AI기반 데이터 분석 및 AI Agent 개발 과정

『1과목 : 』
AI기반 데이터 분석

2025.09.22-10.02(9일, 62시간)

Prepared by DaeKyeong K

Ph.D.

삼성전자 구미사업부 G.제조팀

HS Global
HARDWARE SOFTWARE

목차

『1-3』
데이터 전처리

데이터
데이터 수집, 인제스트
분석 주제 탐색 및 문제해결 단계별 접근
데이터 확인 및 검증
결측값/데이터 분포/이상치

개요

학습목표

- 이 워크샵에서는 Data Cleansing 에 대해 알 수 있다.
- 실제 데이터는 여러가지 노이즈와 문제가 있을 수 있으므로, 적절한 전처리가 필요한다.

눈높이 체크

- Data Cleansing 을 알고 계신가요
- 결측값/데이터 분포/이상치를 알고 계신가요?

1. Data Cleansing

Data Cleansing?

- 데이터 정제는 주어진 데이터에서 노이즈, 오류, 누락된 값 또는 불필요한 정보를 식별하고 수정하여 데이터의 정확성과 완전성을 향상시키는 과정을 의미한다.
- 데이터 정제는 데이터 분석 및 모델링 과정에서 정확한 결과를 얻기 위해 필수적인 단계로, 데이터의 품질을 향상시켜 유용한 인사이트를 도출하는 데 도움을 준다.
- 이 프로세스는 데이터의 이상치 제거, 중복 제거, 누락된 값 대체, 형식 통일 등의 작업을 포함할 수 있다.

  - 결측치 처리: 누락된 데이터를 확인하고, 대체하거나 삭제하는 방법을 사용하여 데이터의 완전성을 보완한다.
  - 이상치 탐지 및 처리: 이상치를 식별하고, 제거하거나 대체하여 데이터의 정확성을 향상시킨다.

1. Data Cleansing

데이터 조사와 정제Data Invest & Cleansing

- 데이터를 활용할 수 있도록 만드는 과정
- 데이터의 누락값, 불일치, 오류의 수정
- 컴퓨터가 읽을 수 없는 요소의 제거
- 숫자나 날짜 등의 형식에 대해 일관성 유지
- 적합한 파일 포맷으로 변환

출처 : https://www.dataentryoutsourced.com/blog/cxos-guide- to-marketing-and-sales-data-cleansing-and-enrichment/

1. Data Cleansing

데이터 조사와 정제Data Invest & Cleansing

- 속성의 데이터 타입과 도메인(속성 값의 범위)
- 속성 값의 분포 특성(대칭, 비대칭 등)
  - 대칭/비대칭 분포
  - 실제 값의 주요 분포 범위
  - 값의 표준편차
- 속성 간의 의존성
  - 속성 A의 값이 같은 데이터의 속성 B 값이 반드시 같다면, 속성 A와 속성 B 간의 함수적 종속성 존재 (A→B)

2. How does the data cleansing work?

데이터 정제 절차

- 일부 데이터는 올바르게 형식화되어 있어 바로 사용할 수 있지만, 대부분의 데이터는 형식 불일치 혹은 가독성 문제(예: 약어 또는 일치하지 않는 헤더 설명)가 있다. 둘 이상의 데이터 세트에서 데이터를 사용하는 경우 특히 심하다. 따라서 데이터를 정리하면 더 쉽게 저장, 검색 및 재사용을 할 수 있다.

- 일반적으로, 데이터 정제 절차는 다음과 같을 수 있다.

1. Matching 을 할 수 있다.
2. Formatting을 할 수 있다.
3. Filtering & Sorting을 할 수 있다.

2. How does the data cleansing work?

데이터 정제 절차

- Matching (일치):
  - 데이터의 일관성을 유지하기 위해 다른 소스에서 가져온 데이터 간에 일치하는지 확인한다. 예를 들어, 다른 시스템에서 가져온 데이터의 형식이나 구조를 확인하고 일치시키는 작업을 수행할 수 있다.

- Formatting (형식화):
  - 데이터를 일관된 형식으로 변환하거나 조정하여 일관성을 유지하고 분석에 용이하도록 만듭니다. 이는 데이터 타입 변환, 날짜 형식 표준화, 텍스트 형식 통일화 등을 포함할 수 있다.

2. How does the data cleansing work?

데이터 정제 절차

- Matching (일치) 예

```
import pandas as pd

# 두 데이터셋 예시 (data1과 data2)
data1 = pd.read_csv(`datasets/data1.csv')  # 첫 번째 데이터셋
data2 = pd.read_csv('datasets/data2.csv')  # 두 번째 데이터셋

# 일치시킬 기준이 되는 열을 기준으로 Merge (예시: 'key' 열을 기준으로 Merge)
merged_data = pd.merge(data1, data2, on='key_column', how='inner')
# 'key_column'은 두 데이터셋에서 일치시킬 기준이 되는 열 이름입니다.

# Merge된 데이터 확인
print(merged_data)
```

2. How does the data cleansing work?

데이터 정제 절차

- Filtering (필터링):
  - 데이터셋에서 특정 기준을 충족하는 행 또는 열을 선택하는 과정을 의미한다. 예를 들어, 조건에 따라 특정 날짜 범위의 데이터만 선택하거나, 특정 조건을 만족하는 행을 추출하는 것이 필터링에 해당한다.

- Sorting (정렬):
  - 데이터를 특정 기준에 따라 순서대로 배열하는 작업을 말한다. 이는 데이터를 오름차순 또는 내림차순으로 정렬하는 것을 의미하며, 보통 숫자나 날짜 등의 기준으로 데이터를 정렬한다.

3. Finding Outliers and Bad Data

고유한 값(결측값Missing values)

- 값이 존재하지 않고 비어있는 상태
- NA(Not Available)또는 NULL 값
  - NA: 결측값
  - NULL: 값이 없다
- 분석 대상의 속성 값이 상당 부분 비어있게되면, 분석 대상 데이터가 충분하지 않은 상태이므로 제대로 된 분석을 수행하기 어려움

3. Finding Outliers and Bad Data

결측값 구분

- MCAR(Missing Completely At Random)
  - 결측값이 관측된 데이터와 관측되지 않은 데이터와 독립적이며 완전 무작위로 발생
  - 데이터 분석 시 편향되지 않아서 결측값이 문제가 되지 않는 경우
  - 데이터가 MCAR인 경우는 거의 없음
- MAR(Missing At Random or MCARMissing Conditionally At Random)
  - 결측값이 조건이 다른 변수에 따라 조건부로 무작위 발생되는 경우
  - 변수의 조건에 따른 결측값이 설명할 수 있는 경우
  - 데이터 분석 시 편향이 발생할 수도 있음
- MNAR(Missing Not At Random)
  - MCAR 또는 MAR이 아닌 데이터
  - 무시할 수 없는 무응답 데이터 (누락된 이유가 존재)
  - 결측값이 무작위가 아니라서 주도면밀한 추가 조사가 필요한 경우

3. Finding Outliers and Bad Data

결측값 처리 방법

- 결측값 데이터 개체 또는 속성의 제거
  - 결측값이 발생한 데이터 개체를 분석 과정에서 제거하거나 해당 속성을 제거
  - 데이터가 충분히 많이 있다면 고려할만한 방법
  - 데이터 내에 결측치를 가진 데이터나 속성이 많은 경우 대부분의 정보가 제거될 수 있음
  - 실제로는 많이 사용하지 않는 방법
- 수동으로 결측값 입력
  - 결측값이 발생한 데이터를 다시 조사 및 수집하여 입력
  - 매우 고비용의 소모적인 방법
  - 결측값이 많은 경우 비현실적인 방법
- 전역상수global constant를 사용한 결측값 입력
  - 단순하고 명확한 방법
  - 예를 들어, 결측값을 0으로 입력
  - 전역상수 값이 분석 결과를 왜곡할 수 있음

