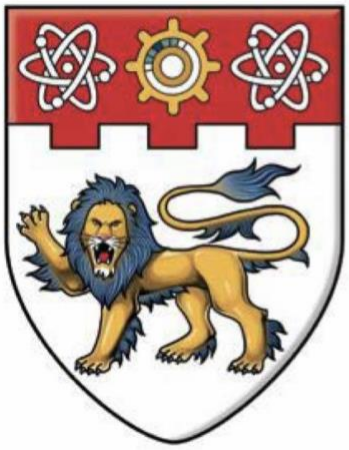


NANYANG TECHNOLOGICAL UNIVERSITY

PROFESSIONAL ATTACHMENT

FINAL REPORT ON PROFESSIONAL

ATTACHMENT WITH

MAF INTELLIGENCE PTE. LTD.

ROLE: FINANCIAL PORTFOLIO RISK MODELLING INTERN

PREPARED BY: AARON ANG WEI JIE

SPMS-MAEC (MATH AND ECONOMICS)

Contents

Table of Contents

Abstract

This report consists of three chapters in which I will describe my twelve-week experience at MAF Intelligence Pte Ltd as a Financial Portfolio Risk Modelling intern. This professional attachment took place in Summer 2019, from 13 May 2019 to 02 August 2019. This report entails the following sections in the order of: Introduction, Projects, and Summary. The Introduction section outlines the company's background and job scope of my internship role. The Projects section explains the details of tasks assigned, accompanied with their respective final output. Lastly, the report closes with the Summary section which includes a summary of the internship and my reflections on the experience.

Keyword: Professional Attachment; FinTech

List of Figures and Tables

1. Introduction

1.1 Purpose of Report

This report has been prepared based on practical working experience at MAF Intelligence Pte Ltd. as part of a 12-week attachment programme read as an Unrestricted Elective (6 AU) at NTU.

1.2 Company Background

MAF Intelligence is a Financial Technology organisation located in Singapore which provides flexible and relevant solutions to quantitative needs. Its informative dashboard with various quantitative models has information readily visualised for professionals or firms, allowing them to gain valuable insights and make wiser decisions.

1.3 Project Scope

As a Financial Portfolio Risk Modelling Intern, my task was to design and craft several financial models using Python language. The common practice is to retrieve relevant data from company's databases (SQLite) and to design an algorithm (create functions and APIs in Python) to generate the output. Subsequently, I will test the output using a software called Postman. Finally, I will work closely with the Front-end team to discuss on how to generate visualisable results such as charts for that specific model. Table 1 and Table 2 show the work plan for my internship role.

Table 1: Initial Plan of Financial Portfolio Risk Modelling Intern, Summer 2019

Schedule of Activities
Familiarize with Project Structure and Backend
Portfolio Performance Table (Create Model)
Portfolio Stress Test (Create Model)
Seasonal Analysis (Create Model)
Portfolios (Table)
Weekly Report will be done to record activities done

Table 2: Modified plan of Financial Portfolio Risk Modelling Intern, Summer 2019

Schedule of Activities
1. Familiarize with Project Structure and Backend
2. CTA Portfolio Performance Table (Create Model)
3. Seasonal Analysis (Create Model)
4. Forward Curve Table (Create Model)
5. Forward Curve (Create Model)
6. Continuous Contract
7. Position Calculation Model
8. Data Transformation
9. US Recession Probability

2. Projects

2.1 Familiarization with Project Structure and Back-end Logic

As I first embarked on the internship, the other interns and I were briefed by the Director and IT advisor on the whole structure of the organisation and the kinds of services the organisation delivers to its customers. With the guidance of the company's Back-end Developer, I managed to understand the functions of various software currently adopted by the company to carry out my tasks.

The following are the software I have learnt:

1. Pycharm (Python) – Explore the folders/scripts and learn what they do. Mainly the programming flow. Learn how to use Debug mode. Learn how RESTFUL APIs and Flask framework works.
2. DB Browser for SQLite – Learn how to collect relevant data by creating queries.
3. SourceTree – Learn how to push and pull commits and create and merge branches. This software allows multiple developers to work on one project simultaneously.
4. Postman – Learn how to test API.
5. Panda and Numpy (libraries in python) – For data manipulation and analysis.

On top of that, I had to read documentation on existing quantitative models employed by the company. I was required to resolve bugs contained in these models from time to time if any occurred. Throughout the internship, I managed to resolve bugs from quantitative models such as Correlation Matrix, Cross Correlation and Value at Risk.

2.2 CTA Portfolio Performance

Commodity Trading Advisor (CTA) is an individual or firm who renders advice and services related to trading in futures contracts and options on futures contracts. (Chen, 2019) The performance and risk of a CTA program is measured with a set of metrics and they are often compared against a benchmark index such as S&P 500. Table 3 shows the performance metrics the company has chosen to calculate for both CTA programs and the benchmark index.

The following illustrates what I have done with regards to the CTA program performance:

1. Understand how CTA works.
2. Create an API to generate the portfolio's daily unrealised profit and loss, realised profit and loss and Value at Risk value. This API will be used by the database team as a daily scheduler to save these data in the company's database.
3. Create a function to retrieve the daily unrealised and realised profit and loss, where these daily data will be converted to monthly data. The monthly total profit will then be calculated.
4. Create a function to retrieve data from the portfolio's general ledger, where these data will be used to calculate the portfolio's monthly ending equity balance.
5. Create a function to calculate the monthly return and rate of return for both CTA programs and benchmark products. Monthly return is defined as monthly profit divided by monthly ending equity balance.
6. Create an algorithm to compute the performance metrics as shown in Table 3.
7. Create an API for the Front-end team to call for the performance output.

