In [ ]:
```
pip install user-agent
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Collecting user-agent
  Downloading user_agent-0.1.10.tar.gz (20 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: six in /usr/local/lib/python3.8/dist-packages (from user-agent) (1.15.0)
Building wheels for collected packages: user-agent
  Building wheel for user-agent (setup.py) ... done
  Created wheel for user-agent: filename=user_agent-0.1.10-py3-none-any.whl size=18982 sha256=b07fcf17c4ad88979392a3c7669838f0bb0c95156716f1d18c8b80b290172b3f
  Stored in directory: /root/.cache/pip/wheels/31/2b/5b/d3d4cef9b2818758251d0d556cf7a01550fa05df2e4bcd012e
Successfully built user-agent
Installing collected packages: user-agent
Successfully installed user-agent-0.1.10
```

In [ ]:
```
! python -m pip install pymongo==3.7.
! python -m pip install pymongo[srv]
```

```
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Collecting pymongo==3.7.
  Downloading pymongo-3.7.0.tar.gz (626 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 626.8/626.8 KB 11.0 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: pymongo
  Building wheel for pymongo (setup.py) ... done
  Created wheel for pymongo: filename=pymongo-3.7.0-cp38-cp38-linux_x86_64.whl size=436211 sha256=8bb6b3bb20a379ec106ab32a523482ccf502a31bc31cb8d5ab4d12014f3a2c24
  Stored in directory: /root/.cache/pip/wheels/33/33/8a/e080ffb7c749ca54a191fbf42095b6e4fcb66bd305a3f2b1b5
Successfully built pymongo
Installing collected packages: pymongo
  Attempting uninstall: pymongo
    Found existing installation: pymongo 4.3.3
    Uninstalling pymongo-4.3.3:
      Successfully uninstalled pymongo-4.3.3
Successfully installed pymongo-3.7.0
Looking in indexes: https://pypi.org/simple, (https://pypi.org/simple,) https://us-python.pkg.dev/colab-wheels/public/simple/ (https://us-python.pkg.dev/colab-wheels/public/simple/)
Requirement already satisfied: pymongo[srv] in /usr/local/lib/python3.8/dist-packages (3.7.0)
Collecting dnspython<2.0.0,>=1.13.0
  Downloading dnspython-1.16.0-py2.py3-none-any.whl (188 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 188.4/188.4 KB 10.0 MB/s eta 0:00:00
Installing collected packages: dnspython
  Attempting uninstall: dnspython
    Found existing installation: dnspython 2.3.0
    Uninstalling dnspython-2.3.0:
      Successfully uninstalled dnspython-2.3.0
Successfully installed dnspython-1.16.0
```

In [ ]:
```
# Import Dependencies
import pandas as pd
from urllib.request import urlopen
from urllib.request import Request
from bs4 import BeautifulSoup
import json
from user_agent import generate_user_agent
import pymongo
from pymongo import MongoClient
import re
from bson.json_util import dumps, loads
```

In [ ]:
```
#randomizing your user agent to be able to download a little more every time
headers={'User-Agent': generate_user_agent()}
```

#Part I: Deploy MongoDB

In [ ]:
```
user='ACF'
password='Qwe123123'

classclient = pymongo.MongoClient(f"mongodb+srv://{user}:{password}@cathyzhang.ekuyu27.mongodb.net/?retryWrites=true&w=majority")
classDb = classclient.classDb
```

In [ ]:
```
classDb
```

Out[6]:
```
Database(MongoClient(host=['ac-ptphd0u-shard-00-01.ekuyu27.mongodb.net:27017', 'ac-ptphd0u-shard-00-00.ekuyu27.mongodb.net:2701
7', 'ac-ptphd0u-shard-00-02.ekuyu27.mongodb.net:27017'], document_class=dict, tz_aware=False, connect=True, authsource='admin',
replicaset='atlas-tmxdt9-shard-0', ssl=True, retrywrites=True, w='majority'), 'classDb')
```

In [ ]:
```
classDb['EtfList_new'].create_index([("ticker", pymongo.ASCENDING)],unique=True)
```

Out[7]:
```
'ticker_1'
```

Part II: Get ETFs Holding dataset

Step 1: Get ETFs Master Name List

From dropbox, get 188 ETFs with AUM bigger than $2bn https://www.dropbox.com/s/1a4u95oj30x68k8/ETF1.xlsx?raw=1 (https://www.dropbox.com/s/1a4u95oj30x68k8/ETF1.xlsx?raw=1)

We download our own newest master list from Bloomberg

By this way we get 246 ETFS with AUM bigger than $2bn

In [ ]:
```
url = 'https://dl.dropboxusercontent.com/s/qu30y8u1a6xtvcm/ETF1_se5gn5mg.xlsx?dl=0'
etf = pd.read_excel(url,sheet_name='Worksheet')
etf
```

Out[8]:

	Name	Ticker	30D Vol	Class Assets (MLN USD)	Fund Assets (MLN USD)	YTD Rtn	YTD Class Flow	(M USD)	Unnamed: 8	1M Flow	(M USD).1	Unnamed: 11	12M Yld	Expe R
0	Median	NaN	21.14k	123.44	121.46	+7.31%	0.00	NaN	NaN	0.00	NaN	NaN	+1.55%	+0.
1	SPDR S&P 500 ETF Trust	SPY US	80.87M	379939.31	379939.31	+6.08%	4205.09	4205.0920	NaN	-3218.21	-3218.21000	NaN	+1.56%	+0.
2	iShares Core S&P 500 ETF	IVV US	4.62M	306688.91	306688.91	+6.10%	-517.09	-517.0875	NaN	-1168.54	-1168.53700	NaN	+1.57%	+0.
3	Vanguard Total Stock Market ETF	VTI US	4.18M	280070.66	280070.66	+6.67%	1659.60	1659.6040	NaN	2417.26	2417.26200	NaN	+1.56%	+0.
4	Vanguard S&P 500 ETF	VOO US	4.24M	278653.5	278653.5	+6.13%	625.47	625.4667	NaN	272.50	272.49690	NaN	+1.59%	+0.
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1922	Subversive Mental Health ETF	SANE US	--	--	0.62	+0.04%	0.00	0.0000	NaN	0.00	0.00000	NaN	--	+0.
1923	Newday Sustainable Development Equity ETF	SDGS US	158.03	--	1.49	+4.72%	0.00	0.0000	NaN	0.00	0.00000	NaN	+0.16%	+0.
1924	Day Hagan/Ned Davis Research Smart Sector Inte...	SSXU US	9.21k	--	12.43	+9.49%	1.33	1.3285	NaN	1.33	1.32850	NaN	+0.60%	+0.
1925	Strive 1000 Value ETF	STXV US	1.92k	--	2.61	+4.76%	0.00	0.0000	NaN	0.00	0.00000	NaN	+0.45%	+0.
1926	Clockwise Capital Innovation ETF	TIME US	19.51k	--	43.6	+11.95%	0.00	0.0000	NaN	5.23	5.22925	NaN	--	+0.

