Data Preprocessing

목차

1. 데이터 전처리의 의미 & 전처리 기법
    - 세부 내용

2. 데이터 전처리의 일반적인 단계
    - 세부 내용

3. 영상처리 & 데이터 전처리
    - 세부 내용

데이터 전처리란?

- 수집된 데이터를 용도에 적합하게 가공하는 과정을 의미함

데이터 전처리 기법

- 결측치(Missing Data)
- 중복된 데이터
- 이상치
- 정규화
- one-hot encoding

결측치(Missing Data)

- 결측치: 데이터에 값이 없는 것을 의미함.
  - “0”, “NA”, “NaN”, “NULL” 이라고 표현

누락된 데이터를 처리하는 이유는?
모델의 정확도를 높이기 위해서.

결측치을 처리하는 방법

1. 누락된 값 삭제
2. 결측값 대체

EX) 결측치

Index	val
1	10
2	15
3	NA
4	22
5	NA

결측치(누락된 모든 값 삭제)

```python
import pandas as pd

df = pd.read_csv("C:\\Users\HOME\OneDrive\Desktop\Melbourne_housing_FULL.csv")
print('결측치 제거 전:',df.shape)
result = df.dropna()
print('결측치 제거 후:', result.shape)
```

```
결측치 제거 전: (34857, 21)
결측치 제거 후: (8887, 21)
```

결측치(결측값 대체)

```python
import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer

data = {
    'Feature1': [1, 2, np.nan, 4, 5],
    'Feature2': [10, 20, 30, np.nan, 50],
    'Feature3': [100, 200, 300, 400, 500]
}

df = pd.DataFrame(data)
print("적용 전: ", df)

imputer = SimpleImputer(strategy='mean') # 평균값 대체

df_imputed = pd.DataFrame(imputer.fit_transform(df), columns=df.columns) # 결측값 대체

print("적용 후: ", df_imputed)
```

Result

```
적용 전:     Feature1  Feature2  Feature3
0       1.0      10.0       100
1       2.0      20.0       200
2       NaN      30.0       300
3       4.0       NaN       400
4       5.0      50.0       500
적용 후:     Feature1  Feature2  Feature3
0       1.0      10.0     100.0
1       2.0      20.0     200.0
2       3.0      30.0     300.0
3       4.0      27.5     400.0
4       5.0      50.0     500.0

종료 코드 0(으)로 완료된 프로세스
```

중복된 데이터 처리

이상치(Outlier) method?

- 이상치란?
  - 값의 범위에서 벗어나거나 극단적으로 데이터가 크거나 작거나 하는 값을 이상치라 고함 ex = [15, 101, 18, 7, 13, 16, 11, 21, 5, 15, 10, 9]

이상치 처리 및 감지

1. 이상치 삭제
2. 다른 값으로 대체
3. Z-score,
4. Boxplots,
5. Inter Quantile Range(IQR)[사분범위]

Z-score method?

```
def outlier(df, col, z):
    # 각 데이터 포인트의 평균으로부터의 거리를 구함
    distance_from_mean = abs(df[col] - np.mean(df[col]))

    # 표준화된(z-score) 값을 계산
    z_scores = distance_from_mean / np.std(df[col])

    # 주어진 임계값 z를 초과하는 데이터포인트의 인덱스를 반환
    outlier_indices = df[z_scores > z].index

    return outlier_indices
```

Z-score는 데이터 포인트가 평균에서 얼마나 떨어져 있는지 나타내는 표준화된 값을 의미함.

abs([1,2,3,4,5,100]−15)=[14,13,12,11,10,85]abs([1,2,3,4,5,100]−15)=[14,13,12,11,10,85]

$$\frac{14,13,12,11,10,85}{\sqrt{406}}$$

결과: 100의 인덱스 5

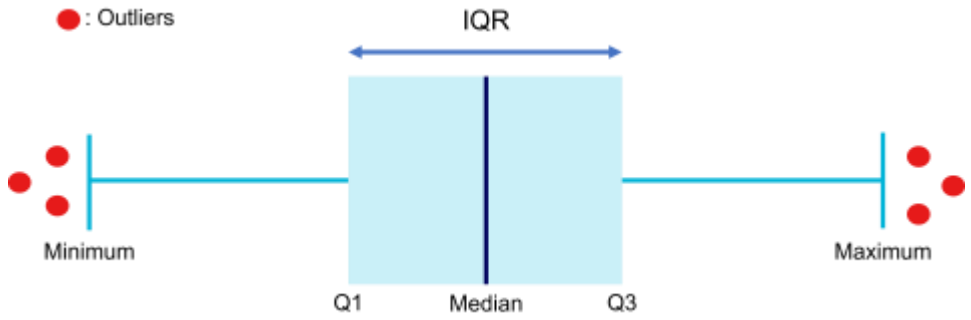
$$\frac{[14,13,12,11,10,85]}{\sqrt{406}} > 2$$

Boxplot method?

