Contents

1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈린
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

1 라이브러리 호출 ¶

In [1]:

```
import numpy as np # Numpy
import pandas as pd # Pandas
import matplotlib as mpl #Matplotlib 세팅
import matplotlib.pyplot as plt # 시각화 .
import seaborn as sns # 시각화 도구
from sklearn.preprocessing import Standa
from sklearn.model_selection import trai
from sklearn.model_selection import KFol
from sklearn.cluster import KMeans # 클러
from sklearn.metrics import silhouette_s
import xgboost as xgb # XGBoost
from sklearn.model_selection import GridS
from sklearn.metrics import accuracy_sco
from sklearn.metrics import recall_score
from imblearn.combine import SMOTEENN, SM
from hyperopt import hp, fmin, tpe, Tria
from nltk.corpus import names # nltk
import nltk
nltk.download("names")
from nltk import NaiveBayesClassifier
from scipy import stats
from collections import Counter
import random

import warnings # 경고문 제거용


%matplotlib inline
%config Inlinebackend.figure_format = 're

# 한글 폰트 설정
mpl.rc('font', family='D2Coding')
# 유니코드에서 음수 부호 설정
mpl.rc('axes', unicode_minus = False)

warnings.filterwarnings('ignore')
sns.set(font="D2Coding", rc={"axes.unico
plt.rc('figure', figsize=(10,8))
```

executed in 5.27s, finished 11:39:14 2022-11-25

```
[nltk_data] Downloading package names to
[nltk_data]     C:\Users\admin\AppData\Roaming
[nltk_data]   Package names is already up-to-d
```

2 데이터로딩

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분스
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈런
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [2]:

```
data = pd.read_excel('train_test_na_fille
data.info()
```

executed in 2.46s, finished 11:39:17 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8693 entries, 0 to 8692
Data columns (total 18 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   PassengerId   8693 non-null   object
 1   HomePlanet    8693 non-null   object
 2   CryoSleep     8693 non-null   bool
 3   Cabin1        8590 non-null   object
 4   Cabin2        8590 non-null   float64
 5   Combi         8590 non-null   object
 6   Cabin3        8590 non-null   object
 7   Cabin         8590 non-null   object
 8   Destination   8693 non-null   object
 9   Age           8693 non-null   int64
 10  VIP           8693 non-null   bool
 11  RoomService   8693 non-null   int64
 12  FoodCourt     8693 non-null   int64
 13  ShoppingMall  8693 non-null   int64
 14  Spa           8693 non-null   int64
 15  VRDeck        8693 non-null   int64
 16  Name          8493 non-null   object
 17  Transported   8693 non-null   bool
dtypes: bool(3), float64(1), int64(6), object(
memory usage: 1.0+ MB
```

In [3]:

```
test = pd.read_excel('train_test_na_fille
test.head()
```

executed in 1.36s, finished 11:39:18 2022-11-25

Out[3]:

	PassengerId	HomePlanet	CryoSleep	Cabin1	Ca
0	0013_01	Earth	True	G	
1	0018_01	Earth	False	F	
2	0019_01	Europa	True	C	
3	0021_01	Europa	False	C	
4	0023_01	Earth	False	F	

3 탐색

3.1 ANOVA 분석

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분스
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [4]:

```
numeric_data = [column for column in data

for column in numeric_data:
  df_anova = data[[column,'Transported']]
  grouped_anova = df_anova.groupby(['Tran
  f_value, p_value = stats.f_oneway(grou
                                     grou
  result = ""
  if p_value < 0.05:
    result = "{}은/는 예측에 중요한 feature
  else:
    result = "{}은/는 예측에 중요하지않은 fe
  print(result)
```

executed in 43ms, finished 11:39:18 2022-11-25

```
Cabin2은/는 예측에 중요하지않은 feature입니다.
Age은/는 예측에 중요한 feature입니다.
RoomService은/는 예측에 중요한 feature입니다.
FoodCourt은/는 예측에 중요한 feature입니다.
ShoppingMall은/는 예측에 중요하지않은 feature입니ㄷ
Spa은/는 예측에 중요한 feature입니다.
VRDeck은/는 예측에 중요한 feature입니다.
```

In [5]:

