EDA 및 전처리

2. 전처리

SK 네트웍스

Data Insight

개요

분석을 위한 데이터 구조 만들기

결측치, 이상치 탐색 및 조치

Feature Engineering

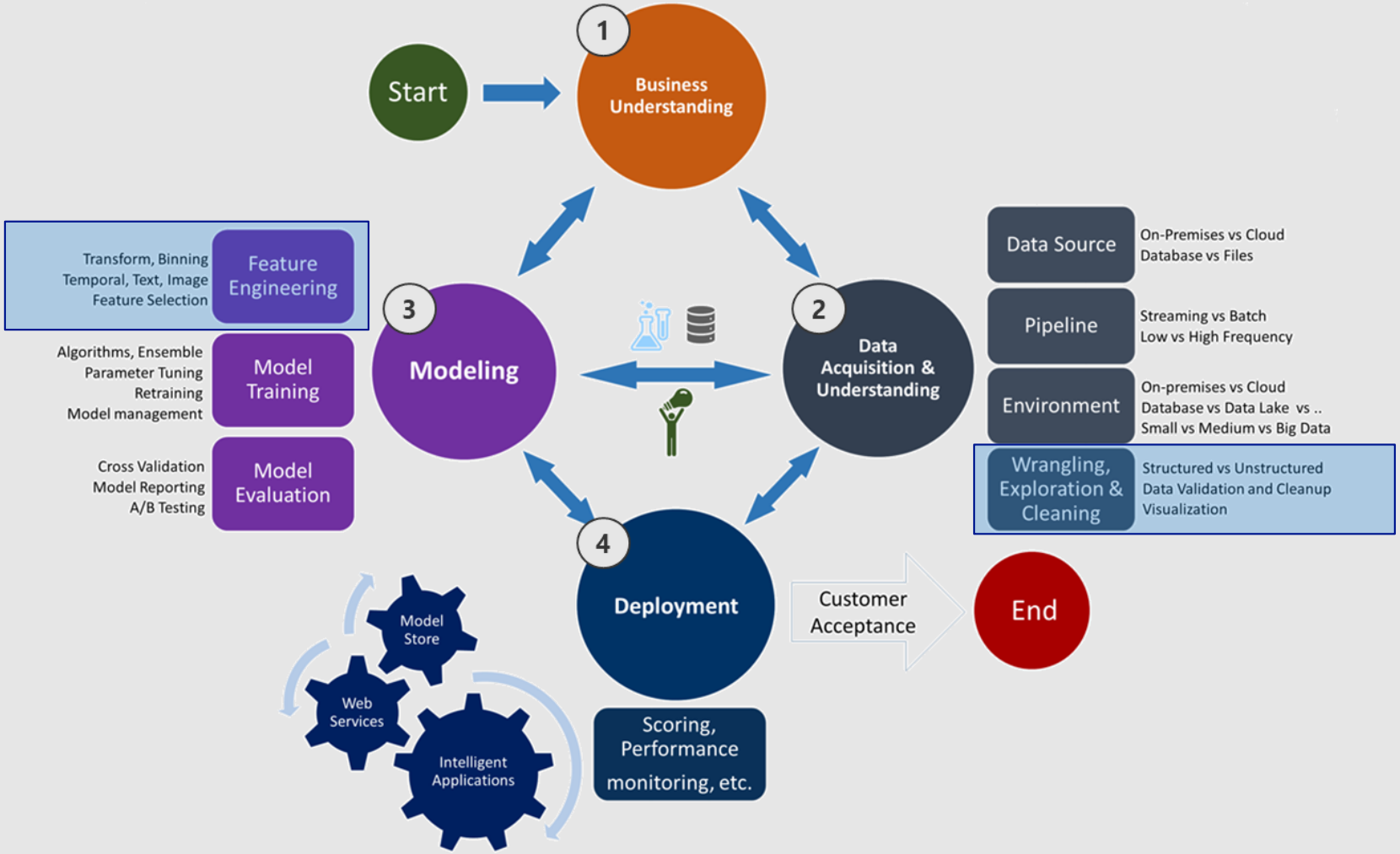
개요

- ✓ 목표
- ✓ 머신러닝 프로세스 Review
- ✓ 데이터 전 처리란?

목표

1. 목표에 맞게 데이터프레임을 구성할 수 있다.
2. 결측치(NA ; NaN)와 이상치를 찾아서 조치할 수 있다.
3. 추가 변수를 충분히 도출해 낼 수 있다.
4. 도출한 변수가 성능에 미치는 영향을 확인할 수 있다.

