AY 2021-2022 Term 1

IS453 Financial Analytics

PROJECT ATLAS

Prepared for:

Prof. Seema Chokshi

Written by Group 8:

Chan Yao Ying
Naomi Han Yuxin
Tan Hua Juan
Tchea Jia Wei
Wee Heng Lee (Henry)

Submitted on:19/11/2021

1. Introduction

1.1 Topic chosen

Our team assumes the role of an asset manager to use stock and fund data to create an interactive application that supports analysis of financial instruments and portfolios.

1.2 Background research

In 2015, Deloitte consulting division published a study on the ten disruptors to the wealth management industry. One of the key disruptors was analytics and big data, and as volume of consumer data continues to grow exponentially, firms are investing in building more descriptive and predictive models to provide more complete insights into markets.

In summary, the rising cost of regulatory burdens and wages has forced firms to find other means of reducing cost. As new technology becomes readily available and more cost efficient, there is a shift from reliance on human resources in order to automate and streamline data collection that has zero human error. Apart from data reconciliation, according to a survey by consulting firm Corporate Insight, the total assets managed by the 11 leading robo advisors in the US rose 65% over the past year, hitting $19 billion in December 2014. The hybrid advisory model that takes into account both science-based and human advice will most likely be the new standards in the coming years.

1.3 Problem Statement and Situation Analysis

Through the background research and interview with an actual asset manager, we have discovered the following problems. Most of the sub-fund portfolios are reviewed on a quarterly basis, the portfolio reviews are usually done manually through various excel sheets connected to the bloomberg terminal service (refer to Appendix D). As such, it requires intensive manual labour to extract data from bloomberg terminal and cross check transactions through various custodian accounts.

Furthermore, the data entries are susceptible to human error and would usually require a few reviews before presenting to the client. Hence, for the purpose of this project we have decided to focus on addressing the following pain points:

1. Slow process to update client portfolio
2. Human Errors from manual work
3. Lack of a centralisation for visualisation

2. Data Sources

2.1 Yahoo Finance

Yahoo is a large network that covers many areas, including financial information, also known as Yahoo Finance. Yahoo Finance provides all types of financial data, updates, stock quotes, charts, and more. Furthermore, by using the “pandas_datareader” package, we are able to extract the data needed for our analysis such as financial statements and stock daily closing price.

2.2 Packages we use to pull data

The key list of packages adopted in our Jupyter notebook can be found in Appendix F.

3. Analytics Solution and Framework

3.1 Consolidation of Information on Tableau

Our team developed a front-end visualisation with Tableau to allow our client to have a quick overview of their portfolio analysis while hiding the underlying complex financial algorithms used to manage their portfolio.

Based on our financial strategies recommended for our client, we have developed three dashboard categories namely Portfolio Allocation, Trading Strategy and Financial Statement.

3.1.1 Portfolio Allocation and Trading Strategy Dashboards

3.1.1.1 Portfolio Allocation Dashboard

Our client is able to view the relative comparison of their portfolio tickers from a combination of financial ratios such as min volatility, max sharpe and sortino ratios to gauge the performance of their portfolio found in appendix a.

3.1.1.2 Trading Strategy Dashboard

This dashboard displays the sector performance for our client to compare their existing portfolio with a particular sector to view their historical returns with filters that they could select such as different types of sectors comparison and date of portfolio rebalance.

In addition to this, the auto arima model that we developed for our client would allow them to view historical price movement and future price prediction from the available tickers found in Appendix B.

3.1.2 Financial Statement Dashboard

To better understand the financial performance and business activities of the companies that our client has invested in, especially in identifying whether the company is profitable, we have included common financial records such as Balance Sheet, Income Statement and Cash Flow (found in Appendix C).

3.2 Solution Architecture

3.2.1 Purpose

In order to limit human errors as much as possible, we sought to provide our solutions via an interactive platform. Tableau serves this purpose well as it allows our clients to not only interact with the output data of our analytics solutions but all the visualizations can be centralised within the Tableau notebook.

Various backend technologies have been utilised to serve the analysis. These backend components serve these purposes:

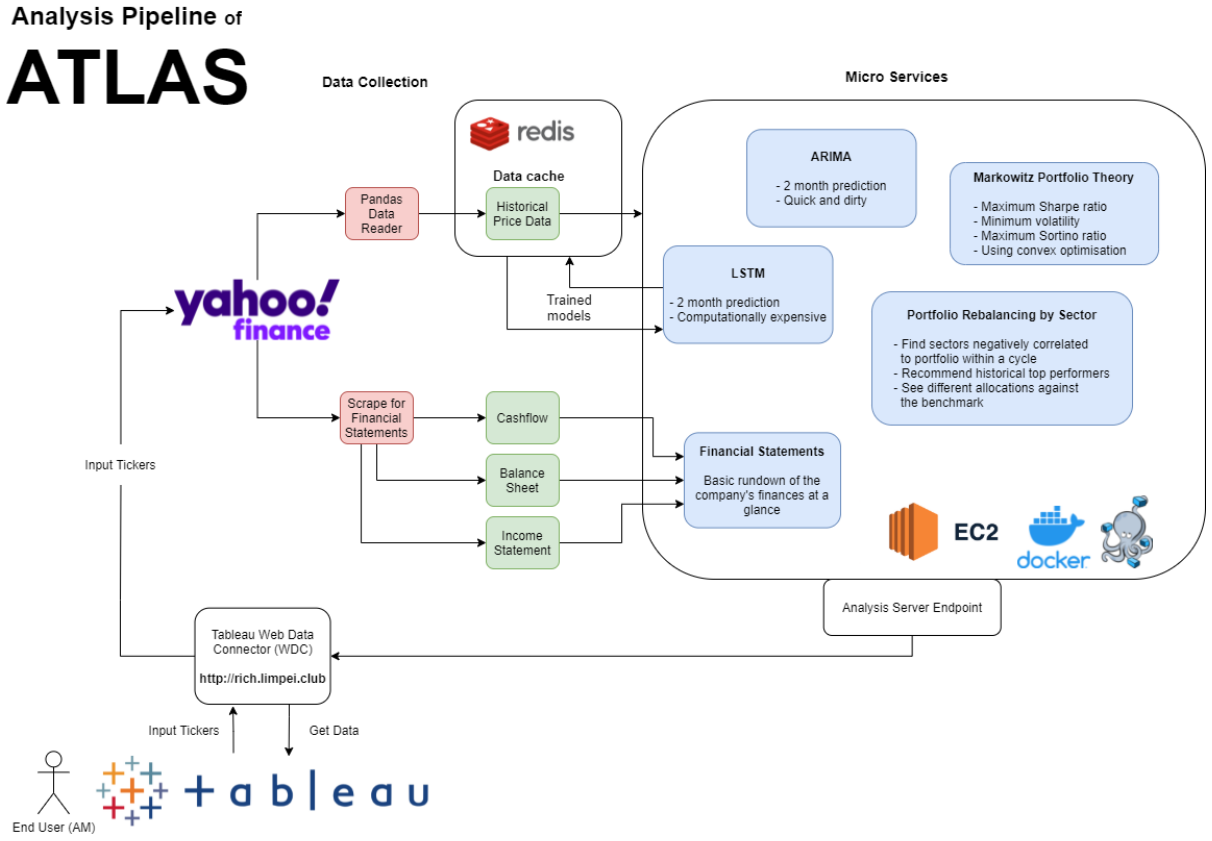
- High availability
- High speed
- High performance
- Low computational overhead costs
- Code-free (for clients)

3.2.2 Architecture Implementation

The code repository of the backend services and Tableau workbook are available here: https://github.com/chanyaoying/ATLAS-Analysis-Servers

Instructions on how to access our solution (or run the solution on your local machine) is accessible on the Github repository README.

