**Feedback for HW #2 Pandas**

Daphene:

**Empty list initialized correctly, header skipped.**

avg\_change variable calculates every iteration, better practise would be to calculate once the loop is excecuted**. This is important when dealing with larger and more complicated data sets and you should always try to make your code as efficient as possible.**

**Dependencies called correctly initially. You might want to split the actual opening of the file on the next cell. Try to use ' i ' when using it within loops as it is common convention and is widely recognized in python. Naming conventions for variables and are on point, good job!**

**Commit messages can be more precise**

**Although this is an easy code to read, it is always a good practise to include comments strategically.**

Overall good job!!

David Savage:

**Empty list initialized correctly, header skipped.**

Good job on experimenting with dictionaries, although the dictionaries were not necessary, could have just created the list. Try not to create multiple loops but try to use less loops which increases overall readability of the code. All the calculations could have been accomodated within 2 loops,

Line 19: is too volumous, try to condense the code.

The code is correct logically, but you havent followed DRY priciple. Use of comments and variable names Is clear. Github commit messages are also clear.

Carlos:

Read of data: Implemented correctly with skipping of header files. **Empty list initialized correctly. Dependencies imported correctly.**

Great job on writing concise code.

Variable naming isnt very clear. pl\_dif\_acum looks very cryptic. Variable named as ‘a’ isnt helpful for fellow developers who want to understand the code. Also variables have been initialized and then randomly added with values which should be avoided. If statement indentation not followed.

Github commit messages are clear.

Alexandra:

Clear and concise code. Implemented correctly with skipping of header files. **Empty list initialized correctly. Dependencies imported correctly. Variable naming and commenting is also very clear. Maybe you could have used less comments, but you can continue doing so if you wish. For loops although not compulsory you might want to use i variable as it is common usage. Overall great submission !!**

**Amit:**

Clear and concise code. Implemented correctly with skipping of header files. **Empty list initialized correctly. Dependencies imported correctly. Commenting is also very clear.**

**Commenting can be improved upon, single letter naming isnt very helpful. Do name variables as profit instead of ‘P’. DRY principle followed. Commenting on github can be better with naming of the project being worked upon. Lastly try to keep characters in one line of code lesser than 80. Overall great job across Reading/Write and Manipulation.**

**Abu:**

Clear and concise code. Implemented correctly with skipping of header files. **Empty list initialized correctly. Dependencies imported correctly. Commenting is also very clear.**

Good use of native functions. There is some redundancy in the snippet.

Mounth = row[0]

amount = int(row[1])

# Append the row

moun.append(Mounth)

line\_num += 1

ChangeL.append(amount)

This the only unclear part of the code.

You can create one variable called month = append(row[i]).

DRY principle followed. Github comments should mention the project name.

Amar:

Good job on using pandas to read and manipulate the data. SInce pandas is a very powerful library, one can access a lot of information in lesser code using native functions. If you are using pandas, then make sure you are using simple code which makes for better readability.

For better readability

banks\_df['Change'] = 0 #initialize the column

for i in range(0,len(banks\_df)-1):

banks\_df['Change'].iloc[i+1] = banks\_df['Profit/Losses'][i+1] - banks\_df['Profit/Losses'][i]

Similarly:

banks\_df.loc[(banks\_df['Change'] == banks\_df['Change'].max())]

banks\_df.loc[(banks\_df['Change'] == banks\_df['Change'].min())]

For finding the min and max for pnl.

The average calculation is fine.

Try to use variables in small cases and try to use \_ if you want to make them more specific.

Ali:

**Read/Write operations:**

**Empty list initialized correctly, header skipped. Reading and Writing logic is clear and concise. The calculations for min, max are volumous.**

**Average change calculations are incorrect. Logic for the above mentioned calcualtions can be done in a more readable way which will help easier collaboration with other developers. You can also try to use more native functions.**

**Eg**

**sum([list])**

**len(list)**

**Github commit messages do not mention the HW name.**

**Coding conventions: Use of variable name is good. You can make more use of DRY principle.**

**Bailey:**

**Read/Manipulate: 60/70**

**Read/write operations are concise. Empty list initialized correctly, header skipped. Dependencies imported correctly. Code logic can be implemented using lesser codes and wo the use of a explicit counter. Adds to the complexity of the code and reduces the readablility. average\_change has been calculated correctly but it is missing from the output**

**Coding Convetions: 8/10**

**All coding conventions followed. Just that code is a bit lengthy which reduces readablity.**

**Deployment:8/10**

**Github commit messages can be more precise which identify the project being worked upon**

**Commenting: 8/10**

**Although commenting is a good practise, make sure to keep it precise. Commenting should act more of a placeholder of code rather than be used to explain code logic. Also make sure comments arent volumonous which improves the overall readabliity.**

**Feed back for HW #4 Whale analysis & Portfolio analysis.**

**Best submission:**

**Richard**

**Veldurai**

**C:/Users/user/Desktop/FinTech-Lesson-Plans/02-Homework/04-Pandas/Instructions/Starter\_Code/Resources**

**Sinthushan – 14, 15, 9, 15, 9, 9, 9 = 80**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly**

**The pct­\_change for S&P500 should have been done after sorting of date index, since this hasn’t been done, all the calculations are incorrect.**

**The rolling 21 day chart for** BERKSHIRE HATHAWAY INC also seems to be different than what I am getting, again this is because the date time hasn’t been sorted.Make sure of DRY principle.

**I did a quick sort with ‘sp500\_history\_df.sort\_index(inplace = True)’ and most of the graphs seem to fixed.**

**The box plot showing the risk is supposed to be plotted for the daily returns and not the cumulative returns. All Sharpe ratio calculations are a bit off. The S&P sharpe ratio is a positive value of 0.648267**

**Coding Conventions/Formating**

**Recomandattions: you can infer\_date within the read\_csv command itself**

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

**Overalll good job, just make sure you are careful with time series analysis. Also refer to charts and figures to cross reference the accuracy of results of your code.**

**Total score 80 / 100**

**Veldurai: 20, 18, 15, 10, 9, 10**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly**

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

**Only thing which was missing was the box plot interpretation in regards to the spread seen.**

**Very neat and concise code, interpretation of the results are also on point. Great job!**

**Total 97/100**

**Nitesh: 20, 17, 1, 10, 10, 9, 10**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly**

**\*Interpretation of cumulative returns charts is missing, which porfolio perfomed better or worse than the index S&P500**

**\*Box plot interpretation is missing.**

**\*Beta graph is to be a rolling function and thus a chart.**

**\*Custom portfolio stocks not picked, expectation is you would be selecting 1 to 3 other stocks other than the ones which have already been provided, usign the google function and uploading the csv on github.**

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

**Overall good job! All calculations are on point, just need to provide more your understanding of the charts as per the rubrick.**

**Total score 91/100**

**Feedback for Richa!**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly**

**\*Interpretation of cumulative returns charts is missing, which porfolio perfomed better or worse than the index S&P500**

**\*Box plot interpretation is missing.**

**\*For sharpe ratios, interpretation is correct, however the for loop isnt required and is incorrect.**

**Custom portfolio: selection of stock HSBC with uploaded sighted.**

**Ananlysis of Beta for the custom stock selection is good, one small error**

elif beta\_custom < 0:

print(f"Custom Portfolio is negatively uncorrelated to the market")

**It should be negatively correlated!**

**Documentation/Comments: One of the interpretation was just put as a comment,**

#The STD of TIGER GLOBAL MANAGEMENT….

**This should be in markdown.**

**Deployment/Submission: Submission was not done through command line. No commit messages. Please refrain from using manual upload, for the homework submissions.**

**For some reason the kernel keeps dying when I try to run the code on local computer.**

**Feedback for Richard: 20, 20, 15, 10, 10, 9, 10**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly**

**Excellent interpretation of quantitative analysis, just make to put the findings/interpretations next below the cumulative chart for better readability. Enjoyed reading your code. Great job!**

**EWA chart is correct, just a suggestion: you can combine the 2 lines of code.**

One thing missing was using Google function to select own stock to build custom portfolio.

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

**Feedback for Shuran 20,15, 12, 15, 10, 9, 10 total = 91/100**

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctl**

**\*Interpretation of cumulative returns charts is missing, which portfolio performed better or worse than the index S&P500**

**\*Box plot interpretation is missing.**

**ewm needs to plotted with std() and not mean()**

**Sharpe Ration:**

On the basis of this performance metric, do our algo strategies outperform both 'the market' and the whales? Type your answer here:

Your answer is missing

I liked that you used own stocks to calculate the custom portfolio. Good analyis. Interpretation is also missing though.

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

Feedback for Musonda: 20, 15, 12, 15, 9, 9, 10 = 90

**All dependencies have been called, Pandas has been used to read the file. Nulls have been removed correctly\*, numeric values formatted correctly. Index for S&P500 has been sorted twice, which is not required.**

**\*Interpretation of cumulative returns charts is missing, which portfolio performed better or worse than the index S&P500**

**\*Box plot interpretation is missing.**

\*Sharpe ratio interpretation is missing.

Custom Porfolio: Good job with the custom portfolio analysis.

Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.

Feedback for Younwoo: 20, 15, 10, 7, 9, 8

# YOUR CODE HERE – This should not be there in the final submission.

\*Error seen in the submission, this needs to be addressed.

Daily returns of S&P500 to be calculated after sorting of index, daily returns calculatation are incorrect.

Beta was to be calculated for just one of the portfolios

**\*Interpretation of cumulative returns charts is missing, which portfolio performed better or worse than the index S&P500**

**\*Box plot interpretation is missing.**

\*Sharpe ratio interpretation is missing.

**Deployment/Submission: Do specify which HW assignment # and name which the upload addresses.**

**Coding Conventions/Formating: Code is lengthy at places.**

Homework 5: API –

Simulations, account summary, portfolio planner, financial report.

Accessing Plaid API

The plaid API consists of public key, a client key and your password. First use the public key and then use your two secure keys to create a public token. This public token is then use to grab a transactions\_response.

public token is used to create a public key and  a request id. together they can be said to be creation\_token\_response. You take this creation\_token\_response and pass [ "public\_token" ] within

creation\_token\_response[ "public\_token" ]. This is used to as a parameter within the *exchange* function of the **client.Item.public\_token.***exchange*(creation\_tkn\_response[ "public\_token" ]).

Now this abomination for lack of better word is called exchange\_response which is made up of

access\_token, item\_id and request\_id. We consider only the access\_token part of abomination and call it accessed\_token.

