K-Means Clustering of Daily Stock Data

Jessica Kwok, Melody Zhao, Kewei Zhou
Harvey Mudd College

Abstract

The purpose of this paper is to sort general trends of daily stock data into clusters, then use the output to predict future trends and make connections with significant events happening during a specific period of time. K-Means Clustering is used to cluster the general trends, and Markov Chain is used to predict future trends. The methodology used is able to sort the trends into 7 general clusters. The accuracy of the Markov Chain prediction model is around 92%. Although the accuracy is relatively high, the quality of the clustering needs to be further investigated.

1 Introduction

It is commonly thought that stock indexes are only measures of stock market performance, or the economy in a country. However, they are also indicators of a country's non-economic conditions. For example, a research done by the University of Utara in 2017, by studying the stock prices in Pakistani for a period of time, has concluded that the stock market returns and volatility have noticeable relationship with a country's political instability (Irshad). History often repeats itself. Therefore, it becomes essential to connect unusual yet significant time periods, such as the 911 terrorist attack in 2001 and the outbreak of the contagious illness SARS in 2003, to the change in stock market performances. If similar incidents happen again, the results of this investigation can not only allow government to reflect and update on the effectiveness of their policies and measures during states of emergencies, but also to allow public and private institutions to be more prepared of what to expect.

However, there are so many events that have happened throughout just the past 20 years that it would be impossible to connect each significant event to the stock performance during that period manually. Furthermore, how would one know which events are noteworthy enough such that it had an impact on the stock market? It would also be useful to know which events had similar reactions from the stock market and for the same event, how did countries react differently. For example, is the stock performance during the outbreak of SARS in 2003 and that of COVID19 in 2020 similar? Did the US and China react similarly to either outbreaks? Finally, based on the previous stock performances, is it possible to predict future trends? This paper aims to investigate and produce both quantitative and qualitative answers to these questions.

In order to identify unusual events in the stock market, this paper approaches the solution using K-Means clustering, a popular unsupervised machine learning algorithm. It sorts stock performance over short period of times into clusters. In other words, each short period of time is labeled with a number that describes the general shape of its stock performance. Finally, by applying a Markov chain on the time series sequence of labels, the model predicts the fifth label based on the previous four.

2 Data Collection

The main source of data collection is Yahoo Finance and Barchart. For the midterm project, daily historical data from 2000 to current for S&P 500 is used. Historical data for Heng Sung Index (Hong Kong), Nikkei 225, SSE Composite Index (Shanghai) and SZSE Component Index (ShenZhen) are also collected for future usage. These stock indexes are selected because they are some of the most commonly used indexes that are able to represent the general stock market in each country. Historical data is downloaded in the form of csv file and contains the following information: date, open price, high, low, closing price, adjusted close, and volume.

3 Methodology

3.1 Data Pre-Processing

3.1.1 Standardizing Data

As mentioned in the data collection section, the following information for a stock is obtained from financial website from 2000 to current: date, open price, high, low, closing price, adjusted close, and volume. In the attempt to build a model that can identify trends in historical data and apply to similar future occurrences, stock prices and volatility are the most important characteristics. Since it is essential to see how one event may affect the stock prices and volatility, the percent change of price and volume are used to train the model. Since these two components are not included directly in the raw data, percent change for price and volume are calculated. The equations to calculate the percent change in volume and price are defined as the following:

$$\text{Percent Change in Price (T)} = \frac{\text{Price(T)- Price(T-1)}}{\text{Price(T-1)}} * 100$$
$$\text{Percent Change in Volume (T)} = \frac{\text{Volume(T)- Volume(T-1)}}{\text{Volume(T-1)}} * 100$$

where T is the date of interest. This is done directly in the csv file through Microsoft Excel. The calculated values are stored in new columns labeled as "PercentPrice" and "PercentVolume" respectively.

3.1.2 Matching Standard Date

The second step in pre-processing involves standardizing the range of dates that would be used in building the machine learning model. Since one of the ultimate goals in this project is to compare stock performance reactions in different countries, it is important to note that due to a variety of reasons, such as countries having different holidays, the days that each stock exchanges open differ. In order to resolve this issue, a system of standard date is enforced to ensure that data across different stock indexes can be easily compared.

