데이터프로그래밍 기초 8일차

2026-1 DS Bootcamp

부산대학교
데이터사이언스전문대학원
석사과정 박민서

PUSAN NATIONAL UNIVERSITY
Graduate School of Data Science

CONTENTS

▪ 데이터 분석 기초 with Pandas

- .describe( ): DataFrame의 수치 데이터에 대한 기초 통계량 제시
- “include=‘all’”: 수치 + 다른 자료형에 대한 기초 통계량 제시

```
# 모든 자료형에 대한 요약 정보
df.describe(include='all')
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
count	150.000000	150.000000	150.000000	150.000000	150.000000	150
unique	NaN	NaN	NaN	NaN	NaN	3
top	NaN	NaN	NaN	NaN	NaN	Iris-setosa
freq	NaN	NaN	NaN	NaN	NaN	50
mean	75.500000	5.843333	3.054000	3.758667	1.198667	NaN
std	43.445368	0.828066	0.433594	1.764420	0.763161	NaN
min	1.000000	4.300000	2.000000	1.000000	0.100000	NaN
25%	38.250000	5.100000	2.800000	1.600000	0.300000	NaN
50%	75.500000	5.800000	3.000000	4.350000	1.300000	NaN
75%	112.750000	6.400000	3.300000	5.100000	1.800000	NaN
max	150.000000	7.900000	4.400000	6.900000	2.500000	NaN

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .count( ): DataFrame, Series가 가진 셀 수 출력
- None, NaN, NaT, np.inf와 같은 결측치는 제외되고 출력

```
# 행 개수 출력
df.count()
```

```
Id               150
SepalLengthCm    150
SepalWidthCm     150
PetalLengthCm    150
PetalWidthCm     150
Species          150
dtype: int64
```

```
# 특정 열의 행 개수 출력
df["SepalLengthCm"].count()
```

```
np.int64(150)
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .max( ): DataFrame이나 Series의 최댓값 출력
- .idxmax( ): 최댓값을 저장한 데이터의 index를 출력

```
# 데이터프레임의 최댓값들을 출력
df.max()
```

```
Id                       150
SepalLengthCm            7.9
SepalWidthCm             4.4
PetalLengthCm            6.9
PetalWidthCm             2.5
Species          Iris-virginica
dtype: object
```

```
# 특정 열의 최댓값 출력
df["SepalLengthCm"].max()
```

```
np.float64(7.9)
```

```
# 특정 열의 최댓값을 가진 행 index 출력
df["SepalLengthCm"].idxmax()
```

```
131
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .min( ): DataFrame이나 Series의 최솟값 출력
- .idxmin( ): 최솟값을 저장한 데이터의 index를 출력

```
# 데이터프레임의 최솟값들을 출력
df.min()
```

✓ 0.0s

```
Id                      1
SepalLengthCm         4.3
SepalWidthCm          2.0
PetalLengthCm         1.0
PetalWidthCm          0.1
Species       Iris-setosa
dtype: object
```

```
# 특정 열의 최솟값 출력
df["SepalLengthCm"].min()
```

✓ 0.0s

```
np.float64(4.3)
```

```
# 특정 열의 최솟값을 가진 행 index 출력
df["SepalLengthCm"].idxmin()
```

✓ 0.0s

```
13
```

▪ 데이터 분석 기초 with Pandas

- .sum( ): DataFrame이나 Series의 합계 출력
- .mean( ): DataFrame이나 Series의 평균 출력
- .std( ): DataFrame이나 Series의 표준편차 출력

```
# 특정 열의 합계 출력
# str Series는 각 문자열이 연결된 상태로 출력
df["SepalLengthCm"].sum()
```

✓ 0.0s

```
np.float64(876.5)
```

```
# 특정 열의 평균 출력
df["SepalLengthCm"].mean()
```

✓ 0.0s

```
np.float64(5.843333333333334)
```

```
# 특정 열의 표준편차 출력
df["SepalLengthCm"].std()
```

✓ 0.0s

```
np.float64(0.828066127977863)
```

▪ 데이터 분석 기초 with Pandas

- .unique( ): 데이터의 열 속의 고윳값들을 출력 (범주형 자료에 용이)
- .nunique( ): 고윳값의 개수를 출력

```
# 범주형 자료의 고윳값 출력
df["Species"].unique()
```

✓ 0.0s

```
<StringArray>
['Iris-setosa', 'Iris-versicolor', 'Iris-virginica']
Length: 3, dtype: str
```