Table 3: CTA performance and risk metrics

	Abbreviation	Formula	Parameters
1. **Rate of Return**	RoR_i	$(R_i / 100) + 1$	R_i
2. **VAMI**	$VAMI_n$	$1000 * RoR_1 * RoR_2 * \dots * RoR_n$	$RoR_{1,\dots,RoR_n}$
3. **Risk Free Rate**	R_{RF}	%, Updated daily	-
4. **S&P Value**	$VS\&P_i$	Updated monthly	Product details
5.**S&P Rate of Return**	$RoRS\&P_i$	$VS\&P_i / VS\&P_{i-1}$	VS&P
6.**S&P Monthly Return %**	$RS\&P_i$	$(RoRS\&P_i - 1) * 100$	RoRS&P
7. **Mean Return**	M	$(R_1 + R_2 + R_3 + \dots R_N) / N$	N, $R_{1,\dots R_N}$
8. **Total Compound Rate of Return**	Total RoR	$(RoR_1 * RoR_2 * RoR_3 * \dots * RoR_N - 1) * 100\%$	N, $RoR_{1,\dots,RoR_n}$
9.**Year to Date Return**	YTD RoR	total compound return since first month of the current year	N, $RoR_{1,\dots,RoR_n}$
10. **1 Year Return**		total compound return for past 12 months (N = 12)	$RoR_{1,\dots,RoR12}$
11. **3 Year Return**		total compound return for past 36 months (N = 36)	$RoR_{1,\dots,RoR36}$
12. **Compounded Monthly Return**		$((RoR_1 * RoR_2 * RoR_3 * \dots * RoR_N)^{1/N} - 1) * 100\%$	N, $RoR_{1,\dots,RoR_n}$
13. **Compounded Annual Return**		$((RoR_1 * RoR_2 * RoR_3 * \dots * RoR_N)^{12} - 1) * 100\%$	N, $RoR_{1,\dots,RoR_n}$
14. **# of Losing Months**		number of months where $R_i < 0$	R_i
15. **# of Winning Months**		number of months where $R_i \geq 0$	R_i
16. **Avg Monthly Loss**		$(R_1 + R_2 + R_3 + \dots R_M) / M$ where $R_i < 0$, M = # Losing months	R_i
17.**Avg Monthly Gain**		$(R_1 + R_2 + R_3 + \dots R_K) / K$ where $R_i \geq 0$, K = # Winning months	R_i
18.**Standard Deviation**		$(((R_1 - M)^2 + (R_2 - M)^2 + \dots + (R_N - M)^2) / (N-1))^{1/2}$ where R_i– return of specific month, M – mean of return, N – number of months.	$R_{i, M, N}$
19. **Annualized Standard Deviation**		Standard Deviation * SQRT(N) where N = number of periods in 1 year.	Standard Deviation
20. **Downside Deviation**		$((L_1^2 + L_2^2 + \dots + L_N^2) / N)^{1/2}$. Where $L_i = \min(R_i - R_{RF}, 0)$, N – number of months in calculation	R_i , $R_{RF, N}$
21.**Sharp Ratio**		$(M - R_{RF})$ / Standard deviation	M, R_{RF}, Standard deviation
22.**Sortino Ratio**		(Compound monthly return - R_{RF}) / Downside deviation	Compound monthly return, R_{RF}, Downside deviation
23. **Maximum Drawdown**		max $((VAMI_i - VAMI_j) / VAMI_i) * 100\%$ where j > i and for any j $VAMI_i > VAMI_j$	$VAMI_j$
24. **Clamar Ratio**		Compound annual return / Maximum drawdown	Compound annual return, Maximum drawdown
25. **Sterling Ratio**		Compound annual return / ABS (Average Drawdown - 10%). Where Average Drawdown = $(DD_1 + DD_2 + DD_3) / 3$. Where DD_1 = Maximum drawdown for first 12 months, DD_2= Maximum drawdown for next 12 months, DD_3 = Maximum drawdown for last 12 months.	Compound annual return, Average Drawdown
26. **Beta**		$((R_1 - M) * (RS\&P_1 - MS\&P) + (R_2 - M) * (RS\&P_2 - MS\&P) + \dots + (R_N - M) * (RS\&P_N - MS\&P)) / ((RS\&P_1 - MS\&P)^2 + (RS\&P_2 - MS\&P)^2 + \dots + (RS\&P_N - MS\&P)^2)$. Where R_i = program return for i-th month, M = mean return of the program, $RS\&P_i$ = return of S&P program for i-th month, MS&P = mean of return of S&P program	

2.3 Seasonal Analysis

When conducting a time series analysis, it is often important to recognise time series patterns such as trends, seasonality and cycles in the data (Hyndman & Athanasopoulos, 2018). In this model, we are particularly interested in identifying the annual seasonality of the selected product.

The following illustrates what I have done with regards to the Seasonal Analysis:

1. Understand how Seasonal Analysis works.
2. Create a function in Python to retrieve relevant data from SQLite.
3. Create a main function in Python to contain algorithm of Seasonal Analysis.
4. Create an API for Seasonality Analysis in Python and test the API on Postman.
5. Once testing is done, commit my scripts using SourceTree to let the Backend Developer merge them with the company's main folder.
6. Communicate with Data Visualisation Intern and modify the code accordingly. The Data Visualisation Intern will then present the data in a chart for users. The final product is as shown in Figure 1.