1927 rows × 18 columns

In [ ]:
```
#Drop first median row
etf = etf.drop(etf.index[0])
etf
```

Out[9]:

	Name	Ticker	30D Vol	Class Assets (MLN USD)	Fund Assets (MLN USD)	YTD Rtn	YTD Class Flow	(M USD)	Unnamed: 8	1M Flow	(M USD).1	Unnamed: 11	12M Yld	Exp
1	SPDR S&P 500 ETF Trust	SPY US	80.87M	379939.31	379939.31	+6.08%	4205.09	4205.0920	NaN	-3218.21	-3218.21000	NaN	+1.56%	+(
2	iShares Core S&P 500 ETF	IVV US	4.62M	306688.91	306688.91	+6.10%	-517.09	-517.0875	NaN	-1168.54	-1168.53700	NaN	+1.57%	+(
3	Vanguard Total Stock Market ETF	VTI US	4.18M	280070.66	280070.66	+6.67%	1659.60	1659.6040	NaN	2417.26	2417.26200	NaN	+1.56%	+(
4	Vanguard S&P 500 ETF	VOO US	4.24M	278653.5	278653.5	+6.13%	625.47	625.4667	NaN	272.50	272.49690	NaN	+1.59%	+(
5	Invesco QQQ Trust Series 1	QQQ US	48.99M	157563.78	157563.78	+11.26%	-4874.20	-4874.1970	NaN	-4245.36	-4245.36300	NaN	+0.72%	+(
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1922	Subversive Mental Health ETF	SANE US	--	--	0.62	+0.04%	0.00	0.0000	NaN	0.00	0.00000	NaN	--	+(
1923	Newday Sustainable Development Equity ETF	SDGS US	158.03	--	1.49	+4.72%	0.00	0.0000	NaN	0.00	0.00000	NaN	+0.16%	+(
1924	Day Hagan/Ned Davis Research Smart Sector Inte...	SSXU US	9.21k	--	12.43	+9.49%	1.33	1.3285	NaN	1.33	1.32850	NaN	+0.60%	+(
1925	Strive 1000 Value ETF	STXV US	1.92k	--	2.61	+4.76%	0.00	0.0000	NaN	0.00	0.00000	NaN	+0.45%	+(
1926	Clockwise Capital Innovation ETF	TIME US	19.51k	--	43.6	+11.95%	0.00	0.0000	NaN	5.23	5.22925	NaN	--	+(

1926 rows × 18 columns

In [ ]:
```
#Drop rows contains '--'
etf = etf[~etf.isin(['--']).any(axis=1)]

#Select etf Fund assets bigger than 2bn
etf = etf[etf['Fund Assets (MLN USD)']>2000]

etf
```

Out[10]:

	Name	Ticker	30D Vol	Class Assets (MLN USD)	Fund Assets (MLN USD)	YTD Rtn	YTD Class Flow	(M USD)	Unnamed: 8	1M Flow	(M USD).1	Unnamed: 11	12M Yld	E
1	SPDR S&P 500 ETF Trust	SPY US	80.87M	379939.31	379939.31	+6.08%	4205.09	4205.092000	NaN	-3218.21	-3218.210000	NaN	+1.56%	
2	iShares Core S&P 500 ETF	IVV US	4.62M	306688.91	306688.91	+6.10%	-517.09	-517.087500	NaN	-1168.54	-1168.537000	NaN	+1.57%	
3	Vanguard Total Stock Market ETF	VTI US	4.18M	280070.66	280070.66	+6.67%	1659.60	1659.604000	NaN	2417.26	2417.262000	NaN	+1.56%	
4	Vanguard S&P 500 ETF	VOO US	4.24M	278653.5	278653.5	+6.13%	625.47	625.466700	NaN	272.50	272.496900	NaN	+1.59%	
5	Invesco QQQ Trust Series 1	QQQ US	48.99M	157563.78	157563.78	+11.26%	-4874.20	-4874.197000	NaN	-4245.36	-4245.363000	NaN	+0.72%	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
264	iShares MSCI Australia ETF	EWA US	2.40M	2047.66	2047.66	+12.19%	169.71	169.705600	NaN	169.71	169.705600	NaN	+4.71%	
265	Global X Copper Miners ETF	COPX US	514.65k	2025.91	2025.91	+16.93%	60.21	60.211600	NaN	74.40	74.403600	NaN	+2.69%	
266	First Trust Energy AlphaDEX Fund	FXN US	971.22k	2015.16	2015.16	+2.81%	-6.87	-6.870397	NaN	4.22	4.222215	NaN	+2.22%	
267	iShares Europe ETF	IEV US	470.02k	2011.65	2011.65	+9.41%	204.58	204.583700	NaN	204.58	204.583700	NaN	+2.80%	
268	WisdomTree Emerging Markets High Dividend Fund	DEM US	479.17k	2010.5	2010.5	+9.85%	89.96	89.955640	NaN	89.96	89.955640	NaN	+7.84%	

246 rows × 18 columns

In [ ]:
```
#Extra ticker from dataset
etf_ticker= etf['Ticker'].map(lambda x:x[0:len(x)-3]).tolist()
print(etf_ticker)
```

```
['SPY', 'IVV', 'VTI', 'VOO', 'QQQ', 'VEA', 'VTV', 'IEFA', 'VUG', 'VWO', 'IEMG', 'IJR', 'IJH', 'VIG', 'IWF', 'VXUS', 'IWM', 'VO',
'IWD', 'VYM', 'EFA', 'SCHD', 'VB', 'XLE', 'VGT', 'XLK', 'ITOT', 'XLV', 'RSP', 'VEU', 'XLF', 'IXUS', 'SCHX', 'SCHF', 'USMV', 'DI
A', 'IVW', 'IWR', 'IWB', 'EEM', 'VT', 'SDY', 'VBR', 'IVE', 'VV', 'DGRO', 'DVY', 'SCHB', 'ESGU', 'MDY', 'JEPI', 'ACWI', 'QUAL',
'VGK', 'VHT', 'VOE', 'XLP', 'EFV', 'SPYV', 'XLU', 'SPLG', 'SPDW', 'SCHG', 'SPYG', 'SCHA', 'XLY', 'VXF', 'XLI', 'GDX', 'IUSV', 'I
WS', 'VBK', 'FVD', 'HDV', 'IWN', 'IWP', 'COWZ', 'MTUM', 'IUSG', 'EFG', 'NOBL', 'IWV', 'GSLC', 'SPLV', 'IDEV', 'FNDX', 'MGK', 'SC
HV', 'VONG', 'IWO', 'VOT', 'EWJ', 'SCHM', 'MCHI', 'SPYD', 'XLC', 'FNDF', 'VFH', 'SCHE', 'IBB', 'IYW', 'FTCS', 'VDE', 'VSS', 'RDV
Y', 'IJJ', 'GUNR', 'VLUE', 'DGRW', 'IJS', 'OEF', 'IJK', 'EFAV', 'BBJP', 'ESGD', 'MOAT', 'EZU', 'SPEM', 'VONV', 'BBEU', 'SOXX',
'AMLP', 'EEMV', 'VOOG', 'QYLD', 'VDC', 'QQQM', 'FXI', 'VPL', 'FNDA', 'XLB', 'IHI', 'PRF', 'BBCA', 'ESGV', 'SPMD', 'VTWO', 'MGV',
'SPTM', 'VYMI', 'VPU', 'FTEC', 'AVUV', 'IJT', 'FDL', 'JIRE', 'ITA', 'ICLN', 'IDV', 'SPSM', 'EWZ', 'IGV', 'IWY', 'BBAX', 'FNDE',
'CIBR', 'IQLT', 'SPHD', 'ACWX', 'ACWV', 'VIGI', 'ESGE', 'SPHQ', 'IXJ', 'XOP', 'EWY', 'SLYV', 'GDXJ', 'IEUR', 'EWT', 'VCR', 'ONE
Q', 'IGF', 'EWC', 'GNR', 'DBEF', 'DLN', 'RPV', 'VIS', 'LIT', 'PAVE', 'SCHC', 'MGC', 'VONE', 'AVUS', 'DSI', 'IOO', 'DON', 'EWU',
'FV', 'EMXC', 'AAXJ', 'SUSA', 'VSGX', 'IYH', 'VOOV', 'XT', 'USSG', 'SUSL', 'FHLC', 'HEFA', 'GSIE', 'BBIN', 'VAW', 'JHMM', 'IXN',
'OIH', 'FNDC', 'VOX', 'AVEM', 'SKYY', 'URTH', 'ASHR', 'EMLP', 'DGS', 'DIVO', 'XME', 'MDYV', 'AVDV', 'IGM', 'SCHK', 'KRE', 'NFR
A', 'AVDE', 'OMFL', 'SLYG', 'SPGP', 'RYT', 'FEZ', 'PBUS', 'XSOE', 'VIOO', 'XYLD', 'RPG', 'FUTY', 'EPP', 'IXC', 'IYF', 'CDC', 'PR
FZ', 'PTLC', 'EWA', 'COPX', 'FXN', 'IEV', 'DEM']
```

