이동통신사 고객이탈 분석 및 예측

TEAM 4. Be 전공자
남예은(팀장), 김영성, 김태리

CONTENTS

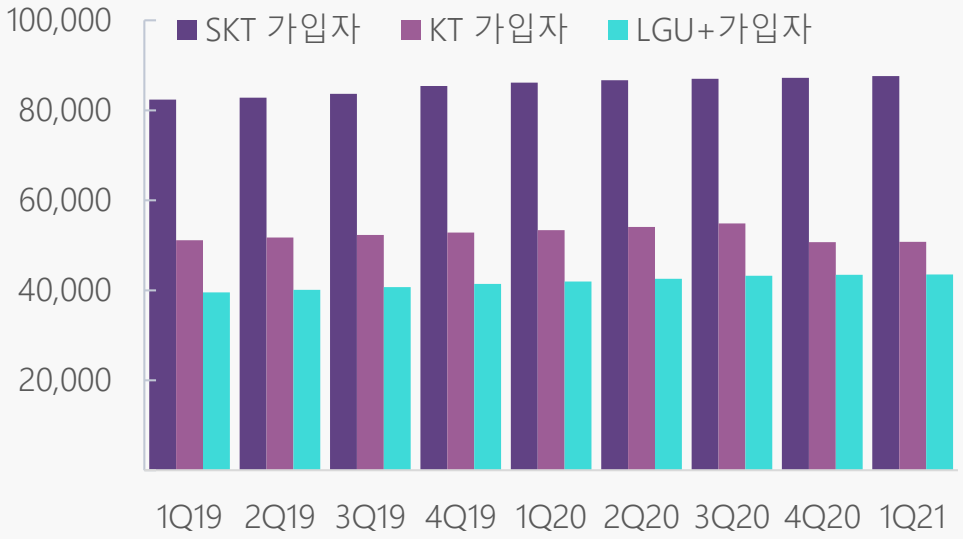
1. 프로젝트 개요

기획 배경 및 목표

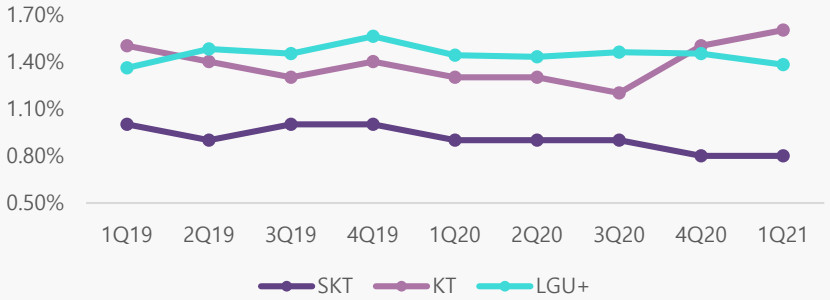
1) 통신사 현황

통신사 가입자 현황 (2020.08기준)

(단위: 만명)



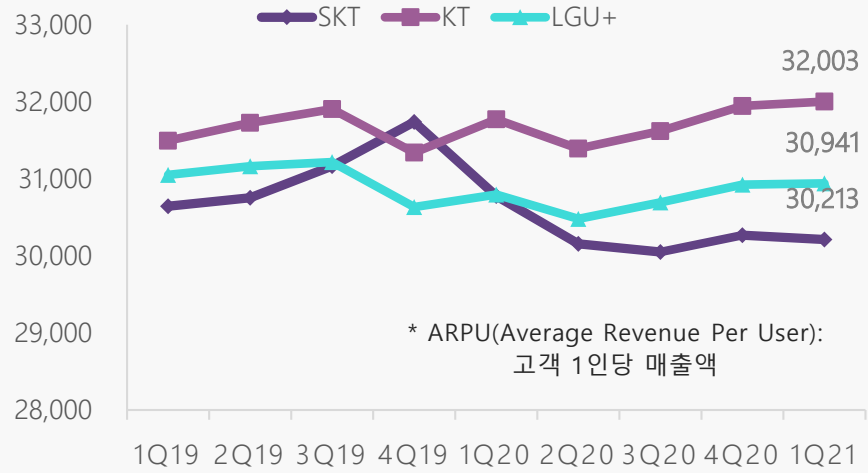
통신사 해지율



통신사 시장 과포화상태로 신규고객 확보가 어려운 실정임

통신사 ARPU* 추이 (2020.08기준)

(단위: 원)



5G 요금제로 1인당 **ARPU**가 상승, 고객 이탈 시 매출액에 큰 타격

기획 배경 및 목표

2) 문제 제기

기존고객	통신사 멤버십 제휴할인, 약정할인, 데이터 쿠폰 등 제공
신규고객	광고선전비, 보조금, 단말기 혹은 약정할인 등 비용 지출

통신사 영업비용 과중

3) 목표

머신러닝 모델을 활용한 고객 이탈 예측 모델을 통해

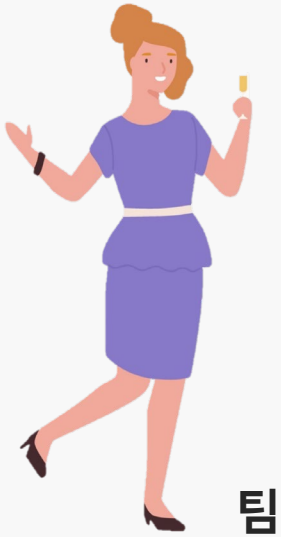
- ✓ **고객이탈예측**
- ✓ **불필요한 영업비용 절감**
- ✓ **고객 차별화 전략 수립**

기획 배경 및 목표

선행연구

배준영(2000)	데이터마이닝을 이용한 해지 예측 모델 비교연구 MLP 신경망과 로지스틱 분석 & 하이브리드 분석엔진을 구축하여 현업에서 사용할 수 있는 예상 해지자 리스트를 추출
유동균(2001)	데이터마이닝을 활용한 이탈고객 스코어링 모델 개발 데이터마이닝을 활용하여 자동차 보험 지원 시스템과 의사결정나무모형, 로지스틱 회귀모형, 신경망모형을 비교하여 최적 성능의 자동차 보험 고객이탈 스코어링 모델 개발
이명미 (2012)	SupportVectorMachine을 이용한 이동 통신사 고객이탈 예측모형연구 데이터 마이닝 기법으로 부터 구축된 모형의 예측 정확도를 비교함으로써 서포트 벡터 머신 (SupportVectorMachine)의 활용 가능성을 제시
김경태(2018)	딥 러닝과 Boosted Decision Tree를 활용한 고객 이탈 예측 모델 RF, XGBoost, RNN, CNN 등 기계학습을 활용하여 이탈할 가능성이 높은 고객을 예측
김형수 & 홍승우 (2020)	이차원 고객충성도 세그먼트 기반의 고객이탈예측 방법론 충성도 세그먼트 기반의 고객이탈예측 프로세스 (CCP/2DL: Customer Churn Prediction based on Two-Dimensional Loyalty segmentation)를 제안

구성원 및 역할



팀장 | 남예은

데이터 전처리 및 EDA

모델 적용

- 로지스틱 회귀
- 나이브 베이즈

데이터 수집

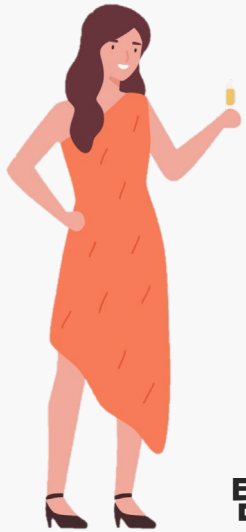
팀원 | 김영성

데이터 전처리 및 EDA

모델 적용

- KNN Classifier
- Random Forest
- SVM

모델 코드 작성 및 개선



팀원 | 김태리

데이터 수집 및 분석

모델 적용

- XGBOOST
- LGBM

PPT 작성 및 발표

WBS

일정: 2021년 08월 09일 ~ 2021년 08월 20일 (12일)

2. 프로세싱

프로세싱

Data Collection

데이터

Telco customer churn

Data Preprocessing

Data Imputation

Telco data 결측치 처리

범주형 -> 수치형 변환

EDA

Data Set 분석 & Feature Engineering

Churn rate (이탈율)

파생변수 생성

SMOTE 오버샘플링

Modeling

이진분류 모델 적용

KNN 모델

로지스틱 회귀 모델

의사결정나무

앙상블 모델

Optimization

Hyper parameter 튜닝

모델 개선 및 최적 성능 모델 선정

데이터 수집 및 변수의 정의

숫자열 (2개)	MonthlyCharges	매월 고객에게 청구되는 금액
	TotalCharges	고객에게 청구되는 총 금액
범주형 (19개)	CustomerID	각 고객의 고유한 고객ID
	gender	성별(Female, Male)
	SenierCitizen	고객이 노인인지 여부(1:노인, 0:아님)
	Partner	파트너 여부(Yes, No)
	Dependents	부양 여부(Yes, No)
	tenure	고객이 회사에 머무른 개월
	PhoneService	전화 서비스 여부(Yes, No)
	MultipleLines	고객의 다중 회선 유무(Yes, No, No phone service)
	InternetService	고객의 인터넷 서비스 제공 업체(DSL, Fiber optic, No)
	OnlineSecurity	온라인 보안 여부(Yes, No, No internet service)
	OnlineBackup	온라인 백업 여부(Yes, No, No internet service)
	DeviceProtection	고객이 기기 보호 기능을 제공 여부(Yes, No, No internet service)
	TechSupport	기술 지원을 받았는지(Yes, No, No internet service)
	StreamingTV	스트리밍TV 여부(Yes, No, No internet service)
	StreamingMovies	스트리밍 영화 여부(Yes, No, No internet service)
	Contact	계약 기간(Month-to-month, One year,Two year)
	PaperlessBilling	종이 명세서 여부(Yes, No)
	PaymentMethod	결제 수단(Electronic check, Bank transfer (automatic), Credit card Mailed check, (automatic))
	Churn (목표변수)	**이탈여부(1: Yes , 0: No)**

