"

Airline Passensger Satisfication

비행 승객 만족도 분석 "

Airline Passenger Satisfication

이성공
김민서
박경희
서태원
장윤식

비행 승객 만족도 분석

Airline Passensger Satisfication

1. 프로젝트 주제소개

2. 데이터 소개

3. 데이터 전처리

4. 적용한 분석 기법 및 모델 소개

5. 향후 과제

1. 프로젝트 주제소개

〈 Project Topic 〉

머신러닝 모델과 딥러닝 모델의 성능비교

만족(또는 불만족)하는 승객과 높은 상관관계가 있는 요소는 무엇일까?

승객의 만족도를 예측할 수 있을까?

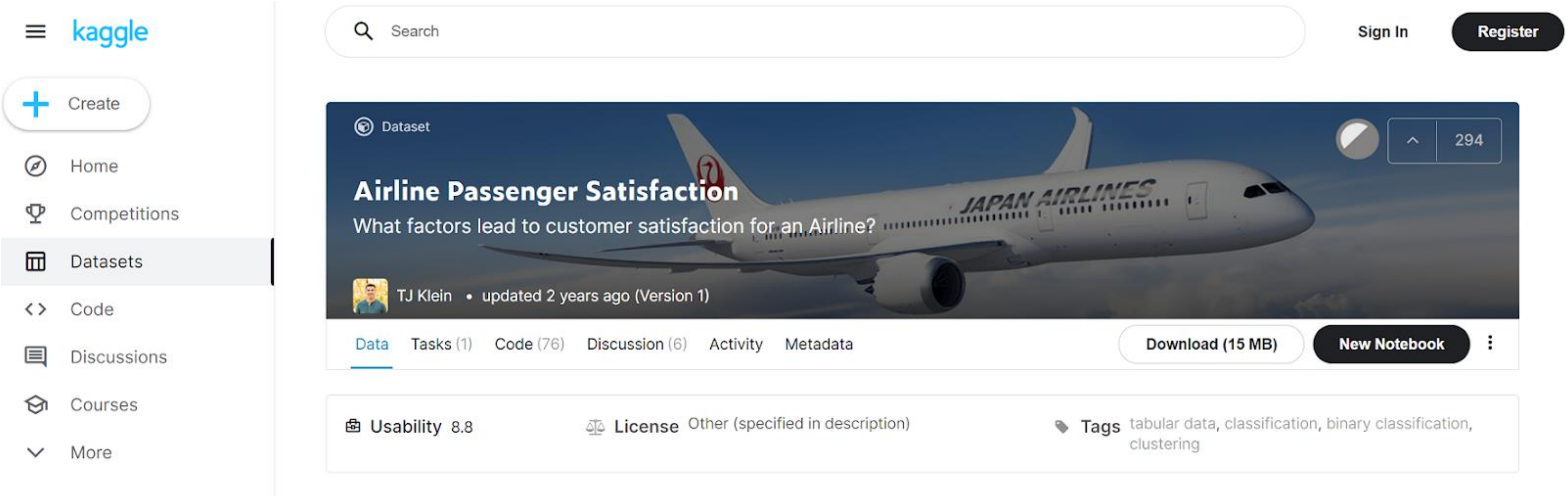
What factors are highly correlated to a satisfied (or dissatisfied) passenger?

Can you predict passenger satisfaction?

비행승객 만족도 분석

Airline Passensger Satisfication

2. 데이터 소개



Airline Passenger Satisfaction

What factors lead to customer satisfaction for an Airline?

TJ Klein • updated 2 years ago (Version 1)

Usability 8.8

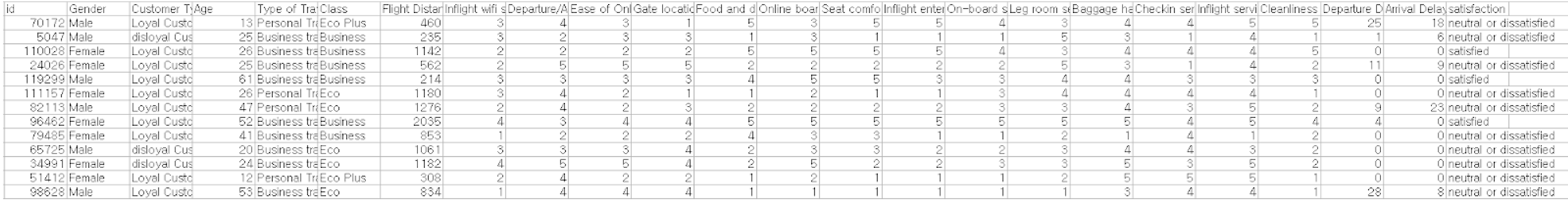
License Other (specified in description)

Tags tabular data, classification, binary classification, clustering

비행 승객 만족도 분석

Airline Passensger Satisfaction

3. 데이터 전처리

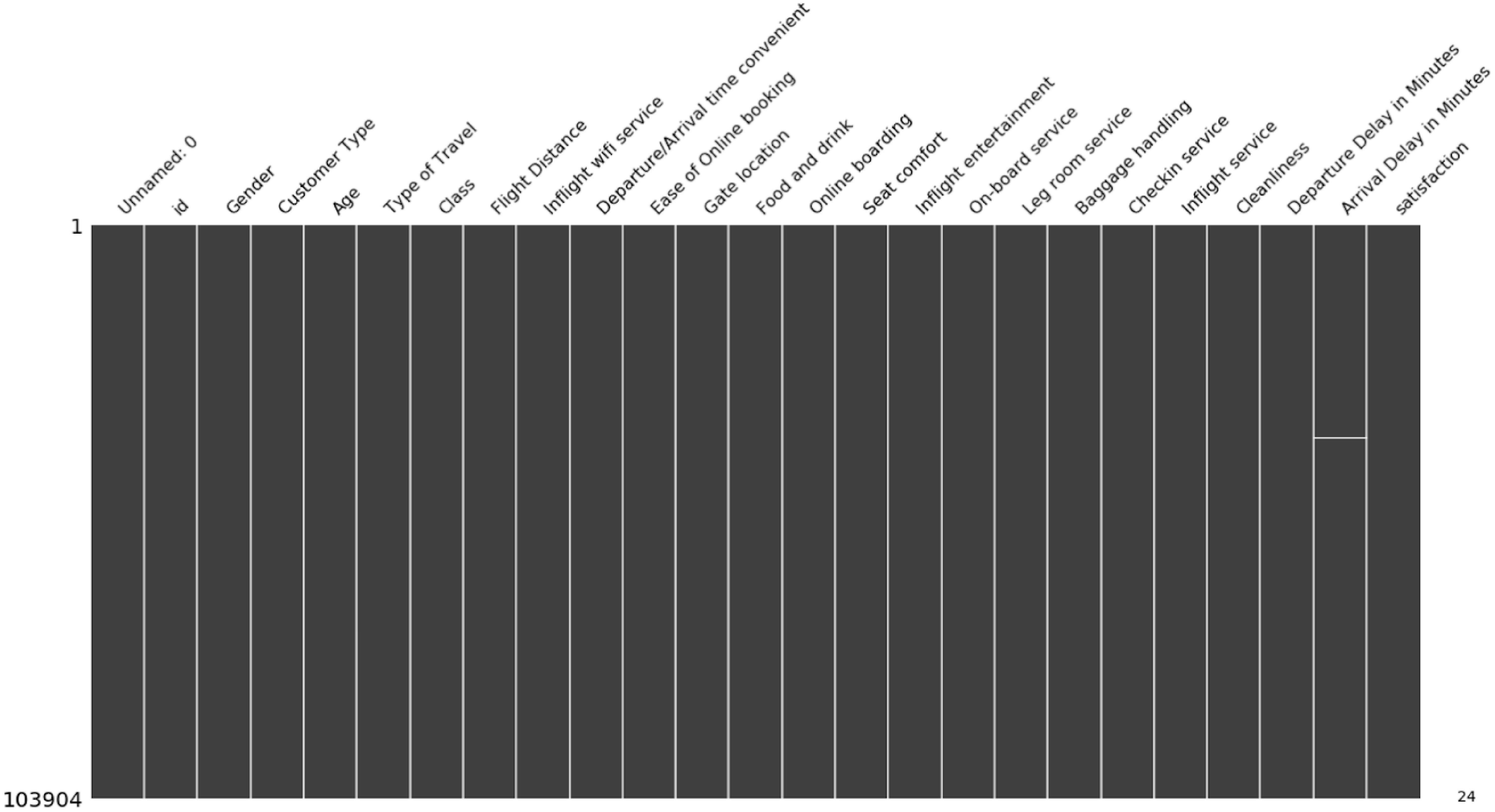


id	Gender	Customer Ty	Age	Type of Tra	Class	Flight Distar	Inflight wifi s	Departure/A	Ease of Onl	Gate locatio	Food and d	Online boar	Seat comfo	Inflight enter	On-board s	Leg room se	Baggage ha	Checkin ser	Inflight servi	Cleanliness	Departure D	Arrival Delay	satisfaction
70172	Male	Loyal Custo	13	Personal Tra	Eco Plus	460	3	4	3	1	5	3	5	5	4	3	4	4	5	5	25	18	neutral or dissatisfied
5047	Male	disloyal Cus	25	Business tra	Business	235	3	2	3	3	1	3	1	1	1	5	3	1	4	1	1	6	neutral or dissatisfied
110028	Female	Loyal Custo	26	Business tra	Business	1142	2	2	2	2	5	5	5	5	4	3	4	4	4	5	0	0	satisfied
24026	Female	Loyal Custo	25	Business tra	Business	562	2	5	5	5	2	2	2	2	2	5	3	1	4	2	11	9	neutral or dissatisfied
119299	Male	Loyal Custo	61	Business tra	Business	214	3	3	3	3	4	5	5	3	3	4	4	3	3	3	0	0	satisfied
111157	Female	Loyal Custo	26	Personal Tra	Eco	1180	3	4	2	1	1	2	1	1	3	4	4	4	4	1	0	0	neutral or dissatisfied
82113	Male	Loyal Custo	47	Personal Tra	Eco	1276	2	4	2	3	2	2	2	2	3	3	4	3	5	2	9	23	neutral or dissatisfied
96462	Female	Loyal Custo	52	Business tra	Business	2035	4	3	4	4	5	5	5	5	5	5	5	4	5	4	4	0	satisfied
79485	Female	Loyal Custo	41	Business tra	Business	853	1	2	2	2	4	3	3	1	1	2	1	4	1	2	0	0	neutral or dissatisfied
65725	Male	disloyal Cus	20	Business tra	Eco	1061	3	3	3	4	2	3	3	2	2	3	4	4	3	2	0	0	neutral or dissatisfied
34991	Female	disloyal Cus	24	Business tra	Eco	1182	4	5	5	4	2	5	2	2	3	3	5	3	5	2	0	0	neutral or dissatisfied
51412	Female	Loyal Custo	12	Personal Tra	Eco Plus	308	2	4	2	2	1	2	1	1	1	2	5	5	5	1	0	0	neutral or dissatisfied
98628	Male	Loyal Custo	53	Business tra	Eco	834	1	4	4	4	1	1	1	1	1	1	3	4	4	1	28	8	neutral or dissatisfied

비행승객 만족도 분석

Airline Passensg
Satisfication

3. 데이터 전처리

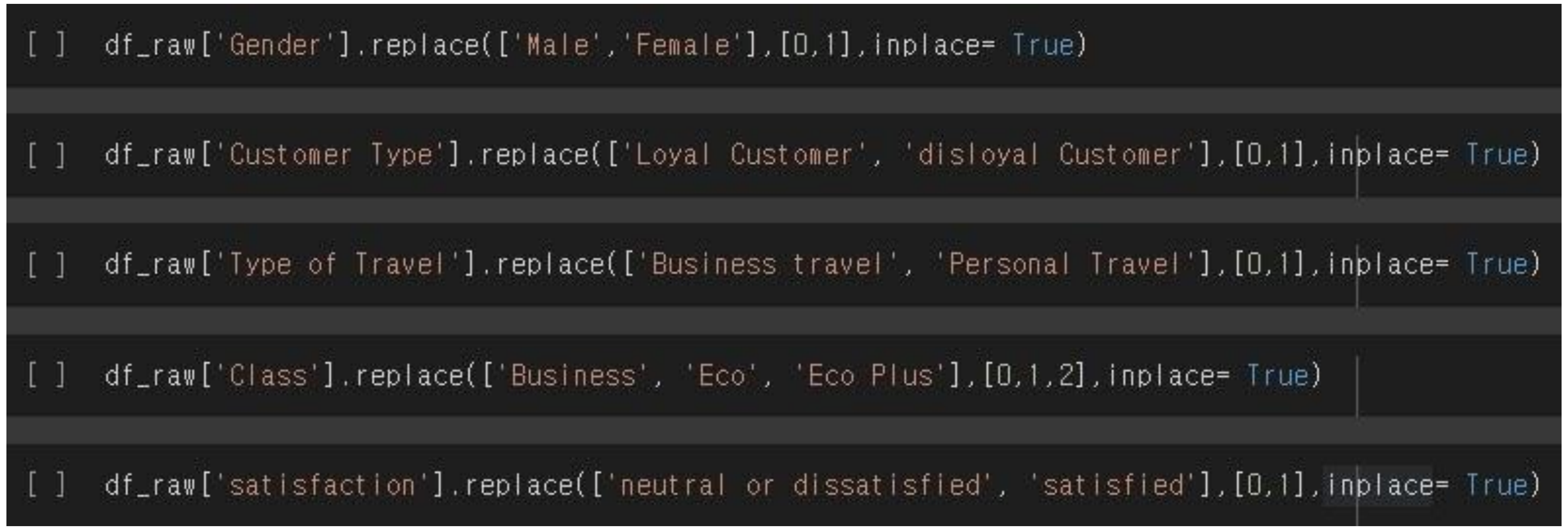


약 10만행 , 24열

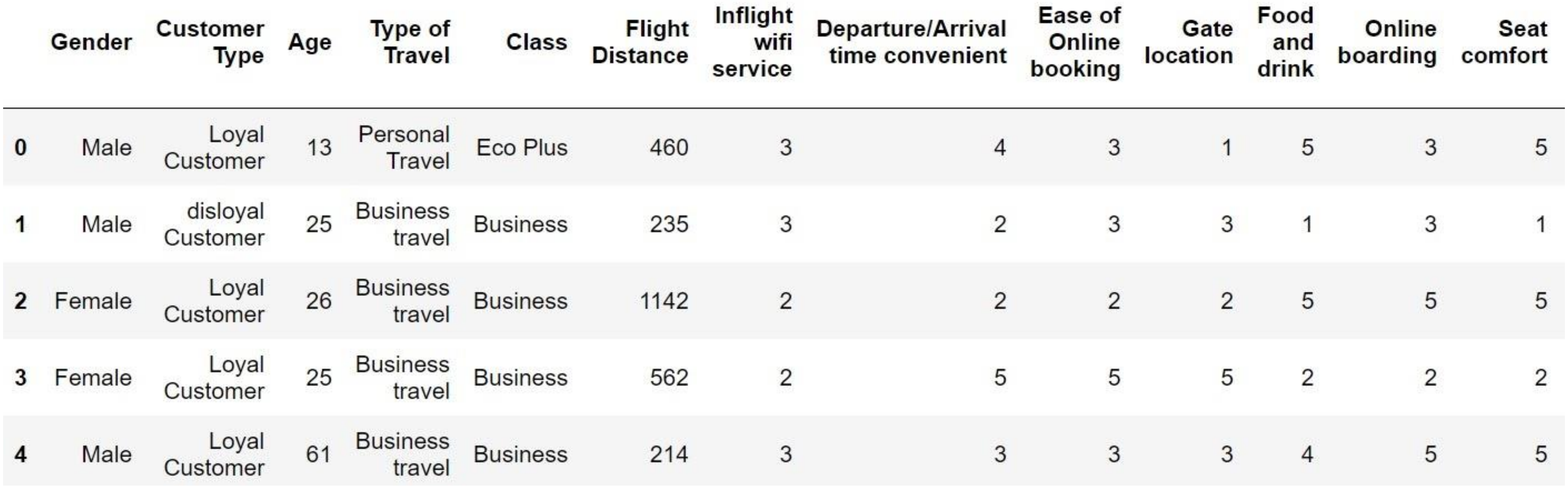
비행승객 만족도 분석

Airline Passensger Satisfication

3. 데이터 전처리



```
[ ]  df_raw['Gender'].replace(['Male','Female'],[0,1],inplace= True)

[ ]  df_raw['Customer Type'].replace(['Loyal Customer', 'disloyal Customer'],[0,1],inplace= True)

[ ]  df_raw['Type of Travel'].replace(['Business travel', 'Personal Travel'],[0,1],inplace= True)

[ ]  df_raw['Class'].replace(['Business', 'Eco', 'Eco Plus'],[0,1,2],inplace= True)

[ ]  df_raw['satisfaction'].replace(['neutral or dissatisfied', 'satisfied'],[0,1],inplace= True)
```



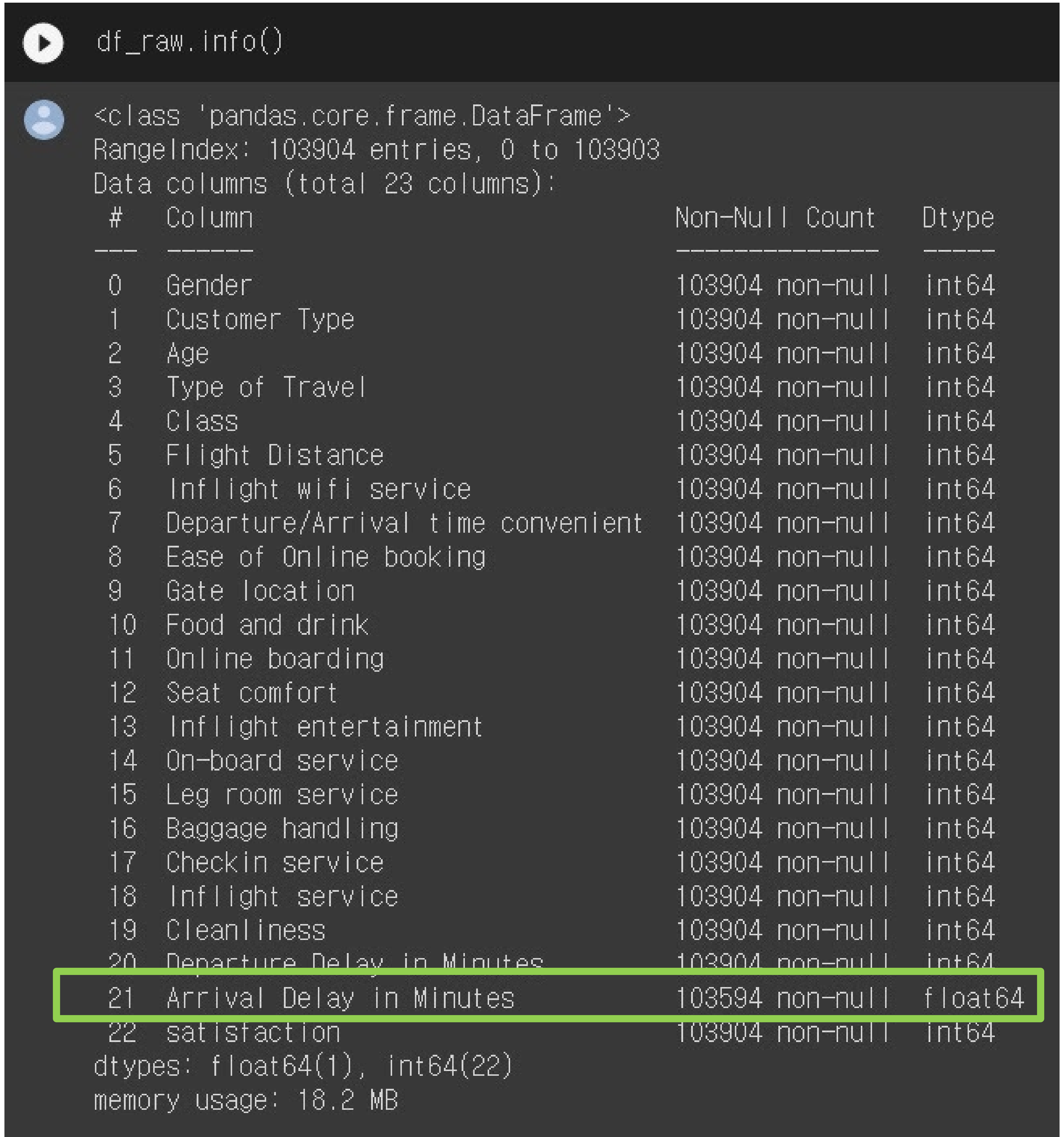
	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	Ease of Online booking	Gate location	Food and drink	Online boarding	Seat comfort
0	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	3	1	5	3	5
1	Male	disloyal Customer	25	Business travel	Business	235	3	2	3	3	1	3	1
2	Female	Loyal Customer	26	Business travel	Business	1142	2	2	2	2	5	5	5
3	Female	Loyal Customer	25	Business travel	Business	562	2	5	5	5	2	2	2
4	Male	Loyal Customer	61	Business travel	Business	214	3	3	3	3	4	5	5

비행승객 만족도 분석

Airline Passensger Satisfication

3. 데이터 전처리

Arrival Delay in Minutes 쪽 결측치 확인



```
df_raw.info()
```

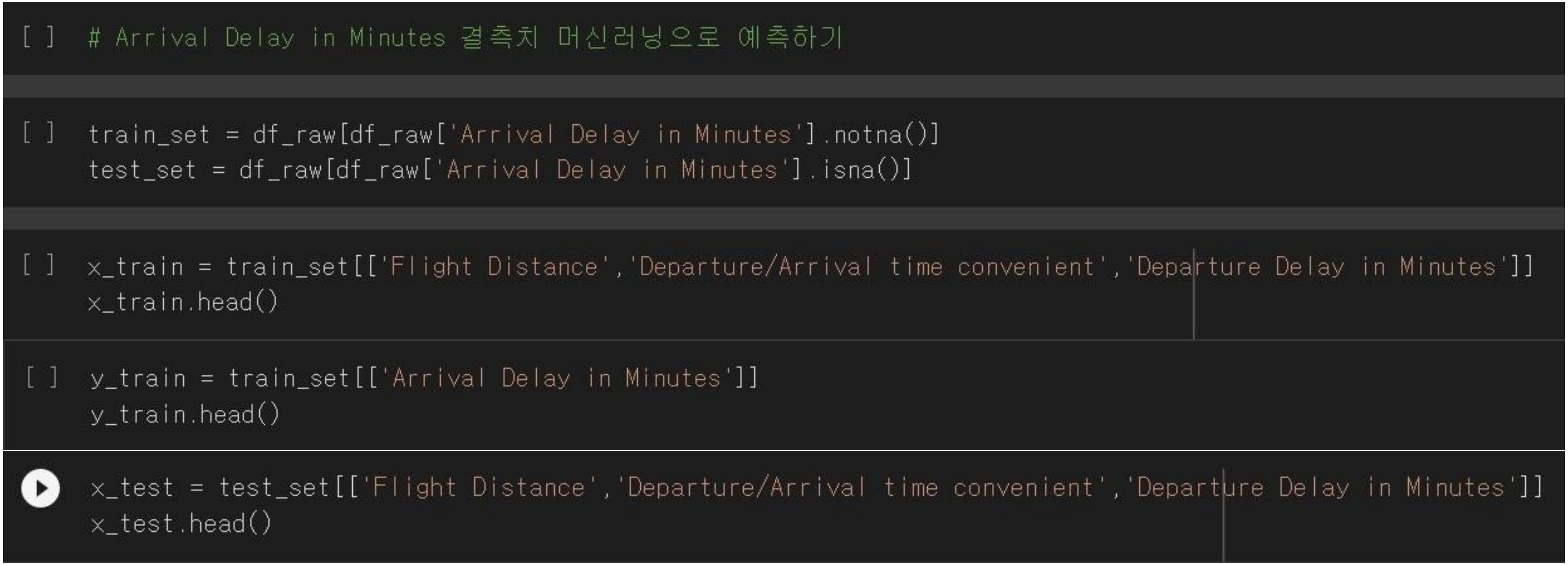
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 23 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Gender                             103904 non-null  int64
 1   Customer Type                      103904 non-null  int64
 2   Age                                103904 non-null  int64
 3   Type of Travel                     103904 non-null  int64
 4   Class                              103904 non-null  int64
 5   Flight Distance                    103904 non-null  int64
 6   Inflight wifi service              103904 non-null  int64
 7   Departure/Arrival time convenient  103904 non-null  int64
 8   Ease of Online booking             103904 non-null  int64
 9   Gate location                      103904 non-null  int64
 10  Food and drink                     103904 non-null  int64
 11  Online boarding                    103904 non-null  int64
 12  Seat comfort                       103904 non-null  int64
 13  Inflight entertainment             103904 non-null  int64
 14  On-board service                   103904 non-null  int64
 15  Leg room service                   103904 non-null  int64
 16  Baggage handling                   103904 non-null  int64
 17  Checkin service                    103904 non-null  int64
 18  Inflight service                   103904 non-null  int64
 19  Cleanliness                        103904 non-null  int64
 20  Departure Delay in Minutes         103904 non-null  int64
 21  Arrival Delay in Minutes           103594 non-null  float64
 22  satisfaction                       103904 non-null  int64
dtypes: float64(1), int64(22)
memory usage: 18.2 MB
```

비행승객 만족도 분석

Airline Passensger Satisfaction

3. 데이터 전처리

결측치 있는 것과 없는 것으로 나누고 x_train,x_test는 결측치 열과 관계성이 높은 열만 가져옴



```
# Arrival Delay in Minutes 결측치 머신러닝으로 예측하기
```

```
train_set = df_raw[df_raw['Arrival Delay in Minutes'].notna()]
test_set = df_raw[df_raw['Arrival Delay in Minutes'].isna()]
```

```
x_train = train_set[['Flight Distance','Departure/Arrival time convenient','Departure Delay in Minutes']]
x_train.head()
```

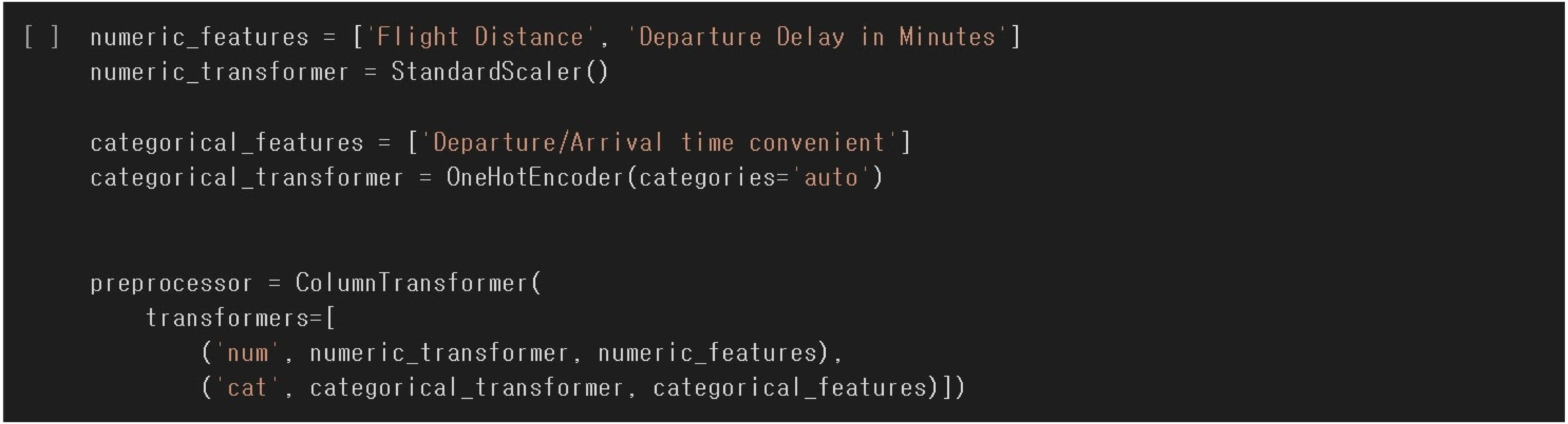
```
y_train = train_set[['Arrival Delay in Minutes']]
y_train.head()
```

```
x_test = test_set[['Flight Distance','Departure/Arrival time convenient','Departure Delay in Minutes']]
x_test.head()
```

비행승객 만족도 분석

Airline Passensger Satisfication

3. 데이터 전처리



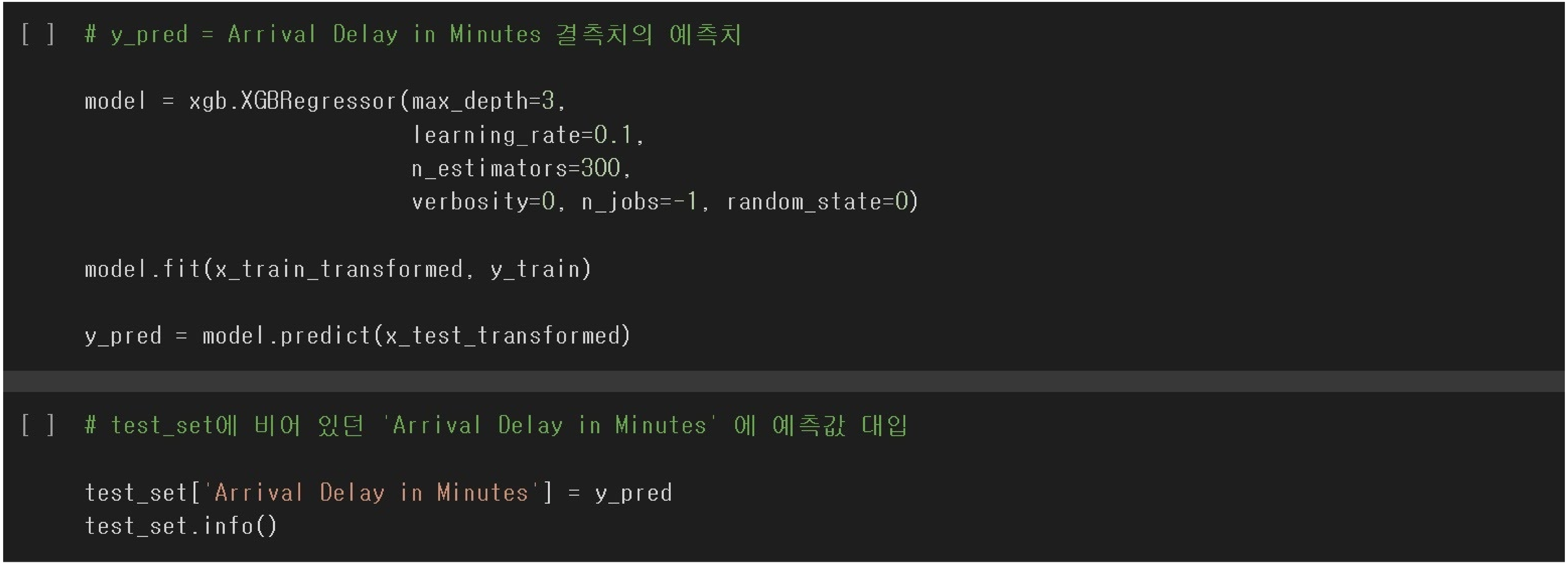
```
[ ]  numeric_features = ['Flight Distance', 'Departure Delay in Minutes']
     numeric_transformer = StandardScaler()

     categorical_features = ['Departure/Arrival time convenient']
     categorical_transformer = OneHotEncoder(categories='auto')


     preprocessor = ColumnTransformer(
         transformers=[
             ('num', numeric_transformer, numeric_features),
             ('cat', categorical_transformer, categorical_features)])
```

파이프라인 형성 / numeric 이랑 categorical 로 나눈 근거

비행승객 만족도 분석

Airline Passensger Satisfaction

3. 데이터 전처리



```
[ ]  # y_pred = Arrival Delay in Minutes 결측치의 예측치

     model = xgb.XGBRegressor(max_depth=3,
                              learning_rate=0.1,
                              n_estimators=300,
                              verbosity=0, n_jobs=-1, random_state=0)

     model.fit(x_train_transformed, y_train)

     y_pred = model.predict(x_test_transformed)
```

```
[ ]  # test_set에 비어 있던 'Arrival Delay in Minutes' 에 예측값 대입

     test_set['Arrival Delay in Minutes'] = y_pred
     test_set.info()
```

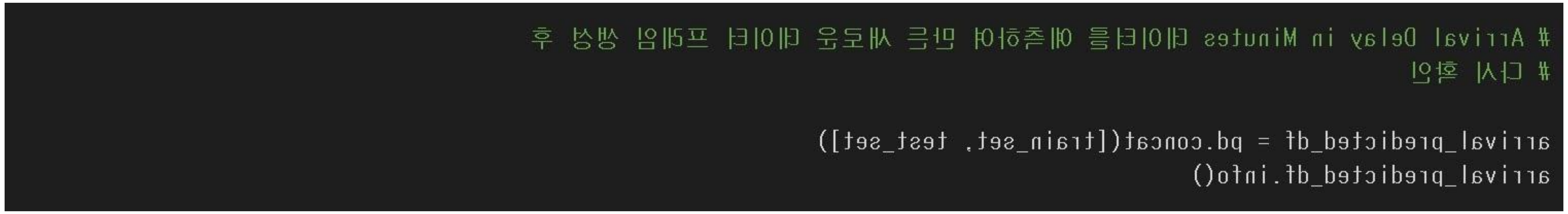
y_pred가 예측값을 대입한 변수라는 걸 설명 + 밑에 변수명 코드까지 (발표자) / y_pred 강조부탁(편집자)

비행 승객 만족도 분석

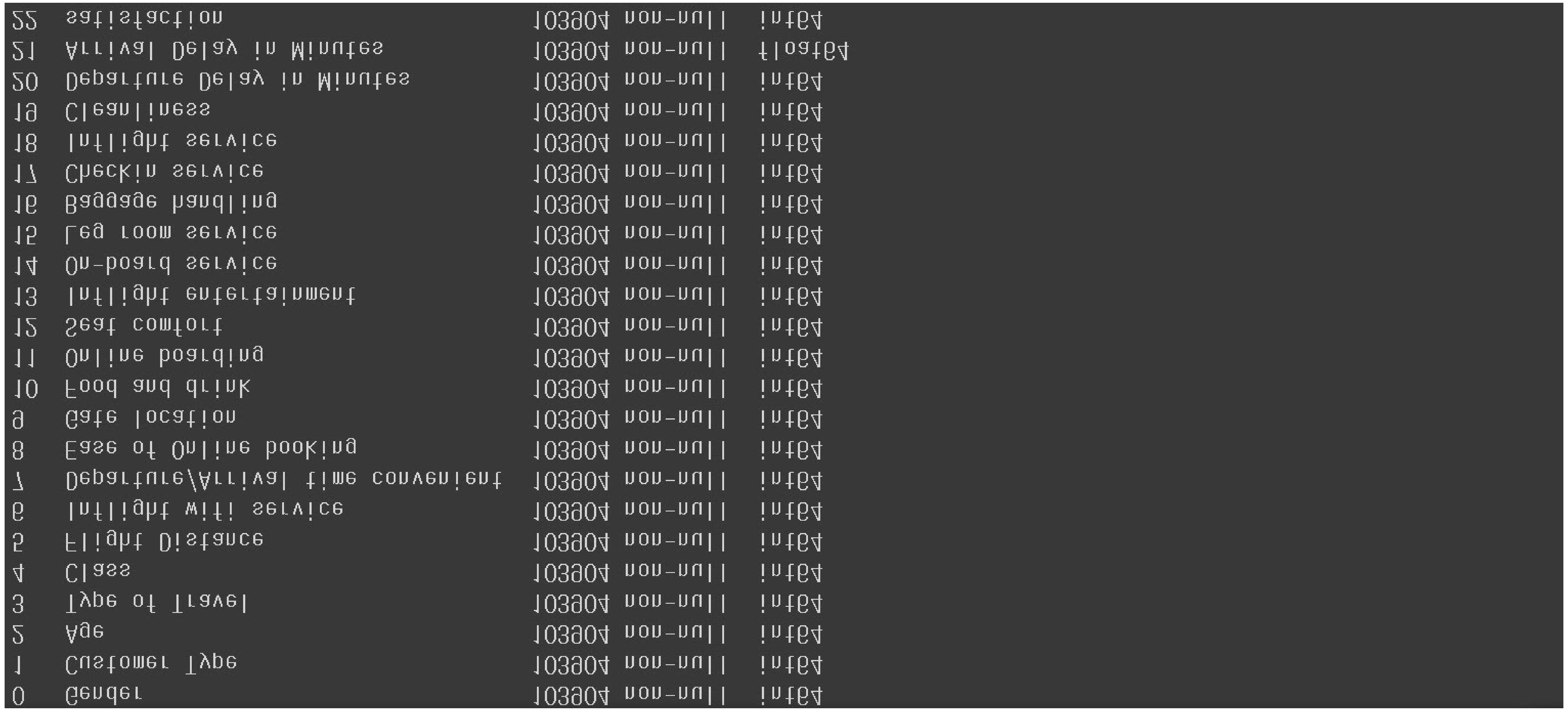
Airline Passensger Satisfication

3. 데이터 전처리

Arrival Delay 열이 결측치가 없음을 강조



```
# Arrival Delay in Minutes 데이터를 예측하여 만든 새로운 데이터 프레임 생성 후
# 다시 확인

arrival_predicted_df = pd.concat([train_set, test_set])
arrival_predicted_df.info()
```



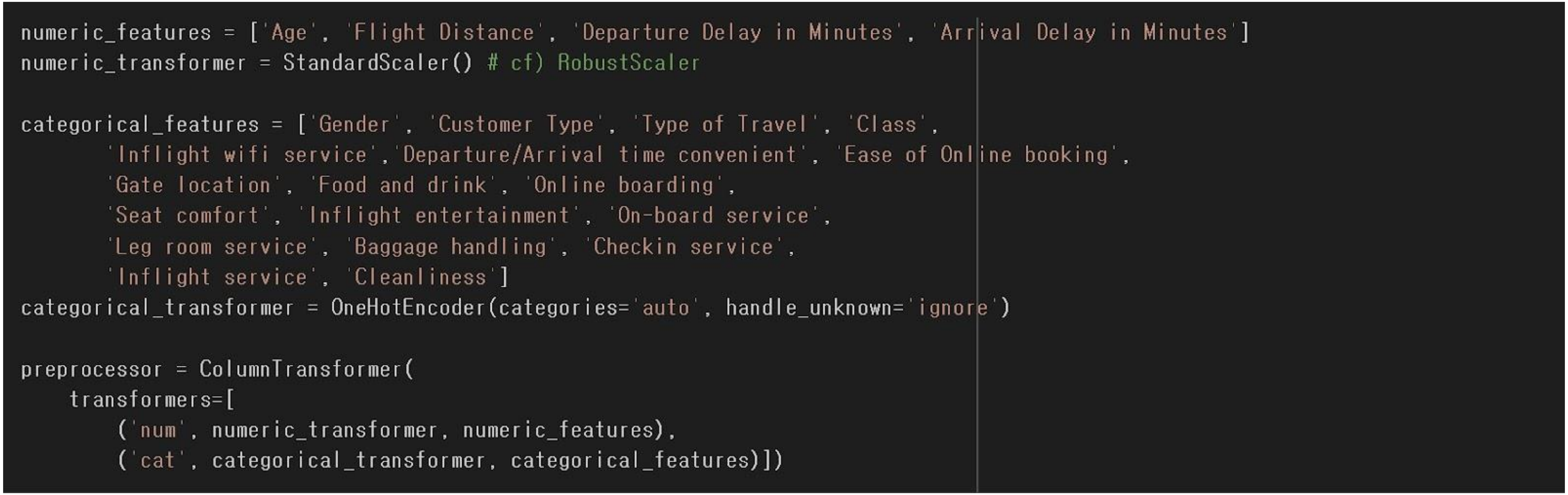
```
0   Gender                             103904 non-null  int64
1   Customer Type                      103904 non-null  int64
2   Age                                103904 non-null  int64
3   Type of Travel                     103904 non-null  int64
4   Class                              103904 non-null  int64
5   Flight Distance                    103904 non-null  int64
6   Inflight wifi service              103904 non-null  int64
7   Departure/Arrival time convenient  103904 non-null  int64
8   Ease of Online booking             103904 non-null  int64
9   Gate location                      103904 non-null  int64
10  Food and drink                     103904 non-null  int64
11  Online boarding                    103904 non-null  int64
12  Seat comfort                       103904 non-null  int64
13  Inflight entertainment             103904 non-null  int64
14  On-board service                   103904 non-null  int64
15  Leg room service                   103904 non-null  int64
16  Baggage handling                   103904 non-null  int64
17  Checkin service                    103904 non-null  int64
18  Inflight service                   103904 non-null  int64
19  Cleanliness                        103904 non-null  int64
20  Departure Delay in Minutes         103904 non-null  int64
21  Arrival Delay in Minutes           103904 non-null  float64
22  satisfaction                       103904 non-null  int64
```

비행 승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개 (머신러닝)

결측치 채운 걸 파이프라인 형성



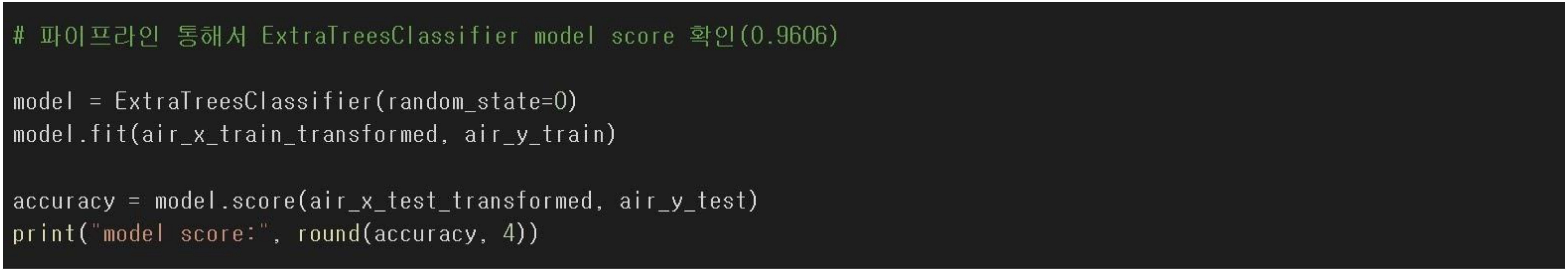
```
numeric_features = ['Age', 'Flight Distance', 'Departure Delay in Minutes', 'Arrival Delay in Minutes']
numeric_transformer = StandardScaler() # cf) RobustScaler

categorical_features = ['Gender', 'Customer Type', 'Type of Travel', 'Class',
       'Inflight wifi service','Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding',
       'Seat comfort', 'Inflight entertainment', 'On-board service',
       'Leg room service', 'Baggage handling', 'Checkin service',
       'Inflight service', 'Cleanliness']
categorical_transformer = OneHotEncoder(categories='auto', handle_unknown='ignore')

preprocessor = ColumnTransformer(
    transformers=[
        ('num', numeric_transformer, numeric_features),
        ('cat', categorical_transformer, categorical_features)])
```

비행승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개 (머신러닝)

파이캐럿 쓰기전에 사용한 4가지 머신러닝 모델 중 가장 높은 것만 찾아본 것 이후 파이캐럿

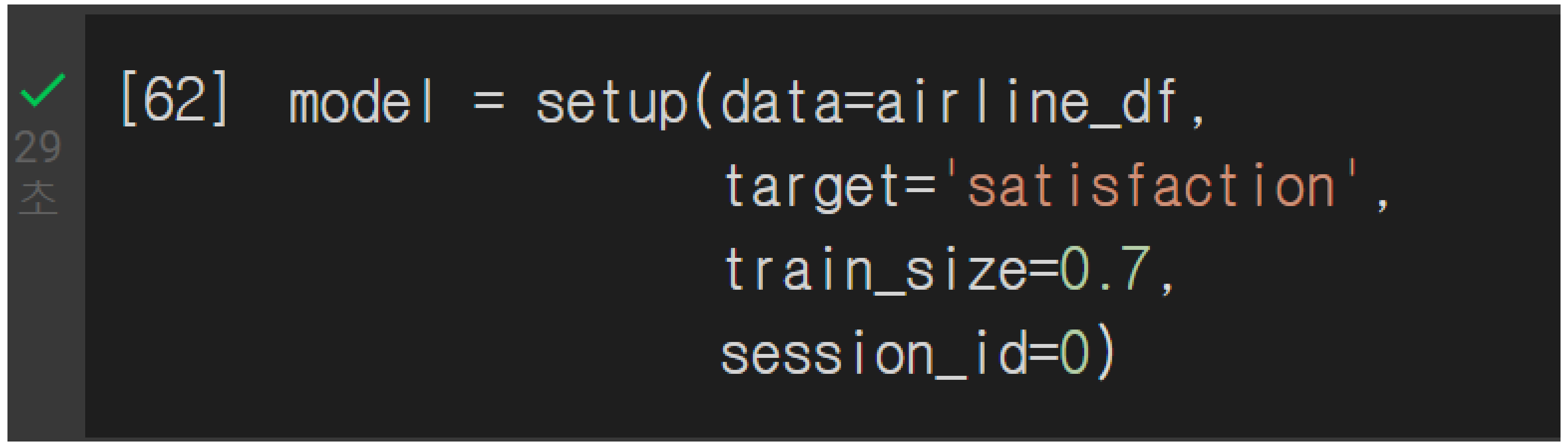


```
# 파이프라인 통해서 ExtraTreesClassifier model score 확인(0.9606)

model = ExtraTreesClassifier(random_state=0)
model.fit(air_x_train_transformed, air_y_train)

accuracy = model.score(air_x_test_transformed, air_y_test)
print("model score:", round(accuracy, 4))
```

비행승객 만족도 분석

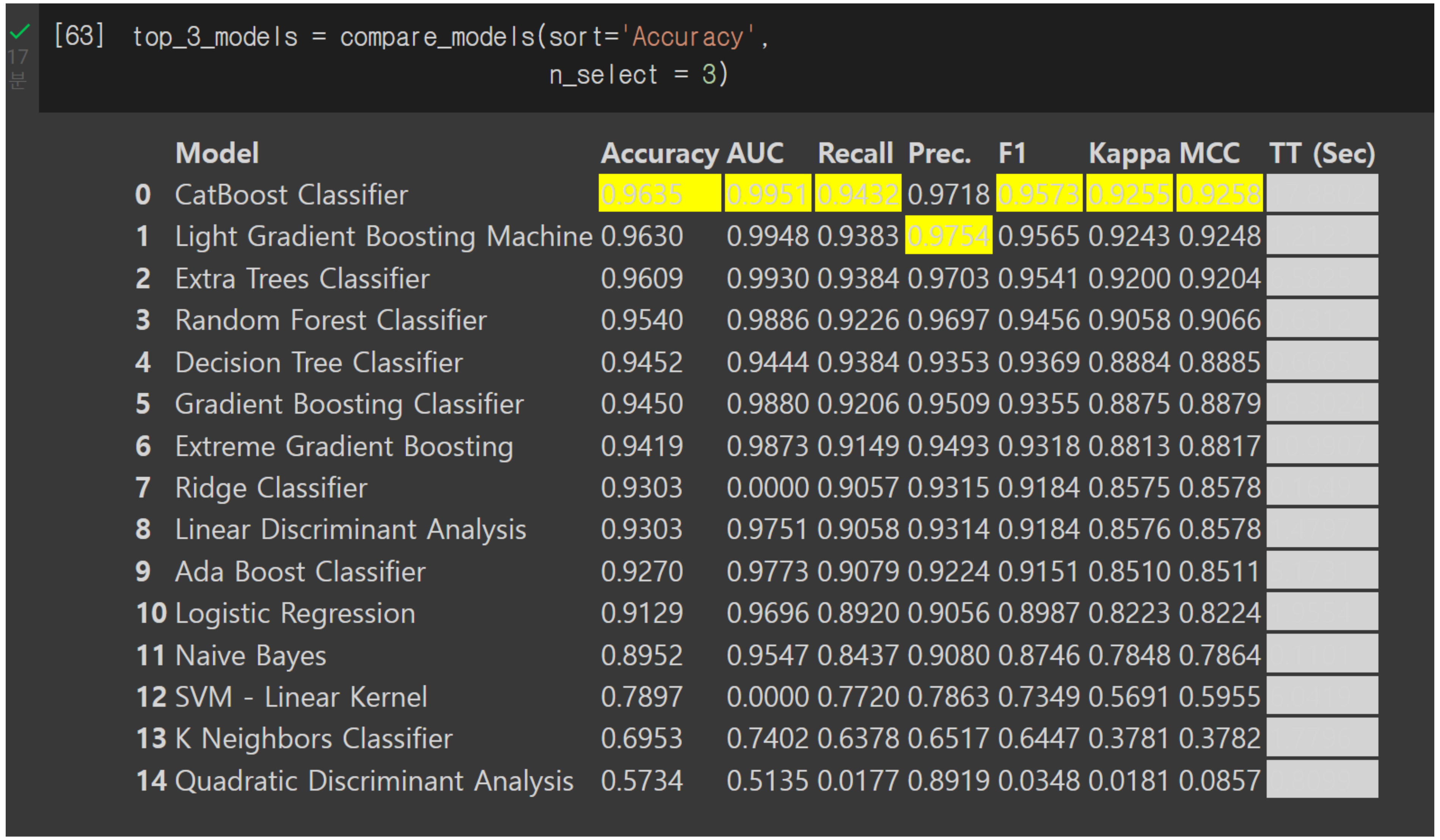
Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개 (머신러닝->파이캐럿)



```
[62]  model = setup(data=airline_df,
                    target='satisfaction',
                    train_size=0.7,
                    session_id=0)
```

모델 setup해주기



```
[63]  top_3_models = compare_models(sort='Accuracy',
                                    n_select = 3)
```

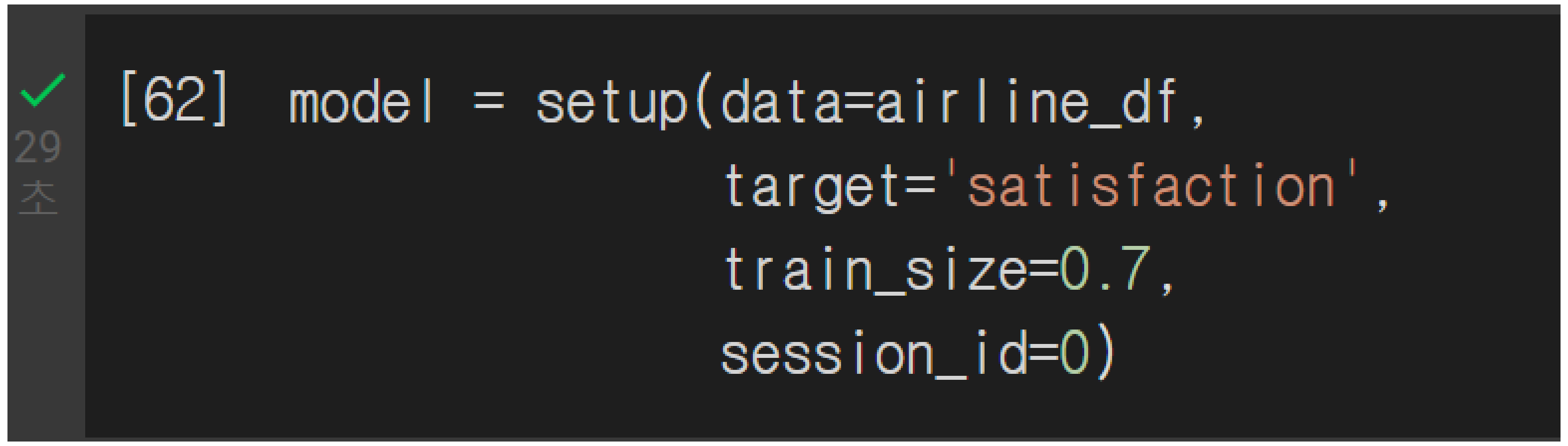
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	CatBoost Classifier	0.9635	0.9951	0.9432	0.9718	0.9573	0.9255	0.9258	17.8802
1	Light Gradient Boosting Machine	0.9630	0.9948	0.9383	0.9754	0.9565	0.9243	0.9248	1.2123
2	Extra Trees Classifier	0.9609	0.9930	0.9384	0.9703	0.9541	0.9200	0.9204	6.5825
3	Random Forest Classifier	0.9540	0.9886	0.9226	0.9697	0.9456	0.9058	0.9066	0.6312
4	Decision Tree Classifier	0.9452	0.9444	0.9384	0.9353	0.9369	0.8884	0.8885	0.6665
5	Gradient Boosting Classifier	0.9450	0.9880	0.9206	0.9509	0.9355	0.8875	0.8879	18.3024
6	Extreme Gradient Boosting	0.9419	0.9873	0.9149	0.9493	0.9318	0.8813	0.8817	10.9907
7	Ridge Classifier	0.9303	0.0000	0.9057	0.9315	0.9184	0.8575	0.8578	0.1649
8	Linear Discriminant Analysis	0.9303	0.9751	0.9058	0.9314	0.9184	0.8576	0.8578	1.4797
9	Ada Boost Classifier	0.9270	0.9773	0.9079	0.9224	0.9151	0.8510	0.8511	5.1731
10	Logistic Regression	0.9129	0.9696	0.8920	0.9056	0.8987	0.8223	0.8224	1.9554
11	Naive Bayes	0.8952	0.9547	0.8437	0.9080	0.8746	0.7848	0.7864	0.1101
12	SVM - Linear Kernel	0.7897	0.0000	0.7720	0.7863	0.7349	0.5691	0.5955	6.0479
13	K Neighbors Classifier	0.6953	0.7402	0.6378	0.6517	0.6447	0.3781	0.3782	1.7796
14	Quadratic Discriminant Analysis	0.5734	0.5135	0.0177	0.8919	0.0348	0.0181	0.0857	0.3059

모델간 성능비교를 위한 Pycaret 돌리기

비행승객 만족도 분석

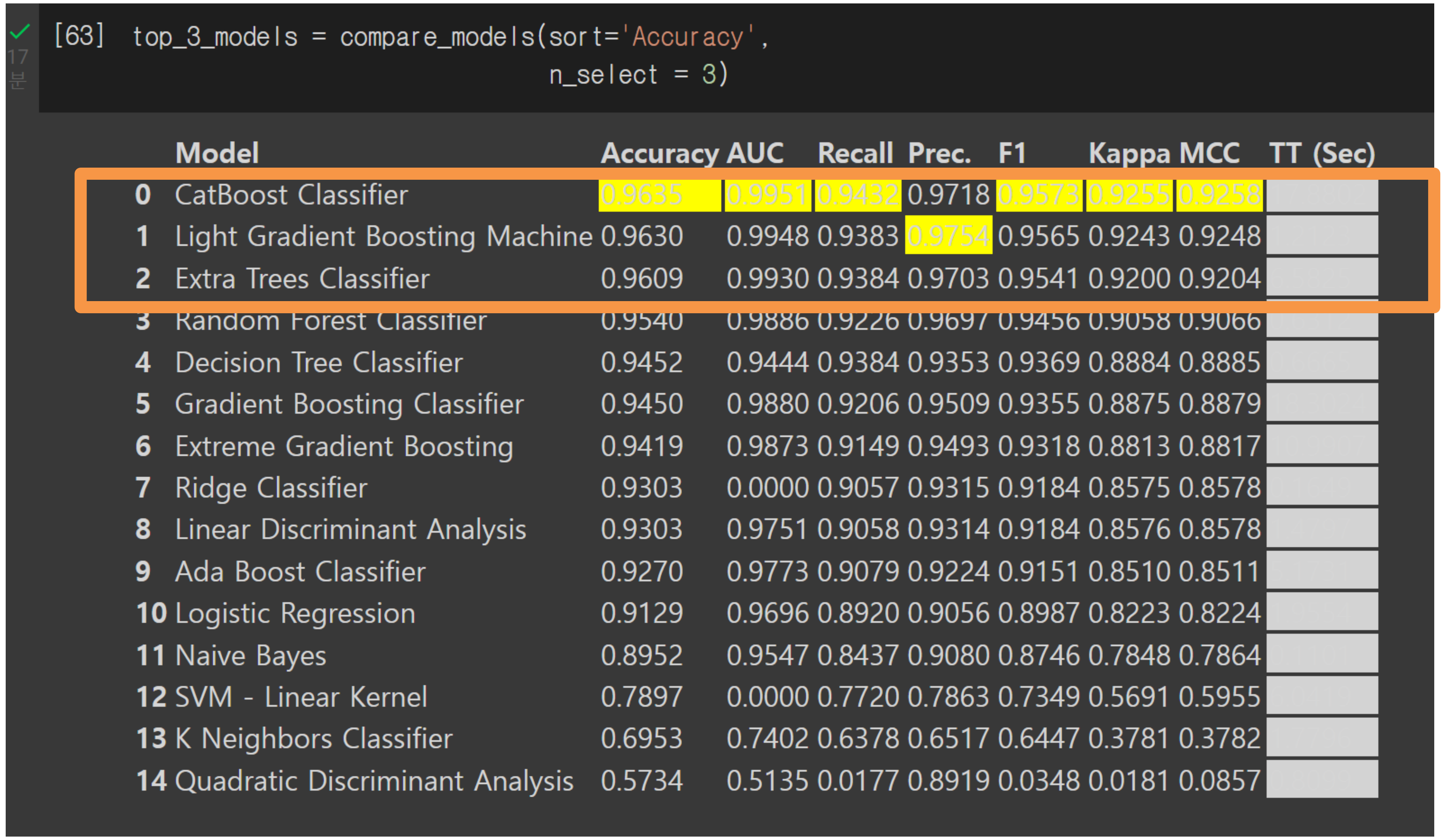
Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개 (머신러닝->파이캐럿)



```
[62]  model = setup(data=airline_df,
                    target='satisfaction',
                    train_size=0.7,
                    session_id=0)
```

모델 setup해주기



```
[63]  top_3_models = compare_models(sort='Accuracy',
                                    n_select = 3)
```

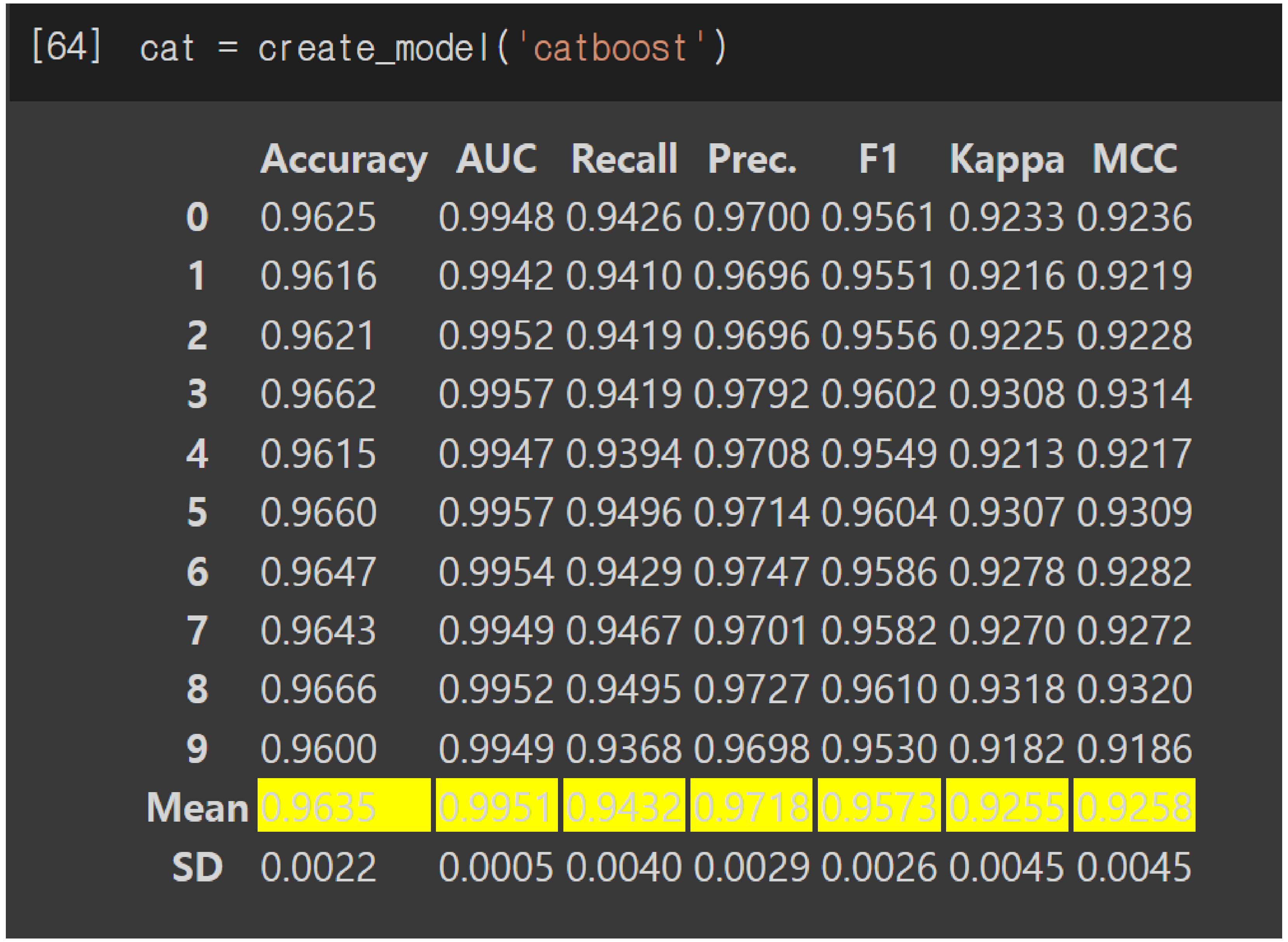
	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
0	CatBoost Classifier	0.9635	0.9951	0.9432	0.9718	0.9573	0.9255	0.9258	
1	Light Gradient Boosting Machine	0.9630	0.9948	0.9383	0.9754	0.9565	0.9243	0.9248	
2	Extra Trees Classifier	0.9609	0.9930	0.9384	0.9703	0.9541	0.9200	0.9204	
3	Random Forest Classifier	0.9540	0.9886	0.9226	0.9697	0.9456	0.9058	0.9066	
4	Decision Tree Classifier	0.9452	0.9444	0.9384	0.9353	0.9369	0.8884	0.8885	
5	Gradient Boosting Classifier	0.9450	0.9880	0.9206	0.9509	0.9355	0.8875	0.8879	
6	Extreme Gradient Boosting	0.9419	0.9873	0.9149	0.9493	0.9318	0.8813	0.8817	
7	Ridge Classifier	0.9303	0.0000	0.9057	0.9315	0.9184	0.8575	0.8578	
8	Linear Discriminant Analysis	0.9303	0.9751	0.9058	0.9314	0.9184	0.8576	0.8578	
9	Ada Boost Classifier	0.9270	0.9773	0.9079	0.9224	0.9151	0.8510	0.8511	
10	Logistic Regression	0.9129	0.9696	0.8920	0.9056	0.8987	0.8223	0.8224	
11	Naive Bayes	0.8952	0.9547	0.8437	0.9080	0.8746	0.7848	0.7864	
12	SVM - Linear Kernel	0.7897	0.0000	0.7720	0.7863	0.7349	0.5691	0.5955	
13	K Neighbors Classifier	0.6953	0.7402	0.6378	0.6517	0.6447	0.3781	0.3782	
14	Quadratic Discriminant Analysis	0.5734	0.5135	0.0177	0.8919	0.0348	0.0181	0.0857	

모델간 성능비교를 위한 Pycaret 돌리기

비행승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개



```
[64]  cat = create_model('catboost')
```

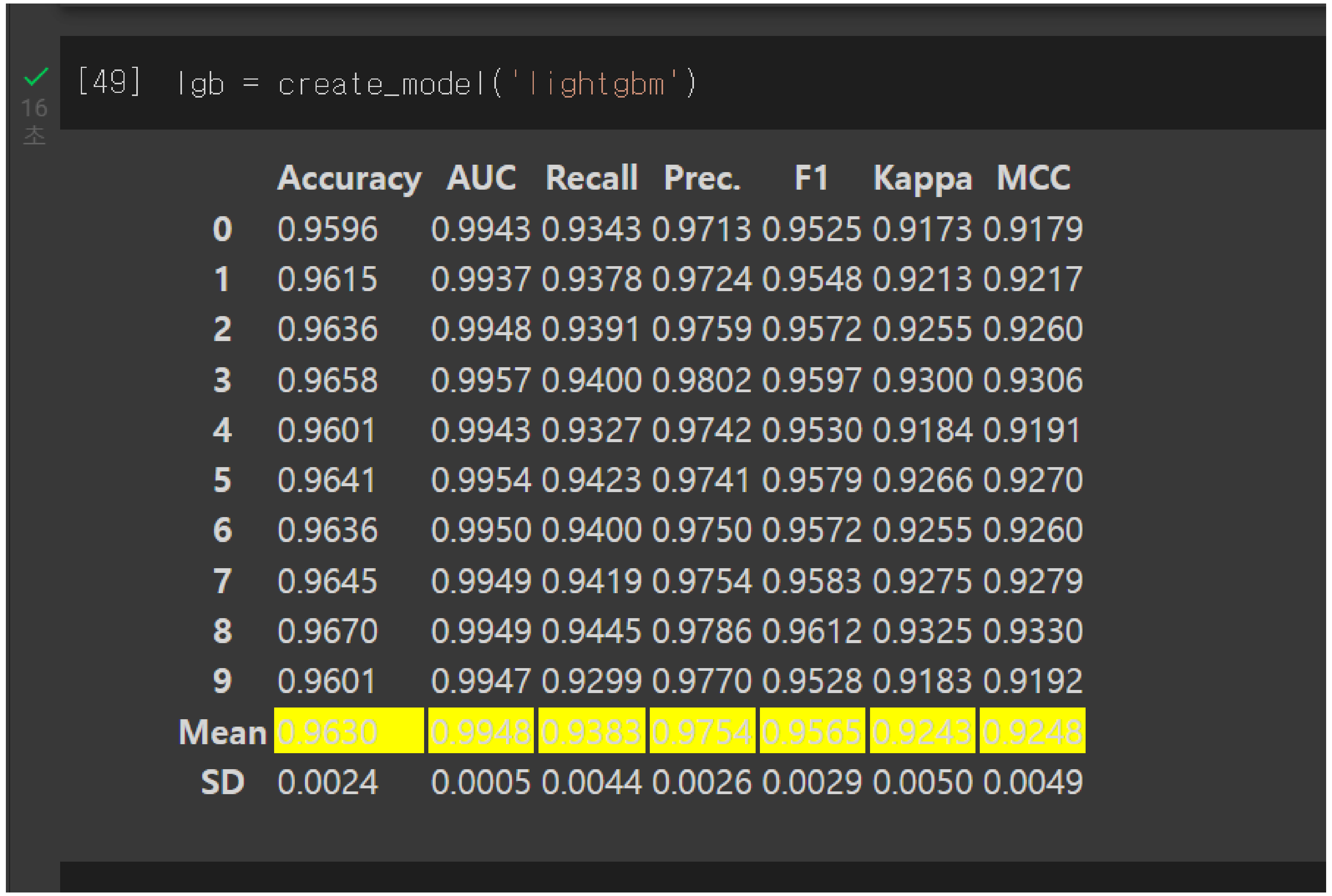
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9625	0.9948	0.9426	0.9700	0.9561	0.9233	0.9236
1	0.9616	0.9942	0.9410	0.9696	0.9551	0.9216	0.9219
2	0.9621	0.9952	0.9419	0.9696	0.9556	0.9225	0.9228
3	0.9662	0.9957	0.9419	0.9792	0.9602	0.9308	0.9314
4	0.9615	0.9947	0.9394	0.9708	0.9549	0.9213	0.9217
5	0.9660	0.9957	0.9496	0.9714	0.9604	0.9307	0.9309
6	0.9647	0.9954	0.9429	0.9747	0.9586	0.9278	0.9282
7	0.9643	0.9949	0.9467	0.9701	0.9582	0.9270	0.9272
8	0.9666	0.9952	0.9495	0.9727	0.9610	0.9318	0.9320
9	0.9600	0.9949	0.9368	0.9698	0.9530	0.9182	0.9186
Mean	0.9635	0.9951	0.9432	0.9718	0.9573	0.9255	0.9258
SD	0.0022	0.0005	0.0040	0.0029	0.0026	0.0045	0.0045

Catboost 모델 생성 후 Accuracy기준으로 최적화 해보기

비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개



16 초

```
[49]  lgb = create_model('lightgbm')
```

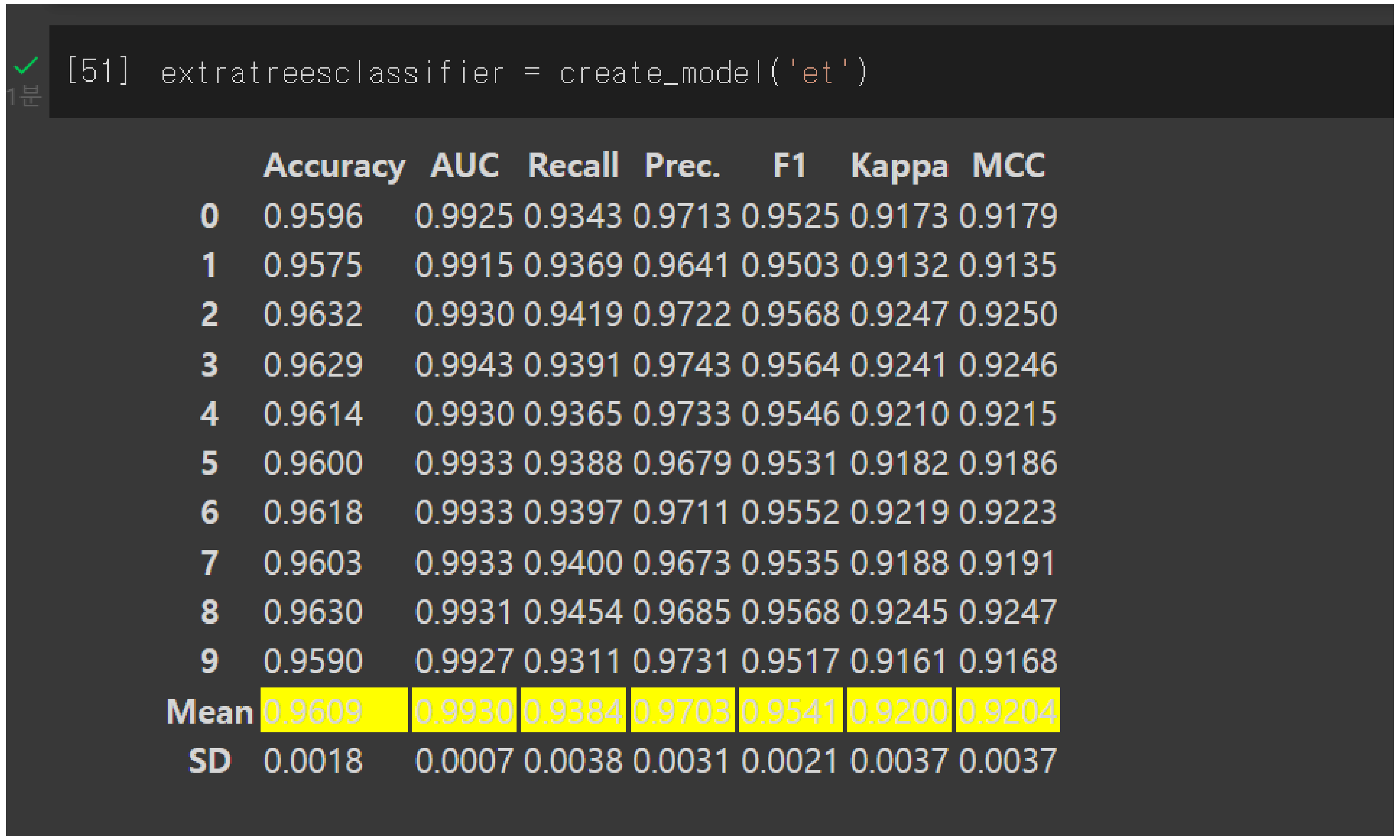
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9596	0.9943	0.9343	0.9713	0.9525	0.9173	0.9179
1	0.9615	0.9937	0.9378	0.9724	0.9548	0.9213	0.9217
2	0.9636	0.9948	0.9391	0.9759	0.9572	0.9255	0.9260
3	0.9658	0.9957	0.9400	0.9802	0.9597	0.9300	0.9306
4	0.9601	0.9943	0.9327	0.9742	0.9530	0.9184	0.9191
5	0.9641	0.9954	0.9423	0.9741	0.9579	0.9266	0.9270
6	0.9636	0.9950	0.9400	0.9750	0.9572	0.9255	0.9260
7	0.9645	0.9949	0.9419	0.9754	0.9583	0.9275	0.9279
8	0.9670	0.9949	0.9445	0.9786	0.9612	0.9325	0.9330
9	0.9601	0.9947	0.9299	0.9770	0.9528	0.9183	0.9192
Mean	0.9630	0.9948	0.9383	0.9754	0.9565	0.9243	0.9248
SD	0.0024	0.0005	0.0044	0.0026	0.0029	0.0050	0.0049

Lightgbm모델 생성 후 Accuracy기준으로 최적화 해보기

비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개



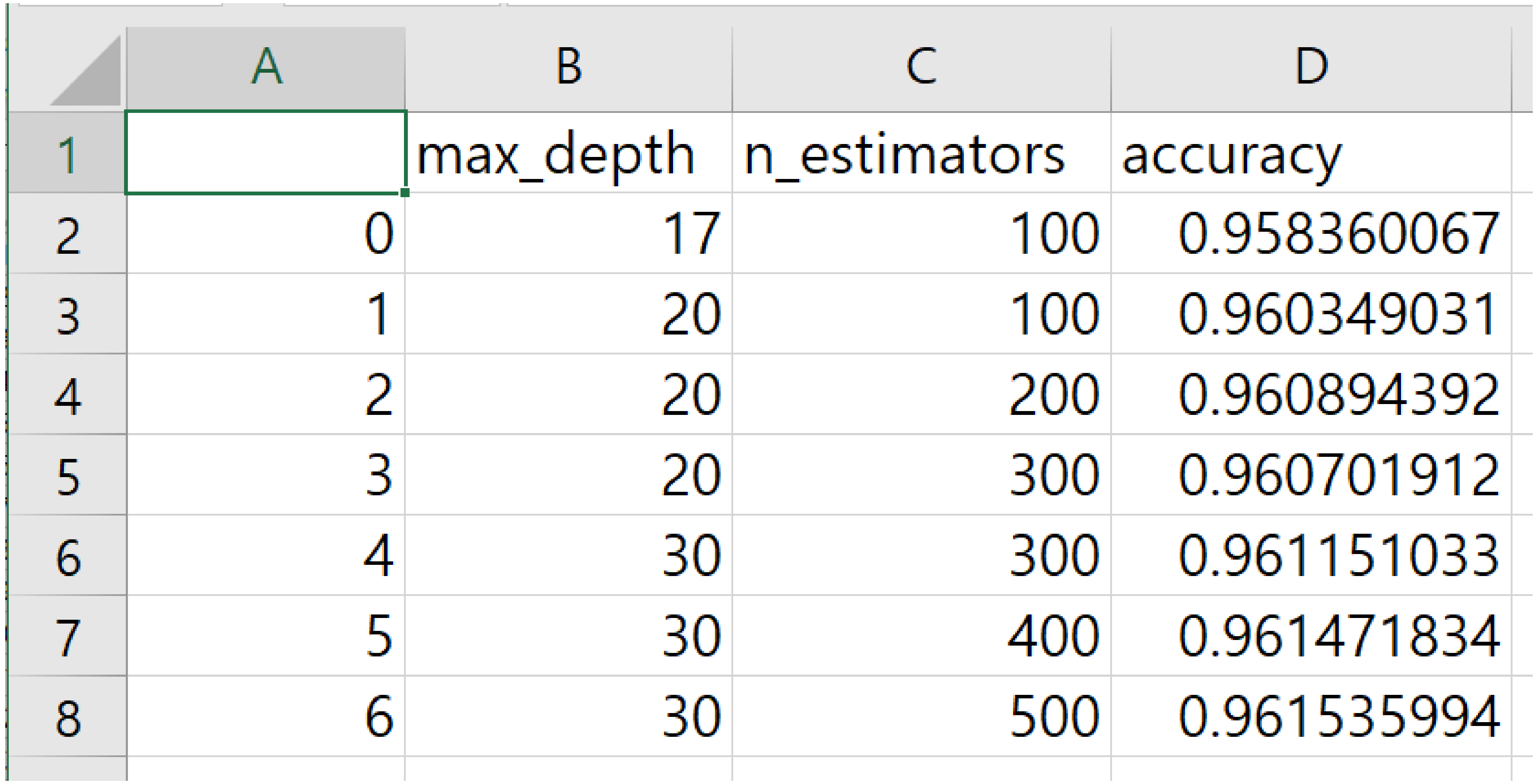
```
[51] extratreesclassifier = create_model('et')
```

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.9596	0.9925	0.9343	0.9713	0.9525	0.9173	0.9179
1	0.9575	0.9915	0.9369	0.9641	0.9503	0.9132	0.9135
2	0.9632	0.9930	0.9419	0.9722	0.9568	0.9247	0.9250
3	0.9629	0.9943	0.9391	0.9743	0.9564	0.9241	0.9246
4	0.9614	0.9930	0.9365	0.9733	0.9546	0.9210	0.9215
5	0.9600	0.9933	0.9388	0.9679	0.9531	0.9182	0.9186
6	0.9618	0.9933	0.9397	0.9711	0.9552	0.9219	0.9223
7	0.9603	0.9933	0.9400	0.9673	0.9535	0.9188	0.9191
8	0.9630	0.9931	0.9454	0.9685	0.9568	0.9245	0.9247
9	0.9590	0.9927	0.9311	0.9731	0.9517	0.9161	0.9168
Mean	0.9609	0.9930	0.9384	0.9703	0.9541	0.9200	0.9204
SD	0.0018	0.0007	0.0038	0.0031	0.0021	0.0037	0.0037

Extratrees Classifier모델 생성 후 Accuracy기준으로 최적화 해보기

4. 적용한 분석 기법 및 모델 소개

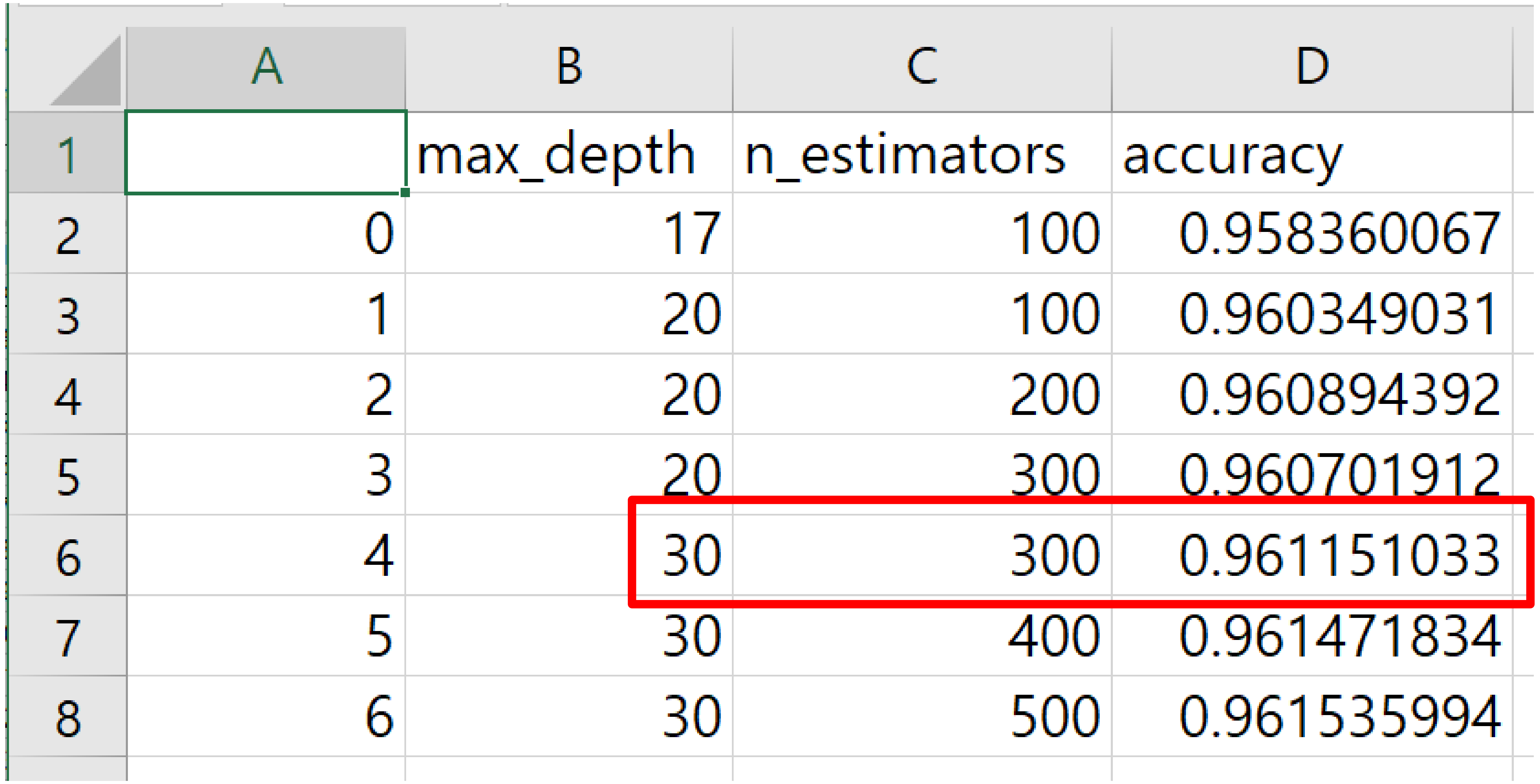
Extratrees Classifier의 hyperparameter 값을 수동으로 조정해 봄



	A	B	C	D
1		max_depth	n_estimators	accuracy
2	0	17	100	0.958360067
3	1	20	100	0.960349031
4	2	20	200	0.960894392
5	3	20	300	0.960701912
6	4	30	300	0.961151033
7	5	30	400	0.961471834
8	6	30	500	0.961535994

4. 적용한 분석 기법 및 모델 소개

Extreme Gradient Boosting의 hyperparameter 값을 수동으로 조정해 봄



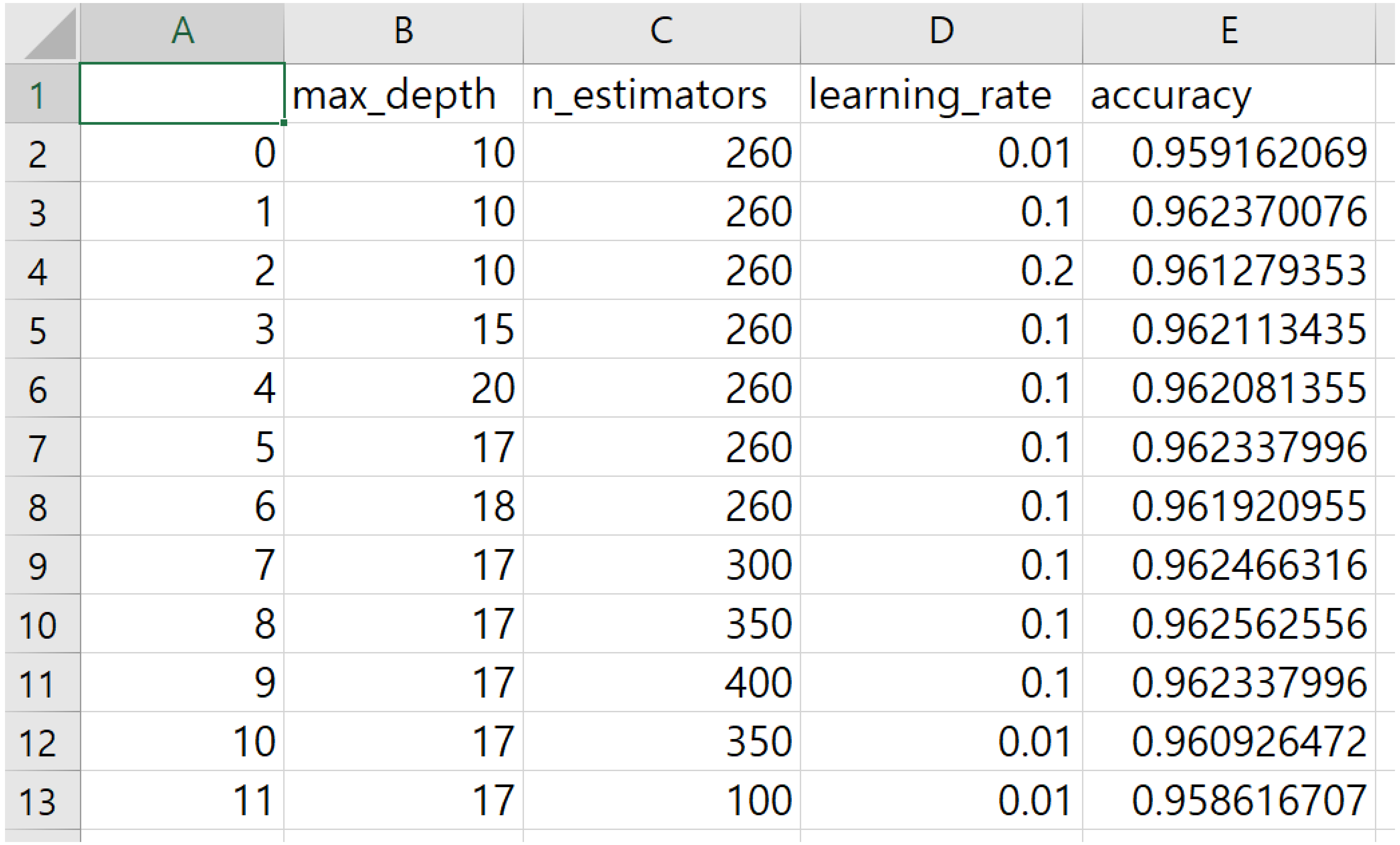
	A	B	C	D
1		max_depth	n_estimators	accuracy
2	0	17	100	0.958360067
3	1	20	100	0.960349031
4	2	20	200	0.960894392
5	3	20	300	0.960701912
6	4	30	300	0.961151033
7	5	30	400	0.961471834
8	6	30	500	0.961535994

비행승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개

Extratrees Classifier의 hyperparameter 값을 수동으로 조정해 봄

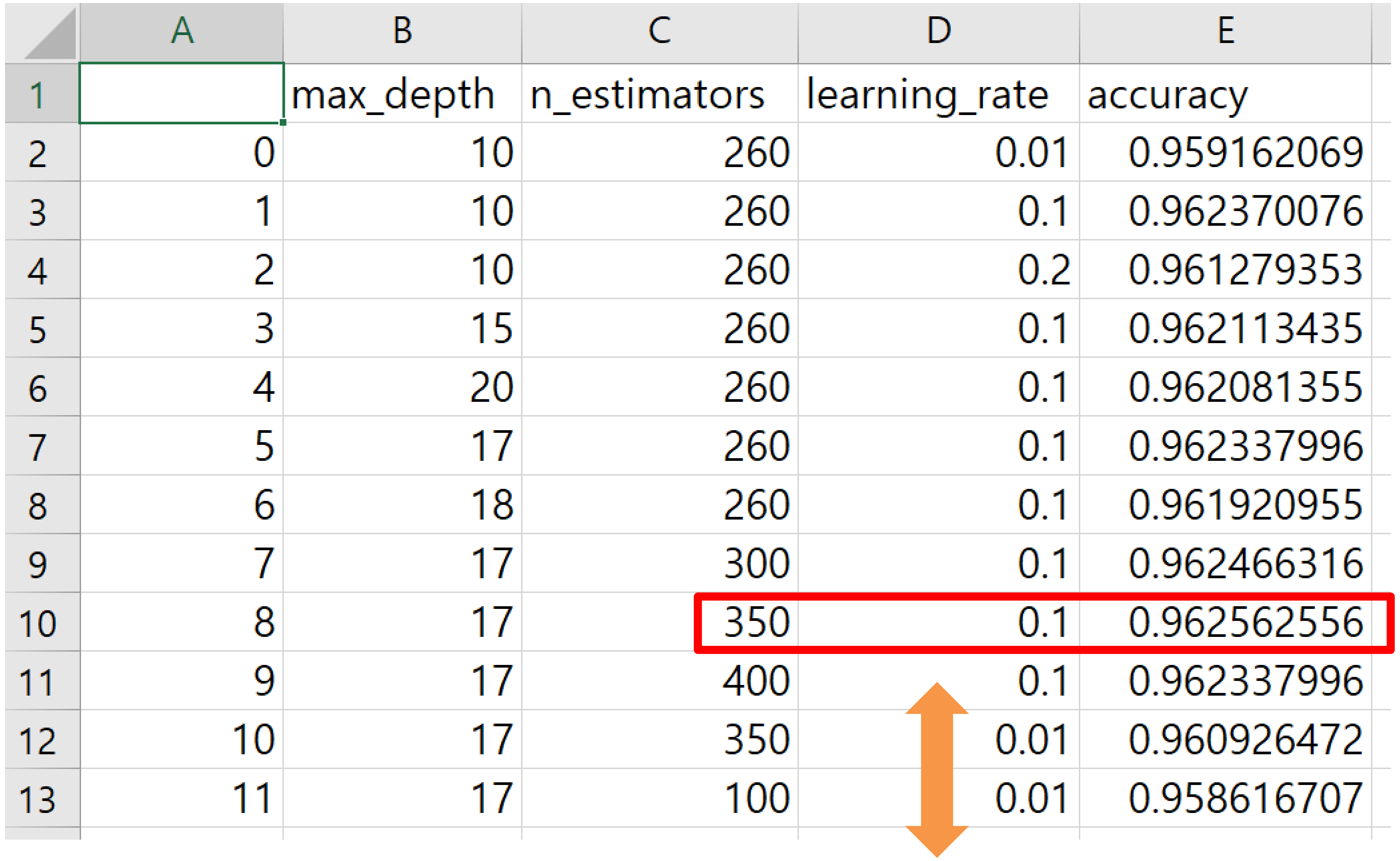


	A	B	C	D	E
1		max_depth	n_estimators	learning_rate	accuracy
2	0	10	260	0.01	0.959162069
3	1	10	260	0.1	0.962370076
4	2	10	260	0.2	0.961279353
5	3	15	260	0.1	0.962113435
6	4	20	260	0.1	0.962081355
7	5	17	260	0.1	0.962337996
8	6	18	260	0.1	0.961920955
9	7	17	300	0.1	0.962466316
10	8	17	350	0.1	0.962562556
11	9	17	400	0.1	0.962337996
12	10	17	350	0.01	0.960926472
13	11	17	100	0.01	0.958616707

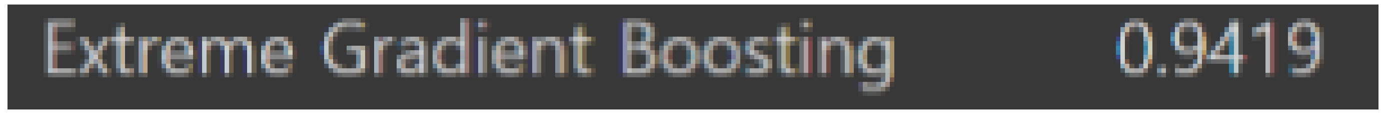
비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개



	A	B	C	D	E
1		max_depth	n_estimators	learning_rate	accuracy
2	0	10	260	0.01	0.959162069
3	1	10	260	0.1	0.962370076
4	2	10	260	0.2	0.961279353
5	3	15	260	0.1	0.962113435
6	4	20	260	0.1	0.962081355
7	5	17	260	0.1	0.962337996
8	6	18	260	0.1	0.961920955
9	7	17	300	0.1	0.962466316
10	8	17	350	0.1	0.962562556
11	9	17	400	0.1	0.962337996
12	10	17	350	0.01	0.960926472
13	11	17	100	0.01	0.958616707



Extreme Gradient Boosting 0.9419

Extreme Gradient Boosting의 hyperparameter 값을 수동으로 조정해 봄

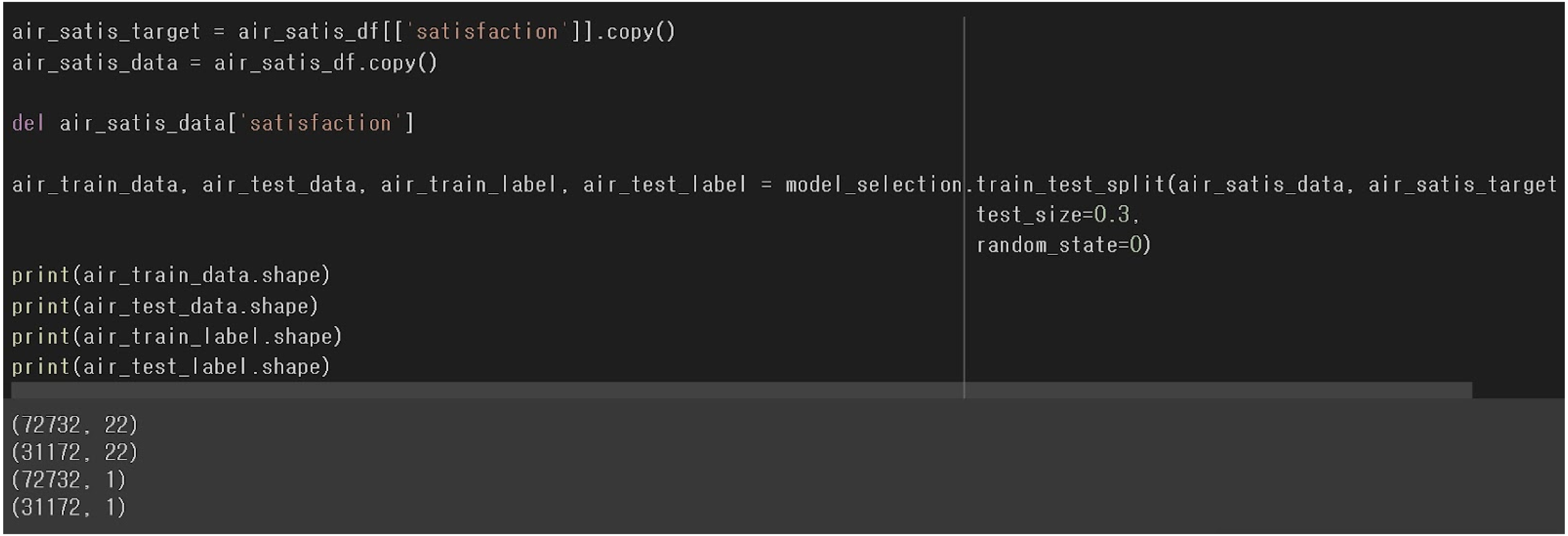
실질적으로 Pycaret에서 계산해준 값보다 Hyperparameter tuning을 통해 성능을 더 높일 수 있다는 것을 확인함

비행 승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)

딥러닝 넣어주기 전 데이터 나눠주기



```
air_satis_target = air_satis_df[['satisfaction']].copy()
air_satis_data = air_satis_df.copy()

del air_satis_data['satisfaction']

air_train_data, air_test_data, air_train_label, air_test_label = model_selection.train_test_split(air_satis_data, air_satis_target
                                                                                 test_size=0.3,
                                                                                 random_state=0)

print(air_train_data.shape)
print(air_test_data.shape)
print(air_train_label.shape)
print(air_test_label.shape)
```

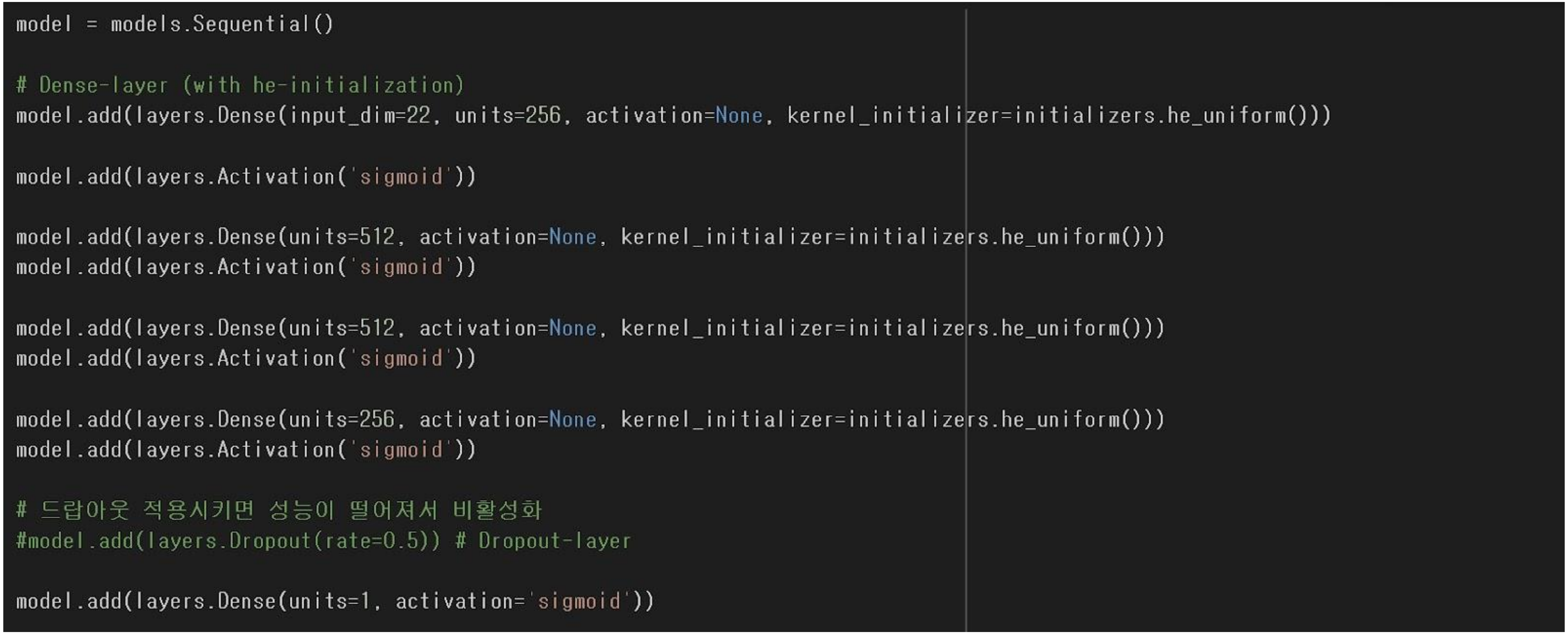
```
(72732, 22)
(31172, 22)
(72732, 1)
(31172, 1)
```

비행 승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개(딥러닝)

input_dim = 22 , activation = sigmoid , 가장밑줄 unit = 1 인 이유설명 (발표자)

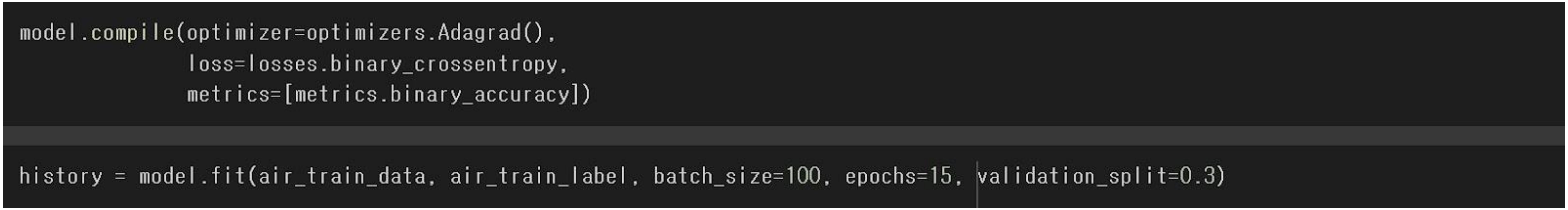


```
model = models.Sequential()

# Dense-layer (with he-initialization)
model.add(layers.Dense(input_dim=22, units=256, activation=None, kernel_initializer=initializers.he_uniform()))

model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=512, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=512, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=256, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))

# 드랍아웃 적용시키면 성능이 떨어져서 비활성화
#model.add(layers.Dropout(rate=0.5)) # Dropout-layer

model.add(layers.Dense(units=1, activation='sigmoid'))
```

비행 승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)



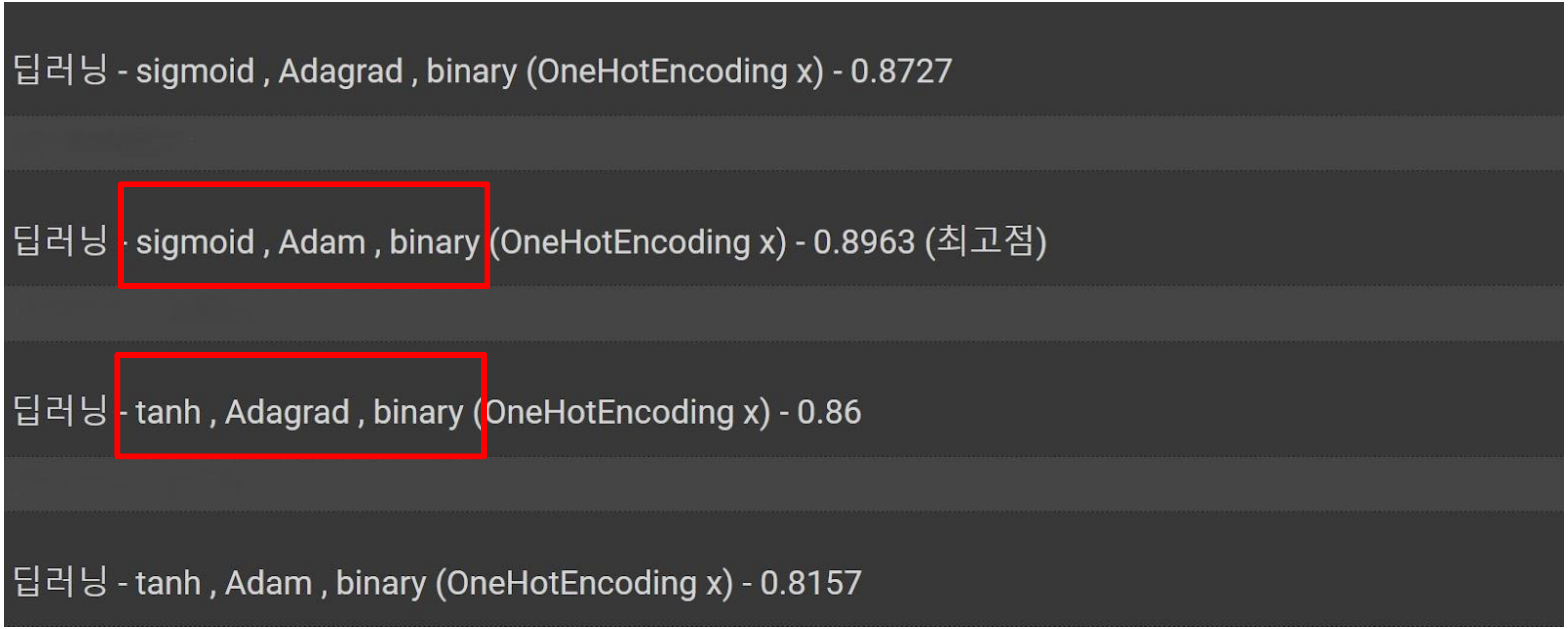
```
model.compile(optimizer=optimizers.Adagrad(),
              loss=losses.binary_crossentropy,
              metrics=[metrics.binary_accuracy])
```

```
history = model.fit(air_train_data, air_train_label, batch_size=100, epochs=15, validation_split=0.3)
```

비행승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개(딥러닝)



딥러닝 - sigmoid , Adagrad , binary (OneHotEncoding x) - 0.8727

딥러닝 - sigmoid , Adam , binary (OneHotEncoding x) - 0.8963 (최고점)

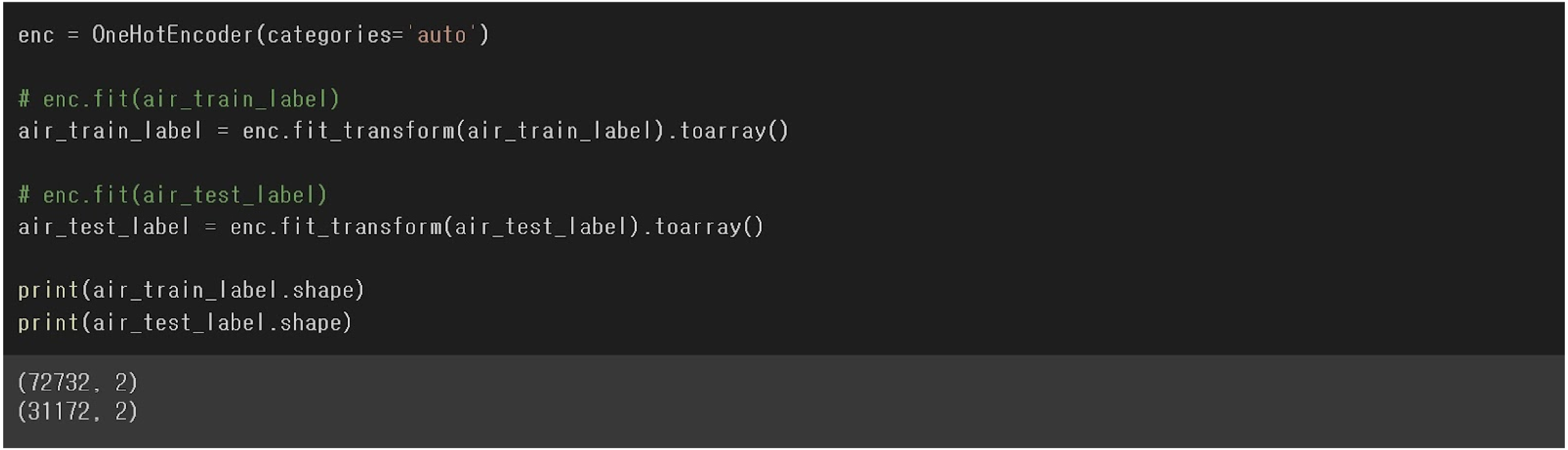
딥러닝 - tanh , Adagrad , binary (OneHotEncoding x) - 0.86

딥러닝 - tanh , Adam , binary (OneHotEncoding x) - 0.8157

비행 승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)



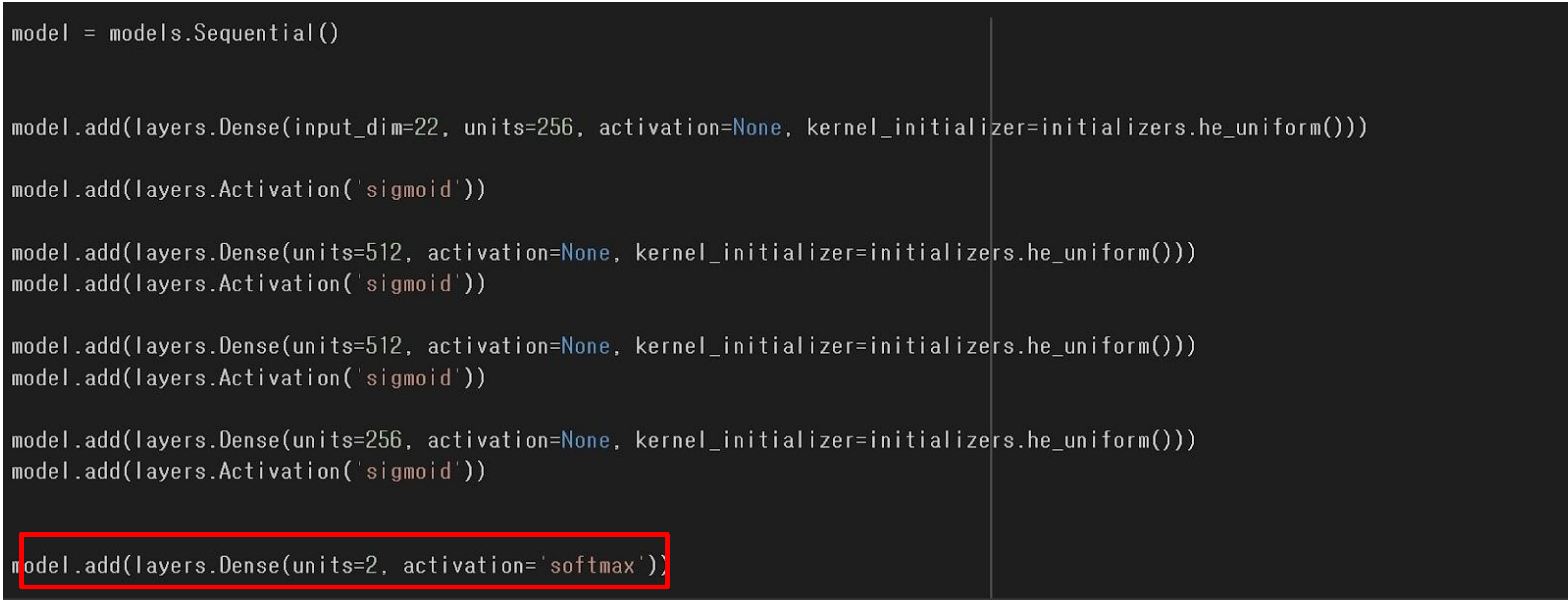
```
enc = OneHotEncoder(categories='auto')

# enc.fit(air_train_label)
air_train_label = enc.fit_transform(air_train_label).toarray()

# enc.fit(air_test_label)
air_test_label = enc.fit_transform(air_test_label).toarray()

print(air_train_label.shape)
print(air_test_label.shape)
```

```
(72732, 2)
(31172, 2)
```

비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)

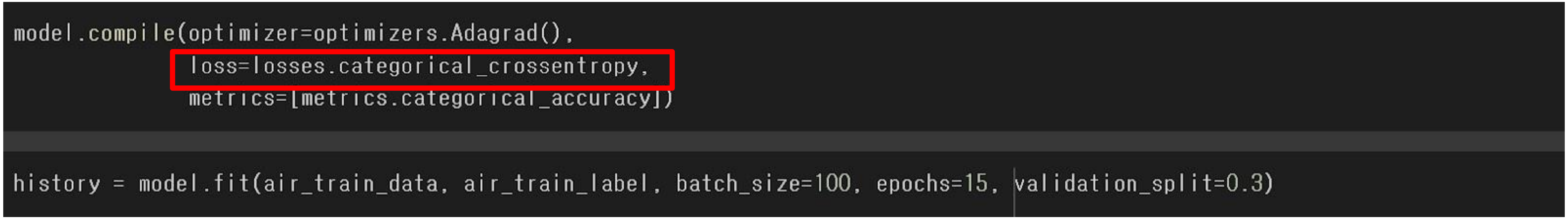


```
model = models.Sequential()


model.add(layers.Dense(input_dim=22, units=256, activation=None, kernel_initializer=initializers.he_uniform()))

model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=512, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=512, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))

model.add(layers.Dense(units=256, activation=None, kernel_initializer=initializers.he_uniform()))
model.add(layers.Activation('sigmoid'))


model.add(layers.Dense(units=2, activation='softmax'))
```

비행 승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)



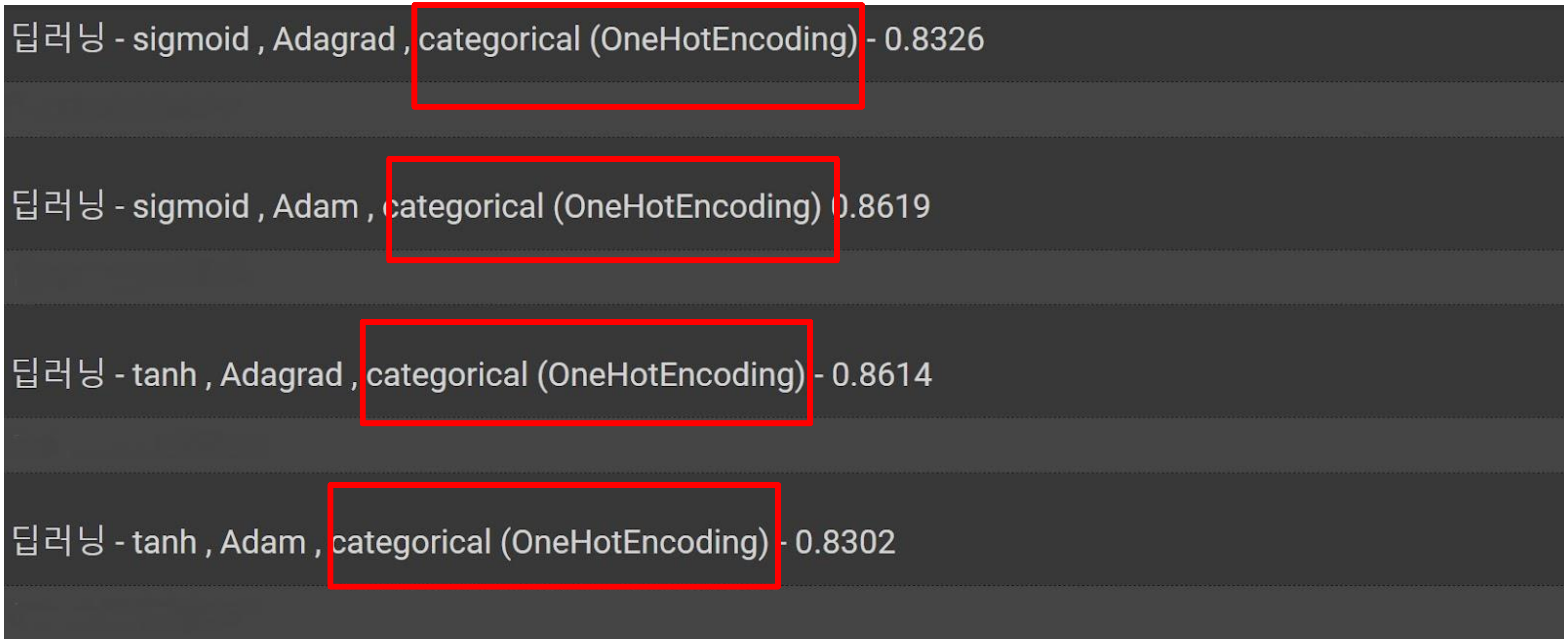
```
model.compile(optimizer=optimizers.Adagrad(),
              loss=losses.categorical_crossentropy,
              metrics=[metrics.categorical_accuracy])
```

```
history = model.fit(air_train_data, air_train_label, batch_size=100, epochs=15, validation_split=0.3)
```

비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)



딥러닝 - sigmoid , Adagrad , categorical (OneHotEncoding) - 0.8326

딥러닝 - sigmoid , Adam , categorical (OneHotEncoding) 0.8619

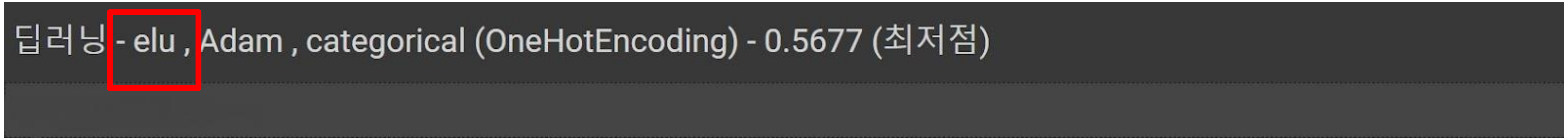
딥러닝 - tanh , Adagrad , categorical (OneHotEncoding) - 0.8614

딥러닝 - tanh , Adam , categorical (OneHotEncoding) - 0.8302

비행승객 만족도 분석

Airline Passensger Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)

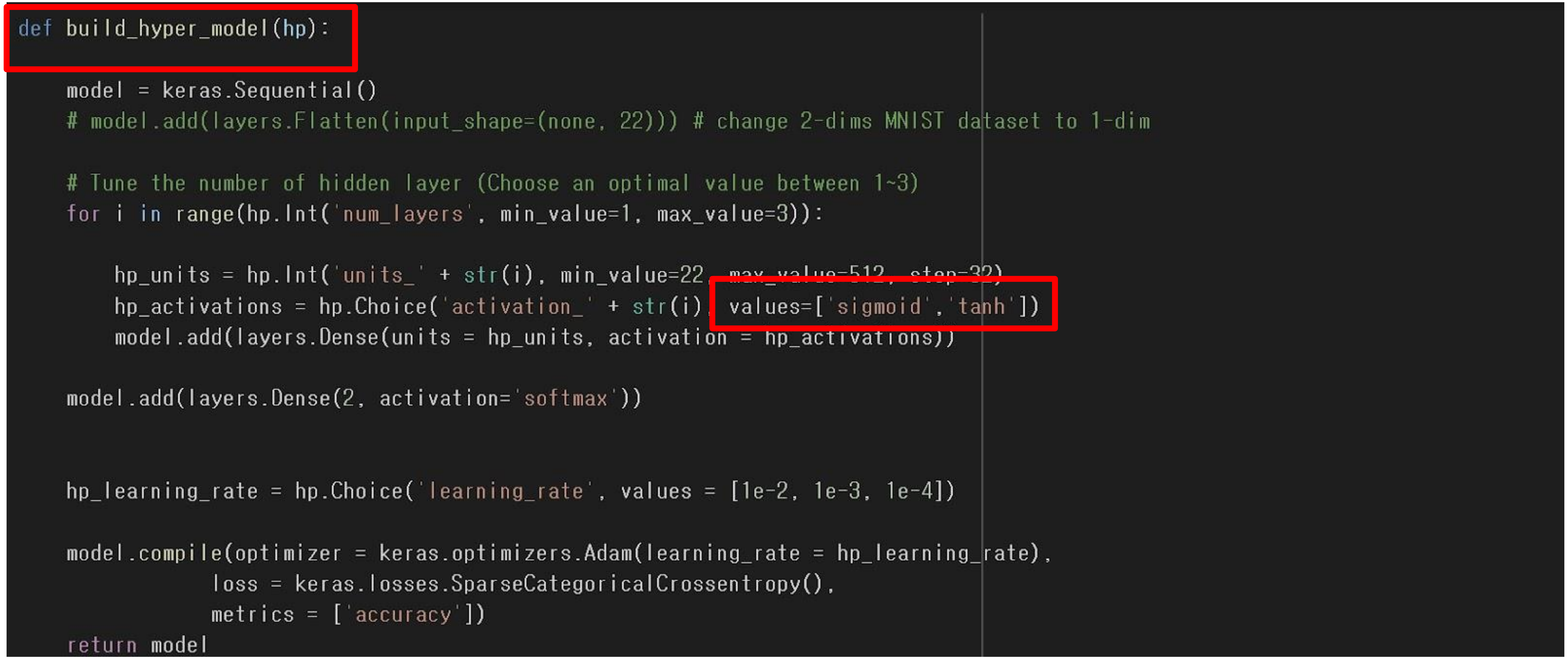


딥러닝 - elu , Adam , categorical (OneHotEncoding) - 0.5677 (최저점)

비행 승객 만족도 분석

Airline Passensger Satisfaction

4. 적용한 분석 기법 및 모델 소개(딥러닝)

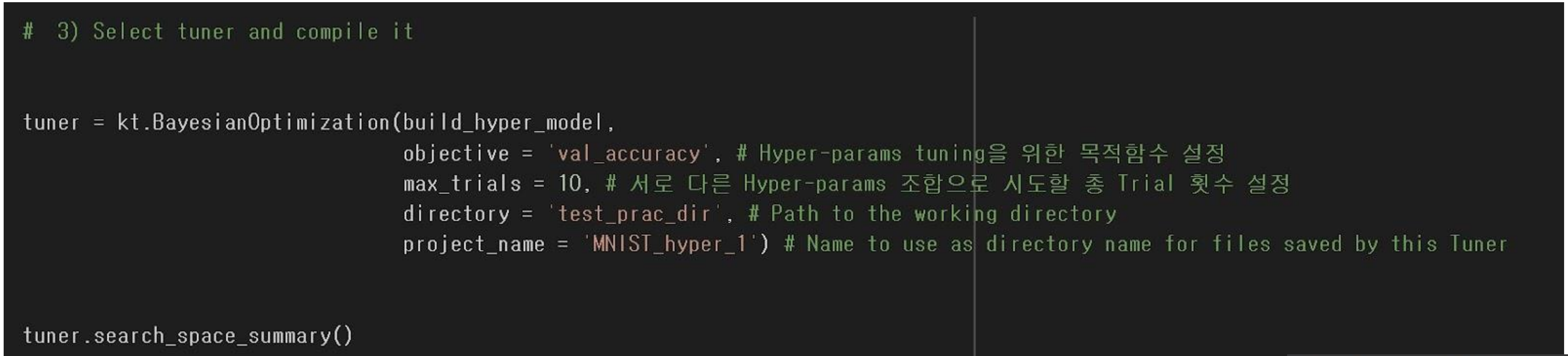


```
def build_hyper_model(hp):

    model = keras.Sequential()
    # model.add(layers.Flatten(input_shape=(none, 22))) # change 2-dims MNIST dataset to 1-dim

    # Tune the number of hidden layer (Choose an optimal value between 1~3)
    for i in range(hp.Int('num_layers', min_value=1, max_value=3)):

        hp_units = hp.Int('units_' + str(i), min_value=22, max_value=512, step=32)
        hp_activations = hp.Choice('activation_' + str(i), values=['sigmoid','tanh'])
        model.add(layers.Dense(units = hp_units, activation = hp_activations))

    model.add(layers.Dense(2, activation='softmax'))


    hp_learning_rate = hp.Choice('learning_rate', values = [1e-2, 1e-3, 1e-4])

    model.compile(optimizer = keras.optimizers.Adam(learning_rate = hp_learning_rate),
                  loss = keras.losses.SparseCategoricalCrossentropy(),
                  metrics = ['accuracy'])
    return model
```

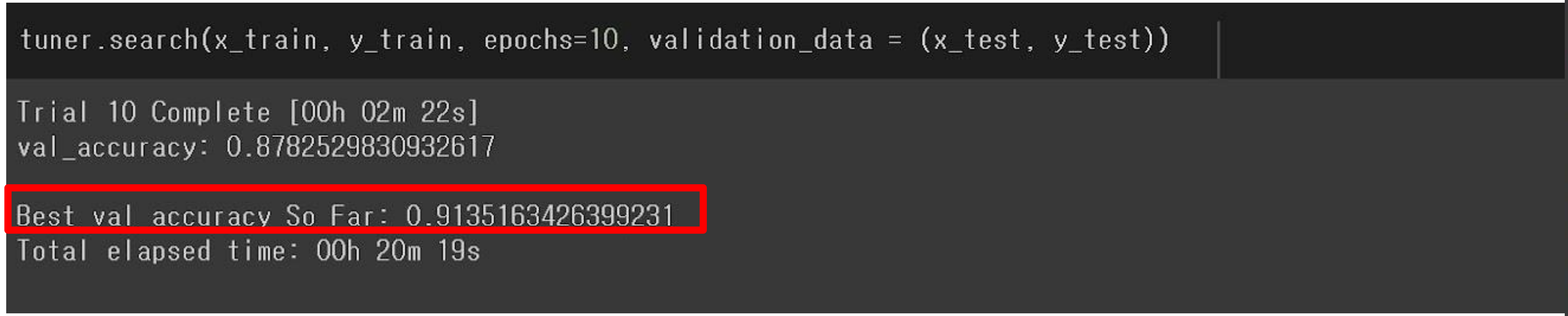
비행승객 만족도 분석

Airline Passensger Satisfication

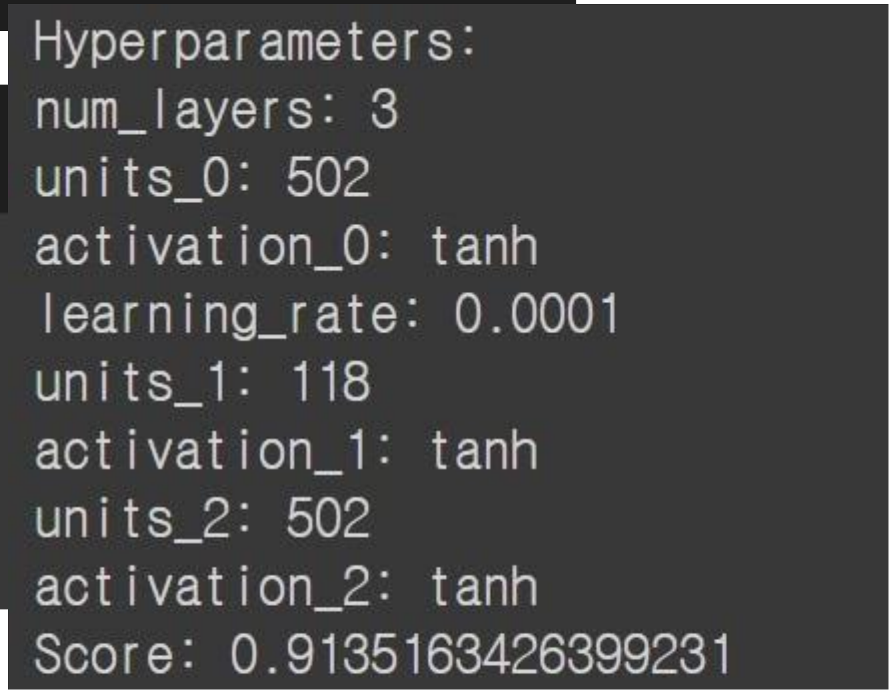
4. 적용한 분석 기법 및 모델 소개(딥러닝)



```
#  3) Select tuner and compile it


tuner = kt.BayesianOptimization(build_hyper_model,
                                objective = 'val_accuracy', # Hyper-params tuning을 위한 목적함수 설정
                                max_trials = 10, # 서로 다른 Hyper-params 조합으로 시도할 총 Trial 횟수 설정
                                directory = 'test_prac_dir', # Path to the working directory
                                project_name = 'MNIST_hyper_1') # Name to use as directory name for files saved by this Tuner


tuner.search_space_summary()
```



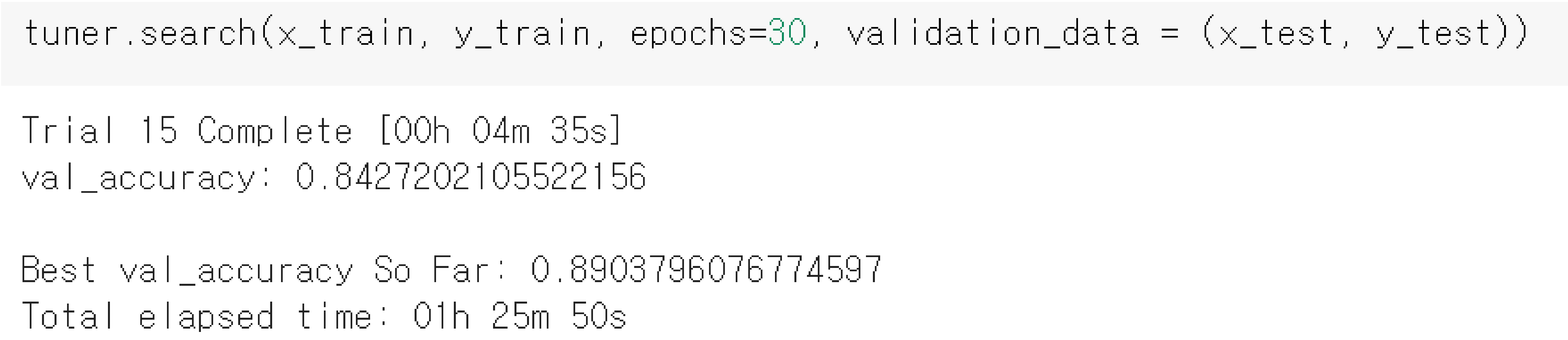
```
tuner.search(x_train, y_train, epochs=10, validation_data = (x_test, y_test))

Trial 10 Complete [00h 02m 22s]
val_accuracy: 0.8782529830932617

Best val accuracy So Far: 0.9135163426399231
Total elapsed time: 00h 20m 19s
```



```
Hyperparameters:
num_layers: 3
units_0: 502
activation_0: tanh
learning_rate: 0.0001
units_1: 118
activation_1: tanh
units_2: 502
activation_2: tanh
Score: 0.9135163426399231
```

4. 적용한 분석 기법 및 모델 소개(딥러닝)

딥러닝 (케라스 튜너)
가장 성능이 잘 나온 tanh 실험(layer의 수, 시도 횟수, epoch의 수 수정)



```
tuner.search(x_train, y_train, epochs=30, validation_data = (x_test, y_test))
```

```
Trial 15 Complete [00h 04m 35s]
val_accuracy: 0.8427202105522156

Best val_accuracy So Far: 0.8903796076774597
Total elapsed time: 01h 25m 50s
```

4. 적용한 분석 기법 및 모델 소개(딥러닝)

딥러닝 레이어 , unit 수 바꾸기

test 파일 넣어서 적용해보기 (지금까지 train 파일 10만행 짜리로 진행했음)

비행승객
만족도 분석
Airline Passensger
Satisfication

4. 적용한 분석 기법 및 모델 소개(딥러닝)

감사합니다.