTE)

	precision	recall	f1-score	support
0.0	0.73	0.96	0.83	9084
1.0	0.52	0.10	0.16	3678
accuracy			0.71	12762
macro avg	0.62	0.53	0.50	12762
weighted avg	0.67	0.71	0.64	12762

5. Improve 4
피처엔지니어링
a) Age feature 추가(AgeHH1, AgeHH2 제거)
- AgeHH1, 2가 모두 결측치 일 때 -> Age = AgeHH1의 중앙값
- AgeHH1만 존재할 경우 -> Age = AgeHH1
- AgeHH1, AgeHH2 모두 존재할 경우 -> Age = AgeHH1, AgeHH2의 평균

	precision	recall	f1-score	support
0.0	0.80	0.60	0.68	9084
1.0	0.39	0.62	0.47	3678
accuracy			0.61	12762
macro avg	0.59	0.61	0.58	12762
weighted avg	0.68	0.61	0.62	12762

b) Subsratio feature 추가

	precision	recall	f1-score	support
0.0	0.80	0.60	0.68	9084
1.0	0.39	0.63	0.48	3678
accuracy			0.61	12762
macro avg	0.59	0.61	0.58	12762
weighted avg	0.68	0.61	0.63	12762

c) Division 추가

	precision	recall	f1-score	support
0	0.80	0.61	0.69	8216
1	0.39	0.62	0.48	3340
accuracy			0.61	11556
macro avg	0.59	0.61	0.58	11556
weighted avg	0.68	0.61	0.63	11556

d) Division 추가 (ServiceAreaNo 제거)

	precision	recall	f1-score	support
0	0.80	0.60	0.69	8216
1	0.39	0.63	0.48	3340
accuracy			0.61	11556
macro avg	0.60	0.62	0.59	11556
weighted avg	0.68	0.61	0.63	11556

6. 최종 라벨링 및 검증

결측치 KNN Imputer (k=10)

범주형 변수 Ordinal encoding

ServiceArea 제거

Train data undersampling (sklearn utils resampling)

Age feature 추가(AgeHH1, AgeHH2 제거)

Subsratio feature 추가

Division 추가 (ServiceAreaNo 제거)

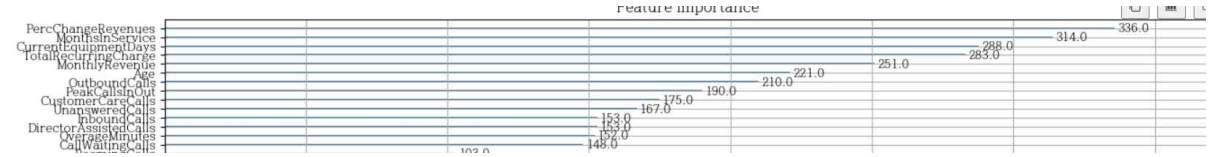
라벨링 : -> 검증

- 신용등급 : 오디널 인코딩
- 프리즘코드, 오큐페이션, 결혼여부, 디비전 : 원핫인코딩

	precision	recall	f1-score	support
0	0.79	0.61	0.69	8216
1	0.39	0.61	0.48	3340
accuracy			0.61	11556
macro avg	0.59	0.61	0.58	11556
weighted avg	0.68	0.61	0.63	11556

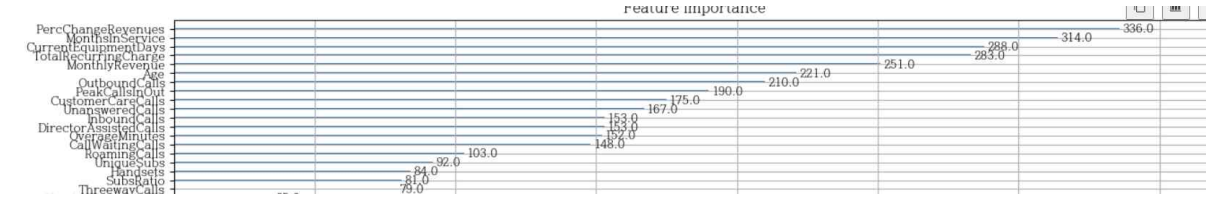
Feature importance 별 Case 분류

Case 1:



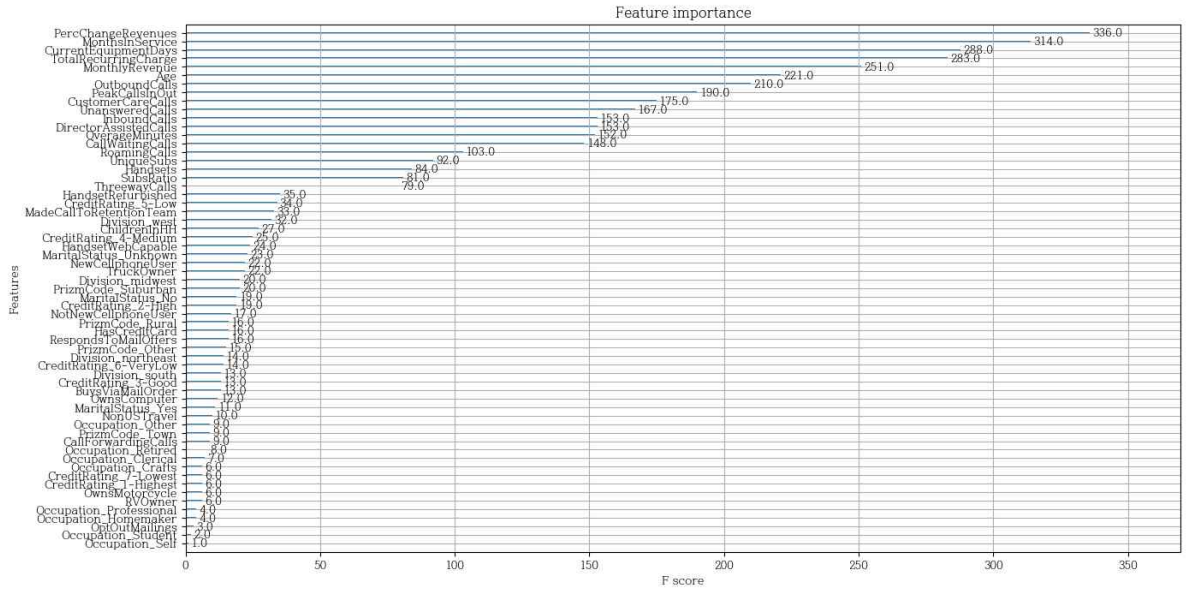
	precision	recall	f1-score	support
0	0.78	0.58	0.67	8216
1	0.37	0.60	0.46	3340
accuracy			0.59	11556
macro avg	0.58	0.59	0.56	11556
weighted avg	0.66	0.59	0.61	11556

Case 2:



	precision	recall	f1-score	support
0	0.79	0.59	0.67	8216
1	0.37	0.61	0.46	3340
accuracy			0.59	11556
macro avg	0.58	0.60	0.57	11556
weighted avg	0.67	0.59	0.61	11556

Case3:



	precision	recall	f1-score	support
0	0.80	0.60	0.69	8216
1	0.39	0.63	0.48	3340
accuracy			0.61	11556
macro avg	0.60	0.62	0.59	11556
weighted avg	0.68	0.61	0.63	11556

Case 3에서 가장 높은 성능을 보여주었으며, 특히 이탈(클래스 1)의 F1 스코어가 0.48로 가장 높습니다. 이를 통해 **Case 3**가 최종적으로 가장 우수한 성능을 낸 모델임을 확인할 수 있다.

클래스 불균형 문제를 해결하기 위해 여러 머신러닝 모델에 샘플링 기법을 적용한 후, 다양한 성능 지표를 기반으로 모델을 평가한 결과를 정리한 것입니다. 주요 성능 지표로는 정확도(Accuracy), AUC(Area Under the Curve), **F1 점수(F1 Score)**가 사용되었으며,특히 F1 점수를 개선하는 것을 목표로 했습니다.

