NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

Variation of the relative stock prices positions on an abstract geometry space

Rui Carlos Nunes Ribeiro

Dissertation proposal presented as partial requirement for obtaining the master's degree in Statistics and Information Management, with a specialization in Risk Analysis and Management

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

NOVA Information Management School

Instituto Superior de Estatística e Gestão de Informação

Universidade Nova de Lisboa

VARIATION OF THE RELATIVE STOCK PRICES POSITIONS ON AN ABSTRACT GEOMETRY SPACE

by

Rui Carlos Nunes Ribeiro

Dissertation proposal presented as partial requirement for obtaining the master's degree in Statistics and Information Management, with a specialization in Risk Analysis and Management

Advisor: Professor Doutor Rui Alexandre Henriques Gonçalves

Advisor: Doutor João Pires da Cruz

February 2022

ABSTRACT

The Markowitz theorem is the basis of current investment fund management. Based on time series pricing treatment, a portfolio is optimized by minimizing the risk for the same expected return or maximizing the return for the same risk. This approach assumes that time series have sections in Gaussian time, with a finite standard deviation and an expected stationary value, assumptions that are not verified. The approach proposed in this paper is to use unsupervised neural networks to make a distributed representation of stocks within a potential portfolio to build an abstract geometry space where the variations of the relative position of shares in that space will be studied. The work ends when the abstract space is built and shows the distance evolution of the set of actions in relation to the others.

KEY WORDS

Portfolio management; Modern Portfolio Theory; Asset Correlation; Artificial Neural Networks; Machine Learning

Index

1. INTRODUCTION

1.1. BACKGROUND AND PROBLEM IDENTIFICATION

When addressing the portfolio creation and management subjects it is inevitable to mention the Markowitz Portfolio Selection Theory (1952) and his map for "Efficient Diversification of Investments" (1959) as the starting point for more advanced concepts. His contribution to the financial community with indirect impact in all the society is acknowledged by everyone that are in some way involved in the financial world as his theory postulation marked a new era, completely changing the way investors perceived and assembled their investments.

The changes were so profound that while, nowadays the meaning of expressions like investing, and investment portfolio are clearly understood as a group of financial assets—such as stocks, bonds, commodities, currencies, and funds—purchased to produce income and meet any financial goal, before Markowitz they would sound very exoteric. During a time when asset prices were still printed in paper and there was no internet to help spread the information at the speed we are used today, these expressions were not only barely used but they also resembled to a betting game, where the goal of most investors was to simply find a good stock and buy it at the best price.

After his postulations in 1952 and 1959 there have been many who have dedicated their time into the study of the portfolio selection which inevitably resulted in adjustments and improvements. However, despite the big innovation of this theory, it is also known that for the current reality and market conditions it presents some limitations with the main critics related to the large estimation errors on the vector of expected returns as well as in the covariance matrix - Chopra et al. (2013). Besides this, there is also the fact that financial returns do not exactly follow a Gaussian distribution as Biró et al. (2007) showed and that the correlation between assets are not fixed but vary depending on external events. Furthermore, there is evidence by Mamum et al. (2015) that the investors are not rational and, as stated by Wilford (2012) the basic assumptions that are not applied when implementing Markowitz Portfolio Theory will lead to misleading conclusions or in some cases completely incorrect.

1.2. MODERN PORTFOLIO THEORY

The reason this theory is so important is because it was the first that focused on maximizing the returns investors get from their investments. And for that, they need not only to choose the right assets to invest in but also to decide what is the ideal amount to allocate to a certain investment. This may sound easy, but it is certainly not and there are numerous the individuals and institutions that have been trying to get to the perfect formula, yet unsuccessfully until this moment.

The difficulty of this relies on various reasons like the incredible volume and speed of information or the unknown risks investors might face, but it all starts with the wide range of different assets classes and instruments available to even a bigger number of sectors from where the investors can choose from. For this reason, the concept of portfolio is the basis for every investor. Which means that everyone who wants to invest will need to choose a collection of financial investments like stocks, bonds, commodities, cash, and cash equivalents, including closed-end funds and exchange traded funds to build their portfolio.

Having chosen the assets to invest in, it is then necessary deciding the portions of the capital that will be allocated to each one of them and here is where portfolio management enters. This means selecting and managing a group of investments to meet the financial objectives considering the owner's tolerance to risk, investment objectives, time horizon and other personal constraints. It is easy to understand that according to all these factors, the type of portfolio will vary from investor to investor. And here is where all Markowitz's, his predecessors as well as successors' work came into play, with a solution to the central concept in portfolio management, which is diversification and consequently, asset allocation.

