

Investing101

Andre Lee
Singapore Management University
andre.lee.2019@mitb.smu.edu.sg

Boey Yi Heen
Singapore Management University
yiheen.boey.2019@mitb.smu.edu.sg

Ng Wee Kien
Singapore Management University
weekien.ng.2020@mitb.smu.edu.sg

ABSTRACT

In making investing decisions, it is always helpful to have seamless tools at investors' disposal to generate insights out of the deluge of financial data available out there. In this project, we developed an R Shiny application using a range of data visualisation techniques to provide some useful tools to investors to aid them in their investing decisions. These tools are not available in traditional sources which retail investors tend to use. The embedded models, visualisation tools and application design of the application are discussed.

1. INTRODUCTION

Most retail investors rely on traditional financial data websites such as Yahoo Finance to retrieve relevant financial data to aid in their investing decisions. However, these sites' offerings tend to be homogeneous and comprise usual information such as price quotes, market commentary and corporate actions. Investors are left on their own to make sense of these information, without tools to further perform actions such as price forecasting or segmenting stocks into groups.

Having identified this gap, we developed Investing101, an interactive visual analytics dashboard to allow investors to not only retrieve basic price information of stocks, but also apply models such as ARIMA forecasting and hierarchical clustering. The application will also allow users to construct bespoke portfolios and see their historical performances. On top of this, the dashboard also provides users with a tool that readily aggregates and interprets candlestick patterns for the user. All these different tools come together to deliver a product that adds value that far exceeds what traditional finance data websites can provide.

2. MOTIVATION & OBJECTIVES

Our research was motivated by the lack of effective and easy to use web-enabled analytics tools for investment planning. We aim to provide investors with an analytical tool for vi-

sually understanding technical indicators of stocks - Specifically, it attempts to support the following analysis requirements:

1. To be able to conduct technical, portfolio analysis and correlation analysis amongst multiple stocks;
2. To create a clustering visualization of related stocks so users can diversify their holdings;
3. To provide a forecast of future stock prices for users to identify suitable entry/exit points.

3. LITERATURE REVIEW

3.1 Technical Indicators

In the literature review conducted, it was found that there have been various attempts by users to provide tools for analysing stock price action. Existing charting or dashboarding tools often focus on only simple charting or technical indicator functionalities. Most open source/ free tools only offer the various functionalities in isolation, and users are unable to have a holistic view uniting technical analysis, portfolio analysis, and the other components to allow for complete decision making when making buying and selling decisions.

In this sub-module, the team will put together various sub-modules spanning across: Charting, Technical Analysis, Correlation Analysis, Clustering, Forecasting, Portfolio Analysis.

3.2 Forecasting

Time series forecasting is not the same as technical analysis and can be seen as a natural extension/logical next step after conducting technical analysis. The main difference is that time series forecasting gives you an exact forecasted price, while technical analysis only predicts the future movement (up/down) of the price (Berdiell, 2015) [1]. The fundamental idea of this method is to seek out patterns in the historical stock prices with a hybrid approach. A hybrid approach combine multiple different models to forecast stock prices. For example, the papers of Markowska-Kaczmar and Dziedzic (2008) [5] and Wang et al (2015) [4] both proposed tackling stock price forecasting with an amalgamation of multiple models instead of just relying on one form of forecasting. This has been shown to result in superior forecasting accuracy and performance as compared to using a dedicated forecasting method. The researches of

Dey et al. (2016) [3] has also shown similar results albeit with some overfitting issues and a limited testing scenario.

The modification made in building the sub-module to the existing hybrid techniques is that the application will do a forecast based on 5 different models then present the mean of the 3 best performing ones.

3.3 Clustering

Da Silva, Cunha, and da Costa (2005) [2] carried out hierarchical clustering on 816 stocks listed in North and South America from 1997-1999 on the following variables: return, risk, earnings-price ratio, book value-price ratio, sales-price ratio, sales-number of stocks ratio and dividend yield. Their research showed that clusters with the best risk-return profile from the time window where clustering was conducted continued to be so in a second time window from 2000-2001. However, their research did not comprise visualisations of the cluster characteristics. In addition, their research only showed results for one fixed combination of variables. Lastly, there was no insight as to how the winning clusters performed against a benchmark index, which is potentially of interest.

This research provides opportunities to address the limitations of existing literature as described above. The team looks to incorporate interactivity features and appropriate visualisations into the sub-module as part of improving on existing work.