Here is a high-level overview of the solution architecture:



3.2.2.1 Cloud Deployment (AWS)

To allow our clients to access our platform, we would either have to package it as a native/desktop application, or deploy it on the cloud. The latter option is more viable as it presents lower computational overhead costs for the clients' machines.

We chose to deploy it on AWS as we are more familiar with it.

In actual production (if we are actually deploying it) we would need to include auto scaling both between instances (with load balancers) and within instances (with Kubernetes pods for example). We are only deploying it on a single instance as a proof of concept.

3.2.2.2 Analysis Services

Each service or model is packaged as a standalone micro service and exists in a docker swarm, deployed on the cloud as an AWS EC2 instance. It is essential that our analytics solutions are deployed as micro services for these reasons:

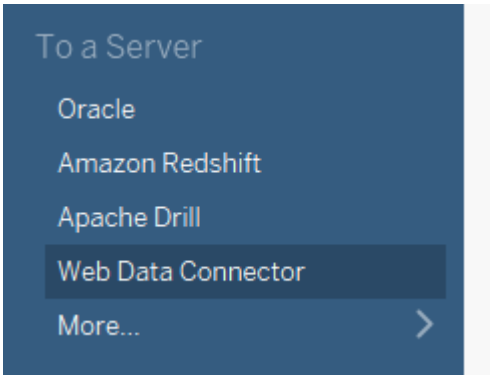
- Much higher scalability
- Improved fault tolerance
- Future proof

As we cannot expect users to use all the micro services at once every time, we will have to allow each micro service to scale according to the demands.

Since each micro service runs autonomously and independently from one another, having a fault in one would not affect the uptime for the other.

Having the option to upgrade each service independently without affecting the others can help us launch part of our platform while we continue to develop the rest of the application. This is an especially important benefit as the financial markets are always quick to evolve, and financial technologies will evolve just as quickly.

3.2.2.3 Tableau Web Data Connector (WDC)



In order for Tableau to talk to our micro services, we have to make use of the Tableau Web Data Connector (WDC). The WDC is essentially a HTML page with javascript code that will parse the requests of our clients via a form, then package the request as a series of API calls to those micro services.

The WDC is accessible via http://rich.limpei.club, but should only be accessed from Tableau. Our clients will just have to reload the WDC from the prepared Tableau workbook and require no further set up. A screenshot of the WDC can be found at Appendix .

3.2.2.4 Redis Cache

Redis is an in-memory data structure store that stores data in key-value pairs. This provides an O(1) time complexity lookup in the RAM.

In the lab exercises done in class, for every analysis that we perform, we would have to pull price data from Yahoo! Finance via pandas_datareader. For our purposes it would be very slow and inefficient to pull new data for every request that our client has. We made use of a cache to store the price data, and this price data is shared among all the microservices that need them for analysis. If our client requests price data that our backend has not seen before, new data is pulled from Yahoo! Finance then stored in the cache. If not, price data is served from the cache in a fraction of a second.

3.2.3 Model deployment

3.2.3.1 Considerations

In order to realise the potential of the analytics solution, we would have to consider the performance of the models in production. For the most part in our project, the nature and distribution of the data we use in development and production is the same. However unlike

traditional NLP or Classification models, time series forecasting models (LSTM & ARMIA) cannot be deployed and evaluated the same way.

With time series forecasting models, we have to retrain our model every time new data comes in. In a cat/dog image classification model like ImageNet, you can have a model that is robustly trained, and once deployed you have the confidence that it will perform similarly to how you have evaluated it in development. Time series forecasting models on the other hand do not work that way as the features of price data are not stationary. Cats and dogs will always look like cats in the future but the mean and standard deviation of price data is always changing.

That said, when evaluating the models, we can only use the train-test split to tune hyperparameters. In production, we are using 100% of the data for training.

3.2.3.2 Our project in deployment

Since our project is just a proof of concept, our project implementation is not enterprise-grade. Our priority is just to explore and find out further analytical concepts in finance and how it can be implemented in real-life. These are the limitations we set to reduce costs:

- LSTM model is not deployed
    - Too much time and computational power needed to train every model
    - Too much space needed to cache/store the models as they will have to be retrained with new data daily
    - Better optimised with GPUs, which is relatively expensive on the cloud
- All historical price data is limited to 2 years from the first day of the current month till the first day of this month
    - E.g. 2019-11-01 to 2021-11-01
    - Prevents new data from being pulled from Yahoo! Finance everyday

3.3 Strategies

Almost all rational investors in the world can be classified into two categories – passive investors and active investors. Passive investors, as the name suggests, often employ buy-and-hold strategies, where they hold a well-diversified portfolio into the long-term and are content with stable returns at a low risk. On the other hand, active investors make frequent trades in an attempt to beat the market and take advantage of short-term price fluctuations to make a profit. This usually entails more risks as profits and losses hinge on the accuracy of the active investor's predictions. Beginner investors similarly can be classified into one of these two categories according to their risk tolerance and time horizon.

For professional funds, the usual approach will be to have the majority of the asset allocated to broad-based ETF to get an overall exposure while allocating to individual names on the side to adjust for beta or alpha exposure. To help funds determine the optimal allocation, we will be utilising the Markowitz Portfolio Theory with a combination of sharpe ratio and sortino ratio to offset some of the inefficiencies discovered during the preliminary analysis which becomes our version of the post modern portfolio theory. Furthermore, we have included two predictive trading strategies, such as Auto Regressive Integrated Moving Average (ARIMA)

and LSTM (Long Short-Term Memory) Forecasting to give us a glimpse of the future price-action.

Lastly, we have included a sector rotation. This function aims to achieve two goals, to benchmark the current portfolio to sector related index on a time series for the asset manager to decide how they should balance or reallocate based on their experience and macro view of the economy. This is because lagging sectors may be due to seasonality.

3.3.1 Asset Allocation

3.3.1.1 Markowitz Portfolio Theory

In class we learnt of using Monte-Carlo simulations to visualise an efficient frontier using sharpe ratio as a means to form that efficient frontier.

In our project, we expanded that idea and sought to use a different method to find a faster way to generate the weights of an optimised portfolio. We have found that by using convex optimisation techniques provided by Scipy, we are able to shave down the time required to get the allocations to less than a second (if we do not need to pull price data from Yahoo! Finance). However, we will be unable to generate an efficient frontier. This would not pose a problem for us as our clients will be able to use their vast economical/financial knowledge to determine the best allocation for their clients, using our optimized allocation as a reference. This also allows us to to use this solution as an intermediate step for stock recommendation in portfolio rebalancing (outlined in 3.3.1.2 Sector Rotation Strategy) when we are performing backtests.

