통계 모델링 및 컨설팅 보고서

Predict US Airline Passenger Satisfaction

3 조

2012110506 고호영

2014110482 장현석

목차

Ⅰ. 서론

1. 데이터셋

1) 데이터셋 설명

데이터셋 출처 : HTTPS://WWW.KAGGLE.COM/JOHNDDDDDD/CUSTOMER-SATISFACTION

Survey 문항에 대한 응답을 보고 고객의 만족도를 예측하는 모형을 만들고 싶어서 Survey 에 관련된 DataSet 을 찾아보았다. 그러던 중 kaggle 웹사이트에서 "US Airline Passenger Satisfaction"이라는 Survey 자료를 찾게 되었다. 일단 관측치가 풍부하였고, 변수의 개수와 종류도 적절하고, 분류변수도 이항변수여서 로지스틱모형을 쓰기도 적합하였으며, classfication data 이므로, 강의안의 내용을 따라 프로젝트를 진행하기에 적합하다고 판단하였다. 원본 파일은 엑셀파일이었으며 분석의 편의를 위해 텍스트로 구성된 범주형 변수값을 우선 이산형변수로 바꾼뒤 더미변수로 변환하여 사용하였다. 수정한 엑셀파일을 Python 으로 불러와 Python 에서 모든 작업을 진행하였다.

데이터는 총 129880 개의 관측치와 ID Number 를 제외한 총 23 개의 변수로 구성되어있다.

2) 변수설명

23 개의 변수는 22 개의 속성변수와 1 개의 분류변수로 이루어져있다.

분류변수는 "Satisfaction" 이고 변수값은 'neutral or dissatisfied' or 'satisfied' 이다. 범주형 변수이고 변수값이 두 개밖에 없으므로 이항분류를 진행할 것이다.

속성변수는 범주형변수와 연속형변수로 이루어져있다.

고객의 기본 신상정보(성별, 여행목적 등), 항공이용정보(로얄고객여부, 좌석클래스 등), 고객의 설문조사 응답정보는 범주형변수다.

나이, 비행거리, 출발지연시간, 도착지연시간은 연속형변수다.

설문조사 응답정보를 담고있는 9~22 번 변수는 범주형 변수이나 숫자의 크기가 고객의 만족도를 나타내므로 연속형변수로 취급하였다.

다음 장에 변수를 정리한 변수표가 있다.

* 변수표

	변수명	변수설명	변수타입	변수값	비고
1	Id	Id Number	명목형		필요없는 변수
2	Satisfaction	Airline satisfaction level	범주형	0 : neutral or dissatisfied 1 : satisfied	분류변수
3	Gender	Gender of the passengers	범주형	0 : Male 1 : Female	
4	Customer Type	The customer type	범주형	0 : disloyal Customer 1 : Loyal Customer	
5	Age	The actual age of the passengers	연속형		
6	Type of Travel	Purpose of the flight of the passengers	범주형	0 : Personal Travel 1 : Business Travel	
7	Class	Travel class in the plane of the passengers	범주형	0 : Business 1 : Eco 2 : Eco Plus	더미변수로 바꾸어 사용
8	Flight distance	The flight distance of this journey	연속형		
9	Seat comfort	Satisfaction level of Seat comfort	범주형	0, 1, 2, 3, 4, 5 0 : Not Applicable 5 : Very Applicable 숫자가 클수록 긍정적 (10~22 번 변수 동일)	
10	Departure/ Arrival time convenient	Satisfaction level of Departure/Arrival time convenient	범주형		
11	Food and drink	Satisfaction level of Food and drink	범주형		

12	Gate location	Satisfaction level of Gate location	범주형		
13	Inflight wifi service	Satisfaction level of the inflight wifi service	범주형		
14	Inflight entertainment	Satisfaction level of inflight entertainment	범주형		
15	Online support	Satisfaction level of Online support	범주형		
16	Ease of Online booking	Satisfaction level of online booking	범주형		
17	On-board service	Satisfaction level of On-board service	범주형		
18	Leg room service	Satisfaction level of Leg room service	범주형		
19	Baggage handling	Satisfaction level of baggage handling	범주형		
20	Check-in service	Satisfaction level of Check-in service	범주형		
21	Cleanliness	Satisfaction level of Cleanliness	범주형		
22	Onlince boarding	Satisfaction level of online boarding	범주형		
23	Departure Delay in Minutes	Minutes delayed when departure	연속형		
24	Arrival Delay in Minutes	Minutes delayed when Arrival	연속형		

Ⅱ. 본론

1. 데이터 전처리

1) 필요한 패키지

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import random
import matplotlib as mpl
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import preprocessing
from sklearn.utils.class_weight import compute_sample_weight
%matplotlib inline
```

벡터와 데이터프레임을 다루기 위해 numpy 패키지와 pandas 패키지를 사용한다. 그리고 표, 그래프로 시각화를 시키기 위해 matplotlib 패키지와 seaborn 패키지를 사용한다. 랜덤샘플링을 위해 random 패키지를 사용하고, 모형구축을 위해 scikit-learn 패키지를 사용한다.

2) 데이터 크기 축소

우리 데이터의 관측치는 총 129880 개 이다. 원본 엑셀파일의 범주형 변수를 이산형 변수로 변환하여 다시 내보내는데 상당히 오랜 시간이 걸리고 나서, 앞으로 분석을 함에 있어서 컴퓨팅 파워부족으로 인한 시간지연문제를 겪을 것으로 예상되어 데이터의 크기를 줄이기로 했다. 그래서 총 관측치의 약 10%정도의 크기인 12000 개의 관측치를 simple random sampling 하여 뽑힌 관측치들만을 사용했다.

```
data.head()
```

id	satisfaction_v2	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	...	Online support	Ease Onlin book
55304	1	0	0	23	1	1	2695	5	4	4	...	5	4
98174	1	1	1	41	1	0	340	1	1	1	...	5	5
114636	1	0	1	45	1	0	2702	1	1	1	...	4	5
102451	1	1	1	11	0	1	1205	4	5	4	...	1	1
48872	1	0	1	55	1	0	83	1	1	1	...	3	3

5 rows × 23 columns

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12000 entries, 55304 to 86560
Data columns (total 23 columns):
satisfaction_v2                   12000 non-null int64
Gender                            12000 non-null int64
Customer Type                     12000 non-null int64
Age                               12000 non-null int64
Type of Travel                    12000 non-null int64
Class                             12000 non-null int64
Flight Distance                   12000 non-null int64
Seat comfort                      12000 non-null int64
Departure/Arrival time convenient 12000 non-null int64
Food and drink                    12000 non-null int64
Gate location                     12000 non-null int64
Inflight wifi service             12000 non-null int64
Inflight entertainment            12000 non-null int64
Online support                    12000 non-null int64
Ease of Online booking            12000 non-null int64
On-board service                  12000 non-null int64
Leg room service                  12000 non-null int64
Baggage handling                  12000 non-null int64
Checkin service                   12000 non-null int64
Cleanliness                       12000 non-null int64
Online boarding                   12000 non-null int64
Departure Delay in Minutes        12000 non-null int64
Arrival Delay in Minutes          11958 non-null float64
dtypes: float64(1), int64(22)
memory usage: 2.2 MB
```

3) 결측치 처리

1] 결측치 탐지

```
print('Missing values: %i' % data.isnull().sum().sum())
```

```
Missing values: 42
```

총 42 개의 결측치가 관측되었다. 어느 변수에서 존재하는지 찾아봤다.

```
data.isnull().sum()
```

```
satisfaction_v2                   0
Gender                            0
Customer Type                     0
Age                               0
Type of Travel                    0
Class                             0
Flight Distance                   0
Seat comfort                      0
Departure/Arrival time convenient 0
Food and drink                    0
Gate location                     0
Inflight wifi service             0
Inflight entertainment            0
Online support                    0
Ease of Online booking            0
On-board service                  0
Leg room service                  0
Baggage handling                  0
Checkin service                   0
Cleanliness                       0
Online boarding                   0
Departure Delay in Minutes        0
Arrival Delay in Minutes         42
```

Arrival Delay in Minutes 변수에서만 42 개의 결측치가 관측되었음을 확인할 수 있다.

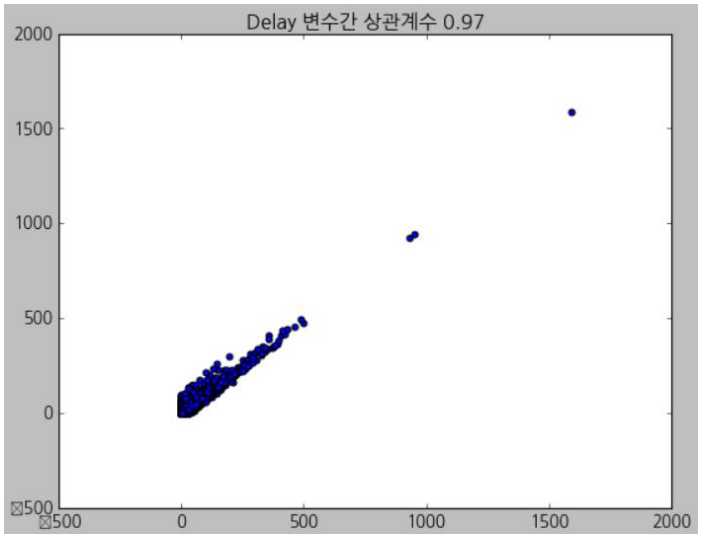
2] 결측치 처리

결측치가 있는 관측치들만 추출해서 따로 만든 데이터셋과 원래 데이터셋의 특성을 비교해봤다. 평균, 표준편차, 분위수 등 특성에서 둘의 데이터셋은 거의 일치하는 수준으로 유사성을 보였으나 Departure Delay in Minutes 의 평균에서만 차이를 보였다. 전체 데이터셋에서 Departure Delay in Minutes 의 Mean 은 14.7615, 결측치만 모은 데이터셋에서 Deparutre Delay in Minutes 의 Mean 은 38.7857 으로 많은 차이를 보였다. 따라서 결측치를 무시하여 제거하기 어렵다고 판단했다.

전체 데이터셋에서 Deparutre Delay in Minutes 와 Arrival Delay in Minutes 는 평균과 표준편차, 그리고 심지어 이상치가 있는 max 도 비슷하였다. 그래서 두 변수를 일치하는 변수로 볼 수 있다면 Arrival Delay in Minutes 의 결측치를 Departure Delay in Minutes 의 값으로 대체할 수 있다고 생각했다.

	Departure Delay in Minutes	Arrival Delay in Minutes
count	12000.000000	11958.000000
mean	14.845583	15.078943
std	40.105745	40.136804
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	13.000000	13.000000
max	1592.000000	1584.000000

상관관계도 0.97 로 매우 높아 두 변수는 같다고 보고 결측치값을 Departure Delay in Minutes 의 값으로 대체하였다.



```
print('Missing values: %i' % data.isnull().sum().sum()) ###missing value가 남지 않았다

Missing values: 0
```

이후 결측치가 더 이상 없음을 확인할 수 있다.