```
# 범주형 자료의 고윳값 개수 출력
df["Species"].nunique()
```

✓ 0.0s

```
3
```

▪ 데이터 분석 기초 with Pandas

- 파생변수 생성

```
# broadcast 적용
df["SepalLengthCm"] * 10
```

✓ 0.0s

```
0      51.0
1      49.0
2      47.0
3      46.0
4      50.0
       ...
145    67.0
146    63.0
147    65.0
148    62.0
149    59.0
Name: SepalLengthCm, Length: 150, dtype: float64
```

```
# 특정 열을 활용한 파생변수 생성 (broadcast)
df["SepalLengthMm"] = (df["SepalLengthCm"] * 10).astype(int)
df["SepalLengthMm"]
```

✓ 0.0s

```
0      51
1      49
2      47
3      46
4      50
       ..
145    67
146    63
147    65
148    62
149    59
Name: SepalLengthMm, Length: 150, dtype: int64
```

▪ 데이터 분석 기초 with Pandas

- 파생변수 생성

```
# 여러 개의 열을 활용한 파생 변수 생성
df["SepalLengthCm"] + df["SepalWidthCm"]
```

✓ 0.0s

```
0      8.6
1      7.9
2      7.9
3      7.7
4      8.6
      ...
145    9.7
146    8.8
147    9.5
148    9.6
149    8.9
Length: 150, dtype: float64
```

```
# 여러 개의 열을 활용한 파생 변수 생성
df["Sepal"] = df["SepalLengthCm"] + df["SepalWidthCm"]
df["Sepal"]
```

✓ 0.0s

```
0      8.6
1      7.9
2      7.9
3      7.7
4      8.6
      ...
145    9.7
146    8.8
147    9.5
148    9.6
149    8.9
Name: Sepal, Length: 150, dtype: float64
```

▪ 데이터 분석 기초 with Pandas

- .groupby( ): 그룹별로 DataFrame을 집계하거나 요약

```
# .groupby() 활용
df.groupby(["Species"])
```

✓ 0.0s

```
<pandas.api.typing.DataFrameGroupBy object at 0x0000016625515A90>
```

```
# .groupby() 활용
# Species를 기준으로 한 그룹의 평균
df.groupby(["Species"]).mean()
```

✓ 0.0s

Species	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	SepalLengthMm	Sepal
Iris-setosa	25.5	5.006	3.418	1.464	0.244	50.06	8.424
Iris-versicolor	75.5	5.936	2.770	4.260	1.326	59.36	8.706
Iris-virginica	125.5	6.588	2.974	5.552	2.026	65.88	9.562

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .groupby( ): 그룹별로 DataFrame을 집계하거나 요약

```
# Species를 기준으로 SepalLengthCm의 최솟값
# 단일 열에 대해서 통계량 출력 시 Series로 반환
# DataFrame 형태로 담고 싶다면 .to_frame()
df.groupby(["Species"])["SepalLengthCm"].min()
```

✓ 0.0s

```
Species
Iris-setosa        4.3
Iris-versicolor    4.9
Iris-virginica     4.9
Name: SepalLengthCm, dtype: float64
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .groupby( ): 그룹별로 DataFrame을 집계하거나 요약

```
# Species, PetalWidthCm를 기준으로 SepalLengthCm의 최댓값
# 단일 열에 대해서 통계량 출력 시 Series로 반환
# DataFrame 형태로 담고 싶다면 .to_frame()
df.groupby(["Species", "PetalWidthCm"])["SepalLengthCm"].max().to_frame()
```

✓ 0.0s

		SepalLengthCm
Species	PetalWidthCm	
Iris-setosa	0.1	5.2
	0.2	5.8
	0.3	5.7
	0.4	5.7
	0.5	5.1
	0.6	5.0
Iris-versicolor	1.0	6.0
	1.1	5.6
	1.2	6.1
	1.3	6.6
	1.4	7.0
	1.5	6.9
	1.6	6.3
	1.7	6.7
	1.8	5.9
Iris-virginica	1.4	6.1

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .groupby( ): 그룹별로 DataFrame을 집계하거나 요약
- .agg( ): 원하는 통계량들만 모아서 출력 가능

```
# Species, PetalWidthCm를 기준으로 SepalLengthCm의 통계량 집합
# .agg([통계량 리스트]) 형태로 사용
df.groupby(["Species", "PetalWidthCm"])["SepalLengthCm"].agg([min, max, np.mean, np.median])
```

