06

로지스틱 회귀분석 실습

[전처리]

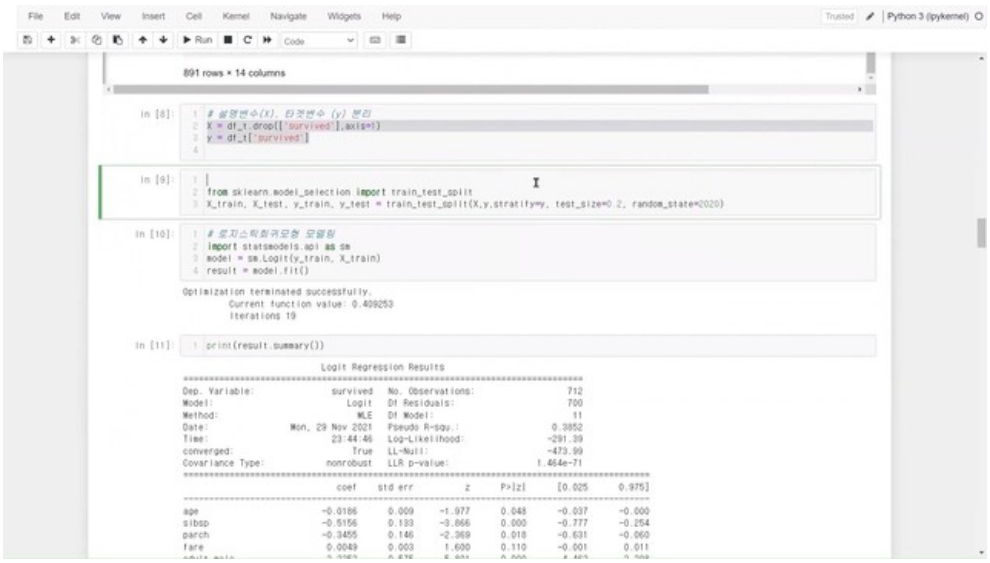
로지스틱 회귀분석을 실습해보겠습니다. 전처리와 핸들링은 책을 안보고도 하실 수 있어야 합니다

00:25

[데이터 분할 및 모델 학습]

홀드아웃 기법을 사용하여 train, test 데이터로 나누어주고 train 데이터를 통해 모델을 학습시켜줍니다.

06:12

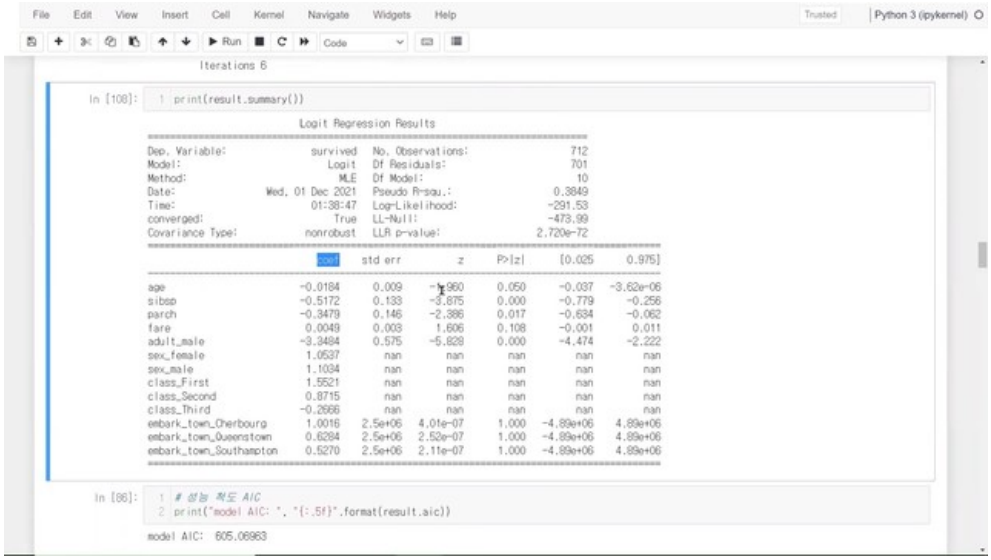


[모델 최적화]

변수선택법을 통해 로지스틱 모델을 최적화 시켜주도록 합니다.

- 머신러닝이 발전하면서, 변수선택법보다 내부적으로 패널티를 주는 방식을 선호합니다. 6-4에서 패널티를 사용하는 모델을 학습하겠습니다.