We used these metrics in portfolio evaluation:

- Sharpe Ratio
- Sortino Ratio
- Minimum Volatility

As opposed to Sharpe ratio which uses the standard deviation of the excess returns, Sortino ratio uses the downside deviation. In our case, we are using the downside risks. Downside risk is defined as the standard deviation of the downside, and the downside is simply the negative returns of an asset. This is arguably a better measure because by using the downside risk we are not penalising the volatility of positive returns, which should not be a source of worry for investors. Using the sortino ratio in MPT is also known as the Post-modern portfolio theory.

The convex optimization method that we used is Sequential Least Squares Programming Algorithm (SLSQP). This method seeks to minimize a multivariate scalar function, with constrained parameters. The scalar cost function in our case is the negative Sharpe ratio and negative Sortino ratio, and our parameters are the weights of the portfolio. We have to use the negatives because in order to maximize these ratios, we have to minimise the negatives.

```
def neg_sortino_ratio(weights, mean_returns, original_returns, risk_free_rate):
    returns = np.sum(mean_returns*weights)
    downside_deviation = downside_risk(weights, original_returns)
    neg_sortino = -(returns - risk_free_rate/252) * \
        np.sqrt(252) / downside_deviation
    return neg_sortino


def max_sortino_ratio(mean_returns, original_returns, risk_free_rate):
    num_assets = len(mean_returns)
    args = (mean_returns, original_returns, risk_free_rate)
    constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
    bound = (0.0, 1.0)
    bounds = tuple(bound for asset in range(num_assets))
    result = sco.minimize(neg_sortino_ratio, num_assets*[
                          1./num_assets, ], args=args, method='SLSQP', bounds=bounds, constraints=constraints)
    return result
```

3.3.1.2 Sector Rotation Strategy

U.S. Sector

ETF or ETN	Today	1 Mths	1 Yr	YTD
Energy (XLE)	-0.7%	8.7%	101.2%	51.6%
Financial Services (XLF)	-0.4%	5.6%	69.2%	36.6%
Technology (XLK)	0.4%	7.4%	42.5%	24.2%
Real Estate (XLRE)	-1.2%	5.9%	39.1%	30.8%
Consumer Discretionary (XLY)	0.3%	10.2%	37.6%	25.1%
Industrial (XLI)	0.0%	4.6%	37.6%	18.0%
Basic Materials (XLB)	-0.5%	5.9%	34.6%	17.6%
Communication Services (XLC)	0.3%	-0.1%	33.3%	19.0%
Healthcare (XLV)	1.0%	3.8%	31.6%	18.0%
Consumer Staples (XLP)	-0.1%	1.6%	14.0%	5.6%
Utilities (XLU)	-0.6%	3.9%	6.3%	6.7%

U.S. Sector - Small Cap

ETF or ETN	Today	1 Mths	1 Yr	YTD
Energy (PSCE)	-1.3%	6.7%	167.1%	85.0%
Consumer Discretionary (PSCD)	0.1%	-3.1%	64.7%	35.1%
Technology (PSCT)	-0.3%	2.0%	58.1%	17.1%
Financials (PSCF)	0.2%	3.8%	54.1%	24.0%
Industrials (PSCI)	0.2%	2.3%	49.8%	19.0%
Basic Materials (PSCM)	0.6%	2.0%	49.7%	14.2%
Utilities (PSCU)	-0.1%	0.2%	47.7%	22.6%
Healthcare (PSCH)	0.2%	-0.4%	33.9%	7.5%
Consumer Staples (PSCC)	1.7%	0.6%	32.1%	18.0%

Global Sector

ETF or ETN	Today	1 Mths	1 Yr	YTD
Energy (IXC)	-1.1%	7.6%	87.9%	43.0%
Financial Services (IXG)	-0.5%	5.0%	58.5%	27.7%
Technology (IXN)	0.1%	6.1%	41.6%	20.9%
Industrial (EXI)	-0.3%	3.2%	34.8%	14.9%
Consumer Discretionary (RXI)	-0.2%	7.6%	33.0%	15.5%
Basic Materials (MXI)	-0.9%	4.0%	30.5%	9.1%
Healthcare (IXJ)	0.6%	3.6%	27.3%	14.2%
Telecommunications (IXP)	0.1%	0.9%	25.4%	14.9%
Consumer Staples (KXI)	-0.3%	1.7%	15.0%	5.8%
Utilities (JXI)	-0.9%	4.3%	7.4%	3.0%

Sector rotation is an investment strategy that is based on predicting that money will move in a particular sector based on the current economic cycle (University of New South Wales, 2018). In most cases, money tends to flow into financials post recession due to the easing of monetary policies and increase in interest rates. As the economy picks up and times get better, spending will naturally increase, bringing liquidity into the consumer cyclical sector (Massey University, 2009). At times of economic weakness, asset managers can choose to rebalance or allocate funds into consumer defensive stocks like utilities or consumer staples.

To execute this strategy, you can invest in individual stocks or mutual funds and exchange- traded funds (ETFs) that target segments of the economy and make it easy to gain diversified exposure. For the purpose of our project, we will assume that the asset manager already has knowledge of the macro environment. In our proposed strategy, we will utilise broad based US ETFs to track the sectors and assume that the underperforming sector YTD has either been rotated out or yet to be rotated in. The strategy will recommend the top 3 stocks in that particular sector if they wish to invest in individual stocks or they can proceed with the ETF(s) to get a diversified exposure. The purpose of this strategy is to ideally increase the alpha of the portfolio if the mandate of the portfolio has leftover risk or volatility based on the client or fund strategy preference. The following details the implementation steps behind this strategy:

- Step 1: Conduct correlation analysis using the return series (based on “Close” price) computed for each ETFs representative of the main sectors in the US market, and your portfolio for the past 3 months to capture the effects of sector rotation within a quarter.

```
# get 3 month close (just to perform correlation analysis)
# cus we're assuming the cycle is 3 months
portfolio_3m_close = portfolio_2y_close.iloc[(
    len(portfolio_2y_close.index) // 6) * 5:-1]

sector_2y_adj_close = pd.DataFrame()
sector_2y_close = pd.DataFrame()

for etf in sector_etf:
    col_adj_close = get_price(etf, 'Adj Close')
    col_adj_close.columns = [etf]
    sector_2y_adj_close.index = col_adj_close.index
    sector_2y_adj_close = sector_2y_adj_close.join(col_adj_close)

    col_close = get_price(etf, 'Close')
    col_close.columns = [etf]
    sector_2y_close.index = col_close.index
    sector_2y_close = sector_2y_close.join(col_close)

sector_3m_close = sector_2y_close.iloc[(
    len(sector_2y_close.index) // 6) * 5:-1]
```

