

전처리 실습

Data Preprocessing

전처리 실습

■ 실습 문제 (1)

■ 이직을 희망하는 직원 예측 문제

- 학습 데이터의 라벨 : 1.0 이직 희망
- 학습 데이터의 라벨 : 0.0 이직 희망하지 않음

■ 평가를 위해 카테고리 정확도 (Categorization Accuracy) 를 사용

Description



과제 설명

최근 N사는 빅데이터 분야와 데이터 사이언스 분야를 활용한 서비스가 많아지면서 해당 분야 개발자가 많이 필요해졌다. 이에 외부에서 직원을 추가 채용하는 부분을 고려하기 전 사내 부서 이동 제도를 통해 빅데이터 분야와 데이터사이언스분야로 옮기고 싶어하는 직원들의 신청을 받기로 했다. 그러나 HR(인사팀)에서 근무하는 솔잎 양은 해당 직원들의 정보가 인공지능 관련 부서로의 이동 뿐만 아니라, 현 부서의 만족도가 낮아(분야, 임금, 동료, 등) 이직을 고려 중인 직원으로 분류 가능하다는 분석 결과를 도출하고 사내 직원들 중 현재 직장을 그만두고 새로운 일자리를 알아보는 직원을 예측하는 소프트웨어를 만들어 보려 한다. 여러분 역시 수업 시간에 배운 <머신 러닝 기술>을 활용하여, test.csv 파일로 제공된 이직을 희망하는 직원을 예측해 주는 인공지능 SW를 작성해주기 바란다.

학습 데이터는 (직원의 개인 정보)와 직원의 (이직 희망 여부)를 제공한다.

테스트 데이터는 (직원의 개인 정보)만 제공하며, 예측된 직원의 (이직 희망 여부)는 submit.csv 파일로 저장하여 캐글 리더보드에 제출해야 한다.

제공되는 (직원의 개인 정보)는 순서대로 직원의 고유 ID, 도시 코드, 도시 개발 지수, 직원의 성별, 직원의 데이터 사이언스분야 관련 경험, 현 대학 등록 여부(풀타임, 파트타임, 없음), 학위(고졸, 대졸 등), 직원의 경력, 현 회사의 직원 수, 고용 유형, 이전 회사 입사 년도, 이수한 교육 시간, 이직 희망 여부 이다.

자, 그럼 테스트 데이터로 주어진 (직원의 개인 정보)에 맞는 직원의 (이직 희망 여부)를 예측하여 보자.

- 예측 결과를 int64로 변환하여 csv파일로 제출하시기 바랍니다

전처리 실습

- 정답 제출시 타입 확인 필요



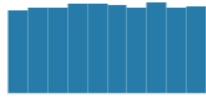
```
# 그리하여 예측된 결과인 Y_test를 형변환을 통해 int64로 변경한 후, csv 파일로 저장합니다.  
  
submit['target'] = Y_test.astype('int64')  
submit.to_csv("submit.csv")
```

+ Code+ Markdown

- 학습 모델

- 베이스라인 학습 모델 : QDA

sample_submit.csv (28.47 kB)

# index	# target
 219.2k	3832 total values
2	0
9	0
10	1
11	1
15	0
17	0
21	
24	
26	

전처리 실습

- 학습데이터 X, Y 나누기

[2]:

```
# 데이터를 읽어옵니다.  
train = pd.read_csv("/kaggle/input/2022-ml-w13p1/train.csv")  
test = pd.read_csv("/kaggle/input/2022-ml-w13p1/test.csv")
```

▷

```
# target 정보는 라벨 정보이므로 별도로 분리합니다.  
X_train = X_train.drop(['target'], axis=1)  
Y_train = train['target']
```

I

전처리 실습

- Pandas 데이터 살펴보기
 - Head()

```
# 학습 데이터를 눈으로 보면서 어떻게 가공할지 고민합니다.
train.head(20)
```

