Algorigo
- 1차 과제

전주혁


Contents

01. EDA

02. train.csv 분류 모델 구현

03. train.csv와 unlabeled.csv 활용 분류 모델 구현

Intro

1. 주제

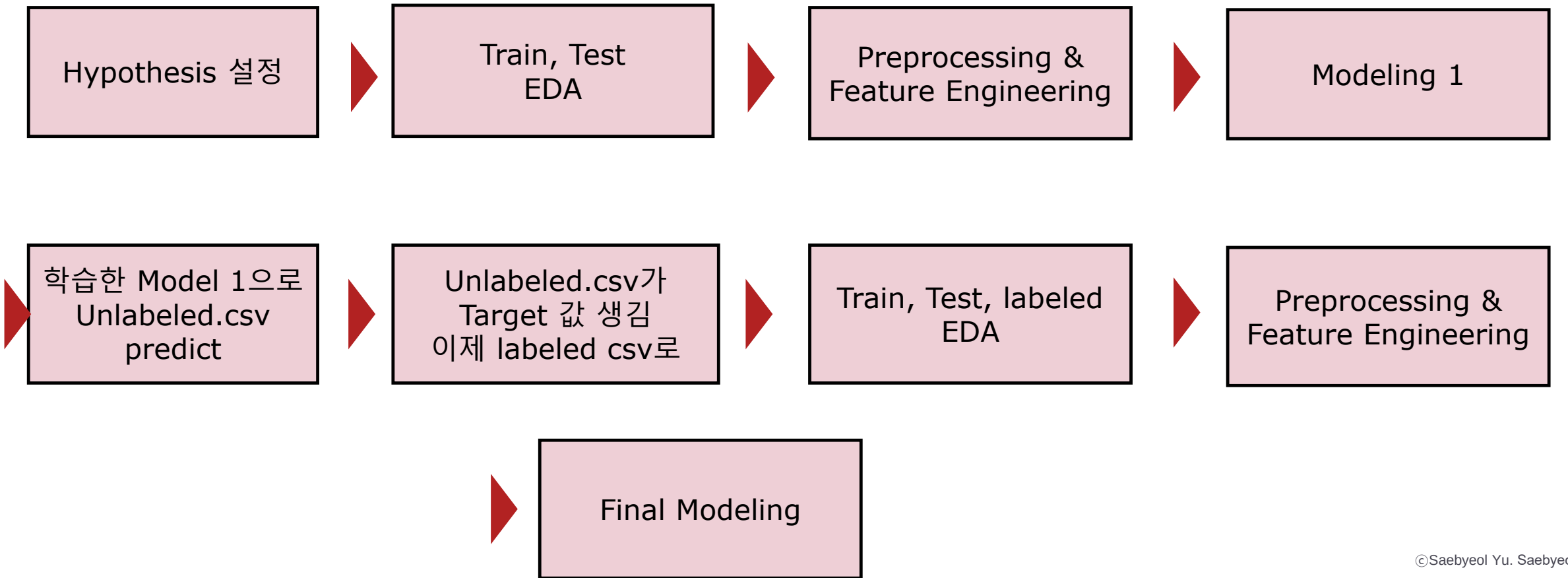
* 승객의 설문 조사 데이터로부터 만족 여부(satisfaction)를 판단

* 제공하는 서비스에 대한 만족의 중요성을 찾는다면 미래에 더 많은 고객 유치 가능

* 승객의 설문 조사 데이터로부터 효율적인 운영이 가능

Intro

2. 전체 Flow



Part 1.

EDA

Part 1,

Hypothesis

1. 좌석별 등급에 따라 만족도가 달라질 것임
-> Eco, Business 좌석의 가격 차이로 인한 제공되는 서비스의 질이 달라지기 때문

2.. Customer Type에 따라 만족도가 달라질 것임
-> Loyal customer는 주로 자주 이용하는 고객층이기 때문에 주로 항공사에 만족한다는 반증으로 생각

3. Food and drink에 따라 만족도가 달라질 것임
-> 음식과 마실것을 많이 줄 수록 다들 만족하기 때문

4. 지연이 덜 될수록 만족할 확률이 높을 것임

5. 다른 피처들의 만족도 값이 높을수록 타겟값(satisfaction)이 만족일 확률이 높을 것임
(반대일 확률도 고려)

Part 1,

EDA(Flow)

1. 먼저 Baseline model을 만든 후 Baseline Score를 만들기

2. EDA를 통해 insight 도출

3. 전처리 or Feature Engineering을 한 뒤 Baseline Score와 비교(검증)