In [ ]:
```
```

Step 2: download holding data from iShare ETF

There are 84 ETFs details holding

In [ ]:
```
url='https://www.ishares.com/us/products/etf-investments#/?productView=etf&pageNumber=1&sortColumn=totalNetAssets&sortDirection=des
resp = urlopen(Request(url=url,headers={'user-agent': 'my-app/0.0.1'}))
```

In [ ]:
```
html = BeautifulSoup(resp, features="lxml")
```

In [ ]:
```
print(html.prettify())
```

```
<!DOCTYPE html>
<html lang="en" prefix="og: http://ogp.me/ns#" (http://ogp.me/ns#") xml:lang="en" xmlns="http://www.w3.org/1999/xhtml">
 <head>
  <title>
   iShares ETF Investments List | iShares - BlackRock
  </title>
  <meta content="text/html;charset=utf-8" http-equiv="Content-type"/>
  <meta content="width=device-width, initial-scale=1" name="viewport"/>
  <meta content="product list, product screener, ishares list, ishares product list" name="keywords"/>
  <meta content="Find the full list of iShares ETFs here. Use these low cost, tax efficient funds to strengthen the core of y
our portfolio." name="description"/>
  <meta content="iShares ETF Investments List | iShares - BlackRock" name="articleTitle"/>
  <meta content="Find the full list of iShares ETFs here. Use these low cost, tax efficient funds to strengthen the core of y
our portfolio." name="pageSummary"/>
  <meta content="iShares ETF Investments List | iShares - BlackRock" name="featureImageAltText"/>
  <meta content="iShares ETF Investments List | iShares - BlackRock" property="og:title"/>
  <meta content="website" property="og:type"/>
  <meta content="/blk-one-c-assets/include/common/images/blackrock_logo.png" property="og:image"/>
  <meta content="BlackRock" property="og:site_name"/>
```

In [ ]:
```
#dictionary of the etf with address of detial csv from ishares
tick_to_url = {}
for row in html.find_all('tr'):
  try:
    for data in row.find_all('a'):
      if len(data.text)>0 and len(data.text)<5:
        tick_to_url[data.text]='https://www.ishares.com/'+data['href']+'/1467271812596.ajax?fileType=csv&fileName=IWM_holdings&data
  except:
    pass

def get_iShare_holdings(etf_name):
  """
  get etf holding data from ishare
  save to dictionary-- key: holding ticker && value: Its weight
  Only look at the Equity
  """
  #First 9 rows are summary and not useful here
  df=pd.read_csv(tick_to_url[etf_name],skiprows=range(0,9), thousands=',')
  df=df[df['Asset Class']=='Equity']
  df['Ticker'] = df['Ticker'].str.extractall(r"([A-Za-z]+)").groupby(level=0).agg(''.join)
  df = df[df['Ticker'].notna()]
  df['Weight'] = df["Market Value"]/df["Market Value"].sum()
  iShare_dict = dict(zip(df["Ticker"].str.strip(), df['Weight']))

  return iShare_dict
```

In [ ]:
```
#test
get_iShare_holdings('IWM')
```

Out[16]:
```
{'IRDM': 0.002988384039871269,
 'MTDR': 0.00298106876614678 3,
 'CROX': 0.00295198829930797,
 'SAIA': 0.0028951607143953917,
 'INSP': 0.0028553325056804076,
 'EME': 0.002805729178440836,
 'RBC': 0.0027806365597935105,
 'HALO': 0.0027539710843730524,
 'TXRH': 0.0026957427630767226,
 'SWAV': 0.002673283192285473,
 'CHX': 0.0026457549290484876,
 'ADC': 0.0026088865036366436,
 'MUR': 0.0025794177224815282,
 'STAG': 0.002570549257064322,
 'CMC': 0.0025420680593351997,
 'LNW': 0.0024676394186059067,
 'KRTX': 0.00240796692455829,
 'KNSL': 0.002406444916725896,
 'CHRD': 0.002381735933689922,
```

In [ ]:
```
#Get 84 ETF holding from iShare
empty_etf = [] # list of etfs that didn't get holding data from iShare
for i in range(0,len(etf_ticker)):
  ticker_name = etf_ticker[i]
  try:
    temp_dic = {}
    value = get_iShare_holdings(ticker_name)
    temp_dic = {'ticker': ticker_name,
                'Holdings': value}
    #Pass result to MangoDB
    try:
      result = classDb['EtfList_new'].insert_one(temp_dic)
    except:
      pass
  except:
    empty_etf.append(ticker_name)
    pass
```

In [ ]:
```
#print(empty_etf)
len(empty_etf)
```

Out[18]: 162

Step3: Get holding data from invesco.com

There are 14 etfs holiding data -- totally 98 etfs

In [ ]:
```
def get_invesco_holding(etf_name):
    """
    get etf holding data from invesco
    save to dictionary-- key: holding ticker && value: Its weight
    Only look at the Equity/Common Stock
    """
    tic_URL = f'https://www.invesco.com/us/financial-products/etfs/holdings/main/holdings/0?audienceType=Investor&action=download&tic
    df = pd.read_csv(tic_URL)
    df = df[df['Class of Shares'] == 'Common Stock']
    df['Holding Ticker'] = df['Holding Ticker'].str.extractall(r"([A-Za-z]+)").groupby(level=0).agg(''.join)
    df = df[df['Holding Ticker'].notna()]
    #market value in this dataset are include "," and type is object
    df["MarketValue"] = df["MarketValue"].str.replace(',', '').astype(float) #change object to float for the following math
    df['weight'] = df["MarketValue"]/df["MarketValue"].sum()
    invesco_dict = dict(zip(df["Holding Ticker"].str.strip(), df['weight']))

    return invesco_dict
```