- Step 2: Get the historical top performers of the top 3 negatively correlated sectors. The idea is to identify stocks within the inversely correlated sectors which would help rebalance the portfolio in the future.

```
correlation = correlation.sort_values(by="portfolio", axis=1)
top3_inverse_sectors = correlation.columns.tolist()[0:3]

lagging_tickers = []
[lagging_tickers.extend(sector_dict[etf]) for etf in top3_inverse_sectors]

new_portfolios = [portfolio_tickers + [ticker]
                  for ticker in lagging_tickers]
new_portfolios_weights = {lagging_tickers[i]: get_optimal_allocation(
    new_portfolios[i]) for i in range(len(lagging_tickers))}
```

Snapshot of our sector_dict that we will be using for our sector rotation strategy- containing main sector ETFs and the top 3 representative stocks within each sector:

```
sector_dict= {

'XLB': ['LIN', 'ECL', 'GOLD'],'XLC': ['GOOGL', 'FB', 'NFLX'],'XLY': ['AMZN', 'TSLA',
'NKE'],'XLP': ['KO', 'WMT', 'PEP'],'XLE': ['XOM', 'CVX', 'PSX'],'XLF': ['V', 'MA',
'JPM'],'XLV': ['JNJ', 'PFE', 'UNH'],'XLI': ['HON', 'GE', 'FDX'],'XLR': ['AMT', 'CCI',
'PSA'],'XLK': ['MSFT', 'AAPL', 'CRM'],'XLU': ['NEE', 'DUK', 'XEL']}
```

- Step 3: Apply Markowitz Theory (covered in 3.2.2.) to get different flavours of allocations to backtest the strategy against benchmarks. Subsequently, the list of the return series with the maximal optimal sharpe/sortino ratio (i.e. Risk-adjusted Returns), and minimal volatility weights for each sector rotation pair will then be passed and displayed onto our dashboard. From which, the asset manager will then perform the final decision on what works best for the client's portfolio diversification, taking into account macro environment factors.

```
for suggested_ticker, weights in new_portfolios_weights.items():

    types = ['max_sharpe', 'min_vol', 'max_sortino', 'equal_weight', 'ticker_only']
    suggested_ticker_2y_adj_close = get_price(
        suggested_ticker, 'Adj Close')

    for i in range(len(types)):
        suggested_portfolio = portfolio_2y_adj_close.copy().join(
            suggested_ticker_2y_adj_close)
        suggested_portfolio = (
            suggested_portfolio.pct_change() + 1).cumprod() - 1
        suggested_portfolio = pd.DataFrame((suggested_portfolio * weights[i]).sum(axis=1)).rename(columns={
            0: 'returns'})
        suggested_portfolio['allocation_type'] = types[i]
        suggested_portfolio['allocation_weights'] = json.dumps(weights[i])
        suggested_portfolio['title'] = f"portfolio + {suggested_ticker}"
        suggested_portfolio['date'] = suggested_portfolio.index
        suggested_portfolio = suggested_portfolio.to_json(orient="records")
        output += json.loads(suggested_portfolio)
```

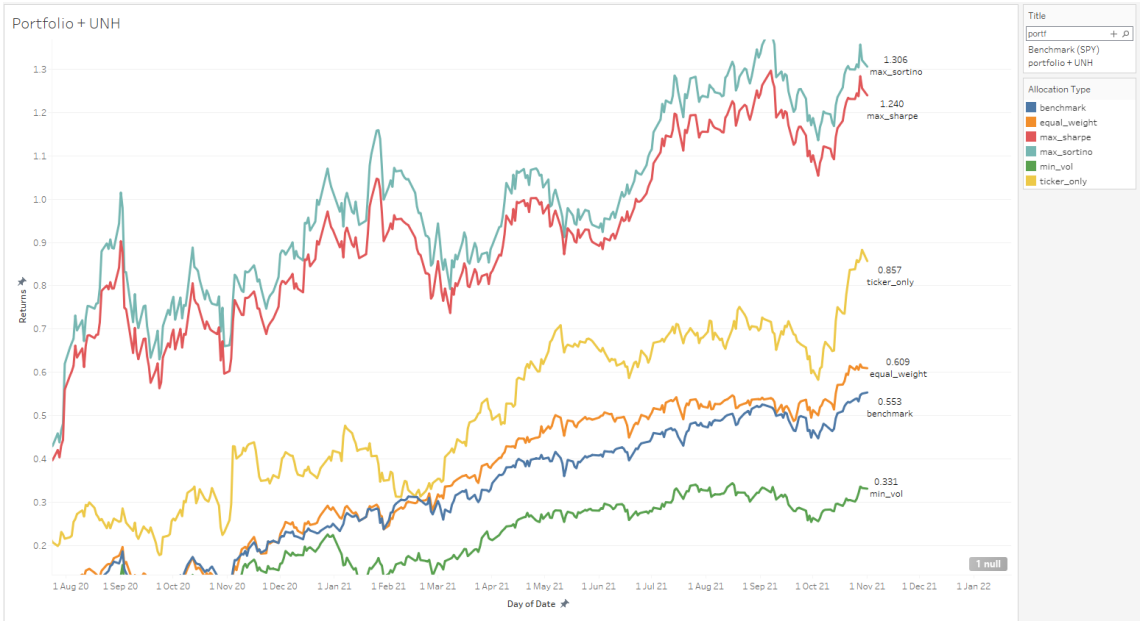


Fig 1: Sample visualization for a sector rotation pair

3.3.2 Staying ahead of the curve

3.3.2.1 ARIMA

Auto Regressive Integrated Moving Average (ARIMA) is a statistical analysis model that is widely used to predict future trends by analysing time series datasets. In the realm of financial modelling, ARIMA models are also known to be robust and efficient in financial time series forecasting, and constantly outperformed other more complex techniques, such as artificial neural networks (ANN) and generalised autoregressive conditional heteroskedasticity (GARCH), in the accuracy of short-term forecasts. As such, we are using the auto-ARIMA model to predict future stock close prices using past close stock prices.

An ARIMA model is denoted in the following form: ARIMA(p,d,q), where p is the parameter used to define the AR term which refers to the past values used to forecasting the next values, q is the parameter defining the MR term which refers to the number of past forecast

errors, and d is the parameter defining the I term which refers to the number of differencing transformations required to perform on the series to make it stationary if it is not. When applying the ARIMA model, we need to use series that are stationary, meaning that the series do not have seasonal effects. We also need to determine the values of the parameters p,d,q. By using Auto ARIMA, it helps us automatically select the best combination of (p,d,q) that provides the least errors.

In our case, we have created an Auto-ARIMA model to forecast a 2 weeks' stock close prices of Apple (2021-01-01 to 2021-01-14) using the past stock close prices (2019-01-01 to 2020-12-31) as shown below.

