2023-1학기 데이터마이닝및분석(001) 과제1

소프트웨어학부 컴퓨터과학전공 2016133 이유진 (Kaggle name: dldbwls00)

<고객이 호텔 예약을 취소할 것인지 예측하는 프로젝트>

✅ To do!

1. Read the training data (train.csv)

```
train = pd.read_csv('/kaggle/input/dataminінghw1/train.csv')
test = pd.read_csv('/kaggle/input/dataminінghw1/test.csv')
```

2. Preprocess the training data

➡️ Categorical attribute 처리 과정

🔹 'booking_status' 타입을 미리 변환: object → integer

```
train = train.replace({'Canceled':1, 'Not_Canceled':0})
train['booking_status'] = train['booking_status'].astype(np.int64)
```

🔹 'room_type_reserved'의 전처리: type1은 1, 나머지는 0으로 변환

```
train = train.replace({'Room_Type 1':1, 'Room_Type 2':0,
        'Room_Type 3':0, 'Room_Type 4':0, 'Room_Type 5':0,
        'Room_Type 6':0, 'Room_Type 7':0})
train['room_type_reserved'] =
train['room_type_reserved'].astype(np.int64)
```

🔹 'market_segment_type'의 전처리:

Offline은 0, Online은 1, 나머지는 2로 변환

```
train = train.replace({'Offline':0, 'Online':1, 'Corporate':2, 'Aviation':2,
        'Complementary':2})
ax = sns.countplot(x = 'market_segment_type', data = train)
```

🔹 'type_of_meal_plan'의 전처리:

Meal Plan 1은 1, 나머지는 0으로 변환

```
train = train.replace({'Not Selected':0, 'Meal Plan 1':1, 'Meal Plan 2':0,
        'Meal Plan 3':0})
train['type_of_meal_plan'] = train['type_of_meal_plan'].astype(np.int64)
ax = sns.countplot(x = 'type_of_meal_plan', data = train)
```

◆ 'no_of_previous_cancellations'의 전처리: 0은 0, 나머지는 1로 변환

```
train = train.replace({0:0, 1:1, 2:1, 3:1, 4:1, 5:1, 6:1, 11:1, 13:1})
train['no_of_previous_cancellations'] =
train['no_of_previous_cancellations'].astype(np.int64)
ax = sns.countplot(x = 'no_of_previous_cancellations', data = train)
```

➡ Attribute들 간 관계 확인

① 'no_of_previous_cancellations'와 'no_of_previous_bookings_not_canceled'는 유사한 정보를 담고 있음

② 'arrival_year', 'arrival_month', 'arrival_date', 'laed_time'은 모두 날짜를 다룸

3. Build any of the following classifiers you want

➡ Decision trees 선택

① 해당 모델 선택 이유 (모델 특성 및 장점 등)

decision tree는 missing value에 취약한 단점이 있지만, train.info()와 test.info()로 두 파일에 missing value가 없음을 확인함.

Ttrain.duplicated().sum으로 중복값이 있는지 확인함.

이상치 / 결측치 여부 확인 및 처리

```
train.info()
test.info()
train.duplicated().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25392 entries, 0 to 25391
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            25392 non-null  object
 1   no_of_adults                          25392 non-null  int64
 2   no_of_children                        25392 non-null  int64
 3   no_of_weekend_nights                  25392 non-null  int64
 4   no_of_week_nights                     25392 non-null  int64
 5   type_of_meal_plan                     25392 non-null  object
 6   required_car_parking_space            25392 non-null  int64
 7   room_type_reserved                    25392 non-null  object
 8   lead_time                             25392 non-null  int64
 9   arrival_year                          25392 non-null  int64
 10  arrival_month                         25392 non-null  int64
 11  arrival_date                          25392 non-null  int64
 12  market_segment_type                   25392 non-null  object
 13  repeated_guest                        25392 non-null  int64
 14  no_of_previous_cancellations          25392 non-null  int64
 15  no_of_previous_bookings_not_canceled  25392 non-null  int64
 16  avg_price_per_room                    25392 non-null  float64
 17  no_of_special_requests                25392 non-null  int64
 18  booking_status                        25392 non-null  object
dtypes: float64(1), int64(13), object(5)
memory usage: 3.7+ MB
```

train.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10883 entries, 0 to 10882
Data columns (total 18 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            10883 non-null  object
 1   no_of_adults                          10883 non-null  int64
 2   no_of_children                        10883 non-null  int64
 3   no_of_weekend_nights                  10883 non-null  int64
 4   no_of_week_nights                     10883 non-null  int64
 5   type_of_meal_plan                     10883 non-null  object
 6   required_car_parking_space            10883 non-null  int64
 7   room_type_reserved                    10883 non-null  object
 8   lead_time                             10883 non-null  int64
 9   arrival_year                          10883 non-null  int64
 10  arrival_month                         10883 non-null  int64
 11  arrival_date                          10883 non-null  int64
 12  market_segment_type                   10883 non-null  object
 13  repeated_guest                        10883 non-null  int64
 14  no_of_previous_cancellations          10883 non-null  int64
 15  no_of_previous_bookings_not_canceled  10883 non-null  int64
 16  avg_price_per_room                    10883 non-null  float64
 17  no_of_special_requests                10883 non-null  int64
dtypes: float64(1), int64(13), object(4)
memory usage: 1.5+ MB
```

test.info()