3) 더미변수 생성

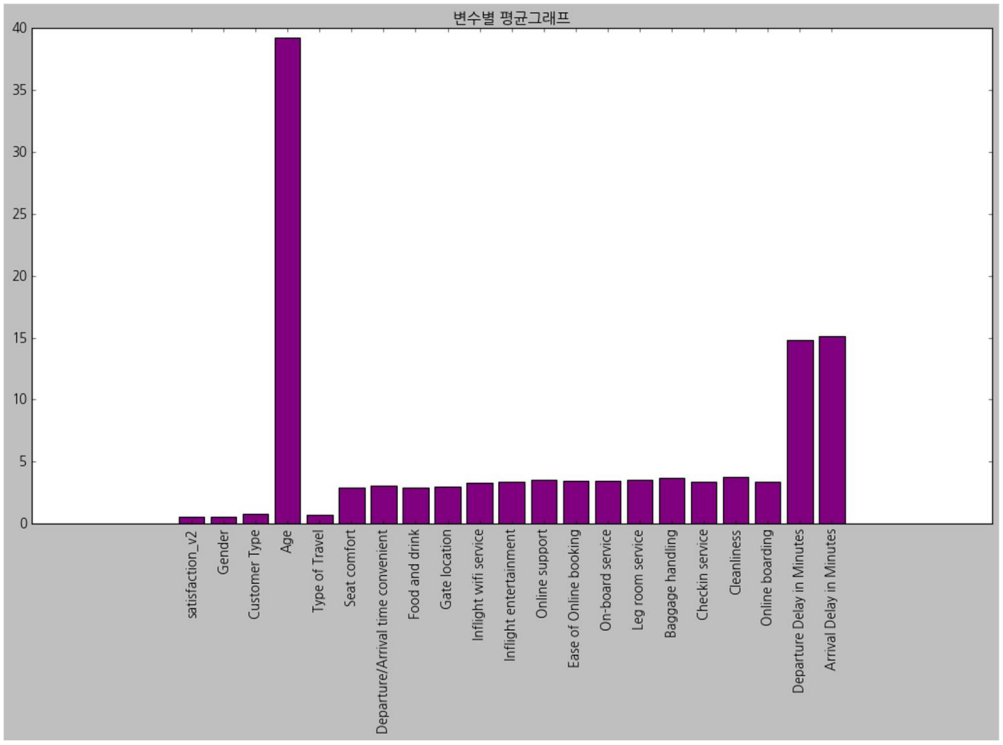
Class 변수의 범주의 개수가 3 개이므로 더미변수를 만들어준다. 나머지 범주형 변수는 범주의 개수가 2 개이므로 변수값을 0, 1 로 만들어주거나, 연속형변수로(변수표 9~22 번) 취급하였다.

Index	Class_Eco	Class_Eco_plus
55304	1	0
98174	0	0
114636	0	0
102451	1	0
48872	0	0
97167	1	0

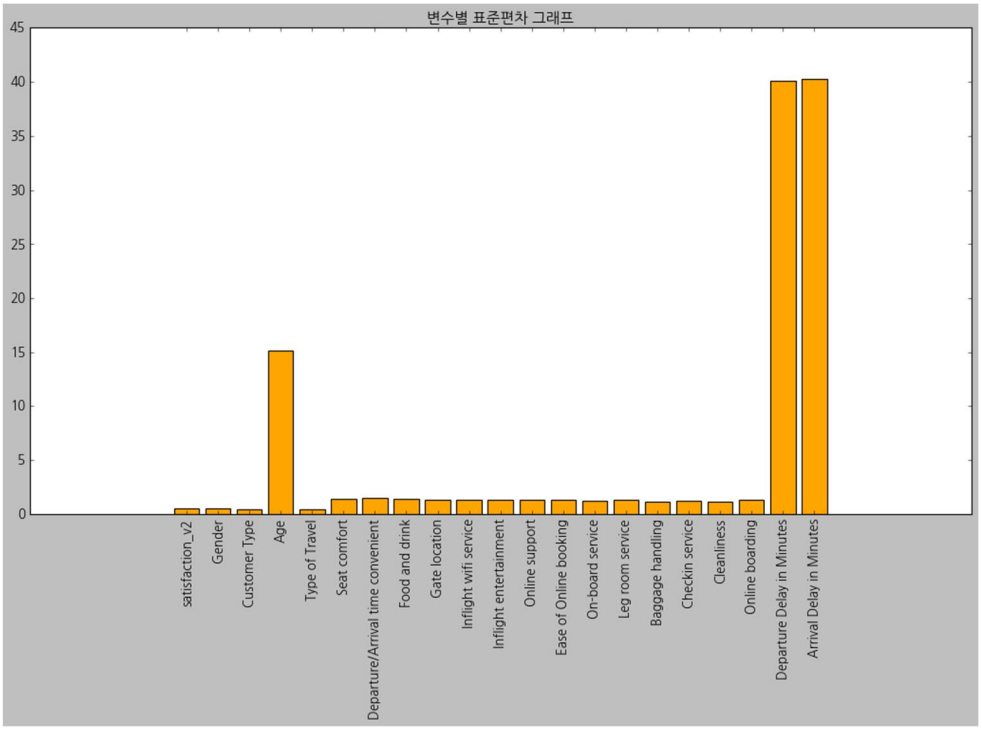
Class 변수는 Class_Eco 변수와 Class_Eco_plus 변수로 바뀌었다.

4) 각 변수의 측도

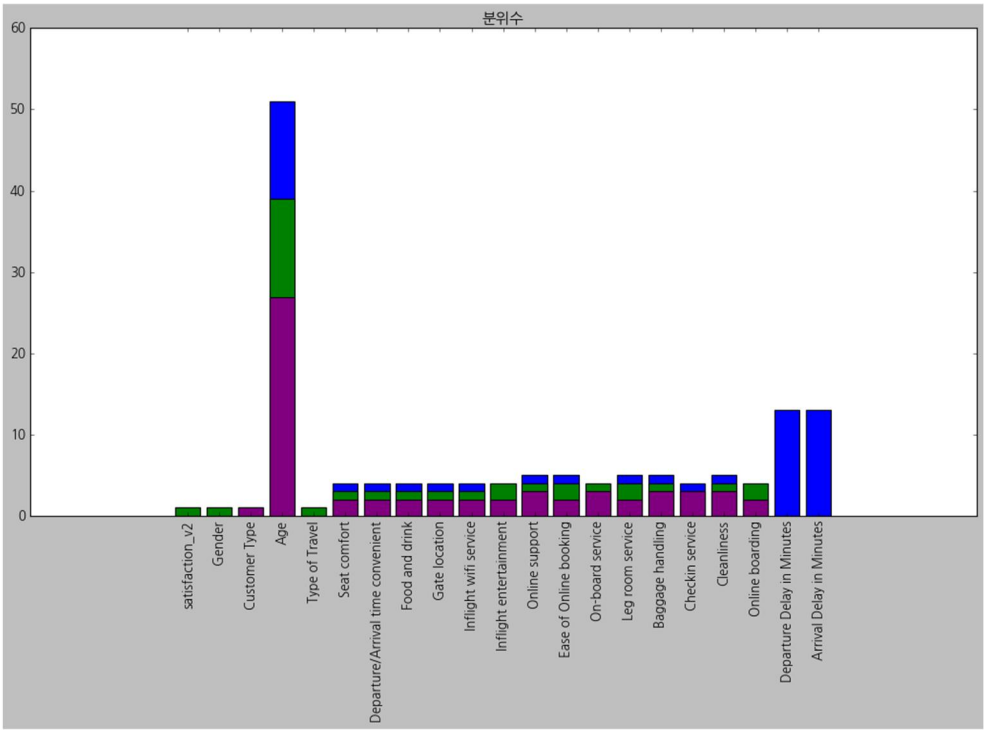
1] 변수별 평균 그래프



2] 변수별 표준편차 그래프

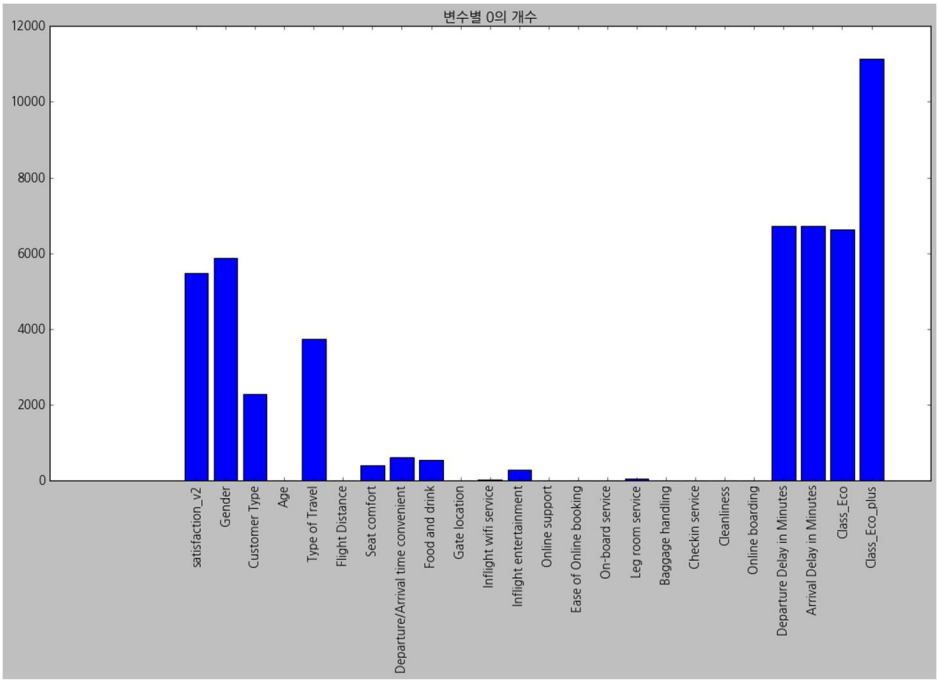


3] 변수별 분위수 그래프



* 보라색영역 : 0~25%, 초록색영역 : 25~50%, 파랑색영역 :50~75%

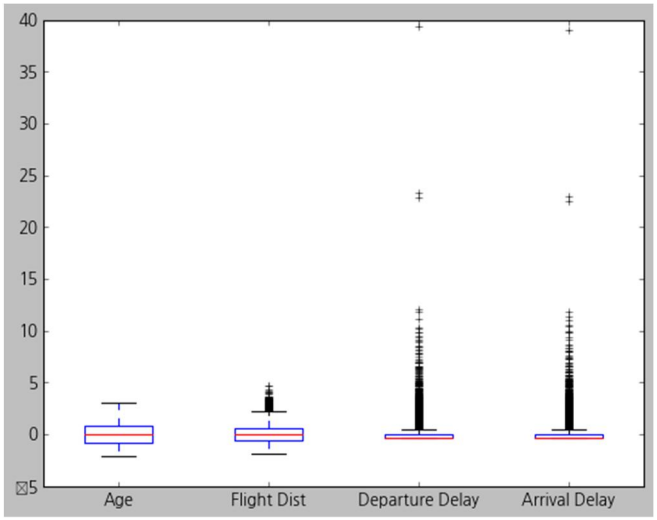
4] 변수별 0 의 개수 그래프



5) 이상치 처리

1] 이상치 탐지

범주형 변수들 중 변수값이 범주외의 값이 있는지 확인한 결과, 범주를 벗어나는 값은 존재하지 않았다. 연속형 변수들은 표준화를 시킨뒤 BoxPlot 을 그려서 이상치가 있는지 확인하였다.



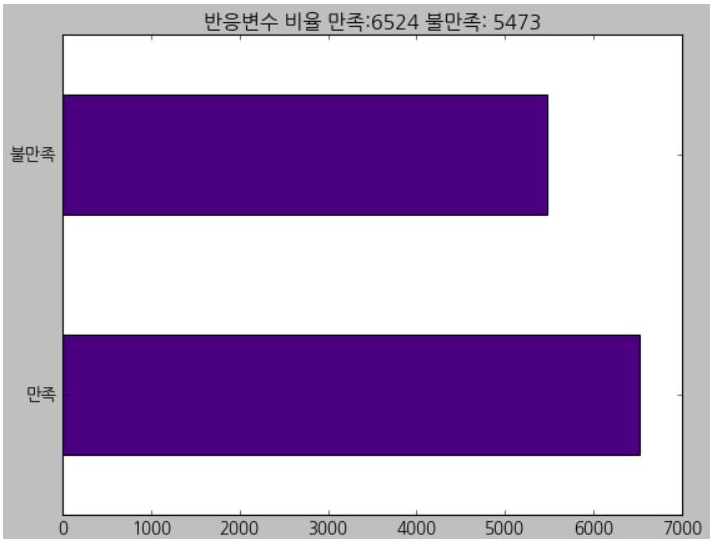
Departure Delay in Minutes 변수와 Arrival Delay in Minutes 변수에서 값이 심하게 큰 관측치들이 발견되었다. 표준화값이 20 이 넘는 관측치들을 확인했다. 비행기 지연시간이 15 시간이 넘는다는 것을 알 수 있었다.

```
id
63689     951
8345      933
73471    1592
Name: Departure Delay in Minutes, dtype: int64
```

2] 이상치 처리

표준화 시킨 후 변수값이 20 이 넘는 관측치들은 분석에 있어서 변수를 왜곡시키기에 값이 충분히 크다고 판단해서 제거했다.

