# === 참고용 ===

energy : 석유, 가스

materials : 화학, 건설자재, 채광, 목재, 제지, 일반패키징

industrials : 자본재, 기계장치, 전자부품, 항공 및 군수산업, 운송산업, 사무용품

discretionary : 자동차, 내구재(가구), 사치품, 관광서비스, 유통서비스(백화점), 소매판매업(음식점)

staples : 식음료, 개인용품(휴지, 물티슈 등)

health care : 의료 및 의료서비스, 의료기기판매업

financial : 은행(투자은행포함), 보험, 자산운용

it : 반도체, IT하드웨어, IT소프트웨어

communication service : 통신업, 미디어 및 언론사, 엔터테인먼트(영화 등)

energy, materials, financials : 김영준

industrials, discretionary : 정새미, 윤용현

staples, health\_care : 조준연

it, communication\_services : 최소윤

# ==== sector summary ===== (여기 밑에 딕셔너리로 저장부탁드림 ~)

sector = dict.fromkeys(["energy", "materials", "industrials", "discretionary", "staples",

"health\_care", "financials", "it", "comm\_services"])

# 금융 지원업

financials\_support\_select = ['유화증권', 'SK증권', '키움증권', '삼성증권', 'DB금융투자', '부국증권', '상상인증권', '신영증권', '교보증권',

'한양증권', '코리아에셋투자증권', '유안타증권', 'KTB투자증권', '대신증권', '메리츠증권', '한화투자증권',

'현대차증권', '미래에셋증권', 'NH투자증권', '유진증권', '이베스트투자증권']

# 기타 금융업

financials\_etc\_name = ['DGB금융지주', '삼성카드', 'BNK금융지주', '나우IB', '한국금융지주', 'KB금융', '미래에셋벤처투자',

'하나금융지주', '메리츠금융지주', '신한지주', '아주IB투자', '우리금융지주', 'JB금융지주', '우리기술투자']

sector["financials"] = corp\_list.find\_by\_sector('신탁업 및 집합투자업') + \

[i for i in corp\_list.find\_by\_sector('금융 지원 서비스업') if i.corp\_name in financials\_support\_select] + \

[i for i in corp\_list.find\_by\_sector('기타 금융업') if i.corp\_name in financials\_etc\_select] + \

corp\_list.find\_by\_sector('보험업') + \

corp\_list.find\_by\_sector('은행 및 저축기관')

# len(sector["financials"])

# 67개, 352초

sector\_corpList["energy"] = corp\_list.find\_by\_sector('석유 정제품 제조업') + \

corp\_list.find\_by\_sector('연료 소매업') + \

corp\_list.find\_by\_sector('연료용 가스 제조 및 배관공급업')

sector\_corpList["energy"] = [i.corp\_name for i in sector\_corpList["energy"]]

# len(sector\_corpList["energy"])

# 17개, 962초

sector["materials"] = corp\_list.find\_by\_sector('기초 화학물질 제조업') + \

corp\_list.find\_by\_sector('건축자재, 철물 및 난방장치 도매업') + \

corp\_list.find\_by\_sector('기타 비금속 광물제품 제조업',) + \

corp\_list.find\_by\_sector('1차 비철금속 제조업') + \

corp\_list.find\_by\_sector('금속 주조업') + \

corp\_list.find\_by\_sector('기타 비금속광물 광업') + \

corp\_list.find\_by\_sector('기타 종이 및 판지 제품 제조업') + \

corp\_list.find\_by\_sector('나무제품 제조업') + \

corp\_list.find\_by\_sector('비료, 농약 및 살균, 살충제 제조업') + \

corp\_list.find\_by\_sector('시멘트, 석회, 플라스터 및 그 제품 제조업') + \

corp\_list.find\_by\_sector('제재 및 목재 가공업') + \

corp\_list.find\_by\_sector('펄프, 종이 및 판지 제조업') + \

corp\_list.find\_by\_sector('플라스틱제품 제조업') + \

corp\_list.find\_by\_sector('합성고무 및 플라스틱 물질 제조업') + \

corp\_list.find\_by\_sector('화학섬유 제조업')