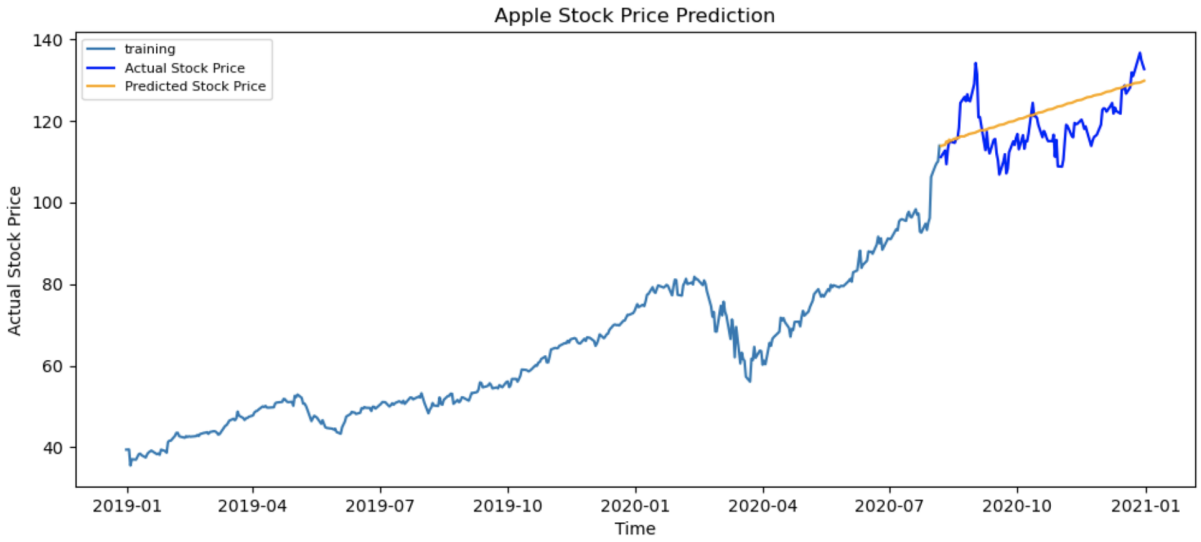


Fig 2: Visualisation of Apple stock price prediction with Auto-ARIMA

The Auto-ARIMA model uses past data to understand the pattern in the time series. Using these values, the model captured an increasing trend in the series as it is evident from the plot. However, the model did not predict the dips and spikes.

To evaluate the performance of the model, we calculated the root mean squared error (RMSE) value using the predicted stock prices and the actual stock prices on the validation set. The RMSE calculated from the model is 7.23, which means that the prediction is on average +/- $7.23 away from the actual price.

These are the limitations of the model. Being only capable of predicting the general uptrend of the stock price is not enough to provide a well-informed insight to the asset manager. The prediction error of 7.23 is also not reliable enough to invest based on the prediction. One of the opportunities for future improvement could be to train the model with various stocks to reduce the prediction error.

3.3.2.2 LSTM Forecasting

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture which has feedback connections attached to certain layers of the network. Unlike conventional RNNs, LSTM is well-suited to learn from experience to predict time series with arbitrary time steps. Furthermore, its memory unit is able to retain the time-related

information for an arbitrary amount of time, solving the vanishing gradient problem and making it an effective method to predict stock prices and profits surpassing that of conventional RNNs. Since stock prices are generally considered to follow a random walk, they do not deviate much from their previous day's values.

LSTM's ability to 'remember' previous stock prices makes it a very powerful tool in stock price prediction, outperforming other memory-free methods such as random forests, deep neural networks and logistic regression classifiers. Even when compared against other state-of-the-art forecasting methods including NAÏVE, ETS, and SARIMA, LSTM still remained superior with lowest root mean squared errors (RMSE) (Fischer, T., & Krauss, C, 2018).

In our project, we used the LSTM model to predict SPY 500 stock next day close price using the previous 6 months close prices. We trained the model with the past 5 years data (2016-11-01 to 2021-11-01).

As seen in the figure below, the LSTM model we created performed quite well as it could predict the dips and the spike accordingly. The model has an RMSE score of 8.11. However, the model may be overfitted as the actual values were fitted into the model during training.

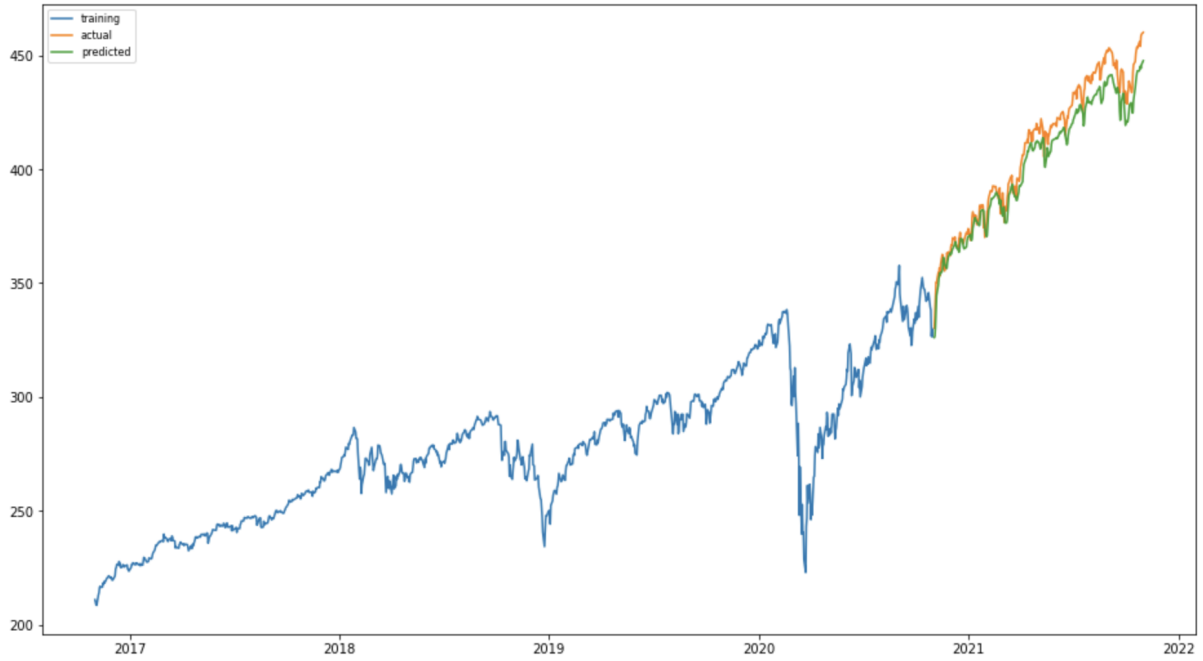


Fig 3: Visualisation of SPY 500 stock price prediction with LSTM model

To test out that theory, we tried to forecast the next 6 month data which is unseen to the model and the model was only able to predict a general downward trend with a slight rise at the end.