In [ ]:
```
#test
get_invesco_holding('PHDG')
```

Out[13]:
```
{'AAPL': 0.06824472342650949,
 'MSFT': 0.05999597268640554,
 'AMZN': 0.026206786032015213,
 'BRKB': 0.01676654054674594,
 'GOOGL': 0.01676335067049753,
 'NVDA': 0.016092898113810573,
 'TSLA': 0.015496700510742988,
 'GOOG': 0.014921013017712448,
 'XOM': 0.014390621218254768,
 'UNH': 0.01372964173063256,
 'JNJ': 0.012622550671676542,
 'JPM': 0.012405085915239535,
 'META': 0.01196832954812615,
 'V': 0.011101497075986342,
 'PG': 0.00984648273908795,
 'HD': 0.00982602642586078,
 'MA': 0.009337408357748997,
 'CVX': 0.009024144245680429,
 'MRK': 0.008239118975534839,
```

In [ ]:
```
remain = []
for i in range(0,len(empty_etf)):
  ticker_name = empty_etf[i]
  try:
    temp_dic = {}
    value = get_invesco_holding(ticker_name)
    temp_dic = {'ticker': ticker_name,
                'Holdings': value}
    #Pass result to MangoDB
    try:
      result = classDb['EtfList_new'].insert_one(temp_dic)
    except:
      pass
  except:
    remain.append(ticker_name)
    pass
```

In [ ]:
```
print(remain)
len(remain)
```

```
['SPY', 'VTI', 'VOO', 'VEA', 'VTV', 'VUG', 'VWO', 'VIG', 'VXUS', 'VO', 'VYM', 'SCHD', 'VB', 'XLE', 'VGT', 'XLK', 'XLV', 'VEU',
'XLF', 'SCHX', 'SCHF', 'DIA', 'VT', 'SDY', 'VBR', 'VV', 'SCHB', 'MDY', 'JEPI', 'VGK', 'VHT', 'VOE', 'XLP', 'SPYV', 'XLU', 'SPL
G', 'SPDW', 'SCHG', 'SPYG', 'SCHA', 'XLY', 'VXF', 'XLI', 'GDX', 'VBK', 'FVD', 'COWZ', 'NOBL', 'GSLC', 'FNDX', 'MGK', 'SCHV', 'VO
NG', 'VOT', 'SCHM', 'SPYD', 'XLC', 'FNDF', 'VFH', 'SCHE', 'FTCS', 'VDE', 'VSS', 'RDVY', 'GUNR', 'DGRW', 'BBJP', 'MOAT', 'SPEM',
'VONV', 'BBEU', 'AMLP', 'VOOG', 'QYLD', 'VDC', 'VPL', 'FNDA', 'XLB', 'BBCA', 'ESGV', 'SPMD', 'VTWO', 'MGV', 'SPTM', 'VYMI', 'VP
U', 'FTEC', 'AVUV', 'FDL', 'JIRE', 'SPSM', 'BBAX', 'FNDE', 'CIBR', 'VIGI', 'XOP', 'SLYV', 'GDXJ', 'VCR', 'ONEQ', 'GNR', 'DBEF',
'DLN', 'VIS', 'LIT', 'PAVE', 'SCHC', 'MGC', 'VONE', 'AVUS', 'DON', 'FV', 'VSGX', 'VOOV', 'USSG', 'FHLC', 'HEFA', 'GSIE', 'BBIN',
'VAW', 'JHMM', 'OIH', 'FNDC', 'VOX', 'AVEM', 'SKYY', 'ASHR', 'EMLP', 'DGS', 'DIVO', 'XME', 'MDYV', 'AVDV', 'SCHK', 'KRE', 'NFR
A', 'AVDE', 'SLYG', 'FEZ', 'XSOE', 'VIOO', 'XYLD', 'FUTY', 'CDC', 'PTLC', 'COPX', 'FXN', 'DEM']
```

Out[22]: 148

Step 4: get holding data from stockAnalysis.com

148 ETFs with holding data

need to reset runtime to obtain full 148 dataset

use mongoDB to store data

In [ ]:
```
def get_stockAnalysis_holding(etf_name):
  """
  get etf holding data from stockAnalysis
  save to dictionary-- key: holding ticker && value: Its weight
  assume all equity, if ticker is letter and less than or equal to 4
  """
  url = f'https://stockanalysis.com/etf/{etf_name}/holdings/'
  headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/
  req = Request(url=url,headers=headers)
  resp = urlopen(req)
  html = BeautifulSoup(resp, features="lxml")

  holdings={}
  ticker = []
  weight = []
  for row in html.find('table',{'class':'svelte-1l0crez'}).find_all('tr'):
    cells=[d.text for d in row.find_all('td')]
    if len(cells)<4:
      continue
    tick = re.sub('[^a-zA-Z]+', '', cells[1])
    wgt = float(cells[3][:-1])
    ticker.append(tick)
    weight.append(wgt)
  data = {'ticker':ticker, 'weight':weight}
  df = pd.DataFrame(data)
  df = df.reset_index()
  df = df[df['ticker'].notna()]
  df['weight_n']= df['weight'] / df['weight'].sum()

  holdings = dict(zip(df['ticker'],df['weight_n']))

  return holdings
```

In [ ]:
```
exist_list = []
cur = classDb.EtfList_new.find({},{"ticker":1})

for doc in cur:
  exist_list.append(doc['ticker'])

exist_list
```

Out[24]:
```
['IVV',
 'IEFA',
 'IEMG',
 'IJR',
 'IJH',
 'IWF',
 'IWM',
 'IWD',
 'EFA',
 'ITOT',
 'IXUS',
 'USMV',
 'IVW',
 'IWR',
 'IWB',
 'EEM',
 'IVE',
 'DGRO',
 'DVY',
 'EGOV'
```

In [ ]:
```
len(exist_list)
```

Out[25]:
```
246
```

In [ ]:
```
#get list of etf without holding
without_holding =[]
for item in remain:
  if item not in exist_list:
    without_holding.append(item)

without_holding
```

Out[26]: []

In [ ]:
```
len(without_holding)
```

Out[27]: 0

In [ ]:
```
#get_stockAnalysis_holding('SPY')
```

In [ ]:
```
remain1 = []
for i in range(0,len(without_holding)):
  ticker_name = without_holding[i]
  try:
    temp_dic = {}
    value = get_stockAnalysis_holding(ticker_name)
    temp_dic = {'ticker': ticker_name,
                'Holdings': value}
    #Pass result to MangoDB
    try:
      result = classDb['EtfList_new'].insert_one(temp_dic)
    except:
      pass
  except:
    remain1.append(ticker_name)
    pass
```

In [ ]:
```
len(remain1)
```

Out[30]: 0

In [ ]:
```
remain1
```

Out[31]: []