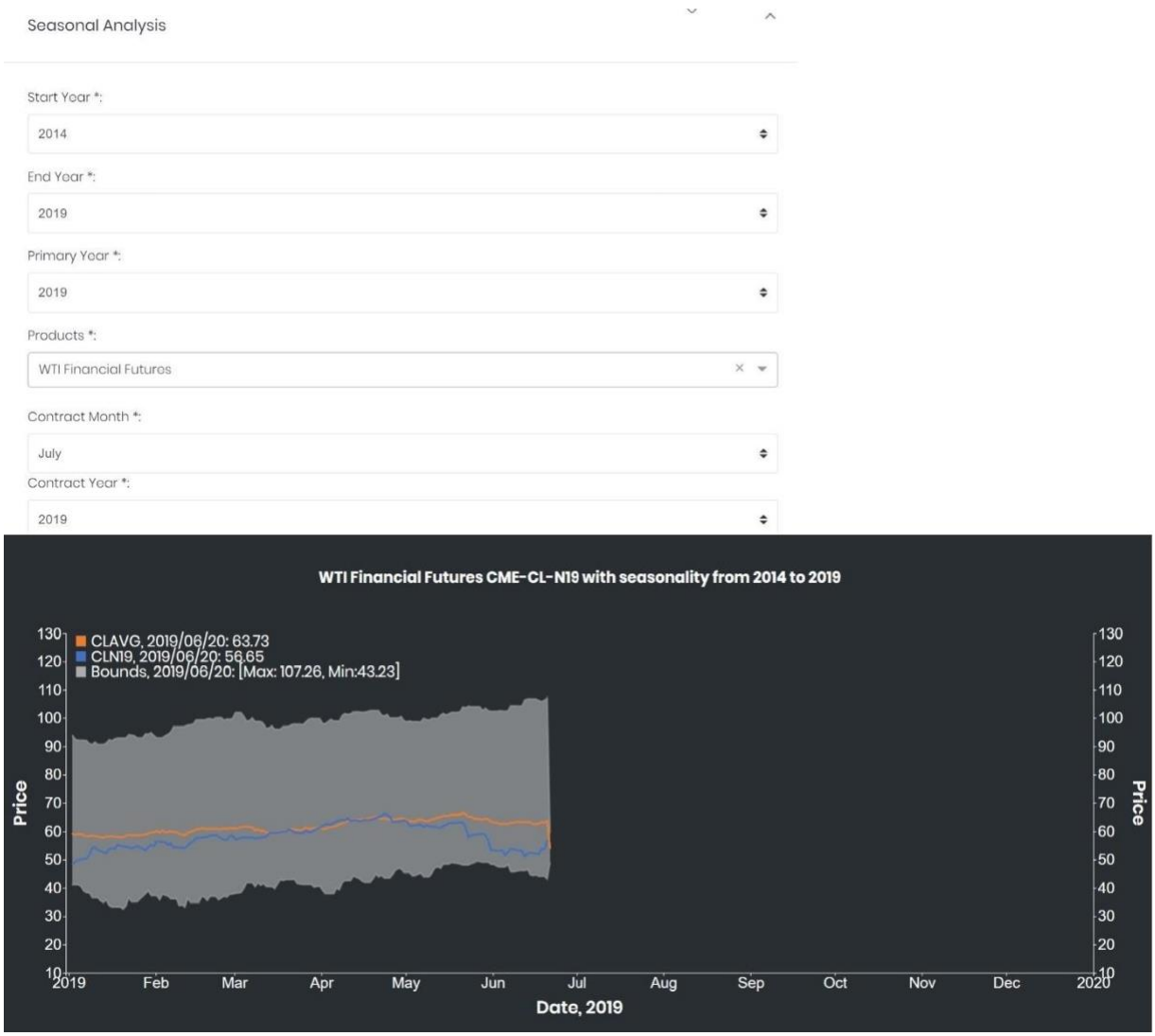


Figure 1: Seasonal Analysis Application

Figure 1 illustrates the seasonal analysis of July contracts of WTI Financial Futures over the period of the year 2014 to the year 2019. The orange line represents the average prices of CLN14, CLN15, …, and CLN19 between January and July. The blue line represents the settlement prices of CLN19, while the lower bound and upper bound represents the minimum and maximum respectively.

2.4 Forward Curve Table

The forward curve table consists of 3 main columns, Settlement Prices, Cumulative Percentage Change and Percentage Change, on each date. Table 4 shows how the values of each cell are derived. The 'Product Code' column is a list of all contracts available for a particular product on each date sorted by the contract's expiry date, with the contract on the first row being the first one to expire. Figure 2 shows the Forward Curve Table Application after the model has been successfully implemented into the company's dashboard.

Table 4: Forward Curve Table Formula

	date 1			date 2		
Product Code	**Settlement Prices**	**Cumulative Percentage Change**	**Percentage Change**	**Settlement Prices**	**Cumulative Percentage Change**	**Percentage Change**
SBN19	P1	(P1 - P1) / P1 = 0	NaN	P1*	(P1* - P1*) / P1* = 0	NaN
SBV19	P2	((P2 - P1) / P1) x 100%	((P2 - P1) / P1) x 100%	P2*	((P2* - P1*) / P1*) x 100%	((P2* - P1*) / P1*) x 100%
SBH20	P3	((P3 - P1) / P1) x100%	((P3-P2) / P2) x100%	P3*	((P3* - P1*) / P1*) x 100%	((P3*-P2*) / P2*) x 100%
SBK20	P4	((P4 - P1) / P1) x 100%	((P4-P3) / P3) x 100%	P4*	((P4* - P1*) / P1*) x 100%	((P4*-P3*) / P3*) x 100%
...	...	...	...	...	...	...

The following illustrates what I have done with regards to Forward Curve Table:

1. Understand how the Forward Curve Table works.
2. Create a function in Python to retrieve data from SQLite.
3. Create a main function in python to contain the algorithm of the Forward Curve Table.
4. Create an API in Python and test the API on Postman.
5. Once testing is done, commit my scripts using SourceTree to let the Back-end Developer merge them with the company's main folder.
6. Communicate with Data Visualisation Intern and modify the code accordingly. The Data Visualisation Intern will then present the data in a chart for users.

Forward Curve Table

Primary Date *: (Take latest available date)

2019-07-04

Products *:

Steel Rebar

No. of Days *:

2

Comments:

Enter comments if any

SHFE-RB Steel Rebar						
	2019-07-03			2019-07-04		
Product Code	Settlement Price	Change%	Cummulative Change%	Settlement Price	Change%	Cummulative Change%
RBN19	3825	-	0	3850	-	0
RBQ19	4110	7.45	7.45	4092	6.29	6.29
RBU19	4088	-0.54	6.88	4072	-0.49	5.77
RBV19	4042	-1.13	5.67	4033	-0.96	4.75
RBX19	3873	-4.18	1.25	3892	-3.5	1.09
RBZ19	3829	-1.14	0.1	3830	-1.59	-0.52
RBF20	3762	-1.75	-1.65	3762	-1.78	-2.29
RBG20	3589	-4.6	-6.17	3584	-4.73	-6.91
RBH20	3591	0.06	-6.12	3584	0	-6.91
RBJ20	3644	1.48	-4.73	3625	1.14	-5.84
RBK20	3594	-1.37	-6.04	3589	-0.99	-6.78
RBM20	3563	-0.86	-6.85	3557	-0.89	-7.61

Figure 2: Forward Curve Table Application

2.5 Forward Curve

The Forward Curve represents the relationship between the price of a future contract which can be concluded today and the time to maturity of that future contract (Geman, 2011). The Forward Curve Graph Application consists of two components; the times series of a particular futures contract and its corresponding forward curves on specific dates as input by the user.