According to Markowitz, for his theory to be valid we need to start by making two assumptions: the first is that investors want to maximize their expected returns and second, that the variance is undesirable. Variance is used in a financial context to measure volatility and it is a very useful indicator of how much risk a particular stock has. The higher the variance, higher the risk is associated to the stock and the potential for higher returns. Markowitz states a rule, "the investor does (or should) diversify his funds among all those securities which give maximum expected return", but also explains that even diversification cannot eliminate all variance (1952). Markowitz then explains that the total expected return and the risk of the portfolio will be affected by the number of securities within a portfolio and their covariance relationships. Other consequence is that if the right securities are chosen, the suitably composed portfolio may have a lower risk as a whole than to the sum of risks of its individual components. Essentially, the MPT can be described as an investment framework for the selection and construction of investment portfolios based on the maximization of expected returns of the portfolio and the simultaneous minimization of investment risk (Fabozzi, Gupta, & Markowitz, 2002).

Covariance is the measure of the strength of the relationship relation of two variables (or securities) and is expressed numerically by the correlation coefficient. This correlation coefficient's values can vary between -1.0 and 1.0. A perfect positive correlation, means that the correlation coefficient is exactly 1 which implies that as one security moves the other security will move in the same direction. A perfect negative, -1, correlation means that two assets move in opposite directions, while a zero correlation implies no linear relationship at all. The goal then is to choose assets that have a negative covariance to one another and build a portfolio that have a standard deviation for the combined portfolio lower than the standard deviation of the individual assets. This way on asset's price return drops, the other will compensate and it will make the rate of return over the years steadier.

The main goal of this dissertation is to study the correlation of a group of stocks, from the Dow Jones Industrial Average Index, in relation with each other in an alternative way from the one previously presented as a first step for a possible improved and innovative approach on asset allocation and portfolio management. For this, a geometry space will be created by resorting to an unsupervised neural network. To do so, having in mind the different perspectives presented in the literature aspects, we will try to answer the following questions:

- How can an alternative method of calculating stock price correlation affect the asset allocation of a portfolio?
- What is the impact of a change in the stock's correlation coefficients in the final portfolio?

Following the known limitations, both explicit and implicit, it is the intention of this work to go in a new direction and build an abstract geometry space which will study variations of the relative position of stock prices in that space, in relation with the remaining stocks of the sample.

2. STUDY RELEVANCE AND IMPORTANCE

The motto of this work was the fact that the economy, like the universe, is an expanding system. Although this may sound simple to comprehend it has deep implications in the way the economy and the financial models work. This is yet, not an original statement as in recent years there has been a closer interrelationship between several areas trying to obtain a more realistic and rich explanation of the natural and social phenomena, namely between physics and financial theory and econophysicists have been applying the principles of physics to the study of financial markets, under the hypothesis that the economic world behaves like a collection of electrons that interact with each other. They do it mainly by using the theory of probabilities and mathematical methods developed in statistical physics to study statistical properties of complex economic systems consisting of many complex units or population (firms, families, households, etc.) made of simple units or humans. In other words, they treat financial and economic systems are as complex systems. Thus, for physicists, studying the economy means studying a wealth of data on a well-defined complex system (Chatterjee et al., 2005). In this field, the analysis of uncertainty, which is crucial in financial analysis, can be made using measures of physics statistics and information theory, namely the Shannon entropy. One advantage of this approach is that the entropy is a more general measure than the variance since it accounts for higher order moments of a probability distribution function.

When dealing with expanding systems like the universe or the economy, due to its constant expansion, the statistical error is not only extraordinarily high, but it is also constantly increasing. Dionisio et al. (2005) analysed the use of entropy as a measure of uncertainty in portfolio management using data collected from the Portuguese Stock Market and concluded that entropy is sensitive to the effect of diversification and is apparently a more general measure of uncertainty than the variance.

Portfolio optimization is a trendy research topic, which has attracted many researchers in recent decades. And it is easy to understand why, better portfolio optimization model means investors can earn more stable profits. This would be easily achievable if it was possible to predict what would any asset price be in the future, but market systems are so complex that they overwhelm the ability of any individual to predict. The causes depend on many factors including but not limited to political conditions, global economy, company's financial reports and performance, the fact that financial data are inherently noisy, that factors can be multicollinear or that relationships between factors and returns can be variable. Practitioners using quantitative factor models have struggled since the 2008 financial crisis, and many traditional factors have become less reliable and less profitable. As a result, some market participants are looking beyond the traditional quantitative approaches to stock selection are developing models that can dynamically "learn" from past data.

