



Portfolio for Great Lockdown

SWCON424-01 금융데이터분석

2015103604 수학과 한지운

2014102714 국제학과 엄세웅

2017103757 소프트웨어융합학과 정희재

Table of Contents

1. 주제: PORTFOLIO FOR RECESSION	2
2. ALL WEATHER PORTFOLIO	2
1) ALL WEATHER PORTFOLIO	2
2) RISK PARITY MODEL	2
3) ALL WEATHER BY RISK PARITY	4
3. 비교군	4
1) 60/40	4
2) GMV	4
4. 코드 설명	5
1) 실행 방법	5
2) 코드 DIRECTORY 구성	5
3) BACKTEST 구현	5
① <i>Asset</i>	6
② <i>Portfolio</i>	6
4) PORTFOLIO 구현	7
① 60/40	7
② GMV	7
③ <i>All Weather</i>	8
5. CASE STUDY	8
6. 결론	11
1) 한계점	11
2) 결론	11

1. 주제: Portfolio for Great Lockdown

IMF는 2020년을 Great Lockdown이라고 말하면서 Great Depression 이후 최악의 경기침체라고 말하였다. 2020년 3월, 코스피, S&P 500, 닷케이 지수가 20% 이하로 떨어지는 사태가 발생하였다. 이러한 경제 침체를 겪으며 세계적인 공황을 방어할 수 있는 포트폴리오를 구성해보고자 주제를 선정하였다.

2. All Weather Portfolio

1) All Weather Portfolio

All Weather Portfolio는 각 자산군의 상관관계가 경제 환경에 따라 달라지지만, 각각의 상황에서는 일관성을 갖는다는 사실을 기반으로 투자를 한다. 인플레이션에 대한 기대감이 높을 때는 주식과 채권의 상관관계가 높아지지만 경제성장에 기대감이 높을 때는 반대로 낮아지게 된다.

경제성장률과 인플레이션이 기대보다 높고 낮음에 따라 4개 상황에 동일한 리스크로 투자할 수 있게 포트폴리오를 구성한다. 4개의 상황에 유리한 자산은 아래 그림과 같다.

	Growth	Inflation
Rising	주식, 회사채 원자재, 이머징국가 채권	물가연동채권 원자재, 이머징국가 채권
Falling	국채 물가연동채권	주식 국채

경제를 예측하지 않고 각각의 상황에 들어갈 확률을 25%로 가정하여 자산을 배분한다. 자산군 내에서도 리스크를 헷지하기 위하여 다양한 자산에 투자할 수 있도록 구성하였다.

2) Risk Parity Model

기존 Mean-Variance Model은 공분산, 기대 수익률, 위험회피계수를 활용하여 모델을 구성한다. 이 때 기대수익률은 추정 에러가 높고 신뢰도가 낮다. M-V 모델은 이러한 기대수익률에 기반한 모델이기 때문에 불안정하다. 반면 Risk Parity Model은 위험기여도로 동일하도록 구성한다. 이때 공분산 행렬만 사용하기 때문에 M-V 모델보다 안정성을 갖는다.

Risk Parity Model은 Marginal Risk Contribution과 Risk Contribution을 활용하여 구성한다. Marginal Contribution은 특정 종목의 비중을 한 단위 늘렸을 때 증가하는 포트폴리오의 변동성(1)이다. Risk Contribution은 개별종목이 전체 포트폴리오의 위험에 기여하는 정도로 MRC에 가중치를 곱하는 값(2)로 나타낸다.

$$MRC = \frac{\partial \sigma_p}{\partial w_i} \text{ --- (1)}$$

$$RC = w_i \times \frac{\partial \sigma_p}{\partial w_i} \text{ --- (2)}$$

Python 으로 Risk Contribution 을 단순하게 계산하기 위해 한가지 계산 과정이 추가로 필요하다.