Use the accessed\_token to retrieve either

client.Accounts.get(accessed\_token)

client.Transactions.get(accessed\_token, start\_date, end\_date )

client.Income.get(accessed\_token)

## Budget Analysis \_/20

## Retirement Planner \_/20

## Retirement Analysis \_/15

## Financial Report \_/15

## Coding Conventions/Formating \_/10

## Deployment/Submission \_/10

## Documentation/Comments \_/10

## Optional Challenge — Early Retirement \_/10

**Feed back for Zeldi:**

## Budget Analysis 20/20

Make sure that you do print any keys. For the accounts response, you can make use of pretty\_print\_response function just like you did for the transactions\_response

to get a more intutive picture of the PLAID json response.

Transactions fetched for last 90 days. All income values are correct

## Retirement Planner 20/20

Monte Carlo is correct. Correct weights considered for the appropriate tickers. Confidence interval is correct. Histogram and the vertical lines have been plotted.

## Retirement Analysis 15/15

Retire analysis has been completed satisfactorily, withdrawal rate calculations correct.

## Financial Report 13/15

The formatting for the percentiles is a bit of. Apart from that everything looks good.

## Coding Conventions/Formating 10/10

## Deployment/Submission 10/10

## Documentation/Comments 10/10

## Optional Challenge — Early Retirement 10/10

**Feedback for Mihir**

## Budget Analysis 18/20

Projected Income before tax not considered, For the accounts response, you can make use of pretty\_print\_response function to get a more intutive picture of the PLAID json response.

## Retirement Planner 20/20

Retirement Income calculations correct

## Retirement Analysis 15/15

Retirement analysis correct.

## Financial Report 0/15

No report was submitted

## Coding Conventions/Formating 9/10

Avoid naming variables like p10, for the most part conventions were followed.

## Deployment/Submission 10/10

Good comments on github.

## Documentation/Comments: 10/10

## Optional Challenge — Early Retirement 0/10

Optional not attempted

## Total: 82/100

**Feedback for Manjari**

## Budget Analysis 20/20

Make sure that you do print any keys on Github. Calculations look good.

## Retirement Planner 19/20

Retirement Income calculations correct. As per the question a histogram of the distribution was to be plotted, however you plotted the probablity distribution. It is not incorrect, however do provide the answers to the questions asked in the starter code.

## Retirement Analysis 15/15

All calculations look good.

## Financial Report 15/15

## Coding Conventions/Formating 9/10

Could use better variables, and make them as distinct as possible. portfolio\_cumulative\_returns could have been replaced with monte\_carlo, too many ‘cumulative’ named variables.

## Deployment/Submission 10/10

## Documentation/Comments 10/10

## Optional Challenge — Early Retirement 8/10

The cumulative returns for 10,50 and 95 percentile is not correct.

projected\_returns = portfolio\_cumulative\_returns.quantile(q=[0.05, 0.50, 0.95],axis =1).T

projected\_returns.plot(title = "Portfolio Performance over time", figsize = [12, 6])

You missed to pass the axis=1 parameter.

**Feedback for Jessica:**

## Budget Analysis 20/20

Make sure that you do print any keys on Github. Calculations look good.

## Retirement Planner 15/20

The weights have been reversed, should be 40 for AGG and 60 for SPY.

For the question # Select the last row for the cumulative returns (cumulative returns at 30 years)

You have considered: # Select the last row for the cumulative returns (cumulative returns at 20 years)

The calculations will be significantly off if you discount 30% of the timeframe. You can use ending\_cumulative\_returns = portfolio\_cumulative\_returns.iloc[-1, :]

## Retirement Analysis 10/15

Again since the ending\_cumulative\_returns have been calculated at 20 years, the calculations are incorrect. The code logic is correct though.

## Financial Report 15/15

Report looks good!

## Coding Conventions/Formating 8/10

You have declared monte\_carlo as dataframe but you didn’t use it in the actual mode. Avoid declaring empty variables.

## Deployment/Submission 10/10

## Documentation/Comments 7/10

Please write more comments in complicated part of the code, eg during the Monte carlo calculations, will prevent errors.

## Optional Challenge — Early Retirement 0/10

Not attempted

## Total 90/100

**Feedback for Joseph:**

## Budget Analysis 15/20

Date hasn’t been converted to date\_time object, projected early income considered is incorrect.

## Retirement Planner 15/20

The **agg\_closing\_price,spy\_closing\_price, avg\_daily\_return\_spy, std\_dev\_daily\_return\_spy** arent calculated correctly, you should be selecting a single value, please review the code.

Weights should be the other way around.

simulated\_agg\_price = simulated\_agg\_prices[-1] \* (1 + np.random.normal(avg\_daily\_return\_agg, std\_dev\_daily\_return\_agg))

The above also was incorrect. Please review.

Please check your code, also do sanity checks at each stage which programming.

## Retirement Analysis 0/15

## Financial Report 0/15

## Coding Conventions/Formating 7/10

Monte\_carlo dataframe has been created but not used. Readabality of the code could be better.

## Deployment/Submission 7/10

Files uploaded, not pushed through git bash.

## Documentation/Comments 7/10

Comment do not correlate to the code with the simulations.

**Feedback for Elizabeth:**

## Budget Analysis 20/20

Make sure that you do print any keys.

## Retirement Planner 20/20

You arent required to plot the probablity distribution, just the frequency distribution.

## Retirement Analysis 15/15

## Financial Report 15/15

The financial report looks good!

## Coding Conventions/Formating 5/10

expected\_cum\_returns = ending\_cumulative\_returns.quantile(q=[0.10, 0.5, 0.90])

Value\_10th\_percentile = expected\_10th\_percentile\_return

Avoid capitals and numbers in declaring variables,

Both variables sound the same. Avoid such declarations.

## Deployment/Submission 10/10

## Documentation/Comments 5/10

Please maintain consistent coding conventions wrt to comments, at times code is on next line, at times there space.

## Optional Challenge — Early Retirement 0/10

Early retirement analysis not done!

**Feedback for IAN**

## Budget Analysis 18/20

The projected income considered is incorrect.

## Retirement Planner 20/20

Make sure that you do print any keys.

## Retirement Analysis 15/15

## Financial Report 15/15

Projected income is wrong, however not reducing the score here as it was done in first part.