✓ 0.0s

		min	max	mean	median
Species	PetalWidthCm				
Iris-setosa	0.1	4.3	5.2	4.833333	4.90
	0.2	4.4	5.8	4.975000	5.00
	0.3	4.5	5.7	4.971429	5.00
	0.4	5.0	5.7	5.300000	5.40
	0.5	5.1	5.1	5.100000	5.10
	0.6	5.0	5.0	5.000000	5.00
Iris-versicolor	1.0	4.9	6.0	5.414286	5.50
	1.1	5.1	5.6	5.400000	5.50
	1.2	5.5	6.1	5.780000	5.80
	1.3	5.5	6.6	5.884615	5.70
	1.4	5.2	7.0	6.357143	6.60
	1.5	5.4	6.9	6.190000	6.25

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 분석 기초 with Pandas

- .to_csv( ): DataFrame을 .csv파일로 저장
- 'index=False' 옵션을 통한 인덱스 저장 유무 선택

```
# 인덱스 제외 후 데이터 저장
df.to_csv("./Iris_work.csv", index=False)
```

Iris_work.csv > data

```
1   Id,SepalLengthCm,SepalWidthCm,PetalLengthCm,PetalWidthCm,Species,SepalLengthMm,Sepal
2   1,5.1,3.5,1.4,0.2,Iris-setosa,51,8.6
3   2,4.9,3.0,1.4,0.2,Iris-setosa,49,7.9
4   3,4.7,3.2,1.3,0.2,Iris-setosa,47,7.9
5   4,4.6,3.1,1.5,0.2,Iris-setosa,46,7.699999999999999
6   5,5.0,3.6,1.4,0.2,Iris-setosa,50,8.6
7   6,5.4,3.9,1.7,0.4,Iris-setosa,54,9.3
8   7,4.6,3.4,1.4,0.3,Iris-setosa,46,8.0
9   8,5.0,3.4,1.5,0.2,Iris-setosa,50,8.4