With the introduction of artificial intelligence and increased computational capabilities, programmed methods of prediction have proved to be more efficient in predicting stock prices. Rasekhschaffe et al. (2019) stated machine learning algorithms (MLAs) may provide a better approach than linear models since they can uncover complex nonlinear patterns that are hard to tease out with traditional statistical techniques. In fact, machine learning algorithms have proven to be more effective than traditional statistical techniques in many areas outside finance. Since Frank Rosenblatt invented the perceptron, a neural network that could classify images, in 1957, deep learning algorithms can now exceed human accuracy for many images classification tasks.

The use of neural networks in finance is a promising field of research especially given the already availability of large mass of data sets and the reported ability of neural networks to detect and assimilate relationships among a large number of variables. However, the realization of the potentials of neural networks in forecasting the market involves more of an art than science. There are many ways and combinations in which a neural network can be specified, and many types of data can be simultaneously used. Much of that aspect is yet unexplored. The principles that can guide us in effectively using the networks in financial applications remain a fertile area of research. In addition, the application of the technology is yet to be evaluated in some promising areas of economics, finance, and business.

Before the introduction of neural networks in price prediction most of the studies used classical algorithms like Random Walk Theory (RWT), Moving Average Convergence / Divergence or linear models like Autoregressive Moving Average (ARMA) and Autoregressive Integrated Moving Average (ARIMA). Recent works that include machine learning techniques typically resort to methos such as Support Vector Machine (SVM), Random Forest (RF) and other based on neural networks such as Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN). However, most of the previous work has been around the use of machine learning techniques to predict the stock market prices and there are not many dedicated to the study of correlation between assets.

Among all the studies the vast majority used Long short-term memory. Long short-term memory (LSTM) neural networks are generated by recurrent neural (RNN) networks and are of relevant significance in many areas. In addition, due to its special storage unit structure, LSTM prevents problems of long-term dependency and helps forecast financial times. A wavelet transformation is used to denote, based on LSTM and a focus mechanism, extract, and train historical inventory data and to create an inventory price prediction model.

Fischer et al. (2018) were the first authors found to use this technique. They applied long-short term memory networks to a large-scale financial market prediction task on the S&P 500, from December 1992 until October 2015 and found that compared to random forests, standard deep nets, and logistic regression, LSTM is the method of choice with respect to prediction accuracy and with respect to daily returns after transaction costs. Following this conclusion others applied the same method to other markets, Wanga et al. (2020) proposed a mixed method consisting of long short-term memory networks and mean-variance model for optimal portfolio formation in conjunction with the asset preselection, in which long-term dependences of financial time-series data can be captured. The experiment used a large volume of sample data from the UK Stock Exchange 100 Index between March 1994 and March 2019. Ta et al. (2020) also proposed a long short-term memory (LSTM) network to predict stock movement based on historical data. The constructed portfolios based on the LSTM prediction model outperformed other constructed portfolios-based prediction models such as linear regression (LR) and support vector machine (SVR) as well as the benchmark Standard and Poor 500 (S&P 500) index in both active returns and Sharpe ratios.

Most of the studies are done using exclusively historical price data but it is also possible to integrate other factors such as news or even to merge different neural networks in order to improve prediction results. Gu et al. (2020) proposed to build an automatic trading system by integrating AI and human stock traders experience where the knowledge and experience of the successful stock traders are extracted from their publications and after that, a Long Short-Term Memory-based deep neural

network is developed to integrate the human stock traders' knowledge in the automatic trading system. Finally, the proposed deep neural network was trained and evaluated based on the historic data of the Japanese stock market. In all the four strategies for stock classification used, the proposed deep neural network-based stock prediction method had better performance than the conventional methods even thought it was also pointed that the best human traders have managed to obtain higher return that the proposed model. On the other hand, H. Wang et al. (2021) proposed a hybrid stock closing price predicting model based on CNN-BiSLSTM. According to the author Bidirectional special long short-term memory was used because it adds "1 − tanh(x)" function to the output gate calculation based on BiLSTM, making it a better predictor of the stock price. CNN is firstly captures and combines the features that influence stock prices of the input data to form high-level data features and then BiSLSTM is used to predict the next trading day stock closing price of the Shenzhen Component Index. The proposed model was considered an efficient model as it performed better than the reference models MLP, RNN, LSTM, BiLSTM, CNN-LSTM, and CNN-BiLSTM.