$$RC = w_i \times \frac{\partial \sigma_p}{\partial w_i}$$

$$\text{pf) } (\sqrt{f(x)})' = \frac{f'(x)}{2\sqrt{f(x)}}, \quad \sigma_p = \sqrt{w' \Sigma w}$$

$$MRC : \frac{\partial \sigma_p}{\partial w_i} = \frac{2 \Sigma w}{2 \sqrt{w' \Sigma w}}$$

$$RC : w_i \times \frac{\partial \sigma_p}{\partial w_i} = \frac{2 w' \Sigma w}{2 \sqrt{w' \Sigma w}} = \sigma_p$$

$$\text{Equal Risk Contribution} = \frac{\sigma_p}{n}$$

모델에서 동일한 Risk Contribution 을 주기 위해서 표준편차를 자산의 개수로 나눈 값을 활용한다. Risk Parity Model 는 최소화 모델로 구축할 수 있다.

$$\begin{aligned} \text{Minimize } \sum \sum (w_i MRC_i - w_j MRC_j)^2 & \qquad \text{Minimize } \sum [w_i - \frac{\sigma(w)^2}{(\Sigma w)_i N}]^2 \\ \text{s.t. } \sum w_i = 1 & \qquad \text{s.t. } \sum w_i = 1 \\ w_i > 0 \text{ for } \forall i & \qquad w_i > 0 \text{ for } \forall i \end{aligned}$$

Risk Parity Model 은 여러가지 최소화 모델로 구축할 수 있다. 좌측의 모델의 경우 i 번째 자산의 RC 가 j 번째 자산의 RC 와 같을 때 minimize 가 되는 형식으로 모든 자산의 RC 가 동일 해진다. 우측의 모델의 경우 $w_i - \frac{\sigma(w)^2}{(\Sigma w)_i N} = 0$ 일 때 최소화된다. 이때 i 번째 자산의 분산은 $w_i (\Sigma w)_i$ 로 $w_i (\Sigma w)_i = \frac{\sigma(w)^2}{N}$ 일 때, 즉 각 자산의 분산이 전체 분산을 자산으로 나눈 값으로 동일 할 때 최소화된다.

이 모델을 활용함으로써 특정 경우에 따라 자산의 RC 가 동일하게 나오지 않을 상황이 생길 수 있다. 이 때 좌측모델과의 결과값의 차이가 발생할 수 있다. 그럼에도 이 프로젝트에서는 코딩에 용이하여 따라 좌측 모델을 활용할 것이다.

3) All Weather by Risk Parity

Risk Parity Model 을 All Weather Portfolio 에 적용하기 위해서 포트폴리오를 2 번 구축해야한다. 첫째로 그림 1 에서 나뉜 대로 총 4 개의 포트폴리오를 구성한다. 이 때 포트폴리오는 동일가중치를 활용한다. 이 후 Risk Parity 를 활용하기 때문에 4 개의 포트폴리오는 동일가중치를 활용해도 괜찮다고 가정한다. 4 개의 포트폴리오를 하나의 자산으로 보고 Risk Parity 모델을 활용하여 가중치를 구한다.

3. 비교군

All Weather Portfolio 와의 기대 수익률, Drawdown 을 비교하기 위한 벤치마크로 두 가지 포트폴리오를 사용하였다. GMV Portfolio 는 수익률을 일부 포기하는 대신 압도적인 안정성을 보장하는 모델이며, 60/40 Portfolio 는 안정성을 기반으로 꾸준히 높은 수익률을 보여주는 모델이다. 두 모델을 벤치마킹하여 All Weather Portfolio 의 상대적인 수익률과 안정성을 비교하고 포트폴리오의 성능을 평가하고자 하였다.

1) 60/40

먼저 첫 번째 벤치마크로 GMV Portfolio(최소 분산 포트폴리오)를 사용하였다. GMV Portfolio 는 포트폴리오의 분산을 최소화함으로써 분산을 통해서 상쇄되지 않는 리스크를 최대한 낮춰 포트폴리오의 안정성을 최대화된다는 장점을 가지고 있다. 수익률은 낮다는 단점을 가지고 있지만 포트폴리오의 리스크가 가장 낮은 모델이기 때문에 코로나 대유행으로 변동성이 높은 주식 시장에서도 안정적인 모습을 보여줄 것이라고 생각했다. 따라서 All Weather Portfolio 와의 drawdown 차이 대비 수익률을 비교하고자 벤치마크 모델로 선정하였다.