The following illustrates what I have done with regards to the Forward Curve:

1. Understand how the Forward Curve works.
2. Create a function in Python to retrieve data from SQLite. For instance, obtain the historical prices of a specific contract and the prices of futures contracts related to that contract.
3. Create a main function in python to contain the algorithm of the Forward Curve
4. Create an API in Python and test the API on Postman.
5. Once testing is done, commit my scripts using SourceTree to let the Back-end Developer merge them with the company's main folder.
6. Communicate with the Data Visualisation Intern and modify the code accordingly. The Data Visualisation Intern will then present the data in a chart for users as shown in Figure 3.

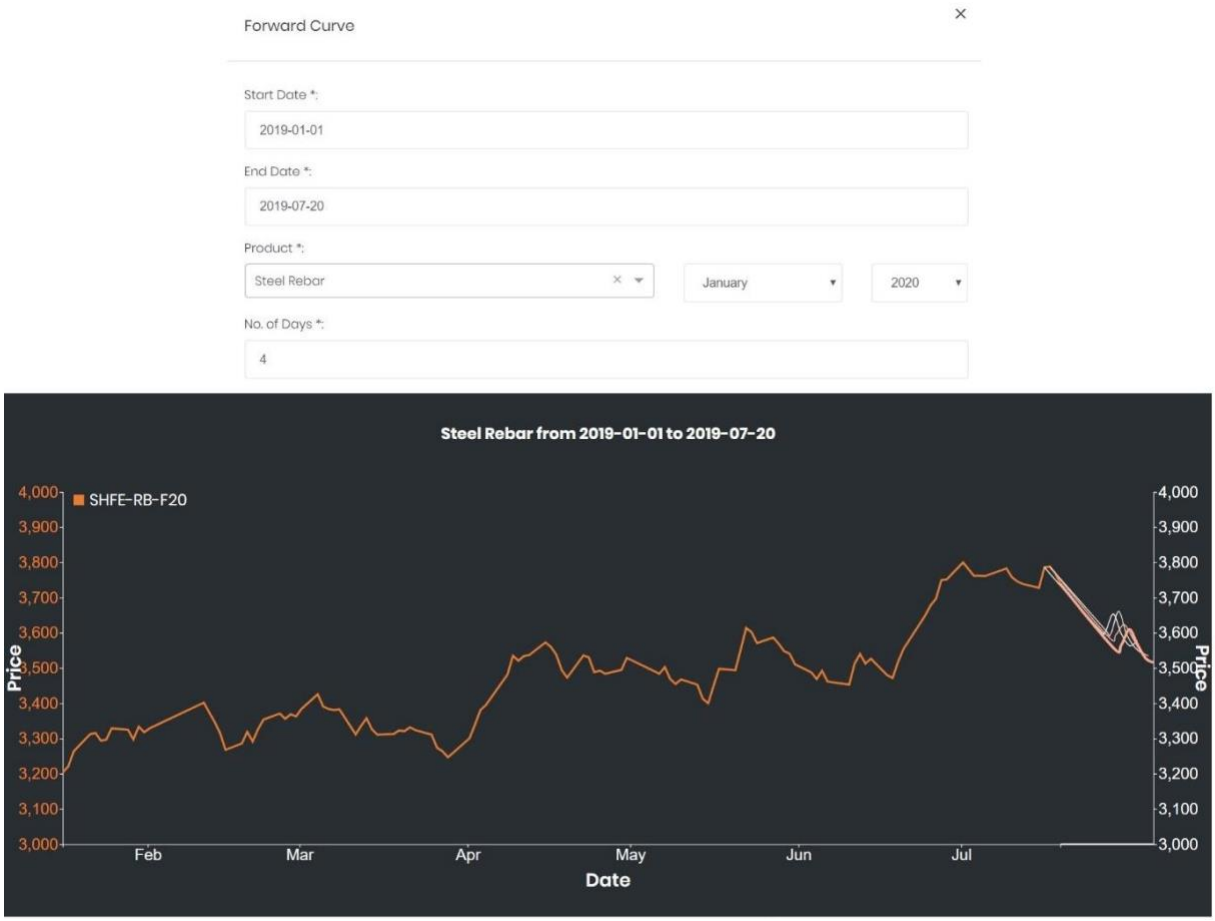
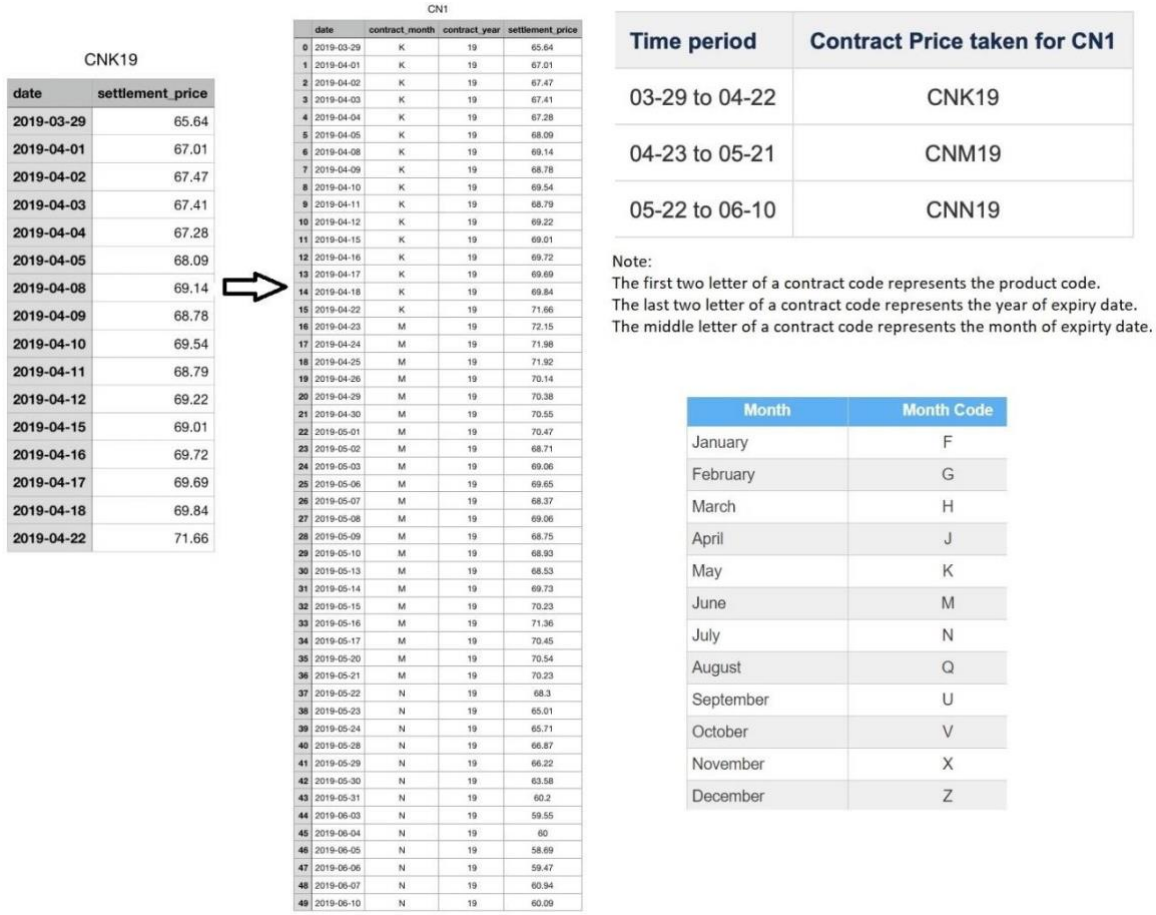