11:51

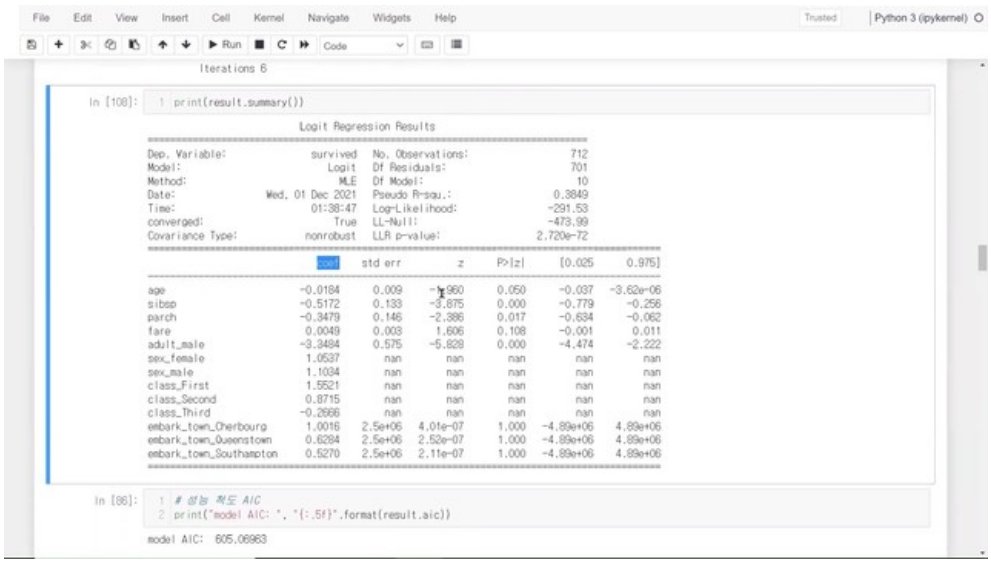


[모델 최적화]

변수선택법을 통해 로지스틱 모델을 최적화 시켜주도록 합니다.

- 머신러닝이 발전하면서, 변수선택법보다 내부적으로 패널티를 주는 방식을 선호합니다. 6-4에서 패널티를 사용하는 모델을 학습하겠습니다.

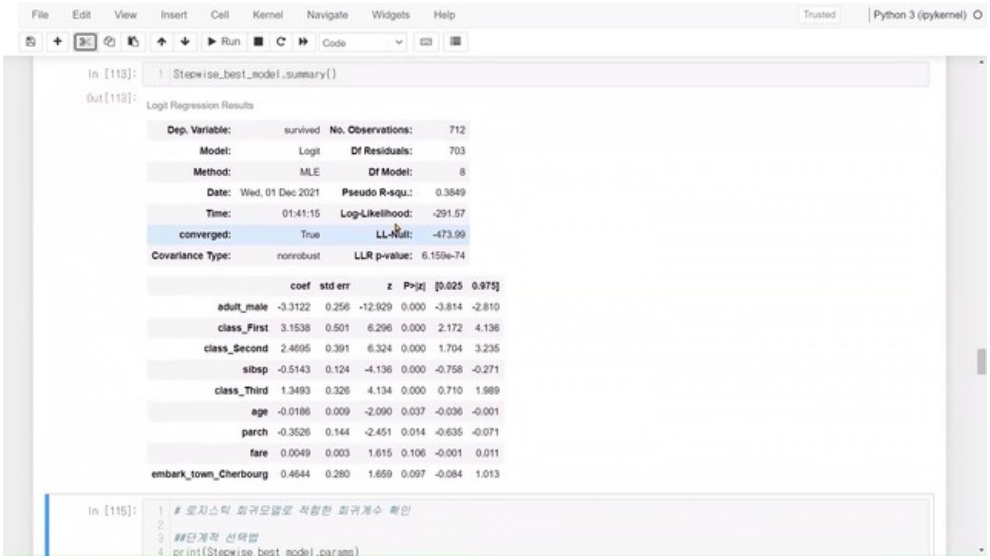
11:51



[회귀계수 설명]

회귀계수는 오즈비로 설명할 수 있습니다.