```
def outlier_detection_train(df, n, column
    rows = []
    will_drop_train = []
    for col in columns:
        Q1 = np.nanpercentile(df[col], 2
        Q3 = np.nanpercentile(df[col], 7
        IQR = Q3 - Q1
        outlier_point = 1.5 * IQR
        rows.extend(df[(df[col] < Q1 - ou
    for r, c in Counter(rows).items():
        if c >= n: will_drop_train.append
    return will_drop_train
```

executed in 14ms, finished 11:39:18 2022-11-25

In [6]:

```
data.drop('Cabin2', inplace=True, axis=1)
```

executed in 14ms, finished 11:39:18 2022-11-25

In [7]:

```
test.drop('Cabin2', inplace=True, axis=1)
```

executed in 14ms, finished 11:39:18 2022-11-25

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분스
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여부
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 쨰려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [8]:

```
data.columns
```

executed in 14ms, finished 11:39:18 2022-11-25

Out[8]:

```
Index(['PassengerId', 'HomePlanet', 'CryoSleep
abin3',
       'Cabin', 'Destination', 'Age', 'VIP', '
t',
       'ShoppingMall', 'Spa', 'VRDeck', 'Name'
      dtype='object')
```

4 또처리~

In [ ]:

```

```

executed in 29ms, finished 14:28:49 2022-11-23

4.1 이상치 확인 및 제거

In [9]:

```
def outlier_detection_train(df, n, columi
    rows = []
    will_drop_train = []
    for col in columns:
        Q1 = np.nanpercentile(data[col],
        Q3 = np.nanpercentile(data[col],
        IQR = Q3 - Q1
        outlier_point = 1.5 * IQR
        rows.extend(df[(df[col] < Q1 - ou
    for r, c in Counter(rows).items():
        if c >= n: will_drop_train.appen
    return will_drop_train
```

executed in 13ms, finished 11:39:18 2022-11-25

Contents

1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [10]:

```
will_drop_train = outlier_detection_trai
will_drop_train
```

executed in 25ms, finished 11:39:18 2022-11-25

Out[10]:

```
[338,
 1390,
 6469,
 7038,
 1936,
 3317,
 3980,
 4762,
 6509,
 7007,
 7065,
 7294,
 7689,
 7957,
 8064]
```

In [11]:

```
data.drop(will_drop_train, inplace = Tru
```

executed in 13ms, finished 11:39:18 2022-11-25

4.2 새로운 feature 생성

4.2.1 총 사용금액, 그리고 사용한 금액에 따라 p

In [12]:

```
data["Total"] = data["RoomService"] + dat
data["VRDeck"]
data["RichPoor"] = data["Total"].apply(l
                                       e

test["Total"] = test["RoomService"] + tes
test["VRDeck"]
test["RichPoor"] = test["Total"].apply(l
                                       e
```

executed in 14ms, finished 11:39:18 2022-11-25

4.2.2 그룹여행객 여부

Contents

1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈
4.2.3.2 테
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [13]:

```
data["GroupId"] = data["PassengerId"].app
test["GroupId"] = test["PassengerId"].app
data["GroupNo"] = data["PassengerId"].app
test["GroupNo"] = test["PassengerId"].app

train_g = data[data["GroupId"].duplicate
test_g = test[test["GroupId"].duplicated(
data["Group"] = data["GroupId"].apply(lam
test["Group"] = test["GroupId"].apply(lam
```

executed in 583ms, finished 11:39:19 2022-11-25

In [14]:

```
data.info()
```

executed in 30ms, finished 11:39:19 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8678 entries, 0 to 8692
Data columns (total 22 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   PassengerId   8678 non-null   object
 1   HomePlanet    8678 non-null   object
 2   CryoSleep     8678 non-null   bool  
 3   Cabin1        8575 non-null   object
 4   Combi         8575 non-null   object
 5   Cabin3        8575 non-null   object
 6   Cabin         8575 non-null   object
 7   Destination   8678 non-null   object
 8   Age           8678 non-null   int64 
 9   VIP           8678 non-null   bool  
 10  RoomService   8678 non-null   int64 
 11  FoodCourt     8678 non-null   int64 
 12  ShoppingMall  8678 non-null   int64 
 13  Spa           8678 non-null   int64 
 14  VRDeck        8678 non-null   int64 
 15  Name          8478 non-null   object
 16  Transported   8678 non-null   bool  
 17  Total         8678 non-null   int64 
 18  RichPoor      8678 non-null   object
 19  GroupId       8678 non-null   object
 20  GroupNo       8678 non-null   object
 21  Group         8678 non-null   bool  
dtypes: bool(4), int64(7), object(11)
memory usage: 1.3+ MB
```