[5]:

	index	enrollee_id	city	city_development_index	gender	relevent_experience	enrolled_university	education_level
0	0	8949	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate
1	1	29725	city_40	0.776	Male	No relevent experience	no_enrollment	Graduate
2	3	33241	city_115	0.789	NaN	No relevent experience	NaN	Graduate
3	4	666	city_162	0.767	Male	Has relevent experience	no_enrollment	Masters
4	5	21651	city_176	0.764	NaN	Has relevent experience	Part time course	Graduate
5	6	28806	city_160	0.920	Male	Has relevent experience	no_enrollment	High School
6	7	402	city_46	0.762	Male	Has relevent experience	no_enrollment	Graduate
7	8	27107	city_103	0.920	Male	Has relevent experience	no_enrollment	Graduate
8	12	25619	city_61	0.913	Male	Has relevent experience	no_enrollment	Graduate
9	13	5826	city_21	0.624	Male	No relevent experience	NaN	NaN
10	14	8722	city_21	0.624	NaN	No relevent experience	Full time course	High School
11	16	4167	city_103	0.920	NaN	Has relevent experience	no_enrollment	Graduate
12	18	2155	city_21	0.624	Male	Has relevent	no_enrollment	Graduate

전처리 실습

- 데이터 삭제하기



```
# index는 삭제하면 될 것으로 생각이 되고, enrollee_id도 삭제하면 될 것 같네요.  
# city 정보는 city_development_index로 대체할 수 있을 것으로 생각되어 삭제하겠습니다.  
|  
X_train = train.drop(['index', 'enrollee_id', 'city'],axis=1)  
X_test = test.drop(['index', 'enrollee_id', 'city'],axis=1)
```

전처리 실습

- Pandas 데이터 타입 확인하기
 - Info()
 - object는 보통 문자열이거나 문자, 숫자가 복합된 형태의 데이터 타입



```
# 범주형 데이터를 눈으로 쉽게 확인하는 방법은!! info() 함수를 사용해서 Dtype 정보를 확인 하는 것
```

```
X_train.info()
```

```
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15326 entries, 0 to 15325
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   city_development_index 15326 non-null  float64
1   gender                 11752 non-null  object
2   relevent_experience     15326 non-null  object
3   enrolled_university    15009 non-null  object
4   education_level        14964 non-null  object
5   major_discipline       13068 non-null  object
6   experience             15276 non-null  object
7   company_size           10612 non-null  object
8   company_type           10445 non-null  object
9   last_new_job           14983 non-null  object
10  training_hours         15326 non-null  int64
```

```
dtypes: float64(1), int64(1), object(9)
```

```
memory usage: 1.3+ MB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3832 entries, 0 to 3831
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   city_development_index 3832 non-null  float64
1   gender                 2898 non-null  object
2   relevent_experience     3832 non-null  object
```

전처리 실습

- 범주형 데이터 수치형으로 변환하기
 - 남성, 여성 → 0, 1

```
# 모델 학습을 위해서는 object로 되어 있는 범주형을 수치형(int64, float64 등)으로 변환해야 합니다.  
# 그런데, X에만 있고 Y에 없는 데이터가 존재할 수 있어, 학습과 테스트 데이터를 합쳐서 범주형을 수치형으로 변경합니다.  
# 우선 for문과 같은 문법을 사용하지 않고, 쉽게 나열하는 방법으로 문제를 해결해 보겠습니다.
```

```
from sklearn.preprocessing import LabelEncoder
```

```
X_data = pd.concat([X_train, X_test], axis=0)
```

```
# gender (성별)
le = LabelEncoder()
le.fit(X_data['gender'])
X_train['gender'] = le.transform(X_train['gender'])
X_test['gender'] = le.transform(X_test['gender'])
```

```
# relevent_experience (관련 경력)
le = LabelEncoder()
le.fit(X_data['relevent_experience'])
X_train['relevent_experience'] = le.transform(X_train['relevent_experience'])
X_test['relevent_experience'] = le.transform(X_test['relevent_experience'])
```