4. THE APPLICATION

4.1 System Design & Architecture

After careful consideration and specially taking into account the need to bring financial literacy to the masses we have decided to design and develop Investing101 based on R Shiny. Shiny is an open source R package that provides an elegant and powerful web framework for building web applications using R. Shiny helps you turn your analyses into interactive web applications without requiring HTML, CSS, or JavaScript knowledge.

The design of the Shiny server allows us to make the application available across the Internet (i.e. to the masses). The Shiny Server will host each app at its own web address and automatically start the app when a user visits the address. When the user leaves, Shiny Server will automatically stop the app. This reduces the need for maintenance and infrastructure costs, thus allowing us to provide our application to a wider audience for FREE.

4.2 Visualizations

The data visualisations within the application are built using a host of open source R packages. They are 'tmap,' 'tidyverse,' 'forecast,' 'tseries,' 'readxl,' 'tidyquant,' 'dygraphs,' 'TSstudio,' 'plotly,' 'tsibble,' 'ggplot2,' 'tidymodels,' 'modeltime,' 'modeltime.ensemble,' 'timetk,' 'glmnet,' 'randomForest,' 'shiny,' 'readr,' 'DT,' 'shinydashboard,' 'CandleStickPattern,' 'ggpubr,' 'dplyr,' 'cluster,' 'formattable,' 'reshape2,' 'zoo,' 'tibble,' 'ggdendro,' 'knitr,' 'kableExtra,' 'dendextend,' 'scales,' 'data.table,' 'corrplot,' 'factoextra.'

The following packages are discussed in detail as they are integral to the workings and logic of the application.

The Tidyquant package integrates the most useful functions from the xts, zoo, quantmod, TTR, and PerformanceAnalytics packages. It is also one of the key sources of data for our application.

The timetk package is part of the modeltime ecosystem for time series analysis and forecasting. The forecasting system is extensive and includes many ready made algorithms, allowing for ensembling and resampling, Machine Learning and Deep Learning. It is mainly used to calculate and analysis financial ratios and also the plotting of time series/financial data.

The dygraphs package is able to handle huge data sets. It also creates interactive plots out of the box, allowing the user to use our app more interactively. dygraphs is also Highly customizable and works in most browsers and mobile devices which once again allows us to deliver our app to the masses. plotly is another package that we use for creating interactive web graphics via the open source JavaScript graphing library plotly.js which allows use to create customizable interactive plots used for visualization in our application.

Modeltime unlocks time series models and machine learning in one framework meaning that is no need to switch back and forth between various frameworks. Modeltime incorporates a streamlined workflow for using best practices to forecast. Its many different algorithm and models means that the application can be easily scaled up to include more forecasting models with little effort.

The dendextend package allows many functions for manipulating and visualizing dendrograms. In the case of hierarchical clustering, it allows for visualizing and comparing trees and adjusting a tree's graphical parameters, such as colour, size, type etc of its branches.

4.3 Application optimisation

4.4 Charting

4.4.1 Charting and Technical Analysis

In this module, users are able to visualize the chosen stock symbol using a candlestick chart. Open, High, Low, and Close prices for the selected period are shown on the chart.

In addition, users are able to choose between 2 sets of technical indicators: Simple moving averages and Bollinger Bands. Users are able to see the chosen symbol on 1 chart, and the accompanying benchmark index chosen on another chart. Both charts are created in an interactive manner using the plotly package.

Finally, multiple features based on candlestick patterns are created and available through a table. Such candlestick patterns inform users of certain bullish or bearish price action patterns observed. This provides a different perspectives to users incline towards technical analysis.

4.4.2 Correlation Analysis

In this sub section, users are able to: - Choose 4 stock symbols and visualize their monthly adjusted prices Visualize any anomalous points for these 4 stock symbols using an interactive chart - Visualize the 6 month rolling correlation between the 4 stock symbols, and the accompanying benchmark index

4.4.3 Portfolio Analysis

In this sub section, users are able to: - Create 3 portfolios with different weightages for the 4 chosen stock symbols plus the accompanying benchmark index - Visualize the monthly returns (in %) for the 3 different portfolios on an interactive plot - Visualize an equity curve of the 3 portfolios, where users are able to choose the starting equity amount

4.5 Time Series Forecasting

This sub-module of the dashboard is a tool to carry out time series forecasting on a stock. For this sub-module, forecasting will be carried out on a chosen ticker. The intent for this sub-module is to allow the user to identify future trends of a stock and entry/exit opportunities. The example given will be based on the technology company APPLE (AAPL).