## Coding Conventions/Formating 10/10

## Deployment/Submission 10/10

## Documentation/Comments 10/10

## Optional Challenge — Early Retirement 10/10

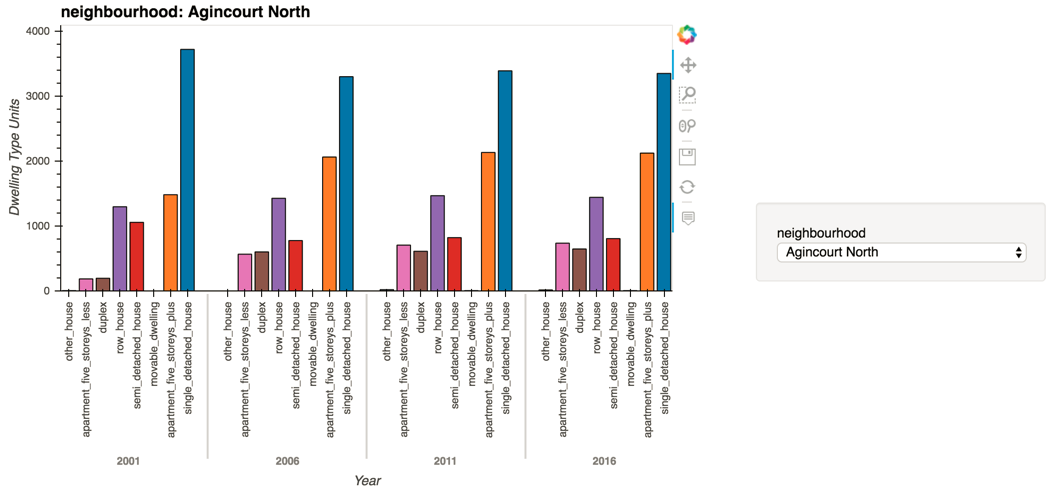
Score: 100/100

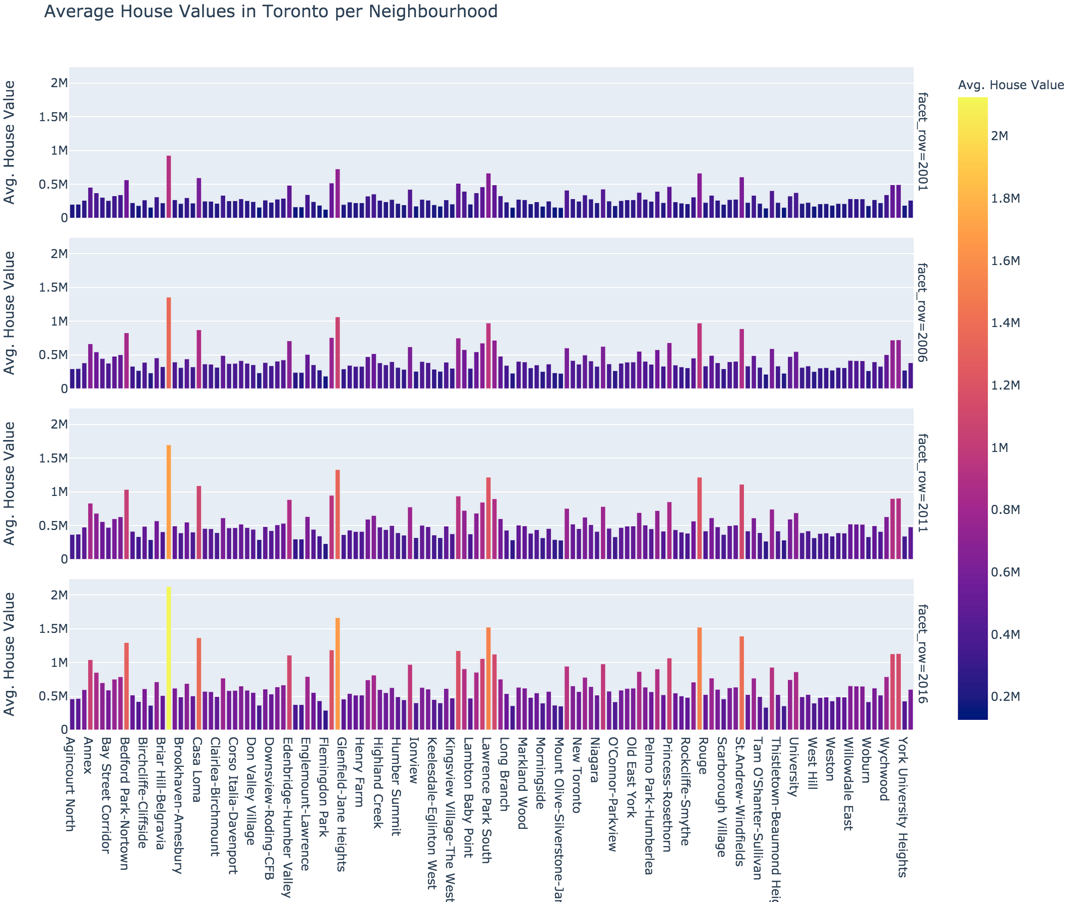
**PYVIZ**

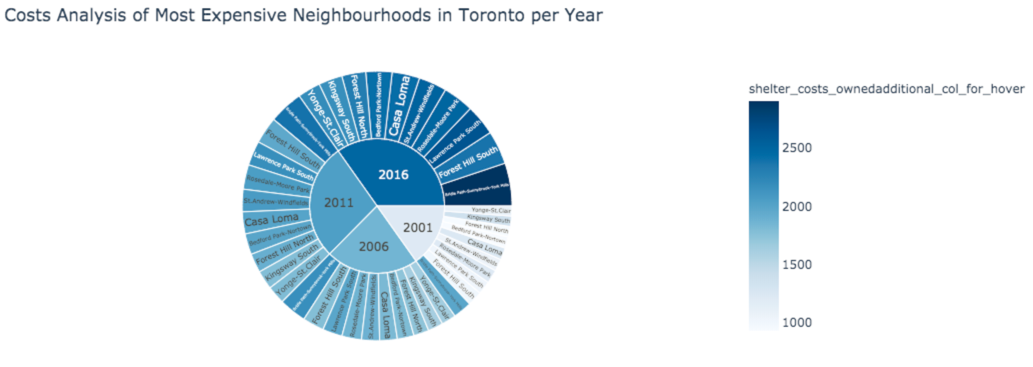
For this homework, learnt some interesting visualization for starburst, interative hvplots with drop down, panel integration for dashboarding.

Facet plots

1







Feedback for Pyviz homework:

Feedback for Daphene.

## Rental Analysis: 20/20

Number of dwelling types/year calculated correctly. Bar chart visualization correct.

## Average House per year: 20/20

Average house calculations correct. Drop down visualization not visible on github, but visible when I ran on local computer.

Monthly shelter costs for owned& rented house calculated and visualized correctly over line chart.

## No of dwelling types/year:15/15

Number of dwelling types/year grouped by year done.

Dwelling type visualized, mean house value for each neighbourhood calculated. Expensive houses idenfied. Bar chart isnt visible, same ran on local computer.

\*\*Neighbourhood Map: 15/15

Interactive map as per requirements. Behaves as expected

\*\*Coding conventions: 10/10

For Dashboard: Make sure to

import hvplot.pandas

after

import panel as pn

pn.extension("plotly")

This will not throw a message.

\*\*Documentation

Code logic is concise.

\*\*Cost Analysis

Average house calculations correct. Drop down visualization not visible on github, but visible when I ran on local computer.

Optional challenge: Excellent logic for creating the sunburst.

**Feedback for David Savage:**

## Rental Analysis: 20/20

All calculations seem correct.

## Average House per year: 20/20

For average house/year, instead of dropping multiple columns to create new dataframe, you can just declare

df[[‘col\_1’, ‘col\_2 ’]]

its more readable and takes less lines of code.

## No of dwelling types/year: 13/15

Most expensive neighbourhoods identified correctly, however the bar chart doesn’t match to the one in the starter code, please use it a reference to guide your logic. The neighbourhoods had to be averaged across the years to arrive at the correct average figures.

## Neighbourhood Map 15/15

Behaves as expected.

## Coding conventions: 7/10

Refrain from using generic terms like ‘array’, instead use variables like

top\_neighbourhood\_list

For Dashboard: Make sure to

import hvplot.pandas

after

import panel as pn

pn.extension("plotly")

This will not throw a message.

## Documentation 10/10

## Cost Analysis – Optional Challenge:

Row Facet plot of the all TO neighbourhoods is a bit hard to read as the chart is default, maybe try formatting the size(dimensions) to make it bigger and so that the lines don’t overlap.

Sunburst chart is good!

## Rental Analysis:

## Average House per year

## Number of Dwelling Types per Year

## Neighbourhood Map

## Coding conventions

## Deployment

## Documentation  
## Cost Analysis – Optional Challenge

**Feedback for Amar:**

## Rental Analysis: 20/20

All calculations look good. Thanks for pointing out the discrepancy in the question and the sample answer.

## Average House per year: 20/20

For the hvplot, ylimits by default are dynamic, realize you set the limits to compare the plots on the same scale but higher scale for lower range of house value reduces the detailing in the individual graphs.

## Number of Dwelling Types per Year.15/15

Charts and calculations are correct.

## Neighbourhood Map 15/15

## Coding conventions 6/10

As per conventions variables need to be lower case with the use of snake case if required. Bar chart not following a standard of space/line between each year function call. Imports not called at the beginning.

You can use spaces between operators for better readablity.

For the dashboard, call

pn.extension("plotly") and then

import hvplot.pandas

to avoid the warning/

In dashboards, you can use 2 separate variable names for better readability instead of calling file\_path.

## Deployment 10/10

## Documentation 10/10

## Cost Analysis – Optional Challenge

Calculations and plot are correct, however the formatting is a bit off and you can use fig.update\_layout to set the width and height for better readabality of the graphs.

In the top 10 neighbourhoods for the sunburst, what is this used for:

g = df\_agg['average\_house\_value'].groupby(level=0, group\_keys=False)

**Feedback for Amit:**

## Rental Analysis: 18/20

Shelter cost code seems to throw an error, the code should be

avg\_monthly\_shelter\_costs= to\_data[["shelter\_costs\_owned", "shelter\_costs\_rented" ]]

average\_cost\_dwellings= avg\_monthly\_shelter\_costs.groupby([to\_data.index]).mean()

Just an oversight, errors were resolved when the code was fixed.

## Average House per year

## Number of Dwelling Types per Year

For the top 10 neighbourhood/year, it was actually top 10 neighbourhood all time, and plot those values across the 4 years. What you have considered are the 4 years top neighbourhoods. However since the question wasn’t communicated clearly, no marks have been deducted.

## Neighbourhood Map 10/15

The dashboard isn’t complete, please review what needs to be changed/added.

## Coding conventions 7/10

Variables like n\_2001 are very generic and hence you can be a bit more specific for better readability.

You can use spaces before and after operators( = , + ) for better readablity.

Import statements can all be called together in a single cell.

For the dashboard, call

pn.extension("plotly") and then

import hvplot.pandas

to avoid the warning/

you can be more specific with regards to the filepath, if you are calling in the same cell eg data\_path and coordinates\_path, instead of calling file\_path as good practise.

## Deployment 9/10

Commit messages are descriptive of what edit/changes have been made.

## Documentation 8/10

Entine line chart function code has been commented out. This isnt good practise. If it is not serving any purpose please remove it from the final submission.

## Cost Analysis – Optional Challenge

Calculations and plot are correct, however the formatting is a bit off and you can use fig.update\_layout to set the width and height for better readabality of the graphs.

**Feedback for Carlos:**

## Rental Analysis: 20/20

In rental analysis, you printed your keys. Please refrain from printing any API keys on github. Calculations and plots look good.

## Average House per year 20/20

Calculations seem correct.

## Number of Dwelling Types per Year 15/15

For the top 10 neighbourhood/year, it was actually top 10 neighbourhood all time, and plot those values across the 4 years. What you have considered are the 4 years top neighbourhoods. However since the question wasn’t communicated clearly, no marks have been deducted.

## Neighbourhood Map 11/15

Map looks good, for dashboard the tabs have been created, however they havent been passed into the dashboard and hence dashboard isnt generating. Please review how to create an instance of the dashboard.

## Coding conventions 9/10

Also as per the conventions you can use spaces before and after operators for better readabliity,

Do follow standard conventions of leaving blank lines between 2 function calls, case in point for sliced\_data for 2001 to 2016(dashboard.ipynb) file.

## Deployment 9/10

Commit messages not descriptive.

## Documentation 9/10

You can remove sample code format from the final submission.

# def create\_bar\_chart(data, title, xlabel, ylabel, color):

You can remove the code after the dashboard as this confuses about the use of that part of code.

## Cost Analysis – Optional Challenge 27/30

Calculations and plot are correct, however the formatting is a bit off and you can use fig.update\_layout to set the width and height for better readabality of the graphs.

# Score 93/100

FB For Alex:

## Rental Analysis: 20/20

All calculations look good.

## Average House per year 20/20

Plots and calculations look good.

## Number of Dwelling Types per Year 15/15

## Neighbourhood Map 8/10

Neighbourhood map looks good, the dashboard doesn’t display all the modules within the panes.

## Coding conventions 10/10

Great job on following standard spacing and indenting throughout the code. Very neat submission.

Another thing which is a standard as per convention is to have spaces before and after the use of each operator ( = , + , - ) etc.

## Deployment

Github commit messages are very descriptive. Good job!

## Documentation 10/10

## Cost Analysis – Optional Challenge

The top 10 neighbourhoods are meant to be for top 10, you considered top 5. Think you missed you putting in 10 for the groupby.head().

FB for ABU

## Rental Analysis:20/20

All calculations look good.

## Average House per year 20/20

Plots and calculation seem correct.

## Number of Dwelling Types per Year 15/15

## Neighbourhood Map 13/15

The map looks good, however the dashboard doesn’t display all the modules.

## Coding conventions 10/10

Most of the coding conventions followed, you can also use spaces before and after use of operators for better readablity.

For the dashboard, call

pn.extension("plotly") and then

import hvplot.pandas

to avoid the warning/

## Deployment: 8/10

Github messages can be more descriptive

## Documentation: 9/10

You can remove # YOUR CODE HERE! From the final submission

## Cost Analysis – Optional Challenge 25/30

Calculations and plot are correct, however the formatting is a bit off and you can use fig.update\_layout to set the width and height for better readabality of the graphs.

In the top 10 neighbourhoods for the sunburst, what is this used for:

For the sunburst of top 10, you considered the top 10 for each year, but actually it is top 10 averaged out and then plot them over 4 years. However no marks deducted as the question wasn’t communicated correctly.

**Need to talk about compensating, from sbyte. Dave if we need more people. Increase the amount.**

**Siju,**

Feedback for Bailey:

**## Rental Analysis 18/20**  
• Number of dwelling types calculated correctly with the Bar chart visualization

**Average Monthly Shelter Cost TO Per Year**• Average monthly shetler owned vs rented is correct.  
• The line chart x axis isnt showing up as year, not very intitutive. However the trend is correct.