전처리 실습

중복데이터 제외한 데이터 확인하기



X_train

```
[14...] rolled_university education_level major_discipline experience company_size company_type last_new_job training_hours
      2              0              5      >20          NaN              6              1              36
      2              0              5       15        50-99              5              >4              47
      3              0              1       <1          NaN              5             never              52
      2              2              5      >20        50-99              1              4              8
      1              0              5       11          NaN              6              1              24
      ...            ...            ...      ...          ...            ...            ...            ...
      2              0              2        7        10/49              1              1              25
      2              0              2       14          NaN              6              1              42
      2              0              5       14          NaN              6              4              52
      2              1              6       <1        500-999              5              2              97
      2              4              6        2          NaN              6              1             127
```



본격적인 변환에 앞서 experience 데이터를 살펴보면, 아래와 같은 데이터로 채워져 있네요.
저는 동호와 nan 을 처리 후, astype으로 수치형 데이터 변환을 시도할 예정입니다.

```
X_train['experience'].unique()
```

```
[15...] array(['>20', '15', '<1', '11', '5', '13', '7', '2', '1', '4', '10', '18',  
       '19', '12', '3', '16', '6', '9', '14', '8', '20', nan, '17'],  
      dtype=object)
```

+ Code

+ Markdown

전처리 실습

- 학습가능한 형태로 데이터 변경하기
 - NAN 데이터 찾기 : `isnull()`
 - 타입변경 : `astype()`

```
# '>20' => 21, '<1' => 0, nan => -1 로 변경해봅시다.
```

```
XX = X_train['experience'].copy()
```

```
XX[XX == '>20'] = 21
```

```
XX[XX == '<1'] = 0
```

```
XX[XX.isnull()] = -1
```

```
X_train['experience'] = XX.astype('int64')
```

전처리 실습

- 학습가능한 형태로 데이터 변경하기

```
[26]: # experience와 동일하게 company_size, company_type 모두 변환을 시도합니다.  
  
X_train['company_size'].unique()  
  
[26]: array([nan, '50-99', '<10', '1000-4999', '10000+', '100-500', '5000-9999',  
       '10/49', '500-999'], dtype=object)
```

```
XX = X_train['company_size'].copy()  
  
XX[XX == '10000+'] = 1  
XX[XX == '5000-9999'] = 2  
XX[XX == '1000-4999'] = 3  
XX[XX == '500-999'] = 4  
XX[XX == '100-500'] = 5  
XX[XX == '50-99'] = 6  
XX[XX == '10/49'] = 7  
XX[XX == '<10'] = 8  
XX[XX.isnull()] = -1  
  
X_train['company_size'] = XX.astype('int64')
```

+ Code

+ Markdown

전처리 실습

- 학습가능한 형태로 데이터 변경하기



```
X_train.info()  
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 15326 entries, 0 to 15325  
Data columns (total 11 columns):  
#   Column                                Non-Null Count  Dtype  
---  -  
0   city_development_index                15326 non-null  float64  
1   gender                                15326 non-null  int64  
2   relevent_experience                    15326 non-null  int64  
3   enrolled_university                  15326 non-null  int64  
4   education_level                       15326 non-null  int64  
5   major_discipline                     15326 non-null  int64  
6   experience                            15326 non-null  int64  
7   company_size                         15326 non-null  int64  
8   company_type                         15326 non-null  int64  
9   last_new_job                          14983 non-null  object  
10  training_hours                       15326 non-null  int64  
dtypes: float64(1), int64(9), object(1)  
memory usage: 1.3+ MB  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 3832 entries, 0 to 3831  
Data columns (total 11 columns):  
#   Column                                Non-Null Count  Dtype  
---  -  
0   city_development_index                3832 non-null  float64  
1   gender                                3832 non-null  int64  
2   relevent_experience                    3832 non-null  int64  
3   enrolled_university                  3832 non-null  int64  
4   education_level                       3832 non-null  int64  
5   major_discipline                     3832 non-null  int64  
6   experience                            3832 non-null  int64  
7   company size                         3832 non-null  int64
```

전처리 실습

- 학습가능한 형태로 데이터 변경하기

```
[30]: X_train['last_new_job'].unique()  
  
[30]: array(['1', '>4', 'never', '4', '3', '2', nan], dtype=object)
```

```
▶ XX = X_train['last_new_job'].copy()  
  
XX[XX == '>4'] = 5  
XX[XX == '4'] = 4  
XX[XX == '3'] = 3  
XX[XX == '2'] = 2  
XX[XX == '1'] = 1  
XX[XX == 'never'] = 0  
XX[XX.isnull()] = -1  
  
X_train['last_new_job'] = XX.astype('int64')
```

+ Code

+ Markdown

전처리 실습

- 학습가능한 형태로 데이터 변경하기

```
X_train.info()
X_test.info()
```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 15326 entries, 0 to 15325
Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	city_development_index	15326 non-null	float64
1	gender	15326 non-null	int64
2	relevent_experience	15326 non-null	int64
3	enrolled_university	15326 non-null	int64
4	education_level	15326 non-null	int64
5	major_discipline	15326 non-null	int64
6	experience	15326 non-null	int64
7	company_size	15326 non-null	int64
8	company_type	15326 non-null	int64
9	last_new_job	15326 non-null	int64
10	training_hours	15326 non-null	int64

dtypes: float64(1), int64(10)
memory usage: 1.3 MB

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3832 entries, 0 to 3831
Data columns (total 11 columns):

#	Column	Non-Null Count	Dtype
0	city_development_index	3832 non-null	float64
1	gender	3832 non-null	int64
2	relevent_experience	3832 non-null	int64
3	enrolled_university	3832 non-null	int64
4	education_level	3832 non-null	int64
5	major_discipline	3832 non-null	int64
6	experience	3832 non-null	int64
7	company_size	3832 non-null	int64
8	company_type	3832 non-null	int64
9	last_new_job	3832 non-null	int64
10	training_hours	3832 non-null	int64

dtypes: float64(1), int64(10)
memory usage: 329.4 KB

+ Code + Markdown

전처리 실습

■ 실습 문제 (2)

- 통신사 고객 이탈 예측 문제
- 평가를 위해 카테고리 정확도 (Categorization Accuracy) 를 사용
- 모델학습 : Logistic Regression

Description



과제 설명

통신 회사의 경우 신규 고객을 유치하는 동시에 수익을 창출 기반을 늘리기 위해 계약 해지(=이탈)를 피하는 것이 중요합니다. 신규 고객이 이탈하는 이유를 살펴보면 더 나은 가격, 더 흥미로운 패키지, 불편한 서비스 경험 또는 고객의 개인적인 상황 변화와 같이 다양한 이유로 고객이 계약을 종료 하게 됩니다. 고객 이탈 분석은 고객 이탈을 예측하고 이탈을 유발하는 근본적인 이유를 정의하는 기능을 제공합니다. 통신사는 기계학습 모델을 적용하여 개별 고객을 기준으로 이탈을 예측하고 할인, 특별제안 또는 기타 만족을 주기 위한 대응 조치를 취하여 고객을 유지할 수 있습니다. 여러분은 지금부터 수업시간에 배운 <머신러닝 기술>을 활용하여 test.csv 파일로 제공된 통신사 고객들의 이탈 여부를 예측하는 인공지능 SW를 작성해 주길 바랍니다.

학습 데이터로는 (통신사 고객 정보)와 해당 고객들의 (최종 이탈 여부)를 함께 제공합니다.

테스트 데이터로는 (통신사 고객 정보)만 제공하며, 예측된 고객의 (이탈 여부)는 submit.csv 파일로 저장하여 캐글 리더보드에 제출하셔야 합니다.