	Model	Accuracy	AUC	F1 Score
0	Logistic Regression	0.580735	0.611305	0.437463
1	KNN	0.547331	0.555763	0.401292
2	Decision Tree	0.539119	0.539347	0.399422
3	Random Forest	0.595383	0.642349	0.463350
4	Gradient Boosting	0.585285	0.663924	0.485191
5	XGBoost	0.594496	0.649184	0.470128
6	LightGBM	0.591832	0.660910	0.477260
7	CatBoost	0.600266	0.666094	0.484840
8	MLP	0.603374	0.602010	0.416204

샘플링 전

다양한 머신러닝 알고리즘을 테스트했습니다: 낮은 F1 Score가 문제점으로 인식되었고, 오버샘플링, 언더샘플링 기법을 적용하여 클래스 불균형을 해소하고자 했습니다

f1 score를 높이기 위해 샘플링 진행

	Model	Accuracy	AUC	F1 Score
0	Logistic Regression	0.714793	0.610265	0.075540
1	KNN	0.544224	0.547411	0.388566
2	Decision Tree	0.603374	0.530528	0.341320
3	Random Forest	0.707358	0.624584	0.174648
4	Gradient Boosting	0.716569	0.644163	0.070597
5	XGBoost	0.713572	0.647152	0.262361
6	LightGBM	0.718899	0.662008	0.131642
7	CatBoost	0.720897	0.664209	0.203862
8	MLP	0.629897	0.608329	0.405420

오버 샘플링 결과

F1 점수를 개선하기 위해 오버샘플링 기법을 적용하여 데이터셋을 균형 있게 만든 후, 모델을 재평가했습니다. 일부 모델에서 F1 점수가 향상되었으나, AUC와 정확도는 다소 감소했습니다.

	Model	Accuracy	AUC	F1 Score
4	Gradient Boosting	0.585396	0.664097	0.485399
7	CatBoost	0.600266	0.666094	0.484840
6	LightGBM	0.591832	0.660910	0.477260
5	XGBoost	0.594496	0.649184	0.470128
3	Random Forest	0.599157	0.645917	0.468980
0	Logistic Regression	0.580735	0.611305	0.437463
1	KNN	0.547331	0.555763	0.401292
2	Decision Tree	0.540118	0.540516	0.400636
8	MLP	0.586838	0.560826	0.368019

언더 샘플링 결과

언더샘플링 기법을 적용하여 다수 클래스를 줄인 후 모델을 재평가한 결과입니다. F1 점수에서 일부 모델, 특히 그래디언트 부스팅과 CatBoost가 가장 좋은 성능을 보였고, 전체적인 F1점수를 확인한결과 오버샘플링보다 언더샘플링에서 더 높은 상승을 보였습니다.

앞의 내용을 반여해 딥러닝 모델(LSTM, GRU, Transformer) 또한 언더샘플링 데이터를 기반으로 평가되었습니다.

Model	Accuracy	ROC-AUC	F1(1) Score
LSTM + Attention	0.6894	0.6141	0.27
BiLSTM + CNN	0.6736	0.6112	0.33
GRU + Attention	0.6874	0.6210	0.31
Transformer	0.7026	0.6270	0.20

딥러닝 모델 결과

딥러닝 모델은 정확도 측면에서 향상된 성능을 보였으나, 전반적으로 **전통적인 머신러닝 모델(예: Gradient Boosting, LightGBM)**이 F1 점수에서 더 우수한 성능을 보였습니다.

모델의 신뢰성을 확인하고 성능을 더 정교하게 평가하기 위해 Stratified K-Fold 교차검증을 적용했습니다. 일반적인 K-Fold 교차검증은 데이터셋을 무작위로 나누는 반면, Stratified K-Fold는 각 폴드에서 클래스 비율을 유지하여, 특히 불균형 데이터에서 보다 정확한 모델 평가를 가능하게 합니다.