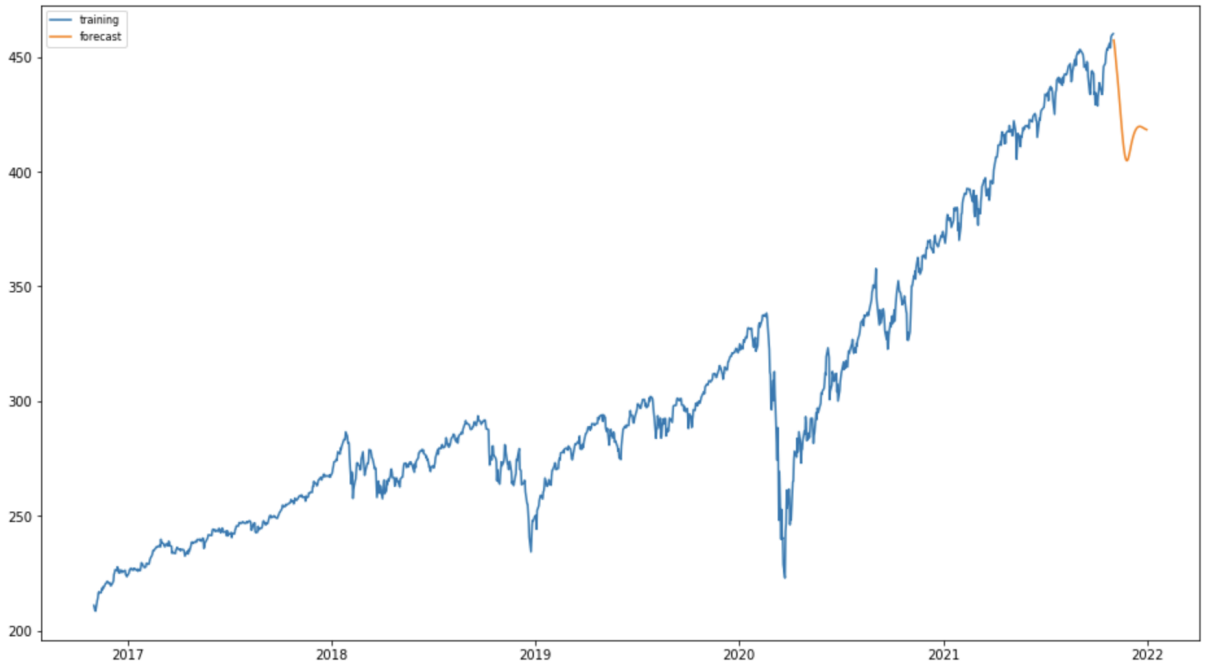


Fig 4. Visualisation of SPY 500 stock price forecasting with LSTM model

As mentioned above (3.2.3 Model Deployment), LSTM is an expensive model to include in production. As new data comes in, we will have to retrain the model from scratch, and for each asset we will have to have a model for it.

4. Future Work and Considerations

4.1. Sentiment Analysis

In the future, we can also consider running social analytics tools to perform sentiment analysis on the market such as analysing what people are talking about in social media and forums in relation to news announcements. This could in broad terms help in figuring out whether the crowd has either a bullish market or bearish market sentiment affecting the price movements which could be technical indicators to help investors to form a long/short term strategy accordingly.

4.2. Algorithmic Trading

Algorithmic trading is the use of automated systems that leverage on technical analysis to open and close positions when criterias are met. According to a research done by Mordor Intelligence, the algorithmic trading market is expected to witness a CAGR of 11.23% over the forecast period (2021-2026). As institutional investors plan to increase efficiency while cutting costs, firms have been actively investing in algorithmic platforms to stay ahead of the curve. It is only a matter of time before it becomes the norm in the financial markets. With the current visualisation being hosted on a local network via API, firms that are utilising ATLAS platform will already be one step closer to having the basic infrastructure in place to work towards cloud-based algorithmic trading. If the opportunity arises, firms that are able to invest in data mining and consider developing quantitative trading systems that rely on advanced mathematical analysis on a variety of datasets.

5. Conclusion

Our solution caters to asset management firms that have yet to take the first step to digitalisation. The tools and strategies introduced in our solution allow firms to visualise data on a single platform, generate useful second opinions and recommendations based on their current holdings and macroeconomic climate. Also, the interactive portal powered by Tableau will make it possible for firms to allocate more funds towards hiring more qualified individuals without compromising standards and focus on developing the company by developing more advanced trading strategies.

6. References

Deloitte. (2015). *10 Disruptive trends in Wealth Management (Vol. 17) [E-book].* Retrieved from https://www2.deloitte.com/content/dam/Deloitte/us/Documents/strategy/us-cons-disruptors-in-wealth-mgmt-final.pdf

Deloitte. (2019, April 8). *What challenges are fund managers facing?* https://www2.deloitte.com/fr/fr/pages/risque-compliance-et-controle-interne/articles/what-challenges-are-fund-managers-facing.html

Fischer, T., & Krauss, C. (2018). *Deep learning with long short-term memory networks for financial market predictions.* European Journal of Operational Research, 270(2), 654-669.

Guggenheim. (2020). *Asset Class Correlation Map. Historical Correlation: January 2011–December 2020.* Retrieved from https://www.guggenheiminvestments.com/mutual-funds/resources/interactive-tools/asset-class-correlation-map

Massey University. (2009). *Sector Rotation over Business Cycles. Department of Finance and Economics.* Retrieved from https://www.researchgate.net/publication/228425439_Sector_rotation_over_business-cycles

Nissa, N. K. (2020, October 29). *Stock Price Prediction using Auto-ARIMA.* Medium. https://nzlul.medium.com/stock-price-prediction-using-auto-arima-5569fcceae59

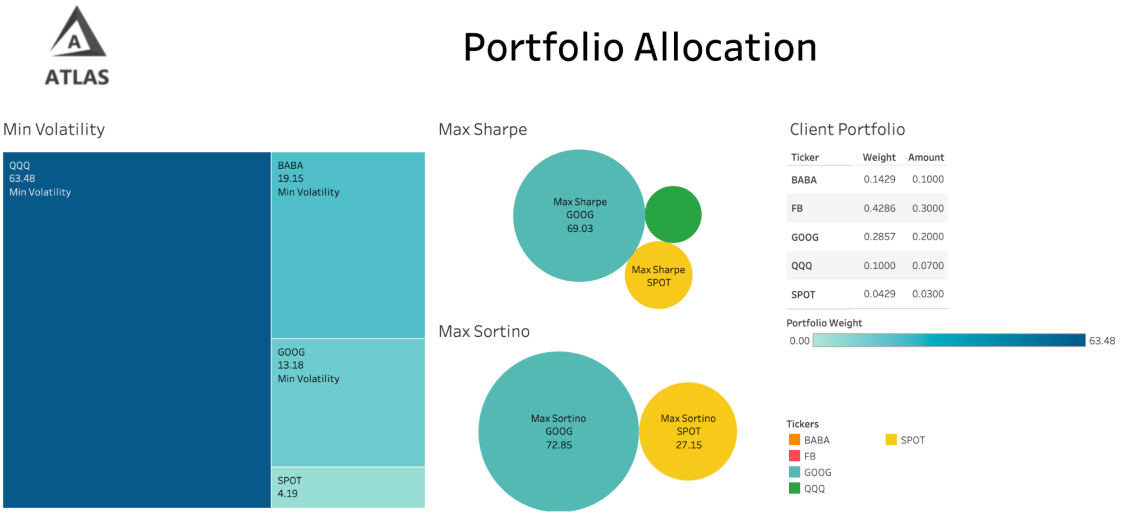
Prabhakaran, S. (2021, November 14). *ARIMA Model - Complete Guide to Time Series Forecasting in Python.* Retrieved from https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/