제공되는 (통신사 고객 정보)는 순서대로 customerId, gender, SeniorCitizen[고령자], Partner, Dependents[부양가족], tenure[계약유지기간], PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract[계약형태], PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn[이탈여부] 입니다.

답안 제출 시, 이탈여부는 1(Yes)과 0(No)으로 제출하셔야 합니다.

전처리 실습

- 실습 문제 (2)
 - 데이터

Overview Data Code Models Discussion Leaderboard Rules Host

Dataset Description

Edit

- 학습 데이터로는 (통신사 고객 정보)와 해당 고객들의 (최종 이탈 여부)를 함께 제공합니다.
- 테스트 데이터로는 (통신사 고객 정보)만 제공하며, 예측된 고객의 (이탈 여부)는 submit.csv 파일로 저장하여 캐글 리더보드에 제출하여야 합니다.
- 제공되는 (통신사 고객 정보)는 순서대로 **customerID**, **gender**, **SeniorCitizen**[고령자], **Partner**, **Dependents**[부양가족], **tenure**[계약유지기간], **PhoneService**, **MultipleLines**, **InternetService**, **OnlineSecurity**, **OnlineBackup**, **DeviceProtection**, **TechSupport**, **StreamingTV**, **StreamingMovies**, **Contract**[계약형태], **PaperlessBilling**, **PaymentMethod**, **MonthlyCharges**, **TotalCharges**, **Churn**[이탈여부] 입니다.
- 답안 제출 시, 이탈여부는 1(Yes)과 0(No)으로 제출하여야 합니다.



Detail Compact Column

23 of 23 columns ▾

	✓ PaperlessBilling	△ PaymentMethod	# MonthlyCharges	# TotalCharges	✓ Churn
54%	 true 2830 59% false 1958 41%	Electronic check 33% Mailed check 23% Other (2072) 43%	 18.3 119	 18.9 8.67k	 true 1269 27% false 3519 73%

전처리 실습

■ 학습 데이터 확인

```
[2]: train = pd.read_csv("/kaggle/input/2022-ml-w13p2/train.csv")
test = pd.read_csv("/kaggle/input/2022-ml-w13p2/test.csv")
```

```
▷ # 데이터 출력하여 확인

train
```

```
[3]:
```

	index	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	...	DeviceProtectio
0	0	1869	7010-BRBUU	Male	0	Yes	Yes	72	Yes	Yes	...	No intern servic
1	1	4528	9688-YGXVR	Female	0	No	No	44	Yes	No	...	Ye
2	2	6344	9286-DOJGF	Female	1	Yes	No	38	Yes	Yes	...	N
3	3	6739	6994-KERXL	Male	0	No	No	4	Yes	No	...	N
4	4	432	2181-UAESM	Male	0	No	No	2	Yes	No	...	Ye

전처리 실습

- 학습 데이터 X, Y 나누기

```
[4]: # Raw 데이터를 데이터와 라벨로 분리

X_train = train.drop(['Churn'], axis=1)
Y_train = train['Churn']
X_test = test
```

```
[3]:
```

Protection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn
No internet service	No internet service	No internet service	No internet service	Two year	No	Credit card (automatic)	24.10	1734.65	No
Yes	No	Yes	No	Month-to-month	Yes	Credit card (automatic)	88.15	3973.2	No
No	No	No	No	Month-to-month	Yes	Bank transfer (automatic)	74.95	2869.85	Yes

```
[7]: Y_train[(Y_train['Churn'] == 'No')] = 0
      Y_train[(Y_train['Churn'] == 'Yes')] = 1
```