데이터 전처리

1) 결측치 처리

1 Total Charges Null 값 확인

```
data = data.drop('customerID', axis=1)
# to_numeric : 똑같은 형식의 숫자로 정렬, errors 옵션:
숫자 이외의 값을 어떻게 처리할지 설정
data['TotalCharges'] = pd.to_numeric(data['TotalCharges'], errors='coerce')
data.isnull().sum()
```

2 Total Charges Null 값 확인

```
data[np.isnan(data['TotalCharges'])]
```

3 결측치 대체값 입력

Total Charges Null 값은 Tenure가 0인 신규가입자임을 확인

>> MonthlyCharges 값을 TotalCharges 값으로 대체

```
data.TotalCharges.fillna(data.MonthlyCharges, inplace=True)
data.isnull().sum()
```

```
(gender            0
 SeniorCitizen     0
 Partner           0
 Dependents        0
 tenure            0
 PhoneService      0
 MultipleLines     0
 InternetService   0
 OnlineSecurity    0
 OnlineBackup      0
 DeviceProtection  0
 TechSupport       0
 StreamingTV       0
 StreamingMovies   0
 Contract          0
 PaperlessBilling  0
 PaymentMethod     0
 MonthlyCharges    0
 TotalCharges     11
 Churn             0
 dtype: int64, (7043, 20))
```

```
gender            0
SeniorCitizen     0
Partner           0
Dependents        0
tenure            0
PhoneService      0
MultipleLines     0
InternetService   0
OnlineSecurity    0
OnlineBackup      0
DeviceProtection  0
TechSupport       0
StreamingTV       0
StreamingMovies   0
Contract          0
PaperlessBilling  0
PaymentMethod     0
MonthlyCharges    0
TotalCharges      0
Churn             0
dtype: int64
```

데이터 전처리

2) 범주형 데이터 > 수치형 데이터로 변환

1 Label Encoding

```
# Label Encoding / Churn : 이탈안함 - 0 , 이탈 - 1
from sklearn.preprocessing import LabelEncoder
def object_to_int(data_ob):
    if data_ob.dtype == 'object':
        data_ob = LabelEncoder().fit_transform(data_ob)
    return data_ob
data = data.apply(lambda x: object_to_int(x))
```

2 Ordinal Encoding

```
data['gender'].replace({'Female':1, 'Male':0}, inplace=True)
# Ordinal Encoding
df = {'Month-to-month':3,'One year': 2,'Two year': 1}
data['Contract'] = data.Contract.map(df)
```

```
Data columns (total 20 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   gender                      7032 non-null   int64
 1   SeniorCitizen               7032 non-null   int64
 2   Partner                     7032 non-null   object
 3   Dependents                  7032 non-null   object
 4   tenure                      7032 non-null   int64
 5   PhoneService                7032 non-null   object
 6   MultipleLines               7032 non-null   object
 7   Contract                    7032 non-null   int64
 8   PaperlessBilling            7032 non-null   object
 9   MonthlyCharges              7032 non-null   float64
 10  TotalCharges                7032 non-null   float64
 11  Churn                       7032 non-null   object
 12  AdditionalService           7032 non-null   int64
 13  InternetService_DSL         7032 non-null   uint8
 14  InternetService_Fiber optic 7032 non-null   uint8
 15  InternetService_No          7032 non-null   uint8
 16  PaymentMethod_Bank_tf       7032 non-null   uint8
 17  PaymentMethod_Electronic    7032 non-null   uint8
 18  PaymentMethod_Mail          7032 non-null   uint8
 19  PaymentMethod_card          7032 non-null   uint8
dtypes: float64(2), int64(5), object(6), uint8(7)
memory usage: 1.1+ MB
```

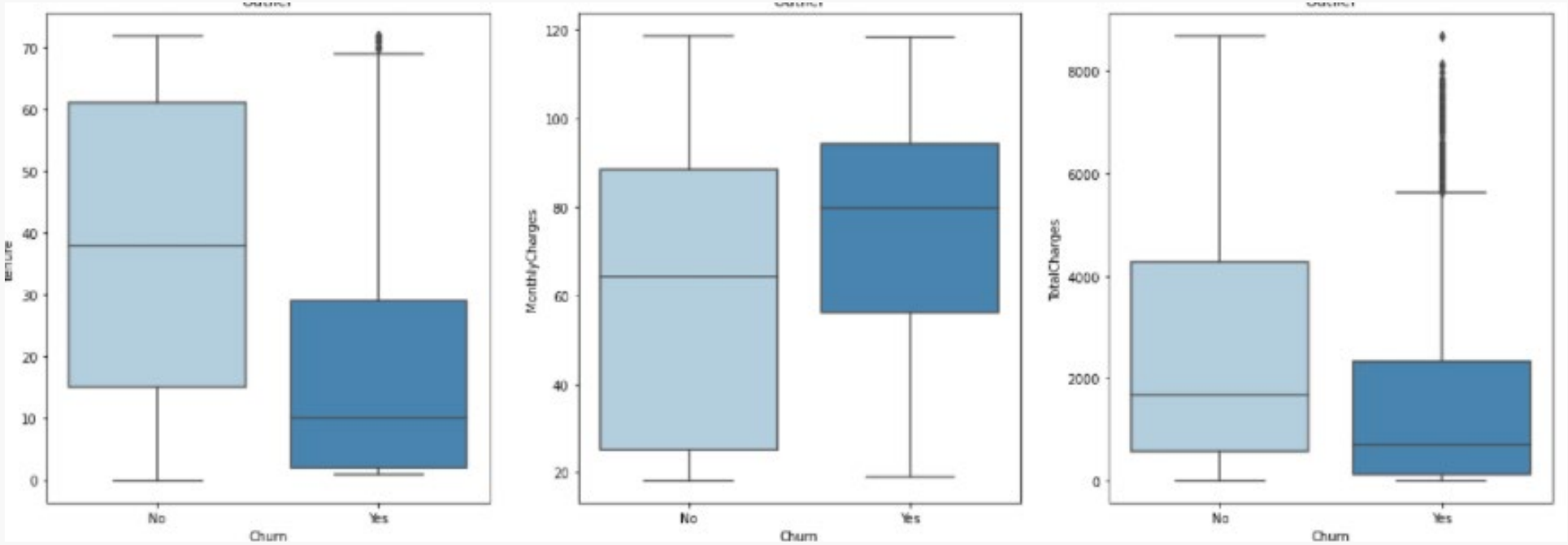
3) 데이터 정규화

1 StandardScaler

```
# StandardScaler
from sklearn.preprocessing import StandardScalerscaler = StandardScaler()
nor_col = ['tenure','MonthlyCharges','TotalCharges']
data[nor_col] = scaler.fit_transform(data[nor_col])
data.head(1)
```

데이터 전처리

4) 이상치 확인



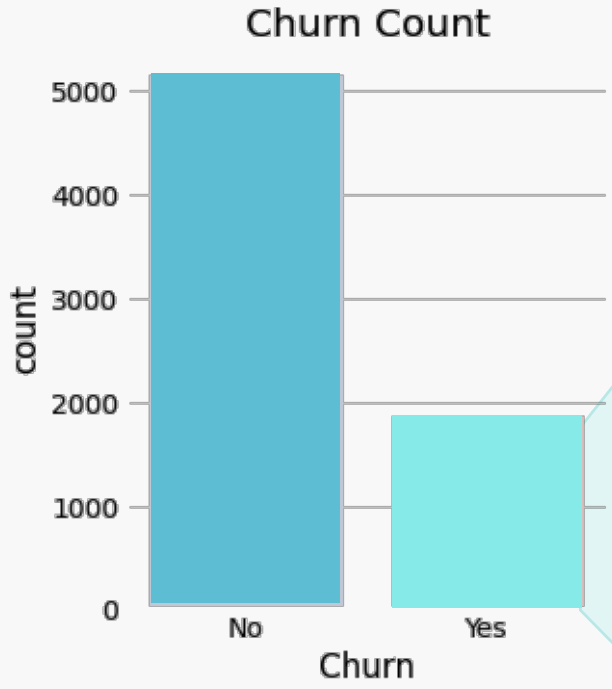
Tenure - Churn : 이탈하지 않은 고객의 중앙값인 38개월보다 아래인 10개월에서 30개월 사이에 많이 분포 되어있다.

Monthly Charges - Churn : 이탈자의 대부분은 60달러에서 90달러 사이의 월 요금을 납부하며 이용했다.