In [ ]:
```
#get_stockAnalysis_holding('NOBL')
```

In [ ]:
```
#specific for 'NOBL'
# we find there is na in weight
url = f'https://stockanalysis.com/etf/NOBL/holdings/'
headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/53
req = Request(url=url,headers=headers)
resp = urlopen(req)
html = BeautifulSoup(resp, features="lxml")

holdings={}
ticker = []
weight = []
for row in html.find('table',{'class':'svelte-1l0crez'}).find_all('tr'):
  cells=[d.text for d in row.find_all('td')]
  if len(cells)<4:
    continue
  tick = re.sub('[^a-zA-Z]+', '', cells[1])
  wgt = cells[3][:-1]
  ticker.append(tick)
  weight.append(wgt)
data = {'ticker':ticker, 'weight':weight}
df = pd.DataFrame(data)
df = df.reset_index()
df = df[df['ticker'].notna()]
df = df[df.weight != 'n/']
df['weight_n']= df['weight'].astype(float) / df['weight'].astype(float).sum()

holdings = dict(zip(df['ticker'],df['weight_n']))
```

In [ ]:
```
holdings
```

Out[34]: {'BEN': 0.019837691614066726,
'CAT': 0.019837691614066726,
'APD': 0.018635407273820258,
'PNR': 0.018034265103697024,
'NUE': 0.01793407474200982,
'SPGI': 0.017633503656948202,
'FRT': 0.017132551848512173,
'AFL': 0.017132551848512173,
'AOS': 0.017032361486824965,
'LIN': 0.01693217112513776,
'O': 0.016631600040076144,
'ITW': 0.016631600040076144,
'PPG': 0.01653140967838894,
'ATO': 0.01653140967838894,
'ROP': 0.01643121931670173,
'DOV': 0.01643121931670173,
'EXPD': 0.016331028955014527,
'ED': 0.016230838593327322,
'CINF': 0.016230838593327322,
'ABT': 0.016230838593327322,
'SWK': 0.016230838593327322,
'KMB': 0.016130648231640118,
'CB': 0.016130648231640118,
'BDX': 0.016130648231640118,
'WST': 0.01603045786995291,
'GWW': 0.015930267508265705,
'PG': 0.0158300771465785,
'TROW': 0.015729886784891293,
'CHD': 0.015729886784891293,
'SHW': 0.015729886784891293,
'VFC': 0.01562969642320409,
'LOW': 0.01562969642320409,
'CTAS': 0.015529506061516882,
'EMR': 0.015529506061516882,
'XOM': 0.015429315699829676,
'MCD': 0.015429315699829676,
'KO': 0.015228934976455265,
'WBA': 0.015228934976455265,
'CAH': 0.015128744614768059,
'AMCR': 0.015128744614768059,
'TGT': 0.015128744614768059,

```
 'CLX': 0.014828173529706442,
 'NEE': 0.014828173529706442,
 'BFB': 0.014828173529706442,
 'IBM': 0.014828173529706442,
 'WMT': 0.014828173529706442,
 'ECL': 0.014727983168019236,
 'CL': 0.01462779280633203,
 'CVX': 0.01462779280633203,
 'ALB': 0.014527602444644825,
 'MKC': 0.014527602444644825,
 'HRL': 0.014427412082957619,
 'MDT': 0.014227031359583206,
 'GPC': 0.014126840997896002,
 'ABBV': 0.014026650636208796,
 'ESS': 0.014026650636208796,
 'SYY': 0.01392646027452159,
 'PEP': 0.013826269912834383,
 'MMM': 0.013826269912834383,
 'JNJ': 0.01372607955114718,
 'ADP': 0.013625889189459974,
 'GD': 0.01352569882777277,
 'ADM': 0.013325318104398357,
 'BRO': 0.013225127742711151}
```

In [ ]:
```
temp_dic = {}
temp_dic = {'ticker': 'NOBL',
            'Holdings': holdings}
#Pass result to MangoDB
try:
  result = classDb['EtfList_new'].insert_one(temp_dic)
except:
  pass
```

##Step 5: Get full dataset from mongoDB

Convert to DataFrame

In [ ]:
```
etf_cur = classDb.EtfList_new.find()
```

In [ ]:
```
list_cur = list(etf_cur)
```

In [ ]:
```
list_cur
```

Out[38]:
```
[{'_id': ObjectId('63d741139b6177006e4ee4e8'),
  'ticker': 'IVV',
  'Holdings': {'AAPL': 0.06406294725137868,
   'MSFT': 0.05430807403894186,
   'AMZN': 0.026639517858015396,
   'GOOGL': 0.01742462669550869,
   'BRKB': 0.016353111976997854,
   'GOOG': 0.015654509735146982,
   'NVDA': 0.014886734729334742,
   'TSLA': 0.01401808078153915,
   'XOM': 0.01397746179479785,
   'UNH': 0.01333232001158127,
   'JNJ': 0.012912344906869929,
   'JPM': 0.012083080476648437,
   'V': 0.011109022011242487,
   'META': 0.010017096941442172,
   'PG': 0.009779163314661946,
   'HD': 0.009517738688560515,
   'CVX': 0.009371785965521476,
```

In [ ]:
```
final_etf = pd.DataFrame(list_cur)
```

In [ ]:
```
final_etf
```

Out[40]:

	_id	ticker	Holdings
0	63d741139b6177006e4ee4e8	IVV	{'AAPL': 0.06406294725137868, 'MSFT': 0.054308...
1	63d741159b6177006e4ee4e9	IEFA	{'NESN': 0.024309578051267204, 'ASML': 0.02037...
2	63d741169b6177006e4ee4ea	IEMG	{'RELIANCE': 0.02691150682230978, 'VALE': 0.02...
3	63d741179b6177006e4ee4eb	IJR	{'ADC': 0.006969187214429515, 'UFPI': 0.005923...
4	63d741189b6177006e4ee4ec	IJH	{'FICO': 0.007450394880033458, 'RS': 0.0060110...
...	...	...	...
241	63d74b67f9a12b008a203509	PTLC	{'AAPL': 0.07736077481840194, 'MSFT': 0.065496...
242	63d74b67f9a12b008a20350a	COPX	{'ANTOL': 0.0601939806019398, 'HK': 0.00449955...
243	63d74b68f9a12b008a20350b	FXN	{'DINO': 0.04439112177564487, 'PDCE': 0.044391...
244	63d74b69f9a12b008a20350c	DEM	{'VALESA': 0.07611990569215224, 'TW': 0.001010...
245	63d74e05f9a12b008a20350d	NOBL	{'CAT': 0.019340615292113436, 'BEN': 0.0188395...