4.2.3 나이브 베어스를 활용한 이름을 통한 성별

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [15]:

```
# Train_Data
names_train_data = []
for n in data["Name"]:
    n = str(n)
    a = n.split()
    names_train_data.append(a[0])
```

executed in 14ms, finished 11:39:19 2022-11-25

4.2.3.1 훈련셋

In [16]:

```
# 이름과 성 분리
names_train_data = []
for i in data["Name"]:
    i = str(i)
    a = i.split()
    names_train_data.append(a[0])
```

executed in 14ms, finished 11:39:19 2022-11-25

In [17]:

```
# NLTK의 names 파일을 활용하여 이름을 여성과
labeled_names = [(name, "female") for nar
[(name, "male") for name in names.words('
random.shuffle(labeled_names)
```

executed in 29ms, finished 11:39:19 2022-11-25

In [18]:

```
# 이름의 마지막 단어 가져오는 함수
def gender_features(word):
    return {'last_letter': word[-1]}
```

executed in 13ms, finished 11:39:19 2022-11-25

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분석
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [19]:

```
names_train_data
```

executed in 29ms, finished 11:39:19 2022-11-25

Out[19]:

```
['Maham',
 'Juanna',
 'Altark',
 'Solam',
 'Willy',
 'Sandie',
 'Billex',
 'Candra',
 'Andona',
 'Erraiam',
 'Altardr',
 'Wezena',
 'Berers',
 'Reney',
 'Elle',
 'Justie',
 'Flats',
 'Carry',
```

In [20]:

```
# 나이브 베어스 모델 학습
featuresets = [(gender_features(n), gend
classifier = NaiveBayesClassifier.train(
```

executed in 89ms, finished 11:39:19 2022-11-25

In [21]:

```
# 성별 feature 생성
names_gender = []
for i in names_train_data:
    names_gender.append(classifier.class

# create new column called 'gender'
data["Gender"] = names_gender
```

executed in 88ms, finished 11:39:19 2022-11-25

In [22]:

```
data.Gender[data.Name.isna()] = 'female'
```

executed in 14ms, finished 11:39:19 2022-11-25

In [23]:

```
data.Gender[data.Name.isna()].unique()
```

executed in 15ms, finished 11:39:19 2022-11-25

Out[23]:

```
array(['female'], dtype=object)
```

4.2.3.2 테스트셋

Contents

1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분석
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [24]:

```
# 이름과 성 분리
names_test_data = []
for i in test["Name"]:
    i = str(i)
    a = i.split()
    names_test_data.append(a[0])
```

executed in 13ms, finished 11:39:19 2022-11-25

In [25]:

```
# NLTK의 names 파일을 활용하여 이름을 여성과
labeled_names = [(name, "female") for nar
[(name, "male") for name in names.words('
random.shuffle(labeled_names)
```

executed in 14ms, finished 11:39:19 2022-11-25

In [26]:

```
# 나이브 베어스 모델 학습
featuresets = [(gender_features(n), gend
classifier = NaiveBayesClassifier.train(
```

executed in 29ms, finished 11:39:19 2022-11-25

In [27]:

```
# 이름의 마지막 단어 가져오는 함수
def gender_features(word):
    return {'last_letter': word[-1]}
```

executed in 14ms, finished 11:39:19 2022-11-25

In [28]:

```
# 성별 feature 생성
names_gender = []
for i in names_test_data:
    names_gender.append(classifier.class
```

executed in 56ms, finished 11:39:19 2022-11-25

In [29]:

```
test["Gender"] = pd.Series(names_gender)
```

executed in 14ms, finished 11:39:19 2022-11-25

In [30]:

```
test.Gender[test.Name.isna()] = 'female'
```

executed in 14ms, finished 11:39:19 2022-11-25

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분석
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [31]:

```
data.Gender[data.Name.isna()].unique()
```

executed in 14ms, finished 11:39:19 2022-11-25

Out[31]:

```
array(['female'], dtype=object)
```

4.3 Cabin 결측값들 제거

In [32]:

```
data.dropna(axis=0, inplace=True)
```

executed in 29ms, finished 11:39:19 2022-11-25

4.4 필요없는 features 제거

In [33]:

```
target = data['Transported']
data = data.drop(["PassengerId", "Name",
                  "GroupNo", "Transporte
test = test.drop(["PassengerId", "Name",
                  "GroupNo", "Transporte