전처리 실습

- 학습 데이터 살펴보며 전처리 필요한 부분 확인

△

데이터 출력하여 확인

train

[3]

	index	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	...	DeviceProtection
0	0	1869	7010-BRBUJ	Male	0	Yes	Yes	72	Yes	Yes	...	No internet service
1	1	4528	9688-YGXVR	Female	0	No	No	44	Yes	No	...	Yes
2	2	6344	9286-DOJGF	Female	1	Yes	No	38	Yes	Yes	...	No
3	3	6739	6994-KERXL	Male	0	No	No	4	Yes	No	...	No
4	4	432	2181-UAESM	Male	0	No	No	2	Yes	No	...	Yes
...
4783	5981	3772	0684-AOSIH	Male	0	Yes	No	1	Yes	No	...	No
4784	5982	5191	5982-PSMKW	Female	0	Yes	Yes	23	Yes	Yes	...	Yes
4785	5983	5226	8044-BGWPI	Male	0	Yes	Yes	12	Yes	No	...	No internet service
4786	5984	5390	7450-NWRTR	Male	1	No	No	12	Yes	Yes	...	Yes
4787	5985	860	4795-UXVCJ	Male	0	No	No	26	Yes	No	...	No internet service
4788 rows x 23 columns												

전처리 실습

- 학습 데이터 살펴보며 전처리 필요한 부분 확인

[3]:

	index	Unnamed: 0	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	...	DeviceProtectio
0	0	1869	7010-BRBUU	Male	0	Yes	Yes	72	Yes	Yes	...	No intern servic
1	1	4528	9688-YGXVR	Female	0	No	No	44	Yes	No	...	Ye
2	2	6344	9286-DOJGF	Female	1	Yes	No	38	Yes	Yes	...	N
3	3	6739	6994-KERXL	Male	0	No	No	4	Yes	No	...	N
4	4	432	2181-UAESM	Male	0	No	No	2	Yes	No	...	Ye
...
4783	5981	3772	0684-AOSIH	Male	0	Yes	No	1	Yes	No	...	N

[5]:

데이터를 출력하여 유의미하지 않을 것으로 생각되는 데이터를 삭제한다.

```
X_train = X_train.drop(['index', 'Unnamed: 0', 'customerID'], axis=1)
X_test = X_test.drop(['index', 'Unnamed: 0', 'customerID'], axis=1)
```

전처리 실습

- 학습 데이터 살펴보고 전처리 필요한 부분 확인
 - Object 타입의 변환

(2) "범주형 데이터" => "숫치형 데이터" 변환하기

```
[14]: # 데이터 Feature 타입을 확인
# 학습가능한 타입으로 변경 (예. 문자열=>실수형 or 정수형)
# 'TotalCharges' 타입에 차이가 존재함
```

```
X_train.info()
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4788 entries, 0 to 4787
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   gender                 4788 non-null   object
1   SeniorCitizen          4788 non-null   int64
2   Partner                4788 non-null   object
3   Dependents             4788 non-null   object
4   tenure                 4788 non-null   int64
5   PhoneService           4788 non-null   object
6   MultipleLines           4788 non-null   object
7   InternetService        4788 non-null   object
8   OnlineSecurity         4788 non-null   object
9   OnlineBackup           4788 non-null   object
10  DeviceProtection       4788 non-null   object
11  TechSupport            4788 non-null   object
12  StreamingTV            4788 non-null   object
13  StreamingMovies        4788 non-null   object
14  Contract               4788 non-null   object
15  PaperlessBilling       4788 non-null   object
16  PaymentMethod          4788 non-null   object
17  MonthlyCharges         4788 non-null   float64
18  TotalCharges           4788 non-null   object
dtypes: float64(1), int64(2), object(16)
memory usage: 710.8+ KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1198 entries, 0 to 1197
Data columns (total 19 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   gender                 1198 non-null   int64
1   SeniorCitizen          1198 non-null   int64
2   Partner                1198 non-null   int64
3   Dependents             1198 non-null   int64
4   tenure                 1198 non-null   int64
5   PhoneService           1198 non-null   int64
6   MultipleLines           1198 non-null   int64
7   InternetService        1198 non-null   int64
8   OnlineSecurity         1198 non-null   int64
9   OnlineBackup           1198 non-null   int64
10  DeviceProtection       1198 non-null   int64
11  TechSupport            1198 non-null   int64
12  StreamingTV            1198 non-null   int64
13  StreamingMovies        1198 non-null   int64
14  Contract               1198 non-null   int64
15  PaperlessBilling       1198 non-null   int64
16  PaymentMethod          1198 non-null   int64
17  MonthlyCharges         1198 non-null   float64
18  TotalCharges           1198 non-null   float64
dtypes: float64(2), int64(17)
memory usage: 178.0 KB
```

