

EMB00004ad874b6

**빅데이터 분석 중간고사 대체 과제**

EMB00004ad874b6

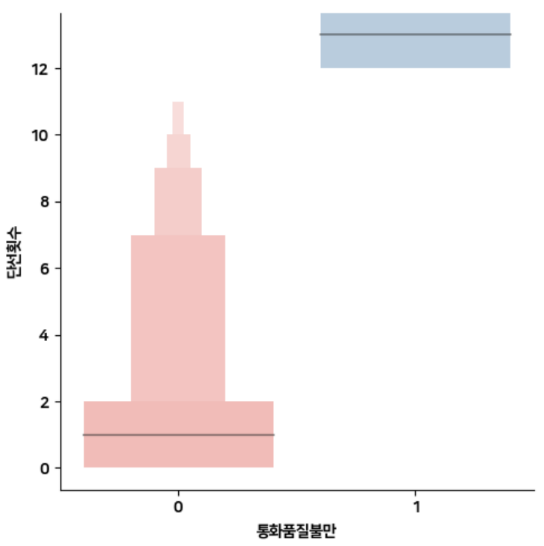
**Telco data를 통한 이탈 고객 특성 분석**

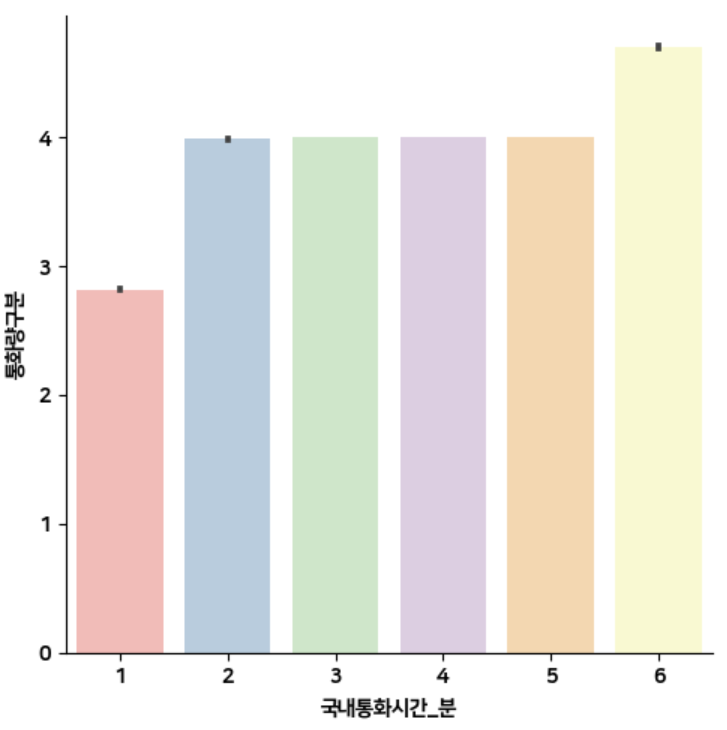
|  |  |
| --- | --- |
| **과목명** | 빅데이터 분석 |
| **교수명** | 김희찬 |
| **학 과** | 소프트웨어학부 |
| **학 번** | 20160272 |
| **이 름** | 김영현 |
| **제출일** | 2020.05.23 |

1. **데이터 전처리**

속성값이 많기 때문에 필요 없다고 생각되는 속성들은 삭제하고, 서로 연관이 있는 속성들은 하나만 남기고 제외하여 데이터를 변경하였다.

sns.catplot(x="통화품질불만", y="단선횟수", data=telco\_data, kind="bar", palette="Pastel1")  
plt.show()



‘

단선횟수’가 11이상이면 모두 ‘통화품질불만’이 T이기 때문에 ‘통화품질불만’ 속성만을 선택했다.

‘id’, ‘고객ID’, ‘개시일’, ‘핸드셋’은 ‘이탈여부’에 영향을 미치지 않기 때문에 삭제했다.

‘지불방법’은 모두 선불이기 때문에 삭제했다.

‘\*\*통화시간\_분’ / ‘\*\*통화횟수’ = ‘평균\*\*통화시간’이기 때문에 ‘\*\*통화시간\_분’ 속성을 선택했다.그리고 ‘주간+야간+주말통화시간\_분’은 ‘국내통화시간\_분’이기 때문에 ‘국내통화시간\_분’ 속성과 ‘국제통화시간\_분’을 선택했다.

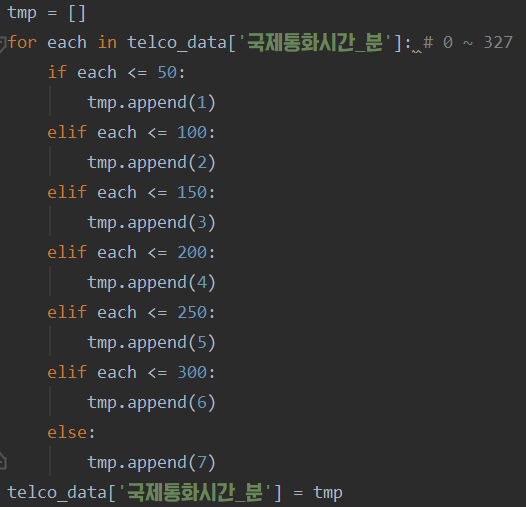
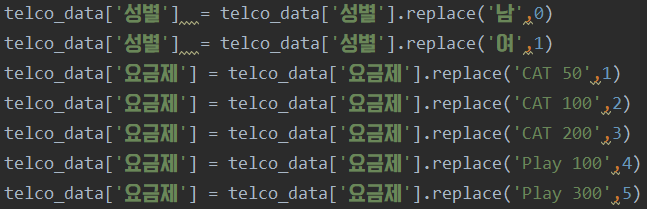
‘총통화시간\_분’은 ‘국제통화시간\_분’ + ‘국내통화시간\_분’이기 때문에 삭제했다.

‘국내통화시간\_분’은 ‘통화량구분’에 많은 영향을 끼치고, 비례관계이기 때문에 삭제했다.

‘요금부과시간’, ‘분당통화요금’, ‘국내통화요금’, ‘총통화요금’, ‘부과요금’은 모두 ‘평균납부요금’과 관련이 있기 때문에 ‘평균납부요금’ 속성만을 선택했다.

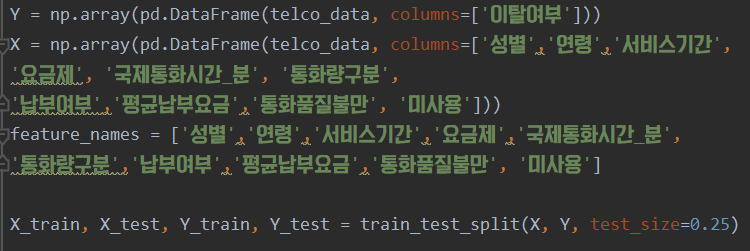
‘\*\*통화비율’은 ‘\*\*통화시간\_분’과 관련이 있기 때문에 삭제했다.

그렇게 해서 모델링을 하기 위해 선택한 속성들은 ‘**성별’, ‘연령’, 서비스기간’, ‘요금제’, 국제통화시간\_분’, ‘통화량구분’, ‘납부여부’, 평균납부요금’, ‘통화품질불만’, ‘미사용’**이다.

그리고 숫자가 아닌 속성값들을 상호연산이 가능하도록 numeric하게 변경해주고 또한 숫자의 범위가 넓은 속성들은 범위를 지정해 스케일링을 진행했다.