Furthermore, Ma et al. (2020) used three deep neural networks (DNNs), i.e., deep multilayer perceptron (DMLP), long short memory (LSTM) neural network and convolutional neural network (CNN) to build prediction-based portfolio optimization models which own the advantages of both deep learning technology and modern portfolio theory. From these 3 approaches, DMLP presented the best results when compared with two models built with support vector regression, used as benchmarks. The paper applied component stocks of China securities 100 index in Chinese stock market as experimental data. The same author extended the first study in Ma et al. (2021) with theoretical research on portfolio optimization with stock return prediction. In order to guarantee the high-quality stocks were selected before building the portfolio optimization model, they used five different models that were already used in the past by many other research but never in conjunction, two machine learning models and three deep learning models (Randon Forest, Support Vector Regression, Deep Multilayer Perceptron, Long Term Short Memory neural network and Convolutional Neural Network). In addition to these five models used for future daily return prediction they used as benchmark the autoregressive integrated moving average (ARIMA) model on historical data of China Securities 100 Index component stocks. Veeresh (2021) studied an opening price prediction mechanism using a LSTM neural network based on GOOGL and NKE stocks. According to the authors the results obtained were better than other neural network existing models.

Leea et al. (2018) were the first to attempt to construct global portfolio management strategies using financial network indicators and to suggest how these can be practically applied. They applied several machine learning approaches (logistic regression, support vector machine, and random forest) proving the network indicators were important supplementary indicators in predicting global stock market particularly during market crisis periods. They were not however the only ones to incorporate multiple neural networks. Guresen et al. (2011) tested the effectiveness of 3 neural network models: multi-layer perceptron (MLP), dynamic artificial neural network (DANN) and the hybrid neural networks which use generalized autoregressive conditional heteroscedasticity (GARCH) to extract new input variables. Each model is evaluated real exchange daily rate values of NASDAQ Stock Exchange index. From the 3, the one that had better results was the MLP with its prediction results being only 0.54% different from the realized price. DANN developed by Ghiassi & Saidane (2005), and the hybrid models (GARCH-ANN, EGARCHANN) developed by Roh (2007).

Krauss et al. (2016) developed a statistical arbitrage strategy based on deep neural network (DNN), gradient-boosted-trees (GBT) and random forests (RF) and each is trained with historical data of all the S&P 500 stocks from 1992 to 2015. They generated one day prediction trading signals and only acted on the signals with a determined number of highest and lowest probabilities. Although they only managed to obtain above average returns until 2001, they still claim the results defy the semi-strong form of market efficiency. Vijh et al. (2020) used an Artificial Neural Network and Random Forest techniques to predict the next day closing price for five companies belonging to different sectors of operation. The financial data: Open, High, Low and Close prices of stock are used for creating new variables which are used as inputs to the model. The models are evaluated using standard strategic indicators: RMSE, MAPE and MBE. The low values of these two indicators show that the models are efficient in predicting stock closing price. Huang (2011) developed a methodology where they first used a Support Vector Regression method to generate predicted returns and build a reliable ranking of stocks and then a Genetic Algorithm for the optimization of the model parameters. It the end they were able to outperform the benchmark (the product of the average yearly returns of the 200 stocks in the investment universe).

Other authors followed less common approaches like Random Matrix approach or Bayesian neural networks. Namaki et al. (2011) studied the stock correlation of the Tehran Stock Market applying the Random Matrix Approach to remove common factors among all stocks and then construct a correlation network of the Tehran Stock Market where the stocks were classified into different groups to better analyse the market structure. Ticknor (2013) proposed a Bayesian regularized artificial neural network combined with the Levenberg–Marquardt algorithm taking daily market prices and financial technical indicators as inputs to predict one day ahead closing prices of single stocks. In this work some of the common improvement points such as overfitting and overtraining are taking in account once the network tends to penalize excessively complex models. To determine the effectiveness of the model they ran tests with Microsoft Corp. and Goldman Sachs Group Inc., concluding that the model performed as well an ARIMA model without the need of preprocessing data.

Li et al. (2019) also succeeded when attempted to predict movement direction based upon a single stock historical data from three popular Chinese stock market indexes together with its correlated stocks utilizing a multi-task recurrent neural network (RNN) - Multi-task Market Price Learner (MMPL) to automatically extract diversified and complementary features from individual stock price sequences with weight high-order Markov random fields (MRFs).

Moghaddama et al. (2016) attempted to forecast the daily NASDAQ stock exchange rate with several feed forward artificial neural networks trained by the back propagation algorithm. Differently from other studies, here the authors considered a short-term historical data of just 6 months. Khan et al. (2019) studied price prediction for 7 days in the future and showed for different stock markets that adding the public sentiment and political events to the inputs improve the accuracy of the models. In addition, they verified a positive correlation between companies of the same industry and concluded that the proposed model performed better than existing ones. Kohli et al. (2019) verified the dependency of the Bombay Stock Exchange (BSE) on factors such as commodity prices, historical data and foreign exchange rates utilizing a statistical classification meta-algorithm called AdaBoost.