4.5.1 Data Preparation

The sole data source for this sub-module is the tidyquant package on R.

We use the tidyquant package to pull historical prices of the stocks. The extracted data is then broken down into training and testing sets to build the forecasting model. The dashboard is reactive and will be able to pull any stock ticker chosen by the user as long as it is available through Tidyquant.

4.5.2 Model building

A few models are used and built in this edition of the application.

1. Auto-ARIMA

ARIMA, short for 'Auto Regressive Integrated Moving Average' is a model that describes a given time series based on its own past values with its own lags and the lagged forecast errors, so that those lags and delay correlations can be plugged into the ARIMA equation to forecast future values.

An ARIMA model is characterized by 3 terms: p, d, q. Which are automatically optimised in the auto-ARIMA model

where,

p is the order of the AR term - We can determine the required number of AR terms by inspecting the Partial Autocorrelation (PACF) plots. PACF can be described as the correlation between the series and its lag, after excluding the contributions from the intermediate lags.

q is the order of the MA term - The ACF tells how many MA terms are required to remove any autocorrelation in the stationarized series.

d is the number of differencing required to make the time series stationary

2. Prophet

Prophet is an open source library published by Facebook that is based on decomposable (trend+seasonality+holidays) models. It gives us the ability to make accurate time series predictions using simple and intuitive parameters and is able to factor in the impact of custom seasonality and holiday. Prophet holds up well against missing data and shifts in the trend, and typically handles outliers well.

$$y(t) = g(t) + s(t) + h(t) + et$$

g(t): piecewise linear or logistic growth curve for modelling non-periodic changes in time series s(t): periodic changes (e.g. weekly/yearly seasonality) h(t): effects of holidays (user provided) with irregular schedules et: error term accounts for any unusual changes not accommodated by the model

3. Elastic Net (GLMNET)

Elastic Net is a combination of the Ridge and Lasso methods. Its name comes from the notion that it is similar in structure to a stretchable fishing net that retains all the big fish (variables) while expanding to let the smaller unimportant fish go away. This model predicts fitted values, logits, coefficients and more from a fitted "glmnet" object - in this case the time series dataset.

4. Random Forest

The Random Forest method is a popular and effective algorithm. Random Forest can be used for time series forecasting, by transforming the time series dataset into a supervised learning problem first. This is expedited using the modeltime package and its readymade functions.

5. Prophet XG Boost

A modified version of the Prophet method from Facebook. The prophet XGBoost algorithm builds on the base prophet and uses xgboost to improve performance on detecting seasonality. XGBoost stands for eXtreme Gradient Boosting and it is used for developing fast and high performance gradient boosting tree models. This algorithm is provided in the base modeltime package.

4.6 Clustering

The next sub-module of the dashboard is a tool to carry out hierarchical clustering on stocks. For this sub-module, clustering analysis will be carried out on 89 current component stocks of the NASDAQ-100 Index. The Nasdaq-100

Index is a basket of the 100 largest non-financial companies listed on the Nasdaq Exchange in the US. Companies listed in the Nasdaq-100 Index largely belong to the technology sector, and some household names include Apple, Google, Tesla and Facebook. There has been huge investor interest in technology stocks following their outperformance in 2020 and the ongoing hype of digitisation. We choose to focus on stocks in the NASDAQ-100 Index for this sub-module to provide users with a tool to aid their decisions in investing in the biggest US-listed technology stocks.

4.6.1 Hierarchical Clustering

Clustering is an unsupervised machine learning technique which finds patterns in unlabelled data, grouping objects in clusters based on their similarity. Cluster analysis can be broadly separated into hierarchical and non-hierarchical methods. For our clustering sub-module, we choose agglomerative hierarchical clustering as the mode of clustering.

In agglomerative hierarchical clustering, each item begins in its own cluster and a distance matrix is computed based on a chosen distance metric. The two closest clusters are merged, and the distance matrix is updated. The process repeats until a single cluster remains.