데이터의 분포를 한 눈에 확인이 가능하게 시각화 하여
이상치(Outlier)등을 탐지할 수 있는 시각화 도구.

Inter Quantile Range(IQR)[사분범위] method?

IQR[사분범위]란 중앙값(Median)을 기준으로 데이터들의 흩어진 정도를 의미함



Q3는 3사분위 수로 전체 데이터의 75%값 의미함
Q1은 1사분위 수로 전체 데이터의 25% 값을 의미함

1. IQR=Q3−Q1
2. Minimum=Q1−(IQR∗1.5)
3. Maximum=Q3+(IQR∗1.5)

Inter Quantile Range(IQR) 정의 5단계

IQR 5단계

- 데이터 세트를 오름차순으로 정렬
- 1사분위수와 3사분위수(Q1, Q3)를 계산
- IQR=Q3-Q1 계산
- 하한 계산 = (Q1-1.5*IQR), 상한 = (Q3+1.5*IQR)
- 데이터 세트의 값을 반복하여 하한값 아래 및 상한값 위에 있는 값을 확인하고 이를 이상값으로 표시

IQR 응용

Data = 5, 7, 10, 15, 19, 21, 21, 22, 22, 23, 23, 23, 23, 24, 24, 24, 24, 25

Median value ← 총 데이터의 개수는 19개 홀수이기 때문에 중앙값의 오른쪽으로 숫자 9개 왼쪽으로 숫자 9개가 있기때문에 중앙값은 23.

Data = 5, 7, 10, 15, 19, 21, 21, 22, 22, 23, 23, 23, 23, 23, 24, 24, 24, 24, 25
[Q1]1사분위? ← 자료의 중앙값 보다 왼쪽에 분포한 값들 즉 9개의 값들의 중앙값
Result = 19

Data = 5, 7, 10, 15, 19, 21, 21, 22, 22, 23, 23, 23, 23, 23, 24, 24, 24, 24,
25
[Q3]3사분위? ← 자료의 중앙값 보다 오른쪽에 분포한 값들 즉 9개의 값들 중 홀수 의 값이 중앙값이다
Result = 24
사분위수 범위(IQR)? ← Q3 – Q1 = 24 – 19
Result = 5

하한 이상치 & 상한 이상치?

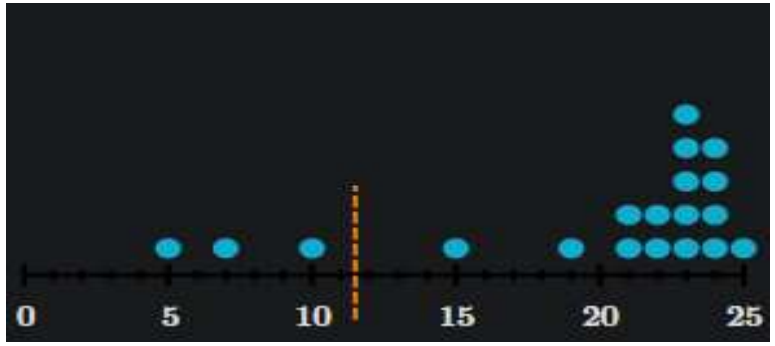
하한 이상치 ← Q1 – 1.5 * IQR

= 19 – 1.5 * 5 = 11.5

상한 이상치 ← Q3 + 1.5 * IQR

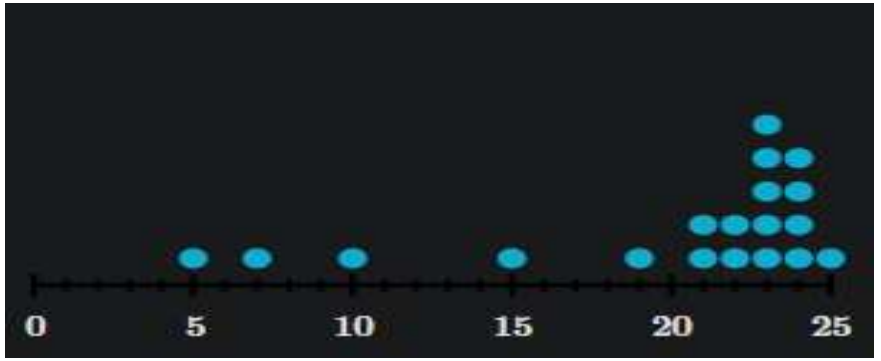
= 24 + 1.5 * 5 = 31.5

하한 이상치의 자료는?



