Data Mining
Assignment Report

Ammar Alyousfi (WQD170064)

2 June 2019

Contents

I Introduction **3**

II Tools Used **3**

III The Project Stages **4**

1 Acquisition of the Data **4**
1.1 Data Sources 4
1.2 Web Crawlers 5
1.3 Volume of Crawled Data 7

2 Storage and Management of the Data **7**
2.1 The Star Schema 8

3 Interpretation of the Data **8**
3.1 The-Star Stock-Prices Data 9
3.2 Twitter Data 9
3.3 The-Edge-Markets New Data 12
3.4 The Company-Info Data 12

4 Exploration and Analysis of the Data **13**
4.1 Creating a Time Series Out of the Data 13
4.2 Covariance and Correlation Analysis 13
4.3 Data Reduction Using SAX and PAA 18
4.4 The Relationship Between Stock Price and Sentiment Polarity . 19
4.5 Stock Price Comparison 21
4.6 Last Price Versus Open Price 22
4.7 Clustering 23
4.8 Stock Market Structure 24

5 Building an Investment Recommendation Program **25**

IV Conclusion **27**

Part I
Introduction

In this assignment, we will build a data mining project to apply predictive and perspective analysis on stock market data to solve real-world problems. The three C's (creating, connecting, and computing) are considered in the implementation of this project.

We start with acquiring relevant data from multiple sources. Then, we find a way to manage and store this data. After that, we process this data and perform analysis and exploration on it to discover interesting insights and patterns. Subsequently, we interpret the analysis results of the data. And finally, we build a model to provide the user with a recommendation for the optimal stock options regarding profits, return on investment, and user preferences. Moreover, we communicate the results of our work in a clear, understandable, and attractive way.

Part II
Tools Used

To do this project and obtain the results, we used Python programming language with many of its data-science and data-analysis libraries: we used Pandas library which is the leading Pyhton library for data representation, data analysis, and data manipulation; we used NumPy library which is a Python library for performing calculations and mathematical functions of arrays; we used Matplotlib which is one of the most popular Python libraries for data visualization; we used TextBlob which is a Python library for text analysis including sentiment analysis; we used Scikit-learn which is one of the most used Python libraries for building supervised and unsupervised machine learning models; we also used Requests library which is a HTTP Python library that can be used to download web pages for web crawling; we used Beautiful Soup which is a Python library for parsing HTML and XML documents; we also used python-twitter library that facilitates connecting to and dealing with Twitter API; we used PyMySQL Python library too to connect to MySQL database and perform related operations.

Along with that, we used many of Python built-in libraries like time to control the time of execution of Python code, re to use regular expressions in dealing with text objects, and other libraries. We used Jupyter Notebook to write the code, show its output, and explain about it in the same document.

Other than Python and its libraries, we used MySQL database system to create databases and tables to store our data; we also used MySQL Workbench which allows us to see our database and tables and perform SQL queries on them using a graphical interface.

In the following sections of this report, we will describe and interpret the processes and results of the stages of our project with code and visualizations when possible.

Part III
The Project Stages

In this part, we will describe the stages of this project with the results of each stage.

1 Acquisition of the Data

As the first step in our project, we need to collect the required data before we can move to next steps in the project. In our case, we will crawl our data from the web using web crawlers. Our web crawlers are programs that can download web pages from the Internet, process and parse them, and then store them for later use. Python programming language—along with its libraries like Requests, Beautiful Soup, and Pandas—were used to build the web crawlers of this project.

1.1 Data Sources

In this project, multiple data sources related to the stock market were used in order to have multiple views and perspectives on the subject of the project. We crawled data from the following sources:

- Stock prices from The Star website
- Stock prices from MalaysiaStock.Biz website
- Tweets from Twitter social media platform
- News articles and headlines from The Edge Markets website
- News articles and headlines from The Star website
- Quarterly financial data from MalaysiaStock.Biz website
- Company information (like ROE and market capital) from MalaysiaStock.Biz website
- Malaysian stock indices from Investing.com website

1.2 Web Crawlers

For each data source mentiond above, a dedicated Python string was written to crawl data from that source, manipulate and store it for later use. The crawler scripts are available on the Github repository associated with this project in "Web Crawlers" directory. Here, we will mention the name of each crawler script and explain what it does.

Crawling of stock prices from The Star website

This crawler script is available under the name "The Star stock price crawler.py" inside "The Star stock price crawler" directory. This script crawls stock-prices data for all KLSE companies available on The Star website which are more than 1,800 companies. It automatically downloads the HTML web pages of each company, parses them, extracts the required data from them, and finally writes the extracted data into text files.

The data extracted for each company includes the last (or closing) stock price of the company at the time of crawling, the opening stock price of the company at the time of crawling, the highest price of the day of the company at the time of crawling, the lowest price of the day of the company at the time of crawling, and the update date and time of the prices.

We uploaded this script to a virtual machine rented on Amazon Web Services cloud platform and scheduled it to run six times each day using Crontab system service. This way, the script will automatically run several times a day without any manual intervention. Whenever we need the crawled data, we can download them from the virtual machine using, for example, FileZilla program which allows downloading files using FTP protocol using a graphical interface.

Crawling of stock prices from MalaysiaStock.Biz website

This crawler script is available under the name "Malaysiastockbiz stock price crawler.py". This script crawls stock-prices data for all KLSE companies available on MalaysiaStock.Biz website which are more than 1,800 companies. It automatically downloads the HTML web pages of each company, parses them, extracts the required data from them, and finally writes the extracted data into text files.

The data extracted for each company includes the last (or closing) stock price of the company at the time of crawling, the opening stock price of the company at the time of crawling, the highest price of the day of the company at the time of crawling, the lowest price of the day of the company at the time of crawling, and the update date of the prices.

As with we did for The Star stock-prices crawler, we uploaded this script to a virtual machine rented on Amazon Web Services cloud platform and scheduled it to run six times each day using Crontab system service. This way, the script will automatically run several times a day without any manual intervention. Whenever we need the crawled data, we can download them from the virtual

machine using, for example, FileZilla program which allows downloading files using FTP protocol using a graphical interface.

Crawling tweets from Twitter social media platform

This crawler script is available under the name "Twitter crawler.py". This script crawls tweets that mention any of the KLSE companies which are more than 1,800 companies. It connects to Twitter API using "pyhton-twitter" library and then it uses the "search" function of the API to search for tweets posted between two specified dates that contain the name of any KLSE company. When it finds such a tweet, it extracts many related information from the tweet and finally it writes all found tweets with their information to a CSV file using Pandas library.

For each relevant tweet, the script extracts the tweet text, the tweet language, the tweet poster, the tweet ID, the creation date of the tweet, the hashtags included in the tweet, and other information.

Crawling of news articles and headlines from The Edge Markets website

This crawler script is available under the name "The Edge Markets news crawler.py". This script crawls news articles from the Corporate section on The Edge Markets website that mention any of the KLSE companies which are more than 1,800 companies.

For each relevant article, the scripts extracts the article headline, the article body text, the article URL, and the date of posting the article. The script writes the extracted data to a CSV file using Pandas library.

Crawling of news articles and headlines from The Star website

This crawler script is available under the name "The Star news crawler.py". This script crawls news articles from the Business section on The Star website.

For each crawled article, the scripts extracts the article headline, the article body text, the article URL, the article category, and the date and timestamp of posting the article. The script writes the extracted data to a CSV file using Python CSV library.

Crawling of quarterly financial data from MalaysiaStock.Biz website

This crawler script is available under the name "Quarterly financial data crawler.py". It crawls quarterly financial data about many KLSE companies. The crawled data includes the financial year, the financial quarter, the revenue, the net profit, the PBT, the EPS, the divident, and the NTA. The script writes the crawled data into a text file.

Crawling of company information from MalaysiaStock.Biz website

This crawler is available under the name "Company Info Scraper.ipynb". It is not just a Python script like the crawlers above. Instead, it is a Python code written inside a Jupyter Notebook. It is suitable to write this crawler in a Jupyter Notebook because it needs to be run one time to get information about KLSE company.

For each company whose information are available on the target website, this crawler obtains the following information on the company: the company symbol, the company full name, the last-financial-quarter date, the last-financial-quarter revenue, the last-financial-quarter net profit, the board of the company, the sector of the company, the sub-sector of the company, the market capital of the company, the number of share of the company, the ROE of the company, and whether the company is Shariah compliant or not.

The crawler writes the obtained data into a CSV file using Pandas.

Crawling of Malaysian stock indices from Investing.com website

This crawler script is available under the name "Malaysian stock indices crawler.py". It crawls price data and the datetime of the price data for Malaysian stock indices. The script extracts data for "KLCI", "Malaysia ACE", "FTSE BM Mid 70", and "Malaysia Top 100" indices and then writes the output to a CSV file.

1.3 Volume of Crawled Data

As of the date of writiing this report, we have more than 1.5 million rows of stock-prices data crawled from The Star and MalaysiaStock.Biz websites. Also, we have more than 18,000 tweets crawled from Twitter talking about different KLSE companies. We also have crawled more than 34,000 news articles talking about KLSE companies from The Edge Markets website.

2 Storage and Management of the Data

We wanted to store and organize the crawled data into a MySQL database. To do so, we first created a MySQL database and then, we created a table in the database for each data source we wanted to store its data in the database. The SQL commands used to create the database and the tables are available inside a file named "SQL commands.txt" in the Github repository of this project. For example, the SQL command used to create a table for the stock prices of The Star website is:

```
CREATE TABLE TheStarData ( ID INT NOT NULL
   AUTO_INCREMENT PRIMARY KEY, Datetime VARCHAR(50)
   NOT NULL, CompanySymbol VARCHAR(50) NOT NULL,
   OpenPrice FLOAT, HighPrice FLOAT, LowPrice FLOAT,
   LastPrice FLOAT );
```

Although not all of the data is stored into the MySQL database, we can draw the star schema of our data to show how different tables interact with each other.

For each data source, a dedicated Python script was written to take the crawled data, structure it, and store it into the corresponding MySQL table. These scripts can be found in "Stoage-in-database scripts" directory in the Github repository of this project.

2.1 The Star Schema

Figure 1 shows the star schema of our data.

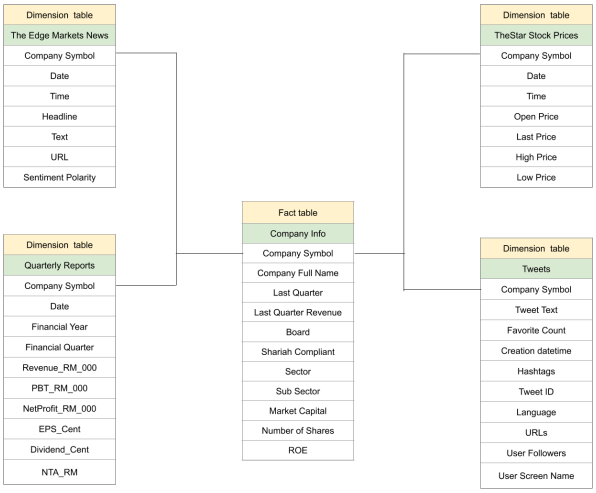

Figure 1: The star schema of our data

We can see that tables connect with each other based on the company symbol which is a unique symbol for each company. When we want to retrieve data about one company, we can use its symbol to get information about it from all tables.

3 Interpretation of the Data

After crawling the data, we looked at the crawled data to interpret it and see if it was suitable for meaningful analysis. In this section, we will show a sample of the crawled data for the data sources that were used in our analysis.

3.1 The-Star Stock-Prices Data

The stock prices data obtained from The Star website looks like the table shown in Figure 2.

	company_symbol	open_price	last_price	high_price	low_price	date	time
300	CMSB-C14	nan	0.085	nan	nan	18-May-2019	05:42
301	CMSB-C15	0.05	0.05	0.05	0.05	18-May-2019	05:42
302	CNASIA	0.43	0.43	0.44	0.43	18-May-2019	05:42
303	CNI	0.07	0.07	0.07	0.07	18-May-2019	05:42
304	CNOUHUA	0.045	0.045	0.045	0.045	18-May-2019	05:42
305	COASTAL	1.05	1.04	1.05	1.04	18-May-2019	05:42
306	COCOLND	1.96	1.96	1.96	1.96	18-May-2019	05:42
307	COMCORP	0.08	0.07	0.08	0.065	18-May-2019	05:42
308	COMFORT	0.83	0.825	0.84	0.825	18-May-2019	05:42

Figure 2: A sample of The-Star stock-prices data

Each row specifies the company symbol, the last (closing) price, the open price, the highest price of the day, the lowest price of the day, and the date and time of the prices.

There are some missing values in the prices of the crawled data as can be seen in the first row of the sample table. This happened because The Star website didn't show that data at that time of crawling. The most important price data for us is the last-price data and fortunately, the last-price field doesn't include any missing value.

3.2 Twitter Data

The tweets data obtained from Twitter looks like the table shown in Figure 3.

	company_fullname	tweet_text	favorite_count	created_at	hashtags	tweet_id	lang	urls	user_followers	user_screen_name	crawling_time
0	AIRASIA X	RT @theworldindex: World's Best Low-Cost Airlines 2018: 1. AirAsia ...	0	Mon Mar 11 23:52:05 +0000 2019	nan	1.10526e+18	en	nan	13770	alexisandmartin	2019-03-16|09:12
1	AIRASIA X	RT @theworldindex: World's Best Low-Cost Airlines 2018: 1. AirAsia ...	0	Mon Mar 11 23:42:34 +0000 2019	nan	1.10525e+18	en	nan	526	AminSkywalker	2019-03-16|09:12
2	AIRASIA X	RT @theworldindex: World's Best Low-Cost Airlines 2018: 1. AirAsia ...	0	Mon Mar 11 23:36:09 +0000 2019	nan	1.10525e+18	en	nan	193	ZiqaFrd	2019-03-16|09:12
3	AIRASIA X	RT @theworldindex: World's Best Low-Cost Airlines 2018: 1. AirAsia ...	0	Mon Mar 11 23:33:49 +0000 2019	nan	1.10525e+18	en	nan	166	hanz_abd_manan	2019-03-16|09:12
4	AIRASIA X	RT @theworldindex: World's Best Low-Cost Airlines 2018: 1. AirAsia ...	0	Mon Mar 11 23:33:38 +0000 2019	nan	1.10525e+18	en	nan	293	shahiraanadia	2019-03-16|09:12

Figure 3: A sample of Twitter data

When we inspected Twitter data, we found that many times, the tweets contain a name that match one of KLSE companies but has different meaning. For example, when we searched for tweets that mention the "White Horse" KLSE company, we got a tweet shown in Figure 4. It is clear that the tweet is not talking about the KLSE "White Horse" company although the words "white horse" appear in the tweet.

Figure 4: Irrelevant tweet example 1

Another example is when we searched for tweets containing the name of "MKH" KLSE company, we got a tweet shown in Figure 5. Again, it is clear that the tweet is talking about something different than the KLSE company.

Figure 5: Irrelevant tweet example 2

Another example is when we searched for tweets containing the name of "MGB" KLSE company, we got a tweet shown in Figure 6. Again, it is clear that the tweet is talking about something different than the KLSE company.

Figure 6: Irrelevant tweet example 3

As a result, we decided to discard Twitter data because we thought that it would introduce noise in the analysis.

3.3 The-Edge-Markets New Data

The news data obtained from The Edge Markets corporate news looks like the table shown in Figure 7.

	company_symbol	headline	url	datetime	text	in_headline	is_fullname
0	BURSA	Foreign funds turn net sellers as outflows exceed inflows	https://www.theedgemarkets.com/article/foreign-funds-turn-net-sellers-outflows-exceed-inflows	March 29, 2019 21:28 pm +08	KUALA LUMPUR (March 29): Foreign institutions turned net sellers in th...	No	Yes
1	BURSA-C18	Foreign funds turn net sellers as outflows exceed inflows	https://www.theedgemarkets.com/article/foreign-funds-turn-net-sellers-outflows-exceed-inflows	March 29, 2019 21:28 pm +08	KUALA LUMPUR (March 29): Foreign institutions turned net sellers in th...	No	Yes
2	BURSA-C20	Foreign funds turn net sellers as outflows exceed inflows	https://www.theedgemarkets.com/article/foreign-funds-turn-net-sellers-outflows-exceed-inflows	March 29, 2019 21:28 pm +08	KUALA LUMPUR (March 29): Foreign institutions turned net sellers in th...	No	Yes
3	BURSA-C21	Foreign funds turn net sellers as outflows exceed inflows	https://www.theedgemarkets.com/article/foreign-funds-turn-net-sellers-outflows-exceed-inflows	March 29, 2019 21:28 pm +08	KUALA LUMPUR (March 29): Foreign institutions turned net sellers in th...	No	Yes
4	BURSA-C22	Foreign funds turn net sellers as outflows exceed inflows	https://www.theedgemarkets.com/article/foreign-funds-turn-net-sellers-outflows-exceed-inflows	March 29, 2019 21:28 pm +08	KUALA LUMPUR (March 29): Foreign institutions turned net sellers in th...	No	Yes

Figure 7: A sample of The-Edge-Markets news data

Each row specifies the company symbol, the headline and the text of the news article, the date and time of the article, the URL of the article, whether the company identifier (symbol or full name) appeared in the headline or in the body text of the article, and whether the company identifier is the company symbol or the company full name.

For this corporate news data, when a news article mentions a KLSE company name, we know that the article is definitely talking about the KLSE company in contrary to the case of Twitter data discussed above.

3.4 The Company-Info Data

The company-info data obtained from MalaysiaStock.Biz website looks like the table shown in Figure 8.

	company_full_name	company_symbol	last_quarter	last_quarter_revenue_000	last_quarter_net_profit_000	board	shariah_compliant	sector	sub_sector	market_capital	num_of_share	roe
0	ASTRAL ASIA BERHAD	AASIA	31 Dec 2018	2,673	692	MAIN	Compliant	Plantation	Plantation	102.30m	659.98m	-2.70
1	AIRASIA X BERHAD	AAX	31 Mar 2019	1,168,804	43,333	MAIN	Compliant	Consumer Products & Services	Travel, Leisure & Hospitality	933.33m	4.148b	-39.42
2	ABF MALAYSIA BOND INDEX FUND	ABFMY1	30 Sep 2018	28,446	0	ETF	Non-Compliant	Exchange Traded Fund-bond	Bond Fund	1.501b	1.265b	-
3	ABLEGROUP BERHAD	ABLEGRP	31 Dec 2018	1,413	-70	MAIN	Compliant	Industrial Products & Services	Building Materials	18.47m	263.90m	-2.18
4	ALLIANCE BANK MALAYSIA BERHAD	ABMB	31 Dec 2018	418,401	148,932	MAIN	Non-Compliant	Financial Services	Banking	5.821b	1.548b	9.67

Figure 8: A sample of the company-info data

Each row specifies the company symbol, the company the company full name, the last-financial-quarter date, the last-financial-quarter revenue, the last-financial-quarter net profit, the board of the company, the sector of the company, the sub-sector of the company, the market capital of the company,

the number of share of the company, the ROE of the company, and whether the company is Shariah compliant or not.

4 Exploration and Analysis of the Data

In this section, we will describe the data-exploration and data-analysis work done on the data.

4.1 Creating a Time Series Out of the Data

This is one of the most important steps in the project. We created a time-series table for each company. This table contains the stock price and average sentiment polarity for the company for each day. Figure 9 shows a sample of the time series of "TM" company.

	company_symbol	open_price	last_price	high_price	low_price	date	news_sentiment_1	headlines_sentiment_1
3514	TM	3.09	3.11	3.12	3.06	2019-02-24	0	0
5368	TM	3.11	3.05	3.12	3.05	2019-02-25	0	0
7222	TM	3.06	3.02	3.06	2.97	2019-02-26	0	0
9076	TM	2.97	2.96	2.98	2.91	2019-02-27	-0.00625	-0.00833333
10930	TM	2.95	3.05	3.08	2.94	2019-02-28	-0.0388889	-0.0518519
12784	TM	3.05	3.03	3.05	2.99	2019-03-03	0	0
14638	TM	2.99	3.09	3.1	2.99	2019-03-04	-0.0689394	0
16492	TM	3.08	3.19	3.21	3.07	2019-03-05	0.125	0
18346	TM	3.19	3.24	3.28	3.15	2019-03-06	0	0

Figure 9: A sample of the time series table of TM company

We will explain later how the sentiment analysis was done.

4.2 Covariance and Correlation Analysis

We calculated the covariance and correlation between the stock-prices (specifically, the last price) time series of KLSE companies to see how the stock prices of KLSE companies are related to each other.

For covariance, we used Pandas library to calculate it between each two KLSE companies. And because the number of companies we are dealing with is large (1800+ companies), we need an efficient and convinient way to look at the covariances of KLSE companies. One of the effective ways to visualize such a thing is the heatmap graph. The heatmap of the covariances of 50 KLSE companies is shown in Figure 10.

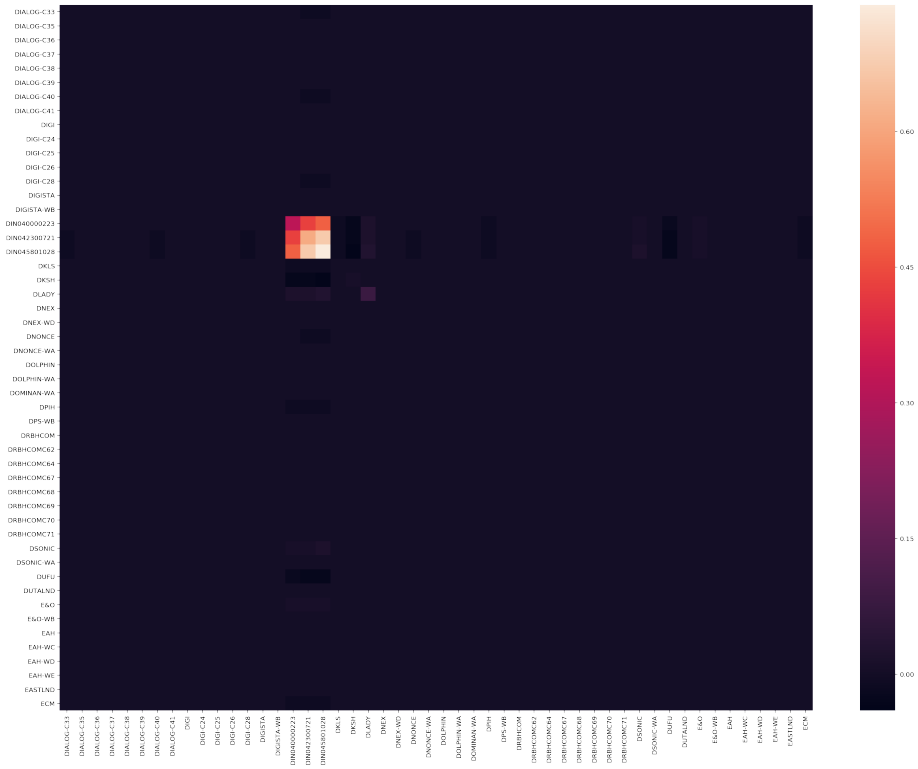

Figure 10: The covariance heatmap of stock prices

The lighter the color of the cell in this heatmap, the higher the covariance between the two corresponding companies.

Correlation is the normalized version of covariance. Correlation value ranges between -1 and +1 where -1 indicates the highest negative correlation and +1 indicates the highest positive correlation; a correlation of 0 indicates that there is no correlation at all. As with covariance, Figure 11 shows the heatmap of correlation for 50 KLSE companies.

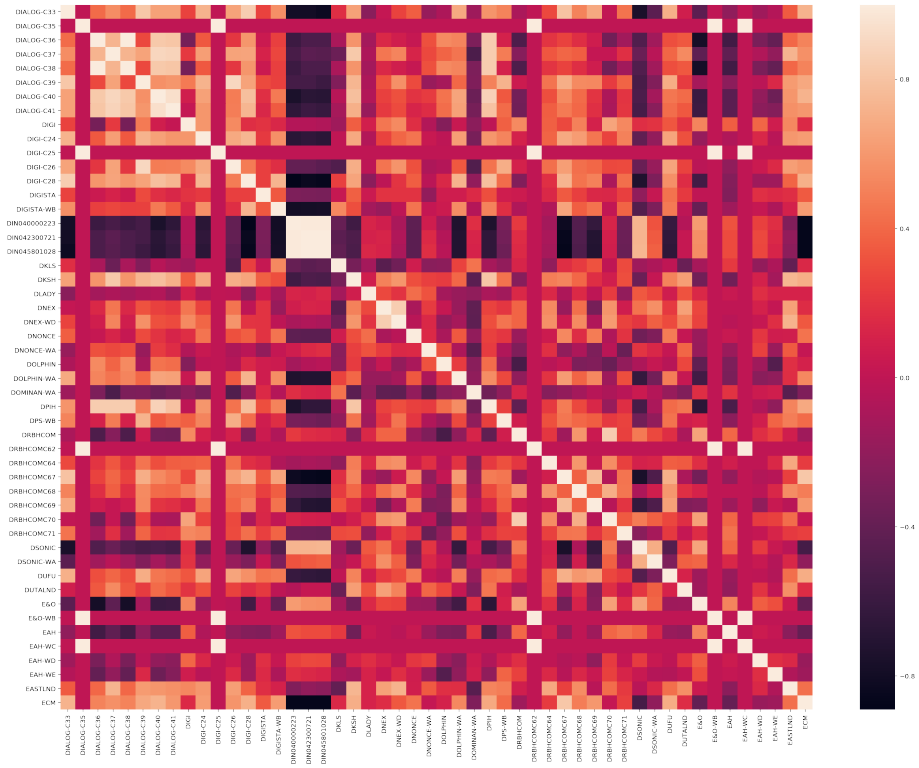

Figure 11: The correlation heatmap of stock prices

In this heatmap, darker colors reflect more negative correlation and lighter colors reflect more positive correlation. We can see that there are some companies that have really high positive and negative correlations between each other.

To find companies that have high positive correlation, a Python code was written which scans the correlation matrix searching for companies with high positive correlation. Figure 12 shows some of the results of this code. It shows companies that have a correlation of $+1$ between each other which is the highest positive correlation possible.

	Company 1	Company 2	Correlation
89950	HIBISCS-CL	SUPERMX-C53	1
89953	HIBISCS-CL	UMW-C10	1
89954	HIBISCS-CL	UNISEM-C13	1
43995	DAIMAN	RHBBANK-C8	1
43996	DAIMAN	YINSON-CV	1
45993	DIALOG-C36	FBMKLCI-C5J	1
59687	FBMKLCI-C5J	HAPSENG-CX	1
185875	SIME-C27	SIMEPROP-CJ	1
137897	MHB-C13	SIME-C25	1
142917	MRCB-C32	TOPGLOV-C33	1
6826	AIRASIAC65	MRCB-C32	1
46691	DIALOG-C38	HAPSENG-CX	1
106654	IJM-C22	MAYBANKC45	1
84400	GTRONIC-CT	IJM-C23	1
56130	EKOVEST-CR	WCT-C14	1
56120	EKOVEST-CR	TM-C33	1
56121	EKOVEST-CR	TOPGLOV-C36	1
56134	EKOVEST-CR	XINQUAN	1
56133	EKOVEST-CR	XINGHE-WA	1

Figure 12: Companies with high positive correlation

Similarly, a Python code was used to retrieve companies with high negative correlation. Some of the results of running this code is shown in Figure 13.

	Company 1	Company 2	Correlation
68070	MRCB-C32	YINSON-CV	-1
2796	AIRASIAC65	YINSON-CV	-1
84337	RHBBANK-C8	TOPGLOV-C33	-1
103176	TOPGLOV-C33	YINSON-CV	-1
2795	AIRASIAC65	RHBBANK-C8	-1
19811	DAIMAN	MRCB-C32	-1
39849	GTRONIC-CT	MISC-C21	-1
33272	FSBM	YFG	-1
33271	FSBM	XINQUAN-WA	-1
33270	FSBM	XINQUAN	-1
33269	FSBM	XINGHE-WA	-1
33268	FSBM	WINTONI-WA	-1
33262	FSBM	VIVOCOM-WD	-1
33273	FSBM	YINSON-CU	-1
33267	FSBM	WINTONI	-1
33274	FSBM	YTL-C19	-1
33266	FSBM	WCT-C14	-1
33265	FSBM	VS-C7	-1
33264	FSBM	VS-C6	-1

Figure 13: Companies with high negative correlation

To show actual examples of companies whose stock prices are highly posi-

tively correlated, we plotted the stock prices of some of these companies. Figure 14 shows a plot of the stock prices of two companies with high positive correlation.

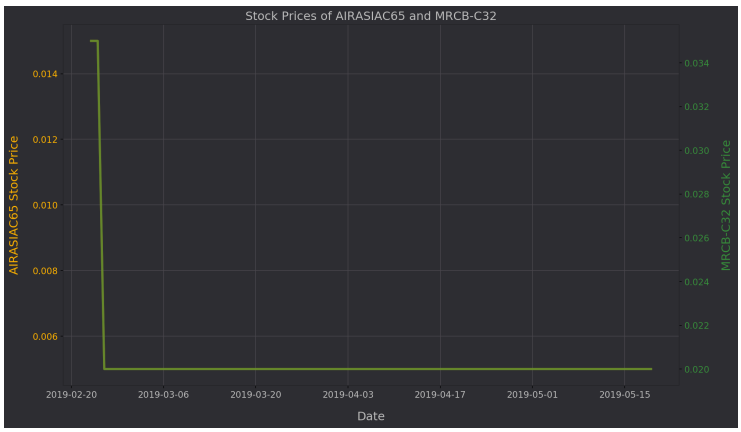

Figure 14: Stock prices of two companies with high positive correlation

We can see that the two stock prices move perfectly together due to the very high positive correlation.

Similarly, Figure 15 shows a plot of stock prices of two companies with high negative correlation.

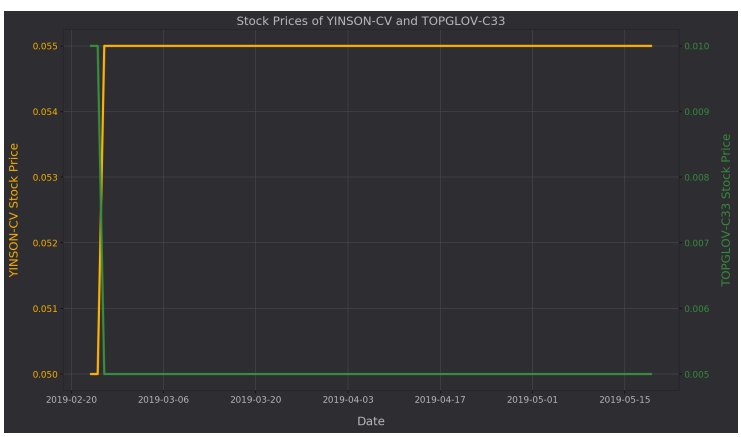

Figure 15: Stock prices of two companies with high positive correlation

We can see that the stock prices of these two companies move in opposite ways: when the price of one of these two companies decreases, the price of the other company increases and vice versa.

4.3 Data Reduction Using SAX and PAA

Data reduction techniques, SAX and PAA, were applied on the stock-prices data of KLSE companies to obtain a representation that is smaller in size. To see an example of data reduction using SAX and PAA, we plot the original stock-price time series of the "DIGI" KLSE company as shown in Figure 16. The results of applying PAA and SAX are shown in Figure 17 along with the original time series.

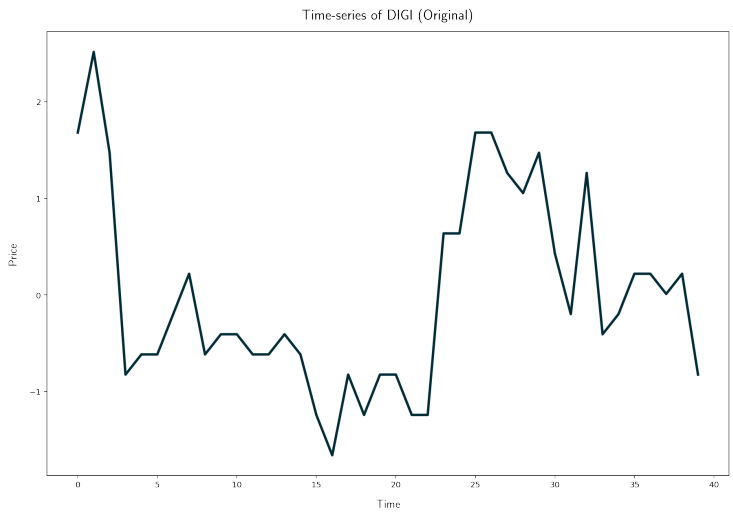

Figure 16: DIGI original stock-price time series

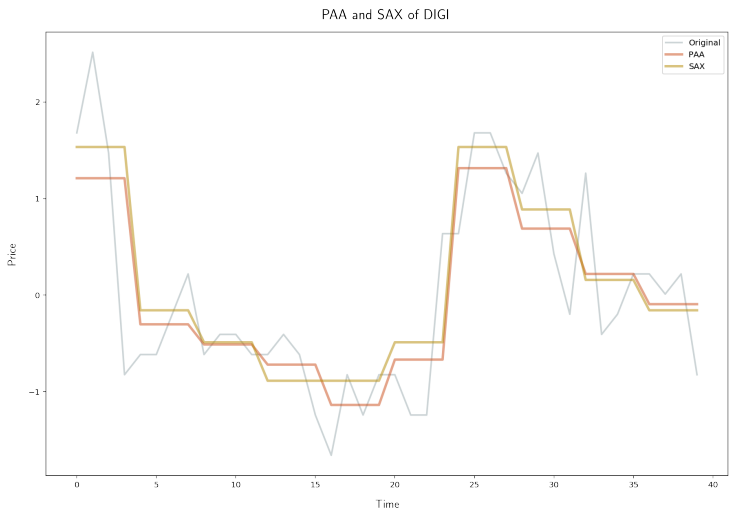

Figure 17: PAA and SAX representations of DIGI stock-price time series

PAA and SAX are useful techniques to use when the data to be analyzed is large so that a smaller version in size is obtained from the original data. But since our data is already small in size and can be easily stored and analyzed, we don't need to apply these reduction techniques for our analysis.

4.4 The Relationship Between Stock Price and Sentiment Polarity

For each KLSE company, we analyzed the daily sentiment polarity regarding this company using the news headlines about it. After that, we calculated the average polarity of news for each company for each day as seen in Figure 9.

TextBlob Python library was used to perform sentiment analysis and get the sentiment polarity. The sentiment polarity is a number between -1 and +1 that represents how positive the text is. If the text is very positive, polarity is 1; if the text is highly negative, the polarity is -1.

After obtaining the average sentiment polarity for each company for each day, and combining this with the stock price (last price) of each company for each day, we can plot the stock price and the sentiment polarity of a company over time to see how the price and sentiment are related.

Figures 18, 19 and 20 show some examples of the stock prices and sentiment polarities of some companies plotted together over time.

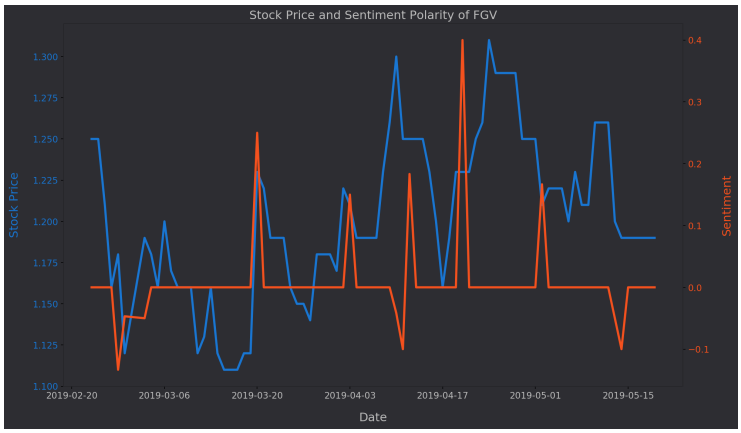

Figure 18: An example of the relationship between stock price and sentiment polarity of a KLSE company

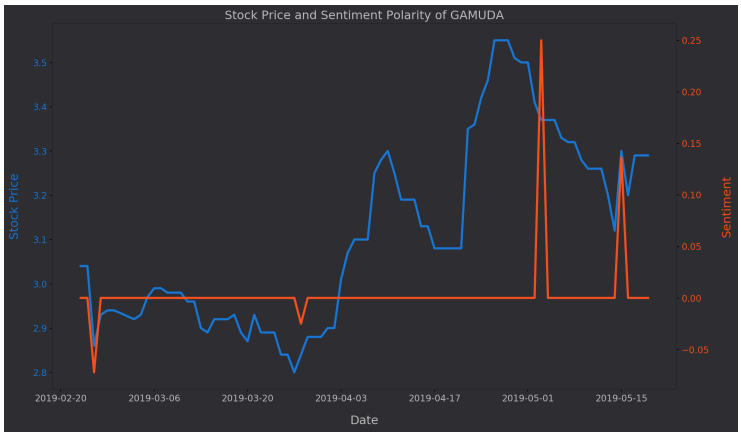

Figure 19: An example of the relationship between stock price and sentiment polarity of a KLSE company

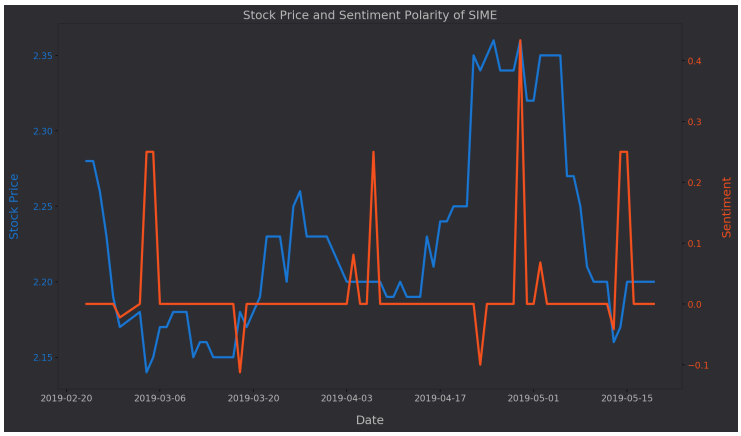

Figure 20: An example of the relationship between stock price and sentiment polarity of a KLSE company

We can see that in Figures 18 and 19, there is a relationship between the stock price and sentiment polarity: we see spikes in sentiment and price happening in close days. However, in Figure 20, we see that the stock-price plot and the sentiment plot are not in sync most of the time. This might be due to some limitations in text and sentiment analysis or might be because the relatively small amount of data used.

As a disclaimer, we don't say that the sentiment analysis done is perfect; we don't say that it reflects the actual sentiment around the KLSE companies.

4.5 Stock Price Comparison

Using the time-series data that we have, we wrote a Python function to perform a comparison between the stock prices of two KLSE companies in order to see how the prices of these two companies change over time. Figures 21 and 22 show some examples of such stock price comparisons.

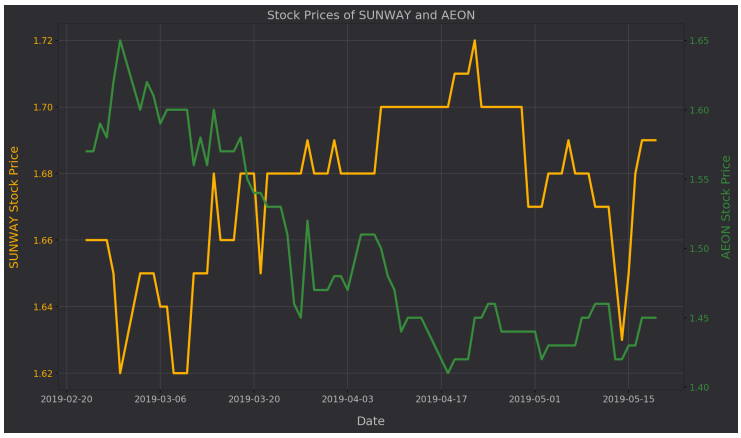

Figure 21: An example of the relationship between the stock prices of two KLSE companies

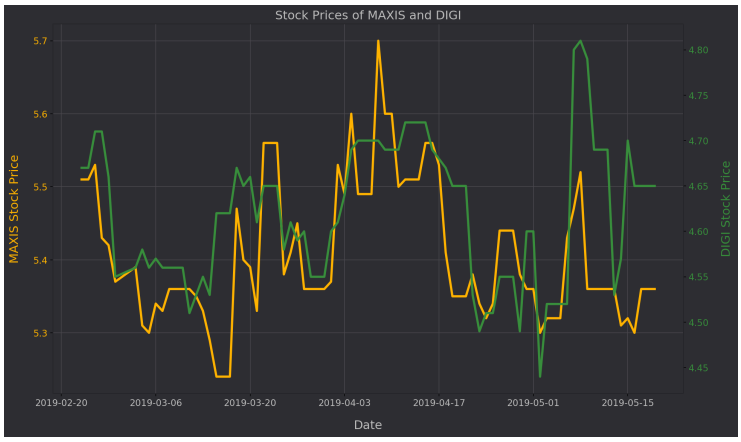

Figure 22: An example of the relationship between the stock prices of two KLSE companies

In Figure 22, where the stock prices of "MAXIS" company and "DIGI" company are plotted together, we can see some correlation between the prices of these two companies. This might be due to the fact that both of these companies are telecommunication companies.

4.6 Last Price Versus Open Price

Using the time series data, we can also plot the last price and open price of some company over time to see how they are related to each other. Figure 23 shows an example of this plot.

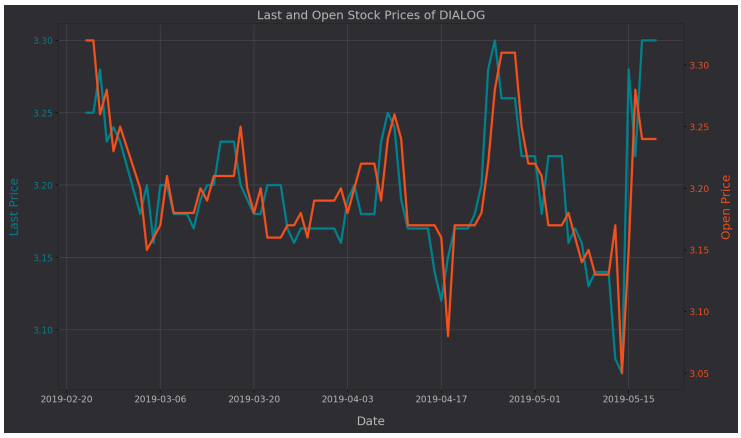

Figure 23: An example of the relationship between the last price and open price of a KLSE company

4.7 Clustering

We applied K-means clustering algorithm on company data to find groups of similar companies. To do so, we needed to extract features for each company from its stock-prices time-series because it is not meaningful to perform clustering using the time series data. The extracted features try to capture the most important information embedded in the time-series data.

Figure 24 shows a sample of the extracted-features table.

	company	mean_last_price	std_last_price	last_minus_open	minimum_last	maximum_last	last_price_skewness	last_argmax
0	3A	0.906667	0.0373145	0.00154618	0.84	0.98	-0.460247	29
1	A50CHIN-C22	0.0989286	0.0720194	-0.0804945	0.055	0.31	1.44093	7
2	A50CHIN-C24	0.79381	0.0686734	0.0242857	0.5	0.915	-2.40879	7
3	A50CHIN-C26	0.531607	0.0799592	0.0788799	0.38	0.615	-0.292595	42

Figure 24: A sample of the table of features extracted for clustering

Figure 25 shows the clustering results of KLSE companies. The number of clusters was chosen to be 3. The two extracted features that were used for clustering are: average last price and the standard deviation of the last price. Figure 25 shows a plot of the aforementioned features for each company. Each company has a color that reflects its cluster. We have 3 colors for 3 clusters.

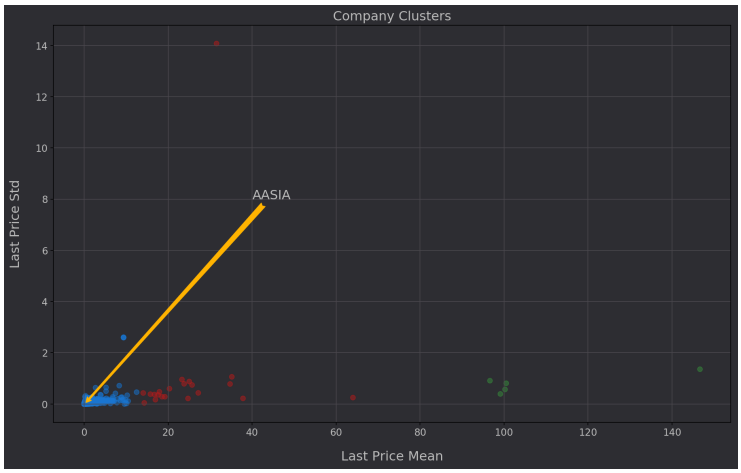

Figure 25: The results of K-means clustering

Figure 26 is similar to Figure 25 but shows the area of each cluster in different color and it shows the centers of the clusters as well.

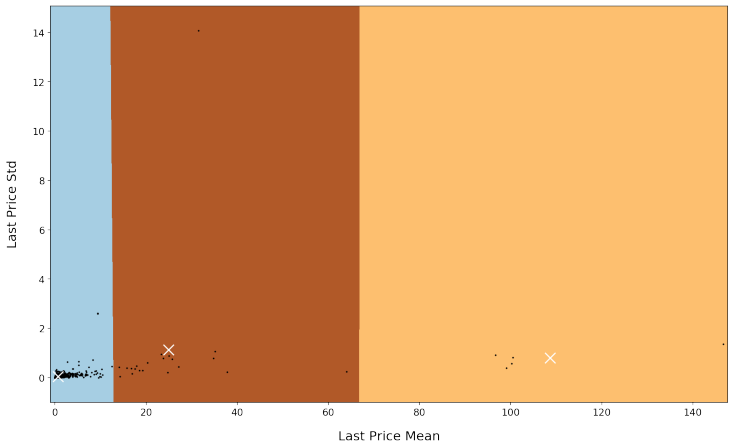

Figure 26: The results of K-means clustering with cluster areas and centers

The results of clustering can be used to understand the common attributes between companies in the same cluster whigh might lead to a better understanding of the market.

4.8 Stock Market Structure

Using a program from Scikit-learn, we created a plot that shows the structure of the market using data of 50 KLSE companies. To produce this plot, the

following techniques were used:

- Sparse inverse covariance to find which stocks are correlated conditionally on the others.
- Affinity Propagation clustering to group together stocks that behave similarly.
- Manifold learning to retrieve 2D embedding for visualization purposes.

In the structure plot which is shown in Figure 27:

- Cluster labels are used to define the color of the nodes
- The sparse covariance model is used to display the strength of the edges
- The 2D embedding is used to position the nodes in the plan

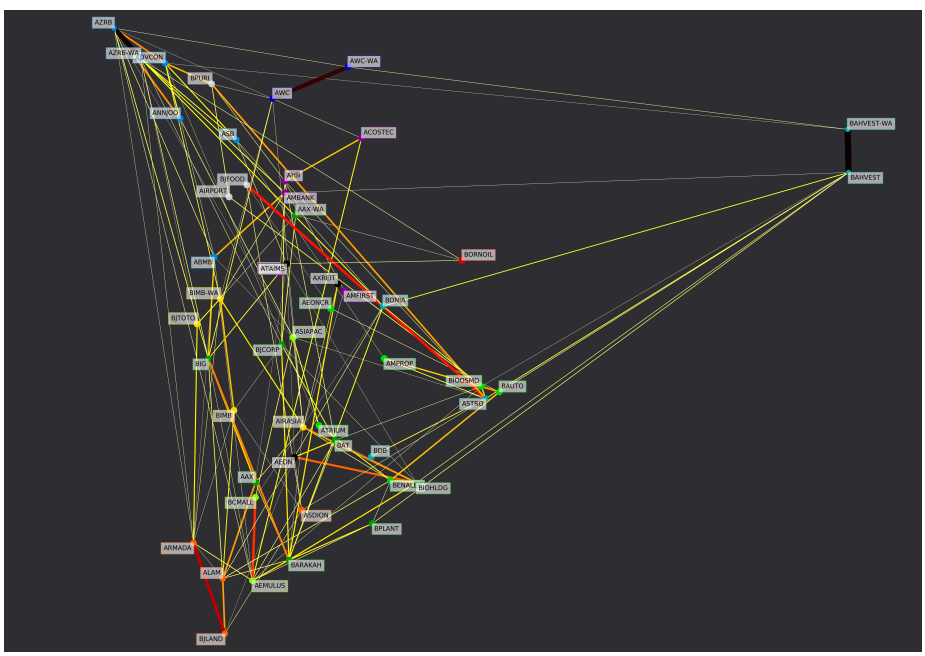

Figure 27: Market structure

5 Building an Investment Recommendation Program

A simple Python program was created to give the user recommendations regarding stock investment. The program recommends stocks whose price was high

then dropped recently then started to become high again in the last few days. The program recommends stocks that are Shariah compliant to give suitable recommendations for everybody. Along with the recommendation, the program outputs useful information about the company and plots the price of the company over time.

For example, when we run the program recently, the program recommended two stocks. The output is shown in Figures 28 and 29.

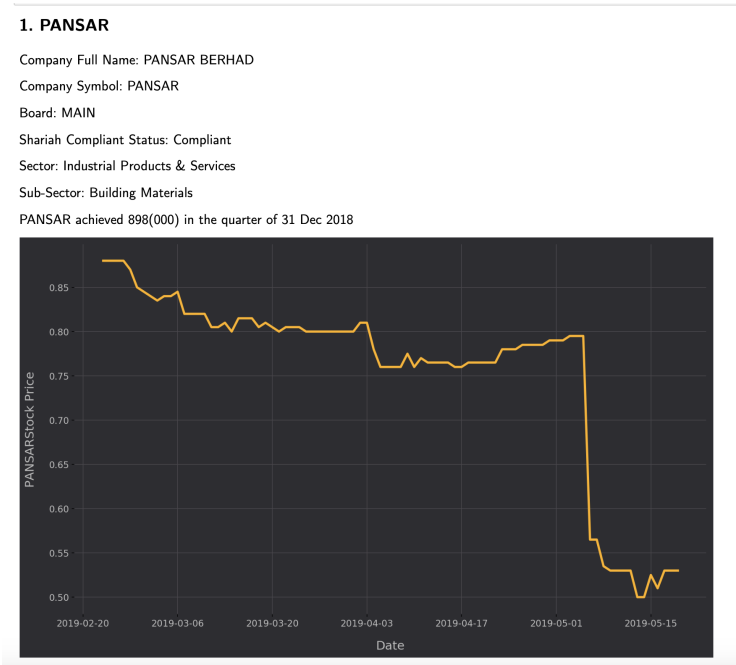

Figure 28: The output of the stock recommendation program

2. SEACERA

Company Full Name: SEACERA GROUP BERHAD

Company Symbol: SEACERA

Board: MAIN

Shariah Compliant Status: Compliant

Sector: Industrial Products & Services

Sub-Sector: Building Materials

SEACERA achieved -18,065(000) in the quarter of 31 Dec 2018

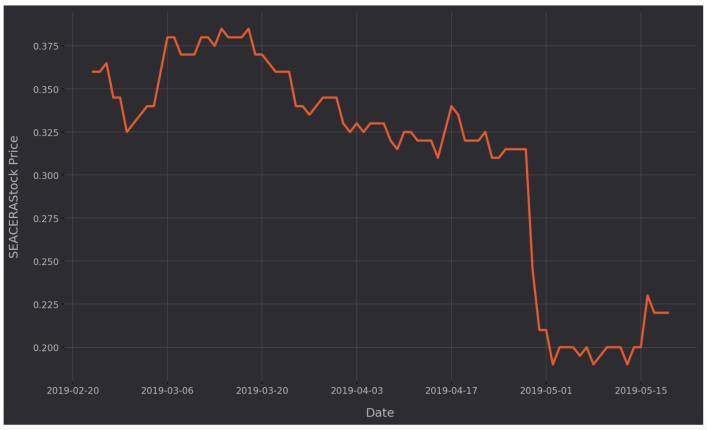

Figure 29: The output of the stock recommendation program

In the two recommendations, we see that the company price was high then dropped then started to rise again. Investing in these stocks might bring profits but that is not guaranteed.

Part IV
Conclusion

In this report, we showed the stages and milestones of our work in this stock-market project. We started with data crawling from multiple sources. Then, we stored and managed the crawled data into MySQL database. After that, we interpreted the data to see what is suitable for our analysis and what is not. Then, we explored and analyzed data using effective tools and visualizations. We performed covarianve and correlation analyses; we performed sentiment analysis on text data; we performed clustering as well. After that, we built a simple stock recommendation program to recommend stocks that might bring profits to the user.