**Project 2 ETL Report  
Stock Market influence of COVID 19**

**Authors**

* Mark Rotil
* Nikita Case
* Eric Lee
* Hadi Hanif
* Murat Cagri KOC

**Introduction**

Conventional thinking reveals that when calamities occur, the stock market is impacted negatively. These calamities stem from a variety of factors ranging from inaccurate economic modelling to climate change.

The Covid-19 pandemic, however, can be classified as a black swan event. News cycles inundate us with recovery rates, death rates, and lockdowns. The goal of this study is to determine just how sensitive the stock market is to these events.

We have identified 9 stocks across following industries:

* Hospitality
* Aviation
* Entertainment
* Technology
* Telecommunications

We will track and observe these stock prices from the 1st of January 2020 to the 4th of December 2020 and then compare it to the previous year.

Afterwards, we will identify key events related to the pandemic (e.g., the first case of Covid-19 in Canada) and look to see if there are any corresponding fluctuations in price.

We hypothesize that fluctuations of stock prices are strongly correlated with key events in Covid-19 such as lockdowns and that the correlation weakens as key events pass. This correlation applies to all industries within the stock market.

**Extraction Process**

Our first step is to extract the stock market data from Yahoo Finance in the form of CSVs.

Once that is done, we proceed to prepare the necessary initial import parameters into Jupyter Notebooks.

**Step 1**



Once completed, we will then import the CSV into Jupyter notebook and check to see if the data is accurate.

**Step 2**



Here we see that the data is accurate after being imported from the CSV.

**Step 2**

We then proceed to scrape the dates of key events from global news source.

****

**Transformation Process**

Now that the data has been loaded, we will begin the Transformation process. 

We drop the “CLOSE” column from the dataset because we will be using use the Adjusted\_Close on a day-to-day basis. The difference between the two is that the Adjusted\_Close factors in the price after dividend issuances.

We also converted the Date format because it will make the querying process much simpler. We also checked to determine how the data will be represented.

This process is replicated 9 times, once for each stock.

**Step 3**

After extracting the data from the news sources, we begin to clean up the news sources by filtering it into a dictionary.

****

**Step 4**

Next, we convert the dates into a data frame with ‘Date’ and ‘News’.



Step 5

After converting the dates into a data frame, we replace the dates format. This was done manually to avoid any errors.



**Step 6**

We then apply a filtration of key words, which includes these four words: *lockdown, shutdown, Toronto, deaths*.

This reduced the list of 129 news articles to 19.

**Loading Process**

After finishing cleansing, we proceed to load the data into ElephantSQL. ElephantSQL allows us to aggregate the data into single SQL table.

**Step 7**

We first create an engine that allows us to export our transformed files onto ElephantSQL.



It is also important to note that ElephantSQl limits the number of users that can concurrently function.

Next, we do a check to see if all the tables have been be placed correctly. To do that, we do a table name check:



Here we see that the tables are correctly showing up.

Step 8

Our first query is to find out if there have been any changes of 7% or more to the prices per each stock.



Step 9

We then determine the negative results which stocks have had only negative changes in 2020.



Another way of doing this is like so:



Step 9

We then calculate the stocks highest price point by quarter.



**Limitations:**

The first challenge we encountered is the user limitations for ElephantSQL. Our team consisted of five members; we were able to have two users fetch the information for ElephantSQL. Our attempts to use other applications for loading purposes, such as MongoDB, were unsuccessful also. We had to constantly keep track of which users were accessing ElephantSQL and terminate connections when the users were done.

Another challenge we faced was in our transformation process. We had multiple iterations of deletions and uploads of our tables. This is due in part to the fact that when we made modifications, we would have to apply them across 9 files, one for each stock. In another cycle we might have merged all the files before putting up on ElephantSQL or restricting the number of stocks.

**Following Steps:**

Following steps would include visualizing the tables on a line graph then plotting all the stocks across a single graph with time and stock prices as the variables. We would then apply dates of key events to determine whether decreases in prices correspond to the dates.

We would also spend more time designing queries that would encompass the entirety our dataset and not just our