데이터 다루기 (1)

Pandas + Seaborn 실습

GOAL

- 데이터의 개념 이해
- Pandas, Seaborn 기초 실습
- 데이터프레임 구조 및 결측치 처리 능력 습득

목차

데이터분석이란?

> 데이터를 정리·요약·해석해 **무슨 일이 있었고 왜 그런지** 설명하고, 의사결정에 바로 쓰일 **근거**를 만드는 활동

- 넷플릭스가 추천 영화를 보여주는 이유는? → 데이터 분석 덕분

데이터사이언스란?

> 데이터를 통해 세상의 패턴을 발견하고,
> 의사결정에 도움을 주는 학문

- 병원에서 질병 확률을 예측하는 이유는? → 데이터 기반 모델

왜 데이터사이언스를 배워야 할까?

- 데이터는 “21세기의 석유”
- AI, 추천시스템, 예측모델의 뿌리는 모두 데이터
- 코딩 실력보다 “데이터로 생각하는 힘”이 중요

데이터 엔지니어링 vs 데이터사이언스

구분	데이터 엔지니어링	데이터사이언스
초점	데이터 수집·저장	데이터 분석·예측
주요 도구	SQL, Spark, ETL	Pandas, Numpy, ML
목표	데이터 파이프라인 구축	인사이트 도출

비유:

- 데이터 엔지니어 → 도로를 깔아주는 사람
- 데이터 사이언티스트 → 그 도로를 달리는 사람

Pandas란?

> "엑셀을 코딩으로 조작하게 해주는 도구"

```
import pandas as pd
df = pd.read_csv("data.csv")
df.head()
```

- 데이터프레임 구조로 쉽게 조작 가능
- 통계, 필터링, 그룹화, 시각화까지 한 번에 가능

왜 Pandas를 써야 할까?

- 리스트나 딕셔너리는 데이터분석에 불편
- Pandas는 **엑셀+SQL+파이썬의 융합체**

데이터프레임의 기본 구조

구성요소	설명
행(Row)	개별 데이터 (예: 한 환자)
열(Column)	Feature / 변수
dtype	데이터 타입 (int, float, object 등)

데이터프레임 생성 방법

```
import pandas as pd

data = {
    "Name": ["Tom", "Jane", "Alice", "Bob"],
    "Age": [28, 34, 29, 42],
    "Score": [85, 92, 88, 75]
}
df = pd.DataFrame(data)
print(df)
```

인덱스 조작

```
df.index = ["a", "b", "c", "d"]
print(df)
```

- 기본 0부터 시작하는 인덱스를 커스텀할 수 있음
- 데이터 구분이 쉬워짐

컬럼 선택

```
df["Name"]
df[["Name", "Score"]]
```

- 단일 컬럼 선택 시 시리즈 반환
- 여러 컬럼 선택 시 데이터프레임 반환

행 선택 (loc / iloc)

```
# 인덱스 이름으로 선택
df.loc["a"]

# 행 번호로 선택
df.iloc[0]
```

- loc: 인덱스 이름 기반
- iloc: 정수 기반 인덱스

행 조건 필터링

```
df[df["Score"] > 80]
```

- True/False 값으로 조건 필터링 가능
- SQL의 WHERE 절과 유사

여러 조건 결합

```
df[(df["Score"] > 80) & (df["Age"] < 35)]
```

- `&` : and, `|` : or
- 괄호 필수!

정렬 (sort_values)

```
df.sort_values(by="Score", ascending=False)
```

- by: 기준 컬럼
- ascending=False → 내림차순 정렬

값 수정하기

```
df.loc[df["Name"] == "Tom", "Score"] = 90
```

- 조건을 만족하는 행의 특정 값만 변경 가능

새로운 컬럼 추가

```
df["Passed"] = df["Score"] >= 80
print(df)
```

- 조건식 결과를 새로운 컬럼으로 바로 추가 가능

컬럼 삭제 (drop)

```
df.drop(columns=["Passed"], inplace=True)
```

- `axis=1` 또는 `columns` 옵션으로 컬럼 삭제

행 삭제

```
df.drop(index="b", inplace=True)
```

- `index` 옵션으로 행 제거 가능

Feature(특징)란?