In our sub-module, since the clustering variables are in continuous form, we use Euclidean distance for computing proximity as a default option.

$$D_{euclidean}(p1, p2) = \sqrt{(x_{p1} - x_{p2})^2 + (y_{p1} - y_{p2})^2}$$

We apply the Ward algorithm as the default option to merge the individual clusters into higher order clusters. Ward's minimum variance criterion minimizes the total within-cluster variance [6]. At each step, the two clusters which lead to the smallest increase in total within-cluster variance after merging are merged.

4.6.2 Clustering variables

For this sub-module, hierarchical clustering will be carried out on 5 variables which have been selected to cover a holistic range of financial indicators of a company. The time horizon for a clustering is one year. The 5 variables are as follows: "1-Year Return," "1-Year Volatility," "Dividend Yield," "Return on Assets," "Total Debt to Total Assets."

4.6.3 Data Preparation

There are 2 main data sources for this sub-module: 1) Bloomberg Terminal 2) tidyquant package on R

The Bloomberg Terminal is used to pull the historical yearly financial ratios and volatility for the stocks. Bloomberg nicely stores these data points which can be easily pulled into csv format. For each stock in our stock universe, yearly historical values from 2015-2020 are pulled for the variables as mentioned above.

We use tidyquant package solely to pull historical prices of the stocks, which we then convert to yearly returns. The reason we use tidyquant rather than using Bloomberg is because tidyquant package has functions which can help us automatically calculate returns for different periodicities, such

as annual or monthly. In this context, we look at yearly returns from 2015-2020 for the stocks in our stock universe.

5. CASE STUDY

5.1 Technical Indicators

5.2 Clustering

We demonstrate how to use the Clustering Analysis sub-module of our dashboard by carrying out a round of clustering analysis on a combination of selectable variables.

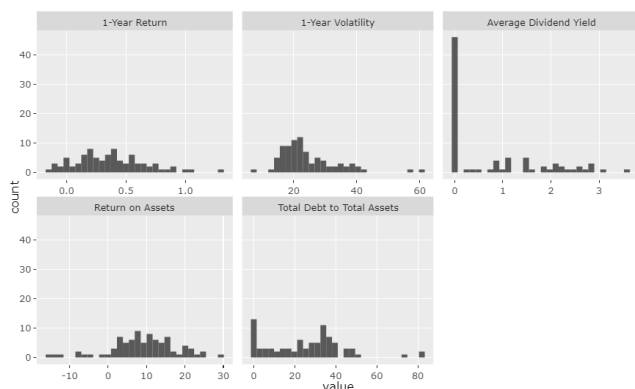
On the side panel of the dashboard, interactivity features are available to allow the user to select the following: "Year," "Variables to cluster by:" "Distance Method," "Linkage Method" and "Number of Clusters."

The screenshot shows a dashboard interface for clustering analysis. It includes a dropdown menu for 'Year' set to 2017. A section titled 'Variables to cluster by:' contains five checked checkboxes: '1-Year Return', '1-Year Volatility', 'Dividend Yield', 'Return on Assets', and 'Total Debt to Total Assets'. Below this, there are two dropdown menus: 'Distance Method' set to 'Euclidean' and 'Linkage Method' set to 'Ward D2'. At the bottom, there is a slider for 'Number of Clusters' ranging from 5 to 20, with the value currently set to 20.

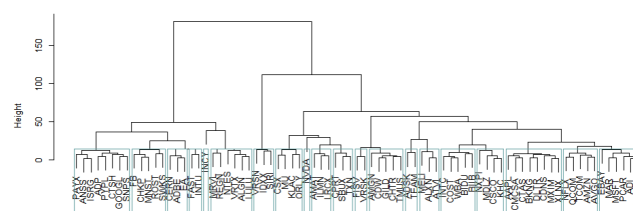
For this case study, we carry out clustering in 2017 on all 5 variables. We select Euclidean to be the distance measure to compute the distance matrix, and Ward D2 as the agglomeration method for the hierarchical clustering algorithm. The resulting dendrogram is cut into 20 clusters.

5.2.1 Pre-Clustering: Univariate and Correlation Analysis

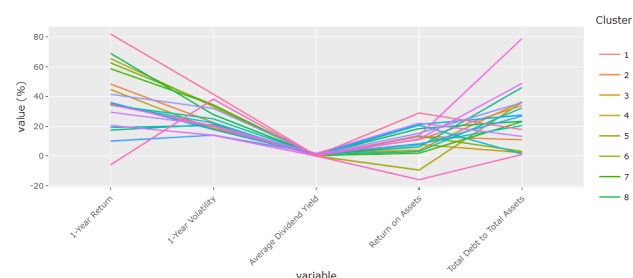
Users will be able to visualise the distribution of the clustering variables they selected for the selected year.