Singh, A. (2020, May 25). *Time Series Forecasting Methods | Arima In Python and R.* Retrieved from https://www.analyticsvidhya.com/blog/2018/08/auto-arima-time-series-modeling-python-r/#h2_3

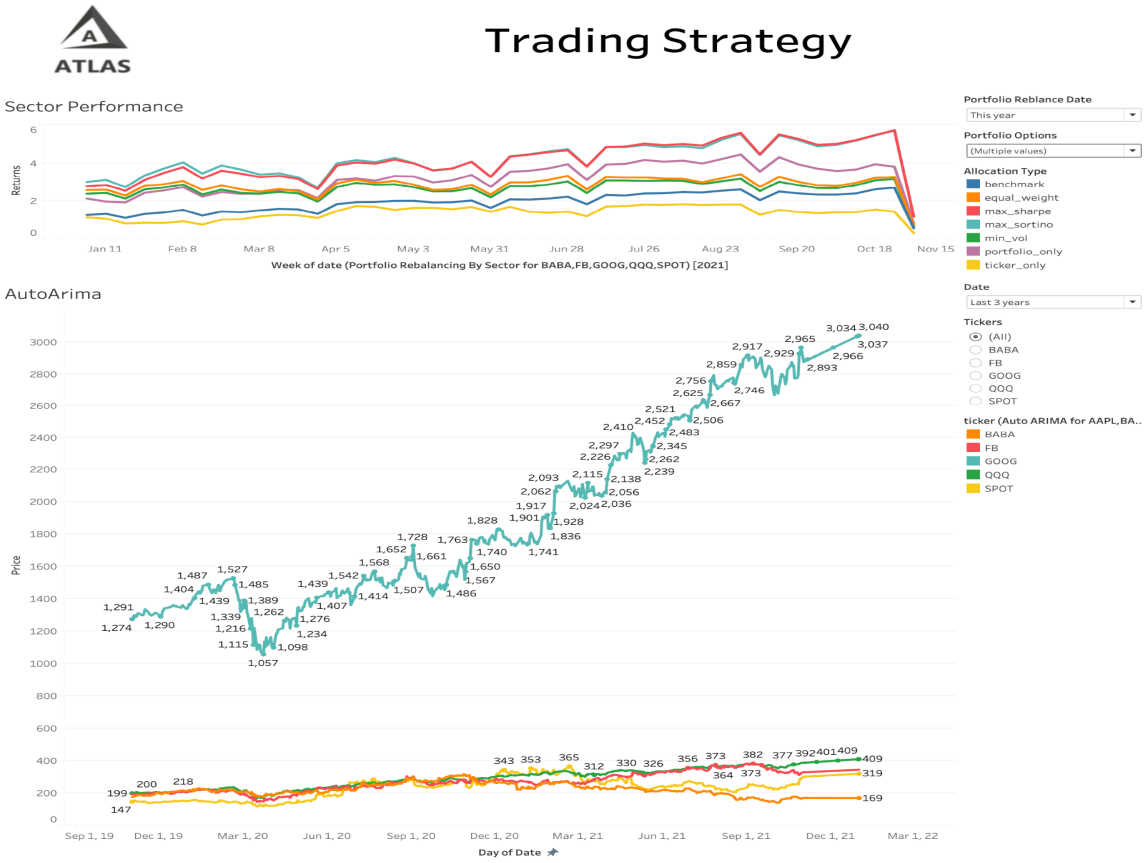
University of New South Wales. (2018). Dynamic Asset Allocation Using Systematic Sector Rotation. IZA Institute of Labor Economics. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract_id=548162

7. Appendix

Appendix A



Appendix B

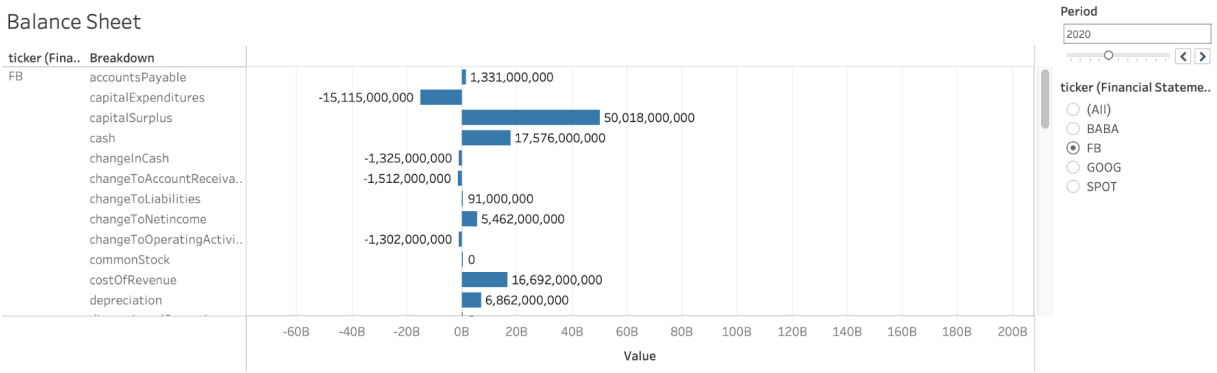


Appendix C

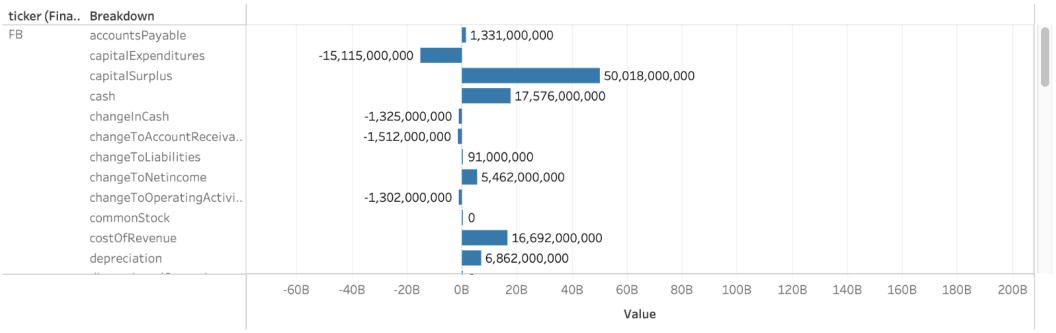
ATLAS

Financial Statement

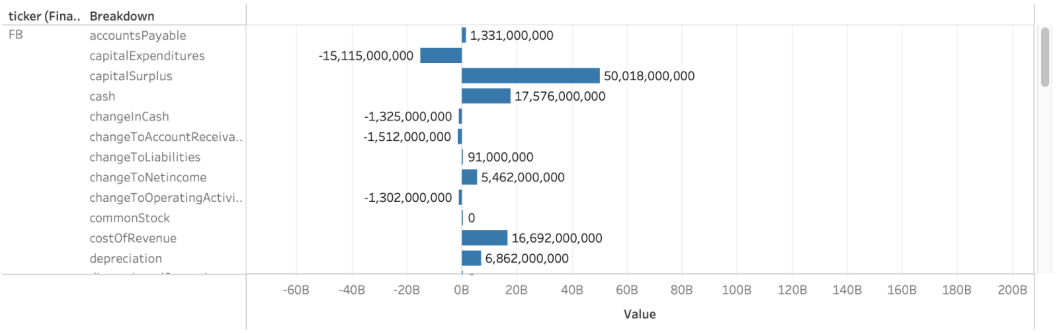
Balance Sheet



Income Statement



CashFlow



Appendix D

Summary as of 28th July 2021 (By Custodian)
截至 2021 年 7 月 28 日的摘要 (托管人提供)