**## Average House Value Price Per Year 20/20**  
Calculation and plot is correct.   
  
Average Prices By Neighborhood calculations and drop down plot sighted.

**## No of dwelling types/year 15/15**

Dwelling types per year visualized in each neighbourhood, drop down selector works.

Mean house value for each neighbourhood calculated

Mean house values sorted for top 10 neighbourhoods on average.

Results plotted as bar chart.

**## Neighborhood Map 13/15**  
• Interactive map with the hover data is correct. I don’t see how avg\_neigh\_list +='lat','lon' is used in the code, if not required you can remove it. Doesn’t impact the map generated but readablity is reduced. Also every single time map is generated, avg\_neigh\_list keeps adding the lat and lon to the list.   
  
Dashboard  
• Dashboard doesn’t seem to display properly, However when I made some changes, I was able to display most of the charts.

**## Coding conventions 8/10**

avg\_neigh\_list declared, but they was no use for it, make sure

For Dashboard: Make sure to

import hvplot.pandas

after

import panel as pn

pn.extension("plotly")

This will not throw a message.

**## Deployment 10/10**

**## Documentation 10/10  
## Cost Analysis – Optional Challenge 30/30**

**Bar chart correct.**

**Sun burst correct.**

**Wasn’t able to see the plots being plotted, with some tweaking, I was able to display.**

**## Rental Analysis**  
• Number of dwelling types per year calculated.   
• Bar chart visualization of rental analysis results.   
  
**## Average Monthly Shelter Cost TO Per Year**• Average monthly shetler owned vs rented.  
• Monthly shelter cost per year visualized in line chart.

**## Average House Value Price Per Year**  
• average\_house\_value calculated for each year.  
• average\_house\_value per year visualized in line chart.   
  
Average Prices By Neighborhood  
• Data grouped by year and neighborhood and average calculated per sqft.  
• Average sales per sqft per year visualized with a neighborhood dropdown selector.

**## No of dwelling types/year**

Dwelling types per year visualized in each neighbourhood

Mean house value for each neighbourhood calculated

Mean house values sorted for top 10 neighbourhoods on average.

Results plotted as bar chart.

**## Neighborhood Map**  
• Interactive map with average neighborhood prices per sqft.  
  
Dashboard  
• Interactive dashboard encompassing each functional visualization.

**## Coding conventions**

**## Deployment**

**## Documentation  
## Cost Analysis – Optional Challenge**

**FOR Group 1**

Presentation Criteria

* Be 8–10 minutes in length (check with the instructor for the official presentation time).
  + 4 points - presentation between 8 to 10 minutes
  + 3 points - presentation 1 minutes over or under
  + 2 points - presentation 2 minutes over or under
  + 1 points - presentation 3 minutes over or under
  + 0 points - presentation 4 minutes over or under
* Describe the project's core message or hypothesis.
  + 4 points - extremely clear core message and/or hypothesis
  + 3 points - very clear core message and/or hypothesis
  + 2 points - somewhat clear core message and/or hypothesis
  + 1 points - not clear core message and/or hypothesis
  + 0 - no core message or hypothesis included
* Describe the questions the group found of interest, and why.
  + 8 points - extremely clear description of questions, reasons for choosing and summary of findings
  + 6 points - very clear description of questions, reasons for choosing and summary of findings
  + 4 points - somewhat clear description of questions, reasons for choosing and summary of findings
  + 2 points - not clear description of questions, reasons for choosing and summary of findings
  + 0 points - none included
* Summarize how, and where, the data was found to answer these questions.
  + 4 points - extremely clear explanation of where data was found and how they found it
  + 3 points - very clear explanation of where data was found
  + 2 points - somewhat clear explanation of where data was found
  + 1 points - not clear explanation of where data was found
  + 0 points - none included

* Describe the data exploration and cleanup process (accompanied by Jupyter Notebook).
  + 20 points
    - Concise and engaging explanation of how data exploration was done, and the data cleaning steps required to prepare the data for analysis.
    - Includes an interesting insight while exploring the data that the group did not anticipate
    - Includes an explanation of the top two major challenges in data cleaning.
  + 15 points
    - Concise and engaging explanation of how data exploration was done, and the data cleaning steps required to prepare the data for analysis.
    - Includes an interesting insight while exploring the data that the group did not anticipate
    - Missing explanation of the top two major challenges in data cleaning.
  + 10 points
    - Somewhat clear explanation of how data exploration was done, and the data cleaning steps required to prepare the data for analysis.
    - Missing an interesting insight while exploring the data that the group did not anticipate
    - Missing explanation of the top two major challenges in data cleaning.
  + 5 points
    - Vague and confusing explanation of how data exploration and data cleaning steps.
    - Missing an interesting insight while exploring the data that the group did not anticipate
    - Missing explanation of the top two major challenges in data cleaning.
  + 0 - none included
* Describe the analysis process (accompanied by Jupyter Notebook).
  + 20 points
    - Concise and engaging explanation of how analysis was conducted on the data.
    - Includes an explanation of the top two major challenges in data analysis.
  + 15 points
    - Concise and engaging explanation of how analysis was conducted on the data.
    - Missing an explanation of the top two major challenges in data analysis.
  + 10 points
    - Somewhat clear explanation of how analysis was conducted on the data.
    - Missing an explanation of the top two major challenges in data analysis.
  + 5 points
    - Vague and confusing explanation of how analysis was conducted on the data.
    - Missing an explanation of the top two major challenges in data analysis.
  + 0 points - None included
* Summarize the conclusions. This should include a numerical summary (i.e., what data did the analysis produce), as well as visualizations of that summary (plots of the final analysis data).
  + 20 points
    - Thorough explanation of each visualization and clearly communicates takeaways and conclusions from each visualization.
    - Visualizations are extremely readable and can be easily interpreted by the audience.
  + 15 points
    - Thorough explanation of each visualization and clearly communicates takeaways and conclusions from each visualization.
    - Visualizations are somewhat readable and can be somewhat interpreted by the audience.
  + 10 points
    - Acceptable explanation of each visualization and communicates takeaways and conclusions from some visualizations.
    - Visualizations are somewhat readable and can be somewhat interpreted by the audience.
  + 5 points
    - Unclear explanation of each visualization.
    - Non readable visualizations that are difficult to interpret by the audience
  + 0 points - None included
* Discuss the implications of the findings.
  + 10 points - Clearly describes implications of findings on industry or on personal lives.
  + 8 points - Somewhat clear implications of findings on industry or on personal lives.
  + 4 points - Unclear implications of findings on industry or on personal lives.
  + 0 points - None included

**Presentation will be marked out of 90, however the technical and presentation marks will be weighed *equally* for calculating your final project grade (i.e. we will take the average of the percent grades).**

Technical Criteria

* README file
  + 15 points
    - Clear and concise ReadMe file. Contents of ReadMe file are properly formatted in markdown language. 0 markdown errors.
    - Purpose of the repo clearly articulated to the audience.
    - Summary of major findings included, along with relevant plots (PNGs from your Jupyter Lab Notebooks).
  + 12 points
    - Clear and concise ReadMe file. Contents of ReadMe file are properly formatted in markdown language.
    - Purpose of the repo clearly articulated to the audience.
    - Summary of major findings included, but missing relevant plots (PNGs from your Jupyter Lab Notebooks) **or** few markdown formatting errors (1-4)
  + 8 points
    - Somewhat clear ReadMe File.
    - Purpose of repo somewhat articulated to the audience
    - Some of the major findings included, no plots
    - Many markdown formatting errors (5+)
  + 4 points
    - Unclear ReadMe file
    - No clear purpose of repo
    - Missing major findings, with no splots
    - Many markdown formatting errors (5+)
  + 0 points
    - None included
* Data exploration and cleanup Jupyter Lab Notebook
  + 40 points
    - Across both notebooks (Data Exploration / Cleanup and Final Data Analysis
      * Readable and DRY (don’t repeat yourself) code, along with intuitive variable names
      * Code is clearly documented with comments describing each step
      * 0 to 1 errors in code
      * At least 1 new Python Library **not** covered in class
    - For Final Data Analysis notebook
      * 6 to 8 visualizations
        + Use of PyViz, Plotly Express (minimum of once, HvPlot minimum of once)
        + 2 visualizations per question
      * Visualizations aggregated into Panel dashboard
  + 30 points
    - Across both notebooks (Data Exploration / Cleanup and Final Data Analysis
      * Readable and DRY (don’t repeat yourself) code, along with ***somewhat*** intuitive variable names
      * Code is clearly documented with ***some*** missing comments
      * ***2 to 5*** errors in code
      * ***Missing*** 1 new Python Library **not** covered in class
    - For Final Data Analysis notebook
      * 6 to 8 visualizations
        + Use of PyViz, Plotly Express (minimum of once, HvPlot minimum of once)
        + 2 visualizations per question
      * Visualizations aggregated into Panel dashboard
  + 20 points
    - Across both notebooks (Data Exploration / Cleanup and Final Data Analysis
      * Readable and DRY (don’t repeat yourself) code, along with ***somewhat*** intuitive variable names
      * Code is clearly documented with ***some*** missing comments
      * ***6 to 10*** errors in code
      * ***Missing*** 1 new Python Library **not** covered in class
    - For Final Data Analysis notebook
      * 4 to 5 visualizations
        + Use of PyViz, Plotly Express (minimum of once, HvPlot minimum of once)
        + 2 visualizations per question
      * Visualizations aggregated into Panel dashboard
  + 10 points
    - Across both notebooks (Data Exploration / Cleanup and Final Data Analysis
      * Unreadable code
      * Code is **not** documented at all
      * ***11*** errors + in code
      * ***Missing*** 1 new Python Library **not** covered in class
    - For Final Data Analysis notebook
      * Under 4 visualizations
        + Use of PyViz, Plotly Express (minimum of once, HvPlot minimum of once)
        + 2 visualizations per question
  + 0 points
    - None included

**Notebooks will be marked out of 55, however the technical and presentation marks will be weighed *equally* for calculating your final project grade (i.e. we will take the average of the percent grades).**

Group 4 draft

Presentation Criteria 4+2+8+4+15+15+20+10 = 78/90

* Be 8–10 minutes in length (check with the instructor for the official presentation time).
  + **4 points** - presentation between 8 to 10 minutes

* Describe the project's core message or hypothesis.
  + **2 points -** somewhat clear core message and/or hypothesis

Hypothesis : Black swan events affect different sectors of economies in different ways. 2 points. Hypothesis not very clear, as the assumption here is that the Covid-19 is a black swan event. No source/references provided which points to Covid 19 being a black swan. Also what constitutes a black swan event or any qualifiers not discussed.

