

ECS781P CLOUD COMPUTING

Mini Project

Portfolio Analysis

Portfolio Diversification

Portfolio diversification is the practice of spreading your investments around so you're your exposure for one type of asset is limited (Fidelity). The key to portfolio diversification is balancing comfort and risk against a time horizon. Therefore, investors need to take into consideration the weighting of each investment in a portfolio given a fixed amount of capital. It is important to not 'put all your eggs in one basket' and spread risk across different stocks, whilst recognising the risk associated with each stock. Diversification can be across different industries, companies, or even asset classes. For this portfolio, the asset class is fixed as a stock, but the companies can be across many industries. Further, the selection is limited to three stocks. Each stock is represented by a ticker which symbolises the selected company.

The timeseries data is backtested and outputs key performance metrics for a portfolio. Backtesting is the process of seeing how a portfolio would have performed retrospectively. The results are zipped into a dictionary with a backtest key (the following metrics) and the results which are outputted by Empyrial. The metrics are as follows:

- Annual Return: the return on the investment expressed as a time weighted annual percentage.
- Cumulative Return: the aggregate return on the investment given the time period.
- Annual Volatility: measure of variance of returns weighted as a annual percentage
- Winning day ratio: a ratio between the total number of winning trades and losing trades (winners/losers) expressed as a percentage
- Sharpe ratio: average return earned in excess of the risk-free rate per unit of volatility or total risk
- Calmar ratio: function of the portfolios average compounded annual rate of return against its maximum drawdown (explained below)
- Information ratio: measurement of portfolio returns above the returns of a benchmark (e.g. S&P 500) to the volatility of those returns
- Stability
- Max Drawdown: the maximum loss from a peak to a trough before a new peak is attained.
- Sortino ratio: risk adjusted return of an investment asset, portfolio, or strategy. Variation of the Sharpe ratio that differentiates harmful volatility from total overall volatility by using the asset's standard deviation of negative portfolio returns.
- Skew: the relative number positive returns versus negative returns. A negative skew implies numerous negative returns and few large positive returns. A positive skew implies the opposite.
- Kurtosis: the degree to which portfolio returns appear in the tails of distribution. A normal distribution has a kurtosis of 3.
- Tail Ratio

2 . Search Portfolio

Ticker 1:

Ticker 2:

* Please select a different from ticket 1

Ticker 3:

* Please select a different from ticket 1 and 2

3 . Portfolio selection for diversification

Do not diversify

Medium diversify

Optimize

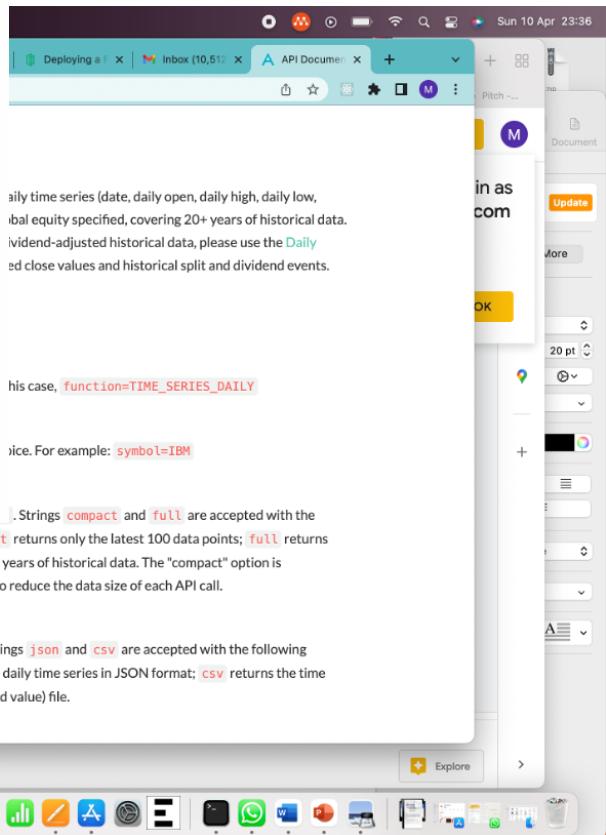
Manage risk

Enter start time for portfolio:

Get Portfolio

The screenshot shows a Mac desktop with a browser window open to a portfolio search page. The sidebar on the left contains links like 'Pitch', 'File', 'Edit', 'External REST API', 'Portfolio Diversification', 'Empirical Module', 'Login Authentication', 'Role-Based Authorization', and 'Security'. The main content area has fields for Ticker 1 (AAPL), Ticker 2 (MSFT), and Ticker 3 (FB). Below these is a section titled 'Portfolio selection for diversification' with four radio button options: 'Do not diversify', 'Medium diversify', 'Optimize' (which is selected), and 'Manage risk'. There's also a date input field for the start time (10/03/2021) and a large blue 'Get Portfolio' button.

- Common sense ratio: the tail ratio multiplied by the pain ratio
- Daily value at risk: calculates the maximum loss expected (or worst case scenario) on an investment, over a given time period and given a specified degree of confidence.
- Alpha: the excess returns earned on investment above the benchmark return



```
{
  "Meta Data": {
    "1. Information": "Daily Prices (open, high, low, close) and Volumes",
    "2. Symbol": "IBM",
    "3. Last Refreshed": "2022-04-08",
    "4. Output Size": "Compact",
    "5. Time Zone": "US/Eastern"
  },
  "Time Series (Daily)": {
    "2022-04-08": {
      "1. open": "128.0100",
      "2. high": "128.7800",
      "3. low": "127.2700",
      "4. close": "127.7306",
      "5. volume": "3143309"
    },
    "2022-04-07": {
      "1. open": "128.8700",
      "2. high": "129.2499",
      "3. low": "126.7300",
      "4. close": "128.5500",
      "5. volume": "3538317"
    },
    "2022-04-06": {
      "1. open": "127.7800",
      "2. high": "129.5150",
      "3. low": "127.2500",
      "4. close": "129.3900",
      "5. volume": "3958057"
    },
    "2022-04-05": {
      "1. open": "129.8500",
      "2. high": "131.2300",
      "3. low": "128.6600",
      "4. close": "128.8900",
      "5. volume": "3102238"
    },
    "2022-04-04": {
      "1. open": "130.2600",
      "2. high": "130.4500",
      "3. low": "128.9000",
      "4. close": "130.2700",
      "5. volume": "2811100"
    },
    "2022-04-01": {
      "1. open": "129.6600",
      "2. high": "130.2700",
      "3. low": "128.0600",
      "4. close": "130.1500",
      "5. volume": "4012373"
    },
    "2022-03-31": {
      "1. open": "130.7200",
      "2. high": "131.8800",
      "3. low": "130.0000",
      "4. close": "130.0200",
      "5. volume": "4274029"
    }
  }
}
```

- Beta: measure of volatility against the market as a whole.

The options for diversification (in terms of weighting) are ‘do not diversify’, ‘medium diversify’, ‘optimise’, and ‘manage risk’. These are the relative weightings for each selected stock. Each selection places a different weighting on each stock according to the appropriate label. Optimise may do so with custom weights based on mean-variance, minimum-variance, global efficient frontier, or hierarchical risk parity (these are all examples of measuring risk). The button is run by importing the Empyrial module.

Empyrial Module

Empyrial is a Python-based open-source quantitative investment library for financial institutions and retail investors (<https://empyrial.gitbook.io/empyrial/>). The library aims to become an all-in-one platform for portfolio management, analysis, and optimisation. Quantstats is another module used in the requirements.txt and Empyrial is mainly a wrapper of financial analysis libraries like Quantstats.

Empyrial’s “Engine” is imported to output the KPIs based on a start date, the portfolio tickers, their respective weights and a benchmark (default is SPY). The output is formatted in the same way as in the app. These figures as discussed give good indications about the financial health, risk, and returns of the backtested, diversified portfolio.

Passlib Library

Passlib is a python library used for password hashing (<https://pypi.org/project/passlib/>). In this project, the hash function SHA-256 was implemented to provide security for user passwords stored in the database.

Flask WTForms library

Flask WTForms is a python-based library involved in form rendering and validation in web development (<https://pypi.org/project/Flask-WTF/>). In this project, it was used to render the new user registration form.