2) GMV

두 번째 벤치마크로는 60/40 Portfolio 를 사용하였다. 60/40 Portfolio 는 주식의 비중을 60%, 채권의 비중을 40%를 두고 투자하는 conventional portfolio 이다. 해당 포트폴리오는 주식 시장의 변동성을 채권을 통해 리스크를 완충 및 분산시키면서도 높은 수익률을 낸다는 장점을 가지고 있다. 특히, 1983 년부터 2010 년까지 S&P 500 지수는 연간 11.2%, 채권 인덱스 지수는 연간 7% 상승하여 평균 연간 9.9%의 높은 수익률을 보여주었다. 주식에 60%의 높은 비중을 두고 있음에도 불구하고 물가연동채 TIPS 와 금을 통해 포트폴리오에 안정성을 추가하여 과거 2007 년의 주식 폭락에도 포트폴리오 수익률을 방어할 수 있었다. 이렇게 높은 수익률과 안정성을 가지고 있는 60/40 Portfolio 를 벤치마킹하여 2020 년 코로나 대유행 기간 동안의 All Weather Portfolio 와의 연간 수익률과 drawdown 을 비교하고자 하였다.

4. 코드 설명

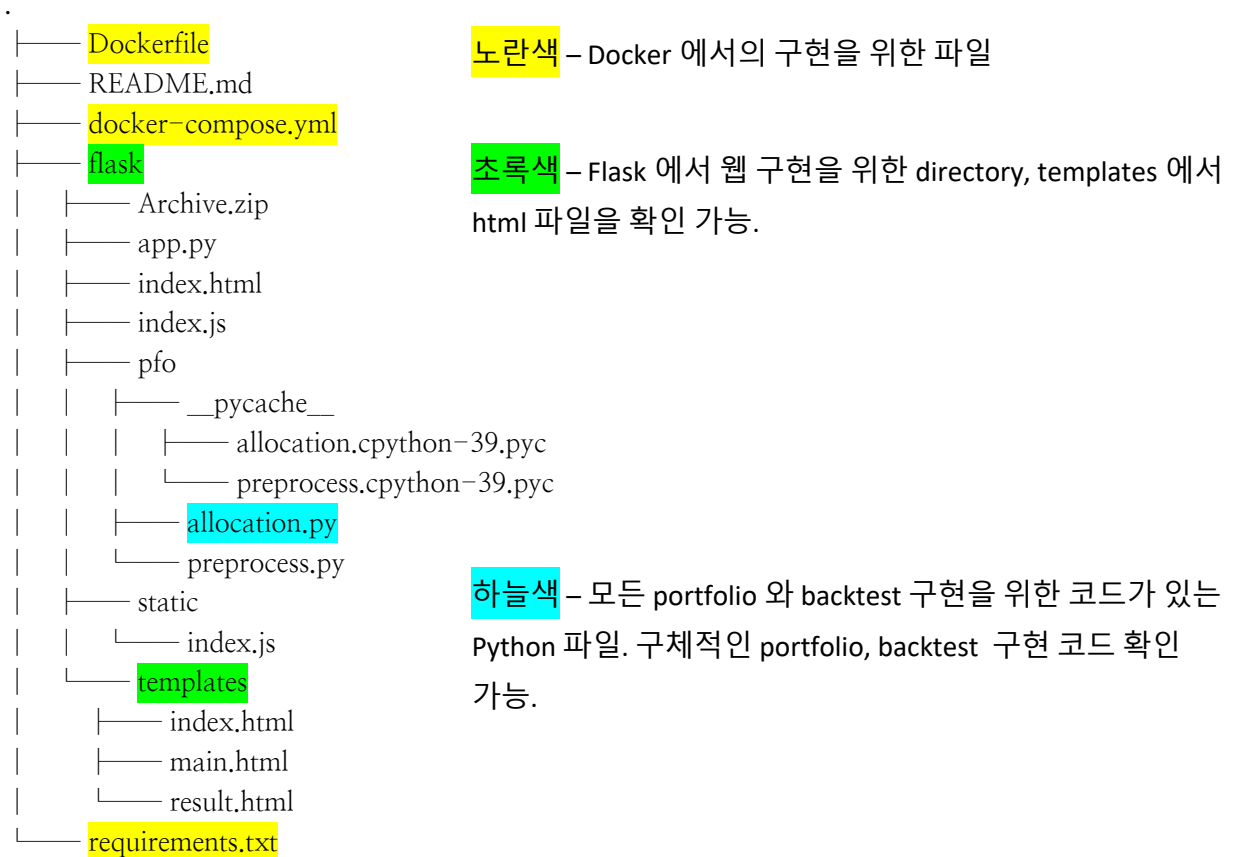
Flask 와 Docker 를 사용하여 input 에 따른 결과를 시각화해주는 웹 사이트를 구현했다.