* Describe the questions the group found of interest, and why.
  + **8 points** - extremely clear description of questions, reasons for choosing and summary of findings

Motivation 2008 crisis

Questions: 1.Can we make money in this scenario

2. Can monte carlo accommodate black swan events  
3. Best strategy for investors, and if new investors should invest.

* Summarize how, and where, the data was found to answer these questions.
  + **4 points** - Extremely clear explanation of where data was found and how they found it

Ecom – AMZN , Banking - RBC and Retail – walmart.ALPACA dataset: Jan 2008 to June 2020

* Describe the data exploration and cleanup process (accompanied by Jupyter Notebook).

**15 -** No interesting insight encountered during the data exploration and clean up. No challenges faced during the data cleaning process.

* Describe the analysis process (accompanied by Jupyter Notebook).

**15 -** One challenge addressed, data preparation.

Analysis with the help of fundamental analysis in regards the B/S and income statements gives us a ground view of the selected companies and it is complementary to the monte carlo analysis driven insights that you have presented. Bonus for using Financial Modelling Prep API.

* Summarize the conclusions. This should include a numerical summary (i.e., what data did the analysis produce), as well as visualizations of that summary (plots of the final analysis data).

**20 points** 4 conclusions presented, Monte Carlo are good accurate and even accomodating crisis events(black swans).

* Discuss the implications of the findings.

**10 points** as it points out the impact to investor who have had market exposure prior to Covid-19, and also for new investors looking to invest in the market.

**Presentation will be marked out of 90, however the technical and presentation marks will be weighed *equally* for calculating your final project grade (i.e. we will take the average of the percent grades).**

Technical Criteria: 48/55

* README file – 13 points

No markdown errors sighted, also Purpose of repo wasn’t outlined in the report.

Comments:

As per the rubric you were suppose to split into **Data Exploration / Cleanup** & **Final Data Analysis.** Understand the data was standard and hence didn’t need much of cleaning.

The notebook presented contains a lot of code and the individual pieces could have been on separate notebook, viz the fundamental analysis.

For data exploration, plot for the closing prices sighted. Volatility, Beta and Sharpe Ratio calculated however no commentary/insights. Interesting plot about comparing historical vs covid 19 volatility, same should have been outlined in the report with possible reasons for the differences in volatility or the lack of it.

Variable names should be more context specific. data\_monte\_carlo\_df is the slice of the original dataframe and should be named accordingly, monte carlo doesn’t seem appropriate.

It was good that you plotted the heatmap for the stocks and the index. However without any summary it doesn’t solve any purpose. Make sure that there is a one line conclusion for any plots you have.

For Beta: Earlier you mentioned about calculating **Beta**, but in the actual calculation its commented as **Covariance of Selected Stock Returns vs. S&P Returns.** The calculations are fine, but you need to communicate effectifvely to all your stakeholders.

Great job creating a function for the monte carlo sims and calling when required.

For some reason I wasn’t able to plot the fundamental analysis plots/dashboard. I got an error while running the data\_clean for RY, AMZN, WT data clean functions.

Good job with using new library Financial Model Prep. Dashboard not visible 35 points. DRY used.

**Overall great job, you could have integrated more of the fundamental analysis to justify/explain the trend you found for the Retail/Banking and Ecom stocks and their performance pre and during Covid-19.**

**Group 4: Bailey, Daphene, Elizabeth, Veldurai**

# Presentation Criteria 78/90

Hypothesis : Black swan events affect different sectors of economies in different ways. The hypothesis is not very clear, as the assumption here is that the Covid-19 itself is a black swan event. No source/references provided which points to Covid 19 being a black swan. Also what constitutes a black swan event or any qualifiers not discussed.

Analysis with the help of fundamental analysis in regards the B/S and income statements gives us a ground view of the selected companies and it is complementary to the monte carlo analysis driven insights that you have presented. Bonus for using Financial Modelling Prep API.

# Technical Criteria: 48/55

README file – 13/15

No markdown errors sighted,

Purpose of repo wasn’t outlined in the report. The README should ideally act as a pointer to the different notebooks and how everything ties together. The README could have summarized more content. You guys did a lot of analysis, the missing part was your conclusions.

## Data Exploration / Cleanup & Final Data Analysis: 35/40

As per the rubric you were suppose to split into **Data Exploration / Cleanup** & **Final Data Analysis.** Understand the data was standard and hence didn’t need much of cleaning.

The notebook presented contains a lot of code and the individual pieces could have been on separate notebook, viz the fundamental analysis.

For data exploration, plot for the closing prices sighted. Volatility, Beta and Sharpe Ratio calculated however no commentary/insights. The plot about comparing historical vs covid 19 volatility is interesting, but this should have been outlined in the report with possible reasons for the differences in volatility or the lack of it.

Variable names should be more context specific. data\_monte\_carlo\_df is the actually slice of the original dataframe and should be named accordingly, monte carlo doesn’t seem appropriate, even confusing.

It was good that you plotted the heatmap for the stocks and the index. However without any summary it doesn’t solve any purpose. Make sure that there is atleast one line conclusion for any plots you have.

For Beta: Earlier you mentioned about calculating **Beta**, but in the actual calculation its commented as **Covariance of Selected Stock Returns vs. S&P Returns.** The calculations are fine, but you need to communicate effectifvely to all your stakeholders.

Great job creating a function for the monte carlo sims and calling when required.

For some reason I wasn’t able to plot the fundamental analysis plots/dashboard. I got an error while running the data\_clean for RY, AMZN, WT data clean functions.

Good job with using new library Financial Model Prep. Dashboard not visible.DRY evident.

**Overall great job, you could have integrated more of the fundamental analysis to justify/explain the trend you found for the Retail/Banking and Ecom stocks and their performance pre and during Covid-19.**

### SQL: FB for Alex

### # Data Modelling 20/20

### ERD diagram is correct.

### # Data Engineering 18/20

### The solution expected was that schema to be designed by you and creation of the tables using reference to primary and foreign keys. However the solution provided is just an export of the Quick DBD.

### Dtypes of Credit card are to be varchar as different credit cards may have different lengths of numbers.

### Primary and foreign keys are present.

### # Data Analysis: 30/30

### The report uploaded is very concise, the ERD is cut of a bit on the left side, make sure relationships between tables are visible and not ambigious. Apart from that, report looks good.

### Fradulent transactions have been identified correctly. For the customer id 25, the conclusions point towards fradulent transactions, you concluded otherwise.

### Use of hvplots & appropriate SQL queries, sighted on the notebook.

### # Coding Conventions/Formating 10/10

### Good job on use of precise variable names.

### # Deployment/Submission 10/10

### # Documentation/Comments 10/10

SQL: Feedback for Ian:

# Data Modelling 20/20

Your representation of Conceptual, Logical and Physical model is correct. One thing which could be changed is the dtype for credit\_card number, this should be varchar as different banks/finacial institiution might have different number of total characters.

# Data Engineering 12/20

The schemas for the tables are missing.

# Data Analysis 25/30

Queries for the views are missing in the solutions. Use of hvplots, SQL alchemy sighted.

# Coding Conventions/Formating

Coding conventions followed. Avoid blank cells in the final notebook.

# Deployment/Submission 9/10

Github comments can be more precise

# Documentation/Comments 10/10

**# Feedback for Amar**

Data Modelling 20/20

### ERD diagram is correct, the ERD is cut of a bit on the left side, make sure relationships between tables are visible and not ambigious.

### # Data Engineering 12/20

Dtypes of Credit card are to be varchar as different credit cards may have different lengths of numbers.

There is complete mismatch between the ERD table names, the table names used in SQL insert queries and also the ipynb files. Please remove files which are rough draft, you have 2 sql table schemas files.

# Data Analysis 30/30

Hvplot and SQL alchemy usage sighted.

Its good that you have done some detailed analysis, but the scope of the data is too less for requiring complex analysis, same insights can be achieved using simple graphs. Readability of the charts is less as one needs to really dig into code to understand the chart. Not very intuitive.

For “Using Plotly Express, create a series of six box plots, one for each month, in order to identify how many outliers per month for cardholder ID 25.”

A simple box plot with outliers is good enough to identify the fraud. Good analysis with the z-score to quantify the fraud.

# Coding conventions 6/10

Table names do not follow the standard of snake case. Use of upper case is not good practise.

Incorrect use of variable names with use of upper case.

Imports not done at the start

Code looks too cluttered with use of lot of temperory dataframes which reduces the overall readability.

# Deployment/Submission 9/10

Good comments. You can remove draft files from the final submission.

# Documentation/Comments 10/10

**# Feedback for Jessica**

Data Modelling 20/20

ERD diagram is correct,

### # Data Engineering 18/20

### The solution expected was that schema to be designed by you and creation of the tables using reference to primary and foreign keys. However the solution provided is just an export of the Quick DBD.

### Dtypes of Credit card are to be varchar as different credit cards may have different lengths of numbers.

### Primary and foreign keys are present.

# Data Analysis 30/30

# Coding conventions 10/10

Good variable names used in the code,.

# Deployment/Submission 10/10

# Documentation/Comments 10/10

**# Feedback for Manjari**

Data Modelling 20/20

ERD diagram is correct,

### # Data Engineering 18/20

Dtypes of Credit card are to be varchar as different credit cards may have different lengths of numbers.

No Schema files sighted in the repo.

# Data Analysis 30/30

### Fradulent transactions have been identified correctly. Good analysis and precise conclusions.

### Use of hvplots & appropriate SQL queries, sighted on the notebook.

# Coding conventions 10/10

Good variable names used in the code,.

# Deployment/Submission 10/10

# Documentation/Comments 10/10

**Home work 7 Time Series:**

**Amar has some interesting insights regarding the JPY and CAD analyses for time series.**

**Richa has done some analysis with the help of ACF and PACF to decide which order is to be used for the time series models.**

Time-Series Forecasting

• Hodrick-Prescott Filter utilized to decompose the settle price into trend and noise.

• ARMA Model used to forecast returns.

• ARIMA Model used to forecast settle price.

• GARCH Model used forecast volatility.

Time Series analysis

• Purchase of the yen analyzed for or against.

• Risk of the yen analyzed.

• Confidence of models as a basis for trading analyzed.

Linear Regression Forecasting

• Data prepared, returns and lagged returns created and data split into training and testing.

• Linear Regression Model fitted..

• Predictions maded using testing data.

• Out-of-sample performance.

• In-sample performance.

Linear Regression Analysis

• Moderm performance analysed for out-of-sample and in-sample data.