11.5보다 왼쪽에 있는 점수로 3개가 있기때문에 총 3개의 하한 이상치 감지

상한 이상치 자료는?



25가 끝이므로 상한 이상치는 없음

Normalization?

예측 및 예측 모델의 성능을 높이는데
중요한 전처리, 매핑, 확장 방법

기계학습에서 정규화를 하는 이유?
모델의 정확도 성능을 높이고 과적합 등을 방지하기 위해서

데이터 정규화
1. Min-max Normalization[최소-최대 정규화]
2. Z-score Normalization[z-점수 정규화]

Min-Max Normalization(최소-최대 정규화)?

최소-최대 정규화는 데이터를 정규화 하는 방법 중 대중적이고 일반적인 방법.

모든 feature[특징]에 대해 각각의 최소값 0, 최대값 1로 그리고 다른 값들은 0과 1사이의 값으로 스케일링 하는 방식

$$x\ normalization = \frac{X\ -\ Xmin}{Xmax\ -\ Xmin}$$

```
def min_max_normalization(lst):
    normalized = []

    for value in lst:
        normalized_num = (value - min(lst)) / (max(lst) - min(lst))
        normalized.append(normalized_num)

    return normalized
```

Min-Max Normalization의 단점?

	GroupA	GroupB
0	0	0
1	1	10
2	2	20
3	3	30
4	4	40
5	5	50
6	6	60
7	7	700
8	8	80
9	9	90
10	10	100

	GroupA	GroupB
0	0.0	0.000000
1	0.1	0.014286
2	0.2	0.028571
3	0.3	0.042857
4	0.4	0.057143
5	0.5	0.071429
6	0.6	0.085714
7	0.7	1.000000
8	0.8	0.114286
9	0.9	0.128571
10	1.0	0.142857

B그룹의 7번 학생의 점수가 700이기 때문에 이상치이다.

정규화전에 이상치을 제거해야한다.
방안: z-score

Z-score Normalization?

Z-score 정규화는 이상치 문제를 피하는 데이터 정규화이고 평균을 0으로 만들고 표준편차를 1로 만든다.

$$Z = \frac{x - \mu}{\sigma}$$ # μ[뮤]: 데이터 평균값
σ[시그마]: 데이터 표준편차

만약 feature의 값이 평균과 일치하면 0으로 정규화 되고 평균보다 작으면 음수 평균보다 크면 양수로 나타낸다.

양수와 음수의 크기는 특징의 표준편차에 의해 결정된다.

```
def z_score_normalize(lst):
normalized = []

for value in lst:
normalized_num = (value - np.mean(lst)) / np.std(lst)
normalized.append(normalized_num)

return normalized
```

원-핫 인코딩(One-hot encoding)

원-핫 인코딩이란?

문자형 데이터를 숫자 타입으로 변환하여 더 좋은 성능을 얻는 방법 중 하나.

각 범주형 값을 새로운 범주형 열로 변환하고 해당 열에 1 또는 0의 이진 값을 할당한다.

One-hot encoding을 사용하는 이유? # 범주형 데이터

One-hot encoding은 언제 사용하지?

Original Data

Team	Points
A	25
A	12
B	15
B	14
B	19
B	23
C	25
C	29

One-Hot Encoded Data

Team_A	Team_B	Team_C	Points
1	0	0	25
1	0	0	12
0	1	0	15
0	1	0	14
0	1	0	19
0	1	0	23
0	0	1	25
0	0	1	29

데이터 전처리의 일반적인 단계

- 1. 데이터 정리 #
- 2. 데이터 통합 #
- 3. 데이터 변환 #
- 4. 데이터 감소 #
- 5. 데이터 이산화 #
- 6. 데이터 정규화 #

영상처리의 전처리

1. 컬러 공간/모델 - Color Space/Model
2. 화소 점처리 기술 - Pixel point processing
3. 히스토그램 처리 기술 - Histogram processing
4. 영역 처리 기술 -Region-based processing
5. 기하 연산 기술 - Geometric processing
6. 다해상도 기술 - Multi-resolution

컬러 공간/모델 - Color Space/Model