3. Finding Outliers and Bad Data

결측값 처리 방법

- 결측값의 무시
  - 알고리즘이나 응용에 따라서는 결측치가 발생한 속성을 무시하고 분석을 수행할 수도 있음
  - 예를 들어, 개체들 사이의 유사성 계산에 있어 많은 수의 속성이 있는 경우 이 중 하나의 속성이 없다면 이를 제외하고 유사성을 계산할 수 있도록 알고리즘을 조정하는 것
  - 하나의 속성 값이 없더라도 유사성을 계산하는데 미치는 영향이 크지 않다면 이러한 방법도 적용 가능
  - 데이터 간 결측값을 가진 속성들이 산재해 있다면 너무 많은 데이터가 제외될 수 있음
  - 속성이 몇 개 없어 하나의 속성이라도 무시하기 힘든 경우라면 이러한 방법의 적용은 좋지 않음
- 결측값의 추정
  - 일반적으로 많이 사용되는 방법
  - 결측값이 발생한 데이터와 유사한 데이터를 사용하여 결측값을 추정하는 방법
  - 결측값을 추정하는 방법에 따라 다양한 형태가 존재

3. Finding Outliers and Bad Data

결측값 추정 방법

- 속성의 평균값을 사용하여 결측값 추정
  - 속성의 평균값을 결측값에 채워넣는 방법
  - 분석 결과를 왜곡시킬 위험성 존재
- 같은 클래스에 속하는 속성의 평균값 사용
  - 주어진 데이터와 같은 클래스(분류)에 속하는 튜플 들의 속성 평균값 사용
  - 동일 유형에 속하는 데이터의 평균값을 사용하므로 왜곡 가능성 줄임
- 가장 가능성이 높은 값으로 결측값 추정
  - 회귀분석, 베이지안Bayesian 기법, 의사결정트리 기법 등의 통계 또는 마이닝 기법을 활용하여 결측값 예측
  - 분석에 의해 가능성이 높은 값을 찾아내는 방법
  - 가장 효과적이고 높은 정확도의 결측값 예측 가능
  - 결측값을 채우기 위한 분석 가설을 세우는 등의 복잡성 존재

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치anomalies/outliers 란무엇인가?
  - **데이터의 나머지 부분과 상당히 다른 데이터 요소 집합**
- 자연적 함의Natural implication가 이상한 것은 상대적으로 드문현상
  - **수 많은 데이터가 있는 경우, 수천 개 중에 하나가 자주 발생**
  - **상황이 중요, 예: 7월에 기온이 몹시 추움**
- 중요하거나 방해가 될 수 있음
  - **10 피트(3.048 미터) 키, 2살**
  - **비정상적으로 높은 혈압**

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 이상치의 원인
- 다른 클래스의 데이터
  - **오렌지의 무게를 측정하지만 자몽이 몇 개 섞여 있음**

- 자연 변형Natural variation
  - **비정상적으로 키가 큰 사람들**

- 데이터 오류Data errors
  - **200 파운드 (약 90kg), 2살**

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 노이즈Noise와 이상치Anomalies의 구분
- 노이즈는 잘못되었거나, 임의적이거나, 값이 있거나 오염된 객체
  - **무게가 잘못 기록됨**
  - **오렌지와 섞인 자몽**
- 노이즈가 반드시 비정상적인 값이나 객체를 생성하지는 않음
- 노이즈는 흥미롭지 않음
- 이상치가 노이즈의 결과가 아닌 경우는 흥미로울 수 있음
- 노이즈와 이상치는 관련이 있지만 별개의 개념

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 일반적인 이슈: 속성의 수
- 많은 이상치가 하나의 속성으로 정의
  - **신장Height**
  - **모양Shape**
  - **색깔Color**
- 모든 속성을 사용하여 이상치를 찾기가 어려울 수 있음
  - **노이즈 또는 관련 없는 속성**
  - **객체는 일부 속성과 관련해서만 이상치를 가짐**
- 그러나 어떤 속성에서는 객체가 이상치가 아닐 수도 있음

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 일반적인 이슈: 이상치 점수
- 많은 이상치 탐지 기술은 단지 이진 분류만을 제공
  - **객체가 이상치이거나 그렇지 않음**
  - **특히 분류 기반 접근법에 해당**
- 다른 접근법은 모든 포인트에 점수를 할당
  - **점수는 객체가 비정상인 정도를 측정**
  - **객체의 순위를 매길 수 있음**
- 결국 이진 결정이 필요할 수 있음
  - **이 신용 카드 거래가 신고되어야 하나?**
  - **여전히 점수를 얻는 데 유용**

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 모델 기반 이상치 탐지
- 데이터에 대한 모델을 생성하고 확인
- 비지도Unsupervised
  - **이상치는 잘 맞지 않는 포인트**
  - **이상치는 모델을 왜곡시키는 포인트**
  - **예제:**
    - 통계분포Statistical distribution
    - 클러스터Clusters
    - 회귀분석Regression
    - 기하학Geometric
    - 그래프Graph
- 지도Supervised
  - **이상치는 희귀한 등급으로 간주**
  - **학습 데이터가 필요**

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

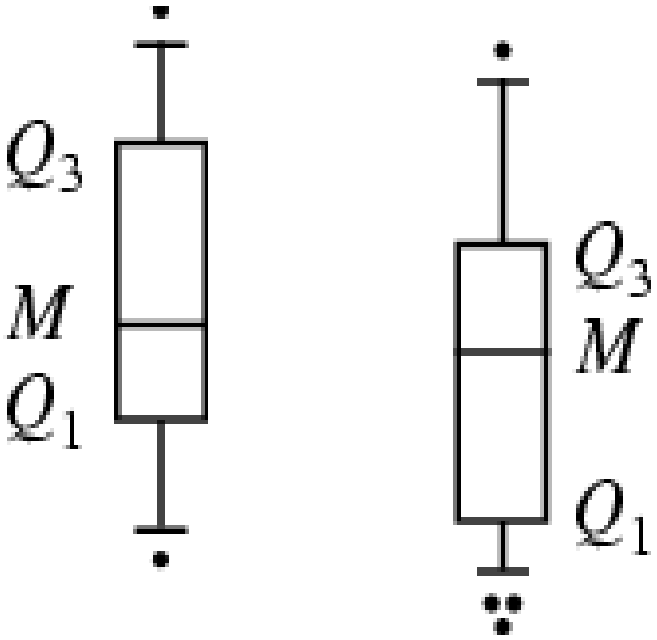
- 이상치 탐지Anomaly/Outlier Detection
- 추가적인 이상치 탐지 기술
- 근접 기반Proximity-based
  - **이상치는 다른 포인트와 멀리 떨어진 지점**
  - **일부 경우 그래픽로도 감지 가능**
- 밀도 기반Density-based
  - **저밀도 포인트는 이상치**
- 패턴 매칭Pattern matching
  - **이례적이지만 중요한 이벤트 또는 객체의 프로필이나 템플릿 생성**
  - **이러한 패턴을 탐지하는 알고리즘은 일반적으로 간단하고 효율적**

3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 시각적 접근방법

◦ **박스 플롯Boxplots 또는 분산형 플롯(산포도)scatter plots**



3. Finding Outliers and Bad Data

이상치 탐지Anomaly/Outlier Detection

- 이상치 탐지Anomaly/Outlier Detection
- 이외 통계적 접근 방식
  - **근접성 기반 이상치 탐지**
  - **밀도 기반 이상치 탐지**
  - **군집 기반 이상 치 탐지 방식이 있다**