Coding Conventions/Formating

• Appropriate header, name, short description at top of the notebook

• Imports are at the top of the file, just after any headers or subheads.

• Files read in from relative file path

• Functions and variable names are descriptive, lowercase, with words separated by underscores

• Clean code, no repetition, maintainable and highly reusable code.

• Appropriate code wrapping and

Deployment/Submission

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

Documentation/Comments

• Code is well commented with concise, relevant comments

10 Points Mastery 9 Points Approaching Mastery 8 Points Progressing 8 > 0 Emerging 0

LETTER

**Feedback for Veldurai**

## Time-Series Forecasting 35/35

HP filter used, ARMA ARIMA and GARCH used to forecast respective values.

Time Series analysis

Purchase of yen argued with the help of analysis of both the models. Risk also explained. Rightly stated that the models are too simple to be used for actual trading.

Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

Linear Regression Analysis

Both performace of in sample and out of sample calculated.

Coding Conventions/Formating 10/10

Clean code. DRY seen. Good formatting.

Deployment/Submission 10/10

Files were submitted in personal repo

Appropriate commit messages sighted.

Documentation/Comments 10/10

• Code is well commented with concise, relevant comments

10 Points Mastery 9 Points Approaching Mastery 8 Points Progressing 8 > 0 Emerging 0

LETTER

**Feedback for Nitesh:**

Time-Series Forecasting **30/35**

HP filter used, ARMA, ARIMA and GARCH used to forecast returns, price and volatility respectively.

Time Series analysis

Purchase of the yen analyses is incorrect, in the shorter term the YEN is to gain against the CAD and once should maybe buy

Risk of the yen analysed.

Confidence of models as a basis for trading rightfully stated as not good.

Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

Linear Regression Analysis

Both performace of in sample and out of sample calculated.

Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

Deployment/Submission 9/10

Commit messages can be more descriptive.

Documentation/Comments 10/10

Code is well commented with concise, relevant comments

Feedback for Richard

## Time-Series Forecasting: 35/35

HP filter used, ARMA ARIMA and GARCH used to forecast respective values.

Time Series analysis

Purchase of yen argued with the help of analysis of both the models. Risk also explained. Rightly stated that the models are too simple to be used for actual trading.

## Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

## Deployment/Submission 9/10

Commit messages can be more descriptive.

## Documentation/Comments 10/10

Code is well commented with concise, relevant comments

Feedback for Sinthushan

## Time-Series Forecasting **30/35**

HP filter used, ARMA, ARIMA and GARCH used to forecast returns, price and volatility respectively.

Time Series analysis

Purchase of the yen analyses is incorrect, in the shorter term the YEN is to gain against the CAD and once should maybe buy

Risk of the yen analysed.

Confidence of models as a basis for trading rightfully stated as not good.

## Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

## Deployment/Submission 10/10

Files were submitted in personal repo

Appropriate commit messages sighted.

## Documentation/Comments 10/10

Code is well commented with concise, relevant comments

Feedback for Richa

Time-Series Forecasting **30/35**

HP filter used, ARMA, ARIMA and GARCH used to forecast returns, price and volatility respectively.

Time Series analysis

Purchase of the yen analyses is incorrect, in the shorter term the YEN is to gain against the CAD and once should maybe buy

Risk of the yen analysed.

Confidence of models as a basis for trading ly stated as not good.

## Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

Comments:

You have done some good analysis with PACF and ACF, however the conclusions with respect to the usage in the models isnt outlined in depth.

For Linear Regressions, your conclusions of overfitting is opposite to the observations. In this case, where the RMSE for out of sample is lower, means the model generalizes well(surprisingly). If this is what you were trying to conclude, then its not very clear and sounds a bit ambigious. But great job overall!

Feedback for Manjari!

## Time-Series Forecasting **30/35**

HP filter used, ARMA, ARIMA and GARCH used to forecast returns, price and volatility respectively.

Time Series analysis

Purchase of the yen analyses is incorrect, in the shorter term the YEN is to gain against the CAD and once should maybe buy

Risk of the yen analysed.

Confidence of models as a basis for trading rightly stated as not good.

## Linear Regression Forecasting 35/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

## Deployment/Submission 10/10

Files were submitted in personal repo

## Documentation/Comments 10/10

Code is well commented with concise, relevant comments

Feedback for Musonda

## Time-Series Forecasting **30/35**

HP filter used, ARMA, ARIMA and GARCH used to forecast returns, price and volatility respectively.

Time Series analysis

Purchase of the yen analyses is incorrect, in the shorter term the YEN is to gain against the CAD and once should maybe buy

Risk of the yen analysed.

Confidence of models as a basis for trading rightly stated as not good.

1. ARIMA model specifications given as order=(5,1,1)

But you have considered order = (5,1,2) hence the trend is a little different however does point to strengthning JPY against the CAD.

## Linear Regression Forecasting 32/35

Data prepped and returns and lagged returns calculated. Train and test data split.

In sample and out of sample performance addressed.

Analysis that model is underfit is not accurate. The model actually performs well on unseen data and generalizes well. Ofcourse one would need to continue monitoring performance to measure if the accuracy holds up in the longer term

## Coding Conventions/Formating 9/10

Code is clean with the use of correct spacings.

Results variables should be in lower case.

## Deployment/Submission 10/10

Files were submitted in personal repo

## Documentation/Comments 10/10

Code is well commented with concise, relevant comments

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**# Machine Learning/ Classification**

**## Resampling /35**

• **Data** Oversampled with **Naive Random Oversampler** and **SMOTE** algorithms**.**

• **Data** Undersampled with **Cluster Centroids** algorithm.

• **Data Over** and **Undersampled** with combination of **SMOTEENN** algorithim.

**Classification Analysis**

• Best Balance accuracy score determined.

• Best Recall score determined.

• Best Geometric mean score determined.

**## Ensemble Learning /35**

• Model **trained** using **Quarterly Data.**

• **Balance Accuracy Score** calculated from ***sklearn.metrics***.

• **Confusion Matrix** printed from ***sklearn.metrics***.

• **Classification Report** generated using ***imbalanced\_classification\_report*** from ***imbalanced learn***.

• **Feature Importance** printed and sorted in descending order for balanced random forest classifier along with

**Feature Score**

**Classification Analysis**

• **Best Balanced Accuracy Score** model determined.

• **Best Recall Score Model** determined.

• **Best Geometric Mean Score** determined.

• **Top Three Features** determinded.

**## Coding Conventions/Formating /10**

• Appropriate header, name, short description at top of the notebook

• Imports are at the top of the file, just after any headers or subheads.

• Files read in from relative file path

• Functions and variable names are descriptive, lowercase, with words separated by underscores

• Clean code, no repetition, maintainable and highly reusable code.

• Appropriate code wrapping and cell sizes

• Appropriate subheads as needed

**## Deployment/Submission /10**

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments /10**

• Code is well commented with concise, relevant comments

**Feedback for MANJARI**

**## Resampling 35/35**

* **Oversampling by Naive Random Oversampler** and **SMOTE** algorithms **sighted.**
* **Undersampling with Cluster centroids sighted.**
* Balanced accuracy scores, Recall and geo mean calculated.

**## Ensemble Learning 35/35**

* ***sklearn.metrics used to calculate relevant scores and confusion matrices.***
* **All relevant reports sighted.**
* **Train test split correctly**
* **BRT feature importance displayed correctly along with feature scores. Accuracy, Recall and Geo mean all sighted.**
* **Top 3 identified.**

**## Coding Conventions/Formating 10/10**

All coding conventions followed.

**## Deployment/Submission 10/10**

Satisfactory

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

Feedback for Elizabeth

**## Resampling 35/35**

• Oversampling by Naive Random Oversampler and SMOTE algorithms sighted.

• Undersampling with Cluster centroids sighted.

• Balanced accuracy scores, Recall and geo mean calculated.

**## Ensemble Learning 35/35**

* ***sklearn.metrics used to calculate relevant scores and confusion matrices.***
* **All relevant reports sighted.**
* **Train test split correctly**
* **BRT feature importance displayed correctly along with feature scores. Accuracy, Recall and Geo mean all sighted.**
* **Top 3 identified.**

**## Coding Conventions/Formating 8/10**

Avoid naming variables like generic model esp when there are multiple models been applied. Try to be more specific eg model\_lr etc. Formatting can be improved, at times there is line after the imports and at times there isnt. Do follow a consistent process which leads to better code readablity.

**## Deployment/Submission 8/10**

Github messages can be more descriptive.

**## Documentation/Comments 9/10**

You can remove redundant comments like # YOUR CODE HERE. From your final submission.

**Feedback for Mihir**

**## Resampling 33/35**

* **Oversampling by Naive Random Oversampler** and **SMOTE** algorithms **sighted.**
* **Undersampling with Cluster centroids sighted.**
* Balanced accuracy scores, Recall and geo mean calculated.

No analysis or comparison between the different models provided in the notebook or a report

Good use of stratification

**## Ensemble Learning 34/35**

* ***sklearn.metrics used to calculate relevant scores and confusion matrices.***
* **All relevant reports sighted.**
* **Train test split correctly**
* **BRT feature importance displayed correctly along with feature scores.**

**Accuracy, Recall and Geo mean all sighted.**

**Going forward please provide some conclusion for the different models which have been used and how each compares to the others in terms of the scores outlined in the starter code.**

**## Coding Conventions/Formating 8/10**

Avoid using generic variable name like model at every model fit stage. You can be more specific with the variable naming by considering model and the sampling technique.

Eg smo\_model or ros\_model instead of just model.

**## Deployment/Submission 8/10**

No Readme sighted which gives a description of the methodology used.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

Feedback for Jessica!

## Resampling 32/35

* **Oversampling by Naive Random Oversampler** and **SMOTE** algorithms **sighted.**
* **Undersampling with Cluster centroids sighted.**
* Balanced accuracy scores, Recall and geo mean calculated.

As per the balanced accuracy scores, after the use of stratification, the model with best accuracy is SMOTE oversampling, next is Naïve Oversampling and the 3rd is SMOTEEN. Your conclusions state SMOTEEN as the one with the best accuracy.

## Ensemble Learning 35/35

Top 3 features identified correctly, however the order of first 2 features have been reversed, reason being the n\_estimators considered is 1000.

## Coding Conventions/Formating 9/10

The notebooks submitted are very neat, however at times there seems to be a line between the comment and the code (or the imports and the next line of code), and most of the time there isnt. Its nothing major, but a consistent formatting will make the code look better.

## Deployment/Submission 10/10

Good commit notes.