Before clustering, it is essential to check for multicollinearity. If there is a high degree of collinearity between variables, they will not be sufficiently unique to identify distinct clusters. In addition, if highly correlated variables are used for cluster analysis, the aspects covered by these variables will be given excess weightage in the clustering output. We see from the following correlation plot that multicollinearity is not an issue.



Below the dendrogram is a line graph showing the combination of variable values for each cluster. For each cluster, the value seen on the line graph for each variable is the average value of that variable for all the stocks in the cluster. For example, if Cluster 19 has a value of -6% for 1-Year Return, it means that the average 1-Year Return for all the stocks in Cluster 19 is -6%. There is an interactivity option to isolate the view for a single cluster to strip out the noise from the other clusters.



Next, the user will be able to see the breakdown of each cluster's constituent stocks in a table. For example, we see from the table below for our case study that Cluster 4 consists of the stocks ADP, ANSS, CTSH, GOOGL, ISRG, PAYX, PYPL and SNPS.

Cluster	Stocks
1	AAPL,BKNG,CDNS,CMCSA,CTAS,DLTR,MXIM,XLNX
2	ADBE,CERN,EA
3	ADI,EBAY,MAR,MCHP,MSFT,PCAR
4	ADP,ANSS,CTSH,GOOGL,ISRG,PAYX,PYPL,SNPS
5	ADSK,TEAM
6	ALGN,LULU,MRVL,NTES,REGN,VRTX
7	ALXN,ATVI,MELI
8	AMAT,ILMN,LRCX
9	AMGN,CDW,CHTR,GILD,TMUS
10	AMZN,AVGO,NFLX,QCOM,TCOM

The final output on the main panel is each individual cluster's return and volatility profile. An additional panel calculating the Returns to Volatility ratio is included, arranged in descending order. This way, users can see instantly which cluster performed the best in terms of risk-adjusted returns for that year. We see that Cluster 4 performed the best in 2017.

5.2.2 Clustering Results

The main panel of the dashboard has 4 main outputs. At the top of the panel is the dendrogram showing how the stocks have been cut into clusters, in this case 20 clusters.

	Cluster	Returns to Volatility	1-Year Return	1-Year Volatility
1	4	2.51	44.75	17.84
2	8	2.48	69	27.83
3	2	2.15	48.33	22.45
4	13	2.03	36	17.75
5	5	1.97	65.5	33.16
6	20	1.97	82	41.58
7	3	1.89	35.5	18.78
8	6	1.83	62.67	34.31
9	18	1.74	34.67	19.87
10	1	1.72	34.88	20.22

Showing 1 to 10 of 20 entries

Previous 1 2 Next

5.2.3 Clustering Results - Backtesting

We then explore a use case as to how to use the clustering results to aid in investing decisions. Following from the finding that Cluster 4 stocks performed the best in 2017, the user may want to use this to explore if outperformance of a cluster continues into the next year 2018. This is an example of momentum investing, a strategy that aims to capitalize on the continuance of existing trends.

5.3 Forecasting

Before using the forecasting module, users would have identified a few candidate stocks that he or she would want to invest in, and so this brings up the questions such as “When do I invest? Is now a good time?” It also leads us into our last and final sub-module which is time series forecasting on a stock pick.

Users will input the selected stock ticker into the side panel. Here users will also need to select a few other things, namely:

1. Data start date (which determines the amount of data ingested for the forecasting model)
2. Accuracy measure (which will determine which forecasting model results to retain based on how users like to quantify accuracy)
3. Forecasting period (the period ahead in which users would like to forecast between 1 - 4 weeks.)

In this case we will use the following inputs:

- AAPL
- 1st Oct 2020
- MAPE
- 1 week

After the forecasting model is run in the server, two outputs will be created.

5.3.1 Accuracy Table

The accuracy table shows the performance breakdown of each one of the 5 forecasting models by different accuracy measures. This gives a brief overview of how each model fared compared to the others and which 3 of the models were actually retained. In the test case below, you can see that based on the selection of MAPE. The ARIMA/Prophet/ Prophet with XG Boost models had the superior performances. This means that those 3 models were eventually taken in to form the ensemble model while the other

2 (GLMNet/Random Forest) were dropped. Of course, the user can change this by simply changing the input “Accuracy Measure.”