Moewsa et al. (2020) reinforced the theory that the S&P 500 stock prices used as sample contains lagged correlations and that it can exploited through the application of deep learning models. The particularity of this study is that they showed that even without using the data of the target stock in the inputs, by using trend approximations as features, it is possible to make higher than-average price trend predictions. This statement goes against the random walk hypothesis and the semi-strong and strong forms of the efficient market hypothesis while at the same time enhances the viability of applying deep learning for trend prediction in cross correlated financial time series

Rout et al. (2015) presented a low complexity recurrent Functional Link Artificial Neural Network, a single layer ANN structure, capable of predicting financial time series data applying different learning methods to obtain the optimal weights for the model. More concretely they tried to predict prices from the Bombay Stock Exchange and Standard & Poor's 500 over different time frame periods ranging from 1 day to 1 month. Patel et al. (2014) used multi-Layer perceptron (a type of artificial neural network) techniques, to predict the stock price of companies listed under LIX15 index of National Stock Exchange (NSE). Freitas et al. (2006) took a different approach than all the other papers, with a named autoregressive moving reference neural network (AR-MRNN) to predict stock returns. They used a large dataset with real data from the Brazilian stock market and showed that their model outperformed the Markowitz portfolio selection model showing better return for the same risk.

More recently, Hamida et al. (2022) realized that the results of comparing volatility forecast from neural networks with the with implied volatility from S&P 500 Index futures options using the Barone-Adesi and Whaley (BAW) American futures options pricing model were not significantly different only in the horizon of 55 days, the longer horizon from 3 that were tested (55, 35, and 15 trading days to maturity of a futures contract).

Despite the vast number of papers dedicated to price prediction there are few that directly address the subject of correlation. Choi (2018) applied a hybrid ARIMA-LSTM model to predict stock price correlation coefficient between two stocks. Different time periods combinations of assets were tested and concluded that model performed better than other equivalent financial models. Puerto et al. (2020) addressed the subject of correlation between assets' returns by adding a new criterion to the portfolio selection and measure the clustering effect of the selected assets in relation with those non-selected. They applied a Mixed-Integer Linear Programming to deal with clustering and portfolio selection simultaneously.

As already mentioned, sentiment analysis is one of the well-known tasks and fast-growing research areas in natural language processing (NLP) and text classifications. This technique has become an essential part of a wide range of applications including politics, business, advertising, and marketing. There are various techniques for sentiment analysis, but recently word embeddings methods have been widely used in sentiment classification tasks. Du et al. (2020) presented a method to encode the influence of news articles through a vector representation of stocks called a stock embedding. The method was built to acquire such vectors from stock prices and news articles, using a neural network framework. They tested the model using Reuters and Bloomberg headlines where the proposed neural network framework selected the news articles that were related to the stock. According to the authors the results obtained suggest that the proposed stock embedding can leverage textual financial semantics to solve financial prediction problems.

Knowing how a variable varies depending on another variable, i.e. the correlation of two variables, is a big part of many science fields, from statistics to neurosciences. This becomes more challenging when dealing with multiple variables. Fontana (2017) proposed a set of criteria so that correlation of two variables could be measured among a random number of variables of a given dataset. The method studied was introduced to generalise the concept of Pearson correlation and it was based on a node-base statistical criterion also called hyper occurrence. Contrary to Pearson correlation which is limited to calculate the correlation among only two variables, this model is not only able to detect correlation among an arbitrary number of variables but also to capture any type of correlation that is not linear.

When searching for past contributins in the different fields, the most common recommendations have been to apply more efficient inputs such as technical indicators, news, exchange rates and economic indicators - Ma et al. (2021) - or interest policy and political and economic reforms - Kohli et al. (2019). Further, model generalization and overfitting are pointed as the biggest technical obstacles by Dindi *et al.* (2015). Rasekhschaffe et al. (2019) presented a solution with two ways of reducing the risk of overfitting – feature engineering and forecast combinations and demonstrated that, when properly applied, Machine Learning Algorithms can use a wide variety of company characteristics to forecast stock returns without overfitting. Because the approach taken in this work does not follow any other, these recommendations were considered as much as possible.

3. MACHINE LEARNING

3.1. INTRODUCTION

Machine learning techniques in financial markets, especially for time-series prediction, have been the focus of many researchers as well as investors. Their main goal is to develop intelligent algorithms that can capture the hidden patterns inherent to stock markets. In general, the approaches used by researchers can follow statistical methods such as Linear Regression (LR), Autoregression Moving Average and ARIMA models that usually have the non-realistic assumption that the financial time-series data follows a linear pattern and is stationary and then there are the predictive models for forecasting market stock prices based on intelligent algorithms that resemble biological processes to solve nonlinear and complex problems. Examples of such algorithms are Genetic Programming (GP), Artificial Neural Networks (ANN) and Support Vector Machines (SVM). In the following sections will be presented a highlight the functionally of Artificial Neural Networks (ANN).