1) 실행 방법

* 해당 코드를 편리하게 실행하기 위해서는 Docker 가 필요하며, 웹사이트는 Chrome 에 최적화 되어있으니 Chrome 에서 전체화면이 가장 용이하다. *

1. https://github.com/lydiahjchung/2020_FDA 에서 git clone, 또는 첨부파일 압축 풀기
2. 다운로드 받은 디렉토리에서 아래 명령어 (총 2 개) 실행
 - > docker-compose build
 - > docker-compose up
3. Chrome 에서 0.0.0.0:5000 접속

2) 코드 Directory 구성



3) Backtest 구현

크게 두 개의 클래스로 구현이 되어 있고, 세부적으로 나누어진 Portfolio 의 개별 클래스들은 이 두 클래스를 바탕으로 생성했다. 데이터 시각화는 전부 flask 의 app.py 에 구현되어 있다.

Plotly 를 사용하여 구현하였고 backtest 의 데이터를 가지고 단순히 시각화해주기 때문에 설명은 제외한다.

① Asset

ticker, start date, end date 를 입력 받아 데이터를 해당 ticker 의 해당 기간 동안의 데이터를 저장해두는 객체이다. Adjusted Close 데이터를 가져와 percentage change 를 진행해 준 데이터를 저장해둔다. 이후 Portfolio 에서 이 asset 클래스를 활용하여 포트폴리오 구성 시 데이터 활용이 가능하며 모든 데이터는 Yahoo Finance 데이터를 활용했다.

② Portfolio

Portfolio 구성으로 backtest 및 시각화 구현을 위해 데이터 처리를 해주는 클래스이다. 아래의 4 가지 함수들이 주요한 계산을 구현하고, 나머지는 모두 getter 함수이다.

○ backtest()

사용자에게 받은 initial balance, rebalancing 주기, look back 주기를 바탕으로 실행된다. 처음 backtest 를 시작하면 사용자에게 받은 initial balance 를 초기 weight 와 asset 으로 맞추어 나누어줘야 한다. 60/40 의 경우는 고정 weight 이지만, GMV 와 All Weather 의 경우는 아니기에 backtest 시작과 동시에 start date 를 기준으로 사용자의 look back 기간만큼의 데이터를 통하여 초기 weight 를 계산해준다. 이 weight 들과 60/40 의 고정 웨이트를 통해 start date 의 balance 를 asset 별로 나누어준다.

이후 rebalancing 주기마다 look back 기간만큼의 데이터를 가지고 weight 를 재설정해준 값으로 balance 를 다시 나누어준다. rebalancing 이 아닌 기간 동안은 설정된 weight 와 asset 별 데이터를 가지고 balance 에 대한 데이터를 계속 쌓아준다.

end date 까지 동일한 과정이 이어지고, 이를 하나의 data frame 로 저장한다. Portfolio 에 해당하는 Total balance 값과 각 asset 별 balance 가 저장되므로 column 으로 Total 과 각 asset 들이 있다.

○ backtest_result() & balance_result()

backtest_result 함수는 backtest 함수 구현을 통해 저장한 backtest 결과의 데이터를 가지고 rate of return, cumulative return, CAGR, drawdown, MDD 의 값을 구하기 위하여 사용된다. balance_result 는 이 backtest_result 함수 내에서 호출된다. Backtest 의 결과로 각 column 별로 balance 가 날짜에 따라 어떻게 변하는 지가 들어있고, 현재 backtest 의 결과에 대한 지표 값들을 계산하고 싶은 것이기에 Total balance 열에 대한 데이터만 balance_result 함수를 통하여 각 지표들의 값을 구해준다.

balance_result 함수에서는 날짜별로 저장된 변화되는 balance 값을 통하여 rate of return, cumulative return, CAGR, drawdown, MDD, standard deviation, sharpe ratio 값을 구해준다. backtest_result 에서는 이 값 중 4 가지만 저장하지만, 향후 다른 값들은 summary 를 통해 사용되도록 구현했다.