# len(sector["materials"])

# 182개, 8897초

sector["industrials"] = corp\_list.find\_by\_sector('무기 및 총포탄 제조업') + \

corp\_list.find\_by\_sector('항공기,우주선 및 부품 제조업') + \

corp\_list.find\_by\_sector('구조용 금속제품, 탱크 및 증기발생기 제조업') + \

corp\_list.find\_by\_sector('건물 건설업') + \

corp\_list.find\_by\_sector('건물설비 설치 공사업') + \

corp\_list.find\_by\_sector('건축기술, 엔지니어링 및 관련 기술 서비스업') + \

corp\_list.find\_by\_sector('건축자재, 철물 및 난방장치 도매업') + \

corp\_list.find\_by\_sector('전기 및 통신 공사업') + \

corp\_list.find\_by\_sector('전기업') + \

corp\_list.find\_by\_sector('전동기, 발전기 및 전기 변환 · 공급 · 제어 장치 제조업') + \

corp\_list.find\_by\_sector('기타 전기장비 제조업') + \

corp\_list.find\_by\_sector('특수 목적용 기계 제조업') + \

corp\_list.find\_by\_sector('그외 기타 운송장비 제조업') + \

corp\_list.find\_by\_sector('일반 목적용 기계 제조업') + \

corp\_list.find\_by\_sector('그외 기타 개인 서비스업') + \

corp\_list.find\_by\_sector('회사 본부 및 경영 컨설팅 서비스업') + \

corp\_list.find\_by\_sector('사업시설 유지·관리 서비스업') + \

corp\_list.find\_by\_sector('시장조사 및 여론조사업') + \

corp\_list.find\_by\_sector('기타 전문 서비스업') + \

corp\_list.find\_by\_sector('기타 사업지원 서비스업') + \

corp\_list.find\_by\_sector('해상 운송업') + \

corp\_list.find\_by\_sector('항공 여객 운송업') + \

corp\_list.find\_by\_sector('육상 여객 운송업') + \

corp\_list.find\_by\_sector('운송장비 임대업') + \

corp\_list.find\_by\_sector('기타 운송관련 서비스업')

# 362개, 17794초 (실수로 시간 못 잼)

sector["discretionary"] = corp\_list.find\_by\_sector('자동차 재제조 부품 제조업') + \

corp\_list.find\_by\_sector('자동차 신품 부품 제조업') + \

corp\_list.find\_by\_sector('자동차 차체 및 트레일러 제조업') + \

corp\_list.find\_by\_sector('자동차용 엔진 및 자동차 제조업') + \

corp\_list.find\_by\_sector('교육지원 서비스업') + \

corp\_list.find\_by\_sector('귀금속 및 장신용품 제조업') + \

corp\_list.find\_by\_sector('기타 교육기관') + \

corp\_list.find\_by\_sector('기타 상품 전문 소매업') + \

corp\_list.find\_by\_sector('기타 생활용품 소매업') + \

corp\_list.find\_by\_sector('기타 섬유제품 제조업') + \

corp\_list.find\_by\_sector('기타 전문 도매업') + \

corp\_list.find\_by\_sector('무점포 소매업') + \

corp\_list.find\_by\_sector('봉제의복 제조업') + \

corp\_list.find\_by\_sector('상품 종합 도매업') + \

corp\_list.find\_by\_sector('생활용품 도매업') + \

corp\_list.find\_by\_sector('섬유, 의복, 신발 및 가죽제품 소매업') + \

corp\_list.find\_by\_sector('스포츠 서비스업') + \

corp\_list.find\_by\_sector('신발 및 신발 부분품 제조업') + \

corp\_list.find\_by\_sector('편조원단 제조업') + \

corp\_list.find\_by\_sector('가구 제조업') + \

corp\_list.find\_by\_sector('편조의복 제조업') + \

corp\_list.find\_by\_sector('직물직조 및 직물제품 제조업') + \

corp\_list.find\_by\_sector('일반 교습 학원') + \

corp\_list.find\_by\_sector('일반 및 생활 숙박시설 운영업') + \

corp\_list.find\_by\_sector('의복 액세서리 제조업') + \

corp\_list.find\_by\_sector('음식점업') + \

corp\_list.find\_by\_sector('유원지 및 기타 오락관련 서비스업') + \

corp\_list.find\_by\_sector('여행사 및 기타 여행보조 서비스업') + \

corp\_list.find\_by\_sector('가전제품 및 정보통신장비 소매업') + \

corp\_list.find\_by\_sector('가정용 기기 제조업') + \

corp\_list.find\_by\_sector('가죽, 가방 및 유사제품 제조업') + \

corp\_list.find\_by\_sector('개인 및 가정용품 임대업')

