Luiss
Libera Università Internazionale degli Studi Sociali Guido Carli

Algorithms A.Y. 2021/2022

Software Project – A Stock Market Data Analyzer

Andrea Coletta, Irene Finocchi
acoletta@luiss.it, finocchi@luiss.it

13 April 2022

LUISS

Dipartimento di Impresa e Management

Group Project Work

The project requires to:

- read a financial dataset
- implement an *efficient* algorithmic solution to study stocks correlation

We release three different parts:

1. February 9 – due to March 9 (not mandatory)
2. *March 9 – **due to April 13 (Today)** (not mandatory)*
3. **April 13 (Today)** – due to first exam session (May, TBA)

Group Project Work: A brief Introduction to Stock Market

The green line represents the price of stock.

The Stock prices are determined in the marketplace, where seller supply meets buyer demand.

And the price of a stock changes over the time according to the market, company performance, and users' actions (buy and sell).

The Figure shows an example of the price of Apple (AAPL) listed at NASDAQ market (US) from 2017 up to February 2022.

LUISS

Software Project: Input Data

You are given four datasets*: small_dataset.txt, medium_dataset.txt, large_dataset.txt, huge_dataset.txt*:

Filename	Size	Rows	Stocks	Days
small_dataset.txt	~400KB	~18K	~50	366
medium_dataset.txt	~3MB	~180k	~500	366
large_dataset.txt	~30MB	~1,8M	~5000	366
huge_dataset.txt	~70MB	~4,3M	~6000	731

In this final release we use all the datasets!

Software Project: Input Data

How we store financial data in a simple way?

You are given as input a .txt file containing a list of stocks and additional details. Each line has:

stock_name, day, price, volume

The values represent the price and volume for the stock_name (e.g., AAPL) in that day.

Stock	Day	Price	Volume
AAPL	458	45	5559100
AAPL	507	288	1938100
TMUS	464	75	3553000
QCOM	723	65	18966800
ROST	397	97	1314100
GOOGL	588	1290	0
GOOGL	581	1290	0
ISRG	727	504	0
GOOGL	643	1398	0

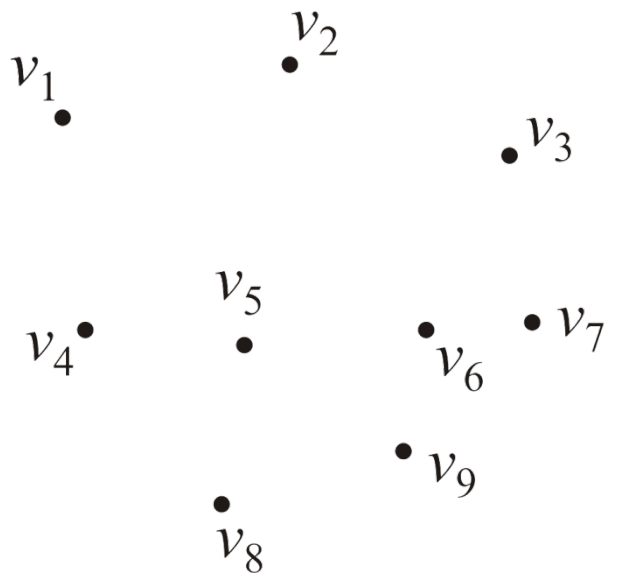
Data are not ordered!!

A brief introduction to Graphs

Consider you're planning a party and you have a group of V friends:

$$V = \{v_1, v_2, ..., v_9\}$$

where $|V| = n$



LUISS

A brief introduction to Graphs

Now you have to create the tables for the party, but you know that some of your friends want to stay on the same table.

You can write down the preferences:

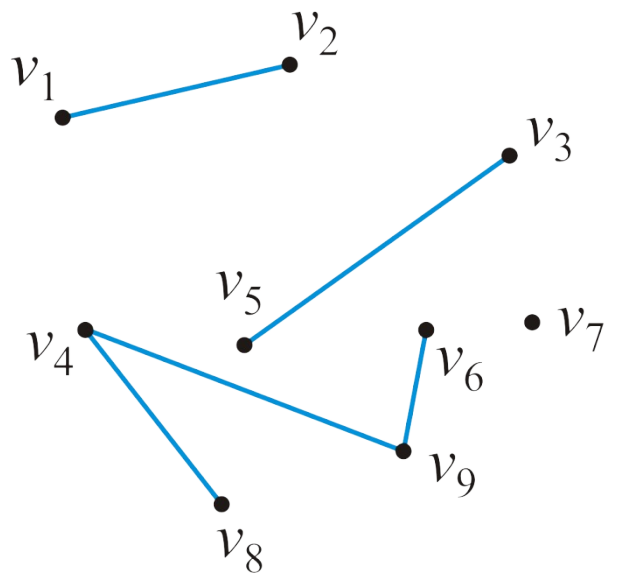
$$preferences = \{\{v_1, v_2\}, \{v_3, v_5\}, \{v_4, v_8\}, \{v_4, v_9\}, \{v_6, v_9\}\}$$

The pair {vj , vk} indicates that vj and vk wants to stay on the same table.

A brief introduction to Graphs

We can use graphs to create and store relationships:

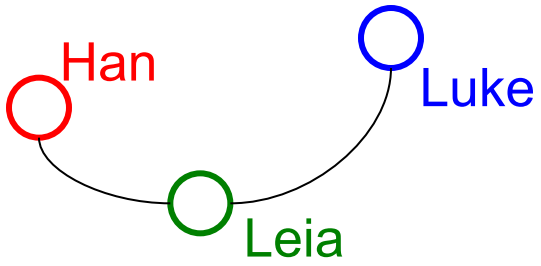
$$E = \{\{v_1, v_2\}, \{v_3, v_5\}, \{v_4, v_8\}, \{v_4, v_9\}, \{v_6, v_9\}\}$$



LUISS

A brief introduction to Graphs

A graph is a formalism for representing relationships among items



A graph is a pair: G = (V, E)

A set of vertices, also known as nodes: $V = \{v_1, v_2, \ldots, v_n\}$

A set of edges $E = \{e_1, e_2, \ldots, e_m\}$

- Each edge e_i is a pair of vertices (v_j, v_k)
- An edge "connects" the vertices

$$V = \{\text{Han}, \text{Leia}, \text{Luke}\}$$
$$E = \{(\text{Luke}, \text{Leia}), (\text{Han}, \text{Leia})\}$$

Graphs can be *directed* or *undirected*

LUISS

Example of Graphs

For each example, what are the vertices and what are the edges?

- Facebook friends

-Road maps

-Airline routes

-Web pages with links

Group Project Work : Final Task

In the final project we will study portfolio management and correlations between stocks!

The goal is to answer a fundamental question:

If I have a stock X in my portfolio, which stocks should I avoid/sell to reduce risk?

Portfolio and Correlation

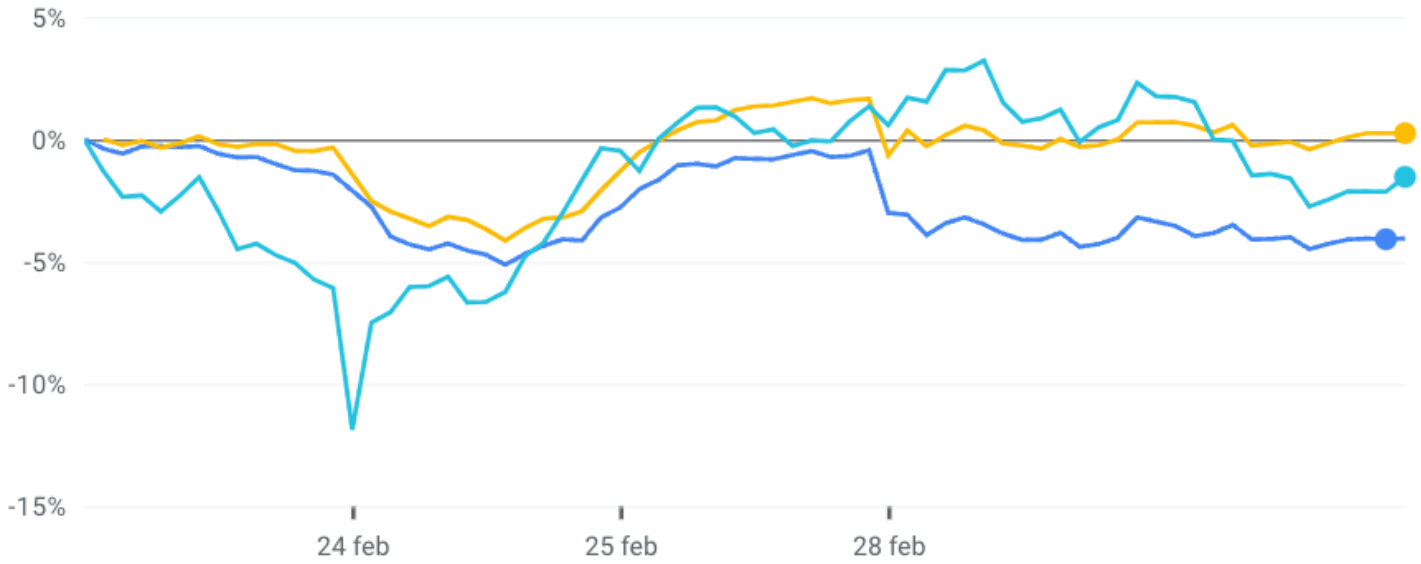
When investing into the financial market we usually create a ***Portfolio.***

A portfolio is a collection (i.e., a set) of financial investments like stocks, bonds, commodities, or exchange traded funds (ETFs).

One of the key concepts in portfolio management is the wisdom of **diversification and risk management!**

Portfolio and Correlation

Which is the best couple of stocks to reduce risk?



PepsiCo	162,22 $	-6,90 $	↓4,08%
The Coca-Cola Co...	61,97 $	+0,16 $	↑0,26%
NVIDIA	234,70 $	-3,64 $	↓1,53%

LUISS

Stock correlation

We first define the return $\underline{r}_s$ for a stock s over a period of N = {0, .. , n} days as follows:

$$r_s = \begin{cases} (x_n - x_0) \,/\, x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

For AAPL for example we have:

Stock	Day	Price	Volume
AAPL	371	200	0
TSLA	369	275	13038300
TSLA	370	273	0
TSLA	371	273	0
AAPL	369	197	18526600
AAPL	370	200	0
AAPL	365	191	27862000
TSLA	365	289	8110400
TSLA	366	286	5478900
TSLA	367	292	7929900
TSLA	368	268	23720700
AAPL	366	194	22765700
AAPL	367	195	23271800
AAPL	368	196	19114300

Stock correlation

We first define the return $\underline{r}_s$ for a stock s over a period of N = {0, .. , n} days as follows:

$$r_s = \begin{cases} (x_n - x_0) \,/\, x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

For AAPL for example we have:

(200 - 191) / 191 = **0.047**

For TSLA we have:

Stock	Day	Price	Volume
AAPL	371	200	0
TSLA	369	275	13038300
TSLA	370	273	0
TSLA	371	273	0
AAPL	369	197	18526600
AAPL	370	200	0
AAPL	365	191	27862000
TSLA	365	289	8110400
TSLA	366	286	5478900
TSLA	367	292	7929900
TSLA	368	268	23720700
AAPL	366	194	22765700
AAPL	367	195	23271800
AAPL	368	196	19114300

LUISS

Stock correlation

We first define the return $\underline{r}_s$ for a stock s over a period of N = {0, .. , n} days as follows:

$$r_s = \begin{cases} (x_n - x_0) \,/\, x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

For AAPL for example we have:

(200 - 191) / 191 = **0.047**

For TSLA we have:

(273– 289) / 289 = **-0.055**

Stock	Day	Price	Volume
AAPL	371	200	0
TSLA	369	275	13038300
TSLA	370	273	0
TSLA	371	273	0
AAPL	369	197	18526600
AAPL	370	200	0
AAPL	365	191	27862000
TSLA	365	289	8110400
TSLA	366	286	5478900
TSLA	367	292	7929900
TSLA	368	268	23720700
AAPL	366	194	22765700
AAPL	367	195	23271800
AAPL	368	196	19114300

Stock correlation

We first define the return r_s for a stock s over a period of N = {0, .. , n} days as follows:

$$r_s = \begin{cases} (x_n - x_0) \,/\, x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

We compute the return because two stocks are correlated if they have a similar return over the entire period *.

This means that they move together in the market:

LUISS

* *A very simplified definition of correlation!!*

Stock correlation

Therefore, we can define a distance score (similarity) between stock i and stock j as the absolute different of their returns:

$$c_{ij} = \quad | r_i - r_j |$$

LUISS

Stock correlation

Therefore, we can define a distance score (similarity) between stock i and stock j as the absolute different of their returns:

$$c_{ij} = \quad |r_i - r_j|$$

According to the previous example, the score between NVIDIA and TESLA is $|0.52 - 0.46| = 0.06$ (very similar) while the score between AMAZON and NVIDIA is $|-0.08 - 0.52| = 0.6$ (they are much more distant).

Stock correlation

Therefore, we can define a distance score (similarity) between stock i and stock j as the absolute different of their returns:

$$c_{ij} = \quad |\, r_i - r_j \,|$$

According to the previous example, the score between NVIDIA and TESLA is $|0.52 - 0.46| = 0.06$ (very similar) while the score between AMAZON and NVIDIA is $|-0.08 - 0.52| = 0.6$ (they are much more distant).

LUISS

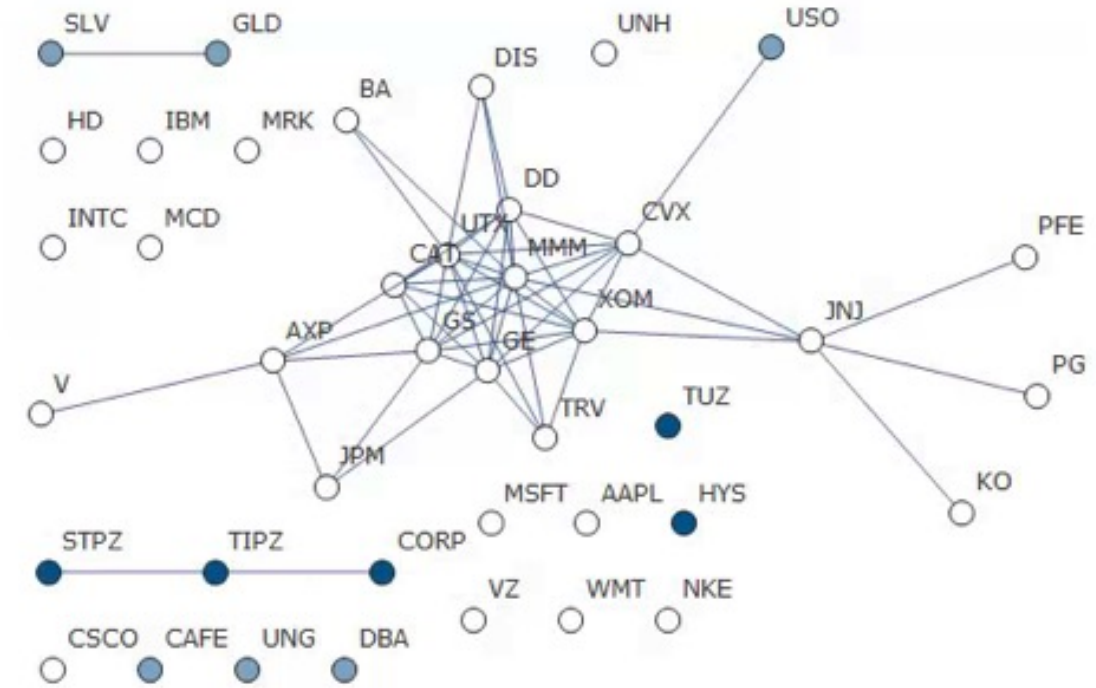
Thus: the less the score the higher the correlation! For simplicity, we'll say that two stocks are correlated if their score c_{ij} is less than a threshold t (i.e., $c_{ij} < t$).

Stock Correlation Graph

To study correlation we often use a graph that maps the interrelations between assets that are correlated at some specified threshold t (0.5 of higher, in this illustration).

We can build a Graph G(V, E)

Where :
- Vertices or Nodes V = stocks
- Edge E = correlations.

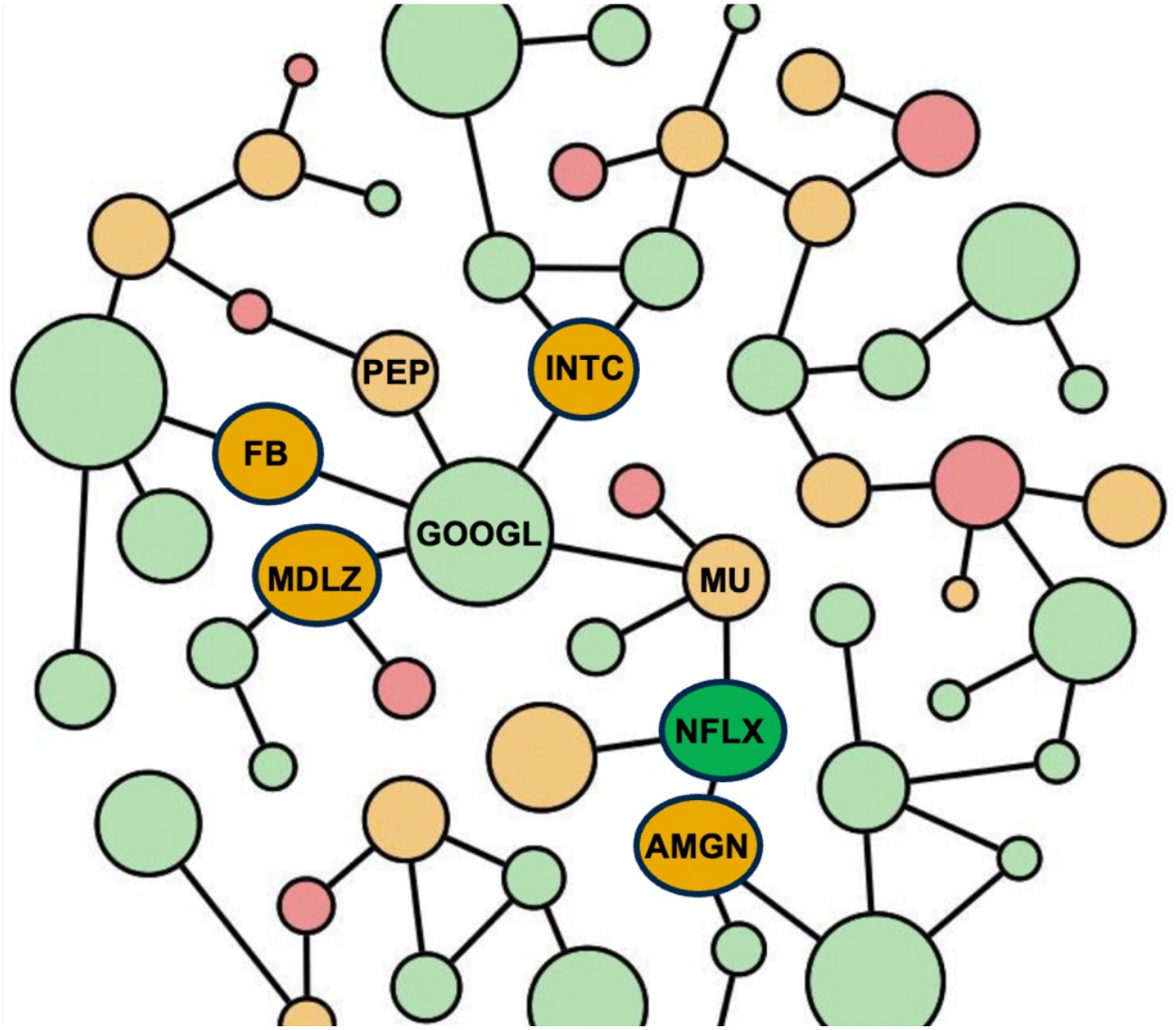


Stock Correlation Graph

Graph are easy to read!

For example, GOOGL is directly correlated with MDLZ, FB, MU, INTC, and PEP. Nevertheless, MU is correlated also with NFLX!
Therefore, we have that:

- If a crash happens on FB is very likely to have a crash also on GOOGL
- If a crash happens on NFLX is very likely to have a crash on MU, and therefore the crash propagates on GOOGL as well!



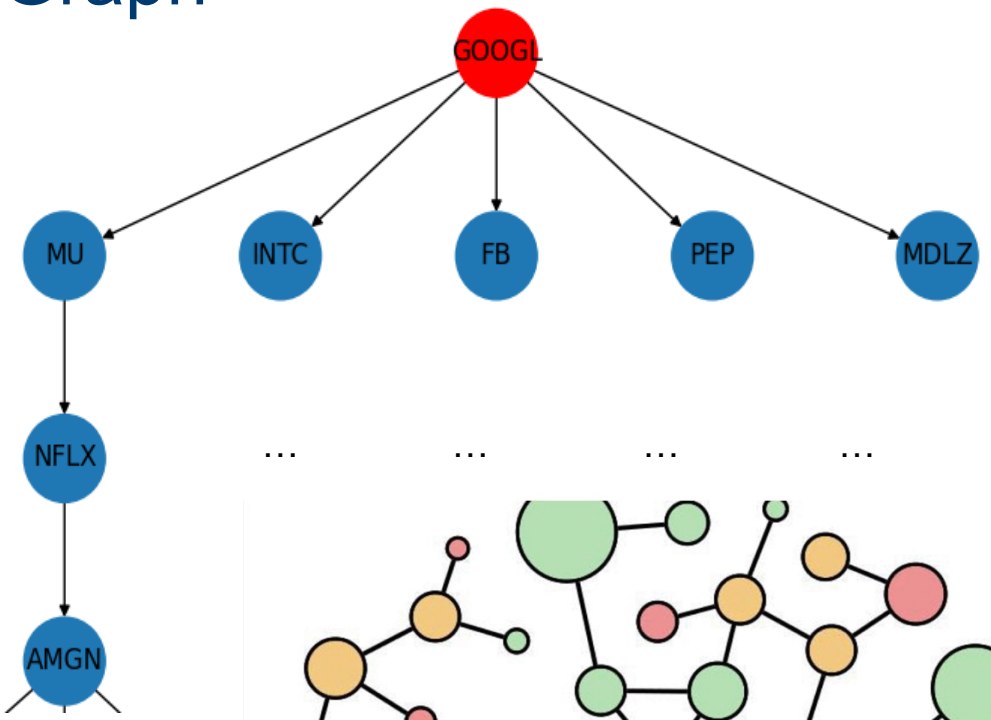
LUISS

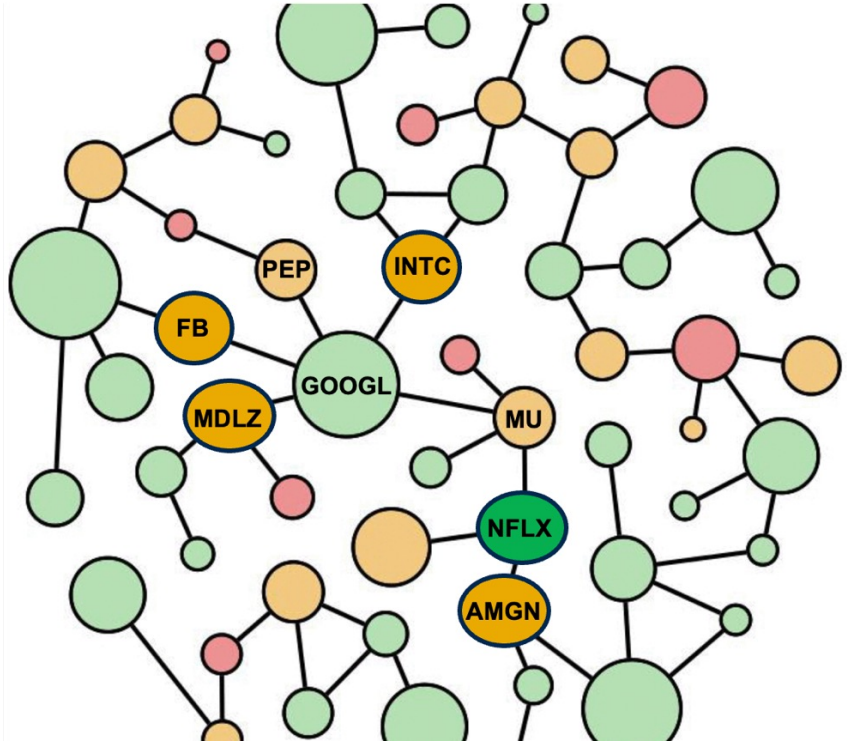
Stock Correlation Graph

We can build a correlation Tree as follows:

- The tree has GOOGL as root;
- At level 1 we have the stock that are directly correlated (MDLZ, FB, MU, INTC, and PEP);
- At level 2 we have the stocks that are correlated through a single node (e.g., NFLX is correlated through MU);
- At level 3 the stocks that are correlated through 2 nodes (e.g., AMGN is correlated to NFLX, that is correlated to MU and finally to GOOGL);

In general, at level i we have the stocks correlated through i-1 nodes to the root!





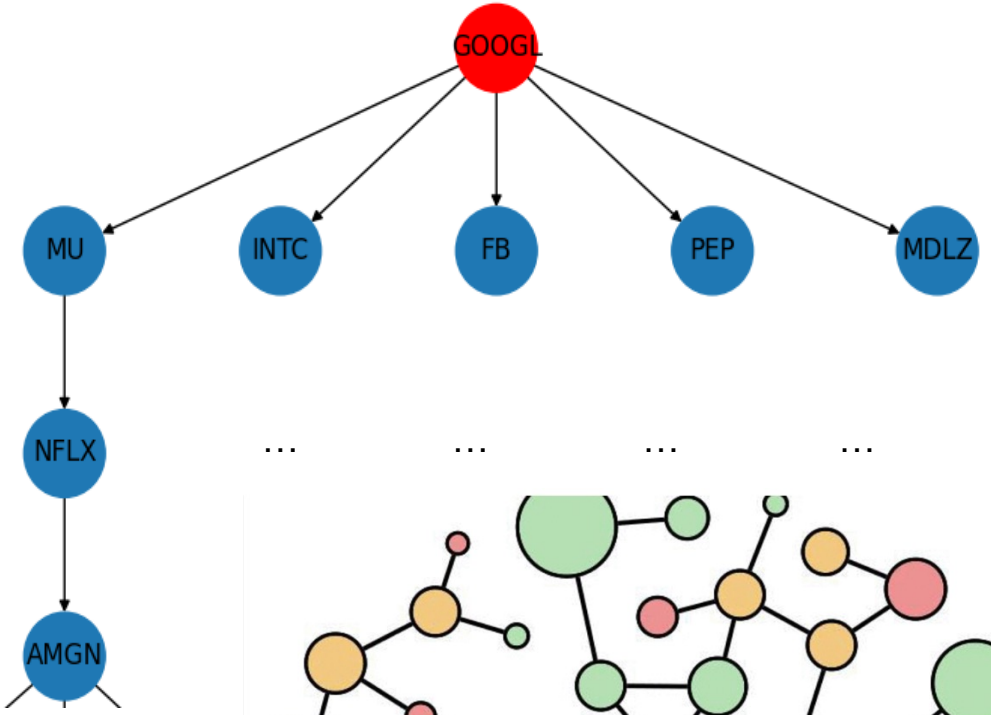
LUISS

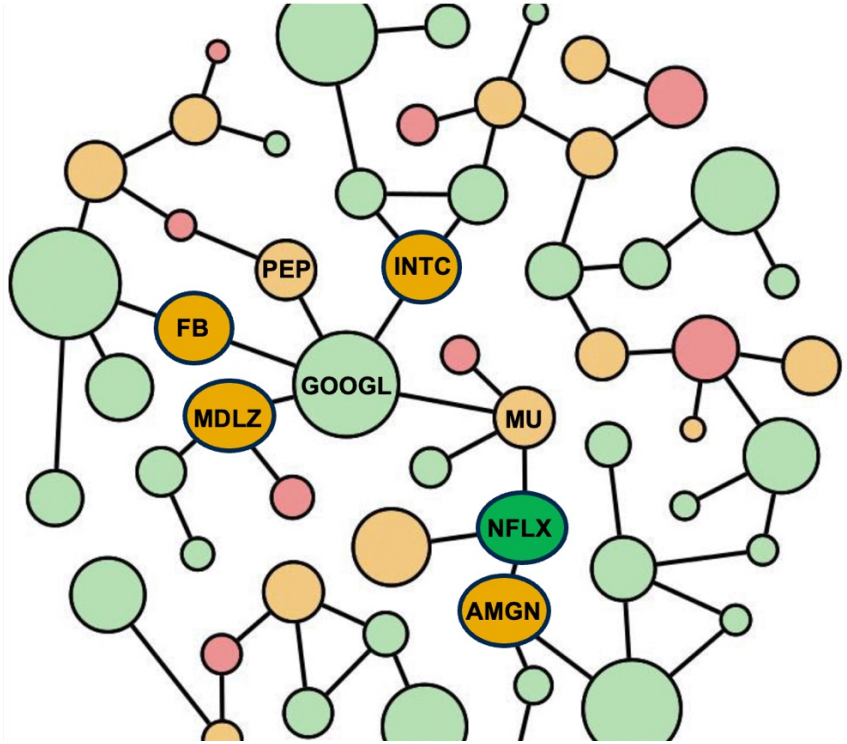
Final TASK!

You have to design and implement a Python code that reads the TXT file, extracting the relevant information, compute the correlation between all pairs of stocks, and store them in a suitable data structure.

The data structure should make it possible to answer queries of the form:

“ Given a stock X, which are the all the correlated stocks at level i? ”





LUISS

Final TASK!

Tips:

- Read the dataset

Stock	Day	Price	Volume
AAPL	371	200	0
TSLA	369	275	13038300
TSLA	370	273	0
TSLA	371	273	0
AAPL	369	197	18526600
AAPL	370	200	0
AAPL	365	191	27862000
TSLA	365	289	8110400
TSLA	366	286	5478900
TSLA	367	292	7929900
TSLA	368	268	23720700
AAPL	366	194	22765700
AAPL	367	195	23271800
AAPL	368	196	19114300

Final TASK!

Tips:

- Read the dataset
- Compute the returns

$$r_s = \begin{cases} (x_n - x_0) / x_0 & if \;\; x_0 > 0 \\ 1 & else \end{cases}$$

Final TASK!

Tips:

- Read the dataset
- Compute the returns
- Compute the correlation using the threshold t

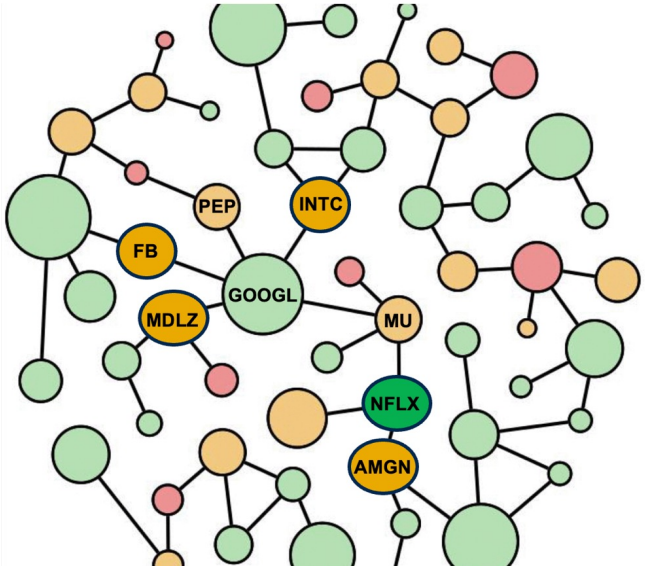
$$c_{ij} = \quad |\, r_i - r_j \,|$$

And $c_{ij} < t\ (thresholds)$

Final TASK!

Tips:

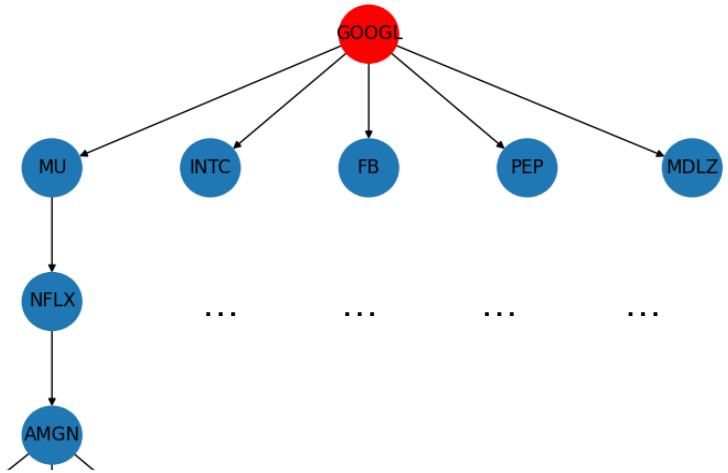
- Read the dataset
- Compute the returns
- Compute the correlation using the threshold t
- Create a suitable data structure for the stock correlation (Tip: Graph)



Final TASK!

Tips:

- Read the dataset
- Compute the returns
- Compute the correlation using the threshold t
- Create a suitable data structure for the stock correlation (Tip: Graph)
- Answer the query by visiting the graph.

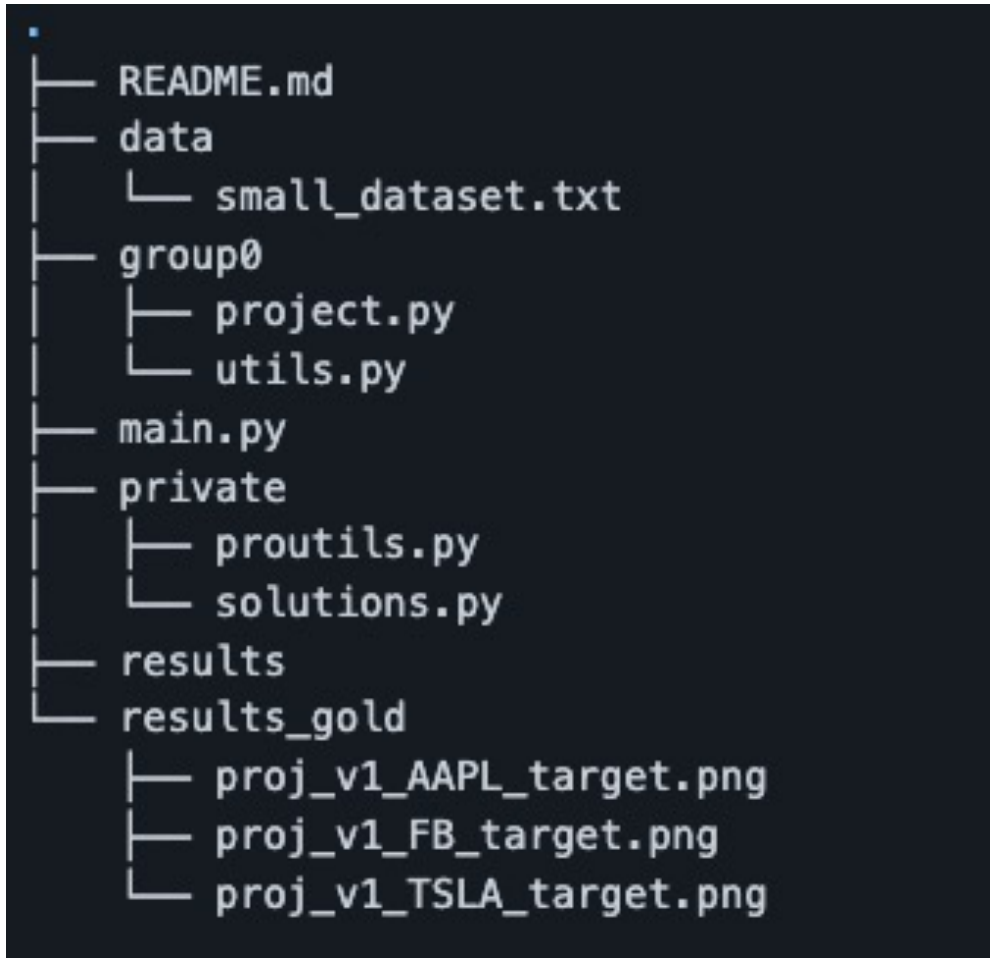


LUISS

Group Project Work : Implementation

We are providing you **the same** skeleton of the code (right picture). You should modify the code we provide, adding the missing parts.

In particular, you have to implement **“group0/project.py”**. But you can also use “**group0/utils.py**” to add needed structures and algorithms.



You can download the code from Luiss-learn

Group Project Work : Implementation

The python file **“group0/project.py”:**

The function prepare is called **once** to load the dataset: it can be used to prepare and read the input file (e.g., *“data/small_dataset.txt”*).

Note: *filename* contains the name of the file to read (e.g., *data/small_dataset.txt) while threshold is a float and must be used to compute correlations.*

```
def prepare(filename : str, threshold : float):
    """ The method is responsable to create the needed data structures
        answer the queries.

    Args:
        filename (str): the input file with the dataset
        threshold (float): a special threshold used to compute the corr
    """
    global your_variables_here
    your_variables_here = {}
```

LUISS

Group Project Work : Implementation

The python file **“group0/project.py”:**

The function **query** implements your query algorithm!

It receives as input the stock name (e.g., *”AAPL”*) and the **corr_level!** it outputs the **ordered (alphabetical order) correlated stocks** at the input level.

The output is a lists of string.

```
def query(stock : str, corr_level : int) -> list:
    """
    Some stocks can be correlated and have a similar behavior.

    This method aims at finding all the correlated stocks respect to the input stock.
    In particular we define the corr_level as follows:
        corr_level = 1 identify all the stocks that are directly correlated to the input sto
        corr_level = 2 identify all the stocks that are NOT directly correlated to the input
        corr_level = 3 identify all the stocks that are NOT directly correlated to the input
        ..
        corr_level = i identify all the stocks that are NOT directly correlated to the input

    Returns:
        list: a list of correlated stocks at corr_level. The list should be in an alphabetic

    E.g., :
        The execution query(GOOGL, 4) may returns: ['AMZN', 'FISV', 'XEL']
        Notice the output is ordered!

    """
    return []
```

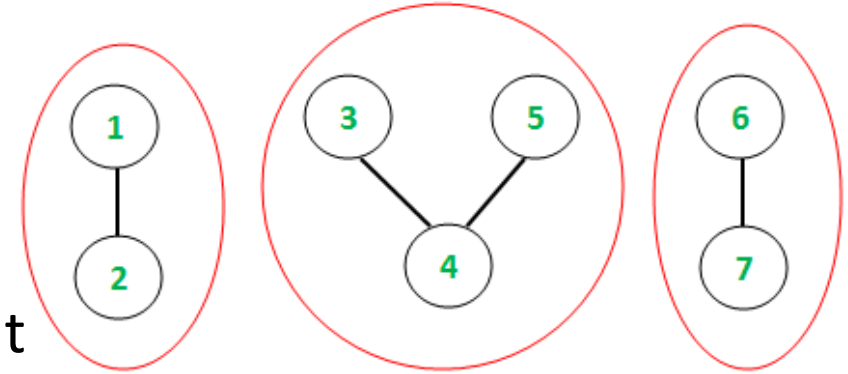
LUISS

Group Project Work : Optional

Correlation graphs highlight inter-correlated stocks!

They are crucial to understand market behaviors and build a good portfolio.

The optional part of this project requires to compute the number of groups (i.e., connected components) that are in the correlation graph.



LUISS

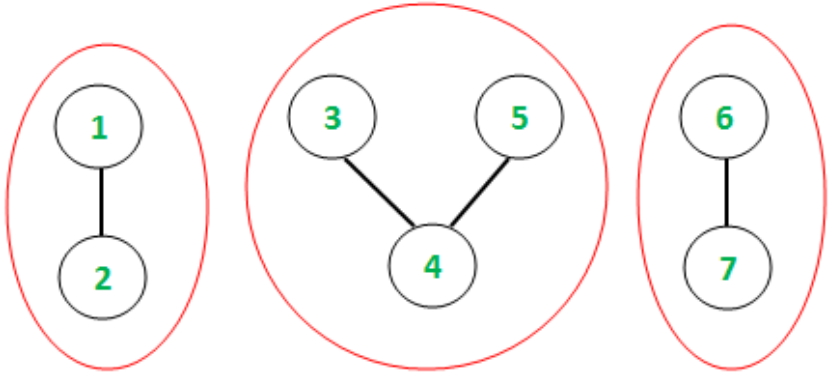
Group Project Work : Optional

The python file **“group0/project.py”:**

The function **num_connected_components** implements the optional query!

If implemented, please enable the tests in the main.py:

```
OPTIONAL_TEST = True
```



The output is an integer (i.e., number of connected components of the correlation graph).

Group Project Work : Execution

Once implemented, you can execute your project using the following command:

“python3 main.py”

This will test your code on some sample queries.

You receive a **text** feedback.

```
Reading file data/medium_dataset.txt
Prepare: 673ms
Query: 1ms
Solution stats for stock TSLA with threshold: 0.1 and corr_level : 5
 IS CORRECT, Congratulations!
Optional query: 1ms
Solution stats for stock TSLA with threshold: 0.1 and corr_level : 5
 IS CORRECT, Congratulations!
Starting large dataset test...
Reading file data/large_dataset.txt
Prepare: 55750ms
Query: 36ms
Solution stats for stock PEP with threshold: 0.05 and corr_level : 5
 IS CORRECT, Congratulations!
Optional query: 36ms
Solution stats for stock PEP with threshold: 0.05 and corr_level : 5
 IS CORRECT, Congratulations!
------------------------------------------------------------------------------------------------------------
Final score: 18 / 18 correct solutions!
```

LUISS

Group Project Work : Execution

Once implemented, you can execute your project using the following command:

“python3 times.py”

Loading and Building data - Time: 35779ms
Query - Time: 75ms

To test the execution time! You are not required to match this times!!

LUISS

Group Project Work

Tools

- For the final release is MANDATORY TO implement all the algorithms/structures from scratch.
- You cannot use external or build-in libraries.

For any doubts contact us by email

Group Project Work

Report:

- The final release is MANDATORY TO PASS the Project, and you must provide a presentation with at most 6 slides.

Describe your algorithmic idea, main implementation details, and experiments. You should try to analyze the asymptotic cost of your implementation.

Group Project Work

Score:

- The project contributes up to 8 points (added to the theory score).
- If you miss the May deadline, the maximum grade is lowered: you can achieve max 6 points (if you deliver the project by the second exam session), max 5 points (third session), max 4 points (fourth - and last - session)
- For **top projects**, we might consider assigning an extra **1-point bonus**.

You should work in groups of 3 students.

LUISS

More details on Luiss-learn and Project guidelines pdf.

Group Project Work

Submission (NEW):

-Please submit a zip file called “**Algorithms _project_final_groupX.zip**”, where X is your group ID assigned.

-Inside the zip file please include both your code “**groupX/**” folder and the **pdf** report file.

Submit the zip through by email using "Algorithms 2022: group X” subject , where X is your group ID assigned by the instructors.

LUISS

More details on Luiss-learn and Project guidelines pdf.

Project-v2 Q/A ?

Thank you!

LUISS