- summarize()

Portfolio 의 전체 summary 를 출력해준다. 이를 통해 initial balance, portfolio 의 final balance, CAGR, MDD, sharpe ratio 의 값을 출력해준다.

- periodic_result(mode)

Annual return, monthly return 의 값을 향후 시각화해주기 위하여 구현한 함수이다. Mode 를 통하여 annual, monthly return 중 한 가지를 골라 데이터를 출력할 수 있다. backtest 를 통해 구한 total balance 의 데이터를 가지고 각 year / month 별로 의 balance 데이터를 통해 하나의 data frame 에 저장해준다.

4) Portfolio 구현

모든 포트폴리오는 3 번에서의 두 클래스를 바탕으로 구현이 되기에, 다른 점은 rebalancing 주기에 맞춰 weight 를 다시 구하는 과정에만 차이가 존재한다.

① 60/40

60/40 포트폴리오의 경우 주식 60%, 채권 40% 의 고정 비율로 backtest 가 진행된다. 따라서, rebalancing 또한 고정된 weight 로 진행되어 복잡한 weight 계산 과정이 따로 없다. 하지만 input 을 받는 check box 들에서 볼 수 있듯이, 총 6 가지의 자산군으로 나누어 입력을 받기에, 60/40 의 경우에는 선택된 input 중 Nominal Bond 와 Global Equities 에서 선택된 자산들만을 고려한다. 한 자산군당 여러 input 을 받을 수 있기에, 각 bond 와 equities 를 40%와 60%의 비율로 가져가되, 각 자산군 안에서의 개수에 맞게 그 비율을 조정해준다.

예를 들어, Nominal Bond 에서 총 2 개를 선택하고, Global Equities 에서 총 3 개를 선택한 경우, bond 는 40%, equities 는 60%의 비율을 유지해야하기에 Nominal Bond 내의 각 자산은 20%씩의 비율을 가지게 되고, Global Equities 내의 자산 또한 각 20%씩의 비율을 가지고 portfolio 의 계산이 진행된다.

② GMV

먼저 Asset 클래스를 상속받는 GMV_Asset 클래스에 주식 리스트와 시작 일, 마지막 일을 input 으로 사용하여 주어진 포트폴리오의 데이터 프레임을 얻는다. 이후 각 주식의 adjusted close price 에 percentage change 를 하고 numpy 의 mean()과 cov() 모듈을

사용하여 포트폴리오의 평균 수익률과 공분산 데이터 프레임을 만든다. 이렇게 얻은 평균 수익률과 공분산 데이터프레임을 GMVPortfolio 클래스에 넣고 각각을 연 단위 수치로 변환해 준 다음, scipy 의 optimize.minimize 모듈을 통해 포트폴리오의 최소 분산을 만족하는 weight 값을 얻는다. 이 과정을 rebalancing 기간마다 실행하여 dynamic weight allocation 을 구현했다.