ID와 NULL로만 이루어져 있는 피처 Drop 후 train으로 학습 test로 평가

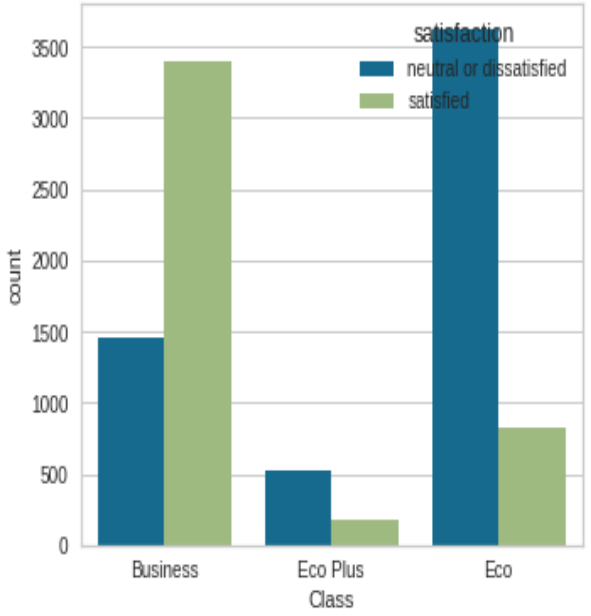
```
Acc Score : 0.938450
```

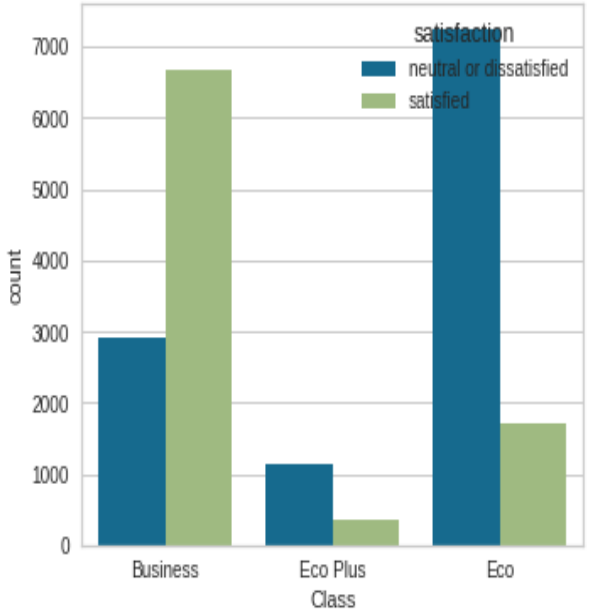
<- Baseline Score

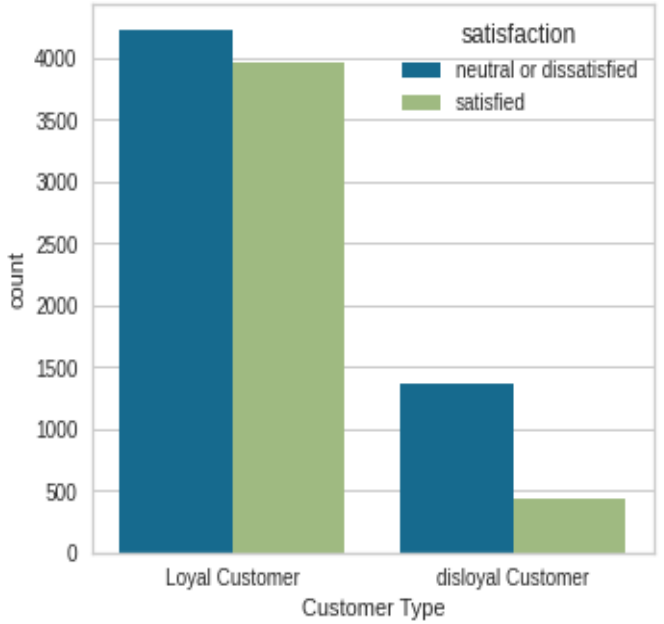
Baseline Model : pycaret(catboost),
10fold

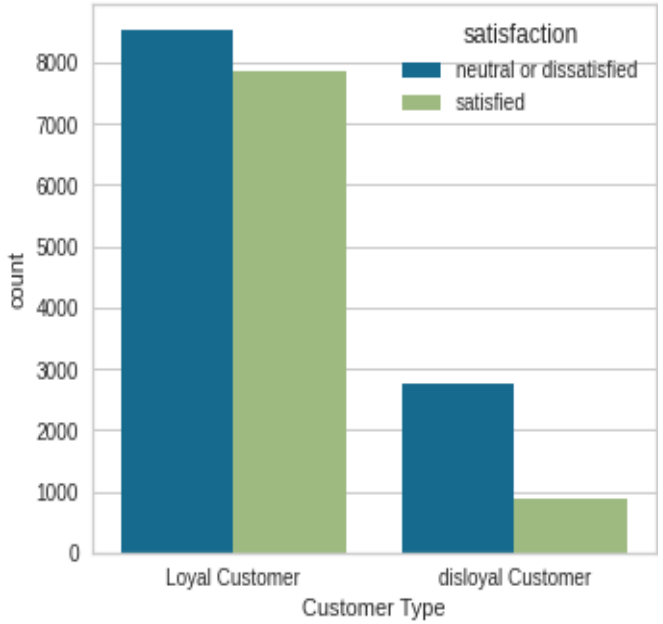
Part 1,

EDA(가설기반)









1. 비즈니스 클래스가 만족 비율이 높지만

Eco 클래스는 불만족이 매우 심함

-> Eco class를 위한 서비스가 부족함을 의미

-> Business는 비싸기 때문에 그만큼 서비스가 좋음

2. Loyal Customer은 불만족 비율이 약간 많지만

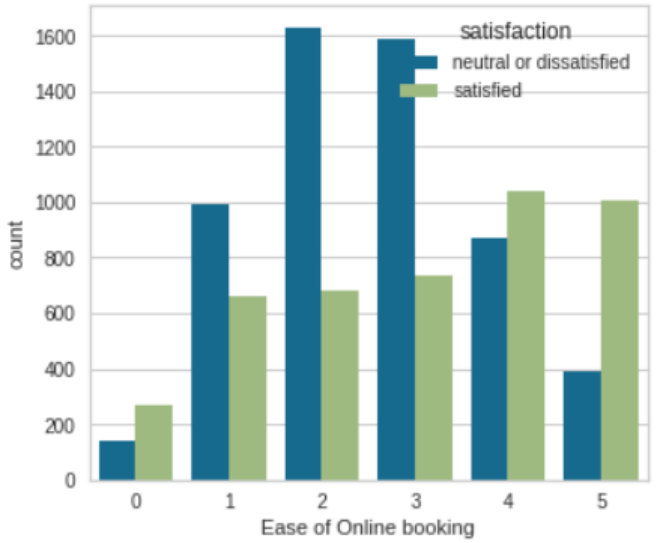
disloyal은 불만족 비율이 매우 높음

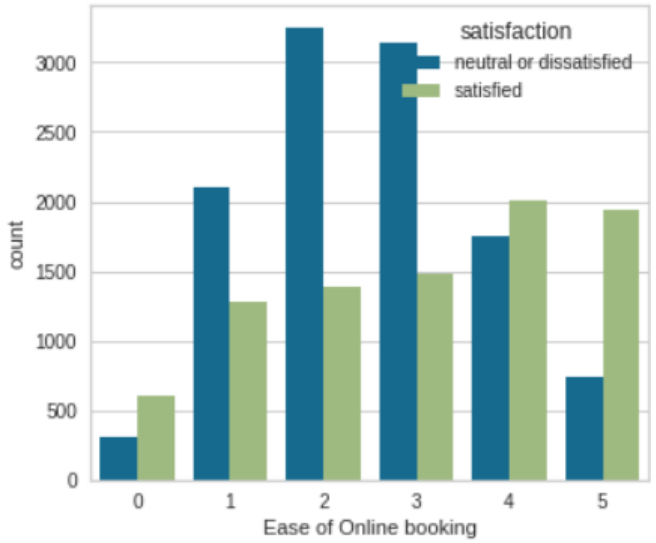
-> disloyal을 위한 서비스가 부족함을 의미

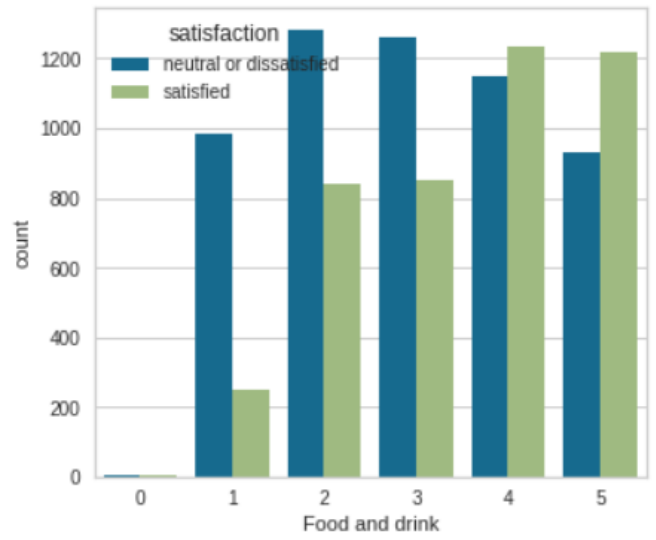
-> Loyal도 만족 비율이 더 높진 않으니 개선책이 필요

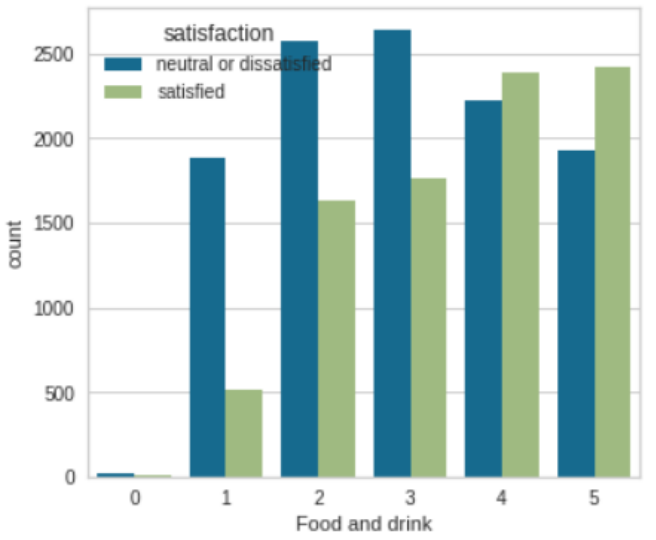
Part 1,

EDA(가설기반)







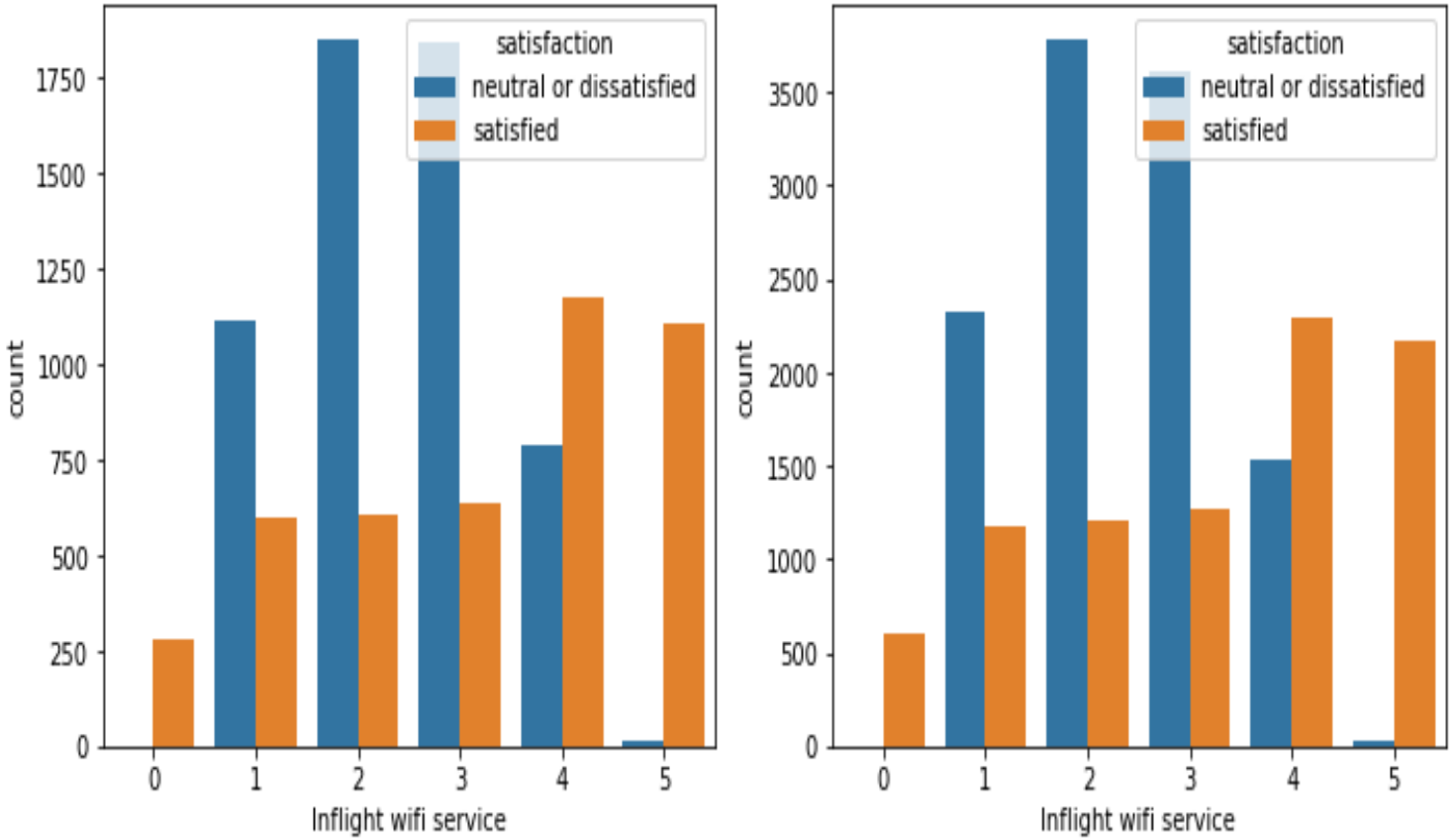


가설대로 보통 만족도의 숫자가 증가할수록

만족도(satisfied)도 증가하는 경향이 존재

Part 1,

EDA



Value가 0일 때는 가설이 맞지 않은 경우 존재

Inflight wifi service 피처의 0의 값은

satisfied만 존재

-> 가장 중요한 피처

EDA

중요 피처 확인(Drop)

1. Gender Drop `Acc Score : 0.938300`
2. Customer Type Drop `Acc Score : 0.928800`
3. Age Drop `Acc Score : 0.936200`
4. Type of Travel Drop `Acc Score : 0.931450`
5. Class Drop `Acc Score : 0.927650`
6. Flight Distance Drop `Acc Score : 0.938150`
7. Inflight wifi service Drop `Acc Score : 0.853300`
8. Departure/Arrival time convenient Drop `Acc Score : 0.937350`
9. Ease of Online booking Drop `Acc Score : 0.938800`
10. Gate location Drop `Acc Score : 0.935200`
11. Food and drink Drop `Acc Score : 0.936250`

insight

EDA 결과와 마찬가지로
Inflight wifi service
피처가
가장 중요

Conclusion

- 대부분 가설과 동일하게 데이터 분포를 띔

- 만족도는 값이 클 수록 만족(satisfied)하는 비율이 올라감

- 만족도에서 0은 missing value로 가정

-> 수기로 하는 설문조사의 경우 값을 기입하지 않거나, 복수로 잘못 기입 하는 경우가 존재할 것임

Part 2,

Train 분류 모델 구현

Feature Engineering

Missing Value

```
train['Ease of Online booking'][train['Ease of Online booking'] == 0] = 5
test['Ease of Online booking'][test['Ease of Online booking'] == 0] = 5
```

다양한 Feature Engineering 기법을 시도해보았지만

위의 코드 외의 다른 피처들을 통해 파생변수들을 만들었지

만

성능 향상이 보이지 않아

위 코드만 사용하였습니다.

Modeling

pycaret

모델 별 평가지표를 쉽게 보기 위해 pycaret 사용

pycaret 이란?

- Machine Learning Workflow를 자동화하는 오픈소스 라이브러리

- Classification, Regression, Clustering 등의 Task에서 사용하는 여러 모델들을 동일한 환경에서 실행을 자동화한 라이브러리

- 여러 모델을 비교 가능

Part 2,

Validation

계층별 K-fold 교차 검증 활용

(회귀) 일반적 방법론 – KFOLD 교차 검증

CV1	Test data	Train data	Train data	Train data	Train data
CV2	Train data	Test data	Train data	Train data	Train data
CV3	Train data	Train data	Test data	Train data	Train data
CV4	Train data	Train data	Train data	Test data	Train data
CV5	Train data	Train data	Train data	Train data	Test data

장점

- 특정 데이터셋에 과적합 방지
- 일반화된 모델 생성
- 과소적합 방지(데이터셋 규모가 작을경우)