	Model	Accuracy	AUC	F1 Score
4	Gradient Boosting	0.612762	0.658138	0.631332
6	LightGBM	0.616855	0.661019	0.630557
7	CatBoost	0.617279	0.661947	0.628173
3	Random Forest	0.603614	0.644015	0.609160
5	XGBoost	0.598826	0.638249	0.605362
0	Logistic Regression	0.574313	0.606660	0.569574
8	MLP	0.558524	0.584817	0.557673
2	Decision Tree	0.545438	0.545438	0.545780
1	KNN	0.534319	0.548897	0.533351

10-fold 교차검증

이 Stratified K-Fold 교차검증 결과, 그래디언트 부스팅, LightGBM, CatBoost 모델이 일관된 성능을 보였으며, F1 점수에서 우수한 성능을 기록했습니다.교차검증케이스가 더 높은 성능을 보여 딥러닝모델에도 동일하게 적용했습니다.

딥러닝 모델 활용

	Model	Accuracy	AUC	F1 Score
10	DNN	0.735498	0.806032	0.742367
9	ANN	0.648439	0.709249	0.655623
12	RNN	0.646006	0.705447	0.653132
14	MLP_NN	0.648245	0.707851	0.646739
11	CNN	0.635235	0.686569	0.644062
4	Gradient Boosting	0.612762	0.658094	0.631305
6	LightGBM	0.616855	0.661019	0.630557
7	CatBoost	0.617279	0.661947	0.628173
3	Random Forest	0.602223	0.642188	0.607857
5	XGBoost	0.598826	0.638249	0.605362
8	MLP_Sklearn	0.565511	0.589375	0.571121
0	Logistic Regression	0.574313	0.606660	0.569574
13	SLP	0.575780	0.606335	0.568562
2	Decision Tree	0.541268	0.541268	0.541090
1	KNN	0.534319	0.548897	0.533351

여러 딥러닝 모델을 사용하여 성능평가를 진행. 특히 DNN(Deep Neural Networks) 모델이 F1 점수 0.7424로 가장 우수한 성능을 기록했으며, RNN, CNN 등의 모델도 강력한 성능을 보였다.
딥러닝 모델이 복잡한 패턴을 학습하는 데 강점을 가지고 있음을 시사함

최종 결론

1. Stratified K-Fold 교차검증을 통해 클래스 불균형 문제를 해결하면서 신뢰성 있는 성능 평가를 할 수 있었으며, 특히 Gradient Boosting과 LightGBM이 매우 우수한 성능을 보였습니다

2. 데이터에 따라 다르지만, 이번 케이스에서는 오버샘플링보다 언더샘플링이 더 효과적이었고, 전통적인 머신러닝 모델이 클래스 불균형을 다루는 데 있어 여전히 강력한 대안임을 확인했습니다.

3. 향후 연구에서는 딥러닝 모델과 전통적 모델을 결합한 앙상블 모델을 통해 성능을 더욱 향상시킬 수 있을 것으로 기대됩니다.

기대효과 및 한계점

기대효과

1. 고객 이탈 요인 분석
- 고객의 서비스 이탈에 영향을 미치는 요인을 체계적으로 분석하여 이탈 가능성이 높은 고객을 사전에 파악할 수 있습니다.

2. 효율적인 고객 관리
- 이탈 가능성이 높은 고객에게 집중적으로 맞춤형 고객 관리 프로그램을 실시함으로써, 마케팅 비용을 효과적으로 줄일 수 있습니다.

3. 매출 증대 및 고객 충성도 강화
- 장기적으로 충성도 높은 고객을 확보하여 매출을 증대시키고, 브랜드의 지속 가능한 성장을 도모할 수 있습니다.

한계점

1. 변수 조합 선정의 어려움
- 최적의 변수 조합을 선정하는 것이 어려워, 파라미터 튜닝과 모델 학습에서 제약을 받을 수 있습니다.

2. 예측 정확도 문제
- 예측 정확도 및 정확도 상승률이 미비할 경우, 모델의 신뢰성에 영향을 미칠 수 있습니다.

3. 데이터 비식별화의 한계
- 고객 데이터의 비식별화 처리로 인해 추가적인 외부 변수를 수집하는 데 한계가 있으며, 이는 모델의 성능에 영향을 미칠 수 있습니다.