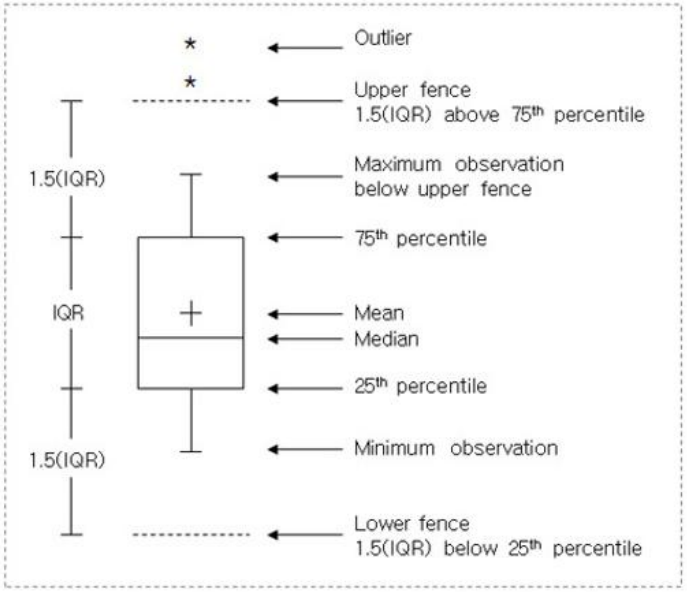
3. Finding Outliers and Bad Data

극단값 절단(trimming)

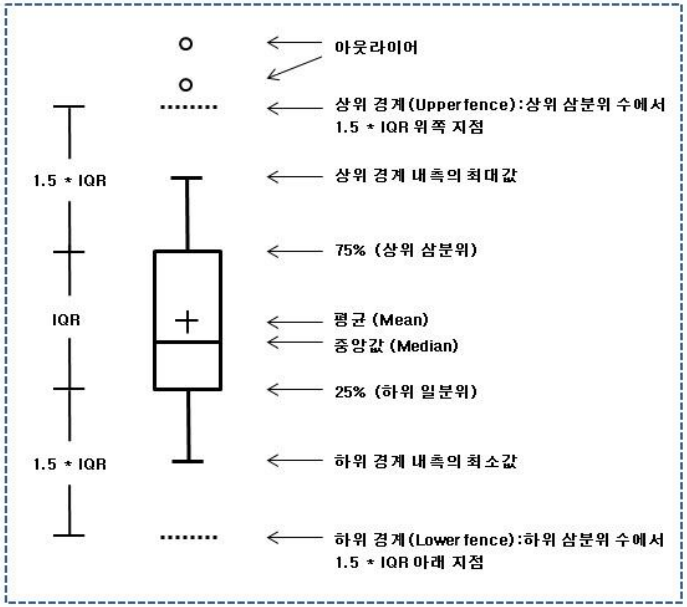
- 기하평균을 이용한 제거
- 상, 하위 5%에 해당되는 데이터 제거

극단값 조정(winsorizing)

- 상한값과 하한값을 벗어나는 값들을 상한 , 하한값 으로 바꾸어 활용하는 방법



※ IQR : Inter Quantile Range



IQR : 사분위 범위

4. 실습

실습12: ChatGPT를이용한 데이터 전처리

문제
최근5년 동안 판매된 건강기능식품에 대해 수집한 원시 데이터를 분석 목적에 맞게 데이터를 전처리하고 싶다.

[건강기능식품_매출액__국내_판매액_및_수출액_20240723113020.csv]

실습13: 결측값 처리하기

문제

[첨부 파일 : data.xlsx]
이 데이터의 결측값을 분석해 주세요.

실습14: 이상치 처리하기

문제

[첨부 파일 : 결측값제거데이터.xlsx] 이상치 분석을 위한 칼럼을 선정해 주세요.

『1-4』
데이터 분석

탐색적 데이터 분석

개요

학습목표

- 이 워크샵에서는 Exploratory Data Analysis , CDA와 EDA에 대해 알 수 있다.

눈높이 체크

- CDA와 EDA에 대해 들어보셨나요?

1. CDA와 EDA

개관

- 데이터를 이용하여 크고 복잡한 현상에서 의미 있는 패턴을 찾고, 의사 결정에 필요한 통찰을 얻는 데이터 분석이 중요해지고 있다. 데이터 분석에는 크게 두 가지의 접근방법이 있다
- CDA (Confirmatory Data Analysis):
  - CDA는 일반적으로 이전에 설정된 가설 또는 이론을 검증하기 위해 사용되는 데이터 분석 방법론이다.
  - 이 방법론은 **사전에 예측된 가정이나 가설을 검증하고 증명하기 위해 사용**된다. 가설을 설정한 후, 수집한 데이터로 가설을 평가하고 추정하는 전통적인 분석이다. 예를 들어, 이전의 연구 결과를 검증하거나 특정 가설이 사실임을 확인하기 위해 사용된다.CDA는 통계적 가설 검정, 회귀 분석 등의 방법을 사용하여 데이터에 대한 사전 가정을 확인하고 검증하는 데 주로 활용된다.

1. CDA와 EDA

개관

- EDA (Exploratory Data Analysis):
  - EDA는 **데이터를 탐색하고 이해하기 위한 접근 방법으로, 새로운 통찰력을 얻고 패턴을 발견**하는 것에 중점을 둔다. 데이터의 구조, 패턴, 관계, 이상치 등을 탐색하고 시각적 및 통계적 도구를 사용하여 데이터를 탐구한다. 목표는 데이터의 복잡성을 이해하고 다양한 변수 간의 관계를 파악하는 것이다.
  - EDA는 데이터의 특성을 파악하고 다음 단계의 분석을 계획하는 데 도움이 된다. 주로 시각화 기법을 사용하여 데이터를 탐색한다.

- 두 접근 방법은 데이터 분석의 다른 단계에서 중요한 역할을 한다. CDA는 이전의 **가정을 확인하고 검증**하는 데 중점을 두며, EDA는 **새로운 통찰력을 얻고 데이터의 패턴을 탐색**하는 데 사용된다. 이 두 방법론은 데이터 분석 프로세스의 서로 다른 측면을 보완하여 데이터의 이해를 높이고 분석 결과를 신뢰할 수 있게 한다.

2. EDA

개관

- 탐색적 데이터 분석이란 벨 연구소의 수학자 존 튜키가 제안한 데이터 분석 방법으로 통계적 가설 검정 등에 의존한 기존 통계학으로는 새롭게 나오는 많은 양의 데이터의 핵심 의미를 파악하는 데 어려움이 있다고 생각하여 이를 보완한 탐색적 데이터 분석을 도입했다고 한다. 데이터를 분석하고 결과를 내는 과정에서 원 데이터에 대한 탐색과 이해를 기본으로 가지는 것이 가장 중요하다.

  - “ '탐색적 데이터 분석(EDA)’은 우리가 존재한다고 믿는 것들은 물론이고 존재하지 않는다고 믿는 것들을 발견하려는 태도, 유연성, 그리고 자발성이다. “ - 존 튜키 (도서 Doing Data Science 중)

2. EDA

개관

- 이에 따라 탐색적 데이터 분석은 데이터의 분포와 값을 다양한 각도에서 관찰하며 데이터가 표현하는 현상을 더 잘 이해할 수 있도록 도와주고 데이터를 다양한 기준에서 살펴보는 과정을 통해 문제 정의 단계에서 미처 발견하지 못한 다양한 패턴을 발견하고 이를 바탕으로 기존의 가설을 수정하거나 새로운 가설을 추가할 수 있도록 한다.
- 데이터에 대한 관찰과 지식이 이후에 통계적 추론이나 예측 모델 구축 시에도 사용되므로 데이터 분석 단계 중 중요한 단계라고 할 수 있다.
- EDA의 목표는 관측된 현상의 원인에 대한 가설을 제시하고, 적절한 통계 도구 및 기법의 선택을 위한 가이드가 되며, 통계 분석의 기초가 될 가정을 평가하고 추가 자료수집을 위한 기반을 제공한다.

2. EDA

분석 방법

- 탐색적 데이터 분석은 한 번에 완벽한 결론에 도달하는 것이 아니라 아래와 같은 방법을 반복하여 데이터를 이해하고 탐구하는 과정이다.

1. 데이터에 대한 질문 & 문제 만들기
2. 데이터를 시각화하고, 변환하고, 모델링하여 그 질문 & 문제에 대한 답을 찾아보기
3. 찾는 과정에서 배운 것들을 토대로 다시 질문을 다듬고 또 다른 질문 & 문제 만들기

- 이러한 과정을 기반으로 데이터에서 흥미 있는 패턴이 발견될 때까지, 더 찾는 것이 불가능하다고 판단될 때까지 도표, 그래프 등의 시각화, 요약 통계를 이용하여 전체적인 데이터를 살펴보고 개별 속성의 값을 관찰한다. 데이터에서 발견되는 이상치를 찾아내 전체 데이터 패턴에 끼치는 영향을 관찰하고, 속성 간의 관계에서 패턴을 발견한다.

