



**A trading app that shows “experienced”
analytics for the stocks / portfolio**

Date 10/20/2022

Executive summary

Main conclusion

Goal of the Project

- To apply all the modules learned so far in Fintech Bootcamp
- To be able to trade stocks, perform stock / portfolio analysis and generate relevant reports
- Idea is to generate a trading app that helps user look at the history, present risk/ratio metrics and future possibilities for individual stocks as well the user portfolio
- App would enable user to look at analytics and make informed decision on his/her stock investment choices

```
$ python app.py
-----
Welcome to 'XPlytics'!!!
-----
.....Inside authentication module.....
? Are you a new user? No
Please sign in
? Please enter your username test1
? Please enter your password test1
Login successful
Funds available to trade for you are 2911.339988708496
.....End of authentication module.....
? What would you like to do? (Use arrow keys)[]
» Update available amount for trading
  Trade Stocks
  View current portfolio
  Stock Analysis
  Portfolio Analysis
  Delete User
  Exit the application
```

Team

Awesome team members

Team members

- Pravin Patil
- Esteban Lopez
- Jorge Villacreses
- Liset Lopez
- William Wolfenbarger

Data Collection

Decided not to use alpaca since it only retrieves up to 3 years data only.

Data Input

- Project used Fire and Questionary libraries to collect inputs from user via command line interface. For instance,

```
# function to load user options
def load_user_options():
    signed_in_user_choices = [
        'Update available amount for trading',
        'Trade Stocks',
        'View current portfolio',
        'Stock Analysis',
        'Portfolio Analysis',
        'Delete User',
        'Exit the application'
    ]

    user_choice = qs.select(
        "What would you like to do?",
        choices=signed_in_user_choices
    ).ask()
```

Market Data Collection

- Code uses 'YFinance' API to download market data.

```
## Use Yahoo finance to retrieve 5 yrs of price data
portfolio = yf.Tickers(tickers)
portfolio_history_df = portfolio.history(period = "5y")
prices_df = portfolio_history_df["Close"]
```

Data Cleanup

- DataFrame function `‘.dropna()’` was used for data cleanup (especially on returns data).

Data Exploration

- On multiple occasions, following tactics were used to filter out irrelevant data:
 - `_df.drop(columns = ["x"])`
 - `_df[x]`
- Local functions were defined to get data out of panel widgets
 - `_wd = pn.widgets.Select(options=`
 - `def _name(_wd):`
 - `name = pn.pane.Markdown("# " + yf.Ticker(_wd).info['longName'])`
 - `return name`
- SQL Lite database (with `‘sqlalchemy’` library) was used to interact with database

```
# retrieve user portfolio details
user_portfolio_query = f"""
SELECT
    ticker, number_of_shares
FROM
    portfolio
WHERE user_name = '{user_df['user_name'].iloc[0]}'
"""
portfolio_df = pd.read_sql_query(user_portfolio_query, db_engine)
```