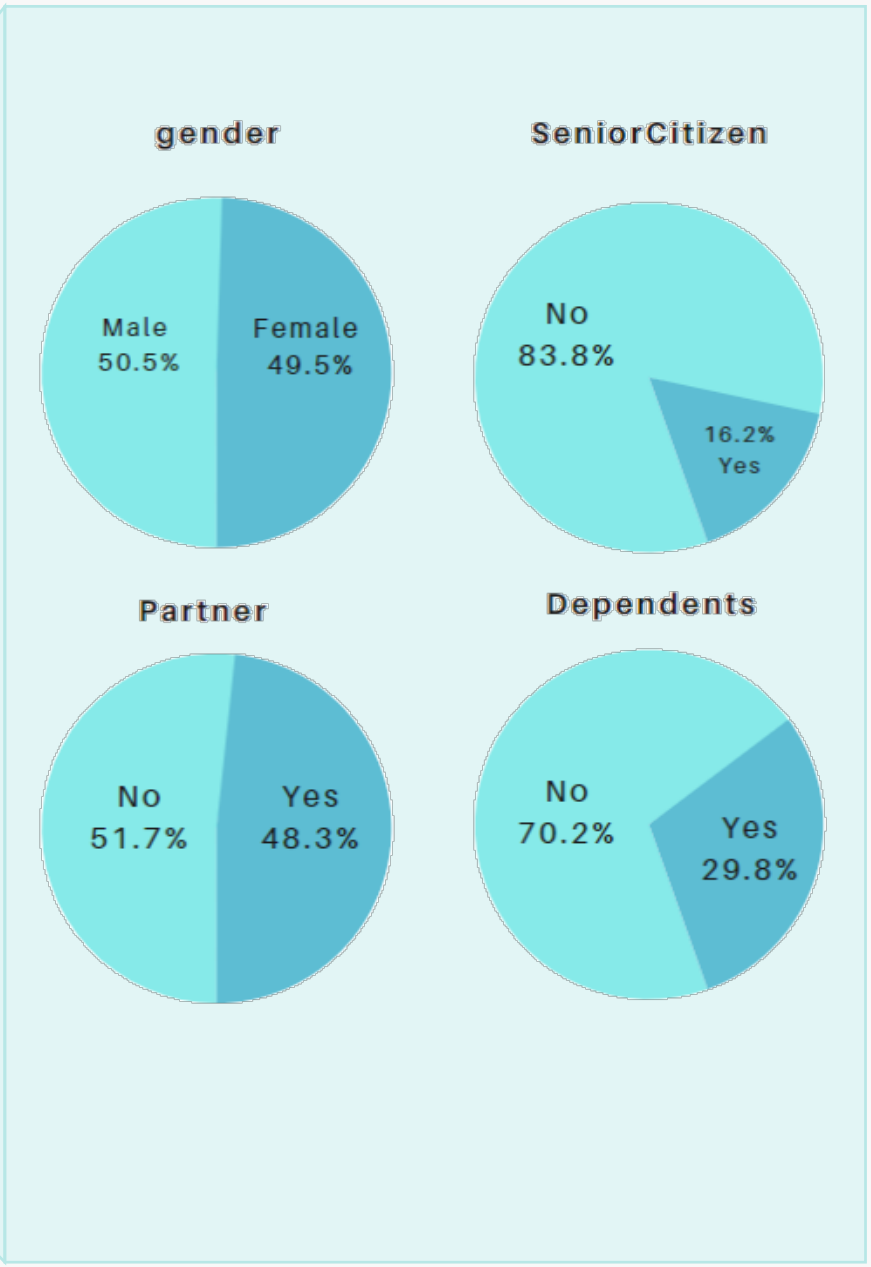
Total Charges - Churn : 월 요금의 누적 결과로 데이터가 불균형 함을 확인

EDA

1) 데이터의 관계파악

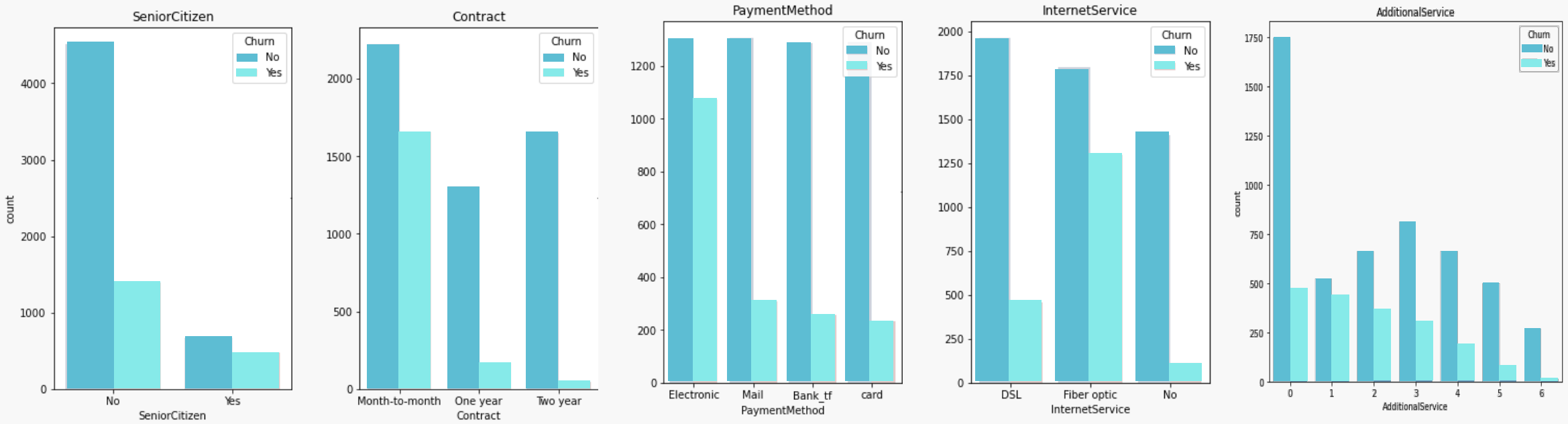


이탈율: 26.6 % (1,869 건/ 전체 7,043 개)



EDA

1) 데이터의 관계파악

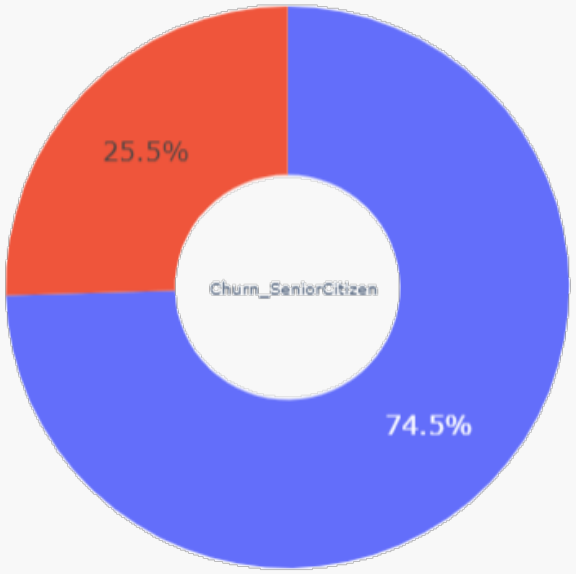


- 노년층 가입자 내 이탈비율은 높음
- 계약기간이 짧을 수록 이탈율이 높음
- Electronic Check 의 이탈율이 매우 높음 (Electronic Check : 전자수표)
- Fiber Optic을 사용하는 이용자의 이탈율이 매우 높음 (속도 : Fiber Optic > DSL)
- Additional service를 이용하지 않는 고객의 이탈율이 매우 높음

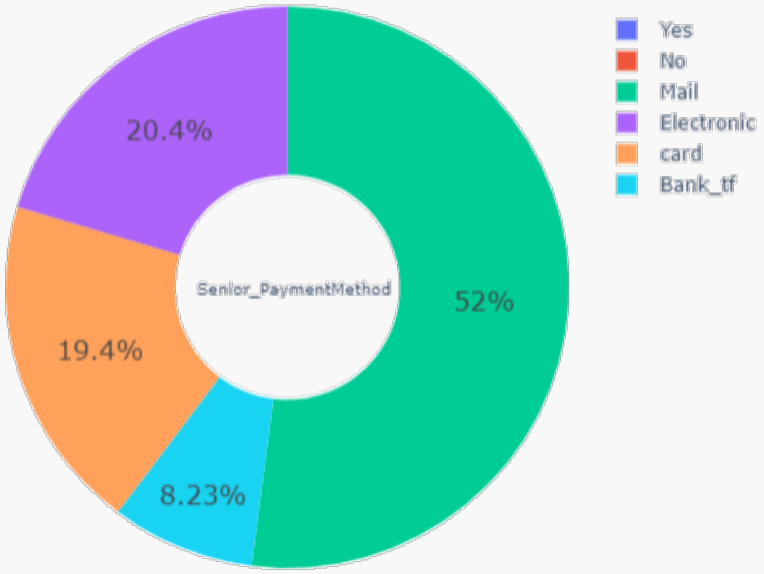
EDA

1) 데이터의 관계파악

노년층내 이탈비율



노년층 지불 방법



- 노년층 가입자 중 이탈자 74.5%
- 노년층 선호 지불 방법은 고지서(Mail) 지불방법 , 전자수표(Electronic Check) 지불 순으로 이루어짐