Figure 3: Forward Curve Application

The orange line in the chart as shown in Figure 3 represents the historical prices of January contract of Steel Rebar, RBF20. There are 4 Forward Curves and each Forward Curve would start on the time series of the input product. The next few subsequent points of the Forward Curve would be the settlement prices of the next available contracts on that specific date. For example, on that specific date, the next few points would be the settlement prices of RBG20, RBH20, RBJ20, …, respectively. On a particular date, the ending point of a forward curve should be the settlement price of the contract with expiry date furthest from that date.

2.6 Continuous Contract

As some Futures contracts have a very short life span, it is often not sufficient for analysts to get valuable insights from a time-series analysis. The minimum number of data points for any time-series analysis is recommended to be more than 50 (Senter, n.d.). Thus, this data transformation involves splicing contracts together to create a long-term chart (Hill, 2011). For example, the Crude Oil Houston WTI has product code CN and has several contracts with different expiry dates on each date. The Crude Oil Houston WTI contract expiring on July 2019 has contract code CNK19. The duration of the CNK19 contract is about 1 month, which is equivalent to approximately 16 data points. Analysts who wish to conduct time-series analysis on Crude Oil Houston may use CN1 instead of CNK19. CN1 would take the price of the nearest contract on each date, while CN2 would take the price of the second nearest contract on each date.



CNK19

date	settlement_price
2019-03-29	65.64
2019-04-01	67.01
2019-04-02	67.47
2019-04-03	67.41
2019-04-04	67.28
2019-04-05	68.09
2019-04-08	69.14
2019-04-09	68.78
2019-04-10	69.54
2019-04-11	68.79
2019-04-12	69.22
2019-04-15	69.01
2019-04-16	69.72
2019-04-17	69.69
2019-04-18	69.84
2019-04-22	71.66

CN1

	date	contract_month	contract_year	settlement_price
0	2019-03-29	K	19	65.64
1	2019-04-01	K	19	67.01
2	2019-04-02	K	19	67.47
3	2019-04-03	K	19	67.41
4	2019-04-04	K	19	67.28
5	2019-04-05	K	19	68.09
6	2019-04-08	K	19	69.14
7	2019-04-09	K	19	68.78
8	2019-04-10	K	19	69.54
9	2019-04-11	K	19	68.79
10	2019-04-12	K	19	69.22
11	2019-04-15	K	19	69.01
12	2019-04-16	K	19	69.72
13	2019-04-17	K	19	69.69
14	2019-04-18	K	19	69.84
15	2019-04-22	K	19	71.66
16	2019-04-23	M	19	72.15
17	2019-04-24	M	19	71.98
18	2019-04-25	M	19	71.92
19	2019-04-26	M	19	70.14
20	2019-04-29	M	19	70.38
21	2019-04-30	M	19	70.55
22	2019-05-01	M	19	70.47
23	2019-05-02	M	19	68.71
24	2019-05-03	M	19	69.06
25	2019-05-06	M	19	69.65
26	2019-05-07	M	19	68.37
27	2019-05-08	M	19	69.06
28	2019-05-09	M	19	68.75
29	2019-05-10	M	19	68.93
30	2019-05-13	M	19	68.53
31	2019-05-14	M	19	69.73
32	2019-05-15	M	19	70.23
33	2019-05-16	M	19	71.36
34	2019-05-17	M	19	70.45
35	2019-05-20	M	19	70.54
36	2019-05-21	M	19	70.23
37	2019-05-22	N	19	68.3
38	2019-05-23	N	19	65.01
39	2019-05-24	N	19	65.71
40	2019-05-28	N	19	66.87
41	2019-05-29	N	19	66.22
42	2019-05-30	N	19	63.58
43	2019-05-31	N	19	60.2
44	2019-06-03	N	19	59.55
45	2019-06-04	N	19	60
46	2019-06-05	N	19	58.69
47	2019-06-06	N	19	59.47
48	2019-06-07	N	19	60.94
49	2019-06-10	N	19	60.09

Time period	Contract Price taken for CN1
03-29 to 04-22	CNK19
04-23 to 05-21	CNM19
05-22 to 06-10	CNN19

Note:
The first two letter of a contract code represents the product code.
The last two letter of a contract code represents the year of expiry date.
The middle letter of a contract code represents the month of expirty date.

Month	Month Code
January	F
February	G
March	H
April	J
May	K
June	M
July	N
August	Q
September	U
October	V
November	X
December	Z

Figure 4: Continuous Contract Transformation

The following is what I have done with regards to continuous contract:

1. Understand how Continuous Contract works.
2. Create a function in Python to retrieve data from SQLite. For instance, obtain the historical prices of a specific contract and the prices of futures contracts related to that contract.
3. Create a main function in Python to contain algorithm of Continuous contract
4. Create an API in Python for Frontend.

2.7 Position Calculation Model

The purpose of the Position Calculation Model is to calculate the unrealised and realised profit and loss for a specific portfolio at the current date. Thus, it is imperative to identify which positions are open and which are closed on the current date. This model takes into account 5 different instruments, namely, outright futures, SWAP, physical transactions (Fixed mode, Average mode and Floating mode). As the process of closing each aforementioned instrument is different, I am tasked to create an algorithm to close the position based on their instrument type. Table 5 outlines the algorithm of the Position Calculation Model while Figure 5 shows the final output.