전처리 실습

- 학습 데이터 살펴보며 전처리 필요한 부분 확인
 - Object 타입의 변환

```
from sklearn.preprocessing import LabelEncoder
```

```
columns = ['gender', 'Partner', 'Dependents', 'tenure',  
           'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity',  
           'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV',  
           'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod']
```

```
for column in columns:  
    le = LabelEncoder()  
    X_train[column] = le.fit_transform(X_train[column].values)  
    X_test[column] = le.transform(X_test[column].values)
```

⇒

```
X_train.info()  
X_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 4788 entries, 0 to 4787  
Data columns (total 19 columns):  
#   Column                Non-Null Count  Dtype  
---  ---  
0   gender                 4788 non-null   int64  
1   SeniorCitizen          4788 non-null   int64  
2   Partner                4788 non-null   int64  
3   Dependents             4788 non-null   int64  
4   tenure                 4788 non-null   int64  
5   PhoneService           4788 non-null   int64  
6   MultipleLines          4788 non-null   int64  
7   InternetService        4788 non-null   int64  
8   OnlineSecurity         4788 non-null   int64  
9   OnlineBackup           4788 non-null   int64  
10  DeviceProtection       4788 non-null   int64  
11  TechSupport            4788 non-null   int64  
12  StreamingTV            4788 non-null   int64  
13  StreamingMovies        4788 non-null   int64  
14  Contract               4788 non-null   int64  
15  PaperlessBilling       4788 non-null   int64  
16  PaymentMethod          4788 non-null   int64  
17  MonthlyCharges         4788 non-null   float64  
18  TotalCharges           4788 non-null   object  
dtypes: float64(1), int64(17), object(1)  
memory usage: 710.8+ KB
```

전처리 실습

- 학습 데이터 살펴보며 전처리 필요한 부분 확인

```
# 'TotalCharges' 변환 시도
# float타입으로 변경하려 했으나 280번째에 " " 공백이 포함되어 변경이 안되는것을 확인
X_train["TotalCharges"] = pd.to_numeric(X_train["TotalCharges"], downcast="float")
# X_train["TotalCharges"] = X_train["TotalCharges"].astype('float64')
```

```
ValueError                                Traceback (most recent call last)
/opt/conda/lib/python3.7/site-packages/pandas/_libs/lib.pyx in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string " "

During handling of the above exception, another exception occurred:

ValueError                                Traceback (most recent call last)
/tmp/ipykernel_33/1421859493.py in <module>
      1 # 'TotalCharges' 변환 시도
      2 # float타입으로 변경하려 했으나 280번째에 " " 공백이 포함되어 변경이 안되는것을 확인
----> 3 X_train["TotalCharges"] = pd.to_numeric(X_train["TotalCharges"], downcast="float")
      4 # X_train["TotalCharges"] = X_train["TotalCharges"].astype('float64')

/opt/conda/lib/python3.7/site-packages/pandas/core/tools/numeric.py in to_numeric(arg, errors, downcast)
    182     try:
    183         values, _ = lib.maybe_convert_numeric(
--> 184             values, set(), coerce_numeric=coerce_numeric
    185         )
    186     except (ValueError, TypeError):

/opt/conda/lib/python3.7/site-packages/pandas/_libs/lib.pyx in pandas._libs.lib.maybe_convert_numeric()

ValueError: Unable to parse string " " at position 280
```

```
# 학습데이터 280번째 데이터 눈으로 확인 => ' '
X_train['TotalCharges'][280]
```