6) 분류변수 분포



분류변수 0 과 1 의 비율이 1:1 에서 많이 벗어나지 않는다. 따라서 우선 weight 을 주지 않고 분석을 하기로 했다.

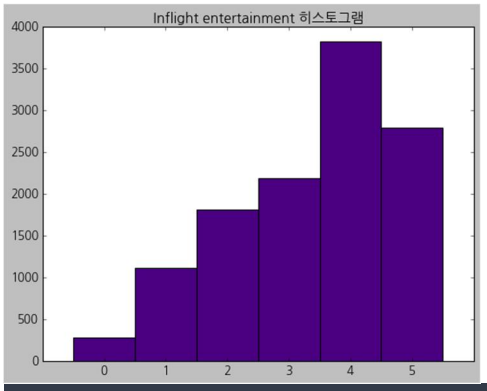
7) 유력변수 파악

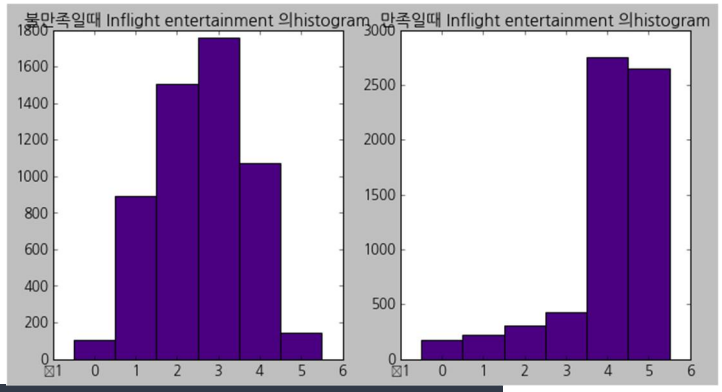
```
satisfaction_v2                 1.000000
Inflight entertainment          0.531620
Ease of Online booking          0.430382
Online support                  0.394540
On-board service                0.343112
Online boarding                 0.341930
Leg room service                0.303594
Customer Type                   0.294204
Seat comfort                    0.266109
Class_Eco                       0.265459
Checkin service                 0.253339
Baggage handling                0.251268
Cleanliness                     0.243887
Inflight wifi service           0.227387
Gender                          0.217215
Food and drink                  0.139887
Age                             0.116492
```

분류변수 satisfaction 과 각 변수간의 상관관계를 파악하여 높은 순으로 나열해 보았다.

상관계수가 가장 높은 변수 3 가지만 뽑아 특성을 히스토그램과 오즈비를 통해 파악해보았다.

1] 첫번째 변수 Inflight entertainment

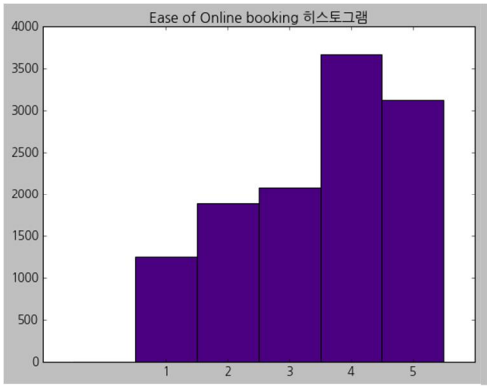


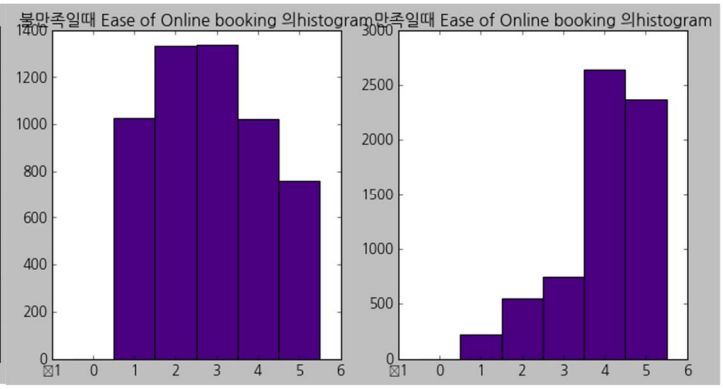



```
Inflight entertainment의 값이 0 이 277 개 있고 오즈는 1.6634615384615385 이다
Inflight entertainment의 값이 1 이 1111 개 있고 오즈는 0.24831460674157305 이다
Inflight entertainment의 값이 2 이 1813 개 있고 오즈는 0.20545212765957446 이다
Inflight entertainment의 값이 3 이 2182 개 있고 오즈는 0.24188958451906659 이다
Inflight entertainment의 값이 4 이 3823 개 있고 오즈는 2.5629077353215286 이다
Inflight entertainment의 값이 5 이 2791 개 있고 오즈는 18.248275862068965 이다
```

Inflight entertainment 의 값이 5 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 18.2 배 큰 반면, 1 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 1.7 배 큰 것으로 많은 차이가 있다.

2] 두번째 변수 Ease of Online booking

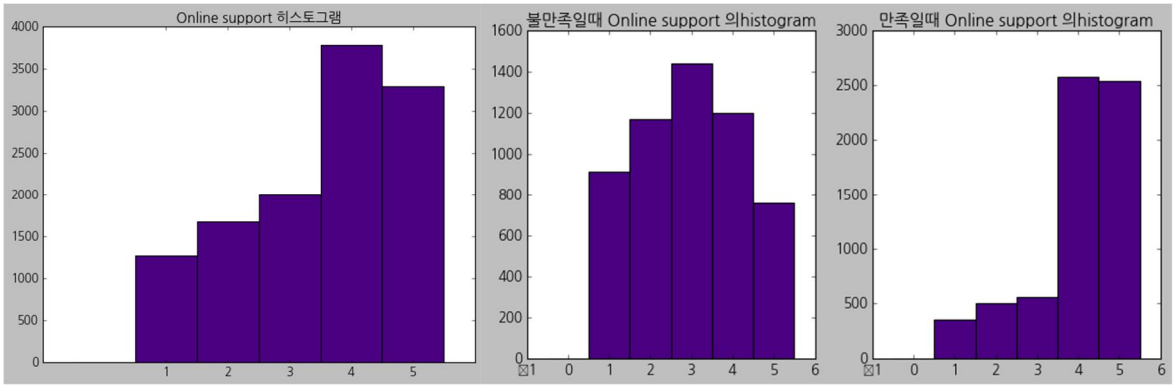





```
Ease of Online booking의 값이 1 이 1248 개 있고 오즈는 0.21875 이다
Ease of Online booking의 값이 2 이 1884 개 있고 오즈는 0.4154770848985725 이다
Ease of Online booking의 값이 3 이 2078 개 있고 오즈는 0.5553892215568862 이다
Ease of Online booking의 값이 4 이 3662 개 있고 오즈는 2.583170254403131 이다
Ease of Online booking의 값이 5 이 3125 개 있고 오즈는 3.111842105263158 이다
```

Ease of Onlie booking 의 값이 5 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 3.1 배 큰 반면, 1 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 0.2 배로 오히려 더 낮은 것을 알 수 있다.

3] 세번째 변수 Online support




```
Online support의 값이 1 이 1271 개 있고 오즈는 0.38907103825136613 이다
Online support의 값이 2 이 1671 개 있고 오즈는 0.43310463121783876 이다
Online support의 값이 3 이 1991 개 있고 오즈는 0.3864902506963788 이다
Online support의 값이 4 이 3775 개 있고 오즈는 2.1458333333333335 이다
Online support의 값이 5 이 3289 개 있고 오즈는 3.3505291005291005 이다
```

Online support 의 값이 5 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 3.4 배 큰 반면, 1 일 때 고객이 만족할 가능성은 만족하지 않을 가능성보다 0.4 배로 오히려 더 낮은 것을 알 수 있다.

종합하여 보면, 충분히 유의한 변수들이 있을 것이라는 점을 예측할 수 있다.

8) 변수변환

분류알고리즘을 적용하기 전에는 raw 데이터를 정규화하여 각 변수값들의 상대적인 크기를 같게하는 작업을 거치는 것이 좋다고 판단하여 변수변환을 했다. 변수변환방법은 Standard Scaler 를 사용했다.

```
X.head()
```

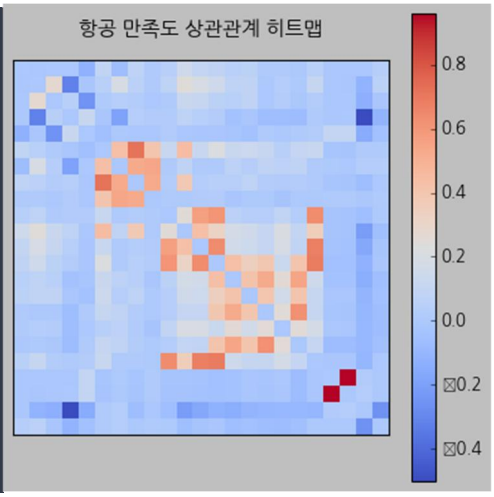
	Gender	Customer Type	Age	Type of Travel	Flight Distance	Seat comfort	Departure/Arrival time convenient	Food and drink	Gate location	Inflight wifi service	...	On-board service	
0	-3.062847	-7.340629	-2.649579	-0.033367	-1.939676	-0.951589	-1.550510	-1.438388	-2.892789	-2.023242	...	-2.419142	-1
1	0.938849	-0.833217	-2.571658	-0.033367	-1.941916	-3.026499	-2.841283	-2.878887	-3.482032	-2.598387	...	-1.790601	-1
2	-3.062847	-0.833217	-2.554342	-0.033367	-1.939669	-3.026499	-2.841283	-2.878887	-3.482032	-2.598387	...	-1.790601	-1
3	0.938849	-0.833217	-2.701527	-4.694752	-1.941093	-1.470316	-1.120253	-1.438388	-3.482032	-3.748675	...	-2.419142	-1
4	-3.062847	-0.833217	-2.511053	-0.033367	-1.942160	-3.026499	-2.841283	-2.878887	-3.482032	-3.173531	...	-3.047683	-2

9) 상관관계

1] 상관관계 파악

속성변수들 간에도 상관관계가 어떤 형태를 가지고 있는지 알아보았다. 다음은 서로 상관관계가 0.6 이 넘는 변수쌍들과 그 상관관계를 나타낸 것이다.

Seat comfort	Food and drink	0.719802
Food and drink	Seat comfort	0.719802
Inflight wifi service	Ease of Online booking	0.606855
	Online boarding	0.638152
Online support	Ease of Online booking	0.635854
	Online boarding	0.679442
Ease of Online booking	Inflight wifi service	0.606855
	Online support	0.635854
	Online boarding	0.692356
Baggage handling	Cleanliness	0.623317
Cleanliness	Baggage handling	0.623317
Online boarding	Inflight wifi service	0.638152
	Online support	0.679442
	Ease of Online booking	0.692356
Departure Delay in Minutes	Arrival Delay in Minutes	0.961269
Arrival Delay in Minutes	Departure Delay in Minutes	0.961269



상관관계가 높은 변수쌍이 많은 것을 확인했다.

2] 공분산검사 및 변수제거

다중공선성을 판단하기 위해 VIF 를 사용하였다. VIF 값이 10 이상이면 해당 변수가 다중공선성이 존재하는 것으로 판단하였다. 따라서 While 문으로 10 이상의 VIF 값이 존재하지 않을 때 까지 변수를 제거하여 총 7 개의 변수를 제거하여 16 개의 변수를 남겼다.