3.2. ARTIFICAL NEURAL NETWORKS

The term “Artificial Neural Network” is derived from the biological neural networks that occur in the human brain and allows computer programs to recognize patterns and solve common problems in the fields of AI, machine learning and deep learning. Similar to human brain that has interconnected neurons, artificial neural networks also have neurons that are interconnected to one another in various layers of the networks. These neurons are known as nodes.

Artificial neural networks (ANNs) are composed by node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network. Multilayer perceptrons (MLPs) are the ANNs most used for a wide variety of problems. A common feature among most NLP methods such as Word2Vec is their foundation upon Artificial Neural Networks (NN).

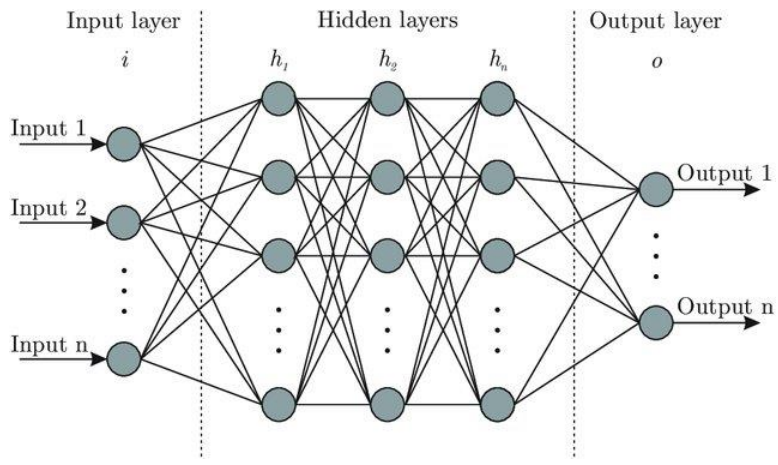

Figure 3.1: Neural network structure

The foundation of neural networks is that information is received from the outside world by the neuron and is then interpreted and processed before producing a result. This process happens multiple times until there is a final output with a result. This idea can be easily visualised through figure 3.2.

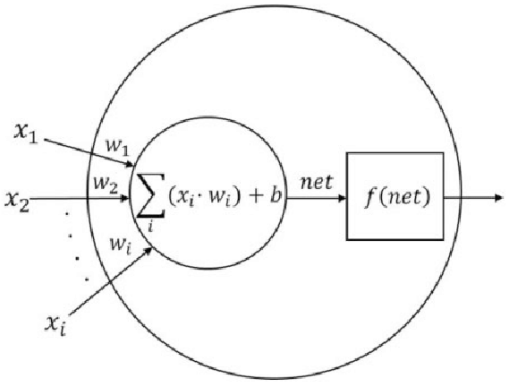

Figure 3.2: Schematic of a neuron

Inside the neuron, all the inputs are summed according to their relative weights, represent as u_j,

$$u_i = \sum_{i=1}^{r} w_{i,j} x_i$$

A sigmoid function $f(u_i - \tau_j)$ is then used to limit the amplitude of the output y_i of the neuron as it shows below.

$$y_i = f(u_i - \tau_j) = \begin{cases} 0 \text{ if } u_i \leq \tau_j \\ 1 \text{ if } u_i > \tau_j \end{cases},$$

Being τ_j the threshold of the specific neuron. Different sigmoid functions can be used but because different functions have different complexities that influence the power required, it is important to choose the correct sigmoid function. Depending on the purpose and the complexity, there can be several layers of neurons between the input and output layers in the NN's architecture, and these are called hidden layers.

3.3. WORD2VEC

Developed by Tomas Mikolov and a team of researchers, this approach was first published in 2013 in the paper 'Efficient Estimation of Word Representations in Vector Space'. Word2vec has two different methods, Continuous Bag of Words which predicts a word based on the inputted context and Skip Gram that works in the symmetric way, predicting a context given a word. Both methods receive a text corpus as input and outputs a vector representation for each word which makes it an effective and efficient technique of representing words as vectors. All vectors have a certain orientation in the vector space making possible to establish a relationship between them and therefore between the corresponding words.

The key to the implementation of word2vec is the construction of 2 complimentary weight matrices to represent words as input and as context or targets. Embedding dimension could be any arbitrary dimension depending upon vocabulary size. The output or prediction of the model is different from the similarity or distances.

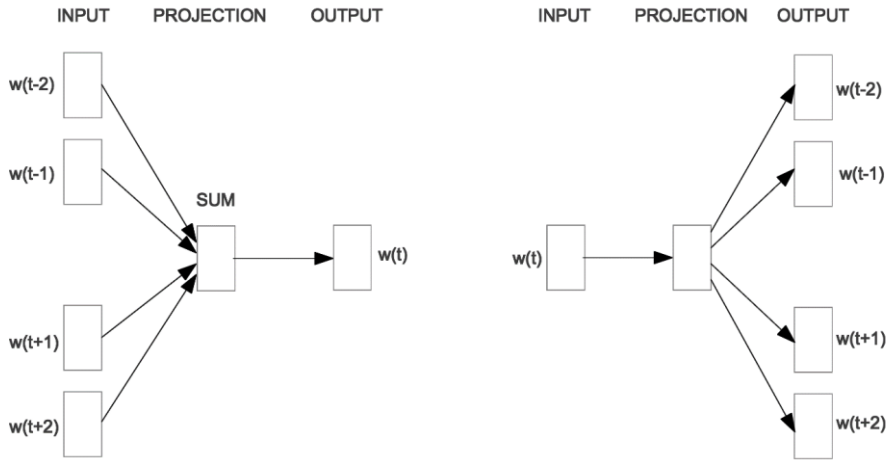

Figure3.1: Schematic of CBOW and Skip-gram models. Mikolov et al. (2013)

3.3.1. Skip Gram

The goal of the Skip − gram (SG) is to predict the context of a given word. Using this methos it is possible to learn a hidden layer that will be used to calculate how probable a word is to occur as the context of the input. Considering a sequence of words $w_1 w_2 w_3 \dots w_i$ as training data, the context of a word w_j is given by the words at its left and right.

Training the models means therefore to maximizing the average logarithmic probability of the context words occurring around the input word over the entire vocabulary:

$$L_{SG}(S) = \frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} \log p(w_{t+j}|w_t)$$

where $S = \{w_1, w_2, w_3, \dots, w_i\}$ is the sequence of the words trained, T is the total number of trained words and c is the training context window, i.e., the number of words that make the interval of context. The size of this window can affect the training accuracy and computational cost. Larger windows lead to higher accuracy as well as higher computational cost. The conditioned probability $p(w_{t+j}|w_t)$ can be written using the softmax function, a function that converts the number it receives into a sum of probabilities.

$$p(w_O|w_I) = \frac{e^{({v'_{w_O}}^{\mathrm{T}} v_{w_I})}}{\sum_{w=1}^{W} e^{({v'_w}^{\mathrm{T}} v_{w_f})}}$$

where $v_n$and v'_n are the vector representations of the word w as the input and output, respectively. Also, W is the number of words in the entire vocabulary.

Equivalently, this means to minimize the *loss* function:

$$E = -\frac{1}{T}\sum_{t=1}^{T}\sum logp(w_j|w_i)$$

The intuition is that words that appear in the same context will have similar vector representations.

However, the main problem of this approach is to compute all the probabilities of the context of each word when the size of the vocabulary has hundreds of thousands or even millions of words. This is where negative sampling comes into play and makes this computation feasible.