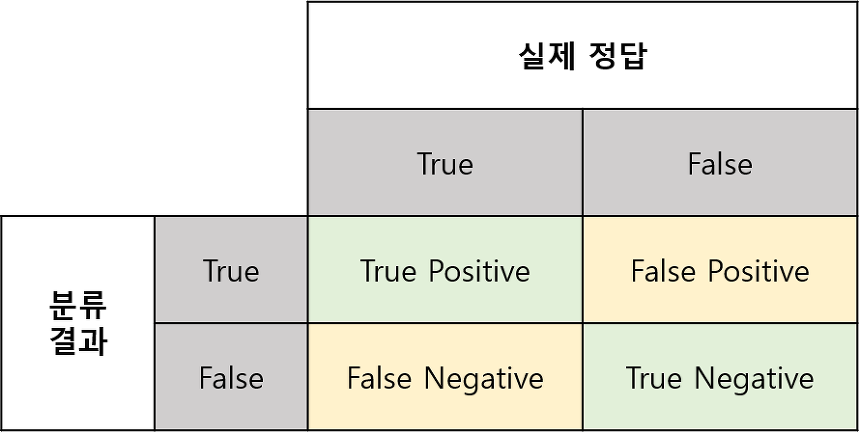
(같은 내용의 코드는 생략했다.)

마지막으로 test data의 비율을 0.25로 하여 전처리를 완료했다.



1. **모델 튜닝과 학습 및 평가**

모델을 평가하는 요소는 모델이 내놓은 답과 실제 정답의 관계로써 정의를 내릴 수 있다. 정답이 True와 False로 나누어져 있고, 분류 모델 또한 True False의 답을 내놓는다. 그렇게 하면, 아래 그림과 같이 case를 나눌 수 있다.



* True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)
* False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)
* False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)
* True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

정밀도(Precision)은 모델이 True라고 분류한 것 중에 실제 True의 비율을 말한다. (이 데이터에서의 Precision은 모델이 이탈할 것이라고 분류한 고객들 중 실제로 이탈한 고객의 비율)

재현율(Recall)은 실제 True 중 모델이 True라고 예측한 것의 비율을 말한다. (이 데이터에서의 Recall은 실제로 이탈한 고객들 중 모델이 이탈할 것이라고 분류한 고객의 비율)

정확도(Accuracy)는 False를 False라고 예측한 것도 고려하는 지표이다. (이 데이터에서의 Accuracy는 테스트 데이터의 모든 고객들 중 이탈, 유지 예측에 성공한 고객의 비율)

# =============================== 모델 튜닝 ===============================  
  
train\_result = []  
test\_result = []  
model\_criterion = []  
model\_max\_depth = []  
  
insert\_criterion = ['gini','entropy']  
list\_max\_depth = [4, 8, 12, 15, 20] # depth가 깊어질수록 train score는 상승 하나 test score는 하락한다. 과적합 되었기 때문  
for i in insert\_criterion:  
 for n in list\_max\_depth:  
 DesTree = tree.DecisionTreeClassifier(criterion=i, max\_depth=n)  
 DesTree.fit(X\_train,Y\_train)  
 train\_result.append(DesTree.score(X\_train,Y\_train))  
 test\_result.append(DesTree.score(X\_test,Y\_test))  
 model\_criterion.append(i)  
 model\_max\_depth.append(n)  
  
result = pd.DataFrame()  
result["Criterion"] = model\_criterion  
result["Depth"] = model\_max\_depth  
result["TrainAccuracy"] = train\_result  
result["TestAccuracy"] = test\_result  
print("\n -- 모델 튜닝 결과 -- ")  
print(result)  
  
  
sorted\_result = result.sort\_values(by='TestAccuracy', ascending=False)  
  
print("\n -- 최적 모델 -- ")  
print(sorted\_result.head(1))  
  
criterion = sorted\_result.head(1)['Criterion'].iloc[0]  
max\_depth = sorted\_result.head(1)["Depth"].iloc[0]  
  
clf = tree.DecisionTreeClassifier(criterion=criterion,max\_depth=max\_depth)  
clf.fit(X\_train, Y\_train)  
  
pred = clf.predict(X\_test)  
prec = precision\_score(Y\_test, pred, average='macro')  
rec = recall\_score(Y\_test,pred, average='macro')  
f1 = f1\_score(Y\_test, pred ,average='macro')  
acc = accuracy\_score(Y\_test,pred)

Decision Tree는 결과를 해석하고 이해하기 쉽다. 그리고 안정적이며 대규모의 데이터 셋에도 잘 동작하기 때문에 이 모델을 선택했다. DecisionTreeClassifier()은 파라미터에 criterion과 max\_depth를 주어 튜닝을 할 수 있는데, criterion에는 gini와 entropy를 부여하고 max\_depth에는 4, 8, 12, 15, 20을 부여하여 가장 높은 Test Accuracy를 찾아 최적의 모델을 사용하였다.

텍스트, 앉아있는, 검은색, 테이블이(가) 표시된 사진

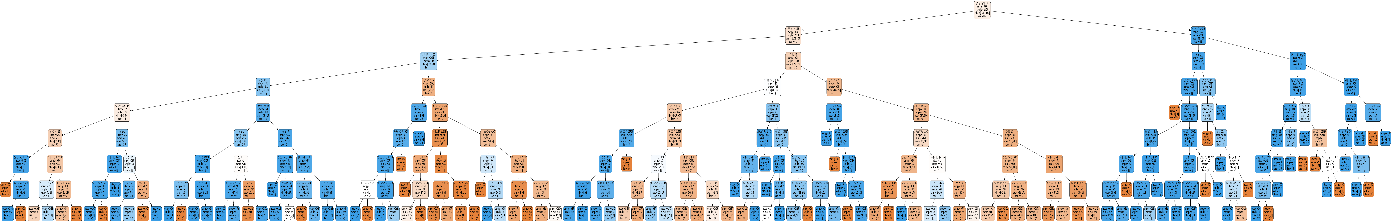
자동 생성된 설명

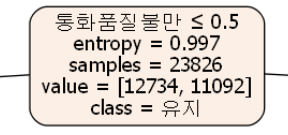
지니 불순도는 집합에 이질적인 것이 얼마나 섞였는지를 측정하는 지표이다. 어떤 집합에서 한 항목을 뽑아 무작위로 라벨을 추정할 때 틀릴 확률을 말한다. 집합에 있는 항목이 모두 같다면 지니 불순도는 최솟값(0)을 갖게 되며 이 집합은 완전히 순수하다고 할 수 있다.

엔트로피는 분자의 무질서함을 측정하는 것으로 원래 열역학의 개념이다. 정확히 잘 예측했다면 entropy는 최솟값(0)이 된다. 즉 어떤 세트가 한 클래스의 샘플만 담고 있다면 엔트로피가 0이 된다.

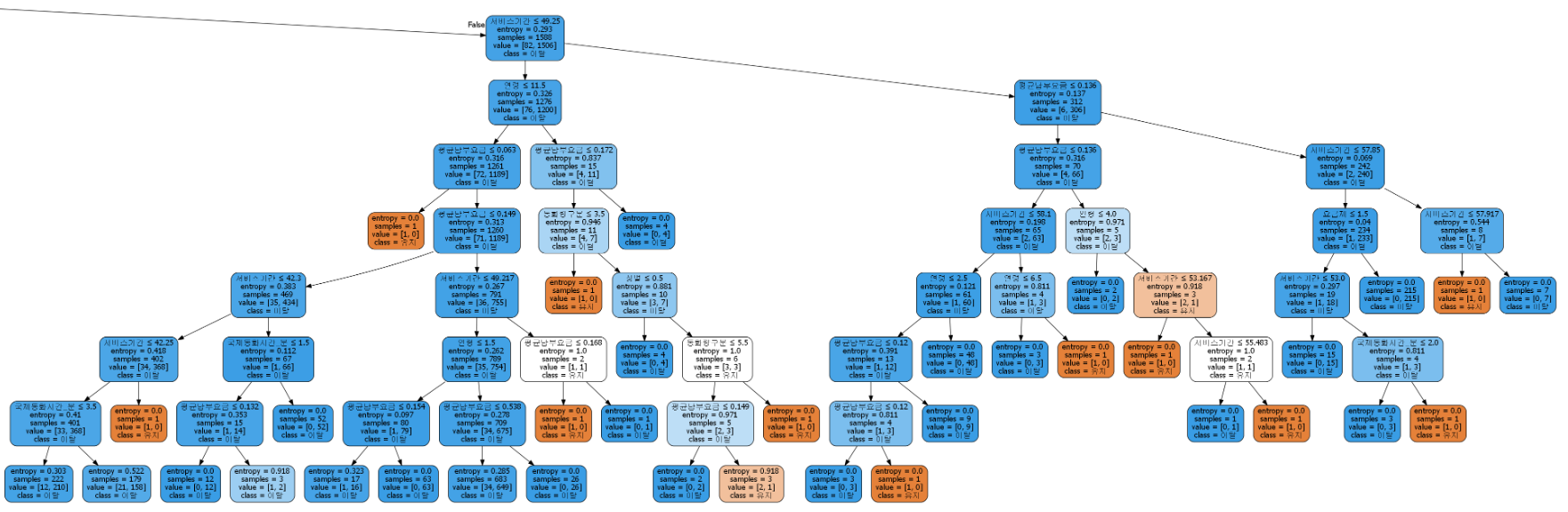
Decision tree의 마디가 지나치게 많을 경우, 즉 과적합(overfitting)이 된 트리는 test data의 예측 정확도를 오히려 떨어트릴 수 있는데 이를 방지하기 위해 max\_depth를 통해 트리의 깊이를 조절했다.

최적의 모델은 criterion = entropy, max\_depth = 8을 부여한 모델이다. 이 결과, Test Accuracy가 0.76이 나왔다. 그리고 precision은 0.77, recall은 0.75의 결과값이 나왔다.

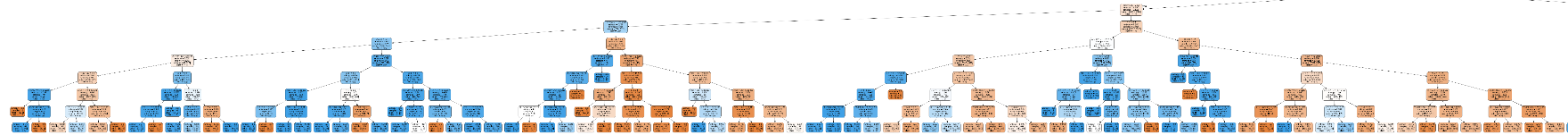


이 모델의 Decision Tree이다. Depth가 커서 잘 보이지 않기 때문에 파일을 따로 첨부했다.

Decision Tree의 root node는 information gain이 가장 큰 통화품질불만이다. 통화품질불만이 0.5이하이면(F이면) 왼쪽 자식 node로 이동하고, 초과이면 오른쪽 자식 node로 이동한다.



Root node에서 오른쪽 자식 node로 이동했을 때부터 leaf node까지 보여지는 tree의 부분이다. leaf node를 보면 class는 대부분 이탈이다.



Root node에서 왼쪽 자식 node로 이동했을 때부터 leaf node까지 보여지는 tree의 부분이다. 통화품질불만이 F라면 T보다 더 많은 속성들을 고려해야 예측을 할 수 있다는 것을 보여준다.

1. **속성의 중요도와 이탈하는 고객의 특성**

스크린샷이(가) 표시된 사진

자동 생성된 설명

**앉아있는, 검은색, 쥐고있는, 테이블이(가) 표시된 사진

자동 생성된 설명**

Decision Tree는 data의 속성을 선택해서 분기해 나아간다. 이 때 어떤 속성을 선택할지에 따라 Tree의 성능에 영향을 미치는데 여기서 어떤 속성을 선택할지의 기준이 되는 것이 information gain이다. information gain 은 부모 노드의 불순도(impurity) - 자식 노드의 불순도(impurity)로 계산할 수 있으며 impurity decrease라고도 한다. 이 information gain이 클수록 정보 이득이 큰 것이다. sklearn은 tree에서 feature\_importances을 제공하는데 이는 impurity decrease가 노드에 도달할 확률을 가중치로 하여 계산하며 아래와 같은 연산을 통해 0~1 사이 값으로 정규화 한다. 이 값이 클수록 중요한 속성이며 이를 기준으로 Decision tree를 그린다.