10  9,4.4,2.9,1.4,0.2,Iris-setosa,44,7.300000000000001
11  10,4.9,3.1,1.5,0.1,Iris-setosa,49,8.0
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

Data Preprocessing

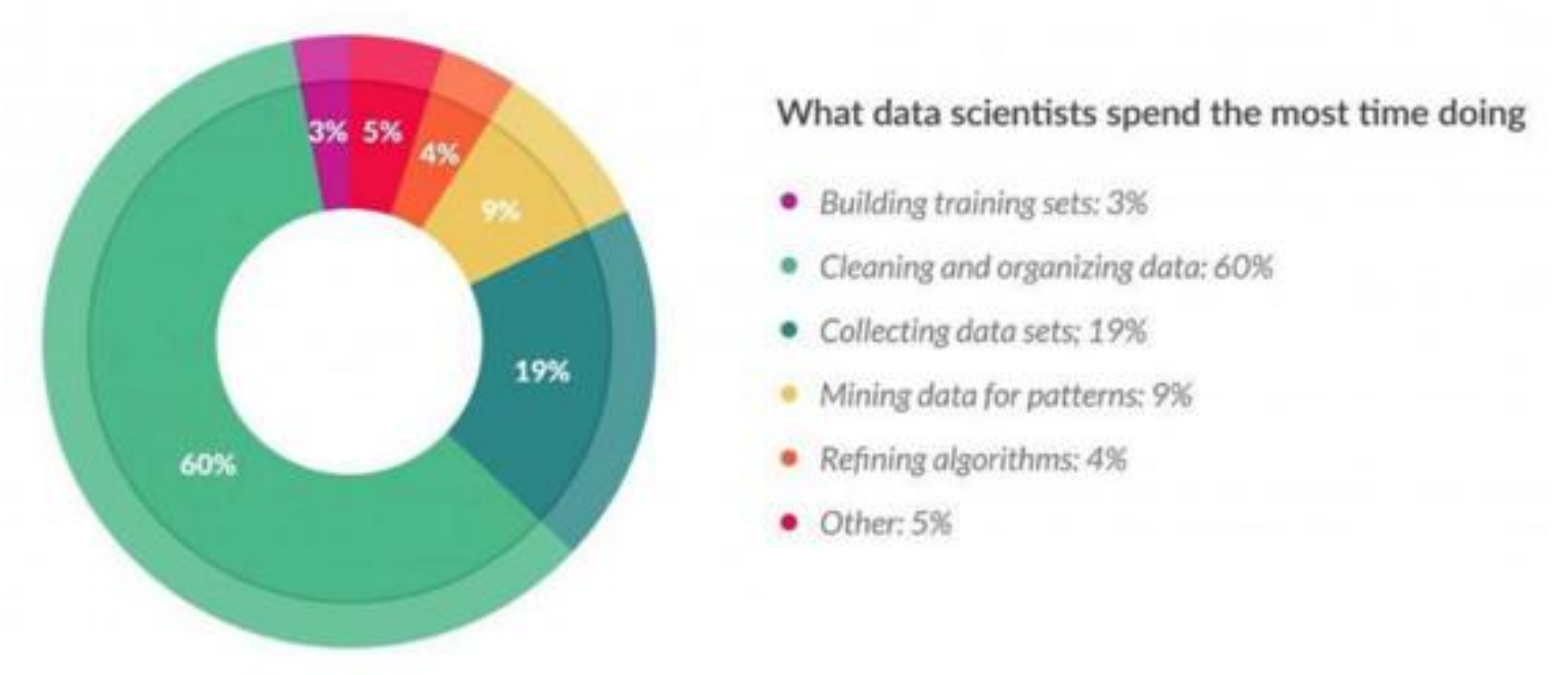
▪ 데이터 수집

- 공개 데이터 수집 가능한 사이트
- Competition: https://www.kaggle.com/ https://dacon.io/
- 공공데이터포털: https://www.data.go.kr/
- AI-Hub: https://www.aihub.or.kr/

Data Preprocessing

▪ 데이터 전처리 (Preprocessing)

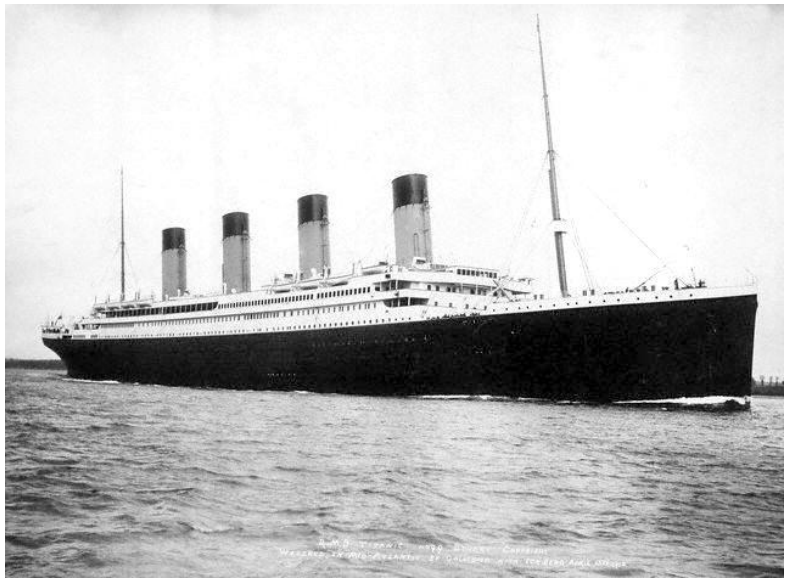
- 데이터 분석 전 과정의 80%는 “데이터 준비”와 “데이터 정제”에 쏟아 붓는다.
- 분석 결과의 인사이트와 모델의 성능에 직접적인 영향을 미치는 과정



https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#7a9598f66f63

Data Preprocessing

Titanic Data



- https://www.kaggle.com/competitions/titanic/data
- 1912년에 발생한 타이타닉 호 침몰 사고에서의 승객들의 생존 정보
- 주요 Column 정보:
- PassengerId: 승객 고유 ID
- Survived: 승객이 생존(1)했는지 사망(0)했는지 여부
- Pclass: 선실 등급(1:1등급, 2:2등급, 3:3등급)
- Name: 이름 Sex: 성별 Age: 나이 SibSp: 형제자매/배우자의 수
- Parch: 부모/자녀의 수
- Ticket: 티켓 번호 Fare: 티켓에 대해 지불한 금액
- Cabin: 객실 카테고리
- Embarked: 승객이 탑승한 항구(C = Cherbourg, Q = Queenstown, S = Southampton)

Q&A

Data Preprocessing

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우