## Documentation/Comments /10

• Code is well commented with concise, relevant comments

**Feedback for Zeldi**

**## Resampling 35/35**

* **Oversampling by Naive Random Oversampler** and **SMOTE** algorithms **sighted.**
* **Undersampling with Cluster centroids sighted.**
* Balanced accuracy scores, Recall and geo mean calculated.

Good analysis, all points made are very relevant.

**## Ensemble Learning 35/35**

* ***sklearn.metrics used to calculate relevant scores and confusion matrices.***
* **All relevant reports sighted.**
* **Train test split correctly**
* **BRT feature importance displayed correctly along with feature scores. Accuracy, Recall and Geo mean all sighted.**
* **Top 3 identified.**

Great job with scaling the data and fitting the model to test the results.

**## Coding Conventions/Formating 10/10**

Good presentation on the notebooks, all imports are at the start, formatting is very consistent.

**## Deployment/Submission /10**

Good commit notes.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Unit 12 – Natural Language Processing (NLP)**

**## Sentiment Analysis /35**

Coin with highest MPS, neg score, positive score.

**## NLP /25**

Tokenize:

NLTK to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK to produce n-grams.

Top 10 words for each Coin

Cloud Generation:

Word cloud for each coin.

**## NER /10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating /10**

• Appropriate header, name, short description at top of the notebook

• Imports are at the top of the file, just after any headers or subheads.

• Files read in from relative file path

• Functions and variable names are descriptive, lowercase, with words separated by underscores

• Clean code, no repetition, maintainable and highly reusable code.

• Appropriate code wrapping and cell sizes

• Appropriate subheads as needed

**## Deployment/Submission /10**

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments /10**

• Code is well commented with concise, relevant comments

Random learnings: In the NLP, while using the NLTK part, there is this concept of lemmitizer, which strips the plural word to the singluar form. Another preprocessing function is stemming which is basically reducign the word to its root.

Eg randomizer to random, cleaning to clean.

Feedback for Daphene – Unit 11 NLP

**## Sentiment Analysis 35/35**

**BTC and ETH mean positive score, negative and postive score calculated.**

**## NLP 21/25**

Tokenize:

Tokenize function defined correctly to lower case and remove punctuations and stopwords.

N-grams:

Nltk library used to find n-grams and top 10

Cloud Generation:

Word cloud for each coin generated.

**## NER 10/10**

NER for BTC and ETH articles displayed.

**## Coding Conventions/Formating 10/10**

Coding conventions followed

**## Deployment/Submission 10/10**

• Files submitted in personal repo

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

**Feedback for David**

**## Sentiment Analysis /35**

Coin with highest MPS, neg score, positive score.

**## NLP /25**

Tokenize:

NLTK to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK to produce n-grams.

Top 10 words for each Coin

Cloud Generation:

Word cloud for each coin.

**## NER /10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating /10**

• Appropriate header, name, short description at top of the notebook

• Imports are at the top of the file, just after any headers or subheads.

• Files read in from relative file path

• Functions and variable names are descriptive, lowercase, with words separated by underscores

• Clean code, no repetition, maintainable and highly reusable code.

• Appropriate code wrapping and cell sizes

• Appropriate subheads as needed

**## Deployment/Submission /10**

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments /10**

• Code is well commented with concise, relevant comments

Feedback for Amit:

**Unit 12 – Natural Language Processing (NLP)**

**## Sentiment Analysis 35/35**

Coin with highest MPS, neg score, positive score calculated and appropriate coins identified.

**## NLP 25/25**

Tokenize:

NLTK used to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK was used to produce n-grams ( bi-gram) with the most common top 10 identified. Good job!

Top 10 words for each Coin identified.

Cloud Generation:

Word cloud for each coin generated.

**## NER 10/10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating 10/10**

All or Most of conventions followed.

**## Deployment/Submission 10/10**

Appropriate messages for the commits seen.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

Feedback for Bailey:

**## Sentiment Analysis 35/35**

Coin with highest MPS, neg score, positive score calculated and appropriate coins identified.

**## NLP 21/25**

Tokenize:

NLTK used to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK was used to produce n-grams ( bi-gram). One piece that was missing was the top 10 or 20 bi-grams for each word.

Top 10 words for each Coin identified.

Cloud Generation:

Word cloud for each coin generated.

**## NER 10/10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating 10/10**

All or Most of conventions followed.

**## Deployment/Submission 10/10**

Appropriate messages for the commits seen.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

Feedback for Abu

**## Sentiment Analysis 35/35**

Coin with highest MPS, neg score, positive score.

**## NLP 20/25**

Tokenize:

NLTK used lower\_case, remove punctuations and stopwords.

N-grams:

NLTK to produce n-grams not visible in the code.

Top 10 words for each Coin idenfied.

Cloud Generation:

Word cloud generated for each coin.

**## NER 10/10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating 8/10**

Report isnt completed. Its completely empty, all you had to do was summarize your findings in the report.

**## Deployment/Submission 10/10**

All or most of conventions followed

**## Documentation/Comments 8/10**

# Create a newsapi client

# YOUR CODE HERE!

Feedback for Alex:

Please refrain from using variables names like btc\_ngrams and eth\_ngrams variables when calculating top 10 words, causes confusion and reduces the readability of the code.

Feedback for Amar:

**## Sentiment Analysis 32/35**

Coin with high compund score, positive score identified correctly, However analysis for Mean positive score isnt correct. As per your table, BTC has been mean positive score of 0.070632 while ETH has 0.048750. Hence BTC is the correct answer.

**## NLP 25/25**

Tokenize:

NLTK used to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK was used to produce n-grams ( bi-gram) with the most common top 10 identified. Good job!

Top 10 words for each Coin identified.

Cloud Generation:

Word cloud for each coin generated.

**## NER 10/10**

Model built for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating /10**

All or most coding conventions are followed. Code looks uniform and readability is good!

**## Deployment/Submission 8/10**

Readme is not displaying the images.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

Feedback for Carlos:

**## Sentiment Analysis 35/35**

Coin with highest MPS, compound score, positive score calculated and identified.

**## NLP 21/25**

Tokenize:

NLTK used to lower\_case, remove punctuations and stopwords.

N-grams:

NLTK was used to produce n-grams ( bi-gram). One piece that was missing was the top 10 or 20 bi-grams for each word.

Top 10 words for each Coin identified.

Cloud Generation:

Word cloud for each coin generated.

**## NER 10/10**

Model build for BTC and ETH – Tags Visualilzed using SpaCy.

**## Coding Conventions/Formating 7/10**

Importing packages in functions is bad practise, imports should be done at the start of the code or atleast the module.

**## Deployment/Submission 8/10**

No Readme has been submitted which summarizes the results.

**## Documentation/Comments 9/10**

• Code is well commented with concise, relevant comments. However comments are arent spaced properly and in functions the readabliity is low.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**13 AWS Lex and Lambda**

## Initial Robo Advisor Configuration /35

• RoboAdvisor created with proper parameter. • RecommendPortfolio created and configured with proper name utterances. • RiskLevel custom slots created with proper card slots. • RoboAdvisor tested after build with error handling configuration.

## Enhance RoboAdvisor with Amazon Lambda Function

• User Input Validated. • Investment Portfolio Recommendation given on selected risks. • Lambda Function tested with sample test cases. • Lambda Function integrated to the RoboAdvisor.

**## Coding Conventions/Formating 10 / 10**

All or most conventions followed

**## Deployment/Submission 10 / 10**

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments /10**

• Code is well commented with concise, relevant comments

**Feedback for Veldurai:**

## Initial Robo Advisor Configuration 35/35

RoboAdvisor attempted, with all proper configurations. Custom slots created with appropriate risk levels. Video showing error handling has been uploaded.

## Enhance RoboAdvisor with Amazon Lambda Function 35/35

Inputs validated and lambda is integrated into the advisor. Appropriate recommendations as per the user input risks. Good short video with all cases.

## Coding Conventions/Formating 10/ 10

All or most conventions followed.

## Deployment/Submission 10 / 10

README sighted. Thanks for explaining the reason for the less commit history

## Documentation/Comments 10 /10

• Code is well commented with concise, relevant comments

**Feedback for Sinthushan:**

## Initial Robo Advisor Configuration 35/35

RoboAdvisor attempted, with all proper configurations. Custom slots created with appropriate risk levels. Video shows happy path, it would have been better if you could also show the error handling for the age/amount in the video.

## Enhance RoboAdvisor with Amazon Lambda Function 35/35

Inputs validated and lambda is integrated into the advisor. Appropriate recommendations as per the user input risks.

## Coding Conventions/Formating 10/ 10

All or most conventions followed.

## Deployment/Submission 10 / 10

README sighted. Thanks for explaining the reason for the less commit history

## Documentation/Comments 10 /10

• Code is well commented with concise, relevant comments

Feedback for Richa! – Clustering with Sagemaker!

## Data Preprocessed 35/35

• Data loaded using Pandas DataFrame into crypto\_df. • Data Preprocessed with the assigned preprocessing tasks. Data Dimension Reduced

• PCA algorithm from sklearn used to reduce dimensions.

• pcs\_df DataFrame created and crypto\_df.index used as the index for pcs\_df DataFrame.

## Cryptocurrency Clustered 34/35

• K-Means used to cluster the cryptocurrencies using PCA data.

• Elbow Curve used to find the best value for k, Seems like the best k value is 4, you have selected 5. Although the accuracy maybe better for 5 clusters, one should consider the diminishing value of additional clusters, 4 seems like a better option looking at the elbow chart.

• Kmeans algorithm used to predict the k cluster for cryptocurrency data. • New DataFrame created named clustered\_df, includes assigned columns and index.

Visualizing Results

• 3D-Scatter plotted created using Plotly using clustered\_df DataFrame, paramaters used as directed. • Data table created using hvplot.table for all current tradable cryptocurrencies, columns used as directed. • Scatter plot created using hvplot.scatter, to present clustered data with directed parameters.

## Optional Challenge 10/10

• Jupyter notebook uploaded to Amazon SageMaker and deployed

## Coding Conventions/Formating 7/ 10

Please avoid printing the entire response, it is bloating the whole code and reduces the readability.

## Deployment/Submission 10 / 10

README sighted

## Documentation/Comments 10 /10

• Code is well commented with concise, relevant comments

**DEEP LEARNING – HW 14**

**## Data Prep for Training and Testing /35**

• **FNG** values used to predict future closing prices.

• **Past Closing Prices** used to predict furture closing prices.

• **MinMaxScaler** applied to the X and Y values to scale the data for the model.

• **X\_train** and **X\_test** reshaped to fit the model requirements (samples, time steps, features)

**## Build and train custom LSTM RNNS /35**

• **Notebook created to fit data using FNG Values.**

• **Notebook created to fit data using closing prices.**

**Evaluate the Perfomance of each model.**

• **Lowest loss model determined.**

**• Model** with best accuracy tracking determined.

• **Window Size** determined for the model.

**## Coding Conventions/Formating / 10**

• Appropriate header, name, short description at top of the notebook

• Imports are at the top of the file, just after any headers or subheads.

• Files read in from relative file path

• Functions and variable names are descriptive, lowercase, with words separated by underscores

• Clean code, no repetition, maintainable and highly reusable code.

• Appropriate code wrapping and cell sizes

• Appropriate subheads as needed

**## Deployment/Submission / 10**

• Files submitted in personal repo

• Appropriate directory structure with correct files needed to run scripts

• Appropriate commit messages

• Appropriate README

**## Documentation/Comments /10**

• Code is well commented with concise, relevant comments

**Feedback for Ian**

**## Data Prep for Training and Testing 35/35**

**FNG and closing prices used to plot the predictions. Scaling used for all data slices individually. Reshaping done appropriately to fit the model.**

**## Build and train custom LSTM RNNS 35 /35**

2 notebooks sighted for FNG and closing respectively

**Multiple model with differnet parameters showcased. Excellent work for collating all the scenarios with different models**

**Lowest loss model was determined.**

Great job with experimenting with the **Window Size**

**## Coding Conventions/Formating 10 / 10**

Most or all conventions followed

**## Deployment/Submission 10 / 10**

Readme sighted.

**## Documentation/Comments 10 /10**

Code is well commented with concise, relevant comments

**FEEDBACK FOR ELIZABETH**

**Data Prep for Training and Testing 35/35**

**FNG and closing prices used to plot the predictions. Scaling used for all data slices individually. Reshaping done appropriately to fit the model.**

**Build and train custom LSTM RNNS 32/35**

2 notebooks for FNG and closing prices sighted.

**Evaluations of the model was missing with the details on the README. The link to loss and accuracy isnt working. Just a table with comparison with the different metrics would have sufficed. Window Size** determined for the model was 10. Please do experiment with more window sizes.

**Coding Conventions/Formating 10 / 10**

Most or all conventions followed

**Deployment/Submission 8 / 10**

Read me was uploaded. However presentation can be better

**Documentation/Comments 10 /10**

• Code is well commented with concise, relevant comments

**Feedback for jessica**

**## Data Prep for Training and Testing 35/35**

**Code doesn’t work in the second cell, as you need to call random.set\_seed(2) instead of random.set\_random\_seed(2). Apart from that the entire code works.**

**FNG and closing prices used to plot the predictions. Scaling used for all data slices individually. Reshaping done appropriately to fit the model.**

**## Build and train custom LSTM RNNS 30/35**

2 notebooks for FNG and closing prices sighted.

**Evaluations of the model was missing with the details on the README. Please do review the rubrik answer all questions if possible. The code does work but you need to make notes for the models and how they perform.**

**## Coding Conventions/Formating 10 / 10**

Most or all conventions followed

**## Deployment/Submission 5/ 10**

No README sighted. No notes on the different models and the interpretation.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

**Feedback for Manjari/Zeldi**

**## Data Prep for Training and Testing 35/35**

**FNG and closing prices used to plot the predictions. Scaling used for all data slices individually. Reshaping done appropriately to fit the model.**

**## Build and train custom LSTM RNNS 35/35**

2 notebooks for FNG and closing prices sighted.

**Performance of each model commented with lowest loss model determined. Accuracy and window size also determined in each case**

**## Coding Conventions/Formating 10 / 10**

Most or all conventions followed

**## Deployment/Submission 10 / 10**

README sighted. README is very concise and to the point. Great job!

**## Documentation/Comments 10 /10**

• Code is well commented with concise, relevant comments

Feebback for Mihir

**## Data Prep for Training and Testing 35/35**

**FNG and closing prices used to plot the predictions. Scaling used for all data slices individually. Reshaping done appropriately to fit the model.**

**## Build and train custom LSTM RNNS 32/35**

2 notebooks for FNG and closing prices sighted.

**Performance of each model not commented and lowest loss model hasn’t been determined. Accuracy scores also not compared.**

**## Coding Conventions/Formating 10 / 10**

Most or all conventions followed

**## Deployment/Submission 5/ 10**

No README sighted. No notes on the different models and the interpretation.

**## Documentation/Comments 10/10**

• Code is well commented with concise, relevant comments

1. **olidity Homework Grading Rubric**

**\*IAN’S HW is the best and explains everything in great details.**

## 1st Contract Setup

• public variables defined per assigned criteria. • Constructor function created accepting address payable \_one. • Constructor function created accepting address payable \_two. • Constructor function created accepting address payable \_three.

## 1st Contract Functionality

• balance function created with assigned criteria. • deposit function created with assigned criteria. • fallback function created with assigned criteria. • Contract tested; screenshots provided.

## 2nd Contract Setup

• Number of points calculated by dividing msg.value. • uint amount set to points for each employee. • amount added to total used to calculate the msg.value distribution. • amount transfer to each employee, amount set to the points and multiplied by given percentage.

## 2nd Contract Functionality

• remainder sent employee with highest percentage. • Contract Deployed. • Contract Tested, screenshots provided

## 3rd Contract Setup 3

• Human Resources set as the constructor msg.sender. • employee initiation variables created. • uint start\_time and uint public distributed\_shares defined to assigned criteria. • distribute function created with defined criteria.

## 3rd Contract Functionality

• Contract deployed locally, screenshots provided. • Contract Tested with fakenow testing logic. • Contract deployed to a live Testnet, screenshots provided.

## Coding Conventions/Formatting

• Appropriate header, name, short description at top of the notebook • Imports are at the top of the file, just after any headers or subheads. • Files read in from relative file path • Functions and variable names are descriptive, lowercase, with words separated by underscores • Clean code, no repetition, maintainable and highly reusable code. • Appropriate code wrapping and cell sizes • Appropriate subheads as needed

## Deployment/Submission

• Files submitted in personal repo • Appropriate directory structure with correct files needed to run scripts • Appropriate commit messages • Appropriate README

## Documentation/Comments

• Code is well commented with concise, relevant comments

**Feedback for Elizabeth**

## 1st Contract Setup 35/35

Variables assigned public appropriately. All 3 addresses called once using a constructor.

## 1st Contract Functionality 30/35

Balance function created correctly. Deposit function created with correct criteria and called using a fallback function.

Screenshots are provided, however no screenshots provided which show the split of ether among the 3 addresses.

9 ether being deposited, should show 3 ethers/address provided.

## 2nd Contract Setup 35/35

Basis points calculated correctly. Uint amount set for each employee and correct amounts considered.

• Number of points calculated by dividing msg.value. • uint amount set to points for each employee.

## 2nd Contract Functionality 35/35

Remainder is given to employee with highest percentage ie CEO with 60% share.

Screenshots of deployment is there but none of testing of contract logic.

## 3rd Contract Setup /35

• Human Resources set as the constructor msg.sender. • employee initiation variables created. • uint start\_time and uint public distributed\_shares defined to assigned criteria. • distribute function created with defined criteria.

## 3rd Contract Functionality 35/35

Contract deployed locally and screenshots provided with Testnet

## Coding Conventions/Formatting 10/10

Most or All appropriate conventions

## Deployment/Submission 10/10

ReadME sighted.

## Documentation/Comments 10/10

• Code is well commented with concise, relevant comments

Feedback for Ian

## 1st Contract Setup 35/35

Variables assigned public appropriately. All 3 addresses called once using a constructor.

## 1st Contract Functionality 35/35

Balance function created correctly. Deposit function created with correct criteria and called using a fallback function.

## 2nd Contract Setup 35/35

Basis points calculated correctly. Uint amount set for each employee and correct amounts considered.

## 2nd Contract Functionality 35/35

Remainder is given to employee with highest percentage ie CEO with 60% share.

Screenshots of deployment is there but none of testing of contract logic.

## 3rd Contract Setup 35 /35

The contract has been setup correctly.

## 3rd Contract Functionality 35/35

Contract deployed locally and screenshots provided with local deployment.

## Contract Setup 20/35

2 of the contracts arent uploaded.

## Contract Functionality 20/35

Cannot test the same.

## Coding Conventions/Formatting 10/10

Most or All appropriate conventions

## Deployment/Submission 10/10

ReadME sighted.

## Documentation/Comments 10/10

• Code is well commented with concise, relevant comments

## Feel free to resubmit for re-assesment.

## GREAT PRESENTATION - Explaination was detailed but crisp.

Explanation

Feedback for Jessica!

Do provide screenshots as they are important as per the rubrik. Ideally screenshots should be provided for deployment and also testign the logic. Just giving the addresses for contract is not enough to test it.

## Coding Conventions/Formatting 10/10

Most or All appropriate conventions

## Deployment/Submission 10/10

ReadME sighted.

## Documentation/Comments 10/10

• Code is well commented with concise, relevant comments

Feedback for Mihir

## Contract Setup 35/35

Setup looks good.

## Contract Functionality 35/35

Functionality and screenshot confirmed

## Coding Conventions/Formatting 10/10

Most or All appropriate conventions

## Deployment/Submission 10/10

ReadME sighted.

## Documentation/Comments 10/10

• Code is well commented with concise, relevant comments

21- Advanced Solidity

Best HW is from Zeldi!

You need to compile the contracts using the exact names else it wont work.

Also for some reason the initial supply doesn’t seem to work. Good thing is that I was able to make all the contracts work.

Pupper coin is fine, initail supply fails.

Order of sequencing.

First you can create the coin and mint,

Next you fix the rate and be able to sell it(other addresses can buy it).

Lastly you can output the details of the contracts.

Feedback for Carlos:

Designing the contracts: 35/35

Both the contracts are submitted in sol format.

Testing the contracts 20/35

No screenshots in README provided, finalize not used. Contracts have been tested but the submission isnt very neat.

Coding Conventions: 10/10

All or most conventions followed.

Deployment: 5/10

Github looks cluttered with no README

Documentation: 10/10

Amar 18

## ## Case Study 40/40

You did a good job of presenting the business model/use case of HIve. It would have been better if you would have mentioned about the different domains that are currently existing with the blockchain ecosystem and what made you choose this project. However good research and presentation. Report is concise with a lot of relevant details.

## ## Custom Testnet Blockchain 40/40

Screenshots attached within a README.

## ## Deployment/Submission 10/10

## ## Documentation/Comments 10/10

• Code is well commented with concise, relevant comments