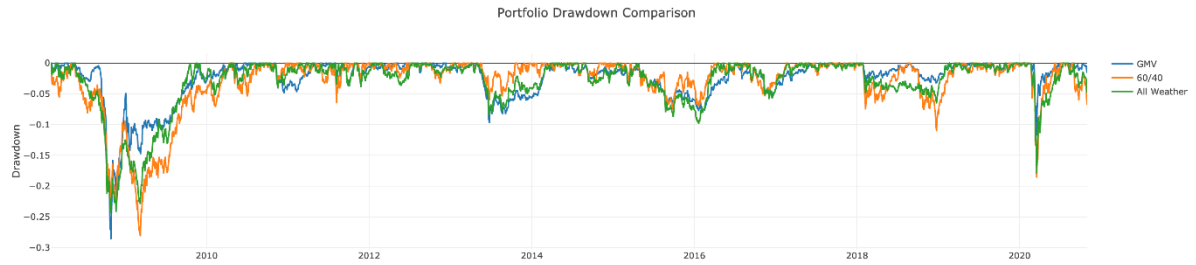
③ All Weather

GMV 의 경우 주어진 포트폴리오를 주식, 채권, 회사채, 원자재 등으로 구분하지 않고 하나로 묶어서 관리하지만 All Weather Portfolio 의 경우 각각을 구분해서 계산하기 때문에 각 카테고리를 dictionary 형태로 구분했다. Asset 클래스에서 상속받는 AWF_Asset 클래스에 주식 리스트와 시작 일, 마지막 일을 input 으로 사용하여 각 자산의 수익률을 나타내는 dataframe 을 dictionary 형태로 저장하여 반환한다. 이후 AllWeatherPortfolio 클래스를 통해 각 카테고리를 growth, rising, emerging, falling 에 따라 4 개로 구분한다. 이렇게 구분한 데이터를 RiskPartiy 함수에 넣고 scipy 의 optimize.minimize 를 통해 카테고리 4 개에 대한 weight 값을 얻는다. 마지막으로 4 개의 weight 값을 각 카테고리 내의 ticker 들에 맞게 한번 더 쪼개어준다. 예를 들어 주식의 개수가 총 4 개이고 주식에 대한 weight 이 0.2 라면 각 주식에 대해 0.05 의 weight 을 할당한다. 이 과정을 GMV 와 마찬가지로 rebalancing 기간마다 실행하여 dynamic weight allocation 을 구현했다.

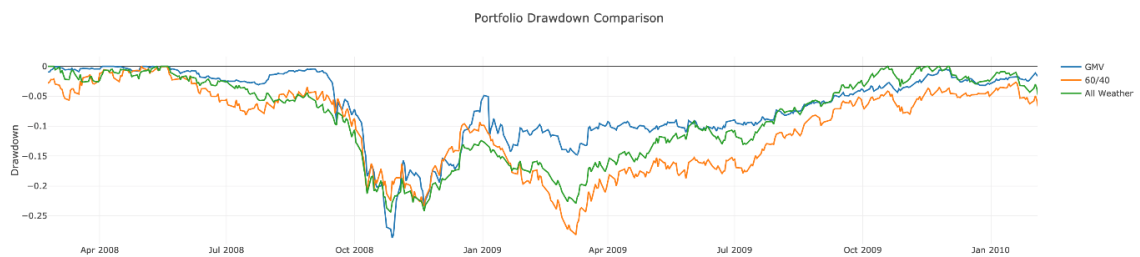
5. Case Study

Case Study 를 진행하기 위해 자산을 임의로 지정하였다. 기간은 2008 년 2 월 1 일부터 2020 년 10 월 30 일까지로 설정하였고 Rebalance 기간은 6 개월 Rebalance 때 lookback 기간은 6 개월로 설정하였다.

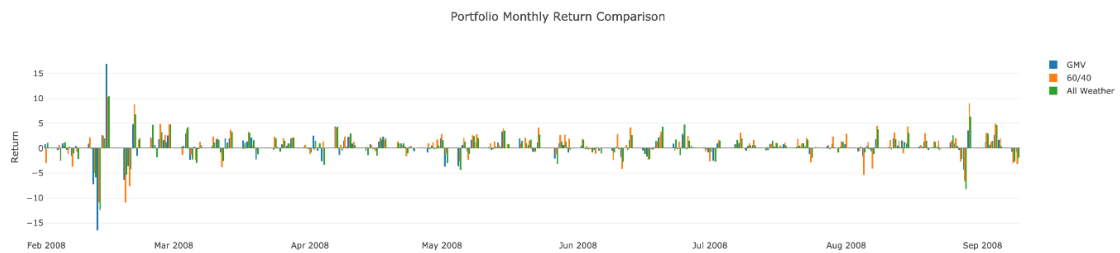
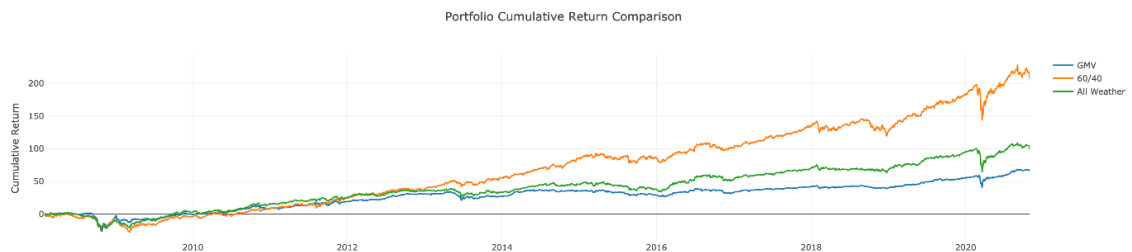
Nominal Bond	iShares 1-3 Year Treasury Bond ETF
Inflation-linked Bond	iShares TIPS Bond ETF
US Corporate Debt Spreads	iShares iBoxx \$ Investment Grade Corporate Bond ETF
Global Equities	Vanguard Total Stock Market Index Fund ETF Shares
EMD Spreads	iShares MSCI Brazil ETF
Commodities	Invesco DB Commodity Index Tracking Fund