Forecasting Sub-Models Accuracy Measure Results

Download Current Page Download Full Results Search:

	.model_id	.model_desc	.type	mae	mape	mase	smape	rmse	rsq
1	1	ARIMA(0,1,0)	Test	5.2169979	3.9081019	3.0098144	3.9916197	5.4067554	
2	2	PROPHET	Test	14.918734	11.192956	8.6069848	11.868532	15.056655	0.3757721
3	3	GLMNET	Test	5.8753307	4.4038416	3.3896228	4.5096115	6.0710207	0.02695616
4	4	RANDOMFOREST	Test	8.2255008	6.1684791	4.7454937	6.3702698	8.3456734	0.05942373

PROPHET W/

Showing 1 to 5 of 5 entries

Previous 1 Next

5.3.2 Forecasting plot

After understanding what goes into the ensemble model. The user can then see what the ensemble model gives as an output in the final plot of the application. As seen in the chart below, the redline represents the forecasted data. The ensemble forecast takes in the results of the 3 top performing models and calculates the mean, which is then presented as the forecast of the ensemble model.

While it is transparent to the user, under the surface, all the testing and training of the models are still done and evaluated by the accuracy measure defined by the user. The models are also calibrated and refitted on the actual data before being presented to users as the final ensemble forecast.

Looking at the forecast for AAPL over the next week. The prices are expected to remain stable and hence it might be good for the users to wait before taking up a position.



6. CONCLUSION AND FUTURE WORK

This paper set out the development of a web application targeted at investors to provide useful tools to aid them in their investing process that traditional sources do not provide. This was motivated by the heightened interest in investing following the market volatility which was brought about by COVID-19.

The application was developed using the Shiny architecture on R, with an emphasis on the usage of interactivity features and data visualisations to allow users to derive insights from financial data. Going forward, we have the following ideas to further improve and expand the scope of the application:

1. For the Charting sub-module, we can include a wider range of technical indicators for the investor to overlay on the candlestick charts. A backtesting module can

also be incorporated to simulate trading performance based on technical analysis.

2. For the Time Series Forecasting sub-module, we can potentially include more choices of forecasting models which the investor may wish to explore. This will also allow for comparison of forecasts across models.
3. For the clustering sub-module, explore incorporating a wider range of financial variables to cluster upon. The financial variables data used in this sub-module was manually pulled from Bloomberg as there is no current R package that allows for the pulling of historical financial variables. Work can be done to pull data from a wider range of both financial variables as well as stocks, so that investors can do cluster analysis on more classes of stocks other than technology stocks, and also on more financial variables than is currently available on our application.

7. TABLES

Financial Ratio	Definition
1-Year Return	Return on an investment generated over a period of one year, expressed as a percentage of the initial amount of investment.
1-Year Volatility	Statistical measure of the rate of fluctuation in the value of an investment over 1 year. Volatility is also known as standard deviation.
Dividend Yield	Shows how much a company pays out in dividends each year relative to its stock price, expressed as a percentage.
Return on Assets	A type of return on investment (ROI) that measures the profitability of a business in relation to its total assets. The higher the return, the more productive and efficient the use of economic resources.
Total Debt to Total Assets	A leverage ratio that indicates the percentage of a company's total assets that are financed with debt. The higher the ratio, the higher the leverage and financial risk.

References

- [1] Berdiell, A.G. 2015. ANALYSIS OF FINANCIAL MARKETS: TECHNICAL ANALYSIS AGAINST TIME SERIES. Department of Probability, Logic and Statistics - University of Barcelona. (Jul. 2015), 1–75.
- [2] Da Silva, S. et al. 2005. Stock selection based on cluster analysis. Economics Bulletin. 13, (Nov. 2005), 1–9.
- [3] Dey, S. et al. 2016. Forecasting to classification: Predicting the direction of stock market price using xtreme gradient boosting. (Oct. 2016).
- [4] J. J. Wang, Z.G.Z., J. Z. Wang 2012. Stock index forecasting based on a hybrid model. Omega - The International Journal of Management Science. 40, (Dec. 2012), 758–766.
- [5] Markowska-Kaczmar, U. and Dziedzic, M. 2008. Discovery of technical analysis patterns. (Oct. 2008), 195–200.
- [6] Wikipedia 2021. Ward's method — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Ward's%20method&oldid=991647867>.