머신러닝 프로세스 Review



데이터 전처리란?

데이터를 분석 가능한 형태로 만드는 작업

① NA, 이상치 데이터를 처리하고
② 필요한 변수가 충분히 도출된,
- 기존 변수 : 그대로 이용
- 추가 변수 : 기존 변수에서 가공하거나, 신규로 수집되는 변수

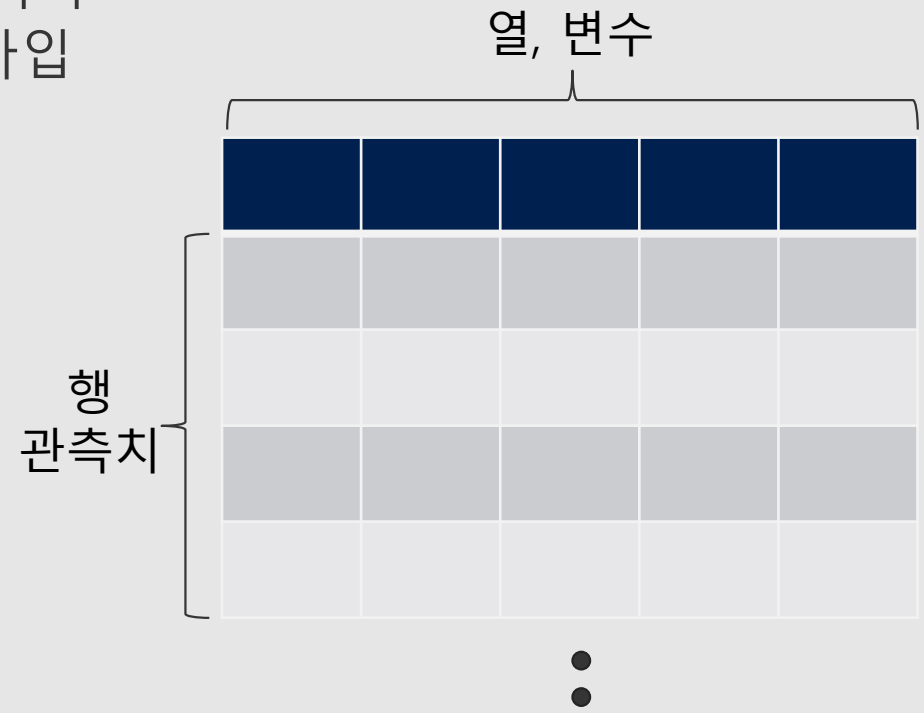
데이터프레임(테이블) 형태

분석을 위한 데이터 구조 만들기

분석 가능한 데이터

✓두 가지 형태의 정보가 저장되는 구조 : Dataframe 혹은 Table

① 데이터가 밑으로 쌓이는 구조
② 변수(열) 의미에 맞는 데이터
③ 변수(열) 마다의 데이터타입



분석 가능한 데이터

- ✓문제 정의
- ✓주제에 맞는 정보 정의
  - 정보 중에서
    - 가용한 정보 : 그대로 사용 가능한 정보, 가공해야 되는 정보
    - 비가용한 정보 : 수집(구입) 가능한 정보, 수집 불가능한 정보
- ✓정보를 dataframe 형태로 만들기
  - SQL : join, group by 등
  - Python :
    - pd.groupby
    - pd.concat
    - pd.merge

분석 가능한 데이터

✓ Dataframe 사례

- 고객 이탈 데이터프레임

고객ID	성별	나이	최근1개월 구매액	최근 1개월 방문횟수	이탈여부
					0
					1

- 이미지 분류 : MNIST

숫자	p1	p2	p3	p4	p5	...	p781	p782	p783	p784
4	0	0	0	0	34	78	167	250	0	0

결측치, 이상치 탐색 및 조치

결측치, 이상치를 어떻게 다룰 것인가?

✓ 데이터 분석 전에 **반드시** 결측치와 이상치를 처리해줘야 한다.

구분	① 제거	② 대체
이상치	제거는 권장하지 않음. 특히 자료가 많지 않은 경우	▪ 자료의 하한/ 상한 값으로 대체 ▪ 비즈니스 의미에 맞는 값으로 대체
결측치		▪ 시계열 데이터 : 같은(비슷한) 시기의 데이터 ▪ 최빈값/ 평균값으로 대체 ▪ 비즈니스 의미에 맞는 값으로 대체

결측치, 이상치를 어떻게 다룰 것인가?