<변수 제거 전>

	VIF Factor	features
0	2.136663	Gender
1	7.139145	Customer Type
2	6.262177	Age
3	4.464305	Type of Travel
4	4.712663	Flight Distance
5	13.124873	Seat comfort
6	8.957879	Departure/Arrival time convenient
7	13.815501	Food and drink
8	5.751200	Gate location
9	13.314338	Inflight wifi service
10	14.031827	Inflight entertainment
11	11.449720	Online support
12	16.466484	Ease of Online booking
13	8.611286	On-board service
14	10.700058	Leg room service
15	11.813318	Baggage handling
16	5.382686	Checkin service
17	12.698733	Cleanliness
18	11.767033	Online boarding
19	15.430323	Departure Delay in Minutes
20	15.498198	Arrival Delay in Minutes
21	2.290718	Class_Eco
22	1.201717	Class_Eco_plus

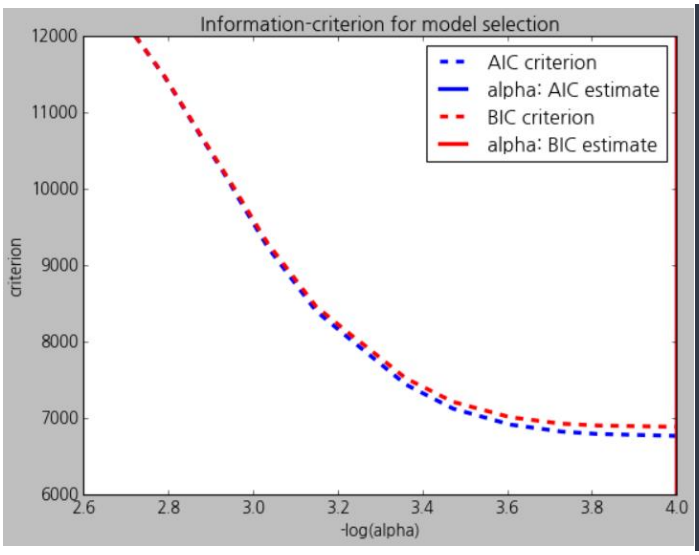
<변수 제거 후>

	VIF Factor	features
0	2.084419	Gender
1	6.522945	Customer Type
2	6.129958	Age
3	4.119284	Type of Travel
4	4.535324	Flight Distance
5	7.173796	Seat comfort
6	8.183512	Departure/Arrival time convenient
7	5.155573	Gate location
8	9.349988	Online support
9	7.033486	On-board service
10	8.526102	Baggage handling
11	4.911955	Checkin service
12	8.037106	Online boarding
13	1.185062	Departure Delay in Minutes
14	2.092704	Class_Eco
15	1.177107	Class_Eco_plus

10) AIC, BIC

다중공선성을 제거한 후 남은 변수로 회귀식에 적합한지 확인하기 위해 AIC, BIC 기법을 사용해봤다.

<AIC>



Customer Type	0.147491
Seat comfort	0.124562
Departure/Arrival time convenient	0.081363
On-board service	0.078037
Gender	0.076553
Online support	0.065166
Checkin service	0.061111
Type of Travel	0.060844
Class_Eco	0.059732
Online boarding	0.056422
Baggage handling	0.043231
Class_Eco_plus	0.038367
Departure Delay in Minutes	0.021530
Flight Distance	0.020235
Age	0.012840
Gate location	0.009915

분석결과 더 이상 제거할만한 변수는 발견되지 않았다.

11) 최종 속성변수

총 16 개

Gender : Gender of the passengers (0 : Male, 1 : Female)

Customer Type : The customer type (0 : disloyal Customer, 1 : Loyal Customer)

Age : The actual age of the passengers (연속형)

Type of Travel : Purpose of the flight of the passengers (0 : Personal Travel, 1 : Business Travel)

Flight distance : The flight distance of this journey (연속형)

Departure/Arrival time convenient : Satisfaction level of Departure/Arrival time convenient (0:Not Applicable;1-5)

Seat comfort : Satisfaction level of Seat comfort (0:Not Applicable;1-5)

Gate location : Satisfaction level of Gate location (0:Not Applicable;1-5)

Online support : Satisfaction level of Online support (0:Not Applicable;1-5)

On-board service : Satisfaction level of On-board service (0:Not Applicable;1-5)

Baggage handling : Satisfaction level of baggage handling (0:Not Applicable;1-5)

Check-in service : Satisfaction level of Check-in service (0:Not Applicable;1-5)

Online boarding : Satisfaction level of online boarding (0:Not Applicable;1-5)

Departure Delay in Minutes : Minutes delayed when departure (연속형)

Class : Travel class in the plane of the passengers 의 더미변수 : Class_Eco, Class_Eco_plus

2. 예측 모델 구축 및 평가

1) 사전작업

1] 모델 적합을 위한 패키지들

예측모형을 위한 패키지들 imort

```
from sklearn.model_selection import train_test_split
import sklearn
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB, BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neural_network import *
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import roc_auc_score
from sklearn.metrics import average_precision_score
from sklearn.metrics import accuracy_score
from sklearn.model_selection import GridSearchCV
```

모델을 만들기 위해 사이킷런 패키지에서 로지스틱 회귀모형, KNN, 나이브 베이즈 모델, 의사결정나무 , 랜덤포레스트, 아다부스트, 그래디언트 부스트, 서포트 벡터 머신등을 가져왔다.

평가를 위해 roc 커브를 그려주고 auc 점수를 계산해주는 함수, PR 커브와 AP 점수를 계산해주는 함수, 그리고 정확도를 계산해주기 위한 함수와 교차표 작성을 위한 함수를 불러와 주었다.

하이퍼 파라미터 결정을 위해 교차검증을 통해 최적의 파라미터를 찾아주는 GridSearchCV 함수를 불러왔다.

2] 이익도표(함수)

그리고 모델 평가를 위해 사용할 이익도표를 만들기 위한 함수를 따로 정의해 준다.

이익도표를 그리기 위해 먼저 모델의 예측값이 확률 형태로 나오거나 아님 로지스틱 모형처럼 오즈값이 나올경우 그 결과로 모든 관측치에 순서를 매겨서 10 개의 분위수로 나눠 관측치 구간을 나눠준다. 그리고 각 구간별 관측치의 개수, lift 값, 활성화 비율, 실제로 만족한 사람이 얼마나 있는지를 구해준다. 그리고 그에 따른 누적 지표들도 구해준다.

```
calc_lift(x,y,clf,bins=10):
 #Actual Value of y
y_actual = np.hstack(y)
#Predicted Probability that y = 1
y_prob = clf.predict_proba(x)
#Predicted Value of Y
y_pred = clf.predict(x)
cols = ['ACTUAL','PROB_POSITIVE','PREDICTED']
data = [y_actual,y_prob[:,1],y_pred]
df = pd.DataFrame(dict(zip(cols,data)))
#Observations where y=1
total_positive_n = df['ACTUAL'].sum()
#Total Observations
total_n = df.index.size
natural_positive_prob = total_positive_n/float(total_n)
df['상위구간'] = pd.qcut(df['PROB_POSITIVE'],bins, labels=False, duplicates='drop')
pos_group_df = df.groupby('상위구간')
#Percentage of Observations in each Bin where y = 1
actual=pos_group_df['ACTUAL'].sum().sort_index(ascending=False)
bin_count=pos_group_df['ACTUAL'].count().sort_index(ascending=False)
cumsum=np.cumsum(bin_count)
cumsum_percentage=np.cumsum(bin_count)/np.sum(bin_count)
lift_positive = pos_group_df['ACTUAL'].sum().sort_index(ascending=False)/pos_group_df['ACTUAL'].count().sort_in
cum_active=np.cumsum(pos_group_df['ACTUAL'].sum().sort_index(ascending=False))/np.cumsum(pos_group_df['ACTUAL']
cum_lift_positive=np.cumsum(pos_group_df['ACTUAL'].sum().sort_index(ascending=False))/np.cumsum(pos_group_df['A
cum_lift_positive=cum_lift_positive/natural_positive_prob
lift_index_positive = (lift_positive/natural_positive_prob)

#Consolidate Results into Output Dataframe
lift_df = pd.DataFrame({ '구간 관측치 개수':bin_count
                        ,'구간 활성화 비율(%)':lift_positive*100 ,
                           '구간 LIFT':lift_index_positive,
                         '구간 실제 만족한 사람 수':actual,
                         '누적 관측치 개수':np.round(cumsum,1),
                        '누적 활성화 비율 %':np.round(cum_active,2),
                      '누적 실제 만족한 사람수':np.cumsum(actual),
                      '누적 lift (%)':cum_lift_positive})
lift_df.index=[1,2,3,4,5,6,7,8,9,10]
lift_df.index.name='구간'
return lift_df
```

3] 데이터 분리 (train set, Validation set, test set)

먼저 우리는 앞서 살펴본 대로 변수는 총 16 개를 쓸 것이다.

```
사용할 변수들
Index(['Gender', 'Customer Type', 'Age', 'Type of Travel', 'Flight Distance',
       'Seat comfort', 'Departure/Arrival time convenient', 'Gate location',
       'Online support', 'On-board service', 'Baggage handling',
       'Checkin service', 'Online boarding', 'Departure Delay in Minutes',
       'Class_Eco', 'Class_Eco_plus'],
      dtype='object')
사용할 변수개수: 16
```

Training Set 과 Validation Set 그리고 Test set 의 비율을 0.7:0.15:0.15 로 분리 하였다.

```
X and y Input Data:    (11997, 16) (11997,)
Training Set Shape:    (8397, 16) (8397,)
Validation Set Shape:  (1800, 16) (1800,)
Test Set shape: (1800, 16) (1800,)
```

각 Set 별 반응변수의 비율은 다음과 같다.

```
training set 반응변수 비율 0: 3811   1: 4586
Validation set 반응변수 비율 0: 826   1: 974
Test set 반응변수 비율 0: 836   1: 964
```