```
train.duplicated().sum()
```

```
[13]: 0
```

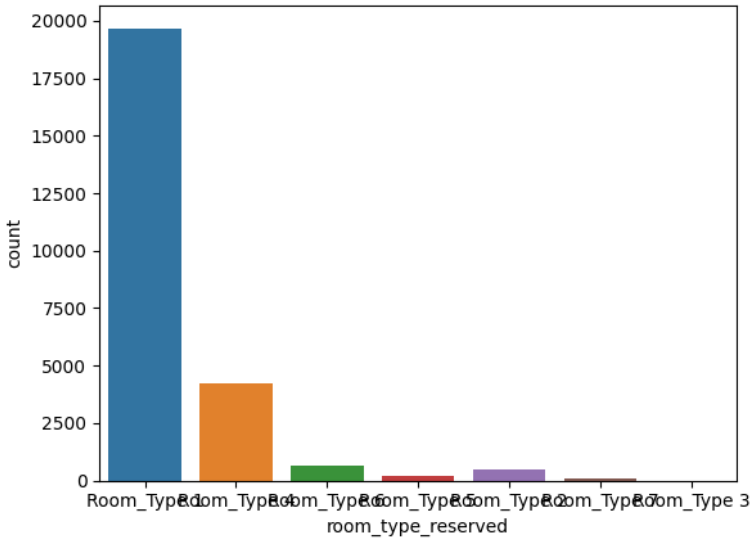
② categorical value와 numeric value 모두 preprocessing 없이 사용할 수 있다는 장점이 있어 코드 구현이 편리함

③ 데이터 분포 확인

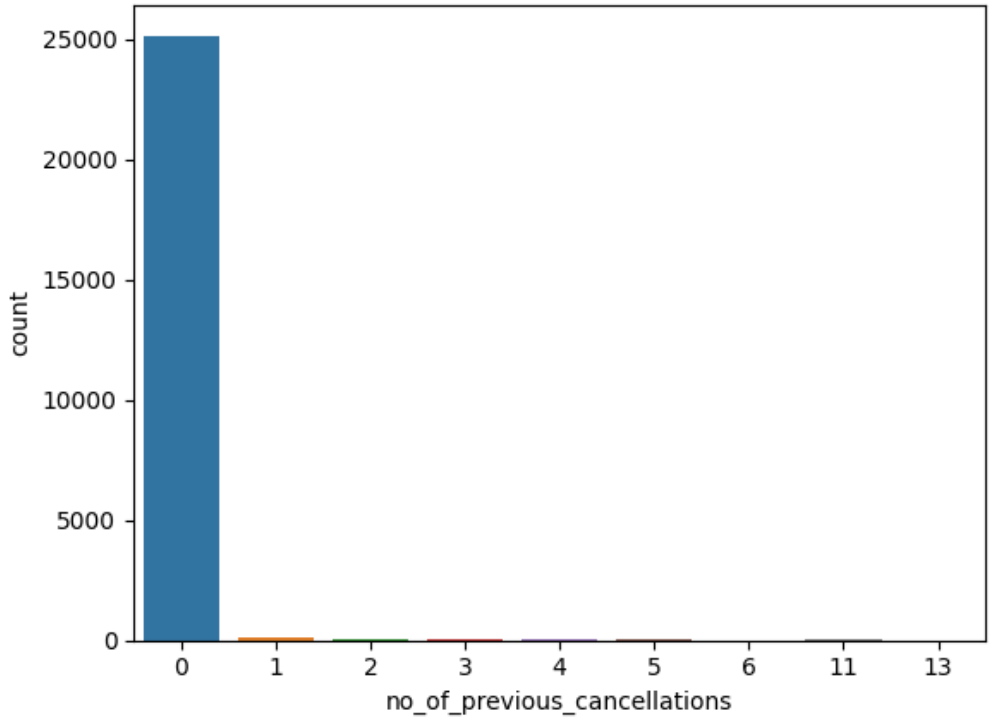
두 파일의 일부 column이 skewed된 부분이 있다고 판단했는데, decision tree classifier는 skewed된 데이터에 영향을 받지 않음

ex) room_type_reserved의 데이터 분포

```
## 분포를 plot으로 볼 수도 있음
ax = sns.countplot(x = 'room_type_reserved', data = train)
```



ex) 'no_of_previous_cancellations'의 데이터 분포



➡ 모델에 사용할 attribute 선택 : 모델에 사용할 attribute들을 선정하게 된 근거

① 'repeated_guest': 재방문율이 높은 예약자가의 취소 확률이 낮을 것이라는 판단

② "no_of_previous_cancellations": 이전의 취소 경험이 예약자의 성향을 나타낸다고 판단

③ 'room_type_reserved', 'type_of_meal_plan', 'required_car_parking_space', 'no_of_special_requests' 등 : 예약자의 demand와 직결된 특성

④ 'lead_time' : 예약일~도착일의 일 수와 취소율의 관계

➡ 모델의 hyperparameter 선택 : 해당 모델의 hyperparameter 선정 과정과 근거

트리를 잘 가를 것이라고 생각되는 attribute들을 골라 hyperparameter 선정

: 총 8개의 attribute(hyperparameter) 사용

4. Read the test data (test.csv) and make predictions

```
prediction_list= dt_model.predict(x_test)

submit = pd.read_csv("/kaggle/input/dataminninghw1/sample_submission.csv")
submit["booking_status"] = prediction_list
submit
```

5. Output a prediction file for the test data

➡ 2016133_summit.csv

```
submit.to_csv("2016133_submit.csv",index=False)
```