First, a separate csv file containing all standard dates is created. This is done purely on Microsoft Excel, and the column of standard date contains all weekdays from 2000 to current. Then, the csv file containing the standard dates, as well as the csv file containing the calculated percent change in price and volume, are passed into the "preprocessed.py" as input. In the python script, the standard list is looped through, and if the historical data csv file contained information for the date of interest, the percent change in price and volume are assigned to that date. However, if the historical data did not contain the date of interest, the associated percent changes in volume and price of that date are defined to be 0. This is based on the assumption that the economic condition would not alter greatly in a day without significant events, then the change in price and volume should be similar for consecutive days. The output of the python file is the input for the machine learning model, in which the standard dates with their associated percentage changes in price and volume values are written to a new csv file named "newProcessed.csv".

3.2 Principal Component Analysis (PCA)

After pre-processing the stock market data, the percentage change of price and volume every day of a specific market are obtained. We divide the data into smaller chunks of 30 days each so that we can look at a market at a specific period of time. By the shift window method, we result in a set of 30-day data starting from any day. The shift window method enables us to study the subtle changes in the stock market over time.

We then use principal components analysis to categorize each chunk of 30-day data. Principal components analysis(PCA) is a statistical procedure that converts a set of data pertaining to possibly correlated variables into a set of linearly uncorrelated variables. These uncorrelated variables are called principal components. For a two-dimensional dataset, we result in a main principal axis, which represents the direction with the largest possible variance, and the second principal axis, which represents the direction with the largest variance under the constraint that it is orthogonal to the main principal axis. The lengths of the principal components are the amount of variance explained by each axis.

We calculated the principal axes of our data and their respective variance using the library sklearn.decomposition.PCA. We input a list of paired percentage change in price and percentage change in volume. Each pair corresponds to the stock market data of a specific stock on a specific day. Using the explained_variance_ function, we were able to extract two variances represented by the principal components in decreasing order. We then found the vector representation of the principal components using the components_ function. Since the two principal components are guaranteed to be orthogonal, we could simplify the representation of the principal components with the angle between the main principal component and the horizontal. Thus, we calculated the angle using

$$\tan^{-1} \frac{\text{y component of the main principal axis}}{\text{x component of the main principal axis}} \tag{1}$$

Along with the mean of percentage change in price and the mean of percentage change in volume, we characterized each chunk of data using a five-dimensional vector: mean of change in price, mean of change in volume, main variance, second variance, and angle of the main principal component.

3.3 Visualize Data

In order to visualize the processed PCA data, a plot with percent change in price as its x-axis and percent change in volume as its y-axis is generated. The plot contains:

1. The 30 data points
2. The 2 eigenvectors (Principal and second Principal)
3. The ellipse corresponding to the eigenvectors, with the eigenvalues as its width and height, and the angle between the principal and horizontal axis as its degree of rotation

An example of the plot looks like the following (Figure 1):

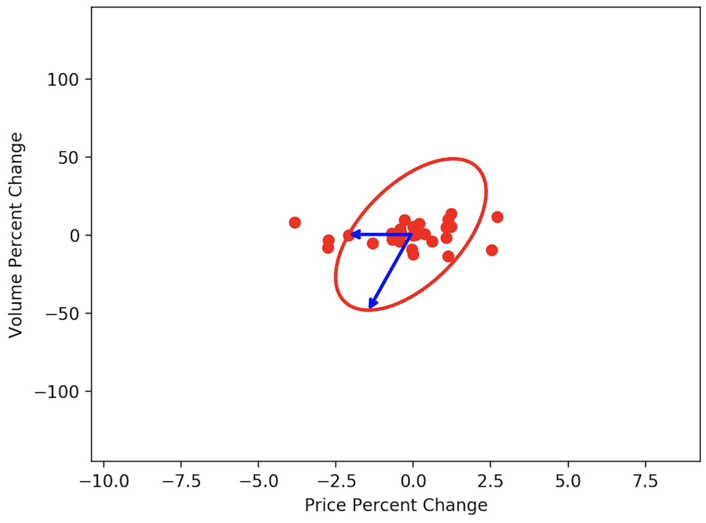


Figure 1

It can be seen that the PCA is performed correctly on the data points since the eclipse is a relatively good representation of the data. In addition, notice that the scale in the y-axis is substantially greater than that of the x-axis. This implies that one of the eigenvalues will also be noticeably bigger than the other.