Table 5: Algorithm of Position Calculation Model

Steps:
1. Convert Trades to Open Positions
1.1 Get all sub nodes related to the portfolio input by the user
1.2 Obtain trades of the relevant portfolios from the database
1.3 Convert each trade derived in Step 1.2 to open positions
2. Translate Positions
2.1 Sort out premiums and other_cost, and translate them to a position object
2.2 Translate Paper Transactions
2.2.1 Outright Futures
2.2.2 SWAP
2.3 Translate Physical Transactions to Paper
2.3.1 Fixed
2.3.2 Floating
2.3.3 Average
3. Group and Close Positions
3.1 Group the positions with up to 6 variables (depending on the grouping method).
3.2 Close positions
3.2.1 FIFO
3.2.2 Average
4. Format Result
4.1 Evaluate the total P/L in closed positions and total unrealised in open positions.
4.2 Check the hedge_status
4.3 Show output

Viewing Portfolio

Portfolio Information

Portfolio Name: Total

P/L Algo: Average

Aggregation Typo: (Select one) Combination

Realised P/L: USD69,471.73

Unrealised P/L: -USD71,975.81

Total P/L: -USD2,504.07

Status: Unhedged Closed

Open Positions

Contract Month	Contract Year	Buy/Sell	Size	Unit	Average Price	Settlement Price	Settlement Date	Unrealised P/L
October	19	buy	5	Lot	2329	2301	2019-07-23	-1400
August	19	buy	50	Lot	13957.5	13455	2019-07-23	-25125
September	19	buy	2	Lot	2780.25	3008	2019-07-23	22775
September	19	buy	8	Lot	2665	2.7	2019-07-23	7075
September	19	buy	4	Lot	523	487.25	2019-07-23	-7150
December	19	buy	3	Lot	65.89	63.73	2019-07-23	-3240
September	19	sell	3	Lot	113	102.3	2019-07-23	12037.5
October	19	buy	90	Lot	12.743	11.98	2019-07-23	-76948.30769

8 total

Closed Positions

Product Name	Contract Code	Valuation Date	Contract Month	Contract Year	Size	Bought Price	Sold Price	Realised P/L
SGX FTSE China A50 Index Futures	CN		May	19	30	12737.5	12735	-75
Sugar No. 11 Futures	SB		May	19	6	12.78	12.8	134.4
SGX MSCI Singapore Free Index Futures	SG		May	19	5	361.35	361.4	18.326
Chicago SRW Wheat Futures	W		July	19	4	437.25	438.75	-100
Cotton No. 2 Futures	CT		July	19	3	76.42	76.34	-120
SGX FTSE China A50 Index Futures	CN		July	19	50	13206.5	13962.5	37800
Crude Oil Futures	CL		December	19	3	53.493	54.169999999999995	2030
SGX FTSE China A50 Index Futures	CN		May	19	35	13572.5	12566.42857142857	-35212.5

20 total

Trade

Trade ID	Trade Date	Instrument	Trade Type	Exchange	Contract Code	Valuation Date	Contract Year	Contract Month
t15947	2019-03-29	paper	none	ICEUS	KC		2019	July
t29897	2019-03-29	paper	none	ICEUS	SB		2019	May
t48476	2019-03-28	paper	none	ICEUS	KC		2019	July
t308	2019-02-26	paper	none	CME	HO		2019	May
t29280	2019-04-01	paper	none	CME	HG		2019	May
t28394	2019-03-04	paper	none	ICEUS	SB		2019	May
t75569	2019-03-04	paper	none	ICEUS	SB		2019	May
t18753	2019-03-12	paper	none	CME	HG		2019	May
t23880	2019-03-18	paper	none	CME	HG		2019	May
t26735	2019-03-18	paper	none	ICEUS	SB		2019	May
t1071	2019-02-19	paper	none	CME	HG		2019	May

Figure 5: Position Calculation Model

2.8 Data Transformation

Data transformations could be used for proper statistical analysis of data from the finance discipline (Ozdemir, 2016). This model would transform raw data into output desired based on the formulae defined on Table 7.

The following illustrates what I have done with regards to the Data Transformation model:

1. Understand how Data Transformation works.
2. Create a function in Python to retrieve data from the SQL database. Identify the frequency type of data retrieved and set its corresponding interval as shown in Table 6.
3. Create a function to perform the 9 data transformations as shown in Table 7.

Table 6: Frequency of Data and their Interval

Frequency(type of data)	Interval within a year
daily	260
weekly	52
monthly	12
quarterly	4

Table 7: Data Transformations Formulae

Units	Formula
1. Change	p_t - p_{t-1}
2. Change from Year ago	p_t - $p_{t-interval}$
3. Percent Change	$((\frac{p_t}{p_{t-1}}) - 1) \times 100$
4. Percent Change from Year ago	$((\frac{p_t}{p_{t-interval}}) - 1) \times 100$
5. Compounded Annual Rate of Change	$((\frac{p_t}{p_{t-1}})^{interval} - 1) \times 100$
6. Continuously Compounded Rate of Change	$(\ln(p_t) - ln(p_{t-1})) \times 100$
7. Continuously Compounded Annual Rate of Change	$(\ln(p_t) - ln(p_{t-1})) \times 100 \times interval$
8. Index Scale Value to 100 for chosen date	$\frac{p_t}{p_{cd}} \times 100$
9. Natural Log	$\ln(p_t)$

2.9 US Recession Prediction Probability

It is common for economists to employ mathematical models to predict economic events. To estimate the probability of a recession, the Probit model, with yield curve spread as an explanatory variable, could be used (Estrella & Mishkin, 1996). The Linear Probability Model (LPM) is an alternative to the Probit model. Both LPM and the Probit model have binary dependent variables. However, there are two problems with LPM. Firstly, under the LPM, the predicted probability could be less than zero or more than one. This violates the basic probability law. Secondly, the partial effect of explanatory variable is constant. To overcome these two problems with LPM, the Probit model is used (Dustan , 2010). Under the Probit model, the probability of a US recession is given by $P(y=1|x) = G(\beta x + \alpha)$, where G() is the function of CDF of standard normal, x is the value of spread, α is a constant and β is a coefficient of the value of spread.

