프로젝트 명: Disease Prediction (질병 예측)

단백질간에 관계, 유전자, 단백질 구조 등 데이터 분석으로 질병을 예측하는데 목적을 두고 있다.

---

1. 라이브러리 로드

```
import pandas as pd
import numpy as np
```

pandas: 데이터프레임 처리 numpy: 수치 연산용 배열 처리

```
#파일 불러오기
dg_data = pd.read_csv(data_path + "DG-AssocMiner_miner-disease-gene.tsv",
sep="\t")
protein_data = pd.read_csv(data_path + "9606.protein.links.detailed.v12.0.txt",
sep= ' ')
gene_protein_id_mapping = pd.read_csv(data_path +
"HUMAN_9606_idmapping_selected.tab", sep= '\t')
```

dg_data: 질병-유전자 연관

protein_data: 단백질 링크 정보

gene_protein_id_mapping: 유전자 ↔ 단백질 ID 매핑

```
uniprot_df = gene_protein_id_mapping[['UniProt_ID', 'Gene ID', 'GO_Terms',
'PDB_IDs', 'PubMed_IDs', 'Ensembl_Protein_ID']]
uniprot_df.head()
```

분석 대상 칼럼 필터링

데이터 파일 칼럼에서 유전자, 단백질 정보 중 필요한 정보만 필터링하여 uniprot_df로 새로운 dataframe으로 구성한다.

```
dg_data['Gene ID'] = dg_data['Gene ID'].astype(str)
uniprot_df['Gene ID'] = uniprot_df['Gene ID'].astype(str)
```

병합을 위한 타입 통일

Gene ID 칼럼으로 양쪽 데이터셋에서 문자열로 맞춰주고 병합이 정확하게 수행된다.

```
# 병합 (Gene ID 기준으로)
merged_df = dg_data.merge(uniprot_df, on='Gene ID', how='left')

# 결과 확인
print(merged_df.shape)
merged_df.head()
```

유전자 정보 병합

질병-유전자 연관 데이터와 단백질 정보를 `Gene ID` 기준으로 병합한다.

```
# 1. explode
merged_df_exploded = merged_df.copy()
merged_df_exploded['Ensembl_Protein_ID'] =
merged_df_exploded['Ensembl_Protein_ID'].fillna('')
merged_df_exploded = merged_df_exploded.assign(
    Ensembl_Protein_ID=merged_df_exploded['Ensembl_Protein_ID'].str.split(';')
).explode('Ensembl_Protein_ID')
merged_df_exploded['Ensembl_Protein_ID'] =
merged_df_exploded['Ensembl_Protein_ID'].str.strip()

# 2. 버전 번호 제거 (.3 등 제거) → 정규식 활용
merged_df_exploded['Protein_ID_Clean'] =
merged_df_exploded['Ensembl_Protein_ID'].str.replace(r'\.\d+$', '', regex=True)

# 3. protein_data에 맞추기 위해 앞에 "9606." 붙이기
merged_df_exploded['Protein_ID_Formatted'] = '9606.' +
merged_df_exploded['Protein_ID_Clean']

# 4. # Disease ID에서 # 삭제
merged_df_exploded = merged_df_exploded.rename(columns={'# Disease ID': 'Disease
ID'})

merged_df_exploded.head()
```

단백질 ID 병합하기 위한 전처리 및 확장

칼럼명을 통일성 있게 문자 제거, '9606' 인간을 나타내는 접두어 추가

단백질 상호작용 정보 병합 (PPI 데이터)

---

1. 데이터 분석에 사용할 컬럼만 추출

```
protein_data = protein_data[['protein1', 'protein2', 'combined_score']]
merged_df_exploded = merged_df_exploded[[
    "Disease ID", "Disease Name", "Gene ID", "UniProt_ID",
```

```
    "GO_Terms", "PDB_IDs", "PubMed_IDs", "Protein_ID_Formatted"
]]
```

분석에 불필요한 컬럼은 제거하여 메모리를 절약한다. 병합에 필요한 컬럼만 남겨둔다.

```
# 2. 병합용 리스트 생성
ppi_chunks = []

# 3. chunk 단위 병합 시작
chunk_size = 5000
for i in range(0, len(merged_df_exploded), chunk_size):
    chunk = merged_df_exploded.iloc[i:i+chunk_size].copy()

    # protein1 병합
    join1 = chunk.merge(protein_data, left_on='Protein_ID_Formatted',
right_on='protein1', how='inner')

    # protein2 병합
    join2 = chunk.merge(protein_data, left_on='Protein_ID_Formatted',
right_on='protein2', how='inner')

    # 두 병합 결과 저장
    ppi_chunks.append(pd.concat([join1, join2]))

# 4. 최종 병합 결과 결합
ppi_merged = pd.concat(ppi_chunks, ignore_index=True)
```

데이터가 커서 5000개 단위로 잘라서 처리한다. protein1 ,protein2 모두 연결되도록 양방향으로 병합을 수행한다. 모든 결과를 병합하여 concat()으로 리스트에 저장한다.