2. EDA

전체적인 데이터 살펴보기

- 데이터 항목의 개수, 속성 목록, NAN 값, 각 속성이 가지는 데이터형 등을 확인하고, 데이터 가공 과정에서 데이터의 오류나 누락이 없는지 데이터의 head와 tail을 확인한다. 또한, 데이터를 구성하는 각 속성값이 예측한 범위와 분포를 갖는지 확인한다.

- EDA과정에서 사용하는 판다스 함수
  - 구조 확인: shape, columns, info()
  - 결측치 확인: isnull().sum(), isnull().mean()
  - 타입 확인: dtypes, df['col'].dtype
  - 샘플 확인: head(), tail()
  - 분포 확인: describe(), value_counts(), unique(), min(), max()

3. EDA 종류

1차원 데이터 탐색하기

- 일변량 1차원 양적 자료를 분석할 때에는 빈도와 백분율의 정보가 있는 표를 사용한다. 하지만 1차원 양적 자료는 질적 자료와 다르게 자료가 가지는 각각의 값에 대한 빈도와 백분율을 구하지 않는다.
- 그 이유로는 질적 자료는 데이터가 많다고 하더라도 데이터가 가지는 값의 종류는 몇 개가 되지 않지만, 1차원 양적 자료는 데이터가 가지는 값의 종류가 많기 때문이다.

◦ 예를 들면, 5000명에게 성별(gender)과 신장(height)을 조사했고, 5000명 모두가 성별과 신장에 대해서 응답해 주었다고 하자. 성별이라는 질적 자료에는 5000개의 자료가 있고, 5000개가 가지는 값은 두 개의 값인 '남자(또는 1)' 나 '여자(또는 2)'로 구성되어 있다. 신장이라는 1차원 양적 자료에도 5000개의 자료가 있지만 5000개가 가지는 값은 성별처럼 **두 개의 값이 아니라 훨씬 더 많은 다양한 값**들로 구성되어 있다.

3. EDA 종류

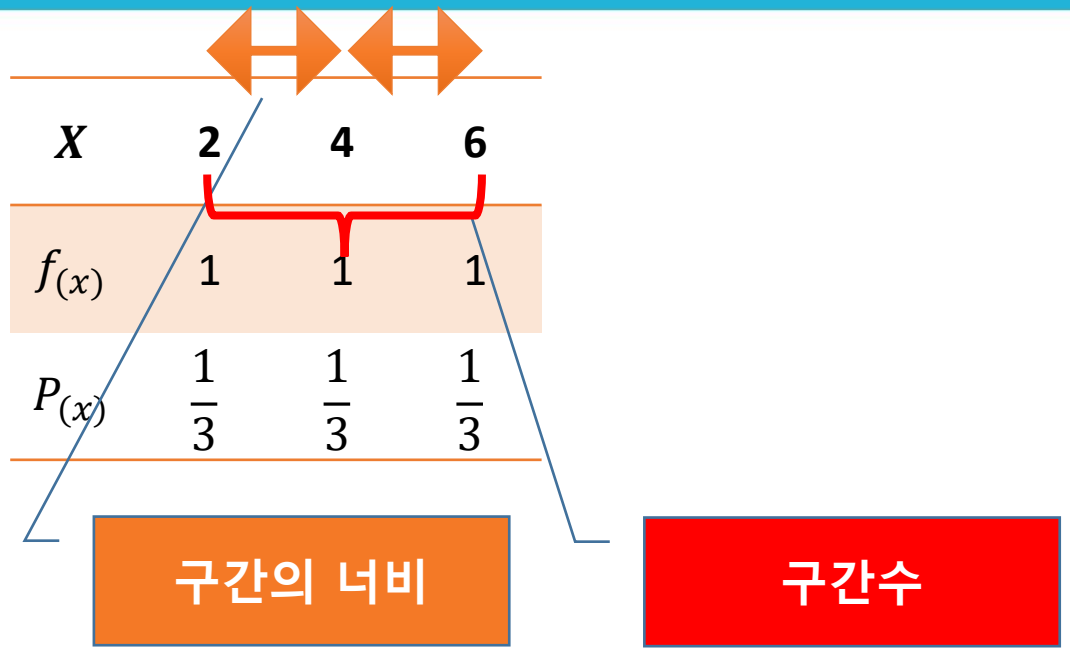
1차원 데이터 탐색하기

- 1차원 데이터 탐색 시 가장 먼저 요약 통계치를 계산할 수 있다. 하지만 요약 통계치가 무의미한 경우도 있다. 그런 경우에는 범위를 몇 개의 구간으로 나누고 각 구간 내 데이터 개수를 살펴보는 히스토그램을 만들어 볼 수 있다.

- 일반적으로, 각 구간의 너비는 다음과 같이 한다.
  - (최대값 – 최소값) / 구간의 개수
  - 첫 번째 구간에는 양적 자료의 최소값이 포함되도록 하고, 제일 마지막 구간에는 최대값이 포함되도록 구간을 만들면 된다. 이렇게 만들어진 구간을 이용하여 각 구간에 있는 양적 자료의 빈도와 백분율을 구해서 표를 작성하면 된다. 작성된 표를 이용하여 양적 자료에 있는 특징을 파악하면 된다.

3. EDA 종류

1차원 데이터 탐색하기



- 구간을 만들 때에 구간을 몇 개로 할 지에 대해서는 다양한 방법이 있다. 여기서는 Herbert A. Sturges(1926)가 「The Choice of a Class Interval」라는 제목으로 발표한 논문에서 제시한 공식은 다음과 같다. 참고로 n은 데이터의 개수를 의미한다.

$$1 + 3.3\log_{10}(n)$$

- 예 : 100개의 자료일 경우 1+3.3log10(100) = 7.3 --> 7~8개

3. EDA 종류

1차원 데이터 탐색하기

- 아래의 예제 데이터를 가지고 구간을 만들고 빈도와 백분율을 구해보도록 한다.

54 56 57 59 73 74 76 78 78 82 82 85 86 91 94 98 116

- Sturges 공식을 따르면, 구간의 개수는 다음과 같다.

$$1 + 3.3\log_{10}(17) = 5.06$$

- 구간의 개수는 반올림해서 5개로 한다. 구간의 너비는 (116-54)/5로 하면 대략적으로 15가 된다

구분	빈도	백분율(%)
50이상 ~ 65이하	4	23.5
65초과 ~ 80이하	5	29.4
80초과 ~ 95이하	6	35.3
95초과 ~ 110이하	1	5.9
110초과~125이하	1	5.9
합계	17	100.0

3. EDA 종류

1차원 데이터 탐색하기

- 도수분포표frequency distribution table : 1차원 양적 자료를 적당한 간격으로 집단화하여 계급, 도수, 상대도수, 누적도수, 누적상대도수, 계급값 등을 기입한 표
- 계급class
  : 양적자료를 적당한 간격으로 집단화하여 나타낸 범주
- 계급간격class width
  : 이웃하는 두 계급의 위쪽 경계에서 아래쪽 경계를 뺀 값
- 계급 상대도수class relative frequency
  : 계급의 도수를 전체 자료수로 나눈 값
- 도수(frequency) : 각 계급에 속하는 자료의 개수

$$\text{계급상대도수} = \frac{\text{계급의 도수}}{\text{전체 도수}}$$

3. EDA 종류

1차원 데이터 탐색하기

- 누적도수cumulative frequency : 이전 계급까지의 모든 도수를 합한 도수
- 누적상대도수cumulative relative frequency : 이전 계급까지의 모든 상대도수를 합한 상대도수
- 계급값class mark : 각 계급의 중앙값 즉, 다음에 의하여 결정되는 값

$$\text{계급값} = \frac{\text{위쪽 경계} + \text{아래쪽 경계}}{2}$$

- 계급의 수 결정 방법
  - 일반적으로 자료의 수(n)가 200 미만이면 $k = \sqrt{n} \pm 3$ 에 가까운 정수를 택하고, 200이상이면 스터지스 공식이라 부르는 $k = 1 + 3.3\log_{10} n$ 에 가까운 정수를 택한다.

3. EDA 종류

1차원 데이터 탐색하기

- 수량화되어 있는 전체 자료를 그 값의 크기에 따라 일정한 계급 class으로 나누고, 각 계급에 속하는 자료의 도수frequency를 대응시켜 작성한 표
- 도수분포표 작성 순서

❶ 자료에서 최댓값 $x_{\max}$와 최솟값 $x_{\min}$을 찾아서 범위 $R = x_{\max} - x_{\min}$을 구한다.

❷ 계급의 수 k를 정한다.

❸ 계급구간 $c = \frac{R}{k}$을 결정한다.

❹ 계급경계를 결정한다.

❺ 계급값 $x_i = \frac{(\text{계급의 양 끝값의 합})}{2}$을 구한다.

❻ 계급도수 f_i를 구한다.

3. EDA 종류

1차원 데이터 탐색하기

- 다음 자료는 하천 유역의 수리시설을 점검하기 위해 지난 30년 동안 누적강우강도를 조사한 것이다.
  - 이 누적강우강도에 대하여 도수분포표를 작성하라.

(단위 : 인치)

43.30	43.11	58.71	42.96	53.20	54.49
47.38	45.93	50.37	48.21	43.93	53.29
63.52	45.05	58.83	49.57	39.91	43.11
40.78	41.31	50.51	51.28	67.72	59.12
55.77	48.26	54.91	44.67	46.77	67.59

❶ 자료에서 최댓값과 최솟값을 찾으면 $x_{\max} = 67.72$, $x_{\min} = 39.91$이므로 범위 R은 다음과 같다.

$$R = x_{\max} - x_{\min} = 67.72 - 39.91 = 27.81$$

❷ 자료의 수가 30이므로 계급의 수를 $k = 6$으로 정한다.

❸ 계급구간 $c = \dfrac{27.81}{6} = 4.635$이므로 대략 5로 정한다.

❹ 자료의 최솟값과 최댓값을 포함하면서
계급이 중첩되지 않도록 다음과 같이 계급경계를 정한다.

3. EDA 종류

1차원 데이터 탐색하기

❺ 각 계급별로 계급값

$$x_i = \frac{(\text{계급의 양 끝값의 합})}{2}$$ 을

구하여 다음과 같이 표에 작성한다.

❻ 계급도수 f_i를 구하여 다음과 같이 도수분포표를 완성한다.

계급(인치)	계급값	도수
$35^{\text{이상}} \sim 40^{\text{미만}}$	37.5	1
40 ~ 45	42.5	8
45 ~ 50	47.5	7
50 ~ 55	52.5	7
55 ~ 60	57.5	4
60 ~ 65	62.5	1
65 ~ 70	67.5	2
합계	–	30

3. EDA 종류

1차원 데이터 탐색하기

- 일변량 1차원 양적 자료의 특징을 파악하기 위해서 작성하는 그래프에는 히스토그램(histogram), 줄기와 잎 그림(stem and leaf plot), 상자그림(boxplot)이 있다.
- 히스토그램은 구간의 빈도나 백분율을 이용하여 작성하며, x축은 1차원 양적 자료의 구간, y축은 각 구간의 빈도 또는 백분율이 된다. 도수히스토그램frequency histogram : 수평축에 도수분포표의 계급간격, 수직축에 각 계급의 도수를 높이로 갖는 사각형으로 작성한 그림

3. EDA 종류

1차원 데이터 탐색하기

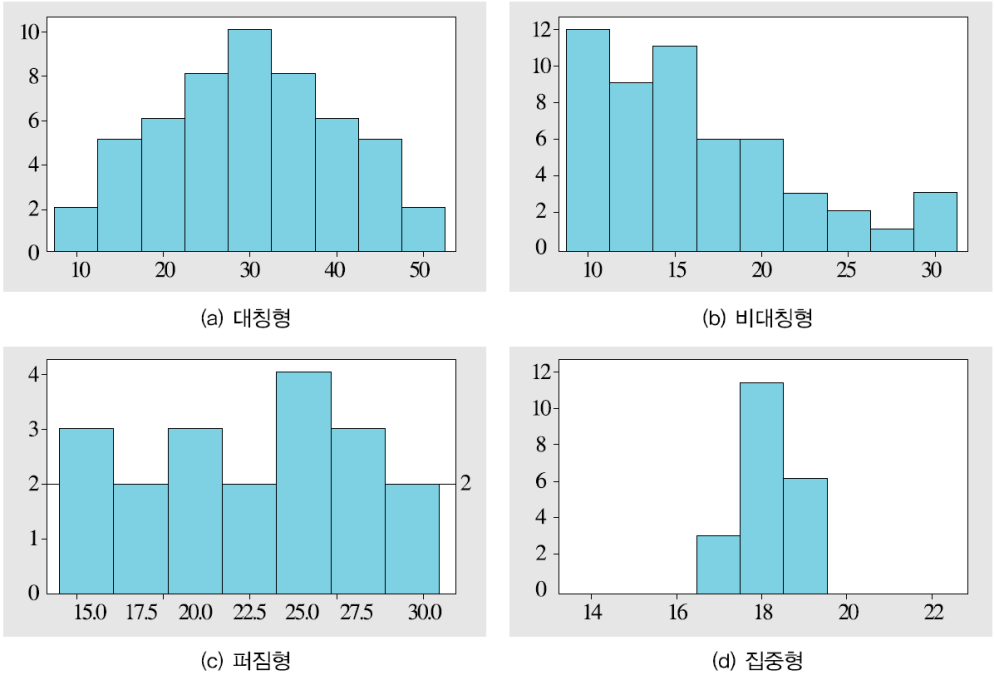
- 히스토그램이 보여주는 내용

  - 각 구간의 현황
  - 빈도가 가장 많은 구간
  - 빈도가 가장 작은 구간
  - 최소값을 포함하는 구간
  - 최대값을 포함하는 구간
  - 무게 중심
  - 대칭 여부
  - 이상치(outlier) 유무 : 이상하게 크거나 이상하게 작은 값의 유무
  - 단봉 : 봉우리가 한 개
  - 쌍봉 : 봉우리가 두 개

3. EDA 종류

1차원 데이터 탐색하기

- [장점] 다음과 같은 사항을 시각적으로 쉽게 알 수 있다.
  ① 수집한 자료의 대칭성 또는 치우침
  ② 자료들의 흩어진 모양
  ③ 자료의 집중 경향
  ④ 틈새(gap)를 갖는 계급
  ⑤ 다른 계급들로부터 멀리 떨어진 계급



(a) 대칭형 (b) 비대칭형

(c) 퍼짐형 (d) 집중형

3. EDA 종류

2차원 데이터 탐색(Exploring 2D data)

- **2차원 데이터의 개념**
  - 정의: 2차원 데이터는 행(Row)과 열(Column)로 구성된 테이블 형태의 데이터.
  - 비유: 엑셀 시트와 비슷함 → 행은 레코드(개별 데이터), 열은 속성(특징, feature).
  - 개발자 관점: Pandas의 DataFrame, SQL의 Table, Numpy의 2D 배열.

- **행(Row)과 열(Column)의 역할**
  - Row: 데이터의 한 단위 (예: 한 명의 고객, 한 건의 거래).
  - Column: 데이터의 속성 (예: 이름, 나이, 매출액).
  - Index: 행을 구분하는 고유 번호.

3. EDA 종류

2차원 데이터 탐색(Exploring 2D data)

- 탐색 방법 (EDA 기초)
  - 데이터 기본 구조 확인
    - df.shape # (행 개수, 열 개수)
    - df.columns # 열 이름 목록
    - df.info() # 데이터 타입, 결측치
  - 데이터 일부 확인
    - df.head() # 앞부분 샘플
    - df.tail() # 뒷부분 샘플
    - df.sample(5) # 임의 샘플 5개
  - 요약 통계 확인
    - df.describe() # 수치형 데이터 요약
    - df.describe(include='all') # 범주형 데이터 포함 요약
  - 행·열 접근
    - df['컬럼명'] # 특정 열 선택
    - df.loc[0] # 첫 번째 행 선택
    - df.iloc[0, 1] # (행 0, 열 1)의 값

3. EDA 종류

2차원 데이터 탐색(Exploring 2D data)

- **시각적 탐색**
  - 2차원 데이터를 더 잘 이해하기 위해 그래프 활용 권장.
  - 산점도: df.plot.scatter(x='col1', y='col2')
  - 상관관계 히트맵: sns.heatmap(df.corr(), annot=True)
  - 분포 확인: df['col'].hist()

- 2차원 데이터는 각 변수를 따로 살펴볼 수도 있지만 두 변수가 2차원 공간상에서 어떻게 분포를 이루는지 살펴보는 것도 의미가 있다.

3. EDA 종류

다차원 데이터 탐색하기

1. 다차원 데이터의 개념

•**정의**: 3차원 이상 데이터를 의미.
- 예: 이미지(가로 × 세로 × 색상채널), 시계열 센서데이터(시간 × 센서ID × 측정값), 고객 데이터(고객 × 제품 × 시간).

- **개발자 비유:**
  - **Numpy**: 다차원 배열(ndarray).
  - **Pandas**: DataFrame은 기본적으로 2차원이지만, 멀티인덱스(MultiIndex)나 Panel 형식으로 확장 가능.
  - **딥러닝**: Tensor(다차원 배열).

2. 왜 다차원 데이터가 필요한가?

•현실 데이터는 단순히 행/열 구조로 설명되지 않음.
•예:
- 의료: 환자 × 검사종류 × 시간 → 3차원
- 영상: width × height × channel × time → 4차원
- 추천시스템: 사용자 × 아이템 × 맥락(시간/장소) → 3차원

3. EDA 종류

다차원 데이터 탐색하기

- 탐색 방식은 크게 3단계
  - 구조 확인 (shape, ndim)
  - 부분 선택/요약 (슬라이싱, 통계)
  - 시각화/차원축소 (PCA, t-SNE)

- 다차원 데이터의 경우 각 차원이 서로 어떻게 연관되어 있는지 살펴볼 수 있다. 가장 간편한 방법은 상관관계 행렬을 살펴보는 것이다.

3. EDA 종류

이상치(Outlier) 분석

- 전체적인 추세와 특이사항을 관찰한다. 데이터가 많다고 특정 부분만 보게 되면 이상치가 다른 부분에서 나타날 수도 있으므로 앞, 뒤, 무작위로 표본을 추출해서 관찰해야 한다. 이상치들은 작은 크기의 표본에서는 나타나지 않을 수도 있다.
- 두 번째로는 적절한 요약 통계 지표를 사용한다. 데이터의 중심을 알기 위해서는 평균, 중앙값, 최빈값을 사용하고, 데이터의 분산도를 알기 위해서는 범위, 분산 등을 이용한다. 통계 지표를 이용할 때에는 평균과 중앙값의 차이처럼 데이터의 특성에 주의해서 이용해야 한다.
- 세 번째로는 시각화를 활용한다. 시각화를 통해 데이터의 개별 속성에 어떤 통계 지표가 적절한지를 결정한다. 시각화 방법에는 Histogram, Scatterplot, Boxplot, 시계열 차트 등이 있다.
- 이외에도 기계학습의 K-means 기법, Static based detection, Deviation based method, Distance based Detection 기법을 이용하여 이상치를 발견할 수 있다.

3. EDA 종류

속성 간의 관계 분석

- 속성 간의 관계 분석을 통해 서로 의미 있는 상관관계를 갖는 속성의 조합을 찾아낸다. 분석에 대상이 되는 속성의 종류에 따라서 분석 방법도 달라져야 한다. 변수 속성의 종류는 다음과 같다.

범주형 변수 (Categorical)	명목형 데이터
	순서형 데이터
이산형 변수 (Numeric)	연속형 데이터
	이산형 데이터

3. EDA 종류

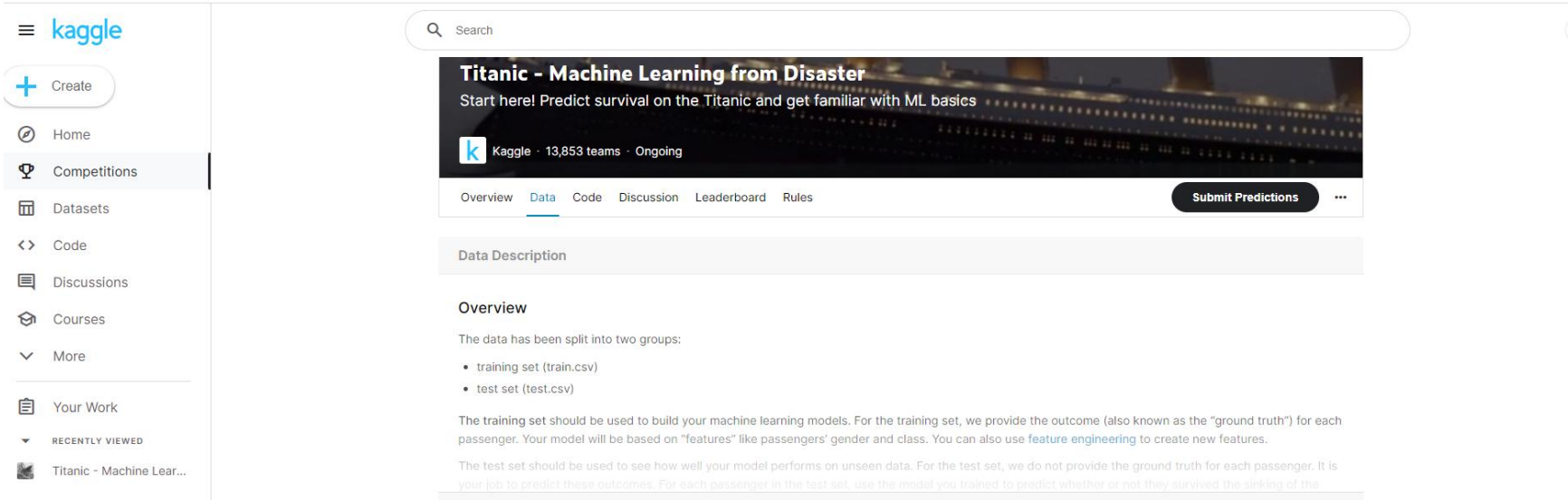
속성 간의 관계 분석

- 먼저 이산형 변수- 이산형 변수의 경우 상관계수를 통해 두 속성 간의 연관성을 나타낸다. Heatmap이나 Scatterplot을 이용하여 시각화할 수 있다.
- 다음으로 이산형 변수 - 범주형 변수는 카테고리별 통계치를 범주형으로 나누어서 관찰할 수 있고, Box plot, PCA plot 등으로 시각화할 수 있다.
- 마지막으로 범주형 변수- 범주형 변수의 경우에는 각 속성값의 쌍에 해당하는 값의 개수, 분포를 관찰할 수 있고 Piechart, Mosaicplot 등을 이용하여 시각화할 수 있다.

4. EDA와 인사이트 얻기

1. 데이터 가져오기

- 데이터 분석할 때 가장 먼저 해야 할 일은 데이터를 탐색하여 변수와 대상에 대한 개요를 얻는 것이다.
  - 사용할 데이터는 앞 서 소개한 Titanic 데이터 세트를 사용한다.Titanic 데이터 세트는 타이타닉에 탑승한 1309명의 승객의 인구 통계 및 티켓 정보에 대한 정보를 포함하는 매우 인기 있는 데이터 세트이며, 목표는 승객 중 생존 가능성이 더 높은지를 예측하는 것이다. Titanic3 데이터 세트 버전은 Titanic: Machine Learning From Disaster(https://www.kaggle.com/c/titanic)라는 제목의 인기 있는 Kaggle.com 대회를 비롯한 여러 다른 소스에서도 사용할 수 있다.