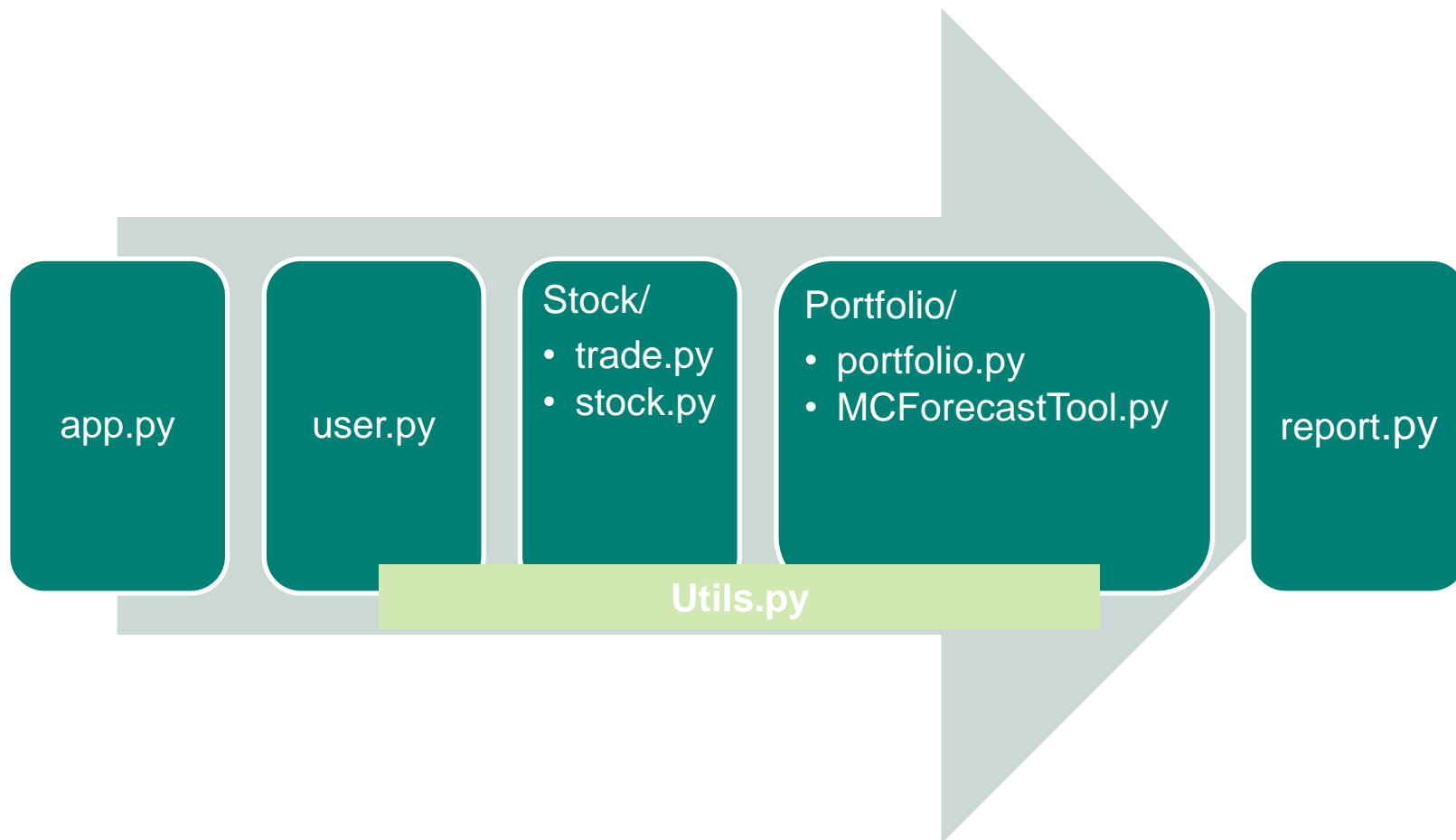
Approach

- We used typical software development lifecycle methodology
 - Defined requirements (user stories, technology details, user interface)
 - Post definition of requirements, various components were envisaged. For instance:
 - user module
 - trade module
 - stock and portfolio analysis module
 - and visualization or reports module
 - These modules were divided amongst members and assigned one project manager to ensure integration of all components

- After individual components were built, we integrated the code into the app
- Next step was code review, QA testing
- And finally, documentation (readme file)

/ Fintech-Workspace / project1

Name	Las
portfolio	a
report	34 m
resources	
stock	
user	
app.py	
MCForecastTools.py	41 m
README.md	
TradingApprequire...	
utils.py	



Libraries

- 'Pandas' library to work with dataframes and analyse timeseries data.
- 'YFinance' API to download market data.
- 'Numpy' library in integration with 'Pandas' to manipulate financial data.
- 'MCForecastTools' file for Monte Carlo Simulations.
- 'Panel' library for creating an interactive web dashboard.
- 'Fire' and 'Questionary' library for Command Line Interface.
- 'SQLAlchemy' toolkit for application development.
- 'hvplot' and 'matplotlib' for data visualization.
- 'statsmodels' for linear regression to calculate stock alphas