# 297개, 14073초

sector["staples"] = corp\_list.find\_by\_sector('기타 식품 제조업') + \

corp\_list.find\_by\_sector('비알코올음료 및 얼음 제조업') + \

corp\_list.find\_by\_sector('담배 제조업') + \

corp\_list.find\_by\_sector('생활용품 도매업') + \

corp\_list.find\_by\_sector('알코올음료 제조업') + \

corp\_list.find\_by\_sector('음·식료품 및 담배 도매업') + \

corp\_list.find\_by\_sector('음·식료품 및 담배 소매업') + \

corp\_list.find\_by\_sector('개인 및 가정용품 임대업') + \

corp\_list.find\_by\_sector('종합 소매업') + \

corp\_list.find\_by\_sector('낙농제품 및 식용빙과류 제조업') + \

corp\_list.find\_by\_sector('기타 화학제품 제조업')

# before; sector["staples"] -> 191

# 191개, 9058초

sector["health\_care"] = corp\_list.find\_by\_sector('의료용 기기 제조업') + \

corp\_list.find\_by\_sector('의료용품 및 기타 의약 관련제품 제조업') + \

corp\_list.find\_by\_sector('의약품 제조업') + \

corp\_list.find\_by\_sector('기타 과학기술 서비스업') + \

corp\_list.find\_by\_sector('기초 의약물질 및 생물학적 제제 제조업')

# sector["health\_care"] = 226

# 226개, 8964초

sector["it"] = corp\_list.find\_by\_sector("기타 정보 서비스업") + \

corp\_list.find\_by\_sector("반도체 제조업") + \

corp\_list.find\_by\_sector("소프트웨어 개발 및 공급업") + \

corp\_list.find\_by\_sector("자료처리, 호스팅, 포털 및 기타 인터넷 정보매개 서비스업") + \

corp\_list.find\_by\_sector("컴퓨터 및 통신장비 수리업") + \

corp\_list.find\_by\_sector("컴퓨터 및 주변장치 제조업") + \

corp\_list.find\_by\_sector("컴퓨터 프로그래밍, 시스템 통합 및 관리업") + \

corp\_list.find\_by\_sector("통신 및 방송 장비 제조업")

##len(sector[“it”]) = 361

# 361개, 15137초

sector["comm\_services"] = corp\_list.find\_by\_sector("광고업") + \

corp\_list.find\_by\_sector("기록매체 복제업") + \

corp\_list.find\_by\_sector("기타 전문 서비스업") + \

corp\_list.find\_by\_sector("영화, 비디오물, 방송프로그램 제작 및 배급업") + \

corp\_list.find\_by\_sector("오디오물 출판 및 원판 녹음업") + \

corp\_list.find\_by\_sector("유원지 및 기타 오락관련 서비스업") + \

corp\_list.find\_by\_sector("전기 통신업") + \

corp\_list.find\_by\_sector("창작 및 예술관련 서비스업") + \

corp\_list.find\_by\_sector("텔레비전 방송업")

##len(sector[“comm\_services”]) = 82

# 82개, 3894초

# ===== 재무데이터 로드 및 저장 예시 =====

# 임포트

import sys

import os

# import IPython

import multiprocessing

import copy

import pickle

import warnings

from datetime import datetime

from time import time

from matplotlib import font\_manager as fm, rc, rcParams

import matplotlib.pyplot as plt

import seaborn as sns

import numpy as np

from numpy import array, nan, random as rnd, where as which

import pandas as pd

from pandas import DataFrame as dataframe, Series as series, isna, isnull, read\_csv

# 재무데이터 로드 함수

def getFinancialStatement(tickers, from\_date, report\_type=["annual"]):

# dart fss module setting and loading the corporate list

corp\_list = dart.get\_corp\_list()

# create dictionary for dataset

fs\_dic = dict.fromkeys(tickers, None)

# for loop on tickers

for i in tickers:

print("Financial Statement Loading --->", i)

try:

# 재무제표 로딩

corp\_obj = corp\_list.find\_by\_corp\_name(i, exactly=True)[0]. \

extract\_fs(bgn\_de=from\_date, report\_tp=report\_type, separator=False)