단점

- 데이터 클래스가 불균형한 경우 학습 데이터가 고루 분할되지 못함

Validation

계층별 K-fold 교차 검증 활용

활용 방법론 – 계층별 KFOLD 교차 검증

-> 데이터 Target 피처의 특성을 반영하기 위해
원본 데이터의 분포를 반영하는 계층별 KFOLD 교차 검증 활용

Part 2,

Modeling

Model Metrics

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
catboost	CatBoost Classifier	0.9391	0.9818	0.9374	0.9255	0.9313	0.8767	0.8769	3.395
lightgbm	Light Gradient Boosting Machine	0.9344	0.9807	0.9284	0.9234	0.9258	0.8670	0.8672	0.197
rf	Random Forest Classifier	0.9337	0.9767	0.9245	0.9252	0.9248	0.8655	0.8656	1.020
xgboost	Extreme Gradient Boosting	0.9300	0.9784	0.9245	0.9175	0.9209	0.8581	0.8583	1.051
gbc	Gradient Boosting Classifier	0.9204	0.9748	0.9300	0.8938	0.9115	0.8392	0.8399	0.950
et	Extra Trees Classifier	0.9204	0.9705	0.9105	0.9092	0.9098	0.8386	0.8387	0.582
lr	Logistic Regression	0.8916	0.9462	0.9089	0.8544	0.8808	0.7815	0.7828	0.498
ada	Ada Boost Classifier	0.8871	0.9459	0.8921	0.8578	0.8745	0.7720	0.7726	0.357
ridge	Ridge Classifier	0.8863	0.0000	0.9164	0.8402	0.8766	0.7715	0.7741	0.038
lda	Linear Discriminant Analysis	0.8863	0.9424	0.9164	0.8402	0.8766	0.7715	0.7741	0.052
dt	Decision Tree Classifier	0.8836	0.8820	0.8687	0.8675	0.8680	0.7638	0.7640	0.038
nb	Naive Bayes	0.7934	0.9175	0.6363	0.8589	0.7307	0.5691	0.5862	0.022
knn	K Neighbors Classifier	0.6297	0.6558	0.5478	0.5856	0.5659	0.2437	0.2442	0.087
svm	SVM - Linear Kernel	0.6139	0.0000	0.6350	0.5749	0.5490	0.2337	0.2791	0.116
dummy	Dummy Classifier	0.5592	0.5000	0.0000	0.0000	0.0000	0.0000	0.0000	0.021
qda	Quadratic Discriminant Analysis	0.5171	0.5394	0.7286	0.4675	0.5576	0.0734	0.0928	0.049

10 fold

Stratified k-fold

각 모델별 Metrics

Modeling

Accuracy

Soft voting, hard voting, stacking 등 여러 앙상블 기법 사용을 하였지만

Catboost 단일 모델이 가장 정확도가 좋게 나옴

EDA와 F.E를 진행한 후 성능 향상

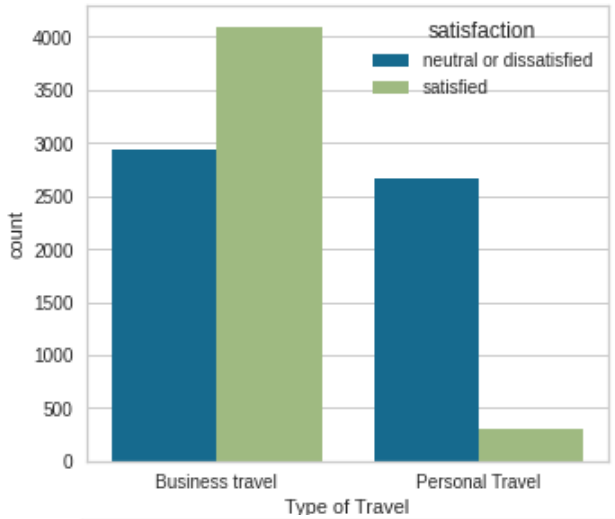
Part 3,

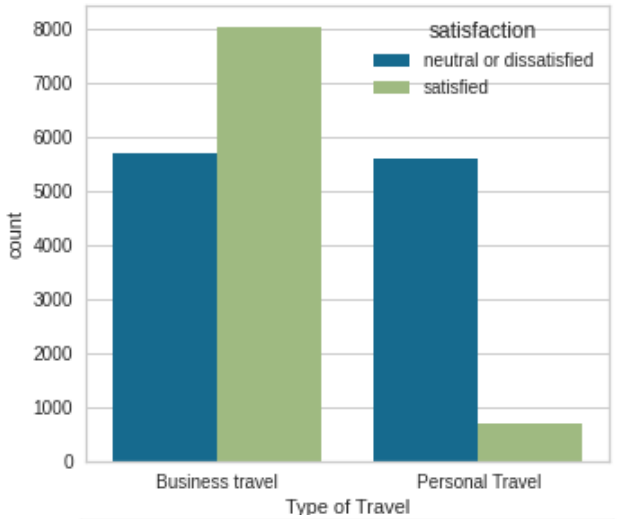
unlabeled

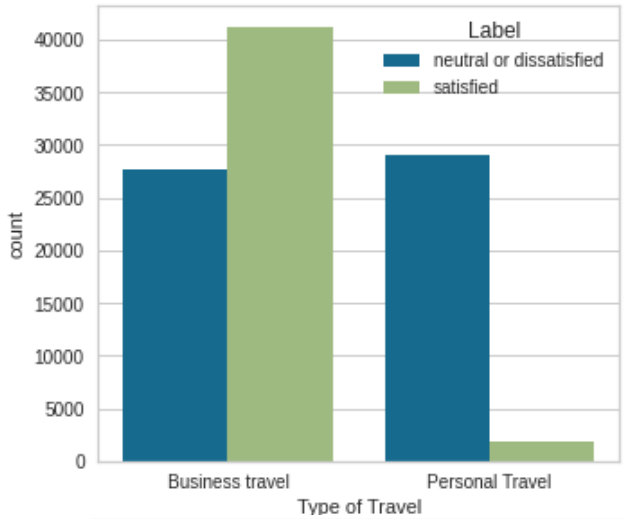
Part 3,

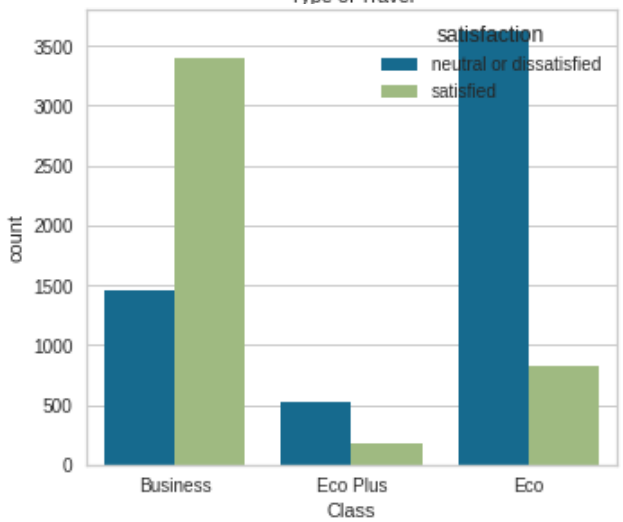
unlabeled

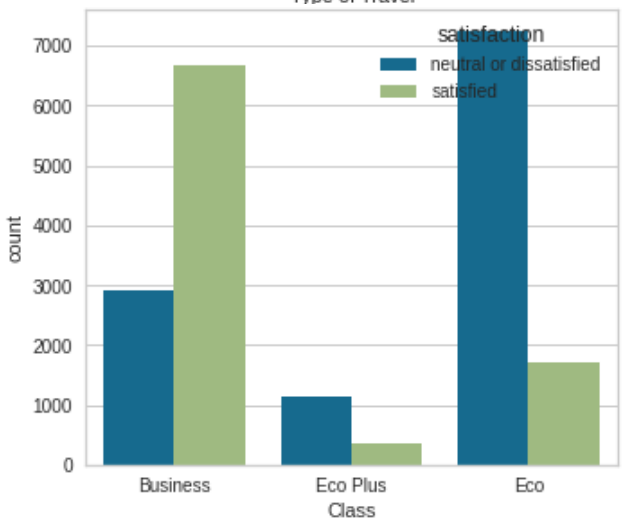
EDA

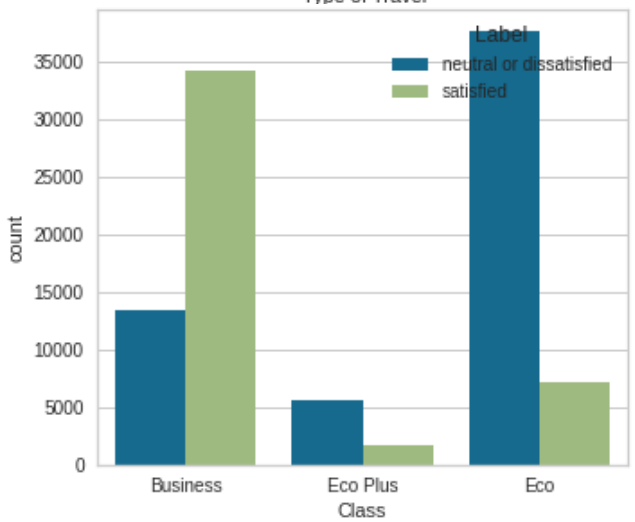












Model1으로

unlabeled.csv를 예측 시

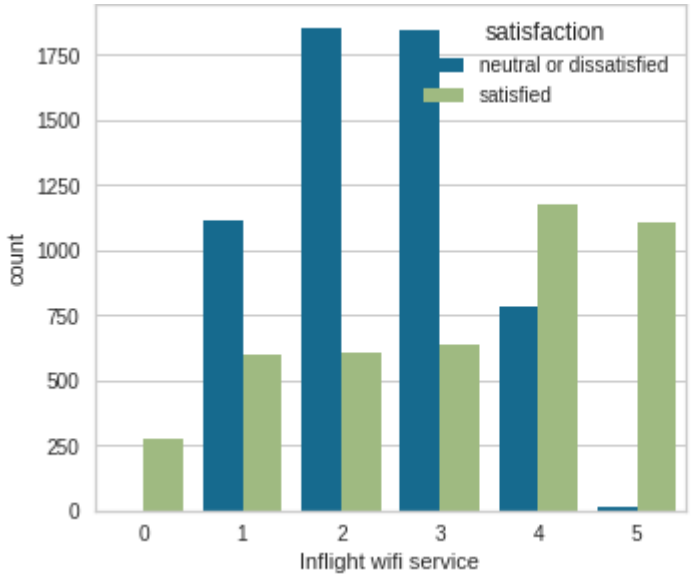
Target값 분포가

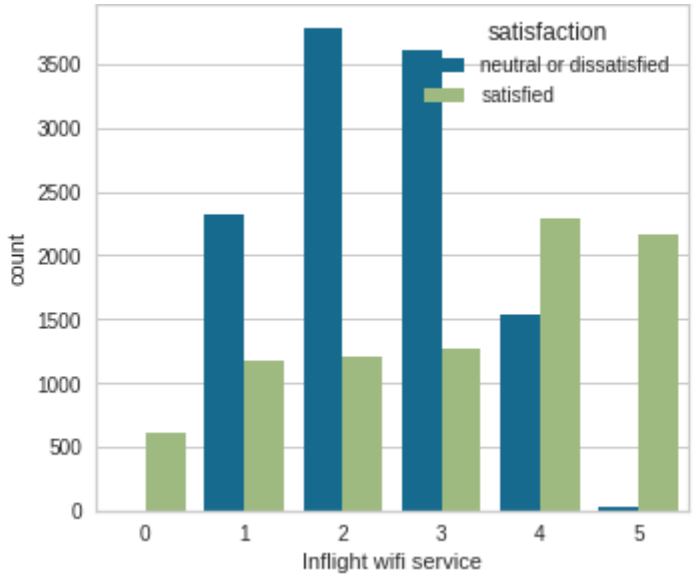
실제 분포와 거의 비슷

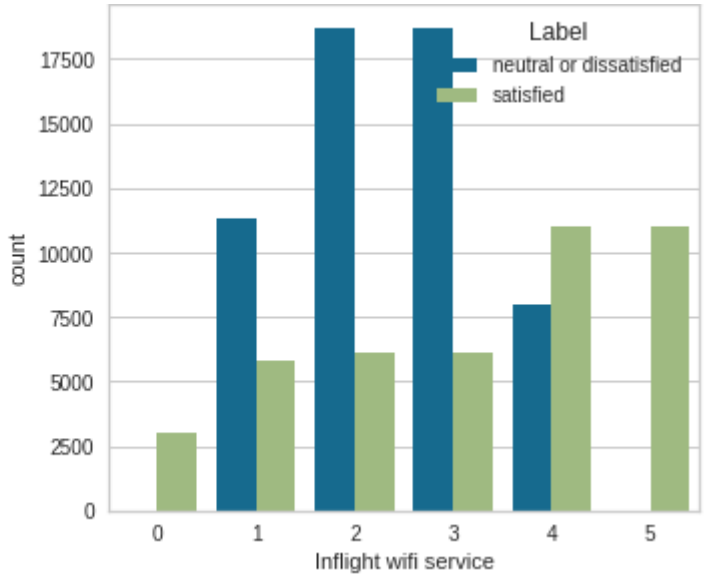
Part 3,

unlabeled

EDA







이 전에서 가장 중요했던 피처 분포도 거의 비슷하게 나온 것으로 보

아

이번 모델에도 가장 중요한 피처로 보임

Part 3,

unlabeled

Unlabeled -> labeled

1 . Part 2의 모델로 Unlabeled 예측

-> 이제 target값이 생김

2. 하지만 이 Target 값은 100%의 정확한 값이 아님

3. unlabeled Target값이 될 확률값을 구간을 나눠서 drop

-> 부정확한 Target 값을 어느정도 filtering 가능

	A	B	C
1	id	Label	Score
2	101480	satisfied	0.9656
3	5952	satisfied	0.9992
4	117026	neutral or	0.8205
5	72662	neutral or	0.8521
6	98343	satisfied	0.9031
7	100140	satisfied	0.8713
8	39699	satisfied	0.9993
9	54375	satisfied	0.6966
10	14026	neutral or	0.9765
11	127499	neutral or	0.9966
12	42154	neutral or	0.9945
13	48012	neutral or	0.9978
14	98131	neutral or	0.9965
15	99072	satisfied	0.9594
16	71792	satisfied	0.9413
17	13996	neutral or	0.823
18	5048	satisfied	0.995
19	113639	neutral or	0.9986
20	44964	satisfied	0.998
21	129183	satisfied	0.7837

Ex)

- 2번 index의 id 5952 는 satisfied일 확률이 99%임 (정확)

- 9번 index의 id54375는 satisfied일 확률이 69%임 (부정확)

Part 3,

unlabeled

Filtering

```
#Acc Score = 0.941950  Filtering 안한 Acc Score

# Acc Score = 0.944650      >  0.825
# Acc Score = 0.943450      >  0.85
# Acc Score = 0.943250      >  0.80
# Acc Score = 0.942650      >  0.88
# Acc Score = 0.942550      >  0.70
# Acc Score = 0.942350      >  0.75
# Acc Score = 0.942300      >  0.65
# Acc Score = 0.939850      >  0.92
```