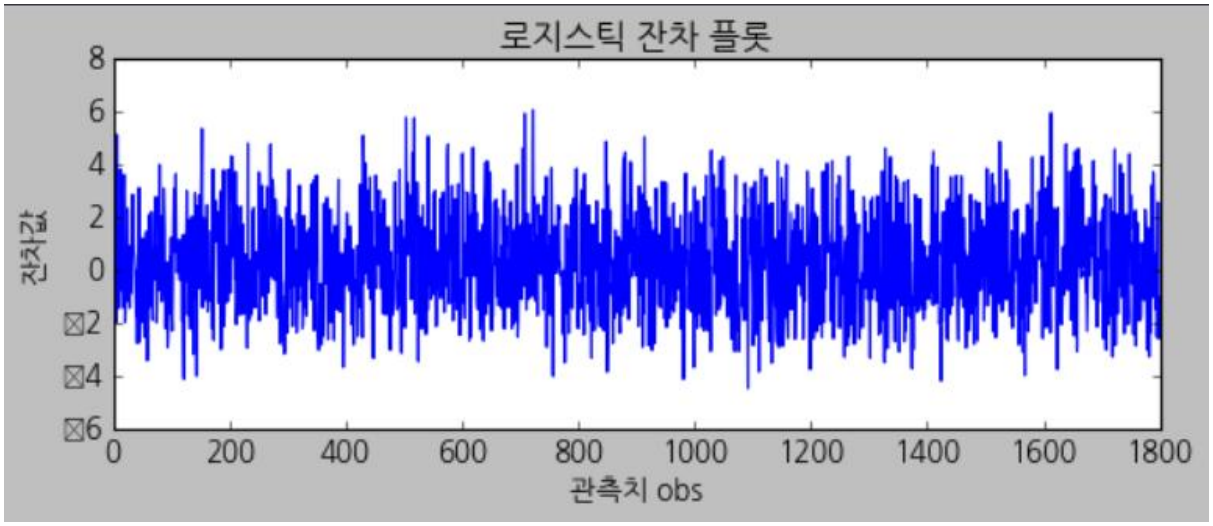
2) 모델구축 및 평가

1] 로지스틱 회귀모형

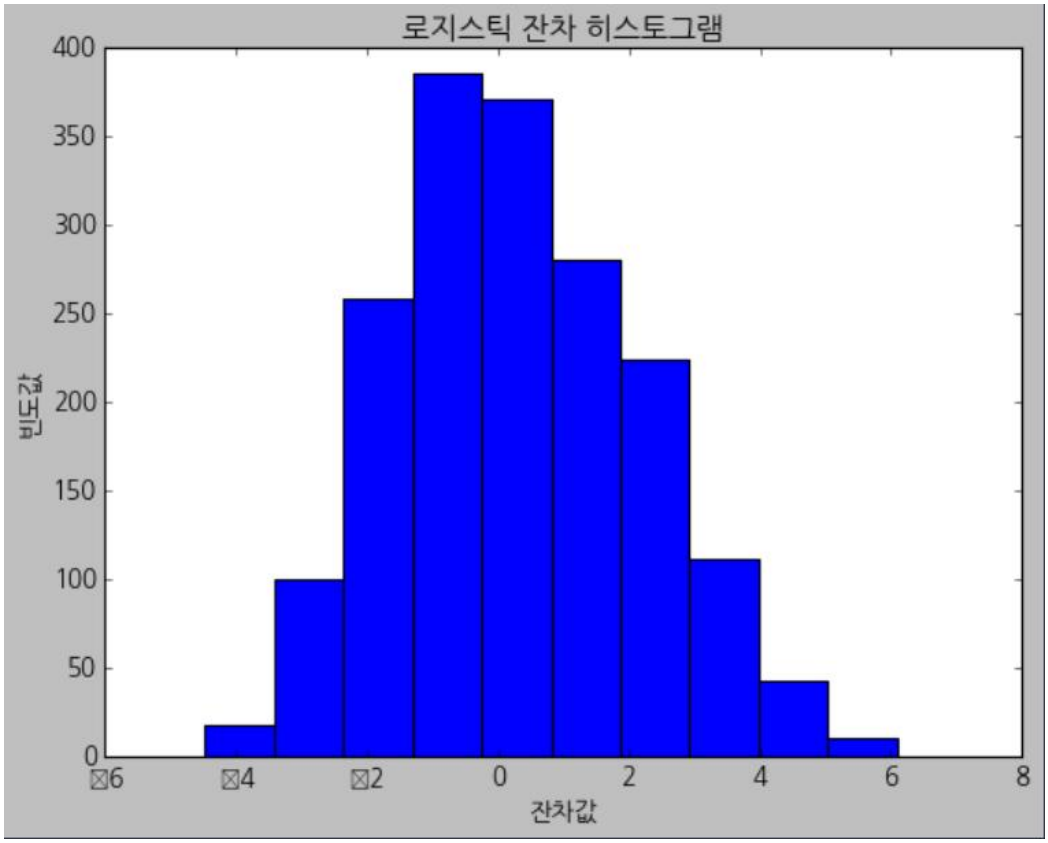
먼저 로지스틱 회귀 모형에 적합해 보았다. 아래는 로지스틱모형에서의 변수별 계수와 유의확률이다.

	Columns	Coefficients	p_Value
0	Gender	0.552474	0.000
1	Customer Type	0.922497	0.000
2	Age	-0.033884	0.298
3	Type of Travel	0.439030	0.000
4	Flight Distance	-0.113213	0.001
5	Seat comfort	0.793209	0.000
6	Departure/Arrival time convenient	-0.510503	0.000
7	Gate location	-0.032541	0.362
8	Online support	0.416263	0.000
9	On-board service	0.539116	0.000
10	Baggage handling	0.258404	0.000
11	Checkin service	0.439569	0.000
12	Online boarding	0.387884	0.000
13	Departure Delay in Minutes	-0.145883	0.000
14	Class_Eco	-0.377584	0.000
15	Class_Eco_plus	-0.219845	0.000

Gate location, Age 만을 제외하고 모두 a=0.05 라 했을 때 유의하게 나온 걸 볼 수 있다. 제대로 적합된 회귀식인지 살펴보기 위해 잔차를 살펴보기 위해 먼저 관측치별 잔차값을 그려보면



별다른 주기없이 잔차들이 독립성을 가진 걸 확인 할 수 있다.



잔차 값의 히스토그램을 그려보면 0 을 중심으로 정규분포 모양이 생기는걸 볼 수 있으므로 회귀식의 가정에 어긋나지 않음을 볼 수 있다.

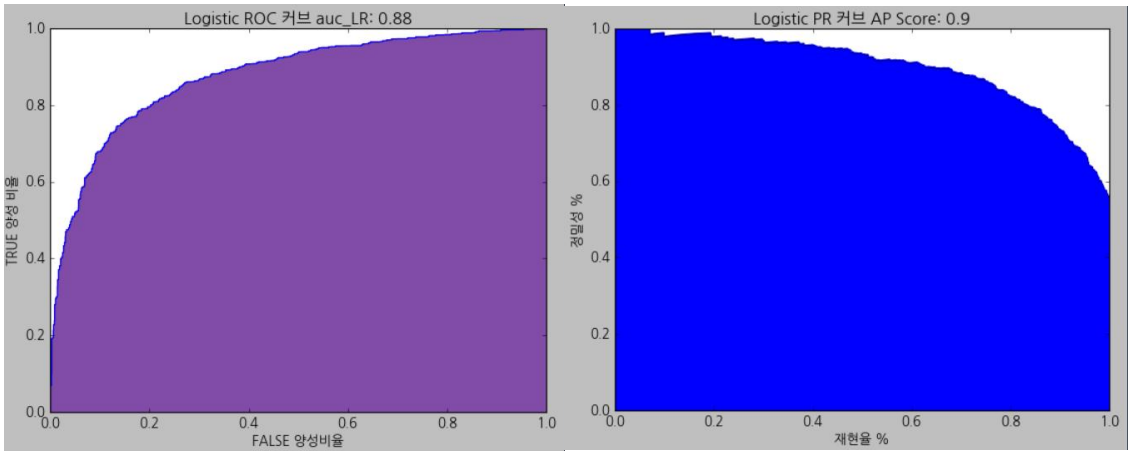
* 모형 평가

모델의 정확도는 : 80%

Confusion Matrix 는

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	628	198
	Positive	169	805

하지만 정확도와 교차표는 임계값을 달리함에 따라 언제든지 바뀔수 있는 지표이다. 그래서 모델의 종합적인 성능을 보기 위해 ROC 커브와 PR 커브를 그려보았다.



ROC 커브는 x 축에 False 양성비율 y 축에 True 양성비율을 놓음으로서 실제 활성화 값을 찾기 위해선 거짓 참값이 늘어 날 수 밖에 없다는 trade-off 관계를 나타낸다. 우리의 경우 True 양성비율은 실제 만족한 사람들 중 맞게 만족한다 대답한 사람들의 비율이고 False 양성비율은 불만족하는데 우리가 만족한다 잘못 예측한 사람들의 비율이다.

ROC 커브는 모형의 임계값을 어디에 두어야 할지 판단할 때 True 양성비율과 False 양성비율의 비교를 통해 적절한 cut off 지점을 찾을 수 있게 도와준다.

PR 커브는 실제 재현율을 얼마나 얻고 싶은지에 따라 정밀도를 얼마나 희생해야 하는지를 잘 나타내어 준다. 만약 실제 만족하는 사람의 50%정도만을 찾고 싶으면 PR 커브의 재현율 50%에 해당하는 임계값을 구해서 그 기준으로 판별하면 된다. 이 임계값을 기준으로한 우리의 분류모델이 반응변수가 1 인 값들을 모델이 찾아내면 그 모델이 1 이라고 분류한 관측치들중 90% 정도가 실제로 1 값을 가지게 되는 것이다.

Auc 는 ROC 커브 아래의 면적이고 AP 는 Average Precision 으로서 PR 커브의 아래의 면적을 나타낸다.

이익도표: Train Set:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	840	97.976190	1.793951	823	840	0.98	823	1.793951
2	840	95.000000	1.739457	798	1680	0.96	1621	1.766704
3	839	92.133492	1.686971	773	2519	0.95	2394	1.740148
4	840	84.523810	1.547637	710	3359	0.92	3104	1.692006
5	839	64.600715	1.182844	542	4198	0.87	3646	1.590246
6	840	42.142857	0.771639	354	5038	0.79	4000	1.453757
7	840	31.785714	0.581999	267	5878	0.73	4267	1.329178
8	839	18.831943	0.344814	158	6717	0.66	4425	1.206224
9	840	11.785714	0.215797	99	7557	0.60	4524	1.096133
10	840	7.380952	0.135146	62	8397	0.55	4586	1.000000

Validation Set:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	98.888889	1.827515	178	180	0.99	178	1.827515
2	180	93.888889	1.735113	169	360	0.96	347	1.781314
3	180	85.000000	1.570842	153	540	0.93	500	1.711157
4	180	81.111111	1.498973	146	720	0.90	646	1.658111
5	180	61.111111	1.129363	110	900	0.84	756	1.552361
6	180	47.222222	0.872690	85	1080	0.78	841	1.439083
7	180	28.888889	0.533881	52	1260	0.71	893	1.309768
8	180	21.666667	0.400411	39	1440	0.65	932	1.196099
9	180	15.555556	0.287474	28	1620	0.59	960	1.095140
10	180	7.777778	0.143737	14	1800	0.54	974	1.000000

이익도표는 관측치들을 모델로 평가해 10 개의 구간으로 나눠준다. 1 구간에는 분류모델이 제일 만족한다고 대답할 확률이 높은 사람 10%를 모아놓은 것 이고 10 번쨔 구간에는 제일 만족한다고 대답할 확률이 낮은 사람들을 모아 놓은 것 이다. 구간 Lift 값은 전체 관측치 중 활성화 값의 비율대비 구간에서의 활성화 비율의 대비를 나타낸다. 1 구간과 10 구간의 lift 값 차이가 클수록 좋은 모델이라고 볼 수 있을 것이다. 그리고 이익도표를 통해서 우리가 제대로 모형을 적합시켰나 살펴볼 수 있다. 만약 구간별 lift 값이 차이가 나지 않는다거나 만족할 확률이 더 낮다고 구분한 구간이 상위 구간보다 더 높은 lift 값을 가진경우 즉 lift 값이 역전된 경우 모델이 제대로 학습이 되지 않았거나 모델의 성능이 떨어짐을 알 수 있다.

Train Set 과 Validation Set 의 이익도표를 모두 그려봄으로서 얻을 수 있는 효과는 여러가지가 있다. 먼저 train set 의 이익도표를 통해서 우리가 제대로 모형을 적합시켰나 살펴볼 수 있다. 그리고 Train Set 의 이익도표와 Validation Set 의 이익도표 비교를 통해서 train set 이 혹시 과적합된건 아닌지 판별 할 수있다. 만약 과적합되었다면 test set 의 이익도표는 과도하게 좋고 validation set 의 이익도표는 제대로 분포가 되어 있지 않은걸 확인 할 수 있을것이다.

다른모형들에 대해서도 Logistic 회귀와 마찬가지로 정확도 confusion matrix, ROC 커브 ,PR 커브, 이익도표를 그려줄 것이다. 다만 이익도표는 모델이 관측치들을 10 개 구간으로 확률을 통해 나누므로 decision tree 나 knn 같이 확률값으로 결과를 배출하지 않는 모델들은 이익도표를 그려줄 수 없다.