4. EDA와 인사이트 얻기

1. 데이터 가져오기

- Kaggle 버전은 학습 및 검증 세트로 미리 섞이고 미리 분할할 수 있는 이점이 있다. 훈련 세트는 train.csv라는 파일에 포함되어 있고 검증 세트는 test.csv이다. 컴퓨터에 있는 CSV 파일의 Titanic 훈련 세트를 Pandas 데이터 프레임으로 로드한다.

```
import numpy as np
import pandas as pd

file_in = './datasets/titanic.csv'

titanic_df = pd.read_csv(file_in)
```

4. EDA와 인사이트 얻기

1. 데이터 가져오기

titanic_df

>>

	PassengerId	Survived	Pclass	Name	Gender	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

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- Obtaining an Overview of the Data
  - 가장 먼저 할 일은 데이터 세트의 행과 열 수에 대한 정보를 얻는 것이다. shape, dtype 함수를 통해 데이터 항목의 개수와 type을 알아보겠다. 데이터 프레임에 891개의 행과 12개의 열(또는 속성)이 있는 것을 볼 수 있다.

```
titanic_df.shape
>>
(891, 12)
titanic_df.dtypes
>>
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Gender          object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- Titanic 데이터셋의 Kaggle 버전 속성에 대한 설명은 다음과 같다.
  - PassengerId: 행 식별자 역할을 하는 텍스트 변수.
  - Survived: 재해에서 살아남은 사람을 나타내는 부울 변수이다.
    - 0 = 아니오, 1 = 예.
  - Pclass: 티켓 등급을 나타내는 범주형 변수.
    - 1 = 1급, 2 = 2급, 3 = 3급.
  - Name: 승객의 이름이다.
  - Gender: 승객의 성별을 나타내는 범주형 변수이다.
  - Age: 승객의 연령을 나타내는 숫자 변수이다.
  - SibSp: 함께 여행하는 형제/배우자의 수를 나타내는 숫자 변수이다.
  - Parch: 부모와 자녀가 함께 여행하는 인원수를 나타내는 수치변수.
  - Ticket: 티켓 번호를 포함하는 텍스트 변수이다.
  - Fare: 1970년 이전 영국 파운드로 지불된 요금을 나타내는 숫자 변수이다.
  - Cabin: 객실 번호를 나타내는 텍스트 변수이다.
  - Embarked: 승선항을 나타내는 범주형 변수.
    - C = 셰르부르, Q = 퀸스타운, S = 사우샘프턴.

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- 다음 스니펫을 사용하여 열 이름을 가져올 수 있다.

```
titanic_df.columns.values
>>
array(['PassengerId', 'Survived', 'Pclass', 'Name', 'Gender', 'Age',
       'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype=object)
```

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- 때때로 데이터에 대한 느낌을 얻는 가장 좋은 방법은 데이터 프레임의 내용을 시각적으로 검사하는 것이다. 데이터 프레임 객체의 head() 함수를 사용하여 데이터 프레임의 처음 몇 행의 내용을 볼 수 있다. head, tail 함수를 이용해서 앞 5행, 뒤 5행의 데이터를 살펴보도록 하겠다.

titanic_df.head()
>>

PassengerId	Survived	Pclass	Name	Gender	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

titanic_df.tail()

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- Pandas가 한 줄에 가로로 맞추기에 열 수가 너무 많은 경우 Pandas는 기본적으로 데이터 프레임의 열 하위 집합을 표시한다. 하위 집합은 데이터 프레임의 왼쪽에서 몇 개의 열과 오른쪽에서 몇 개의 열로 구성된다. 30개의 열이 있는 데이터 프레임에서 head() 함수를 사용할 때의 효과를 보여준다. display.max_columns Pandas 속성을 설정하여 Pandas가 표시할 최대 열 수를 변경할 수 있다. 예를 들어 다음 스니펫은 Pandas가 4개 이하의 열을 표시하도록 한다.

```
pd.set_option('display.max_columns',4)
titanic_df.head()
>>
```

	PassengerId		Survived	...	Cabin	Embarked
0	1	0	...	NaN	S	
1	2	1	...	C85	C	
2	3	1	...	NaN	S	
3	4	1	...	C123	S	
4	5	0	...	NaN	S	

5 rows × 12 columns

4. EDA와 인사이트 얻기

2. 데이터 개요 얻기

- 모든 열을 표시하려면 display.max_columns 값을 None으로 설정한다.

```
pd.set_option('display.max_columns',None)
titanic_df.head()
>>
```

PassengerId	Survived	Pclass	Name	Gender	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

4. EDA와 인사이트 얻기

3. 결측값 정보

- 데이터 과학자가 처리해야 하는 가장 일반적인 문제 중 하나는 결측값 문제이다.
  - 원시 데이터 세트에는 종종 하나 이상의 열에 누락된 값이 있다.
  - 값이 누락된 데는 인적 오류부터 해당 관찰에 사용할 수 없는 데이터에 이르기까지 여러 가지 이유가 있을 수 있다.
  - CSV 파일을 Pandas 데이터 프레임에 로드하면 Pandas는 NaN을 누락된 값을 나타내는 마커로 사용한다.
  - 데이터 프레임의 열에 누락된 값이 포함되어 있는지 확인하는 다양한 방법이 있다. 한 가지 방법은 다음과 같이 info() 함수를 사용하는 것이다.

4. EDA와 인사이트 얻기

3. 결측값 정보

titanic_df.info()
>>

```
<class 'pandas.core.frame.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    object
 4   Gender       891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

info() 함수의 결과를 보면 대부분의 열에 891개의 값이 있는 반면 Age, Cabin 및 Embarked의 세 열에는 891개 미만의 값이 있음을 알 수 있다.

4. EDA와 인사이트 얻기

3. 결측값 정보

- 누락된 값의 수를 결정하는 또 다른 방법
  - 다음 스니펫은 데이터 프레임의 각 열에서 누락된 값의 수를 얻기 위해 isnull() 및 sum() 함수를 연결한 결과를 보여준다.

```
titanic_df.isnull().sum()
>>
PassengerId      0
Survived         0
Pclass           0
Name             0
Gender           0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

누락된 값에 대한 정보를 얻는 또 다른 방법은 Pandas isnull() 함수의 출력을 sum() 함수와 연결하는 것이다. isnull() 함수는 데이터 프레임에 적용될 때 원래 데이터 프레임과 동일한 차원을 갖는 부울 값의 데이터 프레임을 반환한다. 새 데이터 프레임의 각 위치는 원래 데이터 프레임의 해당 위치 값이 None 또는 NaN인 경우 True 값을 갖다. 부울 값의 새 데이터 프레임에 적용할 때 sum() 함수는 각 열의 값이 True인 목록을 반환한다.

4. EDA와 인사이트 얻기

4. 색인

- Titanic 데이터 세트의 PasengerId 속성이 숫자 행 식별자이고 모델 구축에 관한 한 유용한 입력을 제공하지 않다. 모든 Pandas 데이터 프레임은 데이터 프레임의 각 행에 대해 고유한 값을 포함하는 인덱스를 가질 수 있다. 기본적으로 Pandas는 데이터 프레임에 대한 인덱스를 생성하지 않다. 다음 스니펫을 사용하여 데이터 프레임에 인덱스가 있는지 확인할 수 있다.

```
print(titanic_df.index.name)
>>
None
```

4. EDA와 인사이트 얻기

4. 색인

- PassengerId 속성을 df_titanic 데이터 프레임의 인덱스로 만들려면 다음 스니펫을 사용하십시오. set_index() 함수를 실행한 후 데이터 프레임의 인덱스를 검사하면 PassengerId 속성이 이제 인덱스임을 알 수 있다.