③ **데이터셋을 분리한다.** (대체할 방법이 없고, 중요한 변수라면)

고객ID	이름	성별	나이	구매액
1	###	남	10	1000
2	@@@	여	24	2200
3	$$$	여	54	3400
4	%%%	남	38	5000
5	&&&	NA	18	1000
6	***	NA	22	2200

고객ID	이름	성별	나이	구매액
1	###	남	10	1000
2	@@@	여	24	2200
3	$$$	여	54	3400
4	%%%	남	38	5000

모델1

고객ID	이름	나이	구매액
5	&&&	18	1000
6	***	22	2200

모델2

NaN값 찾기

✓NaN값 찾기

1

	0	1	2	3	4	5
0	0.520113	0.884000	1.260966	-0.236597	0.312972	-0.196281
1	-0.837552	NaN	0.143017	0.862355	0.346550	0.842952
2	-0.452595	NaN	-0.420790	0.456215	1.203459	0.527425
3	0.317503	-0.917042	1.780938	-1.584102	0.432745	0.389797
4	-0.722852	1.704820	-0.113821	-1.466458	0.083002	0.011722
5	-0.622851	-0.251935	-1.498837	NaN	1.098323	0.273814
6	0.329585	0.075312	-0.690209	-3.807924	0.489317	-0.841368
7	-1.123433	-1.187496	1.868894	-2.046456	-0.949718	NaN
8	1.133880	-0.110447	0.050385	-1.158387	0.188222	NaN
9	-0.513741	1.196259	0.704537	0.982395	-0.585040	-1.693810

2

	0	1	2	3	4	5
0	False	False	False	False	False	False
1	False	True	False	False	False	False
2	False	True	False	False	False	False
3	False	False	False	False	False	False
4	False	False	False	False	False	False
5	False	False	False	True	False	False
6	False	False	False	False	False	False
7	False	False	False	False	False	True
8	False	False	False	False	False	True
9	False	False	False	False	False	False

3

```
0    False
1     True
2    False
3     True
4    False
5     True
dtype: bool
```

4

```
0    0
1    2
2    0
3    1
4    0
5    2
dtype: int64
```

Python

```
import pandas as pd
import numpy as np

df =
pd.DataFrame(np.random.randn(10,6))
# Make a few areas have NaN values
df.iloc[1:3,1] = np.nan
df.iloc[5,3] = np.nan
df.iloc[7:9,5] = np.nan

df

df.isna()

df.isna().any()

df.isna().sum()
```

NaN값 채우기 : 특정값으로.

✓NaN값 채우기

- pandas.dataframe.fillna()
- 단일 값으로 채우기 : .fillna(0)
- 열 별로 특정 값으로 채우기 : 열 별로 채울 값을 dictionary 형태로 만들고 채우기

1

	0	1	2	3	4	5
0	-0.727980	0.445696	-2.217746	-0.519223	0.233715	-0.708322
1	-0.838894	0.000000	-0.744479	-1.234386	0.466655	-0.373184
2	0.518281	0.000000	-1.952165	-0.298062	1.297094	0.527580
3	-1.244352	0.310189	2.055420	-0.826300	1.134687	1.350048
4	0.507994	2.199799	-0.114116	1.150390	0.480438	0.318925
5	0.324406	0.911641	0.438620	0.000000	-0.1	
6	-0.500391	-2.453221	1.150779	0.094536	2.1	
7	0.343893	-0.541244	-1.066677	1.125763	0.5	
8	0.260250	-0.373172	-2.337928	1.260688	-0.6	
9	-1.005624	-0.599445	0.388457	-1.552684	2.0	

2

	0	1	2	3	4	5
0	1.467203	-0.262560	-0.656085	-1.609935	-0.334550	0.194194
1	1.129105	0.500000	-1.443350	0.236572	-0.655849	0.048304
2	-0.347040	0.500000	-1.816984	-0.957809	2.088772	-0.770964
3	0.443977	2.336354	0.655078	2.229757	1.841941	1.244056
4	0.141338	0.759605	0.772148	0.408642	-0.473196	1.549767
5	0.053538	-1.538381	-1.014288	1.500000	-1.423205	-0.613535
6	0.999037	0.221433	1.661506	-0.533480	-0.674640	-0.012705
7	-1.831335	-1.253603	-0.923675	-0.013066	0.014115	2.500000
8	0.251765	0.847531	-0.872550	-0.149659	-1.071100	2.500000
9	-0.661387	0.372526	-1.417917	0.170983	-1.747402	0.617964