3.4 K-Means Clustering

K-Means clustering is an unsupervised machine learning technique that uses input vectors in order to label the dataset. K-Means defines k number of centroids and cluster data points according to the distance function. The data point is labeled such that the distance between the point and the centroid of a cluster is minimized. The code for K-Means is adapted from an online source code from Visual Studio Magazine. [1]

3.4.1 Normalize Data

K-Means Clustering is called on an array of array in the following format:

$$\begin{bmatrix} p_1 & v_1 & \lambda_{prin,1} & \lambda_{minor,1} & \theta_1 \\ p_2 & v_2 & \lambda_{prin,2} & \lambda_{minor,2} & \theta_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ p_n & v_n & \lambda_{prin,n} & \lambda_{minor,n} & \theta_n \end{bmatrix}$$

where each row is an output of the PCA and thus a vector that represents the stock performance trend in a period of 30 days. Note that p is the average percent change in price, v is the average percent change in volume, λ_{prin} being the principal eigenvalue, λ_{minor} being the minor eigenvalue, and theta being the angle between the horizontal axis and the principal eigenvector, in degrees. Before running the clustering algorithm, all of the raw data are normalized using the Min-Max normalization method, which utilizes the following technique: $x' = \frac{x-min(x)}{max(x)-min(x)}$. Doing so would scale the range of the data to between 0 and 1. The normalized array of array would then be used as input for clustering.

3.4.2 Distance Function

In order to determine which cluster a data point belongs to, a function is defined to calculate the data point's distance from the mean of each cluster:

$$D = W_{center} \cdot \Delta center + W_{\lambda_1} \cdot \Delta\lambda_1 + W_{\lambda_2} \cdot \Delta\lambda_2 + W_\theta \cdot \Delta\theta$$

where the difference between each variable is

$$\begin{aligned} \Delta center &= \text{norm}(p_i - \bar{p}, v_i - \bar{v}) \\ \Delta\lambda_1 &= \text{abs}(\lambda_{1,i} - \bar{\lambda_1}) \\ \Delta\lambda_2 &= \text{abs}(\lambda_{2,i} - \bar{\lambda_2}) \\ \Delta\theta &= \text{abs}(\theta_i - \bar{\theta}) \end{aligned}$$

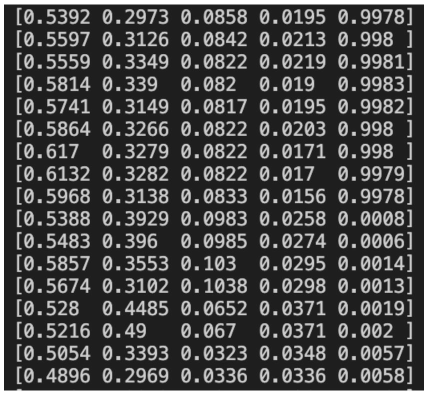
and $\bar{p}, \bar{v}, \bar{\lambda_1}, \bar{\lambda_2}, \bar{\theta}$ are mean of each variable in the specific cluster. In addition, the constant weight for the variables are $W_{center}, W_{\lambda_1}, W_{\lambda_2}, W_\theta$ respectively, such that

$$W_{center} + W_{\lambda_1} + W_{\lambda_2} + W_\theta = 1$$

Before using the distance function, one important question remains - what are the weight for each variable such that the result of the clustering makes the most sense? Insights to this question can be extracted from looking at the normalized data mentioned in 3.4.1. A snippet of the data is shown below (Figure 2), where the columns are center of price, center of volume, λ of principal axis, λ of minor axis, and the angle respectively. Although each of these values are reanged from 0 to 1, it is obvious that, after normalizing it, the percent change in volume (third column) becomes extremely small comparing to the others. The explanation for this is briefly mentioned in the PCA visualization section (3.3) - it is due to the fact that the range of percent change in volume is considerably bigger than all the other variables. If not dealt with properly, this would result in λ of the minor axis and especially the angle completely over-powering it. After several trials and erros, it can be concluded

[1] Garbade, Michael J. "Understanding K-Means Clustering in Machine Learning." Medium, Towards Data-Science

that,in order to compensate for the issue, the weight for the principal axis is set to 0.78, while minor_W and angle_W are set to 0.015 and 0.005 respectively. Since the centroid should play a relatively big role in cluster, its weight is set to 0.2.



```
[0.5392 0.2973 0.0858 0.0195 0.9978]
[0.5597 0.3126 0.0842 0.0213 0.998 ]
[0.5559 0.3349 0.0822 0.0219 0.9981]
[0.5814 0.339  0.082  0.019  0.9983]
[0.5741 0.3149 0.0817 0.0195 0.9982]
[0.5864 0.3266 0.0822 0.0203 0.998 ]
[0.617  0.3279 0.0822 0.0171 0.998 ]
[0.6132 0.3282 0.0822 0.017  0.9979]
[0.5968 0.3138 0.0833 0.0156 0.9978]
[0.5388 0.3929 0.0983 0.0258 0.0008]
[0.5483 0.396  0.0985 0.0274 0.0006]
[0.5857 0.3553 0.103  0.0295 0.0014]
[0.5674 0.3102 0.1038 0.0298 0.0013]
[0.528  0.4485 0.0652 0.0371 0.0019]
[0.5216 0.49   0.067  0.0371 0.002 ]
[0.5054 0.3393 0.0323 0.0348 0.0057]
[0.4896 0.2969 0.0336 0.0336 0.0058]
```

Figure 2

3.4.3 Determine K

K values represent the number of clusters the ellipses are divided into. There is no existing theoretical k value, so we determined k value by numerically finding a balance between a small average distance from each ellipse to their corresponding cluster center and a reasonably small k value.

We defined a cluster_distance function that finds the average distance from the ellipses to their corresponding cluster center associated with a specific value. It takes in a list of clusters, where each cluster is a list of 5-dimensional vectors representing the ellipses in the cluster. The function finds the mean of ellipses' centers in each cluster as the cluster center and sums the distance from each ellipse to the cluster center. Cluster_distance then sums the average distance in each cluster and divide it by the number of clusters. The output gives us a measure of how close ellipses in the same cluster are for a specific k value, weighted by the number of clusters. Then, we clustered the ellipses using k values ranging from 1 to 7. We found the cluster distance of each k value and plotted the cluster distance against the k value.

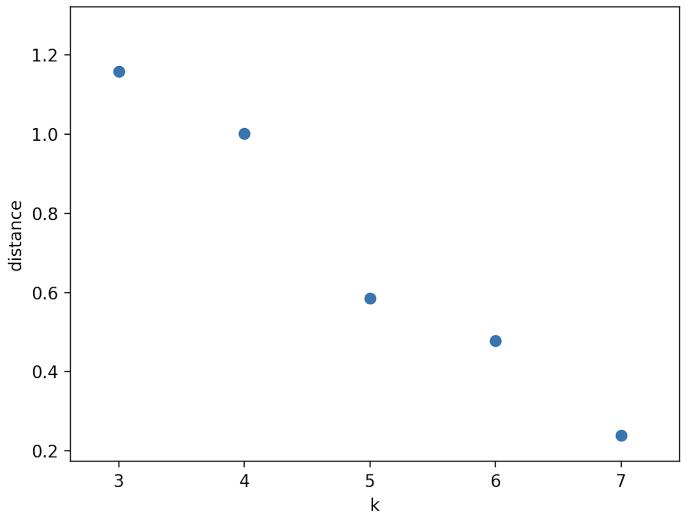


Figure 3

It can be seen that from Figure 3 that the average distance within a cluster is the lowest when k=7. Thus, it can be concluded that k=7 is the optimal k to cluster the data in.

3.5 Tying Together

Tying all the strategies discussed from 3.4.1 to 3.4.3, K-means clustering is performed. First, the normalization method is applied so that the raw data imported from the csv file is processed and be fit into the range from 0 to 1. Then, using the strategy described in 3.4.3, the optimal k value determined is 7. The clustering algorithm first randomly labels all of the data into 7 clusters and calculate an initialized mean. Then the cluster will be updated. The updating cluster technique uses the initialized mean and the distance function to calculate all the data points' distances relatives to the means of each cluster. Then, each data point is re-clustered based on the distance. After re-clustering, a new mean is generated for each cluster, and the process repeats with re-clustering following by updating the means. Clustering is done and exits when there is no update in clusters, meaning that no data points are being relabeled.

3.6 Markov Model of Time Series

Having grouped the ellipses into 7 clusters, we resulted in a time series of cluster labels. Since clustering a 30-day period stock market data categorizes this period by its volatility, the time series represents the trend of stock market volatility. We believe that the volatility of the market at a given period is dependent on the previous periods, so this time series is likely to repeat itself.

We used the Markov model to predict what clustering the future stock market data would belong to. Our training set is a time series of clusters, and our model consists of a dictionary, of which the keys are chunks of 4 consecutive cluster labels and the values are an array such that the first element is the most probable cluster that follows this 4-cluster prefix and the latter is the probability of the most probable cluster happening. To construct this dictionary, we looked that the time series data in chunks of 5 clusters using the shift window method. Then, we recorded the type of the fifth cluster under the label of the 4-cluster prefix. We chose to investigate the time series data in chunks of five clusters for it resulted in an optimized balance between a high coverage of possible prefixes and a high probability of the most probable prediction.

We then tested our Markov model on a testing set, which consists of stock market data later in time than the training set. Similarly, we divided the time series of clusters into chunks of 5 clusters using the shift window. Comparing the fifth cluster predicted by our model and the actual fifth cluster, we found the number of correct predictions and wrong predictions, as well as the number of times the 4-cluster prefix was not found in the model. With this result, we were able to calculate the accuracy of our model.

4 Results

4.1 K-Means Clustering

After determining the optimal K value (reference 3.4.3) and performing K-Means clustering, the data is sorted into 7 clusters. In order to verify the correctness of this process, every ellipse from the data is plotted on the same graph, where each color represents a different cluster (see Figure 4).

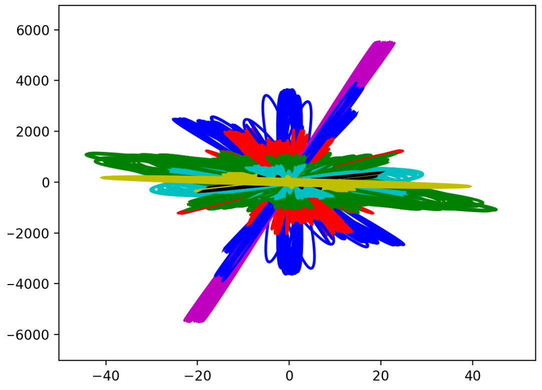


Figure 4

A pattern can be somewhat seen. For example, all the ellipses that are extremely long and in the same direction are colored purple. However, since the scale of the x-axis is considerably smaller than that of the y-axis (around 200 times smaller), this graph is quite misleading and hard to interpret. Thus, an attempt was made to re-scale the plot such that the x and y-axes have the same scale (see Figure 5).