The following illustrates what I have done with regards to the US Recession Prediction Probability model:

1. Understand how the Probit model works.
2. Create a function in Python to retrieve the value of spread (10-Year Treasury Constant Maturity Minus 3-Month Treasury Constant Maturity) from the company's database.
3. Create an algorithm in Python to calculate the predicted probability given the value of spread.
4. Create an API for the Front-end Team to call for the predicted probability output. The chart in Figure 6 was used to give the Front-end Team an idea of what the chart should look like.

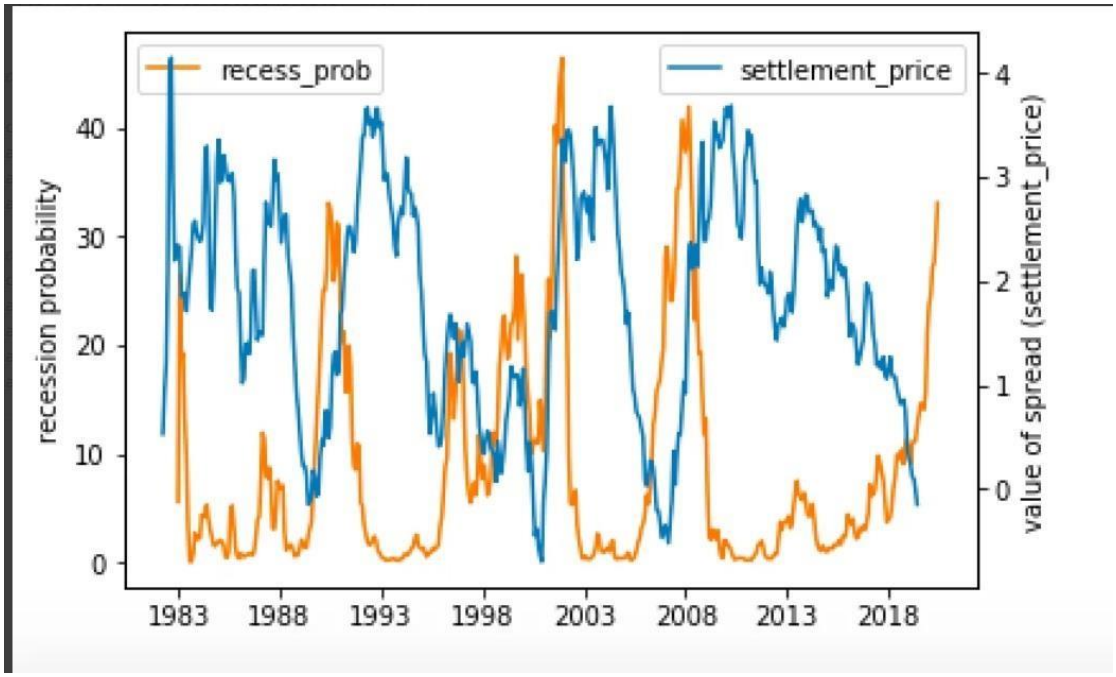


Figure 6: US Recession Probability in Python

2.10 Weekly Reports

The following are weekly reports that have been prepared, as required by the organization. Some confidential information has been omitted.

Week 1 Report (13 May – 17 May)

Done:
• Set up and explore necessary softwares//libraries (Sourcetree, Pycharm & Anoconda, Postman, DB browser for SQL, Macs Terminal(pips), panda and numpy)
• Familiarize with company's project: - Project structure - Understand programming flow of various service codes - Learn basic SQL (CRUD-create read update delete)
• Learn how RESTful APIs and Flask framework works: - Flask tutorials regarding Routing, Variable Rules, URL binding, HTTP methods (GET-retrieve, POST-create, PUT-update, DELETE-remove), database and Blueprints - Create an API route and its relevant service and conduct API testing on Postman
Doing:
• Debug Correlation Matrix
To Be Done:
• Create Seasonal Analysis model

Week 2 Report (21 May – 24 May)

Done:
• Create Seasonality Analysis model
• Test Seasonality Analysis Model with market data (Success)
• Seasonality Analysis Model documentation
Doing:
• Discuss with Frontend team on how to present the data/output in charts
To Be Done:
• Forward Curve model

Week 3 Report (27 May – 31 May)

Done:
• Seasonality Model: - Frontend managed to plot the output - Modify Seasonality Model - Completed Seasonality Model Documentation
• Forward Curve Model (for futures contracts): - Understand how Forward Curve works (meeting with Director and research) - Created a function to retrieve relevant data from the database - Created algorithm to implement forward curve model - Implement the aforementioned functions with current API - Test output using Postman
• Forward Curve Table: - Understand how Forward Curve Table works (meeting with Director and research) - Similar to Forward Curve model but percentage changes in futures contract will be displayed - Created a function to retrieve relevant data from the database - Created algorithm to implement forward curve model - Implement the aforementioned functions with current API - Test output using Postman
Doing:
• Communicating with Data Visualisation Intern to implement Forward Curve Table and Forward Curve
To Be Done:
• Forward Curve Table and Forward Curve Documentation

Week 4 Report (3 June to 7 June)

Done:
• Forward Curve Table and Forward Curve Documentation
• With the help of Frontend/Data Visualization Intern, Forward Curve Table and Forward Curve successfully implemented on company portal's Dashboard
• Debug VaR (Value at Risk) module: -Understand how Values at Risk works. -Had true value of series ambiguous error in existing model, managed to found the error and resolve the bug.
• Continuous Contract - Understand how continuous contract/ data transformation works (meeting with investor and research) - Created function to retrieve data from database (algorithm)
Doing:
• Incorporate Continuous Contract into Time Series App
To be Done:
• Implement Continuous Contract to other existing applications on company portal

Week 5 Report (10 June to 14 June)

Done:
• Modify APIs of all existing models
• Incorporate Continuous Contract into Time Series and Seasonality Applications
• Modify Forward Curve model, remove one unecessary input parameter.
• Help out with User Guide documentation
Doing:
• Help out with User Guide documentation
To Be Done:
• Position Calculation Model (briefing on next Monday)

Week 6 Report (17 June to 21 June)

Done:
• Position Calculation Model Documentation (Almost done)
• Debug Cross Correlation model
Doing:
• Editing Position Calculation Model Documentation after clarifying with investor
To Be Done:
• Start coding on Position Calculation Model