이탈여부를 결정하는 데에 있어 중요한 속성은 **‘통화품질불만’, ‘국제통화시간\_분’, ‘연령’, ‘통화량구분’, ‘납부여부’**, … 순으로 나왔다. 그 중 상위 5개 속성으로 이탈하는 고객의 특성을 알기 위해 시각화를 진행했다. (같은 내용의 코드는 생략했다.)

sns.catplot(y="이탈여부", x="연령", data=telco\_data, kind="bar", palette="Pastel1")  
plt.show()

스크린샷이(가) 표시된 사진

자동 생성된 설명‘통화품질불만’과 ‘이탈여부’의 관계에 대한 그래프이다. 통화품질불만 1은 T, 2는 F이다. 통화품질에 불만을 갖고 있는 고객은 불만을 갖고 있지 않은 고객들보다 이탈할 확률이 높다. 그렇다고 불만을 갖고 있는 고객들이 모두 이탈을 하는 것은 아니다.

스크린샷이(가) 표시된 사진

자동 생성된 설명‘국제통화시간\_분’과 ‘이탈여부’의 관계에 대한 그래프이다. ‘국제통화시간\_분’은 0과 327 사이의 값을 갖고 있으며 50분 단위로 스케일링을 해주었다. 1은 50분 이하, 2는 50분 초과, 100분 이하, 7은 300분 이상을 나타낸다. 국제통화를 100~200분 사이로 하는 고객은 다른 고객들보다 이탈할 확률이 높다.

개체이(가) 표시된 사진

자동 생성된 설명

‘연령’과 ‘이탈여부’의 관계에 대한 그래프이다. ‘연령’은 12와 80 사이의 값을 갖고 있으며 5살 단위로 스케일링을 해주었다. 1은 10~15살, 2는 16~20살, 14는 76~80살이다. 10~25살의 고객은 이탈할 확률이 다른 고객들보다 높다. 이와 반대로 65~80살의 고객은 비교적 이탈할 확률이 낮게 나왔다.

스크린샷이(가) 표시된 사진

자동 생성된 설명‘통화량구분’과 ‘이탈여부’의 관계에 대한 그래프이다. 1은 무, 2는 저, 3은 중저, 4는 중, 5는 중고, 6은 고의 값을 가진다. 통화량이 적은 편인 무~중저에 속하는 고객들은 통화량이 중간이거나 많은 고객들에 비해 이탈할 확률이 높게 나왔다.

스크린샷이(가) 표시된 사진

자동 생성된 설명‘납부여부’와 ‘이탈여부’의 관계에 대한 그래프이다. 1은 High CAT 50, 2는 High CAT 100, 3은 High PLAY 100, 4는 OK의 값을 가진다. 납부를 하지 않은 고객들은 이탈할 확률이 높다.

# 인용 자료

<https://sumniya.tistory.com/26> - 분류성능평가지표 - Precision(정밀도), Recall(재현율) and Accuracy(정확도)

<https://ko.wikipedia.org/wiki/%EA%B2%B0%EC%A0%95_%ED%8A%B8%EB%A6%AC_%ED%95%99%EC%8A%B5%EB%B2%95> – 결정 트리 학습법

<https://hoony-gunputer.tistory.com/entry/%ED%95%B8%EC%A6%88%EC%98%A8-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-6%EA%B0%95-%EA%B2%B0%EC%A0%95%ED%8A%B8%EB%A6%AC-Decision-tree> - [핸즈온 머신러닝] 6강 결정트리 ( Decision tree)