CRUD Operation:

GET:

When user furnishes 3 stock symbols/ ticker symbols with diversification techniques,(user can also choose not to diversify the weights of stocks), date specification user receives a response as tabular KPI results (json converted to html) for the stocks portfolio analysis.

Sample external API end point call :

Sample response

The screenshot shows a Safari browser window with the address bar displaying "127.0.0.1:5000/portfolio/". The main content area contains a table with the following data:

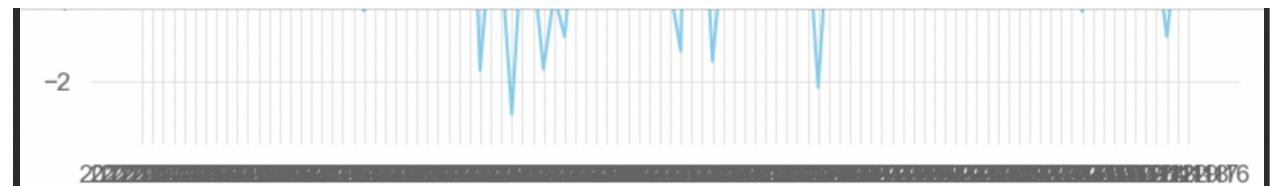
Ticker1	AAPL
Ticker2	MSFT
Ticker3	FB
Diversification_Method	Optimize
Start_Date	2021-03-10
Annual_return	38.6%
Cumulative_return	42.61%
Annual_volatility	24.35 %
Winning_day_ratio	52.92%
Sharpe_ratio	1.46
Calmar_ratio	2.28
Information_ratio	0.0
Stability	0.84
Max_Drawdown	-17.14 %
Sortino_ratio	2.27
Skew	0.2
Kurtosis	0.77
Tail_Ratio	1.03
Common_sense_ratio	1.31
Daily_value_at_risk	-2.0 %
Alpha	0.18
Beta	1.27

After receiving the data from external API it was processed (normalized, cleaned) to calculate daily returns, 30 day EWM for the plots.

Matplotlib is used to create stock graphs converted to IO bytes and presented as png on the HTML container

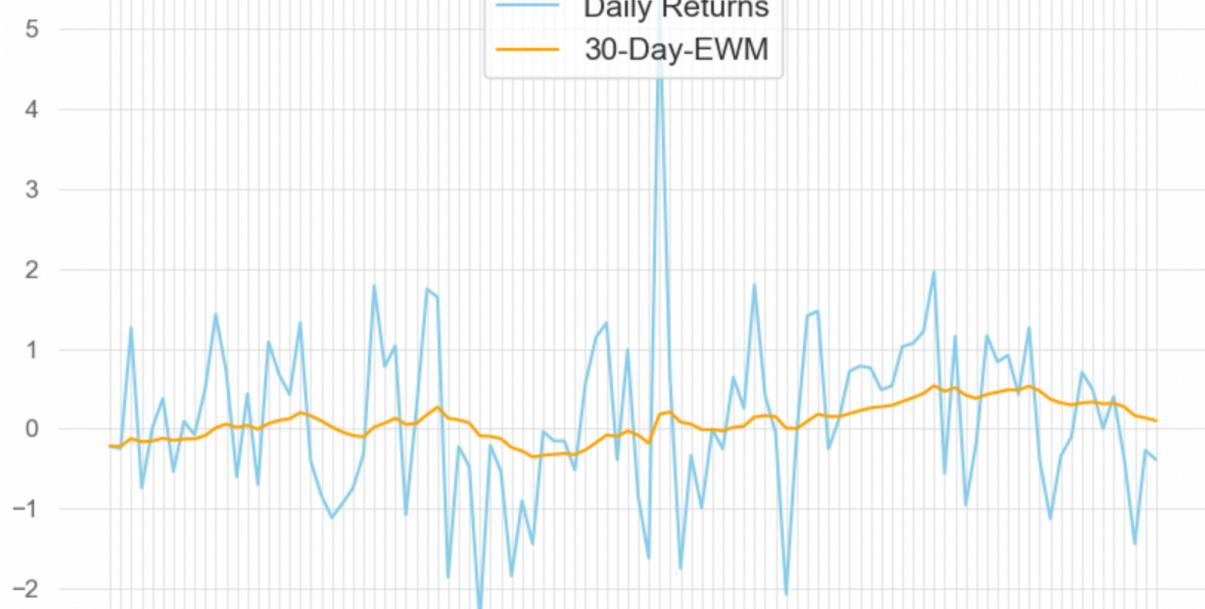
Creation of user class for ORM mapping to be done later by sqlalchemy