```

executed in 13ms, finished 11:39:19 2022-11-25

4.5 째려보기

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [34]:

```
data.info()
```

executed in 14ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8375 entries, 0 to 8692
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   HomePlanet    8375 non-null   object
 1   CryoSleep     8375 non-null   bool  
 2   Cabin1        8375 non-null   object
 3   Cabin3        8375 non-null   object
 4   Destination   8375 non-null   object
 5   Age           8375 non-null   int64 
 6   VIP           8375 non-null   bool  
 7   RoomService   8375 non-null   int64 
 8   FoodCourt     8375 non-null   int64 
 9   ShoppingMall  8375 non-null   int64 
 10  Spa           8375 non-null   int64 
 11  VRDeck        8375 non-null   int64 
 12  RichPoor      8375 non-null   object
 13  Group         8375 non-null   bool  
 14  Gender        8375 non-null   object
dtypes: bool(3), int64(6), object(6)
memory usage: 875.1+ KB
```

In [35]:

```
test.info()
```

executed in 14ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4277 entries, 0 to 4276
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   HomePlanet    4277 non-null   object
 1   CryoSleep     4277 non-null   bool  
 2   Cabin1        4214 non-null   object
 3   Cabin3        4214 non-null   object
 4   Destination   4277 non-null   object
 5   Age           4277 non-null   int64 
 6   VIP           4277 non-null   bool  
 7   RoomService   4277 non-null   int64 
 8   FoodCourt     4277 non-null   int64 
 9   ShoppingMall  4277 non-null   int64 
 10  Spa           4277 non-null   int64 
 11  VRDeck        4277 non-null   int64 
 12  RichPoor      4277 non-null   object
 13  Group         4277 non-null   bool  
```

4.6 원핫인코딩

4.6.1 boolean 타입 피처들 object로 캐스팅

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분스
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈린
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [36]:

```
bool_data = [column for column in data.se

bool_data
```

executed in 13ms, finished 11:39:20 2022-11-25

Out[36]:

```
['CryoSleep', 'VIP', 'Group']
```

In [37]:

```
data["VIP"] = data["VIP"].replace(to_rep
                                  value :
data["CryoSleep"] = data["CryoSleep"].rep
                                  value :
data["Group"] = data["Group"].replace(to_
                                  value :

test["VIP"] = test["VIP"].replace(to_rep
                                  value :
test["CryoSleep"] = test["CryoSleep"].rep
                                  value :
test["Group"] = test["Group"].replace(to_
                                  value :
```

executed in 29ms, finished 11:39:20 2022-11-25

In [38]:

```
data.info()
```

executed in 14ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8375 entries, 0 to 8692
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   HomePlanet   8375 non-null   object
 1   CryoSleep    8375 non-null   object
 2   Cabin1       8375 non-null   object
 3   Cabin3       8375 non-null   object
 4   Destination  8375 non-null   object
 5   Age          8375 non-null   int64 
 6   VIP          8375 non-null   object
 7   RoomService  8375 non-null   int64 
 8   FoodCourt    8375 non-null   int64 
 9   ShoppingMall 8375 non-null   int64 
 10  Spa          8375 non-null   int64 
 11  VRDeck       8375 non-null   int64 
 12  RichPoor     8375 non-null   object
 13  Group        8375 non-null   object
 14  Gender       8375 non-null   object
dtypes: int64(6), object(9)
memory usage: 1.0+ MB
```

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분석
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [39]:

```
test.info()
```

executed in 14ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4277 entries, 0 to 4276
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   HomePlanet    4277 non-null   object
 1   CryoSleep     4277 non-null   object
 2   Cabin1        4214 non-null   object
 3   Cabin3        4214 non-null   object
 4   Destination   4277 non-null   object
 5   Age           4277 non-null   int64 
 6   VIP           4277 non-null   object
 7   RoomService   4277 non-null   int64 
 8   FoodCourt     4277 non-null   int64 
 9   ShoppingMall  4277 non-null   int64 
 10  Spa           4277 non-null   int64 
 11  VRDeck        4277 non-null   int64 
 12  RichPoor      4277 non-null   object
 13  Group         4277 non-null   object
 14  Gender        4277 non-null   object
dtypes: int64(6), object(9)
memory usage: 501.3+ KB
```