Python

```
import pandas as pd
import numpy as np

df =
pd.DataFrame(np.random.randn(10,6))
# Make a few areas have NaN values
df.iloc[1:3,1] = np.nan
df.iloc[5,3] = np.nan
df.iloc[7:9,5] = np.nan

df.fillna(0)  1

values = {1: 0.5, 3: 1.5, 5: 2.5}
df.fillna(value=values)  2
```

NaN값 채우기 : Titanic의 Age 채우기

✓ Titanic["Age"] 데이터에서 NaN값은 어떻게 처리할 것인가?

① 이름에서 타이틀(호칭)을 분리하자. 그리고 별도 칼럼으로 만들자.

- 당시 호칭을 통해 나이대를 가늠할 수 있다.

② 호칭의 평균 나이를 계산한다.

③ 나이가 NaN인 사람의 호칭을 보고 평균나이를 넣어준다.

실습 #8 : NaN 찾기

이상치 검출

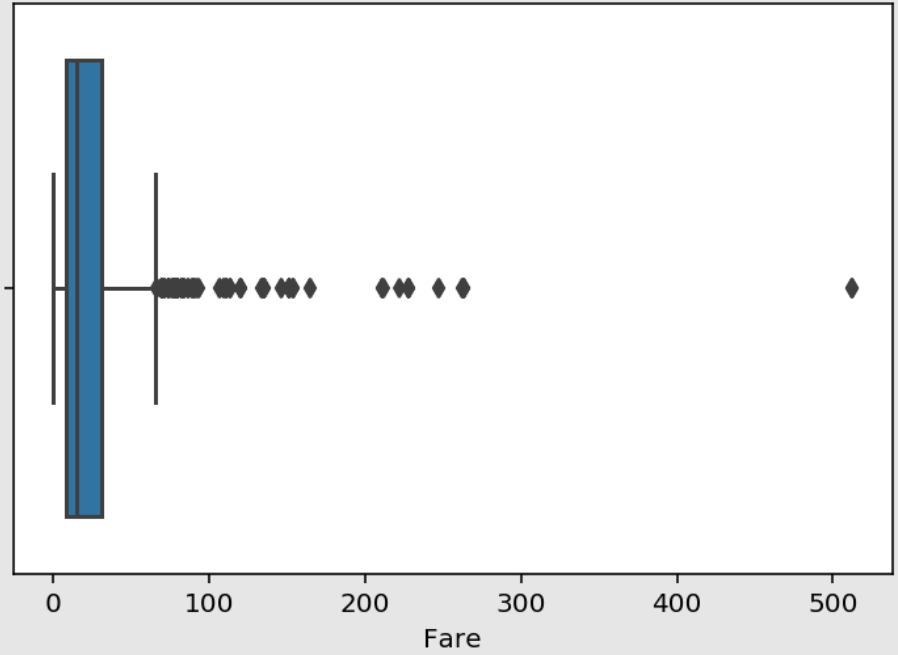
- ✓Chart(scatter, boxplot) 를 통해 살펴보고
- ✓비즈니스 관점에서 이상치인지 판별합니다.
- ✓통계량이 의한 판별은 보조자료로 활용합니다.