246 rows × 3 columns

In [ ]:
```
new_dic = []
for item in final_etf['Holdings']:
  new_dic.append(item)
new_dic
```

Out[41]:
```
[{'AAPL': 0.06406294725137868,
  'MSFT': 0.05430807403894186,
  'AMZN': 0.026639517858015396,
  'GOOGL': 0.01742462669550869,
  'BRKB': 0.016353111976997854,
  'GOOG': 0.015654509735146982,
  'NVDA': 0.014886734729334742,
  'TSLA': 0.01401808078153915,
  'XOM': 0.01397746179479785,
  'UNH': 0.01333232001158127,
  'JNJ': 0.012912344906869929,
  'JPM': 0.012083080476648437,
  'V': 0.011109022011242487,
  'META': 0.010017096941442172,
  'PG': 0.009779163314661946,
  'HD': 0.009517738688560515,
  'CVX': 0.009371785965521476,
  'MA': 0.00932117805384382,
  'LLY': 0.007920486257888252,
```

In [ ]:
```
new = pd.DataFrame(new_dic)
ticker_etf = final_etf['ticker'].tolist()
new['ticker'] = ticker_etf
```

In [ ]:
```
new.set_index('ticker')
```

Out[43]:

ticker	AAPL	MSFT	AMZN	GOOGL	BRKB	GOOG	NVDA	TSLA	XOM	UNH	...	PGASJK	BSANTANDERSN	ISAT
IVV	0.064063	0.054308	0.026640	0.017425	0.016353	0.015655	0.014887	0.014018	0.013977	0.013332	...	NaN	NaN	Na
IEFA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
IEMG	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
IJR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
IJH	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
PTLC	0.077361	0.065496	0.032203	0.021065	0.019734	0.018886	0.017918	0.016949	0.016828	0.016102	...	NaN	NaN	Na
COPX	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	Na
FXN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.016997	NaN	...	NaN	NaN	Na
DEM	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	0.001347	0.001235	0.0012
NOBL	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.015332	NaN	...	NaN	NaN	Na

246 rows × 8080 columns

In [ ]:
```
#convert dataset
final_result = new.T
final_result. columns=final_result. iloc[-1]
final_result = final_result[:-1]
#drop the column all nan
final_result=final_result.drop(columns=['EWJ','FXI','EWY','EWT'])
final_result
```

Out[44]:

ticker	IVV	IEFA	IEMG	IJR	IJH	IWF	IWM	IWD	EFA	ITOT	...	XSOE	VIOO	XYLD	FUTY	CDC	PTLC	COP
AAPL	0.064063	NaN	NaN	NaN	NaN	0.118347	NaN	NaN	NaN	0.053962	...	NaN	NaN	0.076513	NaN	NaN	0.077361	Na
MSFT	0.054308	NaN	NaN	NaN	NaN	0.100193	NaN	NaN	NaN	0.045755	...	NaN	NaN	0.065748	NaN	NaN	0.065496	Na
AMZN	0.02664	NaN	NaN	NaN	NaN	0.049034	NaN	NaN	NaN	0.022452	...	NaN	NaN	0.031254	NaN	NaN	0.032203	Na
GOOGL	0.017425	NaN	NaN	NaN	NaN	0.02806	NaN	0.004048	NaN	0.014682	...	NaN	NaN	0.02072	NaN	NaN	0.021065	Na
BRKB	0.016353	NaN	NaN	NaN	NaN	NaN	NaN	0.029058	NaN	0.013773	...	NaN	NaN	0.01991	NaN	NaN	0.019734	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
FIBRAMQMX	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na
BCHRBK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na
MERPM	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na
FROTOEIS	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na
BANPURBK	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	Na

8080 rows × 242 columns

In [ ]:
```
final_result.to_csv('final.csv',index = True)
```

Part III: Recommendation system

##Step 1: Define a function to calculate the similarity and return the Recommend ETFs

Import Dependecies:

In [ ]:
```
# Import Dependencies:
import scipy
from scipy import spatial
```

Function Definition:

In [ ]:
```
# function: similarity(df: DataFrame, input: DataFrame(1D vector), cutoff: numeric):-> dict
#   Helper do the cosine_similarity for the given index, return a sorted dictionary with the index as the key and the cosine_simila
#   similarity cutoff score must be within (0,1) for function to return a score dict, default value is 0.8. input must be a 1-D vec
#   dictionary will be in descending order.
def similarity(df, input, cutoff):
  candidates = {}
  for i in list(df.columns):
    if i != input:
      simlrty = 1 - scipy.spatial.distance.cosine(df[i].fillna(0), df[input].fillna(0)) #return the score this pkg fn works for 1D
      if (cutoff < 1 and cutoff > 0) and simlrty >= cutoff:
        candidates[i] = simlrty
      elif cutoff > 1 and cutoff < 0:
        print("please input a valid cutoff")
        return None
  #print(dict(sorted(candidates.items(), key=lambda item: item[1], reverse= True)))
  return dict(sorted(candidates.items(), key=lambda item: item[1], reverse= True))

# function: recommend(df: DataFrame, input: DataFrame(1D vector), cutoff: numeric, num: integer):-> dict
#   calling the helper--similarity to get the score and print the recommended ETFs based on the similarity. If the input is invalid
#   If the num is invalid(<0), default cutoff value is 0.8, it will print out the most similar ETF, if the num exceeds the return l
def recommend (df, input, cutoff=0.8, num=1):
  try:
    candidates = similarity(df, input, cutoff)
  except KeyError:
    print("Please input a valid ETF ticker! ")
    return None
  rec = {}
  for i in list(candidates)[0:max(min(num, len(candidates)),1)]:
    print ("ticker {}, similarity {} ".format(i, candidates[i]))
    rec[i] = candidates[i]
  return rec
```

##Step 2. Testing & check for expense ratio:

In [ ]:
```
# Tests:

print(recommend(final_result, 'QQQ',0.8,50))
# print(recommend(etfs, 'SPY', 0.9, 10))
# print(recommend(etfs, 'SPY', 0.9))
# print(recommend(etfs, 'VB', 4))
```

```
ticker QQQM, similarity 0.9999999824575585
ticker QYLD, similarity 0.9585477491146559
ticker ONEQ, similarity 0.9402668297849629
ticker SCHG, similarity 0.925952187876004
ticker IWF, similarity 0.9220408830298149
ticker IWY, similarity 0.9216983533797071
ticker VUG, similarity 0.9182151554581917
ticker MGK, similarity 0.9179983369027024
ticker VONG, similarity 0.9090901072228659
ticker IYW, similarity 0.8824953465631118
ticker ESGV, similarity 0.8815702587479439
ticker VONE, similarity 0.8765800230067662
ticker SCHK, similarity 0.8764315286710781
ticker SCHX, similarity 0.8764106424878356
ticker IGM, similarity 0.8720673285871995
ticker PBUS, similarity 0.8682894744358113
ticker OEF, similarity 0.8660374439801083
ticker SPLG, similarity 0.8653587280779103
ticker SCHB, similarity 0.8652650012902074
ticker SPTM, similarity 0.8651547407825559
ticker PTLC, similarity 0.8649403552330405
ticker SPY, similarity 0.8640977310310077
ticker IVV, similarity 0.8639231817387419
ticker ITOT, similarity 0.8633048120922531
ticker IWB, similarity 0.8622781565563779
ticker IWV, similarity 0.8620613992870476
ticker XYLD, similarity 0.8615416081022113
ticker GSLC, similarity 0.8551008557737322
ticker ESGU, similarity 0.8505049357820963
ticker URTH, similarity 0.8466932795221656
ticker ACWI, similarity 0.8447718883356036
ticker MGC, similarity 0.8434910769130806
ticker VV, similarity 0.841864450093054
ticker VTI, similarity 0.8412773441978566
ticker VOO, similarity 0.8412415787787351
ticker SPYG, similarity 0.8317721921086163
ticker IVW, similarity 0.8315041312318286
ticker IUSG, similarity 0.831185034909481
ticker XLK, similarity 0.828605616857576
ticker IOO, similarity 0.8274591890460972
ticker VGT, similarity 0.8263938040552348
```

```
ticker FTEC, similarity 0.8257953735568307
ticker IXN, similarity 0.822808121862125
ticker VT, similarity 0.8169207543315975
ticker VOOG, similarity 0.8140315551204393
{'QQQM': 0.9999999824575585, 'QYLD': 0.9585477491146559, 'ONEQ': 0.9402668297849629, 'SCHG': 0.925952187876004, 'IWF': 0.9220408
830298149, 'IWY': 0.9216983533797071, 'VUG': 0.9182151554581917, 'MGK': 0.9179983369027024, 'VONG': 0.9090901072228659, 'IYW':
0.8824953465631118, 'ESGV': 0.8815702587479439, 'VONE': 0.8765800230067662, 'SCHK': 0.8764315286710781, 'SCHX': 0.87641064248783
56, 'IGM': 0.8720673285871995, 'PBUS': 0.8682894744358113, 'OEF': 0.8660374439801083, 'SPLG': 0.8653587280779103, 'SCHB': 0.8652
650012902074, 'SPTM': 0.8651547407825559, 'PTLC': 0.8649403552330405, 'SPY': 0.8640977310310077, 'IVV': 0.8639231817387419, 'ITO
T': 0.8633048120922531, 'IWB': 0.8622781565563779, 'IWV': 0.8620613992870476, 'XYLD': 0.8615416081022113, 'GSLC': 0.855100855773
7322, 'ESGU': 0.8505049357820963, 'URTH': 0.8466932795221656, 'ACWI': 0.8447718883356036, 'MGC': 0.8434910769130806, 'VV': 0.841
864450093054, 'VTI': 0.8412773441978566, 'VOO': 0.8412415787787351, 'SPYG': 0.8317721921086163, 'IVW': 0.8315041312318286, 'IUS
G': 0.831185034909481, 'XLK': 0.828605616857576, 'IOO': 0.8274591890460972, 'VGT': 0.8263938040552348, 'FTEC': 0.825795373556830
7, 'IXN': 0.822808121862125, 'VT': 0.8169207543315975, 'VOOG': 0.8140315551204393}
```

In [ ]:
```
recomend = recommend(final_result, 'QQQ',0.8,50)
```

```
ticker QQQM, similarity 0.9999999824575585
ticker QYLD, similarity 0.9585477491146559
ticker ONEQ, similarity 0.9402668297849629
ticker SCHG, similarity 0.925952187876004
ticker IWF, similarity 0.9220408830298149
ticker IWY, similarity 0.9216983533797071
ticker VUG, similarity 0.9182151554581917
ticker MGK, similarity 0.9179983369027024
ticker VONG, similarity 0.9090901072228659
ticker IYW, similarity 0.8824953465631118
ticker ESGV, similarity 0.8815702587479439
ticker VONE, similarity 0.8765800230067662
ticker SCHK, similarity 0.8764315286710781
ticker SCHX, similarity 0.8764106424878356
ticker IGM, similarity 0.8720673285871995
ticker PBUS, similarity 0.8682894744358113
ticker OEF, similarity 0.8660374439801083
ticker SPLG, similarity 0.8653587280779103
ticker SCHB, similarity 0.8652650012902074
ticker SPTM, similarity 0.8651547407825559
ticker PTLC, similarity 0.8649403552330405
ticker SPY, similarity 0.8640977310310077
ticker IVV, similarity 0.8639231817387419
ticker ITOT, similarity 0.8633048120922531
ticker IWB, similarity 0.8622781565563779
ticker IWV, similarity 0.8620613992870476
ticker XYLD, similarity 0.8615416081022113
ticker GSLC, similarity 0.8551008557737322
ticker ESGU, similarity 0.8505049357820963
ticker URTH, similarity 0.8466932795221656
ticker ACWI, similarity 0.8447718883356036
ticker MGC, similarity 0.8434910769130806
ticker VV, similarity 0.841864450093054
ticker VTI, similarity 0.8412773441978566
ticker VOO, similarity 0.8412415787787351
ticker SPYG, similarity 0.8317721921086163
ticker IVW, similarity 0.8315041312318286
ticker IUSG, similarity 0.831185034909481
ticker XLK, similarity 0.828605616857576
ticker IOO, similarity 0.8274591890460972
ticker VGT, similarity 0.8263938040552348
```

```
ticker FTEC, similarity 0.8257953735568307
ticker IXN, similarity 0.822808121862125
ticker VT, similarity 0.8169207543315975
ticker VOOG, similarity 0.8140315551204393
```

In [ ]:
```
etf
```

Out[62]:

	Name	Ticker	30D Vol	Class Assets (MLN USD)	Fund Assets (MLN USD)	YTD Rtn	YTD Class Flow	(M USD)	Unnamed: 8	1M Flow	(M USD).1	Unnamed: 11	12M Yld	E
1	SPDR S&P 500 ETF Trust	SPY US	80.87M	379939.31	379939.31	+6.08%	4205.09	4205.092000	NaN	-3218.21	-3218.210000	NaN	+1.56%	
2	iShares Core S&P 500 ETF	IVV US	4.62M	306688.91	306688.91	+6.10%	-517.09	-517.087500	NaN	-1168.54	-1168.537000	NaN	+1.57%	
3	Vanguard Total Stock Market ETF	VTI US	4.18M	280070.66	280070.66	+6.67%	1659.60	1659.604000	NaN	2417.26	2417.262000	NaN	+1.56%	
4	Vanguard S&P 500 ETF	VOO US	4.24M	278653.5	278653.5	+6.13%	625.47	625.466700	NaN	272.50	272.496900	NaN	+1.59%	
5	Invesco QQQ Trust Series 1	QQQ US	48.99M	157563.78	157563.78	+11.26%	-4874.20	-4874.197000	NaN	-4245.36	-4245.363000	NaN	+0.72%	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
264	iShares MSCI Australia ETF	EWA US	2.40M	2047.66	2047.66	+12.19%	169.71	169.705600	NaN	169.71	169.705600	NaN	+4.71%	
265	Global X Copper Miners ETF	COPX US	514.65k	2025.91	2025.91	+16.93%	60.21	60.211600	NaN	74.40	74.403600	NaN	+2.69%	
266	First Trust Energy AlphaDEX Fund	FXN US	971.22k	2015.16	2015.16	+2.81%	-6.87	-6.870397	NaN	4.22	4.222215	NaN	+2.22%	
267	iShares Europe ETF	IEV US	470.02k	2011.65	2011.65	+9.41%	204.58	204.583700	NaN	204.58	204.583700	NaN	+2.80%	
268	WisdomTree Emerging Markets High Dividend Fund	DEM US	479.17k	2010.5	2010.5	+9.85%	89.96	89.955640	NaN	89.96	89.955640	NaN	+7.84%	