4.6.2 더미화

In [40]:

```
# drop_first 첫번째 범주는 제거하고 더미화
# 다른 범주가 전부 0이면 자동적으로 첫번째 범주
df = pd.get_dummies(data, drop_first = T
t_df = pd.get_dummies(test, drop_first=T
```

executed in 44ms, finished 11:39:20 2022-11-25

Contents
1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈련
4.2.3.2 테스
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [41]:

```
df.info()
```

executed in 14ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 8375 entries, 0 to 8692
Data columns (total 24 columns):
 #   Column                     Non-Null Count
---  ------                     --------------
 0   Age                        8375 non-null
 1   RoomService                8375 non-null
 2   FoodCourt                  8375 non-null
 3   ShoppingMall               8375 non-null
 4   Spa                        8375 non-null
 5   VRDeck                     8375 non-null
 6   HomePlanet_Europa          8375 non-null
 7   HomePlanet_Mars            8375 non-null
 8   CryoSleep_Yes              8375 non-null
 9   Cabin1_B                   8375 non-null
 10  Cabin1_C                   8375 non-null
 11  Cabin1_D                   8375 non-null
 12  Cabin1_E                   8375 non-null
 13  Cabin1_F                   8375 non-null
 14  Cabin1_G                   8375 non-null
 15  Cabin1_T                   8375 non-null
 16  Cabin3_S                   8375 non-null
 17  Destination_PSO J318.5-22  8375 non-null
 18  Destination_TRAPPIST-1e    8375 non-null
 19  VIP_Yes                    8375 non-null
 20  RichPoor_poor              8375 non-null
 21  RichPoor_rich              8375 non-null
 22  Group_Yes                  8375 non-null
 23  Gender_male                8375 non-null
dtypes: int64(6), uint8(18)
memory usage: 605.2 KB
```

Contents

1 라이브러리 호출
2 데이터로딩
3 탐색
3.1 ANOVA 분
4 또처리~
4.1 이상치 확인
4.2 새로운 featu
4.2.1 총 사용
4.2.2 그룹여
4.2.3 나이브
4.2.3.1 훈
4.2.3.2 테
4.3 Cabin 결측
4.4 필요없는 fe
4.5 째려보기
4.6 원핫인코딩
4.6.1 boolean
4.6.2 더미화
4.7 스케일링

In [42]:

```
t_df.info()
```

executed in 29ms, finished 11:39:20 2022-11-25

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4277 entries, 0 to 4276
Data columns (total 24 columns):
 #   Column                     Non-Null Count
---  ------                     --------------
 0   Age                        4277 non-null
 1   RoomService                4277 non-null
 2   FoodCourt                  4277 non-null
 3   ShoppingMall               4277 non-null
 4   Spa                        4277 non-null
 5   VRDeck                     4277 non-null
 6   HomePlanet_Europa          4277 non-null
 7   HomePlanet_Mars            4277 non-null
 8   CryoSleep_Yes              4277 non-null
 9   Cabin1_B                   4277 non-null
 10  Cabin1_C                   4277 non-null
 11  Cabin1_D                   4277 non-null
 12  Cabin1_E                   4277 non-null
 13  Cabin1_F                   4277 non-null
 14  Cabin1_G                   4277 non-null
 15  Cabin1_T                   4277 non-null
 16  Cabin3_S                   4277 non-null
 17  Destination_PSO J318.5-22  4277 non-null
 18  Destination_TRAPPIST-1e    4277 non-null
 19  VIP_Yes                    4277 non-null
 20  RichPoor_poor              4277 non-null
 21  RichPoor_rich              4277 non-null
 22  Group_Yes                  4277 non-null
 23  Gender_male                4277 non-null
dtypes: int64(6), uint8(18)
memory usage: 275.8 KB
```

4.7 스케일링

In [43]:

```
scaler = StandardScaler()
scaler.fit(df)
df = scaler.transform(df)
t_df = scaler.transform(t_df)
```

executed in 29ms, finished 11:39:20 2022-11-25