```
df = pd.read_csv("./titanic_train.csv")
df
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우

```
df.info()
✓ 0.0s
<class 'pandas.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    str
 4   Sex          891 non-null    str
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    str
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    str
 11  Embarked     889 non-null    str
dtypes: float64(2), int64(5), str(5)
memory usage: 83.7 KB
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우
- .isna( ), isnull( ): 해당 데이터가 결측치가 맞는지를 반환
- .notna( ), .notnall( ): 해당 데이터가 결측치가 아닌지를 반환

```
# 해당 데이터가 결측치인지 아닌지 반환
df.isna()
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	False	False	False	False	False	False	False	False	False	False	True	False
1	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	True	False
3	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	True	False
...	...	...	...	...	...	...	...	...	...	...	...	...
886	False	False	False	False	False	False	False	False	False	False	True	False
887	False	False	False	False	False	False	False	False	False	False	False	False
888	False	False	False	False	False	True	False	False	False	False	True	False
889	False	False	False	False	False	False	False	False	False	False	False	False
890	False	False	False	False	False	False	False	False	False	False	True	False

891 rows × 12 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우
- .isna( ), isnull( ): 해당 데이터가 결측치가 맞는지를 반환
- .notna( ), .notnall( ): 해당 데이터가 결측치가 아닌지를 반환

```
# 조건부 인덱싱으로 사용 가능
df[df["Age"].isna()]
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
17	18	1	2	Williams, Mr. Charles Eugene	male	NaN	0	0	244373	13.0000	NaN	S
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C
26	27	0	3	Emir, Mr. Farred Chehab	male	NaN	0	0	2631	7.2250	NaN	C
28	29	1	3	O'Dwyer, Miss. Ellen "Nellie"	female	NaN	0	0	330959	7.8792	NaN	Q
...	...	...	...	...	...	...	...	...	...	...	...	...
859	860	0	3	Razi, Mr. Raihed	male	NaN	0	0	2629	7.2292	NaN	C
863	864	0	3	Sage, Miss. Dorothy Edith "Dolly"	female	NaN	8	2	CA. 2343	69.5500	NaN	S
868	869	0	3	van Melkebeke, Mr. Philemon	male	NaN	0	0	345777	9.5000	NaN	S
878	879	0	3	Laleff, Mr. Kristo	male	NaN	0	0	349217	7.8958	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S

177 rows × 12 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우
- .isna( ), isnull( ): 해당 데이터가 결측치가 맞는지를 반환
- .notna( ), .notnall( ): 해당 데이터가 결측치가 아닌지를 반환

```
# 조건부 인덱싱으로 사용 가능
# ~ = "not"
df[~df["Age"].isna()]
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

714 rows × 12 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- 결측치 (Missing Value): 사용자에 의해서, 관측이 되지 않아서 데이터에 값이 기록되지 않은 경우
- .isna( ), isnull( ): 해당 데이터가 결측치가 맞는지를 반환
- .notna( ), .notnall( ): 해당 데이터가 결측치가 아닌지를 반환

```
# 'Age'열의 결측치 개수 확인
df['Age'].isna().sum()
```

✓ 0.0s

```
np.int64(177)
```

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# index 번호가 3인 행 제거
df.drop(3).head()
# index 번호가 1, 2, 3인 행 제거
# df.drop([1, 2, 3]).head()
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name
0	1	0	3	Braund, Mr. Owen Harris
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...
2	3	1	3	Heikkinen, Miss. Laina
4	5	0	3	Allen, Mr. William Henry
5	6	0	3	Moran, Mr. James

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# index 번호가 3인 행 제거
# df.drop(3).head()
# index 번호가 1, 2, 3인 행 제거
df.drop([1, 2, 3]).head()
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name
0	1	0	3	Braund, Mr. Owen Harris
4	5	0	3	Allen, Mr. William Henry
5	6	0	3	Moran, Mr. James
6	7	0	1	McCarthy, Mr. Timothy J
7	8	0	3	Palsson, Master. Gosta Leonard

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# 'Age'열 제거
df.drop(columns="Age")
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	1	0	PC 17599	71.2833	C85	C

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# 'Age', 'Cabin'열 제거
df.drop(columns=["Age", "Cabin"])
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	1	0	A/5 21171	7.2500	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	1	0	PC 17599	71.2833	C
2	3	1	3	Heikkinen, Miss. Laina	female	0	0	STON/O2. 3101282	7.9250	S