Week 7 Report (24 June – 28 June)

Done:
• Position Calculation Model: - Finished algorithm for type of instruments available in database - SWAP test for paper transactions (Success) - Average Method for Physical Instruments (Success) - Position Calculation Model Documentation
• Read up on London Metal Exchange, especially on products' valuation dates
• Data Transformation: - Derive formula and come out with algorithm for a total of 9 data transformations and for data of different time frame (i.e. daily, monthly, quarterly etc)
Doing:
• Create API for Data Transformation and relevant scripts to pass to frontend
To Be Done:
• Position Calculation Model Algorithm for instruments with valuation dates (i.e. expiry dates)

Week 8 Report (1 July – 5 July)

Done:
• Data Transformation: - Data Transformation Documentation - Create API for Data Transformation
• Position Calculation Model: Algorithm to close instruments with valuation dates
• Briefing on Commodity Trading Advisor (CTA) Performance Model
Doing:
• Standardise the computation of indicators such as realized P/L, unrealized P/L and VaR for CTA performance model
To Be Done:
• CTA performance model

Week 9 Report (8 July – 12 July)

Done:
• US Recession Prediction model: - Understand how the recession prediction probability is derived - Create Algorithm to calculate recession prediction probability based on value of spread using probit model - Create API for US recession prediction model - US Recession Prediction Documentation
• Standardise the computation of indicators such as realized P/L, unrealized P/L and VaR for CTA performance model
Doing:
• Finalising US Recession Prediction model
To Be Done:
• CTA performance model

Week 10 Report (15 July – 19 July)

Done:
• Finalised US Recession Prediction model
• Finalised CTA performance modelling
Doing:
• CTA performance modelling documentation and other documentation
To Be Done:
• Finalise CTA performance indicators calculation

3. Summary

3.1 Summary of Project

This internship took place in Summer 2019. As a Math and Economics student, this internship was a great opportunity to see how these two disciplines interact. Furthermore, I was exposed to less familiar topics like information technology and financial terms. During these 3 months, I have successfully crafted 8 quantitative models in total. Upon the completion of these projects, I find modules like Data Analysis, Programming Skills and Econometric Modelling to be very useful and relevant to my job scope as a Financial Portfolio Risk Modelling intern.

Table 8: Relevant Skill Set

Projects	Programming skills (MH1401, MH1402, MH2401)	Statistics (MH2500, MH3500)	Data Analysis With Computer (MH3511)	Econometrics (HE2005, HE3021, HE3022)	Macro-Economics (HE1002, HE2002, HE4002)	Accounting/Finance Knowledge from my Diploma in Accountancy
Seasonality Analysis	✓		✓	✓		✓
Forward Curve	✓		✓			✓
Forward Curve Table	✓		✓			✓
Data Transformation	✓	✓	✓	✓	✓	✓
Continuous Contract	✓	✓	✓	✓		✓
Position Calculation Model	✓		✓			✓
US Recession Prediction Probit Model	✓	✓	✓	✓	✓	✓
CTA Performance Model	✓	✓	✓			✓

3.2 Reflection

I am glad that I am beginning to understand how these businesses operate day-to-day as it allows me to observe and participate in a variety of real-world experiences. Most importantly, it has been very useful in helping me decide what career route I desire to take.

I feel that this internship has been a very meaningful one to me, and at the same time, it has challenged me in many ways. It placed me outside my usual "comfort zone", as I had to deal with problems that I had never come across before. Therefore, I am glad that I was able to adapt and resolve them with proper guidance from my team. I am grateful that I was able to work for MAF Intelligence Pte Ltd. as the management and co-workers are very helpful and understanding. They made it very clear what was expected of me and were accessible to answer and resolve any doubts I had. I am also impressed by the documentation practice in this company. I have learnt that the documentation processes is highly important for self-evaluation. As I was preparing the documentation, it helped me review my own work and ensure that what I had done was logically correct. If anything went wrong with the model, the documentation aided in debugging bugs, be it a logic bug or a syntax bug. In addition, the weekly reports generated have helped me keep track of my work in progress and ensure that I delivered consistent work. I hope this practice will continue.

All in all, I enjoyed every aspect of this internship and my experience from this internship has imbued me with the skills and confidence I need to become a Data Analyst / Data Scientist. I take pride in what I have contributed so far as I know the models that I have designed will be used by many other users or professionals in the future.

Appendix A: List of Relevant Modules

List of Relevant Modules
MH1401 ALGORITHMS AND COMPUTING I
MH1402 ALGORITHMS AND COMPUTING II
MH2401 ALGORITHMS & COMPUTING III
MH2500 PROBABILITY & INTRODUCTION TO STATISTICS
MH3500 STATISTICS
MH3511 DATA ANALYSIS WITH COMPUTER
HE2005 PRINCIPLES OF ECONOMETRICS
HE3021 INTERMEDIATE ECONOMETRICS
HE3022 ECONOMETRIC MODELLING & FORECASTING
HE1002 MACROECONOMIC PRINCIPLES
HE2002 INTERMEDIATE MACROECONOMICS
HE4002 ADVANCED MACROECONOMICS

References

Chen, J. (15 June, 2019). Retrieved from Investopedia: https://www.investopedia.com/terms/c/cta.asp

Dustan , A. (2010). *EEP/IAS 118 - Section Handout 13.* Retrieved from https://are.berkeley.edu/courses/EEP118/fall2010/section/13/Section%2013%20Handout%20Solved.pdf

Estrella, A., & Mishkin, F. S. (1996). The Yield Curve as a Predictor of U.S. Recessions. *SSRN Electronic Journal*.

Geman, H. (June, 2011). Retrieved from Commodity Forward Curves :The Old and the New: http://www.oxford-man.ox.ac.uk/sites/default/files/events/Oxford%2C%20June%202011.pdf

Hill, A. (10 06, 2011). Retrieved from StockCharts: https://stockcharts.com/articles/mailbag/2011/06/what-is-a-continuous-futures-contract-gold.html

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: principles and practice*. Heathmont, Vic.: OTexts.

Ozdemir, O. (2016). A Comparison Study of Data Transformation Methods to Achieve. *International Journal of Mathematical and Computational Methods*, 382-383.

Senter, A. (n.d.). *Time Series Analysis*. Retrieved from http://userwww.sfsu.edu/efc/classes/biol710/timeseries/timeseries1.htm