EDA

2) Feature Engineering _ 변수생성

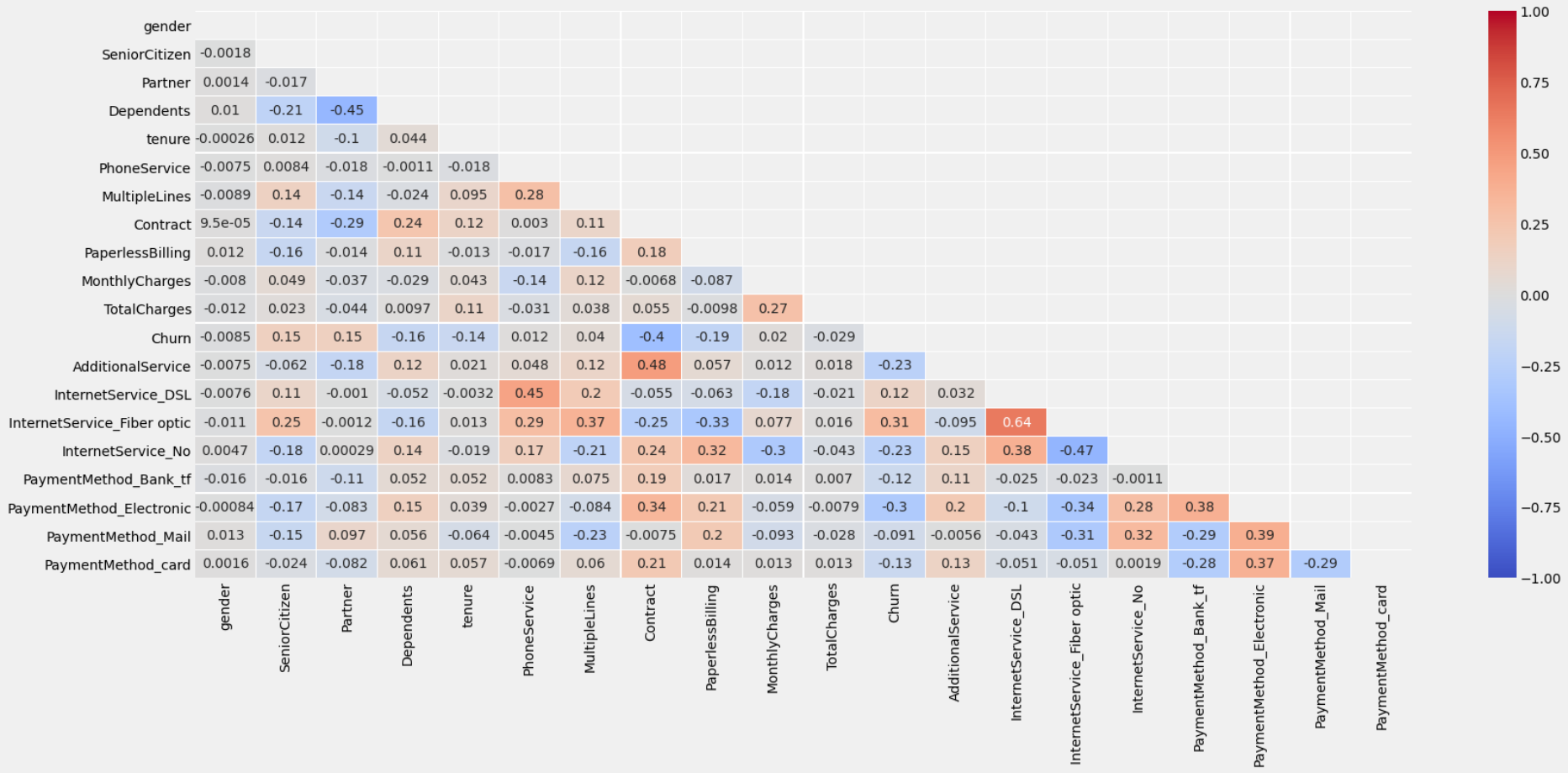
1	CustomerID	각 고객의 고유 한 고객 ID
2	gender	성별(Female, Male)
3	SenierCitizen	고객이 노인인지 여부 (1:노인, 0:아님)
4	Partner	파트너 여부 (Yes, No)
5	Dependents	부양 여부 (Yes, No)
6	tenure	고객이 회사에 머무른 개월
7	PhoneService	전화 서비스 여부 (Yes, No)
8	MultipleLines	고객의 다중 회선 유무 (Yes, No, No phone service)
9	InternetService	고객의 인터넷 서비스 제공 업체 (DSL, Fiber optic, No)
10	OnlineSecurity	온라인 보안 여부(Yes, No, No internet service)
11	OnlineBackup	온라인 백업 여부 (Yes, No, No internet service)
12	DeviceProtection	고객이 기기 보호 기능을 제공여부 (Yes, No, No internet service)
13	TechSupport	기술 지원을 받았는지 (Yes, No, No internet service)
14	StreamingTV	스트리밍 TV여부 (Yes, No, No internet service)
15	StreamingMovies	스트리밍 영화 여부 (Yes , No, No internet service)
16	contact	계약 기간 (Month-to-month, One year,Two year)
17	PaperlessBilling	종이 명세서 여부(Yes, No)
18	PaymentMethod	결제 수단 (Electronic check, Mailed check, Bank transfer (automatic),Credit card (automatic))
19	Churn	이탈여부 (Yes, No)
20	MonthlyCharges	매월 고객에게 청구되는 금액
21	TotalCharges	고객에게 청구되는 총 금액

→ 삭제

→ 파생변수 생성 (Additional Service)

EDA

2) Feature Engineering_변수간 상관관계확인



EDA

3) SMOTE 오버샘플링

```
from imblearn.over_sampling import SMOTE
smote = SMOTE(sampling_strategy='auto', random_state=2056)
X_train_o,y_train_o = smote.fit_sample(X_train,y_train)
print('SMOTE 적용 전 학습용 피처/레이블 데이터 세트: ', X_train.shape, y_train.shape)
print('SMOTE 적용 후 학습용 피처/레이블 데이터 세트: ', X_train_o.shape, y_train_o.shape)
print('SMOTE 적용 후 레이블 값 분포: \n', pd.Series(y_train_o).value_counts())
```

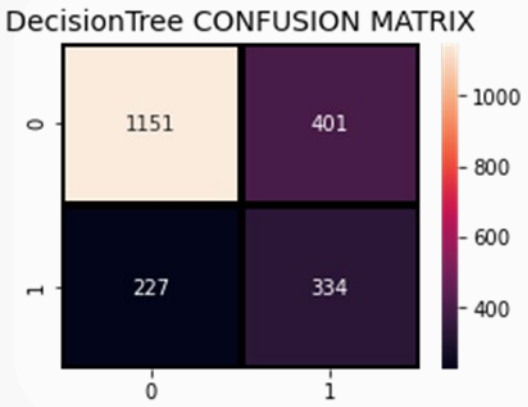
```
SMOTE 적용 전 학습용 피처/레이블 데이터 세트: (4930, 20) (4930,)
SMOTE 적용 후 학습용 피처/레이블 데이터 세트: (7244, 20) (7244,)
SMOTE 적용 후 레이블 값 분포: 1 3622 0 3622 dtype: int64
```

>> 불균형데이터로 인한 결과값의 성능저하를 방지하기 위해 SMOTE 오버샘플링 기법을 사용함

모델 적용 및 개선

1) Decision Tree

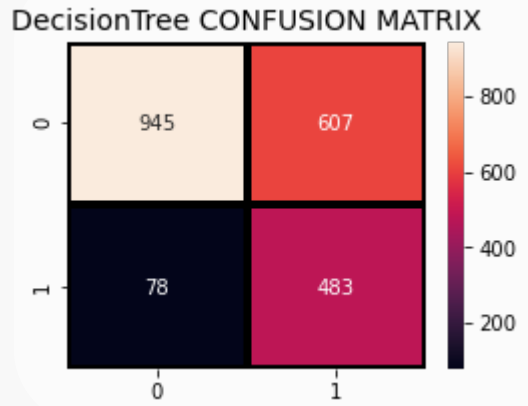
1. 하이퍼파라미터 튜닝 전



정확도: 0.70
정밀도: 0.46
재현율: 0.59
F1: 0.51
AUC:0.67

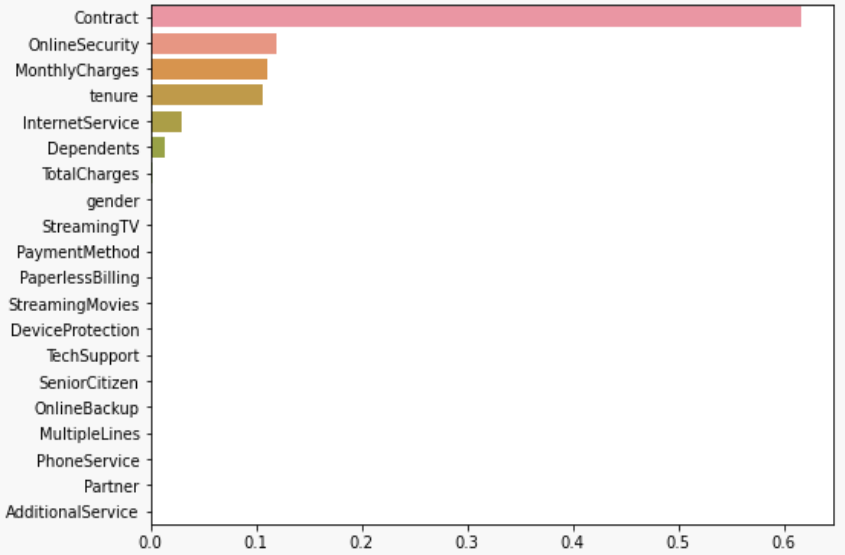
2. 하이퍼파라미터 튜닝 후

(max_depth = 4, min_samples_leaf =2, min_samples_split=2)

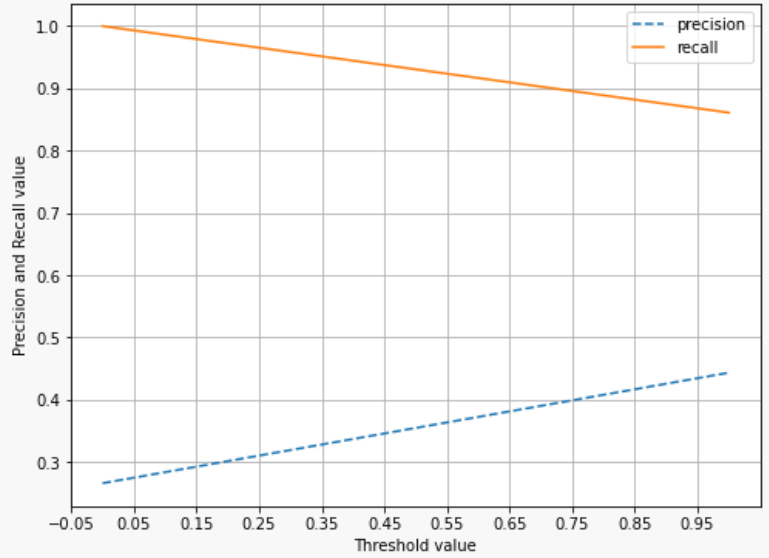


정확도: 0.68
정밀도: 0.44
재현율: 0.86
F1: 0.59
AUC:0.73

변수 중요도



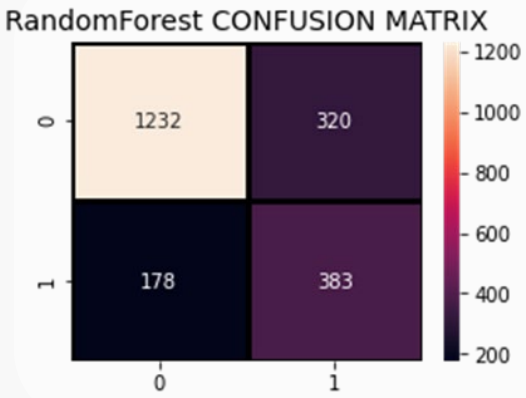
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