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# 결측치 있는 행 제거
df.dropna()
# 결측치 있는 열 제거
# df.dropna(axis=1)
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

183 rows × 12 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .drop( ): DataFrame에서의 특정 행, 열을 제거하는 함수
- .dropna( ): 결측치가 포함된 행, 열을 제거하는 함수

```
# 결측치 있는 행 제거
# df.dropna()
# 결측치 있는 열 제거
df.dropna(axis=1)
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	0	0	373450	8.0500
...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	0	0	211536	13.0000
887	888	1	1	Graham, Miss. Margaret Edith	female	0	0	112053	30.0000
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	1	2	W./C. 6607	23.4500
889	890	1	1	Behr, Mr. Karl Howell	male	0	0	111369	30.0000
890	891	0	3	Dooley, Mr. Patrick	male	0	0	370376	7.7500

891 rows × 9 columns

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

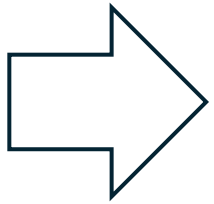
▪ 데이터 전처리 with Pandas

- .fillna( ): 결측치 데이터를 채울 때 사용
- .ffill( ), .bfill( ): 결측치 데이터의 앞의 값, 뒤의 값으로 대체할 때 사용

```
# "Age" 열의 결측치 데이터를 30으로 채우기
display(df["Age"])
df["Age"].fillna(30)
```

✓ 0.0s

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```



```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

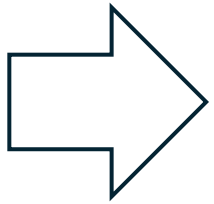
▪ 데이터 전처리 with Pandas

- .fillna( ): 결측치 데이터를 채울 때 사용
- .ffill( ), .bfill( ): 결측치 데이터의 앞의 값, 뒤의 값으로 대체할 때 사용

```
# "Age" 열의 결측치 데이터를 평균으로 채우기
display(df["Age"])
df["Age"].fillna(df["Age"].mean())
```

✓ 0.0s

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```



```
0      22.000000
1      38.000000
2      26.000000
3      35.000000
4      35.000000
         ...
886    27.000000
887    19.000000
888    29.699118
889    26.000000
890    32.000000
Name: Age, Length: 891, dtype: float64
```

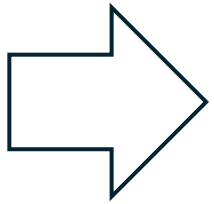
김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .fillna( ): 결측치 데이터를 채울 때 사용
- .ffill( ), .bfill( ): 결측치 데이터의 앞의 값, 뒤의 값으로 대체할 때 사용

```
# "Age" 열의 결측치 데이터를 결측 앞의 값으로 채우기
display(df["Age"])
df["Age"].ffill()
# df["Age"].fillna(method="ffill") [이전 버전 코드]
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```



```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    19.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

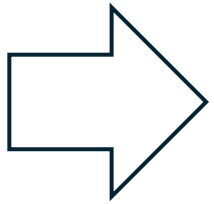
김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .fillna( ): 결측치 데이터를 채울 때 사용
- .ffill( ), .bfill( ): 결측치 데이터의 앞의 값, 뒤의 값으로 대체할 때 사용

```
# "Age" 열의 결측치 데이터를 결측 뒤의 값으로 채우기
display(df["Age"])
df["Age"].bfill()
# df["Age"].fillna(method="bfill") [이전 버전 코드]
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```



```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888    26.0
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 데이터 전처리 with Pandas

- .duplicated( ): 데이터 속 중복이 있는지 없는지를 반환

```
# 중복 값 유무 확인
df.duplicated()
# keep = [옵션]
# 'first' = 중복 표시를 처음 행만 False, 나머지는 True
# 'last' = 중복 표시를 마지막 행만 False, 나머지는 True
# False = 중복된 모든 데이터 True
df['Age'].duplicated(keep='first')
```

✓ 0.0s

```
0      False
1      False
2      False
3      False
4       True
       ...
886     True
887     True
888     True
889     True
890     True
Name: Age, Length: 891, dtype: bool
```

Data Preprocessing

▪ 데이터 전처리 with Pandas

- .drop_duplicates( ): 중복된 데이터 제거
- ‘keep = [옵션]’, ‘subset = [옵션]’에 따른 중복 제거 방법 설정

```
# 전체 속 중복 데이터 제거
# 'Age'와 'Parch'의 값이 같은 경우
# 처음 행만 남기고 이후 중복 행은 제거
df.drop_duplicates(keep='first', subset=['Age', 'Parch'])
```