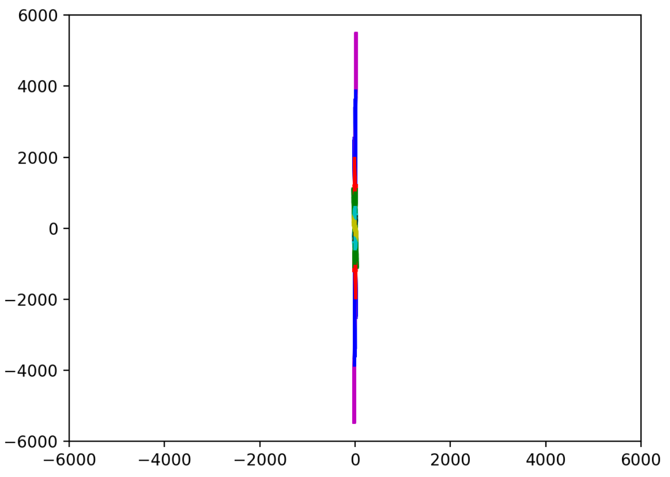


Figure 5

In the scaled graph, there are many things to note of. Similar to what was mentioned in the parameter choosing section, the mean of all the individual eclipses seem to very similar and the relative angles from the horizontal axis to the principal eigenvector are all close to 90 degrees. The biggest difference between eclipses is the length of the principal eigenvector, and this supports our choice of weights as the parameters.

Zooming into figure 5, from the center of the figure and moving outward, seven colors can easily identified in both directions. This may suggest that similar eclipses were successfully classified into their respective clusters based on the distance function.

The K-means algorithm seems to be doing its job, but the two figures seem to suggest some problems that must be looked at in the future. First, our data has drastically different variances, which makes choosing the parameters difficult and the distance function unbalance. The variability in the percentage of price change is relatively small in comparison to the variability in the percentage of volume change, which makes the principal eigenvalues to be almost hundreds of times of the the minor eigenvalues. The differences in magnitude of the two eigenvalues cause the theta value to all be close to 90 degrees, which make the theta value less meaningful and less useful in model training. Another strategy must be come up with in order to find the balance between the variability of the two eigenvalues in order to make the distance function more balanced and demonstrative.

4.2 Markov Model Result

One of the ways that the K-means model can be used is to build a Markov model. With the trained K-means model, all of the eclipses from the data are sorted into 7 clusters. The Markov model is built in order to test whether using four shifting windows' worth of data, the fifth day's general eclipse shape can be predicted. The dataset is split into training set and test set with a 9:1 ratio. When the model is tested on the test set, the accuracy achieved is 92.01% with 1 case in the test set not being able to use the trained dictionary to make a prediction. This accuracy is very high and may suggest that there is a trend in the dataset. It is worthy to note that four days of shift windows is not significant in tying the sequence to a specific type of event, so more testing needs to be done. With this strategy, the number of shift windows needs be increased in the future in order to identify the longest sequence of trends. Then, the confidence of correlating the trend to a specific event would be higher. On the other hand, it is also likely that many eclipses are being clustered into one cluster, and achieving a high accuracy is due to luck. More extensive testing must be done in order to solidify the conclusion.

5 Literature Review

We reviewed literature related to analyzing stock market data with similar approaches. We didn't find any existing articles that categorizes stock market data using PCA, but we found several articles that share other aspects with our approach.

One of the articles uses multiple regression analysis of stock trend prediction, including Data preprocessing was performed before regression analysis. They conducted dimensionality reductions by computing percentage price change, stock volume and dispersion/volatility. These variables are presented in the regression models with optimized regression coefficients from R function Im(). Using the regression model and the KSE 100 index as an input dataset, they obtained a prediction accuracy of 95%. Similarly, when Lucky Cement Stock dataset was used, the model resulted in an accuracy of 89%. This project uses similar data preprocessing method for dimensionality reductions, which serves as indicators for the stock market trend. However, they used a regression model of the market indicators. We could possible use similar indicators in our model, and use similar regression models to train the parameters in our distance function.