2) Random Forest

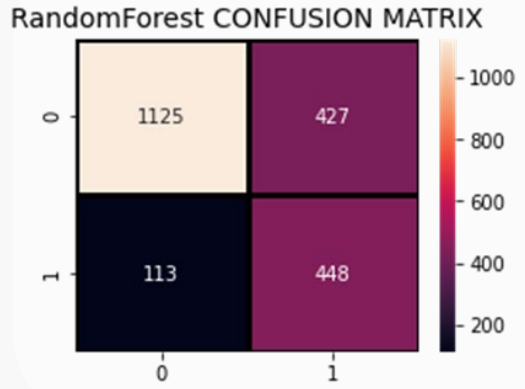
1. 하이퍼파라미터 튜닝 전



정확도: 0.76
정밀도: 0.54
재현율: 0.67
F1: 0.60
AUC:0.82

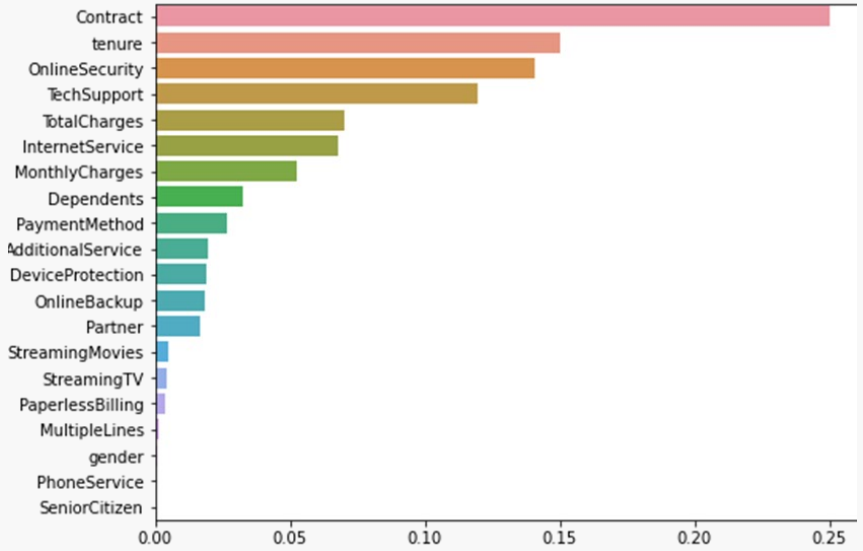
2. 하이퍼파라미터 튜닝 후

(max_depth=4, min_samples_leaf=2, min_samples_split=2)

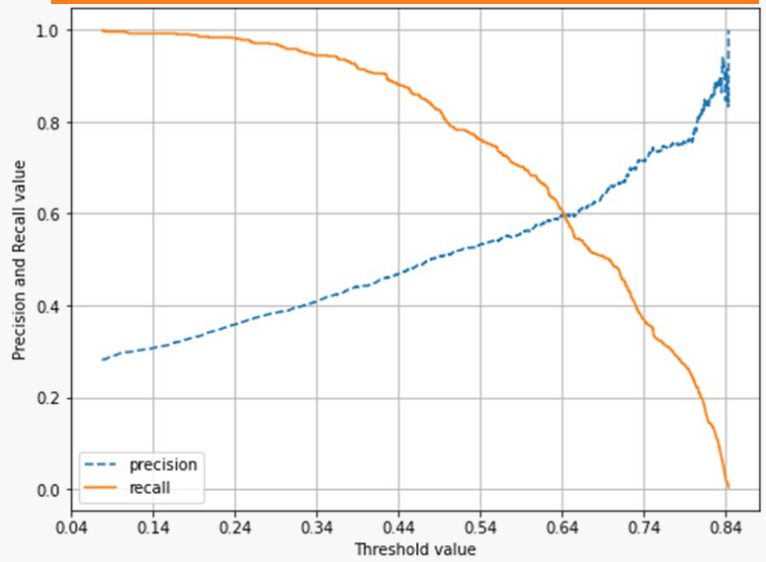


정확도: 0.74
정밀도: 0.51
재현율: 0.80
F1: 0.62
AUC:0.84

변수 중요도



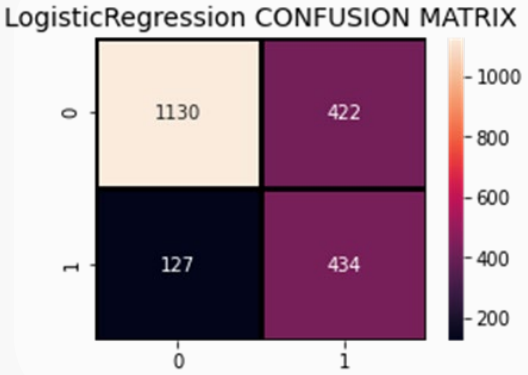
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

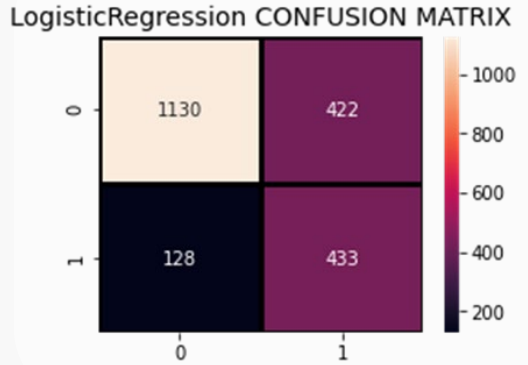
3) Logistic Regression

1. 하이퍼파라미터 튜닝 전



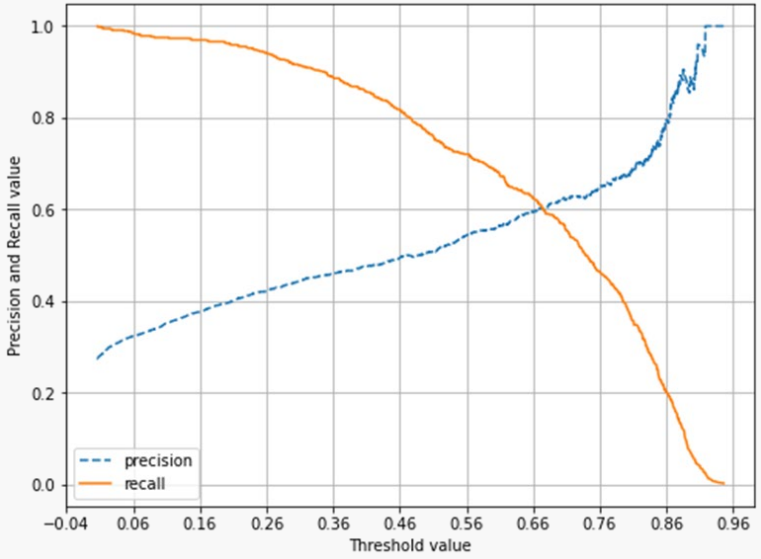
정확도: 0.74
정밀도: 0.51
재현율: 0.77
F1: 0.61
AUC:0.84

2. 하이퍼파라미터 튜닝 후 (C=5, penalty='l2')



정확도: 0.74
정밀도: 0.51
재현율: 0.77
F1: 0.61
AUC:0.84

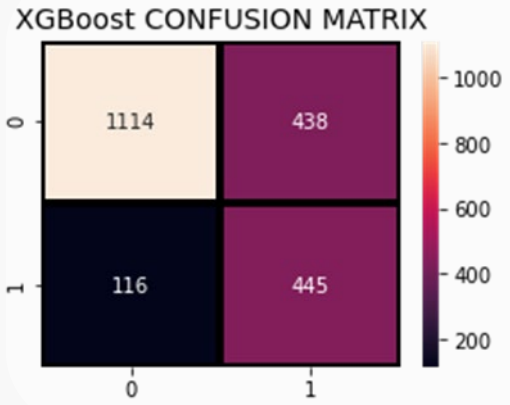
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

4) XGBOOST

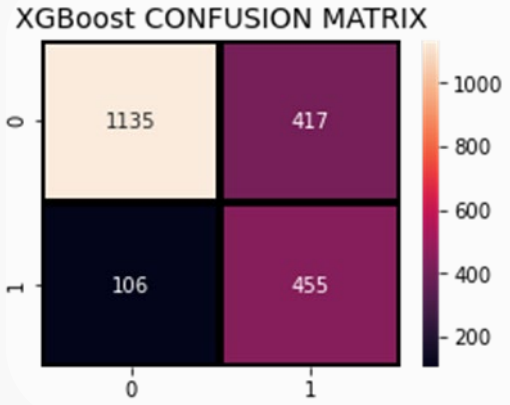
1. 하이퍼파라미터 튜닝 전



정확도: 0.74
정밀도: 0.50
재현율: 0.79
F1: 0.62
AUC:0.84

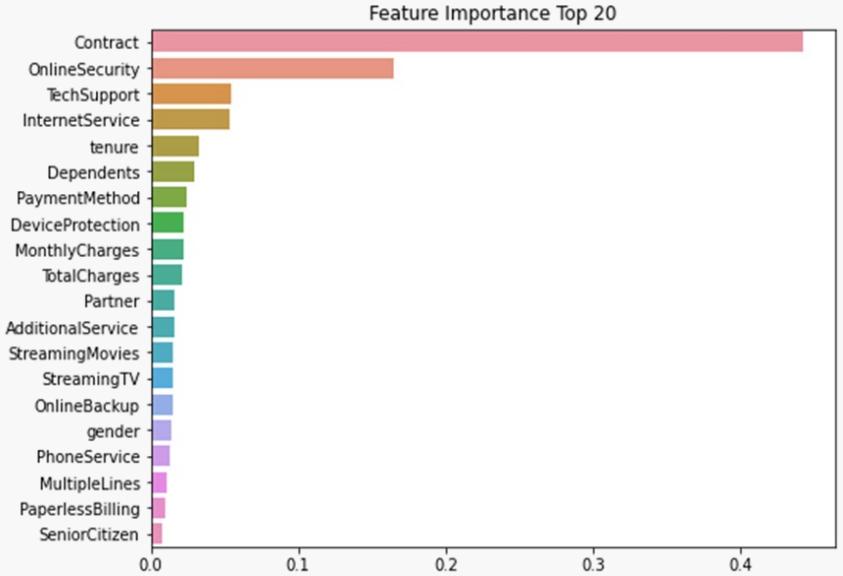
2. 하이퍼파라미터 튜닝 후

(lcolsample_bytree = 0.5,min_child_weight =1)

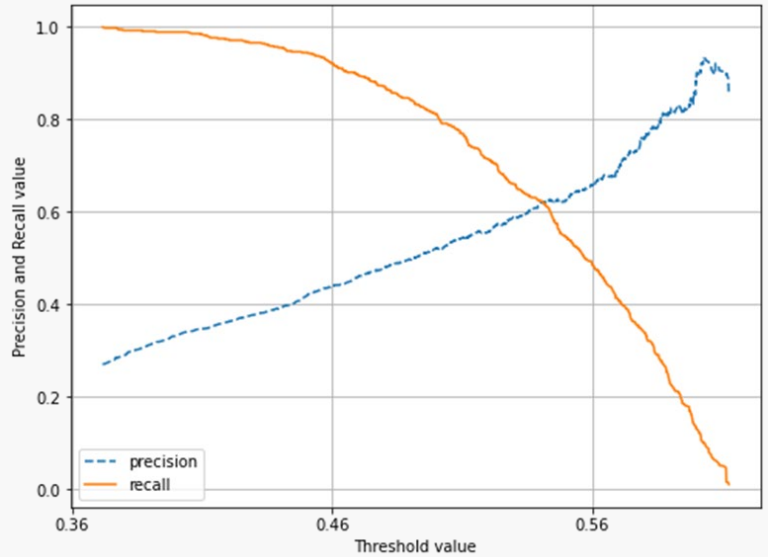


정확도: 0.75
정밀도: 0.52
재현율: 0.81
F1: 0.64
AUC:0.85

변수 중요도



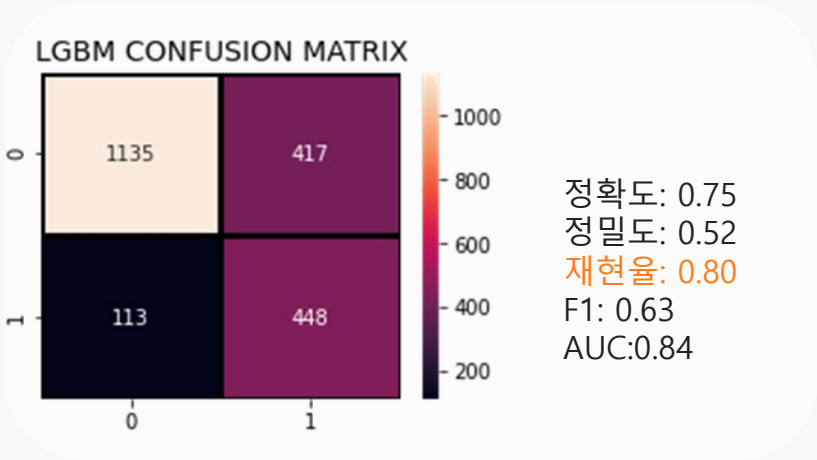
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

5) LightGBM

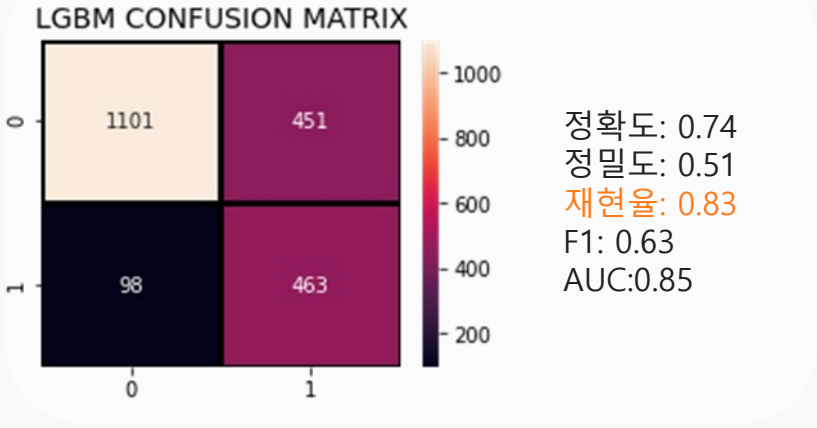
1. 하이퍼파라미터 튜닝 전



정확도: 0.75
정밀도: 0.52
재현율: 0.80
F1: 0.63
AUC:0.84

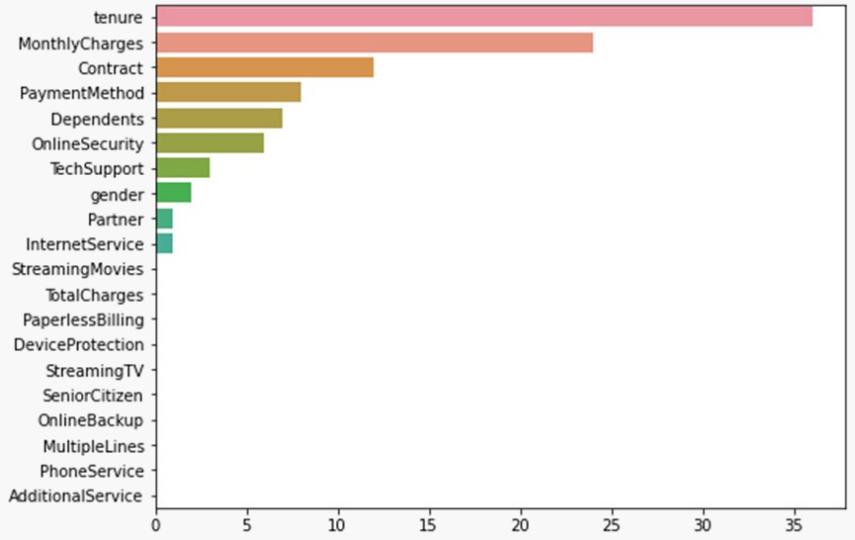
2. 하이퍼파라미터 튜닝 후

(num_leaves = 2,colsample_bytree = 0.5,min_child_weight =1)

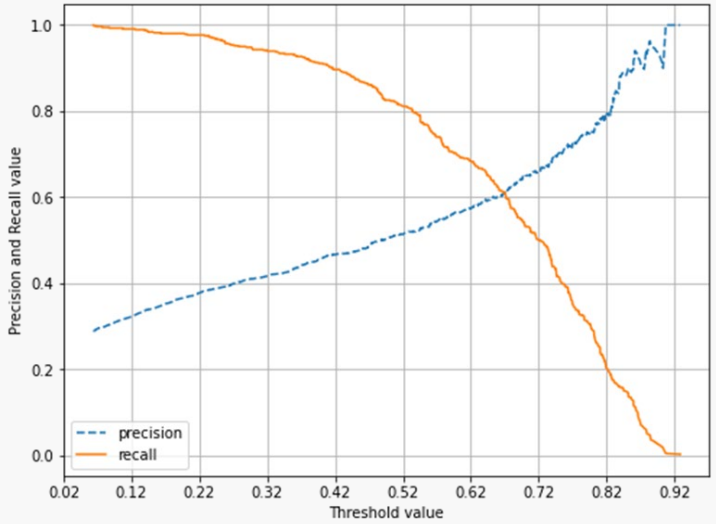


정확도: 0.74
정밀도: 0.51
재현율: 0.83
F1: 0.63
AUC:0.85

변수 중요도



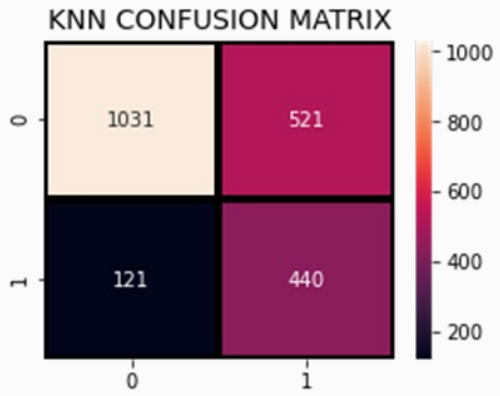
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

6) KNN

1. 하이퍼파라미터 튜닝 전 (n_neighbors = 11)



정확도: 0.70

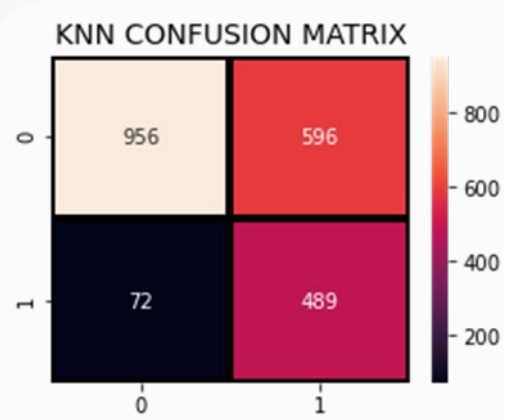
정밀도: 0.46

재현율: 0.78

F1: 0.58

AUC:0.79

2. 하이퍼파라미터 튜닝 후 (n_neighbors = 98)



정확도: 0.68

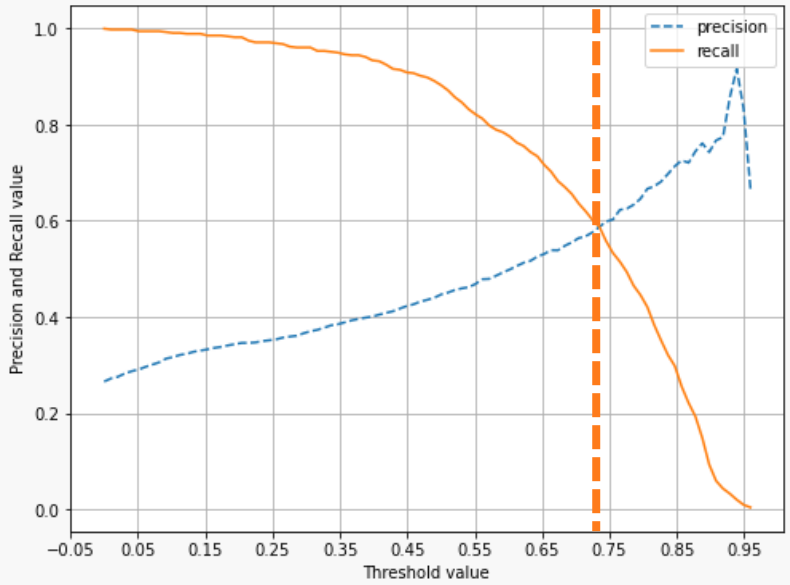
정밀도: 0.45

재현율: 0.87

F1: 0.59

AUC:0.83

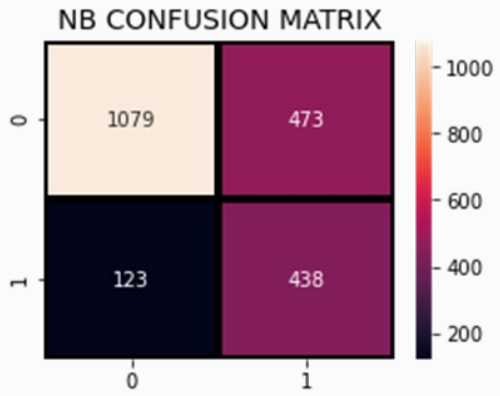
Precision and Recall Curve (정밀도 – 재현율 그래프)