2] K 근접 이웃 분류기

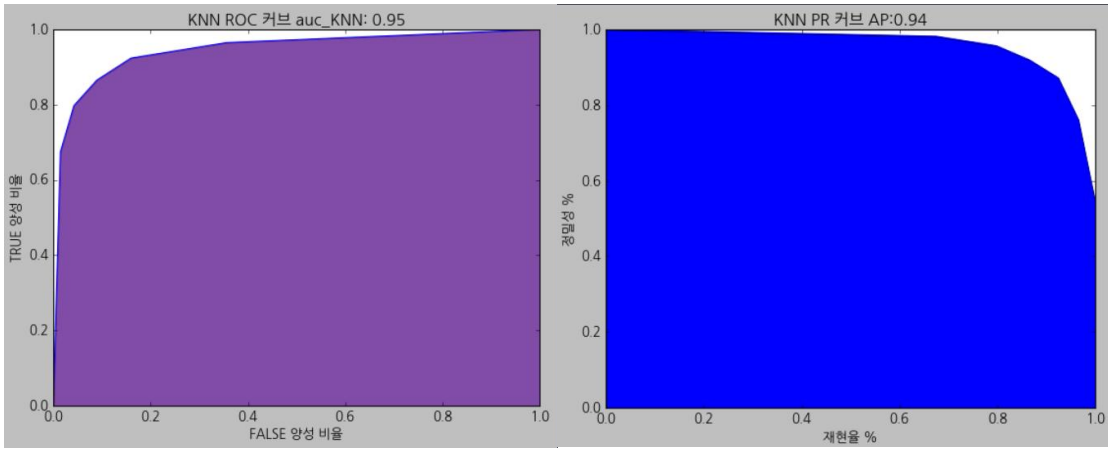
KNN 에서 이웃한 관측치를 평가함에 있어서 가중치를 주는 방법이 있고 안주는 방법이 있다. 우리의 데이터의 경우 사이킷런 패키지의 GRID SEARCH 알고리즘 즉 내부에서 교차검증을 이용해 최적의 하이퍼 파라미터를 구해주는 알고리즘을 통해 평가한 결과 가중치가 없는 것이 결과가 좋다 판별해 주어서 가중치 없는 KNN 을 학습시키도록 한다.

정확도: 89%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	752	74
	Positive	131	843

ROC 커브,PR 커브



Auc 점수: 0.95 AP 점수:0.94

KNN 의 경우 관측치들의 순위를 10 개구간으로 나눠 줄 수 없기 때문에 이익도표를 그려주지 않았다.

3] 의사결정나무 분류기

KNN 의 경우와 마찬가지로 grid search 를 통한 하이퍼 파라미터 튜닝을 통해 의사결정나무의 깊이를 7 이라고 고정해주고 학습을 시켰다.

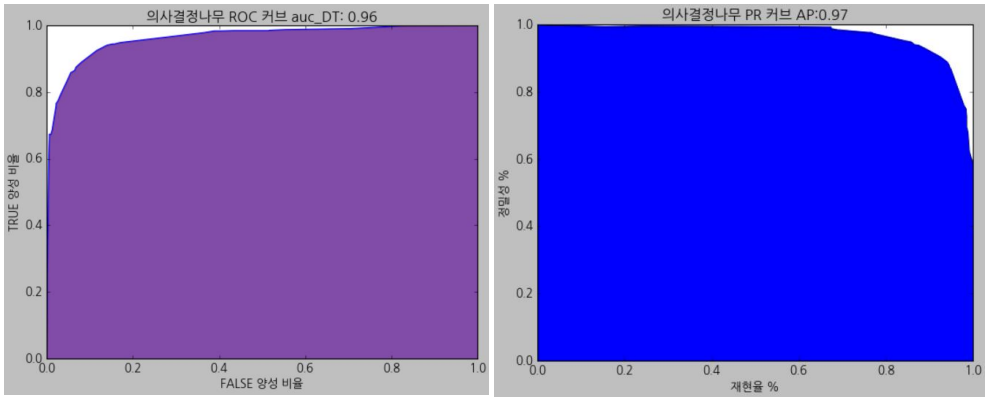
의사결정나무 시각화: 깊이가 7 이나 되는 트리이므로 너무 깊어 보고서에 담긴 크다. 그러므로 따로 그림을 첨부하겠다.

정확도: 90%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	773	53
	Positive	132	842

ROC 커브 PR 커브



auc 점수:0.96 ap 점수:0.97

Decision Tree 도 마찬가지로 관측치들의 순위화가 불가능해서 이익도표를 그려주지 못했다.

4] 나이브 베이즈 분류기

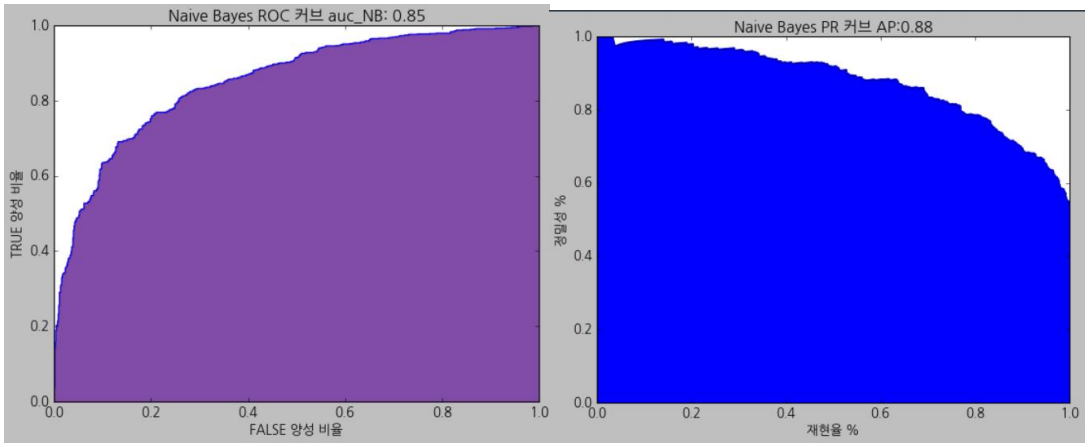
우리는 나이브베이즈 모델중 Bernoulli 에 기반한 나이브 베이즈 분류기를 사용하였다. 더 비모수적 상황에서 활용할 수 있고 우리의 반응변수가 0 또는 1 이기 때문이다.

정확도: 77%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	632	194
	Positive	221	753

ROC 커브 PR 커브



auc 점수: 0.85 ap 점수: 0.88

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	828	98.309179	1.800048	814	828	0.98	814	1.800048
2	852	93.309859	1.708510	795	1680	0.96	1609	1.753626
3	838	89.021480	1.629990	746	2518	0.94	2355	1.712479
4	841	73.127229	1.338965	615	3359	0.88	2970	1.618962
5	839	61.978546	1.134832	520	4198	0.83	3490	1.522205
6	840	48.690476	0.891526	409	5038	0.77	3899	1.417050
7	839	30.274136	0.554322	254	5877	0.71	4153	1.293887
8	839	23.361144	0.427744	196	6716	0.65	4349	1.185684
9	841	17.954816	0.328754	151	7557	0.60	4500	1.090318
10	840	10.238095	0.187460	86	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	98.333333	1.817248	177	180	0.98	177	1.817248
2	180	91.111111	1.683778	164	360	0.95	341	1.750513
3	180	85.555556	1.581109	154	540	0.92	495	1.694045
4	180	72.777778	1.344969	131	720	0.87	626	1.606776
5	180	60.000000	1.108830	108	900	0.82	734	1.507187
6	179	44.692737	0.825944	80	1079	0.75	814	1.394172
7	181	34.254144	0.633033	62	1260	0.70	876	1.284834
8	180	28.888889	0.533881	52	1440	0.64	928	1.190965
9	179	14.525140	0.268432	26	1619	0.59	954	1.088968
10	181	11.049724	0.204204	20	1800	0.54	974	1.000000

이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

5] 신경망 모델

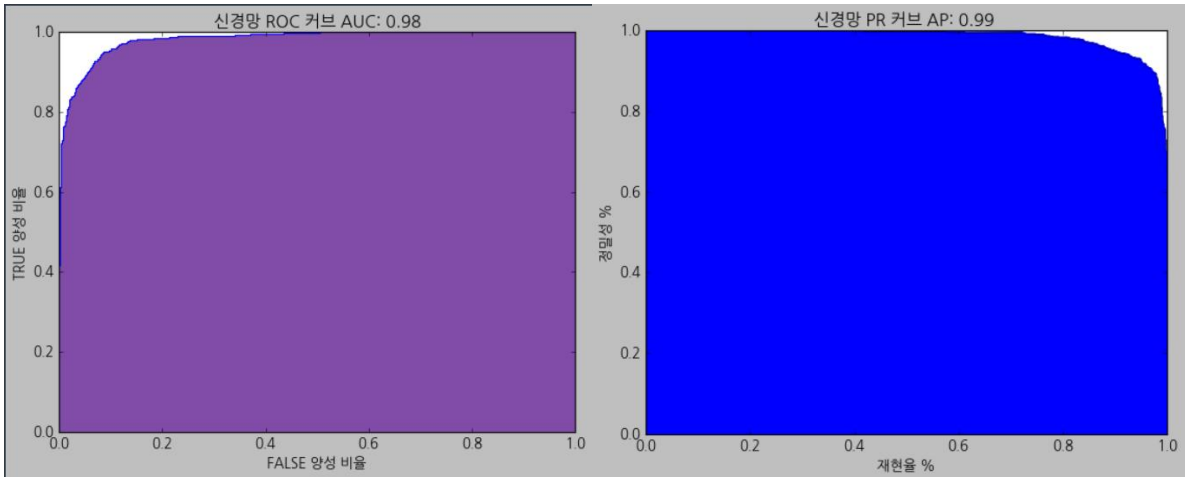
우리는 신경망 모델을 적합함에 있어서 히든레이어 개수는 1 개, 활성화 함수는 Relu 함수, 히든레이어의 노드 개수는 128 개로 학습시켰다.

정확도: 93%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	758	68
	Positive	54	920

ROC 커브 PR 커브



auc 점수: 0.98 ap 점수:0.99

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	840	100.000000	1.831007	840	840	1.00	840	1.831007
2	840	100.000000	1.831007	840	1680	1.00	1680	1.831007
3	839	100.000000	1.831007	839	2519	1.00	2519	1.831007
4	840	99.642857	1.824468	837	3359	1.00	3356	1.829372
5	839	98.092968	1.796090	823	4198	1.00	4179	1.822720
6	840	44.642857	0.817414	375	5038	0.90	4554	1.655103
7	840	3.452381	0.063213	29	5878	0.78	4583	1.427613
8	839	0.238379	0.004365	2	6717	0.68	4585	1.249839
9	840	0.119048	0.002180	1	7557	0.61	4586	1.111155
10	840	0.000000	0.000000	0	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	100.000000	1.848049	180	180	1.00	180	1.848049
2	180	100.000000	1.848049	180	360	1.00	360	1.848049
3	180	99.444444	1.837782	179	540	1.00	539	1.844627
4	180	97.222222	1.796715	175	720	0.99	714	1.832649
5	180	81.666667	1.509240	147	900	0.96	861	1.767967
6	180	52.222222	0.965092	94	1080	0.88	955	1.634155
7	180	6.666667	0.123203	12	1260	0.77	967	1.418304
8	180	3.333333	0.061602	6	1440	0.68	973	1.248717
9	180	0.555556	0.010267	1	1620	0.60	974	1.111111
10	180	0.000000	0.000000	0	1800	0.54	974	1.000000

이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

6] 랜덤 포레스트

랜덤포레스트는 랜덤샘플을 뽑아 랜덤으로 뽑은 변수들로 decision tree 들을 여러 개 만들어 다중투표를 받는 결과를 output 으로 예측하는 모델이다.

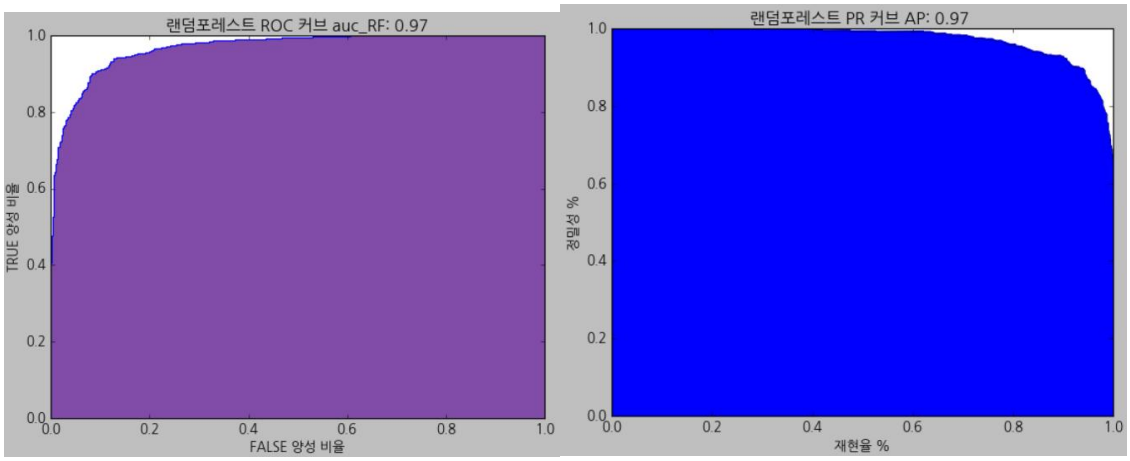
랜덤 포레스트도 트리의 깊이를 7 로 할 때가 제일 좋음을 grid search 기법을 통해 찾아주었다.