14:25



In [113]:
```
Stepwise_best_model.summary()
```

Out[113]:

Logit Regression Results

Dep. Variable:	survived	No. Observations:	712
Model:	Logit	Df Residuals:	703
Method:	MLE	Df Model:	8
Date:	Wed, 01 Dec 2021	Pseudo R-squ.:	0.3849
Time:	01:41:15	Log-Likelihood:	-291.57
converged:	True	LL-Null:	-473.99
Covariance Type:	nonrobust	LLR p-value:	6.159e-74

	coef	std err	z	P>|z|	[0.025	0.975]
adult_male	-3.3122	0.256	-12.929	0.000	-3.814	-2.810
class_First	3.1538	0.501	6.296	0.000	2.172	4.136
class_Second	2.4695	0.391	6.324	0.000	1.704	3.235
sibsp	-0.5143	0.124	-4.136	0.000	-0.758	-0.271
class_Third	1.3493	0.326	4.134	0.000	0.710	1.989
age	-0.0186	0.009	-2.090	0.037	-0.036	-0.001
parch	-0.3526	0.144	-2.451	0.014	-0.635	-0.071
fare	0.0049	0.003	1.615	0.106	-0.001	0.011
embark_town_Cherbourg	0.4644	0.280	1.659	0.097	-0.084	1.013

In [115]:
```
# 로지스틱 회귀모델로 적합한 회귀계수 확인

##단계적 선택법
print(Stepwise_best_model.params)
```

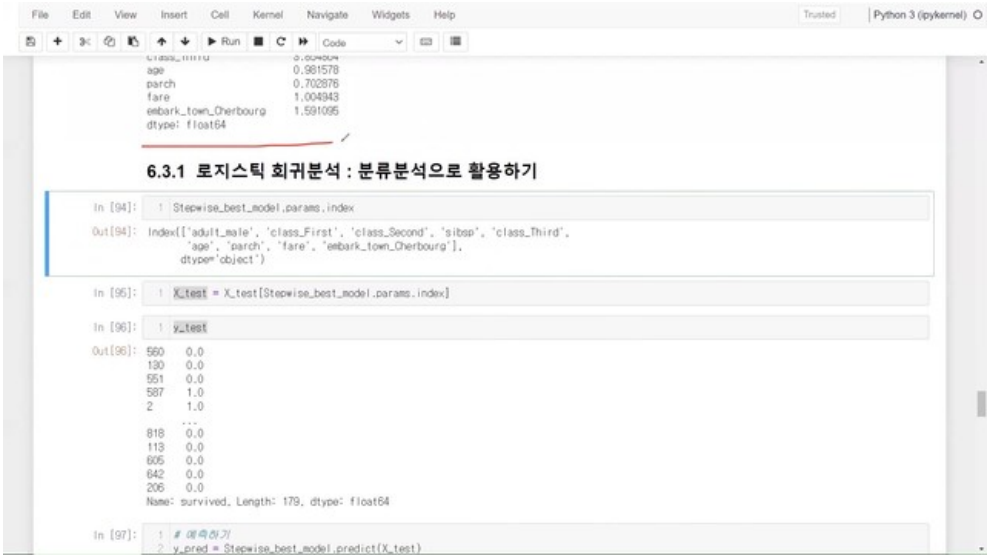
[로지스틱 회귀 모델은 분류분석]

로지스틱 회귀모델은 두가지 목적으로 사용됩니다.

1) 회귀계수의 오즈비를 확인하는 목적

2) 독립변수들의 특징을 사용하여 종속변수를 확률적으로 예측하고자 하는 목적

18:34



```
age                    0.981578
parch                  0.702876
fare                   1.004943
embark_town_Cherbourg  1.591095
dtype: float64
```

6.3.1 로지스틱 회귀분석 : 분류분석으로 활용하기

In [94]:
```
Stepwise_best_model.params.index
```

Out[94]:
```
Index(['adult_male', 'class_First', 'class_Second', 'sibsp', 'class_Third',
       'age', 'parch', 'fare', 'embark_town_Cherbourg'],
      dtype='object')
```

In [95]:
```
X_test = X_test[Stepwise_best_model.params.index]
```

In [96]:
```
y_test
```

Out[96]:
```
560    0.0
130    0.0
551    0.0
587    1.0
2      1.0
      ...
818    0.0
113    0.0
605    0.0
642    0.0
206    0.0
Name: survived, Length: 179, dtype: float64
```

In [97]:
```
# 예측하기
y_pred = Stepwise_best_model.predict(X_test)
```

[분류분석의 평가지표]

분류분석은 평가지표로 혼동행렬을 사용합니다.

- 혼동행렬을 통해 ROC_AUC_SCORE를 구할 수 있습니다.

22:10