Drawdown 은 위와 같이 나왔다. 예상했던 것처럼 2008~2009 년 2020 년초에 큰 Drawdown 이 있었다. 전반적으로 GMV 의 Drawdown 이 안정적이었고 60/40 의 Drawdown 이 가장 불안정 했다.



Recession 이었던 2008 년을 자세히 살펴보면 All Weather Portfolio 가 가장 안정적인 모습을 보였고 GMV 와 60/40 이 25%가 넘어가는 drawdown 을 보였다.



누적수익률의 경우 60/40, All Weather Portfolio, GMV 의 순으로 좋았다. All Weather 의 경우 주식의 비중이 20% 아래로 하락하여 60/40 에 비해 수익률이 낮음을 알 수 있었다.

Fama-French 5 Factor Model 과 회귀분석을 진행한 결과는 다음과 같다.

OLS Regression Results					OLS Regression Results				
Dep. Variable:	Total Rate of Return	R-squared:	0.306		Dep. Variable:	Total Rate of Return	R-squared:	0.980	
Model:	OLS	Adj. R-squared:	0.305		Model:	OLS	Adj. R-squared:	0.980	
Method:	Least Squares	F-statistic:	235.8		Method:	Least Squares	F-statistic:	2.578e+04	
Date:	Mon, 14 Dec 2020	Prob (F-statistic):	4.68e-250		Date:	Mon, 14 Dec 2020	Prob (F-statistic):	0.00	
Time:	22:24:30	Log-Likelihood:	-1612.3		Time:	22:24:23	Log-Likelihood:	2612.3	
No. Observations:	3211	AIC:	3239.		No. Observations:	3211	AIC:	-5211.	
Df Residuals:	3204	BIC:	3281.		Df Residuals:	3204	BIC:	-5168.	
Df Model:	6				Df Model:	6			
Covariance Type: nonrobust					Covariance Type: nonrobust				
	coef	std err	t	P> t [0.025 0.975]		coef	std err	t	P> t [0.025 0.975]
const	0.0132	0.009	1.536	0.125 -0.004 0.030	const	0.0073	0.002	3.142	0.002 0.003 0.012
Mkt-RF	0.2073	0.006	34.330	0.000 0.195 0.219	Mkt-RF	0.5544	0.002	342.254	0.000 0.551 0.558
SMB	-0.0132	0.012	-1.110	0.267 -0.036 0.010	SMB	0.0051	0.003	1.590	0.112 -0.001 0.011
HML	-0.0238	0.010	-2.286	0.022 -0.044 -0.003	HML	0.0046	0.003	1.632	0.103 -0.001 0.010
RMW	0.0518	0.019	2.719	0.007 0.014 0.089	RMW	0.0026	0.005	0.513	0.608 -0.007 0.013
CMA	-2.17e-05	0.024	-0.001	0.999 -0.047 0.047	CMA	0.0011	0.006	0.171	0.865 -0.011 0.014
RF	-1.4336	2.243	-0.639	0.523 -5.832 2.965	RF	-0.1622	0.602	-0.269	0.788 -1.342 1.018
Omnibus:	511.891	Durbin-Watson:	1.864		Omnibus:	681.449	Durbin-Watson:	1.907	
Prob(Omnibus):	0.000	Jarque-Bera (JB):	8305.751		Prob(Omnibus):	0.000	Jarque-Bera (JB):	9763.546	
Skew:	-0.185	Prob(JB):	0.00		Skew:	-0.597	Prob(JB):	0.00	
Kurtosis:	10.870	Cond. No.	444.		Kurtosis:	11.459	Cond. No.	444.	