모델 적용 및 개선

7) Naive Bayse

1. 하이퍼파라미터 튜닝 전



정확도: 0.72

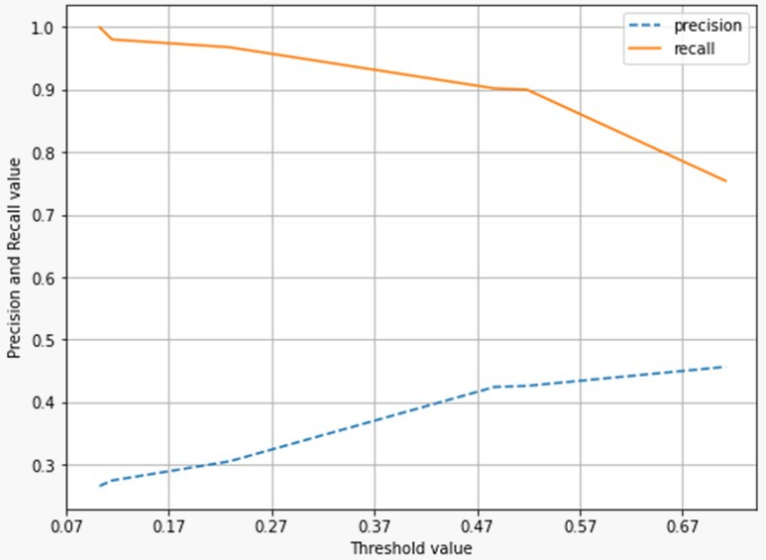
정밀도: 0.48

재현율: 0.77

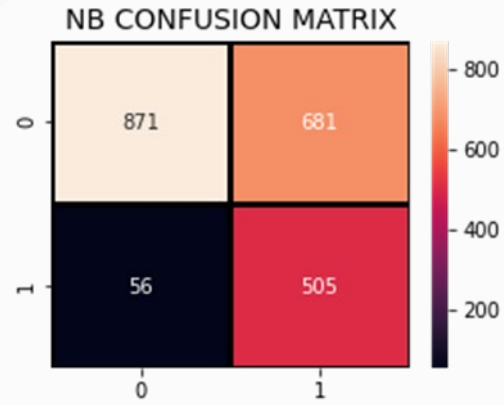
F1: 0.59

AUC:0.81

Precision and Recall Curve (정밀도 – 재현율 그래프)



2. 하이퍼파라미터 튜닝 (alpha=10,binarize=2)



정확도: 0.65

정밀도: 0.43

재현율: 0.90

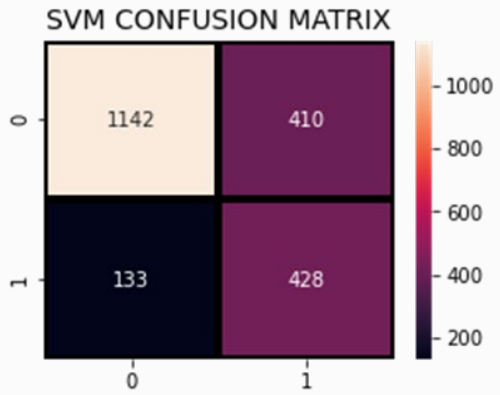
F1: 0.58

AUC:0.75

모델 적용 및 개선

8) SVM

1. 하이퍼파라미터 튜닝 전



정확도: 0.74

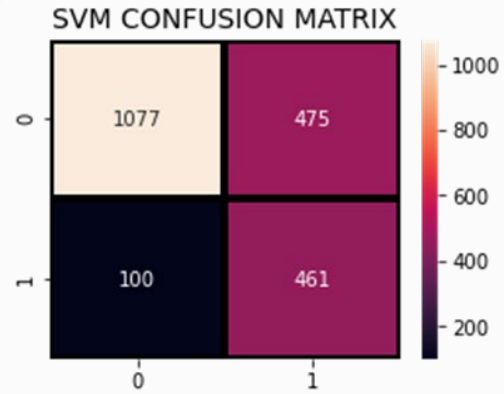
정밀도: 0.51

재현율: 0.76

F1: 0.61

2. 하이퍼파라미터 튜닝 후

(kernel='rbf',gamma=0.001, degree=2, C=1)



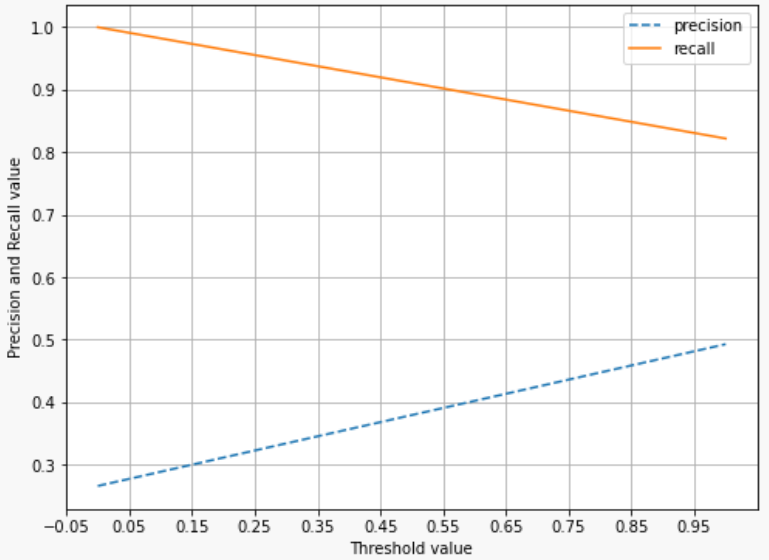
정확도: 0.73

정밀도: 0.49

재현율: 0.82

F1: 0.62

Precision and Recall Curve (정밀도 – 재현율 그래프)

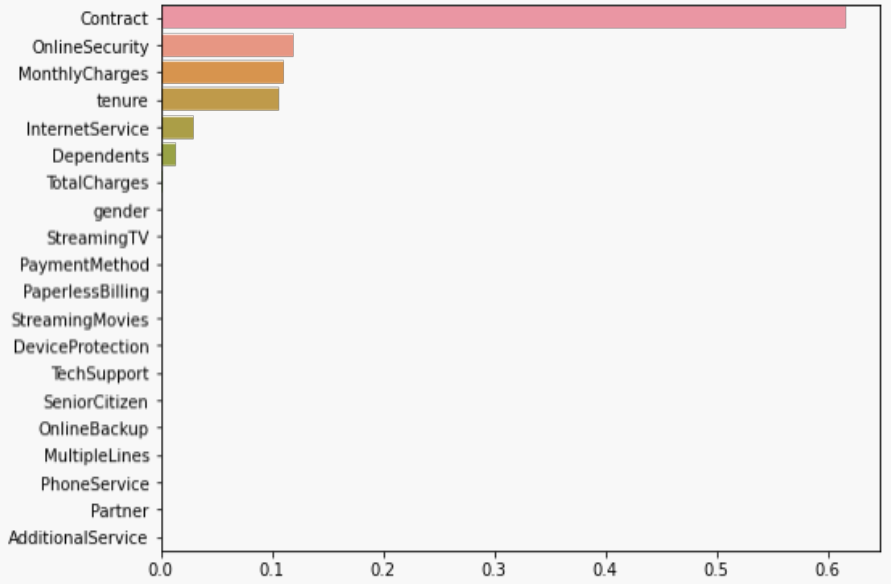


모델 적용 및 개선

9) 모델 성능 비교 표

모델명	재현율	
	개선전	개선후
Decision Tree	0.59	0.86
Random Forest	0.67	0.80
Logistic Regression	0.77	0.77
XGBOOST	0.79	0.81
LightGBM	0.80	0.83
KNN	0.78	0.87
Naive Bayse	0.78	0.90
SVM	0.76	0.82