```
titanic_df.set_index("PassengerId", inplace=True)
print(titanic_df.index.name)
>>
PassengerId
```

4. EDA와 인사이트 얻기

4. 색인

- 인덱스 설정 전후에 head() 함수를 df_titanic 데이터 프레임에 적용한 결과를 봅니다.

titanic_df.head()

PassengerId	Survived	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

- 색인이 설정된 후 데이터프레임의 행과 열 갯수
  - 이제 데이터 프레임의 모양 속성을 사용하여 행과 열의 수를 가져오면 열 수가 이제 12가 아닌 11로 보고되는 것을 알 수 있다. 이는 다음 스니펫에 설명되어 있다.

titanic_df.shape

>>

(891, 11)

5. 데이터 프레임 만들기

- target attribute와 feature variables 데이터 프레임 만들기
  - Survived 속성은 PassengerId가 인덱스로 사용된 후 df_titanic 데이터 프레임에 남아 있는 11개의 속성 중 하나이다.
  - 훈련 과정에서 대상 속성을 입력 기능 중 하나로 포함하지 않도록 해야 한다. 이를 보장할 수 있는 다양한 방법이 있지만 가장 간단한 옵션은 특성 변수와 대상 변수를 별도의 데이터 프레임으로 분리하는 것이다. 다음 스니펫은 titanic_df 데이터 프레임의 Survived 속성을 titanic_df_target이라는 별도의 데이터 프레임으로 추출하고 df_titanic 데이터 프레임의 10개 기능 변수를 titanic_df_features라는 별도의 데이터 프레임으로 추출한다.

```
titanic_df_target =  titanic_df.loc[:,['Survived']]
titanic_df_features =  titanic_df.drop(['Survived'], axis=1)
```

4. EDA와 인사이트 얻기

6. 목표 값의 분포

- 생존 속성은 값 1이 개인이 생존했음을 의미하는 이진 속성이다. 기계 학습 모델이 예측하려는 대상이 범주형(이진 또는 다중 클래스)인 경우 범주별 교육 데이터 세트의 값 분포를 아는 것이 유용한다. 다음 스니펫을 사용하여 이 예에서 대상 값의 분포를 얻을 수 있다.

```
titanic_df_target['Survived'].value_counts()
>>
0    549
1    342
Name: Survived, dtype: int64
```

4. EDA와 인사이트 얻기

7. 데이터의 통계적 특성 가져오기

- 특성 및 대상 변수의 분포에 대한 정보 외에도 이러한 변수의 통계적 특성과 이들 간의 상관 관계는 훈련 데이터에 대한 유용한 통찰력을 제공할 수 있다. Pandas는 데이터 프레임 내에서 숫자 속성에 대한 통계 정보를 얻기 위해 데이터 프레임에서 사용할 수 있는 describe() 함수를 제공한다. 다음 스니펫은 titanic_df_features 데이터세트에 대한 describe() 함수의 결과를 보여준다.

```
titanic_df_features.describe()
>>
```

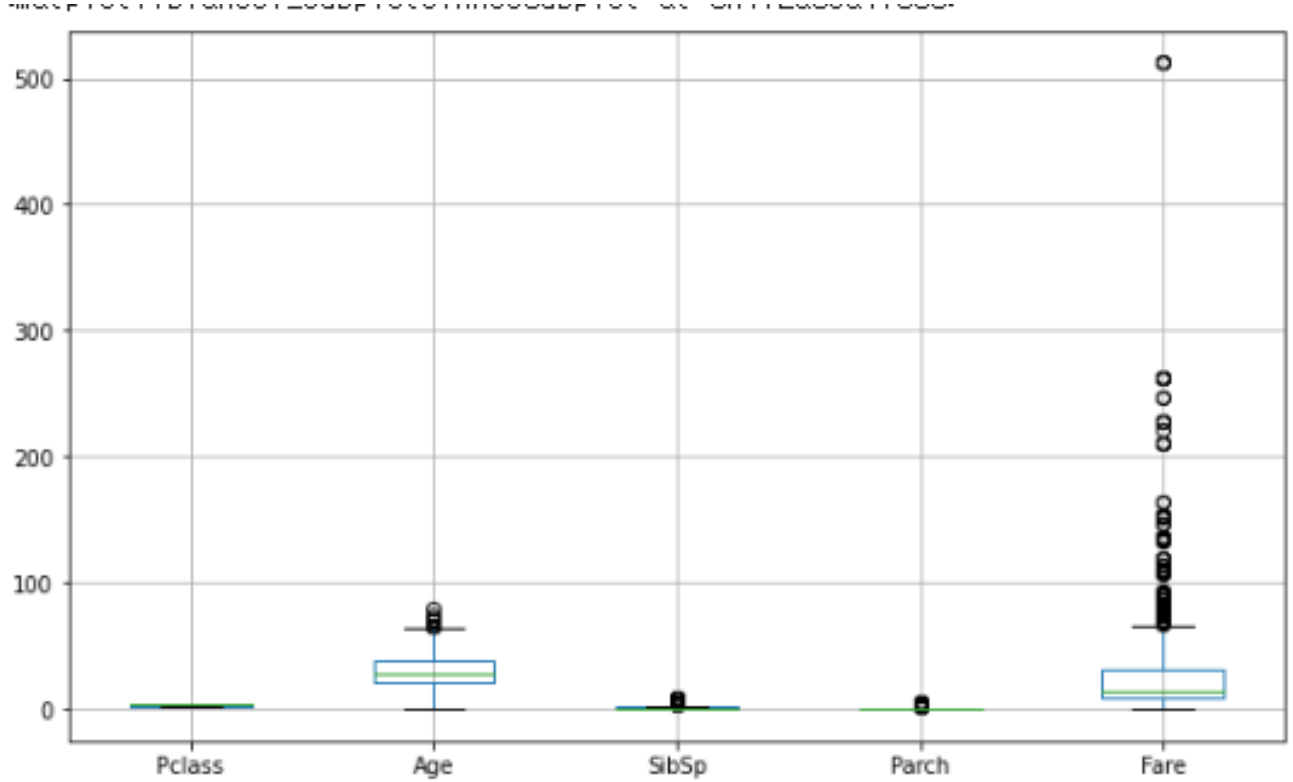
Pclass	Age	SibSp	Parch	Fare		
count	891.000000		714.000000		891.000000	891.000000
	891.000000					
mean	2.308642	29.699118	0.523008	0.381594	32.204208	
std	0.836071	14.526497	1.102743	0.806057	49.693429	
min	1.000000	0.420000	0.000000	0.000000	0.000000	
25%	2.000000	20.125000	0.000000	0.000000	7.910400	
50%	3.000000	28.000000	0.000000	0.000000	14.454200	
75%	3.000000	38.000000	1.000000	0.000000	31.000000	
max	3.000000	80.000000	8.000000	6.000000	512.329200	

4. EDA와 인사이트 얻기

7. 데이터의 통계적 특성 가져오기

- 사분위- create a box plot of numeric features.

titanic_df_features.boxplot(figsize=(10,6))



5. 실습

실습15: ChatGPT를 이용한 탐색적 데이터 분석

문제
건강기능식품에 대한 신제품을 출시하기 위해 소비자 요구, 경쟁 상황, 시장트랜드 등을 파악하고자 설문조사를 실시했다. 수집한 데이터의 특성과 추가 분석을 위한 인사이트를 얻고 싶다.

5. 실습

실습16: ChatGPT를 이용한 탐색적 데이터 분석

문제
[첨부 파일 : 이상치제거데이터.xlsx] 범주형 변수에 대한 통계를 구해 주세요.

THANK YOU.

앞으로의 엔지니어는 단순한 '코더'나 '기계 조작자'가 아니라 뇌-기계 인터페이스를 통해 지식과 능력을 즉각 확장하는 존재(뉴로-인터페이스: Neuro Interface)가 될 수 있습니다.

🎯 목표 달성을 위한 여정이 시작됩니다.
✨ 궁금한 점이 있으시면 언제든 문의해주세요!
🚀 함께 코더와 프롬프트 전문가로 성장해 나갑시다!