< 60/40 >

< All Weather Portfolio >

OLS Regression Results				
Dep. Variable:	Total Rate of Return	R-squared:	0.080	
Model:	OLS	Adj. R-squared:	0.078	
Method:	Least Squares	F-statistic:	46.15	
Date:	Mon, 14 Dec 2020	Prob (F-statistic):	1.77e-54	
Time:	22:24:27	Log-Likelihood:	-1547.1	
No. Observations:	3211	AIC:	3108.	
Df Residuals:	3204	BIC:	3151.	
Df Model:	6			
Covariance Type: nonrobust				
	coef	std err	t	P> t [0.025 0.975]
const	0.0205	0.008	2.425	0.015 0.004 0.037
Mkt-RF	0.0822	0.006	13.890	0.000 0.071 0.094
SMB	-0.0034	0.012	-0.296	0.768 -0.026 0.019
HML	-0.0157	0.010	-1.543	0.123 -0.036 0.004
RMW	-0.0443	0.019	-2.371	0.018 -0.081 -0.008
CMA	-0.0240	0.023	-1.032	0.302 -0.070 0.022
RF	-4.0315	2.198	-1.834	0.067 -8.342 0.279
Omnibus:	985.826	Durbin-Watson:	1.615	
Prob(Omnibus):	0.000	Jarque-Bera (JB):	92491.868	
Skew:	-0.451	Prob(JB):	0.00	
Kurtosis:	29.277	Cond. No.	444.	

< GMV Portfolio >

All Weather Portfolio 와 GMV 둘은 alpha 값이 0 보다 크고 p 값이 유의수준보다 작기 때문에 유의미하다. 이에 따라 두 모델은 Factor Model 보다 수익률이 좋다고 할 수 있다.

세 포트폴리오모델의 정량적 값들을 정리하면 다음과 같다.

	60/40	GMV	AWP
Final Balance	2473.46	1579.73	1837.10
CAGR	7.36%	3.65%	4.88%
MDD	-33.73%	-29.06%	-21.64%
Sharpe Ratio	0.19	0.13	0.16
VaR	-1.12	-0.49	-0.68
CVaR	-1.90	-0.98	-1.10

6. 결론

1) 한계점

Risk Parity 모델 구축에 있어서, 최소화할 때 특정 경우에 따라 자산의 RC가 동일하게 나오지 않을 상황이 생길 수 있다. 이 때 다른 방식으로 최소화 결과값을 구할 경우와 차이가 발생할 수 있다.

실제로 Bridgewater사에서 All Weather Portfolio를 활용할 때 leverage를 적극 활용한다. 프로젝트 간 구성된 모델은 leverage가 활용되지 않아서 60/40 모델보다 저조한 수익률을 나타내고 있다. 또한 ETF를 활용할 때 지출되는 수수료, rebalancing 때 발생하는 거래수수료, 채권에서 나오는 coupon, 주식에서 나오는 배당 등 추가적으로 발생하는 소득과 지출에 대한 내용을 적용하지 않아 실제로 포트폴리오를 구성했을 때와 오차가 발생할 것이다.

2) 결론

세계적인 대공황을 잘 방어하는 포트폴리오를 구성하기 위하여 All Weather Portfolio를 구성하였다. 60/40 Portfolio 모델과 GMV Portfolio 모델과 비교해 보았을 때 수익률과 Sharpe Ratio는 두 모델의 중간의 성격을 띄는 모습을 보였다. 하지만 MDD는 세 모델 중 가장 낮은 모습을 보여주고 있다. MDD는 하락세에서 얼마나 방어할 수 있는지 보여주는 지표이기 때문에 목적에 잘 맞는 포트폴리오를 구성했다고 말할 수 있다.