12th IC Meeting on 31st May 2021
第12次投資委員會會議-2021年5月31日

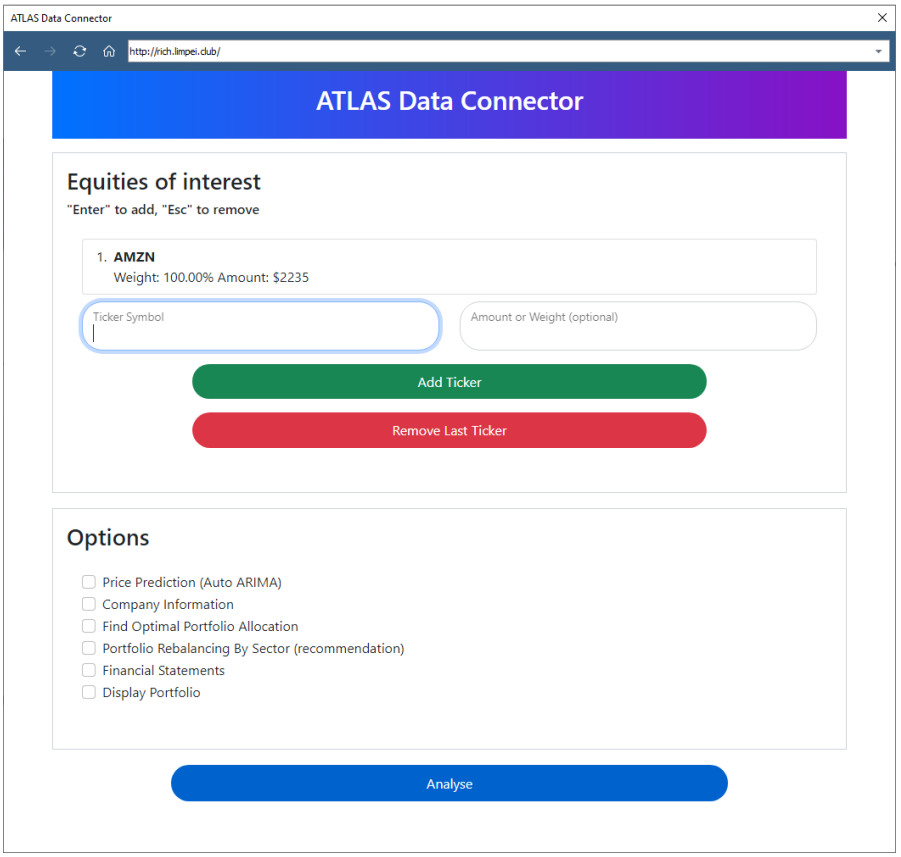
			總額Total	#NAME?
Global 全球的	Dividend Accrual 應計股息		#NAME?	USD 10,843
	Bank Bonds 銀行債券	#NAME?	#NAME?	USD 9,836,266
	Bank Equity 銀行股票		#NAME?	USD 1,378,720
	Real Estate Bonds 房地產債券	#NAME?	#NAME?	USD 696,487
	Real Estate Equity 房地產股票		#NAME?	#NAME?
	Mining, Oil and Gas Bonds	#NAME?	#NAME?	USD 475,765
	Govt Bonds 政府債券	#NAME?	#NAME?	USD 481,730

As of 28th July 2021
截至 2021 年 7 月 28 日

			總額Total	#NAME?
Global 全球的	Dividend Accrual應計股息		#NAME?	USD 90,198.75
	Bank Bonds 銀行債券	#NAME?	#NAME?	USD 9,836,266
	Bank Equity 銀行股票		#NAME?	USD 1,572,400
	Real Estate Bonds 房地產債券	#NAME?	#NAME?	USD 696,487
	Real Estate Equity 房地產股票		#NAME?	#NAME?
	Mining, Oil and Gas Bonds	#NAME?	#NAME?	USD 475,765
	Govt Bonds 政府債券	#NAME?	#NAME?	USD 481,730
	Brokering Equity 經紀公司股票	#NAME?	#NAME?	USD 1,599,688

Asset Class 资产类别	Asset Class % 资产类别 %	Asset sub-Class 資產子類	Asset sub-Class % 資產子類 %	Mkt Value 市場价值
Cash & Eqv	1.6%	Cash 現金	-0.1%	-USD 15,489.59
		Accrued Interest	1.1%	USD 182,388.86
		Dividend Accrual 應計股息	0.5%	USD 90,198.75
Financials	74.7%	Bank Bonds 銀行債券	57.0%	USD 9,836,266
		Bank Equity 銀行股票	9.1%	USD 1,572,400
		Brokering House Equity	9.3%	USD 1,599,688
Real Estate 房地產	18.1%	Real Estate Bonds 房地產債券	4.0%	USD 696,487
		Real Estate Equity	13.5%	USD 2,334,009
Others 其他	5.5%	Mining, Oil and Gas	2.8%	USD 475,765
		Govt Bonds 政府債券	2.8%	USD 481,730
Total 總計	**100.0%**		**100.0%**	**USD 17,253,443**

Appendix E



Appendix F

```
#Data Manipulation and Treatment

import numpy as np

import pandas as pd

import datetime as dt

from datetime import timedelta

from math import sqrt

import math


#Plotting and Visualizations

import matplotlib.pyplot as plt

import seaborn as sns

# render the figures in this notebook

%matplotlib inline

#!pip install plotly

import plotly.express as px

import plotly.graph_objects as go

from plotly.subplots import make_subplots


#Scikit-Learn for Modeling

from sklearn.metrics import mean_squared_error,r2_score,
mean_absolute_error,mean_squared_log_error


#Statistics

import statsmodels.api as sm
```

```
from statsmodels.tsa.api import
Holt,SimpleExpSmoothing,ExponentialSmoothing

#from pmdarima import auto_arima


# Data reader

from pandas_datareader import data

from typing import Dict, List, Any

from matplotlib.ticker import FuncFormatter


# Checking Stationarity - Dicky Fuller Test

from statsmodels.tsa.stattools import adfuller


# Checking Trend and Seasonality

from statsmodels.tsa.seasonal import seasonal_decompose

import json

import datetime
```