POST:



MSFT

Daily Returns
30-Day-EWM



Pr
Ac
Ba

Delete

Y

N

N

FB

Daily Returns

```

    flash("invalid credentials, please use valid username and password!!")
    return render_template("login1.html")

    flash("invalid credentials, please use valid username and password!!")
    return render_template("login1.html")

@app.route('/portfolio/', methods=['post'])
def post():
    global start_date
    global api_key
    global portfolio
    user_id = int(session["uid"])
    tk1 = request.form.get('ticket1')
    tk2 = request.form.get('ticket2')
    tk3 = request.form.get('ticket3')
    diversify = request.form.get('portfolio')
    start_date = request.form.get('date')
    portfolio_name = request.form.get('portfolio_name')

    # Getting portfolio names of currently logged in user
    get_portfolio_name = request.form.get('saved_portfolio_names')

```

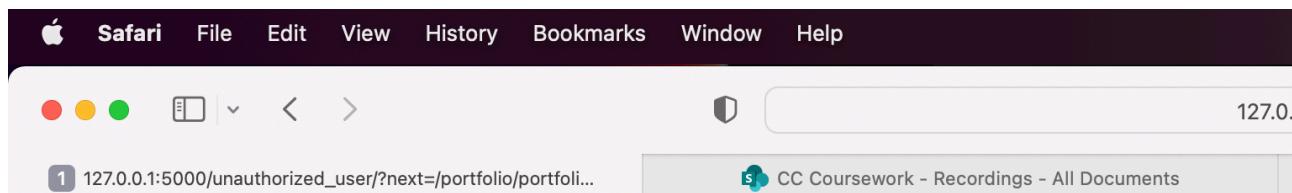
When user furnishes 3 stock symbols/ ticker symbols with date, diversification techniques (user can also choose not to diversify the weights of stocks), date specification a date specification user receives a response whether or not the portfolio details were save successfully .User should use a unique portfolio name as it shouldn't match with other user's portfolio details to avoid ambiguity.

PUT:

5 : Update Portfolio:

The form consists of four input fields:

- A dropdown menu labeled "portfolio_i" with an upward arrow icon.
- A dropdown menu labeled "Medium diversify" with an upward arrow icon.
- An input field containing the date "31/03/2018".
- A button labeled "Update Portfolio" with a rounded rectangular shape.



```
    return None

@app.route('/unauthorized_user/')
def unauthorized_user():
    return ("<h2>Please consider upgrading your access level, contact support@test.com</h2>")
```

Login authentication:

Login manager handler for authenticating user

```
@Loginmgr.user_loader
def load_user(uid):
    try:
        return User_Login.query.filter(User_Login.id == int(uid)).first()
    except:
        return None
```

```
    if user:
        if user is not None and user.password_check(password):
            flash('msg: Successfully logged in')
            login_user(user)
            session['uid'] = user.id
            # portfolio = User_Portfolio_data.query.all()
            # portfolio_by_user = User_Portfolio_data.query.all() # need to add user_id co
            roleid = User_Roles.query.filter(User_Roles.user_id == user.id).first()
```

Test Cases for role based authentication:

Highest level of access: Pro

testuser123

Testuser123

layem39812@sartess.com

Less restricted level of access:Advanced

testuser12

Testuser12

kikani2144@procowork.com

Most restricted level of access:Basic

testuser1

Testuser1

beweja6058@royins.com

Basic

Sample Tickers:

AAPL
GOOGL
MSFT

Upon selecting tickers clicking on “search portfolio” if user clicks on “get charts” button , user would get daily returns vs smoothed daily returns over a period on 30 days