각 칼럼들의 이름과 기능

📋 컬럼 설명 정리 (최종 병합 테이블 기준)

컬럼명	설명	기능 / 용도
`Disease ID`	질병 고유 ID (예: DOID_0050156)	분석 기준이 되는 질병 식별자
`Disease Name`	질병 이름 (예: breast cancer)	결과 해석 및 시각화 시 사용
`Gene ID`	유전자 식별 번호 (NCBI 기준)	병합 키로 사용됨
`UniProt_ID`	단백질 고유 ID (UniProt 기준)	단백질 식별자
`GO_Terms`	단백질의 기능 분류 정보 (Gene Ontology)	기능 기반 분석에 활용
`PDB_IDs`	단백질 구조 ID (Protein Data Bank)	구조 기반 연구 시 / 질병 예측 활용

컬럼명	설명	기능 / 용도
PubMed_IDs	관련 논문 ID	참고 문헌 추적 가능
Protein_ID_Formatted	9606. 접두어가 붙은 단백질 ID	PPI 병합 시 사용되는 표준 포맷
protein1, protein2	상호작용하는 단백질 쌍	PPI 네트워크 연결 정보
combined_score	단백질 간 상호작용 강도 (0~1000)	상호작용 신뢰도 기준

🧬 UniProt 메타데이터 컬럼 설명

컬럼명	설명	용도
entryId	UniProt 내부 entry ID	고유 식별자
gene_x	유전자 이름 (from UniProt)	분석에 사용되는 유전자명
geneSynonyms	유전자 별칭들	보조 검색 키
isReferenceProteome	기준 단백질 여부 (True/False)	표준 단백질 여부 확인
isReviewed	리뷰 여부 (Swiss-Prot vs TrEMBL)	데이터 신뢰도 판단
sequenceChecksum	시퀀스 무결성 검사용 코드	데이터 검증
sequenceVersionDate	시퀀스 버전이 갱신된 날짜	버전 추적
uniprotAccession	공식 단백질 Accession ID	고유 단백질을 추적 가능
uniprotId	단백질 이름 ID	uniprot DB에서 고유 식별 번호
uniprotDescription	단백질 기능 설명	분석에 사용 가능한 설명
taxId	생물종 ID (9606 = 인간)	종 분류
organismScientificName	생물학적 학명	Homo sapiens
globalMetricValue	글로벌 통계 수치	분석 또는 품질 관련 지표
uniprotStart, uniprotEnd	단백질 시퀀스의 일부분 위치	위치 기반 기능 연구 가능
uniprotSequence	단백질 아미노산 서열	생물학적 분석의 핵심
modelCreatedDate	이 데이터가 만들어진 날짜	데이터 생성 시점 추적
organismCommonNames	종의 일반 이름 (human)	종 구분 가능능
proteinFullNames	단백질 전체 이름	설명적 이름
latestVersion, allVersions	최신 및 전체 버전 기록	변경 이력 파악
isAMdata	Additional metadata 포함 여부	추가 분석용
organismScientificNameT	종 학명 (변형 또는 반복)	구조상 중복 필드 가능성
version	내부 DB 버전	시스템용
proteinShortNames	단백질 약칭	간단한 이름 사용 시

컬럼명	설명	용도
uniprotAccession_unchar	미지의 단백질 ID	기능 발견 전 데이터 가능성
entry_name	UniProt entry 이름	주로 유전자 기반 명칭
protein_name	단백질 이름 (일반적 표현)	단백질 이름름
organism	종 이름 (일반 표현)	Human
tax_id	종 ID (중복 필드 가능)	taxId와 유사
gene_y	유전자 이름 (병합된 또 다른 필드)	gene_x와 동일 가능
protein_existence	단백질 존재 근거 (1~5 등급)	신뢰도 판단
sequence_version	시퀀스 버전 정보	변경 이력 추적용
sequence	아미노산 서열 정보	단백질 기능 예측의 핵심