✓ 0.0s

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.00	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.00	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.00	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.00	1	0	113803	53.1000	C123	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
...	...	...	...	...	...	...	...	...	...	...	...	...
831	832	1	2	Richards, Master. George Sibley	male	0.83	1	1	29106	18.7500	NaN	S
843	844	0	3	Lemberopolous, Mr. Peter L	male	34.50	0	0	2683	6.4375	NaN	C
851	852	0	3	Svensson, Mr. Johan	male	74.00	0	0	347060	7.7750	NaN	S
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.00	1	1	11751	52.5542	D35	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.00	0	1	11767	83.1583	C50	C

179 rows × 12 columns

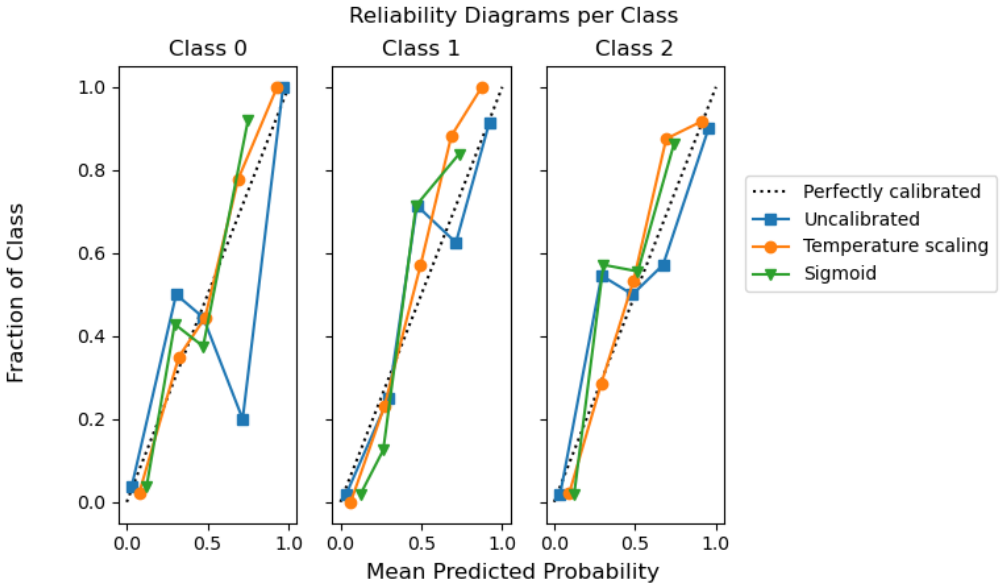
김유지, 「어쩌다 데이터 분석 with 파이썬」, 한빛미디어, 2022, 90-279쪽

▪ 실습 1

- Titanic 데이터를 활용하여 다음과 같은 작업을 수행하세요
  - 데이터 구조를 파악하세요.
    - 데이터 개수, 행 개수, 결측치 개수 등
  - 데이터에서 18살 이상의 생존자 승객만 선택해서 출력하세요
  - 데이터에서 'Embarked'가 'S'이거나 'C' 인 행만 선택해서 출력하세요.
  - 성별('Sex')별로 생존율을 출력하세요.
  - 성별('Sex') 및 객실 등급('Pclass')을 기준으로 각 그룹별 운임('Fare')의 평균, 최대, 최소를 계산하세요.

▪ 사이킷런 (Scikit-learn) 소개

- Python에서 다양한 ML 알고리즘을 다뤄볼 수 있는 대표적인 ML 라이브러리
- Pandas보다 효과적인 데이터 전처리 도구 제공
- Model을 평가할 수 있는 지표, 평가 방법 (ex. 교차 검증 등) 제공
- https://scikit-learn.org/stable/api/index.html



https://scikit-learn.org/stable/auto_examples/release_highlights/plot_release_highlights_1_8_0.html

Homework

▪ 8일차 과제

- GitHub 사이트에서 “8일차_과제.ipynb” 다운로드
- 코드 작성 후, “**본인이름**_8일차_과제.ipynb”로 저장
- 저장한 과제 파일 전송 (이메일 주소: minsuh99@pusan.ac.kr)

  기한: ~ 2/12 PM 13:59:59

Q&A