정확도: 91%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	753	73
	Positive	96	878

ROC 커브 PR 커브



auc 점수: 0.97 ap 점수: 0.97

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	839	100.000000	1.831007	839	839	1.00	839	1.831007
2	841	100.000000	1.831007	841	1680	1.00	1680	1.831007
3	839	99.165673	1.815731	832	2519	1.00	2512	1.825919
4	840	95.833333	1.754715	805	3359	0.99	3317	1.808113
5	839	83.432658	1.527658	700	4198	0.96	4017	1.752062
6	840	48.809524	0.893706	410	5038	0.88	4427	1.608946
7	840	12.738095	0.233235	107	5878	0.77	4534	1.412349
8	839	3.814064	0.069836	32	6717	0.68	4566	1.244660
9	840	1.785714	0.032697	15	7557	0.61	4581	1.109944
10	840	0.595238	0.010899	5	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	100.000000	1.848049	180	180	1.00	180	1.848049
2	179	100.000000	1.848049	179	359	1.00	359	1.848049
3	181	97.790055	1.807208	177	540	0.99	536	1.834360
4	180	92.777778	1.714579	167	720	0.98	703	1.804415
5	180	76.111111	1.406571	137	900	0.93	840	1.724846
6	180	49.444444	0.913758	89	1080	0.86	929	1.589665
7	180	17.777778	0.328542	32	1260	0.76	961	1.409504
8	180	6.111111	0.112936	11	1440	0.68	972	1.247433
9	180	0.555556	0.010267	1	1620	0.60	973	1.109970
10	180	0.555556	0.010267	1	1800	0.54	974	1.000000

이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

7] 아다부스트

부스트 모델들은 미리 정해진 개수의 모형 집합을 사용하는 것이 아니라 하나의 모형에서 시작해서 모형 집합에 포함할 개별 모형을 하나씩 추가한다.

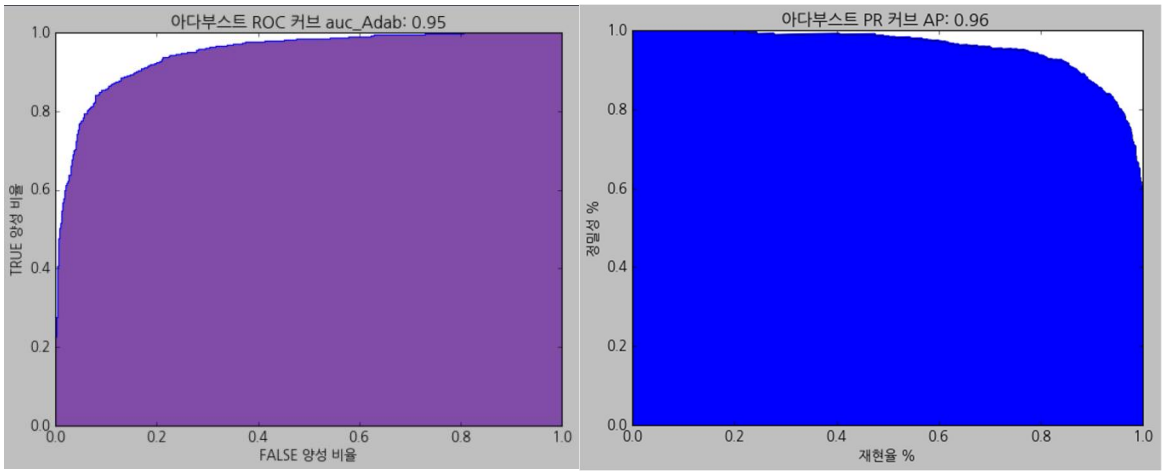
아다부스트는 하나의 모형에서 잘못 분류된 것들을 이어지는 약한 학습기들이 수정해줄 수 있다는 점에서 착안된 아이디어이다. 아다부스트는 한 분류기를 만들고 그 다음에 오분류가 된 데이터들에 가중치를 주어 새롭게 분류하는 파생되는 모델들을 순차적으로 추가해 마지막으론 모델들의 성능에 따라 가중치를 부여해 예측한다. 사이킷런 패키지에서는 아다부스트의 기본 약 분류기로 깊이가 1 인 단순 의사결정나무를 사용한다.

정확도: 88%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	722	104
	Positive	120	854

ROC 커브 PR 커브



auc 점수: 0.95 ap 점수: 0.96

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	840	99.404762	1.820109	835	840	0.99	835	1.820109
2	840	98.571429	1.804850	828	1680	0.99	1663	1.812479
3	839	96.901073	1.774266	813	2519	0.98	2476	1.799752
4	840	92.261905	1.689322	775	3359	0.97	3251	1.772136
5	839	77.711561	1.422904	652	4198	0.93	3903	1.702340
6	840	45.000000	0.823953	378	5038	0.85	4281	1.555884
7	840	20.595238	0.377100	173	5878	0.76	4454	1.387429
8	839	10.607867	0.194231	89	6717	0.68	4543	1.238390
9	840	4.047619	0.074112	34	7557	0.61	4577	1.108975
10	840	1.071429	0.019618	9	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	100.000000	1.848049	180	180	1.00	180	1.848049
2	180	98.333333	1.817248	177	360	0.99	357	1.832649
3	180	96.111111	1.776181	173	540	0.98	530	1.813826
4	180	87.222222	1.611910	157	720	0.95	687	1.763347
5	180	77.777778	1.437372	140	900	0.92	827	1.698152
6	180	44.444444	0.821355	80	1080	0.84	907	1.552019
7	180	23.888889	0.441478	43	1260	0.75	950	1.393370
8	180	6.666667	0.123203	12	1440	0.67	962	1.234600
9	180	5.555556	0.102669	10	1620	0.60	972	1.108830
10	180	1.111111	0.020534	2	1800	0.54	974	1.000000

이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

8] 그래디언트 부스트

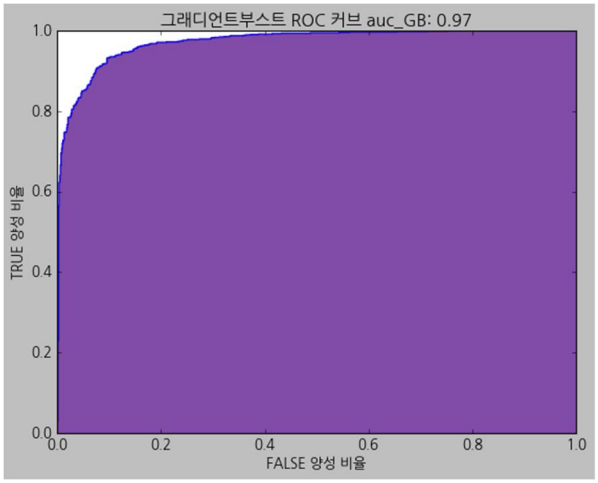
그래디언트 부스트모델도 아다부스트와 같은 개념에서 시작한다. 하나의 모델보다 여러 개 모델들의 조합이 더 낫다 가정하고 여러 개의 분류기들이 전 모델의 결과를 가지고 학습한다는 점이다.
아다부스트와 다른점은 아다부스트는 오분류에 대해 가중치를 줘서 후속 모델을 새로 학습시키는데 반해 그래디언트 부스트는 전 모델이 학습을 했을 때 생긴 잔차들을 예측할 수 있도록 새로운 분류기들을 학습시킨다.

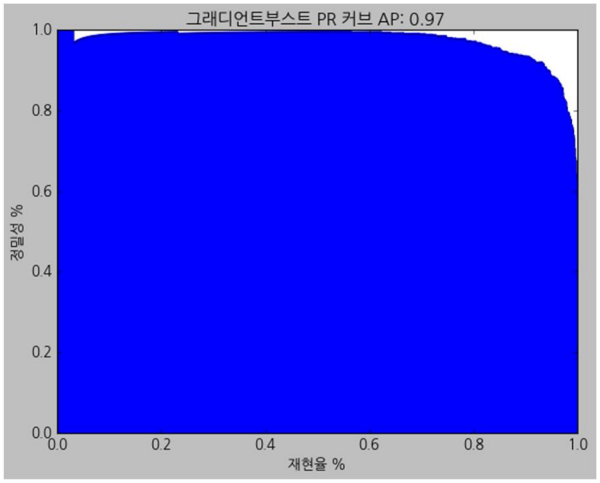
정확도: 92%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	753	73
	Positive	80	894

ROC 커브 PR 커브





auc 점수: 0.97 ap 점수:0.97

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	787	99.364676	1.819375	782	787	0.99	782	1.819375
2	893	99.776036	1.826907	891	1680	1.00	1673	1.823378
3	838	99.880668	1.828822	837	2518	1.00	2510	1.825190
4	841	97.265161	1.780932	818	3359	0.99	3328	1.814109
5	833	82.232893	1.505690	685	4192	0.96	4013	1.752823
6	846	48.345154	0.885203	409	5038	0.88	4422	1.607129
7	839	12.038141	0.220419	101	5877	0.77	4523	1.409162
8	834	5.755396	0.105382	48	6711	0.68	4571	1.247137
9	844	1.658768	0.030372	14	7555	0.61	4585	1.111207
10	842	0.118765	0.002175	1	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	163	99.386503	1.836712	162	163	0.99	162	1.836712
2	197	99.492386	1.838668	196	360	0.99	358	1.837782
3	180	100.000000	1.848049	180	540	1.00	538	1.841205
4	180	95.000000	1.755647	171	720	0.98	709	1.819815
5	178	76.404494	1.411993	136	898	0.94	845	1.738977
6	182	51.648352	0.954487	94	1080	0.87	939	1.606776
7	179	13.407821	0.247783	24	1259	0.76	963	1.413560
8	181	4.419890	0.081682	8	1440	0.67	971	1.246150
9	180	1.666667	0.030801	3	1620	0.60	974	1.111111
10	180	0.000000	0.000000	0	1800	0.54	974	1.000000

이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

9] 서포트 벡터 머신

두 카테고리 중 어느 하나에 속한 데이터의 집합이 주어졌을 때, SVM 알고리즘은 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할지 판단하는 비확률적 이진 선형 분류 모델을 만든다. 만들어진 분류 모델은 데이터가 사상 된 공간에서 경계로 표현되는데 SVM 알고리즘은 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘이다. 다른 모델들과의 가장 큰 차이점은 경계점의 데이터만이 결정곡선을 그리는데 영향을 준다는 것이다.

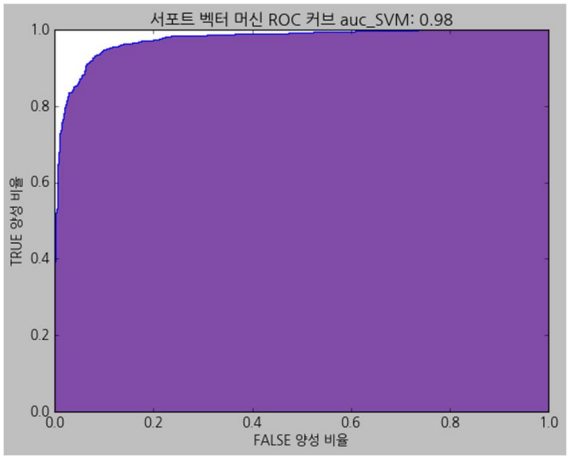
이 경우도 grid search 를 통한 하이퍼 파라미터 튜닝을 통해 오분류 패널티를 결정했는데 c=10 으로 설정하는게 제일 좋았다.