# 데이터를 저장할 딕셔너리 초기화

fs\_dic[i] = {}

fs\_dic[i]["metrics"] = {}

# 한글만 추출하기 위한 정규표현식

expression\_hangul = re.compile('[^가-힣+]')

##### Balance Sheet

selected\_idx = []

selected\_column = []

# 필요한 열만 저장하기 위한 반복문

for idx, name in enumerate(corp\_obj["bs"]):

if name[1] == "label\_ko":

selected\_idx.append(idx)

selected\_column.append(name[1])

if type(name[1]) == tuple and name[1][0] == "연결재무제표":

selected\_idx.append(idx)

selected\_column.append(name[0])

tmp\_df = corp\_obj["bs"].iloc[:, selected\_idx]

# "label\_ko" 열에 대해 한글만 추출하여 재저장

tmp\_df.columns = selected\_column

tmp\_df["label\_ko"] = tmp\_df["label\_ko"].apply(lambda x: expression\_hangul.sub("", x).replace(" ", ""))

fs\_dic[i]["bs"] = tmp\_df.fillna(0.0)

#### Consolidated Income Statement

selected\_idx\_cis = []

selected\_column\_cis = []

for idx, name in enumerate(corp\_obj["cis"]):

# “label\_ko” 라고 되어있는 이름의 인덱스 및 이름 추가

if name[1] == "label\_ko":

selected\_idx\_cis.append(idx)

selected\_column\_cis.append(name[1])

# “연결재무제표” 라고 되어있는 이름의 인덱스 및 이름 추가

if type(name[1]) == tuple and name[1][0] == "연결재무제표":

selected\_idx\_cis.append(idx)

selected\_column\_cis.append(name[0])

tmp\_df = corp\_obj["cis"].iloc[:, selected\_idx\_cis]

tmp\_df.columns = selected\_column\_cis

tmp\_df["label\_ko"] = tmp\_df["label\_ko"].apply(lambda x: expression\_hangul.sub("", x).replace(" ", ""))

fs\_dic[i]["cis"] = tmp\_df.fillna(0.0)

#### Cash Flow Chart

selected\_idx\_cf = []

selected\_column\_cf = []

for idx, name in enumerate(corp\_obj["cf"]):

# “label\_ko” 라고 되어있는 이름의 인덱스 및 이름 추가

if name[1] == "label\_ko":

selected\_idx\_cf.append(idx)

selected\_column\_cf.append(name[1])

# “연결재무제표” 라고 되어있는 이름의 인덱스 및 이름 추가

if type(name[1]) == tuple and name[1][0] == "연결재무제표":

selected\_idx\_cf.append(idx)

selected\_column\_cf.append(name[0])

tmp\_df = corp\_obj["cf"].iloc[:, selected\_idx\_cf]

tmp\_df.columns = selected\_column\_cf

tmp\_df["label\_ko"] = tmp\_df["label\_ko"].apply(lambda x: expression\_hangul.sub("", x).replace(" ", ""))

fs\_dic[i]["cf"] = tmp\_df.fillna(0.0)

except:

print('EXCEPTION : Error occurs, delete the key')

del fs\_dic[i]

return fs\_dic

# easyIO 함수

def easyIO(x=None, path=None, op="r"):

tmp = None

if op == "r":

with open(path, "rb") as f:

tmp = pickle.load(f)

return tmp

elif op == "w":

tmp = {}

print(x)

if type(x) is dict:

for k in x.keys():

if "MLP" in k:

tmp[k] = {}

for model\_comps in x[k].keys():

if model\_comps != "model":

tmp[k][model\_comps] = copy.deepcopy(x[k][model\_comps])

print(F"INFO : {k} model is removed (keras)")

else:

tmp[k] = x[k]

if input("Write [y / n]: ") == "y":

with open(path, "wb") as f:

pickle.dump(tmp, f)

print("operation success")

else:

print("operation fail")

else:

print("Unknown operation type")

# getFinancialStatement 실행 이후 바로 저장

# tickers는 본인이 맡은 sector 선택

tickers = [i.corp\_name for i in sector["financials"]]

rawdata\_financials = getFinancialStatement(tickers, "20160101")

financials = copy.deepcopy(rawdata\_financials)

# 경로 및 파일이름은 본인 환경에 맞춰서 (이후 생성된 pickle 파일을 공유폴더에 업로드)

easyIO(financials, "./kdigital\_project3/fndata\_rawdata\_financials.pickle", op="w")