Github project

- <https://github.com/prpercy/project1>

Results

App produces fantastic interactive report using Panel template

Interactive report

- Report is intuitive, interactive and responsive to user selections
- Report shows past, present and future perspectives providing deep understanding
- Report also allows user to explore more through interactive options to analyze or run a deep dive

XPlytics: Portfolio Analysis

Press Me to Exit App

Ticker
MSFT

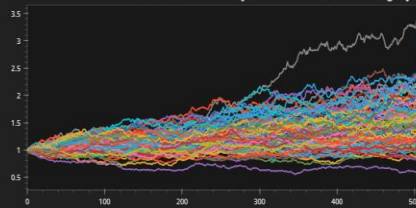
Microsoft
Corporation

<https://www.microsoft.com>

Microsoft Corporation develops, licenses, and supports software, services, devices, and solutions worldwide. The company operates in three segments: Productivity and Business Processes, Intelligent Cloud, and More Personal Computing. The Productivity and Business Processes segment offers Office, Exchange, SharePoint, Microsoft Teams, Office 365 Security and Compliance, Microsoft Viva, and Skype for Business. Skype, Outlook.com, OneDrive, and LinkedIn and Dynamics 365, a set of cloud-based and on-premises business solutions for organizations and enterprise divisions. The Intelligent Cloud segment licenses SQL, Windows Servers, Visual Studio, System Center, and related Client Access Licenses; GitHub that provides a collaboration platform and code hosting service for developers; Nuance provides healthcare and enterprise AI solutions;

Following Monte-Carlo simulation is performed for the user portfolio with weights calculated according to number of shares s/he holds :

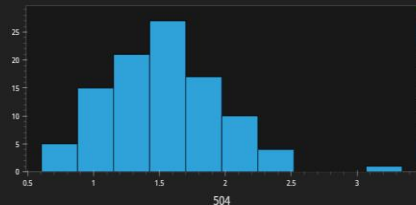
100 Simulations of Cumulative Portfolio Return Trajectories Over the Next 504 Trading Days.



2 year simulated cumulative returns

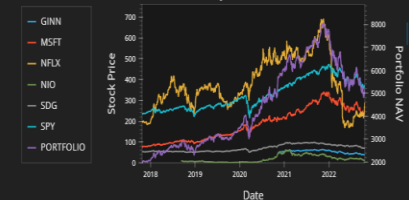


Distribution of Final Cumulative Returns Across All 100 Simulations



Stat Values	
count	100.000000
mean	1.527808
std	0.446593
min	0.604714
25%	1.197569
50%	1.483323
75%	1.778630
max	3.342260
95% CI Lower	0.845952
95% CI Upper	2.369806

Historical Prices - 5 years



Historical Cumulative Returns - 5 years



Index	Correlation	Volatility	Sharpe	Sortino	Covariance	Beta	alpha
GINN	0.891658	0.245207	-0.356267	-0.528003	0.000161	1.177538	-0.000627
MSFT	0.821503	0.270863	0.36394	0.534488	0.000164	1.198405	0.000106
NFLX	0.519853	0.532906	-0.181129	-0.203313	0.000204	1.492024	-0.000738
NIO	0.446619	0.787852	-0.501575	-0.86232	0.000259	1.895072	-0.002019
SDG	0.78767	0.150926	-0.414212	-0.64408	0.000088	0.640253	-0.0004
SPY	1.0	0.185676	0.3228	0.445252	0.000137	1.0	0.0
PORTFOLIO	0.858997	0.252092	-0.067196	-0.094497	0.000016	1.166256	-0.000345

Ratio

Correlation

Key Ratios



Next Steps

Next Steps

- App could be further developed to add:
 - proper user interface
 - and actual market interface to perform actual trading