전처리 실습

■ 학습 데이터 살펴보며 전처리 필요한 부분 확인

(3) Empty 데이터 처리하기

- 비중이 높지 않다면 학습데이터에서 삭제하기 (가장 Naive한 방법론)

```
# Empty 데이터 숫자 확인하기
(X_train['TotalCharges'] == ' ').sum()
```

[16]: 10

```
[17]: # X_train 에 ' ' 값으로 채워진 데이터 삭제

drop_idx = X_train[X_train['TotalCharges'] == ' '].index
drop_idx

[17]: Int64Index([280, 512, 2213, 2470, 2617, 3457, 4297, 4303, 4555, 4759], dtype='int64')
```

```
# 'TotalCharges' 데이터가 비어있는 데이터 삭제
X_train = X_train.drop(drop_idx, axis=0)
Y_train = Y_train.drop(drop_idx, axis=0)
```

+ Code + Markdown

```
[19]: X_train["TotalCharges"] = X_train["TotalCharges"].astype('float64')
```

```
[26]: X_train.info()
X_test.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 4778 entries, 0 to 4787
Data columns (total 19 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   gender                4778 non-null   int64
 1   SeniorCitizen         4778 non-null   int64
 2   Partner               4778 non-null   int64
 3   Dependents            4778 non-null   int64
 4   tenure                4778 non-null   int64
 5   PhoneService          4778 non-null   int64
 6   MultipleLines          4778 non-null   int64
 7   InternetService       4778 non-null   int64
 8   OnlineSecurity        4778 non-null   int64
 9   OnlineBackup          4778 non-null   int64
10   DeviceProtection      4778 non-null   int64
11   TechSupport           4778 non-null   int64
12   StreamingTV           4778 non-null   int64
13   StreamingMovies       4778 non-null   int64
14   Contract              4778 non-null   int64
15   PaperlessBilling      4778 non-null   int64
16   PaymentMethod         4778 non-null   int64
17   MonthlyCharges        4778 non-null   float64
18   TotalCharges          4778 non-null   float64
dtypes: float64(2), int64(17)
memory usage: 746.6 KB
```


전처리 실습

- 학습 데이터 살펴보며 전처리 필요한 부분 확인
 - 빈 데이터 채우는 방법은?

데이터 정제 (Data Cleaning)

- 결측 데이터 채우기 (Empty Values)
 - 결측 데이터: np.nan, npNaN, None
 - 평균(mean), 중위수(median), 최빈수(most frequent value)로 대체하는 기법 사용
 - 사용가능함수
 - sklearn의 Imputer(): 입력인자로 평균, 중위수, 최빈수 선택

```
[ ] 1 # 결측자료 대체 =====  
2 x_miss=np.array([[1,2,3,None],[5,np.NaN,7,8],[None,10,11,12],[13,np.nan,15,16]])  
3 x_miss
```

```
array([[1, 2, 3, None],  
       [5, nan, 7, 8],  
       [None, 10, 11, 12],  
       [13, nan, 15, 16]], dtype=object)
```

```
1 from sklearn.preprocessing import Imputer  
2 im=Imputer(strategy='mean')  
3 im.fit_transform(x_miss) # 열의 평균값으로 대체
```

```
Cr:\Anaconda3\lib\site-packages\sklearn\utils\deprecation.py:66: DeprecationWarning: Class Imputer is deprecated; Im  
warnings.warn(msg, category=DeprecationWarning)  
array([[ 1.,  2.,  3., 12.],  
       [ 5.,  6.,  7.,  8.],  
       [ 6.33333333, 10., 11., 12.],  
       [13.,  6., 15., 16.]])
```