이상치 검출

✓Chart로 살펴보기

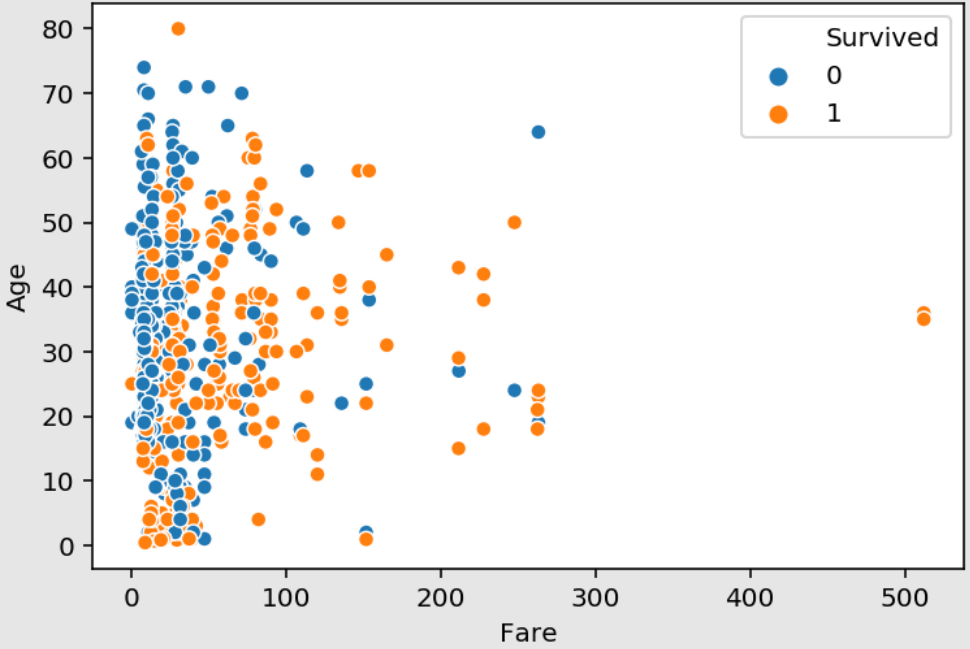
Python

```
sns.boxplot(x=ti["Fare"])
```



Python

```
sns.scatterplot(x="Fare", y="Age",
hue="Survived", data=ti)
```



이상치 검출

✓ IQR을 이용한 이상치 검출

- IQR = 3 사분위수 – 1사분위수
- 1사분위수 – 1.5 * IQR ~ 3사분위수 + 1.5 * IQR

Python

```
q1 = ti['Age'].quantile(.25)
q3 = ti['Age'].quantile(.75)

iqr = q3-q1

min_iqr = q1 - 1.5 * iqr
max_iqr = q3 + 1.5 * iqr
print(min_iqr, max_iqr)
```

Feature Engineering

Feature Engineering

- ✓ 기존 독립변수로 종속변수를 설명하는데 부족하다면 새로운 변수를 만들어 내야 합니다.
  ➔ **대부분의 상황에서는 기존 변수로는 부족!**
- ✓ 도메인 지식 + 경험 + 창의력 +

약간의 마법

- ✓ 모델 성능을 향상시키기 위해 굉장히 중요한 작업
- ✓ 불필요한 변수를 제거하고, 중요한 변수에 집중하도록 함.
- ✓ 통계적으로 만들어 낼 수도 있지만, **도메인지식**에 기반하여야 함

추가변수 사례

- ✓ **중요 값을 기준으로 변수 만들기**
  - 음주 습관에 대한 분석 : age 변수를 이용해서 age >= 20 ➔ 음주가능연령
  - 아파트가격 분석 : 방 수 >=4 & 화장실수 >=2 ➔ Premium
  - 유통 판매분석 : 명절여부, 주요이벤트여부

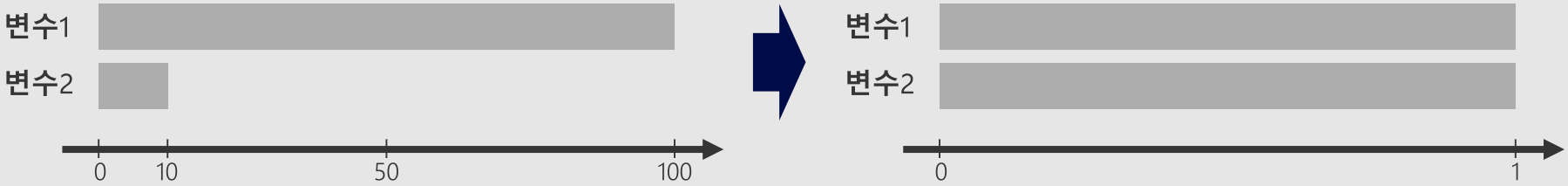
- ✓ **복수의 변수로부터 도출하기**
  - 일교차 = 일최고기온 – 일최저기온

- ✓ **시계열 데이터의 과거 데이터 계산**
  - 주가 데이터 : 최근 7일 이동평균값

- ✓ Dummy variable
  - 범주형 데이터를 명시적인 숫자로 변형

Normalization

✓ **변수 간의 값의 범위를 맞추는 작업**



$$x' = \frac{x - min(x)}{max(x) - min(x)}$$