> 분석 대상의 속성, 즉 “각 열(Column)”

예: 환자 데이터

Feature	설명
Glucose	혈당 수치
BMI	체질량 지수
Age	나이
Insulin	인슐린 수치

머신러닝에서 Feature의 역할

- Feature가 많을수록 예측 정확도 ↑ (하지만 과적합 주의)
- 어떤 Feature가 중요한지를 판단하는 것도 분석의 핵심!

Seaborn이란?

> 통계적 시각화를 쉽게 만들어주는 라이브러리

```
import seaborn as sns
titanic = sns.load_dataset("titanic")
sns.head()
```

- 유명한 샘플 데이터 포함 (titanic, iris 등)
- 시각화와 데이터로드를 동시에!

Seaborn으로 데이터 불러보기

```
titanic = sns.load_dataset("titanic")
print(titanic.head())
```

학습 포인트:

- `load_dataset()` 으로 즉시 로드 가능
- 데이터 구조 탐색은 `.info()`

데이터 확인하기

```
titanic.info()
titanic.describe()
```

`.describe()` → 평균, 중앙값, 표준편차 등 빠르게 확인 가능

결측치(누락된 데이터)란?

> 값이 비어 있거나(None, NaN) 없는 데이터

- 실제 분석에서 거의 100% 등장!
- 결측치를 처리하지 않으면 모델이 제대로 학습되지 않음

결측치 확인

```
titanic.isnull().sum()
```

각 컬럼별로 결측치 개수를 보여줍니다.

결측치 채우기

```
titanic["age"].fillna(titanic["age"].mean(), inplace=True)
```

평균으로 채우는 방법 외에도 중앙값(`median`), 최빈값(`mode`) 가능!

비유:

> 반 친구 중 결석한 사람의 시험점수는 평균으로 대체하는 느낌

산술 평균 (Mean)

> “전체 데이터를 더해서 개수로 나눈 값”

장점

- 가장 많이 쓰이는 방식 (기본값)
- 계산이 간단하고 빠름

단점

- 극단값(Outlier) 의 영향을 많이 받음
  → 한두 개의 이상치가 있으면 평균이 왜곡됨

> 비유: 단체에서 대부분이 20살인데, 누군가 80살이면 평균나이가 갑자기 확 올라가는 느낌!

중앙값 (Median)

> “가운데 값(순서대로 정렬했을 때 중간값)으로 채움”

장점

- 극단값에 영향을 거의 받지 않음
- 데이터가 한쪽으로 치우친 경우 적합

단점

- 분포의 미세한 변화를 반영하지 못함

> 비유: “가장 평범한 사람의 기준값”으로 채우는 것!

최빈값 (Mode)

> “가장 자주 등장하는 값”으로 채움

장점

- 문자형 데이터에도 적용 가능
- 예: “성별”, “지역”, “카테고리” 같은 범주형 데이터 처리

단점

- 여러 최빈값이 존재할 수 있음
- 수치형 데이터에서는 잘 쓰이지 않음

> 비유: “가장 인기 있는 선택지로 채우기"
> 서울 사람이 제일 많으면, 비어 있는 지역은 서울로 채움

결측치 제거

```
titanic.dropna(inplace=True)
```

주의: 데이터를 너무 많이 잃을 수 있음.

→ “버릴지, 채울지”는 상황 판단 필요!

실습: 당뇨병 환자 데이터

파일: train.csv / test.csv

컬럼:

ID, Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI,

DiabetesPedigreeFunction, Age

Step 1. 데이터 불러오기

```
train = pd.read_csv("train.csv")
test = pd.read_csv("test.csv")
train.head()
```

Step 2. 결측치 확인

```
train.isnull().sum()
```

Step 3. 결측치 처리

```
train["Glucose"].fillna(train["Glucose"].mean(), inplace=True)
train.dropna(inplace=True)
```

요약 정리

주제	핵심 내용
데이터사이언스	데이터로 문제 해결
Pandas	데이터프레임 분석 도구
Seaborn	시각화 + 샘플데이터
결측치 처리	dropna(), fillna()
Feature	각 데이터의 속성