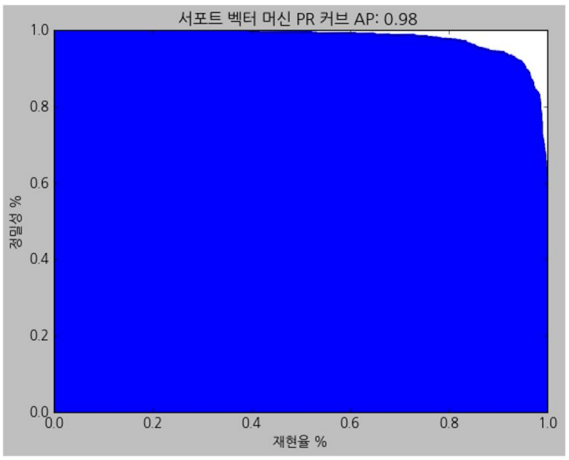
정확도: 92%

Confusion matrix:

		모델이 예측한 값	
		Negative	Positive
실제값	Negative	757	69
	Positive	66	908

ROC 커브 PR 커브





auc 점수: 0.98 ap 점수:0.98

Train 셋의 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	840	100.000000	1.831007	840	840	1.00	840	1.831007
2	840	99.880952	1.828828	839	1680	1.00	1679	1.829918
3	839	100.000000	1.831007	839	2519	1.00	2518	1.830281
4	840	98.928571	1.811389	831	3359	1.00	3349	1.825556
5	839	98.808105	1.809184	829	4198	1.00	4178	1.822284
6	840	45.000000	0.823953	378	5038	0.90	4556	1.655830
7	840	0.952381	0.017438	8	5878	0.78	4564	1.421694
8	839	2.383790	0.043647	20	6717	0.68	4584	1.249566
9	840	0.238095	0.004360	2	7557	0.61	4586	1.111155
10	840	0.000000	0.000000	0	8397	0.55	4586	1.000000

Validation 셋의 이익도표:

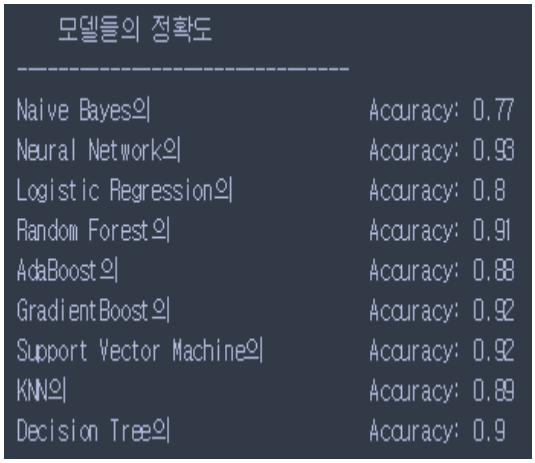
구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	100.000000	1.848049	180	180	1.00	180	1.848049
2	180	100.000000	1.848049	180	360	1.00	360	1.848049
3	180	97.777778	1.806982	176	540	0.99	536	1.834360
4	180	97.222222	1.796715	175	720	0.99	711	1.824949
5	180	80.000000	1.478439	144	900	0.95	855	1.755647
6	180	47.777778	0.882957	86	1080	0.87	941	1.610198
7	180	11.666667	0.215606	21	1260	0.76	962	1.410971
8	180	4.444444	0.082136	8	1440	0.67	970	1.244867
9	180	1.666667	0.030801	3	1620	0.60	973	1.109970
10	180	0.555556	0.010267	1	1800	0.54	974	1.000000

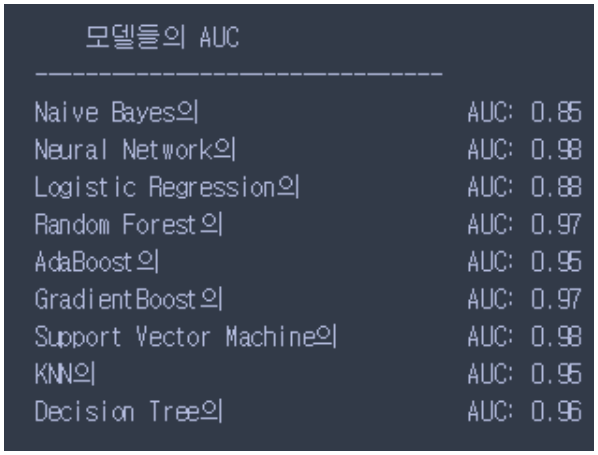
이익도표의 비교를 통해 과적합없이 제대로 학습된 걸 알 수 있다.

Ⅲ. 결론

1. 최종 모형 선택

최종 모형을 선택하기 위해 9 개 모델들의 검증셋에 대한 정확도, auc score, ap score 를 나열해보면

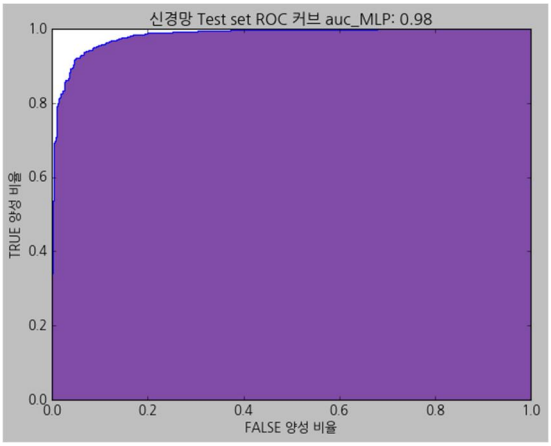


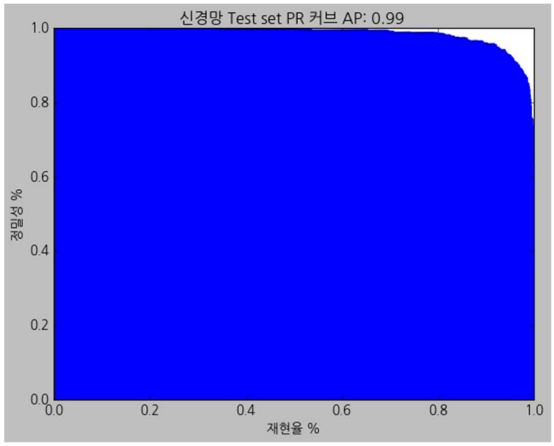


종합적으로 보았을 때 모든 지표에서 Neural Network 가 제일 성능이 좋은점을 알 수 있다. Neural Network 성능이 좋은 이유는 학습해야 할 파라미터 개수가 16* 128 + 128*2=2304 개로 엄청 많지만 다행히 우리의 관측치 개수가 충분해서 신경망 모델을 과적합없이 학습시킬 수 있었던 것 같다.
최종적으로 선택한 모델인 Neural Network 가 실제 어떤 성능을 보일까 확인 해보기 위해 마지막으로 test 셋에 대해 성능을 확인 해보면

[MLP 모델의 Test Set 성능]

정확도: 93%





AUC: 0.98 AP: 0.99

Test 셋 이익도표:

구간	구간 관측치 개수	구간 활성화 비율(%)	구간 LIFT	구간 실제 만족한 사람 수	누적 관측치 개수	누적 활성화 비율 %	누적 실제 만족한 사람수	누적 lift (%)
1	180	100.000000	1.867220	180	180	1.00	180	1.867220
2	180	99.444444	1.856846	179	360	1.00	359	1.862033
3	180	98.888889	1.846473	178	540	0.99	537	1.856846
4	180	97.222222	1.815353	175	720	0.99	712	1.846473
5	180	84.444444	1.576763	152	900	0.96	864	1.792531
6	180	45.000000	0.840249	81	1080	0.88	945	1.633817
7	180	7.222222	0.134855	13	1260	0.76	958	1.419680
8	180	2.222222	0.041494	4	1440	0.67	962	1.247407
9	180	0.555556	0.010373	1	1620	0.59	963	1.109959
10	180	0.555556	0.010373	1	1800	0.54	964	1.000000

이익도표를 확인하면 일단 구간리프트 값이 5 구간 6 구간사이에서 확실히 구분되어있는 것을 볼 수 있다. 4 번째 구간까지 잘못 예측한 관측치의 개수가 총 0+1+2+5=8 밖에 안됨을 알 수 있고 첫번째 구간에 들어가는 관측치는 모두 실제로 만족하는 관측치임을 볼 수 있다. 신경망 모델이 하위구간으로 나눈 마지막 두 구간에 대해선 오분류가 하나씩 밖에 없음을 알 수 있다.

2. 변수 해석

MLP 는 black box 모델이므로 변수들이 반응변수에 어떻게 관여하는지 해석이 불가능하다. 그래서 변수 해석을 위해 두가지 다른 모델을 활용하였다. Random forest 모델에서 변수가 사용된 빈도를 통해 변수 중요도를 평가하고 로지스틱 회귀모형의 계수를 통해 변수의 부호를 결정해 주었다.

<로지스틱 회귀모형의 계수>

	Columns	Coefficients	p_Value
0	Gender	0.552474	0.000
1	Customer Type	0.922497	0.000
2	Age	-0.033884	0.298
3	Type of Travel	0.439030	0.000
4	Flight Distance	-0.113213	0.001
5	Seat comfort	0.793209	0.000
6	Departure/Arrival time convenient	-0.510503	0.000
7	Gate location	-0.032541	0.362
8	Online support	0.416263	0.000
9	On-board service	0.539116	0.000
10	Baggage handling	0.258404	0.000
11	Checkin service	0.439569	0.000
12	Online boarding	0.387884	0.000
13	Departure Delay in Minutes	-0.145883	0.000
14	Class_Eco	-0.377584	0.000
15	Class_Eco_plus	-0.219845	0.000

<랜덤 포레스트 변수 중요도>

Features	Importance
Age	0.01
Gate location	0.01
Departure Delay in Minutes	0.01
Class_Eco_plus	0.01
Checkin service	0.02
Flight Distance	0.03
Departure/Arrival time convenient	0.03
Type of Travel	0.05
Baggage handling	0.05
Class_Eco	0.05
Gender	0.06
Online boarding	0.07
On-board service	0.08
Customer Type	0.11
Online support	0.18
Seat comfort	0.24

전반적으로 모든 변수들이 골고루 쓰인걸 볼 수 있지만 가장 중요한 5 가지 항목을 나열하면 Seat comfort, Online support, Customer Type, on-board service, Online boarding 이다. 모두 Logistic Regression 에서 양의 계수를 가진다. 중요도 순서로 나열해 설명해 보겠다.

1) Seat Comfort

좌석에 대한 만족도다. 로지스틱 회귀모형에서 계수가 0.793209 로 양수이므로 항공사에 대한 만족도는 전반적으로 의자가 편리할수록 항공 서비스에 만족한다 대답할 확률이 증가한다는 해석이 가능하다.

2) Online Support

Online Support 는 온라인으로 진행하는 질의응답이나 포로모션 등 온라인 고객지원 만족도이다. 온라인 고객지원도 로지스틱 회귀에서 계수가 양수이므로 전반적으로 온라인 고객지원 만족도가 증가하면 항공 서비스에 만족한다 대답할 확률이 증가한다는 해석이 가능하다.

3) Customer Type

Customer Type 은 고객이 Loyal 등급에 해당하는지 아닌지를 구분해주는 변수이다. 1 이면 Loyal 등급이고 0 이면 아니다. Loyal 등급의 고객들이 전반적으로 항공 서비스에 만족한다 대답할 확률이 높았다.

4) On-board Service

On-board Service 는 기내 서비스에 대한 만족도를 나타내는 지표이다. 기내 서비스에 대한 만족도가 높을수록 항공 서비스에 만족한다 대답할 확률이 높았다.

5) Online boarding

Online boarding 은 항공사의 온라인 예매에 대한 만족도이다. 온라인 예매가 편리하거나 서비스를 많이 제공할수록 항공 서비스에 만족한다 대답할 확률이 높았다.

이외에도 성별에 대해선 여성이 항공 서비스에 만족한다 대답할 확률이 높고 Eco class 를 타는 사람이 business class 를 활용하는 사람보다 만족할 확률이 높았다.