The second article uses a hybrid feature extraction method, coupled with Support Vector Machine(SVM), in order to predict stock trends. Algorithms are applied before SVM clustering in order to extract features from the dataset. The features are scored based on F-score, and the best feature models are used as training data for SVM. When applying to different stock indexes and exchange rate, SVM achieved accuracy ranging from 83.2% to 87.2% accuracy in predicting NASDAQ stock trend with an average accuracy of 85.4%. The project the article described had a similar goal to our project and used classifying technique. However, instead of using unsupervised technique of clustering like k-means, SVM, a supervised technique in model training, is used. Then, the feature selection step is vital and labels the data before classifying. Their model and our model achieved similar accuracy in predicting stock trend. We could also apply similar strategies as them such as using feature extractions models and supervised learning classification models.

6 Future Work

6.1 More Data Collection

The current model is only trained on daily data, which is only around 5000 data points. Moving forward, we will continue to acquire intraday data (minutes data, hours data) and make appropriate purchases if necessary. Getting intraday data would increase our data size to at least half a million. Training on more data would result in a better model and provide more opportunities for trend seeking. For instance, the key to the dictionary in the Markov Model can be increased to 30 instead of 5. This would significantly increase the possible conclusions that can be made based on the results.

6.2 Better Method for Pre-processing

K-Means clustering usually requires variables to have similar standard deviation. However, in this case, the percent change in price is significantly less variable than the percent change in volume. Volume can easily change for around 10% while changing 10% in price is highly unlikely without significant events. We will seek algorithms that will alter the raw date such that each variable would have similar variability. In addition, instead of percent change in price versus volume, we can also attempt to apply our model on other possibly related variables that might tell us useful information.

6.3 Revising Markov Model

The current trained model predicts the clustering trend of S&P 500 using the Markov Model and the four clusters preceding it. However, four clusters is a rather small number. It is possible that a reoccurring pattern of 5-cluster chunks happened by chance and doesn't have significant implications. Moreover, with an accuracy of 92.01%, we suspect that our clustering method resulted in uneven clusters. If that is the case, there is an even higher probability correctly predicting the trend only by chance.

These circumstances can be reduced if we use longer prefixes in our Markov model, that is looking for more than 4 preceding clusters when making a prediction. Then, in order to maintain the high coverage of possible prefixes in our model, we would need more historical data.

In addition, there might be unobserved ("hidden") states present in our model, we can apply the Hidden Markov model (HMM) for prediction instead.

6.4 Revise Distance Function Parameters

Currently, the parameters for the distance function are estimated through trial and error manually. Since it is a very important part of ultimately the clustering result, manually estimation is not sufficient. In the future, we will try different methods of optimizing these parameters, such as using neural network, or least square method. Then, we can compare the results of our clustering, and see if it is more reasonable. Furthermore, the optimization of parameters could take into account other quantitative data during the respective time period, like the GDP and unemployment rate of a country.

6.5 Explore Additional Variables

Our current model uses day-to-day prices and volume data as the main variables. Other variables that provide a lot of information about the financial market include the daily high and low data. We could utilize those variables as well in order to optimize the k-means model. We could also train k-means model using a subset of the variables and compare the results to our current model.

References

[1] Asghar, Muhammad Zubair, et al. “Development of Stock Market Trend Prediction System Using Multiple Regression.” Computational and Mathematical Organization Theory, vol. 25, no. 3, 2019, pp. 271–301.
[2] Garbade, Michael J. “Understanding K-Means Clustering in Machine Learning.” Medium, Towards Data Science, 12 Sept. 2018, towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1.
[3] Irshad, Hira. “Relationship Among Political Instability, Stock Market Returns and Stock Market Volatility.” Studies in Business and Economics, vol. 12, no. 2, 2017, pp. 70–99., doi:10.1515/sbe-2017-0023.
[4] Lee, Ming-Chi. “Using Support Vector Machine with a Hybrid Feature Selection Method to the Stock Trend Prediction.” Expert Systems with Applications, vol. 36, no. 8, 2009, pp. 10896–10904., doi:10.1016/j.eswa.2009.02.038.
[5] McCaffrey03/27/2018, James. “Data Clustering with K-Means Using Python.” Visual Studio Magazine, visualstudiomagazine.com/articles/2018/03/27/clustering-with-k-means-using-python.aspx.
[6] “Sklearn.decomposition.PCA.” Scikit, scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html.