Decision Tree 의 Feature Importance

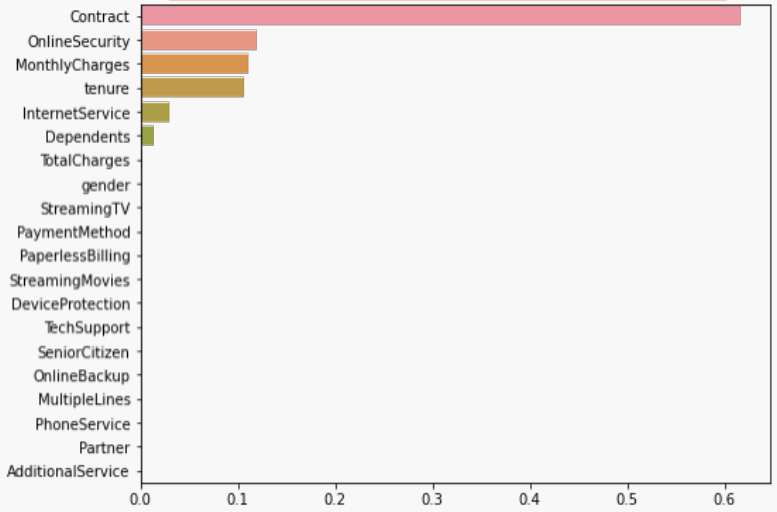


- NB 모델 성능(0.90)로 가장 좋음
- Decision Tree (0.86) 선택

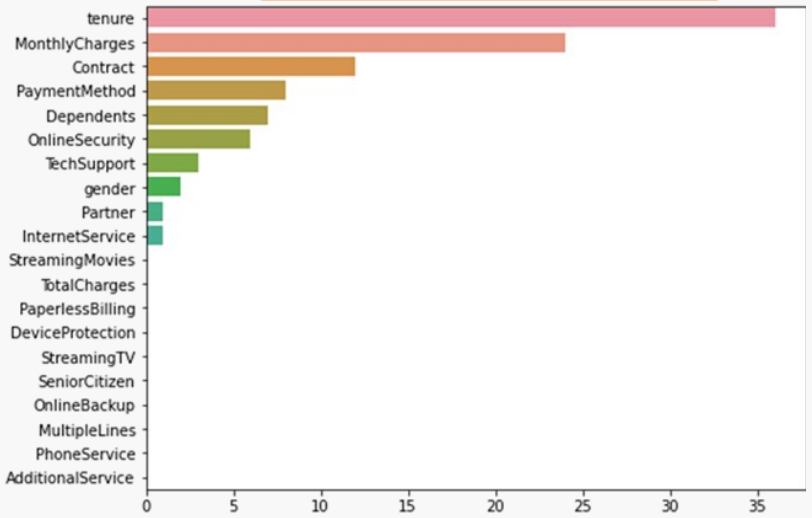
3. 기대효과

분석을 통한 인사이트 도출

Decision Tree 의 Feature Importance

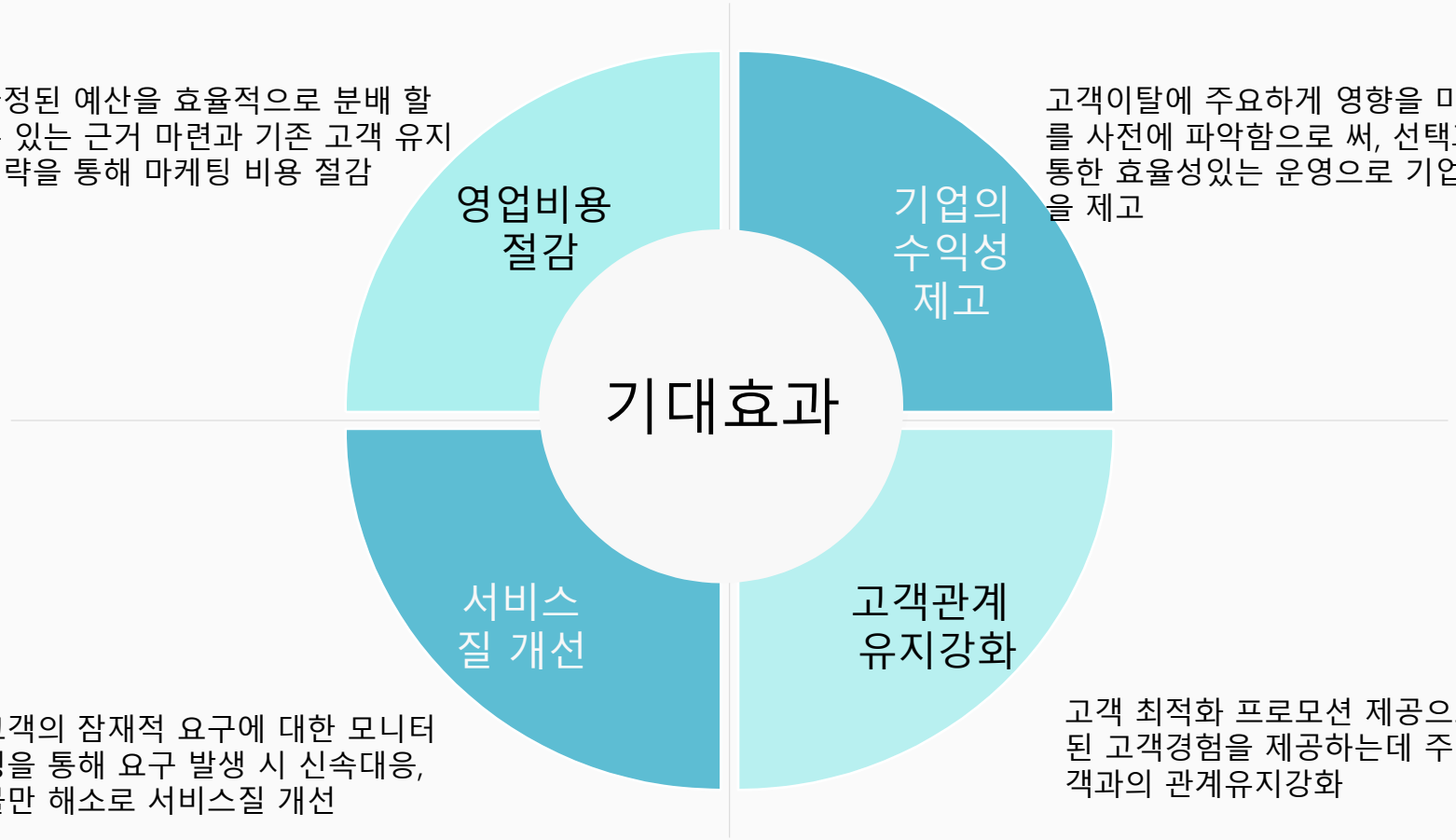


LGBM의 Feature Importance



1. 장기계약 고객에 대한 추가 할인 및 부가서비스 제공
2. 인터넷 보안 시스템 보완 및 강화
3. 인터넷 회선 + 자녀 통신비 결합 할인 혜택 강화

향후 개선사항 및 기대효과



한정된 예산을 효율적으로 분배 할 수 있는 근거 마련과 기존 고객 유지 전략을 통해 마케팅 비용 절감

고객이탈에 주요하게 영향을 미치는 변수를 사전에 파악함으로 써, 선택과 집중을 통한 효율성있는 운영으로 기업의 수익성을 제고

고객의 잠재적 요구에 대한 모니터링을 통해 요구 발생 시 신속대응, 불만 해소로 서비스질 개선

고객 최적화 프로모션 제공으로 맞춤화된 고객경험을 제공하는데 주력하여 고객과의 관계유지강화

개선사항 고객 데이터에 대한 자세한 변수가 누락되어 있음, 추후 테스트에서 연령별, 사용기기별 분석이 추가적으로 필요함

기대효과 국내 데이터를 적용하여 모델 성능의 추가적인 테스트가 진행된다면 국내 현황에 최적화된 모델의 구현이 가능할 것으로 기대

4. 개발 후기 및 느낀점

개발 후기 및 느낀점

남예은 | 팀장

수업시간에 배운 내용을 실제로 활용해 볼 수 있어서 좋았고, 실제로 해봄으로써 놓치고 갔던 부분들을 다시 공부할수있는 시간이 되어서 좋았습니다. 문제에 대한 즉각적인 피드백과 함께 문제를 해결해 나가는 과정에서 협업의 중요성을 배울 수 있었고, 특히나 모델링 기법에 대해 팀원들과 다양한 의견을 나눌 수 있는 기회가 많아서 좋았습니다. 부족했던 저를 많이 도와주고 값진 경험을 함께해준 팀원들에게 정말 감사드립니다.

김영성| 팀원

주제선정부터 모델링, 결과도출 까지 팀원분들과 서로 소통하며 통해 문제를 해결해 나가면서 협업이라는 부분을 직접 느낄수 있는 시간이어서 좋았습니다. 또한 전처리 부터 모델링을 하고 최적의 모델을 튜닝하는 과정을 하며 그동안 위축되었던 심리가 자신감으로 바뀌게 되는 기회였습니다

김태리 | 팀원

평소 궁금하던 사항을 머신러닝 프로젝트를 통해 진행하게 되어서 좋았고, 무엇보다 수업에서 배운 내용을 직접 적용하는 데에서 오류를 겪으면서 더욱 많은 공부가 되었습니다. 본 프로젝트에서는 실제 현업에서 적용해 볼 수 있는 주제를 선택했다는 점에서 보람이 있었으며 프로젝트가 끝나도 해당 예측모델에 대해 더 깊이 알아보고자 하는 마음이 들었습니다. 해당 프로젝트를 진행하면서 스스로 아쉬웠던 점이 많아 앞으로 프로젝트에서는 더욱 열심히 공부하고 실제로 적용해보는 연습을 지속적으로 해보고자 합니다.

참고문헌

김경태, & 이지형. (2018). 딥 러닝과 Boosted Decision Tree 를 활용한 고객 이탈 예측 모델. *한국지능시스템학회 논문지*, *28*(1), 7-12..

김기태, 이보미, & 김종우. (2017). 이진 분류문제에서의 딥러닝 알고리즘의 활용 가능성 평가. *지능정보연구*, *23*(1), 95-108.

김재엽. (2020). *고객이탈 예측 모델링 기반 기대수익 최적화 방안* (Doctoral dissertation, 서울대학교 대학원).

김충영, 장남식, & 김준우. (2002). 이동통신서비스 해지고객 예측 모형의 비교 분석에 관한 연구. *Asia Pacific Journal of Information Systems*, *12*(1), 139-158.

김형수, & 홍승우. (2020). 이차원 고객충성도 세그먼트 기반의 고객이탈예측 방법론. *지능정보연구*, *26*(4), 111-126.

배준영. "데이터마이닝을 이용한 해지 예측 모델 비교연구." 국내석사학위논문 漢陽大學校 大學院, 2000. 서울

이명미. (2012). Support Vector Machine 을 이용한 이동 통신사 고객이탈 예측모형연구.

한상태, 이성건, 강현철, & 유동균. (2001). 데이터마이닝을 활용한 이탈고객 스코어링 모델 개발. *한국통계학회 학술발표논문집*, 155-161.

2021년 2분기 실적발표자료(KT, SKT, LG U+)

수행도구

2021 머신러닝 프로젝트

Be전공자