확률이 0.825 이하인 index들은 Drop하

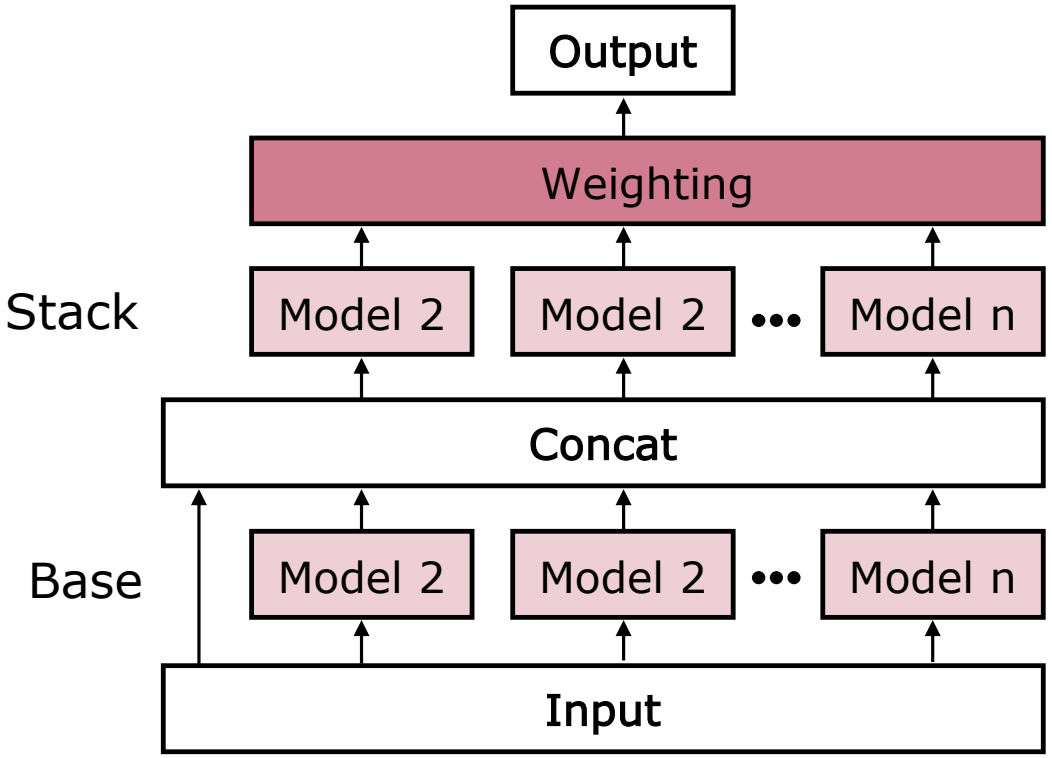
여 Filtering 해줄 때의 정확도가 가장 큼

Part 3,

unlabeled

Modeling

2번째 모델은 AutoGluon-Tabular 사용



- **AutoML** 프레임워크
- 보편적인 AutoML이 주로 사용하는 CASH(Combined Algorithm Selection and Hyper-parameter optimization)이 아닌,
  ensembling과 **stacking** 활용
- 각 모델 학습시 Repeated k-fold ensemble bagging 사용
  -> 과학습 방지

Part 3,

unlabeled

AutoGluion-Tabular

❶ 환경

GPU - NVIDIA GeForce RTX 3060

CPU - AMD Ryzen 9 5950X 16-Core Process or

Python 3.8.6

GPU 학습 시 15~20분 소요

GPU 사양에 따라 정확도의 차이가 발생할 수 있음

❷ 학습결과

★ Acc Score = 0.944650

Modeling1(part2)때의

Acc Score : 0.939150 보다

크게 성능이 향상됨

unlabeled

Feature Importance

	index	importance	...	p99_high	p99_low
0	Inflight wifi service	0.14632	...	0.156033	0.136607
1	Type of Travel	0.09460	...	0.105382	0.083818
2	Customer Type	0.05060	...	0.053958	0.047242
3	Class	0.04892	...	0.052908	0.044932
4	Online boarding	0.00720	...	0.010068	0.004332
5	Gate location	0.00480	...	0.008342	0.001258
6	Seat comfort	0.00388	...	0.006970	0.000790
7	Checkin service	0.00156	...	0.001744	0.001376
8	Cleanliness	0.00144	...	0.003046	-0.000166
9	Food and drink	0.00116	...	0.002375	-0.000055
10	Inflight entertainment	0.00088	...	0.001683	0.000077
11	Departure/Arrival time convenient	0.00084	...	0.001291	0.000389
12	Baggage handling	0.00068	...	0.001232	0.000128
13	Leg room service	0.00052	...	0.001209	-0.000169
14	On-board service	0.00036	...	0.000544	0.000176
15	Flight Distance	0.00028	...	0.000905	-0.000345
16	Age	0.00024	...	0.000585	-0.000105
17	Inflight service	0.00020	...	0.000782	-0.000382
18	Ease of Online booking	0.00008	...	0.000448	-0.000288
19	Departure Delay in Minutes	0.00000	...	0.000000	0.000000
20	Gender	0.00000	...	0.000000	0.000000
21	Arrival Delay in Minutes	0.00000	...	0.000000	0.000000

Feature Importance 결과

Part 4,

outro

outro

1. Model 1과 Model2 에서 Inflight wifi service가 가장 중요한 feature 로 보여짐
 -> 하지만 이는 value값이 0인 이유가 큼

2. 만족도 1~5와 0 구분을 잘 생각

3. Value값이 0인 이유는 두가지가 존재한다고 생각
 가설 1. 단순히 missing value
 가설 2. 예를 들어 wifi인 경우, wifi 서비스가 되지 않아 만족도가 0점임,
와이파이가 잘 될 땐 다수가 핸드폰, 노트북을 사용하기 때문에 쉬고 싶을 때 불빛과 같은 이유로
방해받을 수도 있지만 와이파이가 되지 않기 때문에 옆 사람이 와이파이를 쓸 수 없어
모두가 핸드폰, 노트북의 불빛에 방해받지 않고 쉴 수 있기 때문이라는 가능성을 조금이라도
열어봐야 한다고 생각

outro

4. 항공사는 여행 목적으로 탑승하는 탑승객에 대한 서비스가 필요함

-> 여행 목적으로 탑승하는 탑승객의 만족도가 매우 낮음

-> 목적지의 명소, 맛집, 숨겨진 장소 팜플렛이나 현지 숙박업과의 할인 연계와 같은 서비스를 제공한다면 미래 지향적인 항공사가 될 확률이 높음

5. 항공사는 새로운 고객 유치보다 기존 Loyal Customer들이 떠나지 않게 해야함

-> Loyal Customer의 불만족 비율이 만족보다 높음

-> 한번 떠난 고객은 다시 돌아오기 힘듦

-> 마일리지 이벤트와 같은 기존 고객들이 떠나가지 않게 해야함

“ 감사합니다. ”