246 rows × 18 columns

In [ ]:
```
etf['Ticker'] = etf['Ticker'].map(lambda x:x[0:len(x)-3])
etf
```

Out[63]:

	Name	Ticker	30D Vol	Class Assets (MLN USD)	Fund Assets (MLN USD)	YTD Rtn	YTD Class Flow	(M USD)	Unnamed: 8	1M Flow	(M USD).1	Unnamed: 11	12M Yld	E
1	SPDR S&P 500 ETF Trust	SPY	80.87M	379939.31	379939.31	+6.08%	4205.09	4205.092000	NaN	-3218.21	-3218.210000	NaN	+1.56%	
2	iShares Core S&P 500 ETF	IVV	4.62M	306688.91	306688.91	+6.10%	-517.09	-517.087500	NaN	-1168.54	-1168.537000	NaN	+1.57%	
3	Vanguard Total Stock Market ETF	VTI	4.18M	280070.66	280070.66	+6.67%	1659.60	1659.604000	NaN	2417.26	2417.262000	NaN	+1.56%	
4	Vanguard S&P 500 ETF	VOO	4.24M	278653.5	278653.5	+6.13%	625.47	625.466700	NaN	272.50	272.496900	NaN	+1.59%	
5	Invesco QQQ Trust Series 1	QQQ	48.99M	157563.78	157563.78	+11.26%	-4874.20	-4874.197000	NaN	-4245.36	-4245.363000	NaN	+0.72%	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
264	iShares MSCI Australia ETF	EWA	2.40M	2047.66	2047.66	+12.19%	169.71	169.705600	NaN	169.71	169.705600	NaN	+4.71%	
265	Global X Copper Miners ETF	COPX	514.65k	2025.91	2025.91	+16.93%	60.21	60.211600	NaN	74.40	74.403600	NaN	+2.69%	
266	First Trust Energy AlphaDEX Fund	FXN	971.22k	2015.16	2015.16	+2.81%	-6.87	-6.870397	NaN	4.22	4.222215	NaN	+2.22%	
267	iShares Europe ETF	IEV	470.02k	2011.65	2011.65	+9.41%	204.58	204.583700	NaN	204.58	204.583700	NaN	+2.80%	
268	WisdomTree Emerging Markets High Dividend Fund	DEM	479.17k	2010.5	2010.5	+9.85%	89.96	89.955640	NaN	89.96	89.955640	NaN	+7.84%	

246 rows × 18 columns

In [ ]:
```
recomend
```

Out[96]:
{'QQQM': 0.9999999824575585,
'QYLD': 0.9585477491146559,
'ONEQ': 0.9402668297849629,
'SCHG': 0.925952187876004,
'IWF': 0.9220408830298149,
'IWY': 0.9216983533797071,
'VUG': 0.9182151554581917,
'MGK': 0.9179983369027024,
'VONG': 0.9090901072228659,
'IYW': 0.8824953465631118,
'ESGV': 0.8815702587479439,
'VONE': 0.8765800230067662,
'SCHK': 0.8764315286710781,
'SCHX': 0.8764106424878356,
'IGM': 0.8720673285871995,
'PBUS': 0.8682894744358113,
'OEF': 0.8660374439801083,
'SPLG': 0.8653587280779103,
'SCHB': 0.8652650012902074,
'SPTM': 0.8651547407825559,
'PTLC': 0.8649403552330405,
'SPY': 0.8640977310310077,
'IVV': 0.8639231817387419,
'ITOT': 0.8633048120922531,
'IWB': 0.8622781565563779,
'IWV': 0.8620613992870476,
'XYLD': 0.8615416081022113,
'GSLC': 0.8551008557737322,
'ESGU': 0.8505049357820963,
'URTH': 0.8466932795221656,
'ACWI': 0.8447718883356036,
'MGC': 0.8434910769130806,
'VV': 0.841864450093054,
'VTI': 0.8412773441978566,
'VOO': 0.8412415787787351,
'SPYG': 0.8317721921086163,
'IVW': 0.8315041312318286,
'IUSG': 0.831185034909481,
'XLK': 0.828605616857576,
'IOO': 0.8274591890460972,
'VGT': 0.8263938040552348,

```
 'FTEC': 0.8257953735568307,
 'IXN': 0.822808121862125,
 'VT': 0.8169207543315975,
 'VOOG': 0.8140315551204393}
```

In [ ]:
```
def expense_ratio(etf_ticker,result):
  """
     Passing target etf_ticker and it's result from recommendation system
     filter the expense ratio higher than original etf expense ratio
  """
  ticker = []
  similarity = []
  expense_ratio =[]
  base_line = etf.loc[etf['Ticker'] == etf_ticker, 'Expense Ratio'].item()
  for key,value in result.items():
    new_ratio = etf.loc[etf['Ticker'] == key,'Expense Ratio'].item()
    if new_ratio < base_line:
      ticker.append(key)
      similarity.append(str(value))
      expense_ratio.append(new_ratio)

  final_data={'etfs': ticker, 'similarity': similarity, 'expense_ratio': expense_ratio}
  df = pd.DataFrame(final_data)

  return df
```

In [ ]:
```
base_line = etf.loc[etf['Ticker'] == 'QQQ', 'Expense Ratio'].item()
base_line
```

Out[86]: '+0.20%'

In [ ]:
```
expense_ratio('QQQ',recomend)
```

Out[94]:

	etfs	similarity	expense_ratio
0	QQQM	0.9999999824575585	+0.15%
1	SCHG	0.925952187876004	+0.04%
2	IWF	0.9220408830298149	+0.18%
3	VUG	0.9182151554581917	+0.04%
4	MGK	0.9179983369027024	+0.07%
5	VONG	0.9090901072228659	+0.08%
6	ESGV	0.8815702587479439	+0.09%
7	VONE	0.8765800230067662	+0.08%
8	SCHK	0.8764315286710781	+0.05%
9	SCHX	0.8764106424878356	+0.03%
10	PBUS	0.8682894744358113	+0.04%
11	SPLG	0.8653587280779103	+0.03%
12	SCHB	0.8652650012902074	+0.03%
13	SPTM	0.8651547407825559	+0.03%
14	SPY	0.8640977310310077	+0.09%
15	IVV	0.8639231817387419	+0.03%
16	ITOT	0.8633048120922531	+0.03%
17	IWB	0.8622781565563779	+0.15%
18	GSLC	0.8551008557737322	+0.09%
19	ESGU	0.8505049357820963	+0.15%
20	MGC	0.8434910769130806	+0.07%
21	VV	0.841864450093054	+0.04%
22	VTI	0.8412773441978566	+0.03%
23	VOO	0.8412415787787351	+0.03%
24	SPYG	0.8317721921086163	+0.04%
25	IVW	0.8315041312318286	+0.18%

	etfs	similarity	expense_ratio
26	IUSG	0.831185034909481	+0.04%
27	XLK	0.828605616857576	+0.10%
28	VGT	0.8263938040552348	+0.10%
29	FTEC	0.8257953735568307	+0.08%
30	VT	0.8169207543315975	+0.07%
31	VOOG	0.8140315551204393	+0.10%

In [ ]: