Big Data - Assignment 1

Vlady Veselinov - CS982 - University of Strathclyde

1. Introduction

This report focuses on exploring whether it's better to invest in stocks long-term or short-term. A dataset is selected from the Kaggle repository: New York Stock Exchange. It provides 5 years worth of historical information for S&P 500 companies. It contains rows of stock opening, closing, high and low prices, as well as volume of trades for each day. The Standard and Poor's index is a stable and important indicator for the financial market, which is why it is the target for this project. Investment in the really short term is generally considered borderline gambling because of how difficult it is to predict the market. Long-term investment and more in-general: long-term thinking, has many advocates, some of which are very successful people like Jeff Bezos and Sam Altman. If the stock market as a whole increases in value over time, it would be interesting to see if the same tendency is prevalent in different time segmentations of the data. The project comes with a complementary GitHub repository.

2. Key Challenges

a. Lack of Context

The main problem with financial data is that there is only numbers and no other contextual data. The world economy is driven by events, changes in financial numbers only happen after the events, which can make it hard to draw conclusions based on pricing data alone.

b. Lack of Volume

This report will try to answer how long is long-term and what does it mean to invest in the short-term. It can be certainly said that long-term is longer than a traditional corporate quarter which is the most popular standard for setting milestones in modern business. If Amazon thinks in 5-7 year stretches and this length of time is considered normal for a business to invest resources, the current dataset doesn't contain enough data to lead to any definite conclusions. With that being the case, the analysis can still be applied and observed to explore any relationships between smaller periods of time.

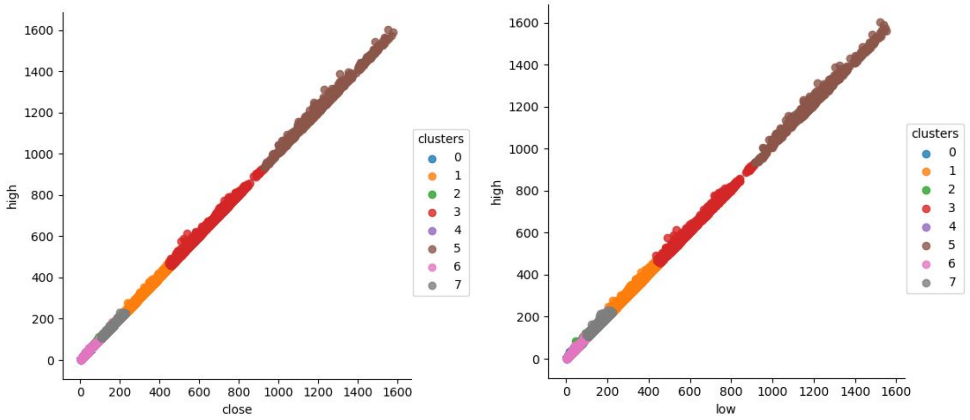
These problems will be addressed by using scalable methods which can be used on bigger datasets. The data will be segmented into time chunks with equal length and the profit for each chunk will be calculated. If long-term profitability can't be proven due to the size of the dataset, it still shouldn't be dismissed if short-term profitability is disproven.

3. Unsupervised Observations

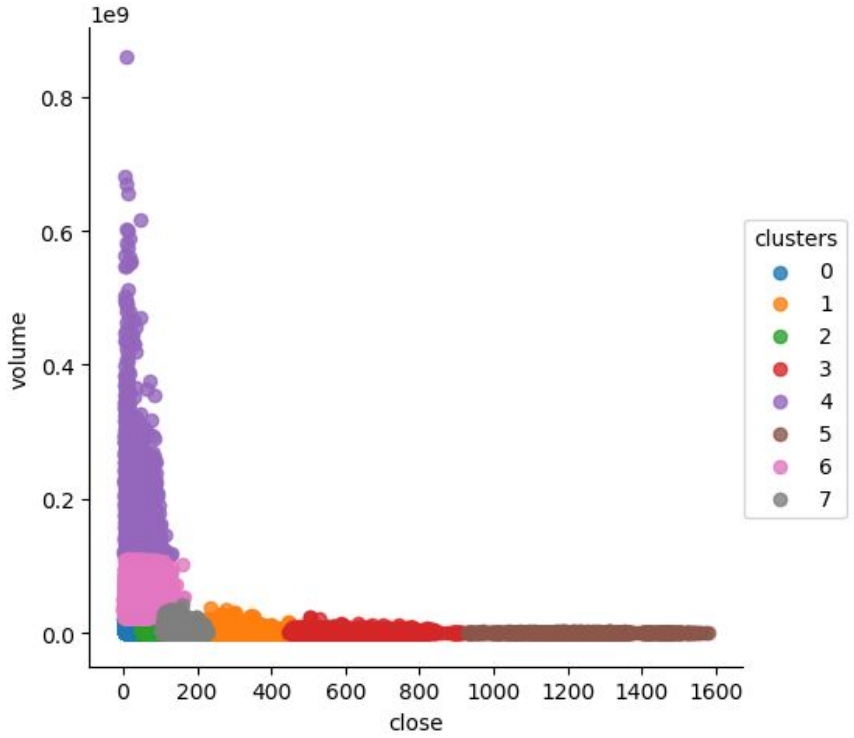
The K-Means method is used in order to gain a better understanding of the data. Using the open, close, low, high and volume values, the data is statistically standardised in order to make it mathematically easier for the learning model.

```
import pandas as pd
import seaborn
from matplotlib import pyplot as plt
from sklearn.cluster import KMeans
from scipy import stats

from data import data

# Replace company symbols with numbers
data = pd.get_dummies(data, columns = ['symbol'])

columns = ['open', 'close', 'low', 'high', 'volume']

# Standardise: i.e. make it mathematically convenient to compare stuff
dataStandardised = stats.zscore(data[columns])

model = KMeans().fit(dataStandardised)
labels = model.labels_

data['clusters'] = labels
columns.extend(['clusters'])
```

As demonstrated above, the data is fit to the K-Means model and is ready to be plotted.



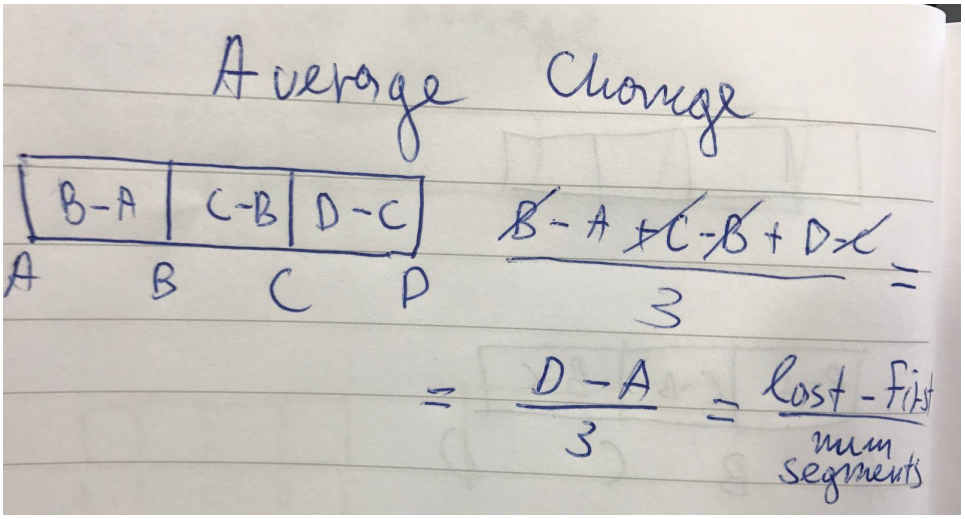
Looking at combinations of values, there are mostly linear relationships. Nothing jumps out in particular except a certain combination: volume and closing price:



There is clearly a relationship indicating a rapid rise in volume relates to a rapid decrease in closing price. This might sound like basic economics, but it turns out that large turnover means lower prices. That relationship can be used to indicate market turmoil and vice-versa, a steady financial state.

4. Segmenting Data and Calculating Profit Description,

In order to simulate stock trades being made, the data is segmented into equal time periods where a stock is bought at the beginning and sold at the end of each period. Essentially, this imitates a person buying stock, holding it for a set amount of time, selling it at the end and checking profitability. If the whole dataset is segmented in 3 segments, the average profit would be calculated like this:



This can be translated to the following function in Python, if the segmentation is done in multiples 30 days:

```
pricesDict = dict(data.groupby('symbol')['close'].apply(tuple))

def getAverageChange(symbol = '', step = 30, prices = pricesDict):
      stepped = prices[symbol][::step]
      length = len(stepped)

      if length > 1:
            return (stepped[-1] - stepped[0]) / (length - 1)

      return stepped[-1] - stepped[0]
```

After the average profit is calculated for each company and for each segmentation, it is statistically analysed in order to judge whether long-profitability is viable as a concept for this particular dataset. The average profit is calculated and put into a dictionary:

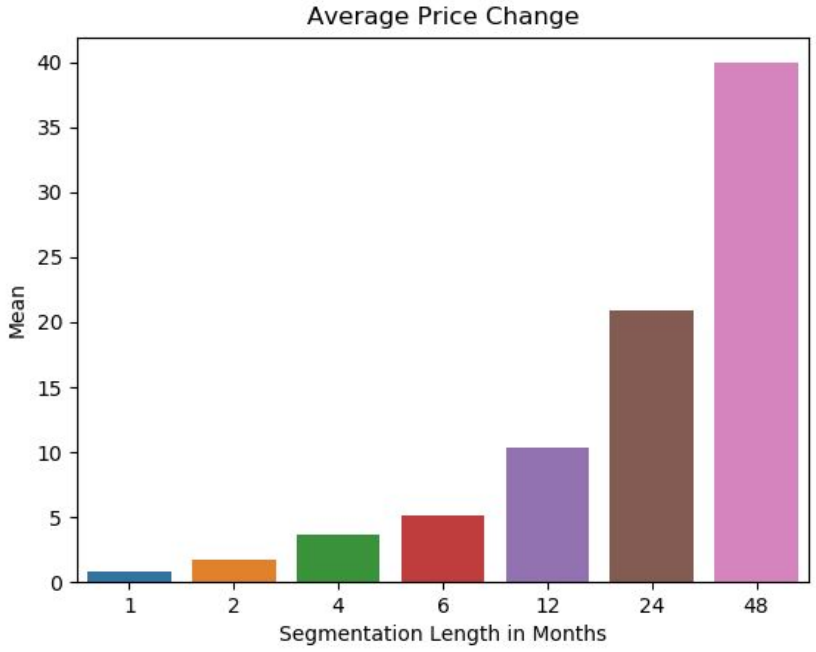
```
# These are the different segmentations in months
monthMultipliers = [1, 2, 4, 6, 12, 24, 48]

# Get steps in days
steps = [number * 30 for number in monthMultipliers]

steppedPrice = {}

averageChanges = {}
for multiplier in monthMultipliers:
    averageChanges[multiplier] = []


for key in pricesDict:
    steppedPrice[key] = {
        'prices': {},
    }

    for index, step in enumerate(steps):
        # Easy to step through the data since no days are missing
        steppedPrice[key]['prices'][monthMultipliers[index]] =
pricesDict[key][::step]


        averageChanges[monthMultipliers[index]].append(
            getAverageChange(key, step)
        )
```

The data is now ready to be statistically analysed.

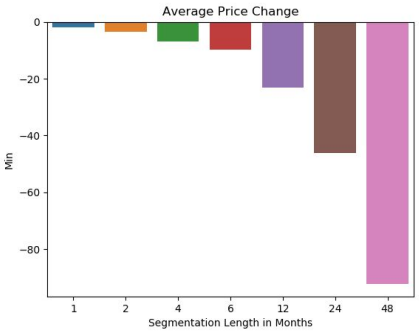
Using SciPy, a collection of descriptive statistical parameters is obtained for each segmentation:

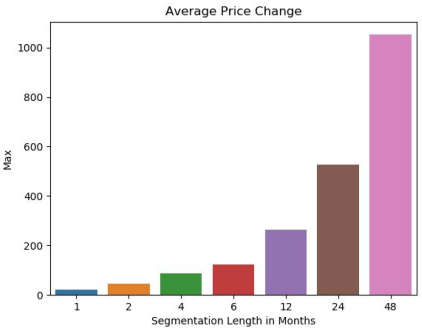
```
statsDict = {}
for multiplier in monthMultipliers:
    # [n, (min, max), mean, var, skew, kurt]
    nonFlatStats = tuple(scipy.stats.describe(averageChanges[multiplier]))

    # Flatten and pop to [min, max, mean, var, skew, kurt]
    statsDict[multiplier] = flatten(nonFlatStats[1:])
```

Looking at the mean values, there is a very prominent increase in average profit as the segmentation gets bigger and there is barely any profit for the smallest segmentations.

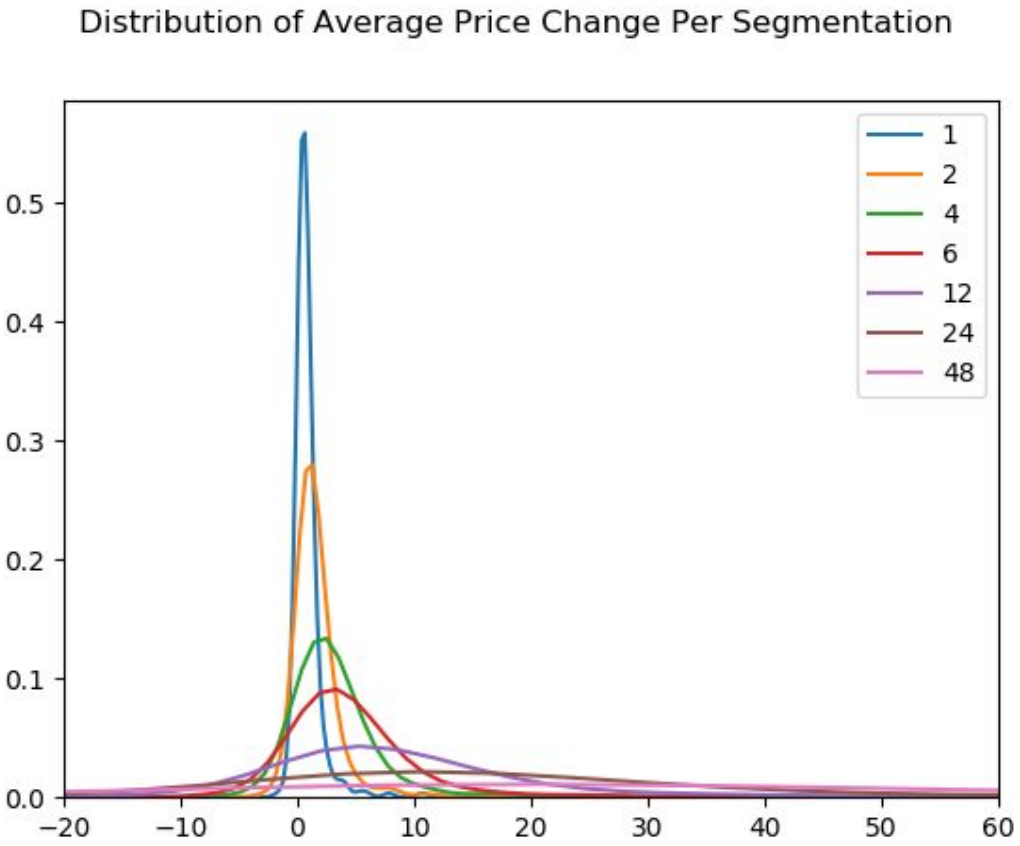


The minimum and maximum values follow the same tendency as the means. Increasing in magnitude with segmentation.

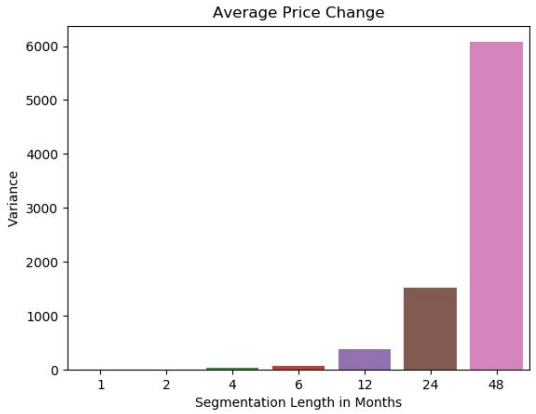




What will give a clearer clue of profitability is the distribution plot:



For each segmentation, the distribution starts shifting towards the positive side, which is good, but the steep decline in frequency means that with each segmentation there is less and less data. Which is true to the dataset, there are more 1 month periods in 5 years than there are 48 month periods, however this is a problem because in order to prove the positive trend, there needs to be more data for longer periods of time. The variance plot further supports this claim:



The high variance in the 48 month segmentation hints that the results are spread out from the mean, which is conflicting with the objective of the project, since the mean is trending upward. It's safe to say that it's not proven that investing very long-term is more profitable. However, looking at the distribution plot, there is a higher chance for profit when not investing very short-term.

5. Reflection on Methods

The most critical part of this assignment is asking the right questions in the beginning, before any code is written. This dataset is very specific and to make any meaningful high-order conclusions there needs to be more context, possibly another dataset that is indirectly connected. This is why the end goal to prove whether it's more profitable to invest in the long term remains incomplete, but the analytical methods applied can shed light on different relationships between values, such as the one with volume and closing price. It would have been more interesting to tie in another dataset and run some unsupervised learning methods. All of the code for this project can be found on the complementary GitHub repo.

References

Erik Marsja. (2017). *Descriptive Statistics using Python*. [online] Available at: https://www.marsja.se/pandas-python-descriptive-statistics/ [Accessed 3 Nov. 2017].

Scikit-learn.org. (2017). *K-means Clustering — scikit-learn 0.19.1 documentation*. [online] Available at: http://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_iris.html [Accessed 6 Nov. 2017].

YouTube. (2017). *Sam Altman : How to Build the Future*. [online] Available at: https://www.youtube.com/watch?v=sYMqVwsewSg [Accessed 1 Nov. 2017].