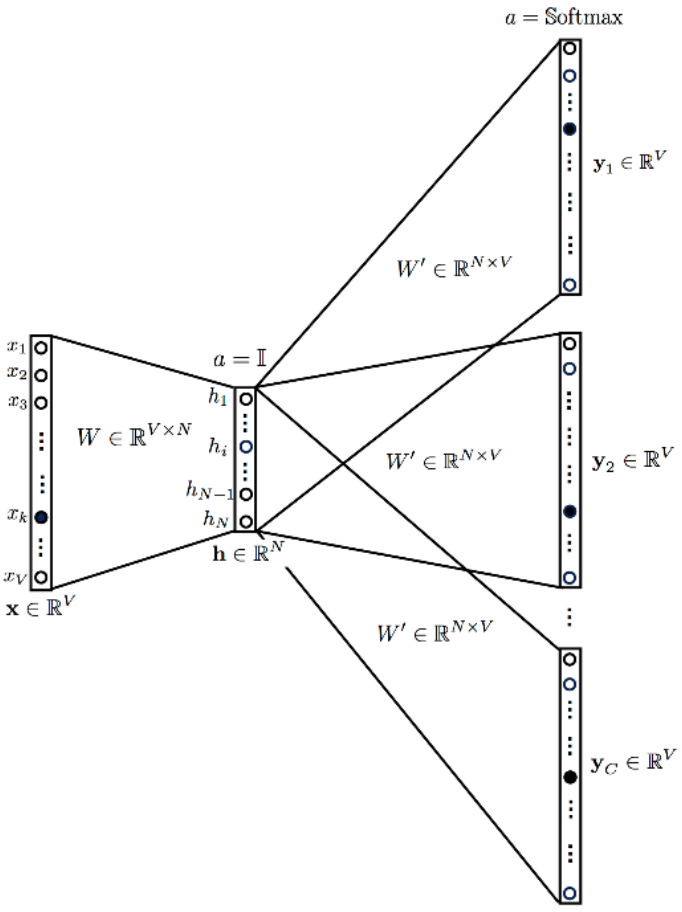

Figure 3.2: The skip-gram model

3.3.2. Softmax

Although both Skip-Gram and CBOW models were revolutionary comparing with the existing ones, in the moment of choosing the best model to use they still had a disadvantage, their computation time duration. Hierarchical softmax (H-Softmax) was the suggested alternative to the flat softmax, inspired by multi-layer binary trees that was proposed by Morin et al. (2005).

H-Softmax essentially replaces the flat softmax layer with a hierarchical layer where the probability of a word is calculated through the product of probabilities on each edge on the path to that node. With this approach, as it is explained in Mnih et al. (2008), the vocabulary is structured in a tree-like arrangement that gives more frequent words a smaller distance from the root to their respective position in the tree.

Mikolov et al. (2002) went one step further defining $p(w|w_I)$ by

$$p(w_O|w_I) = \prod_{j=1}^{L(w)-1} \sigma([\![n(w,j+1) = ch(n(w,j))]\!].{v'_{n(w,j)}}^{\mathrm{T}} v_{w_I}$$

Where $n(w,j)$ is the j^{th} node of the path from the root to the word in question, $L(w)$ the length of this path, $ch(n)$ is an arbitrary fixer child of n, $\sigma(x) = 1/(1+\exp(-x))$ and $[\![x]\!]$ to be 1 if $[\![0]\!]$ is true and -1 otherwise.

3.3.3. Negative Sampling

Negative Sampling is an additional technique used by Mikolov et al. (2013) to improve the Skip-gram model. By defining a new objective function, negative sampling purpose is to maximize the similarity of the words in the same context and minimize it when they occur in different contexts. However, instead of doing the minimization for all the words, it randomly selects a small group of words (usually between 2 and 20, proportional to the corpus size) and uses them to optimize the objective function.

$$L_{SGNS}(s) = \log \sigma\left({v'_{w_O}}^{\mathrm{T}} v_{w_I}\right) + \sum_{i=1}^{k} E_{w_i \sim P_n(w)}[\log(-{v'_{w_O}}^{\mathrm{T}} v_{w_I})]$$

Where SIGMA is the sigmoid function and $P_n(w)$ is called the noise distribution which is defined by:

$$P_n(w) = \frac{U(w)^{\frac{3}{4}}}{Z}$$

Where Z is a normalization constant. Mikolov et al. (2013) stated in their paper that they tried several variations and the one which performed best was to raise the word count to the power of ¾.

Maximizing the objective function will result in maximizing the dot product in its first term and minimizing it in the second term. This means words in the same context will have more similar vector representation while the ones that are found in different contexts will have less similar word vectors.

As it only adds a small number of negative samples in each word, it does not increase the time of computation and improves the accuracy of this model, making it a better option that computing the softmax function over the entire vocabulary.

3.3.4. Cosine Similarity

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is one of the most popular measures of semantic similarity in NLP. Denoting the cosine of the angle between the two vectors, the cosine similarity can take values in the interval [−1, 1].

Cosine similarity has an advantage to other traditional measuring techniques when working with multi-dimensional spaces because it measures the difference in vector orientation.

The formula that describes the cosine similarity is displayed below.:

$$\cos(\theta) = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|\|b\|}$$

The results of this equation may range between [-1,1]. However, in the case of this work, the cosine similarity is employed in a positive space, shaping the outcome of the cosine similarity between [0,1].

4. METHODOLOGY

4.1. LIBRARIES

It is common that in programming projects, open-sourced toolkits and libraries are used to facilitate some processes. Since Python is known for its simplicity, consistency, access to great libraries and frameworks for AI and machine learning (ML), flexibility and platform independence, the selection of python for data preparation, data analysis and model implementations, seems the best decision.

4.1.1. Pandas

Pandas is a quick, flexible, accessible, and open-source python library that allows programmers to better analyse and interpret data. It is used in this work due to its feature of reading data stored in CVS files.

4.1.2. Gensim

Gensim is a free open-source python library designed among others, to represent digital texts as semantic vectors and finding text similarity. It supports a variety of Natural Language Processing such as Word2Vec, Latent Semantic Indexing and Latent Dirichlet Allocation.

4.2. DATA PREPARATION

For this work I used the historical data from the 30 companies that compose the Dow Jones Industrial Average (DJIA) Index from January of 2000 until January of 2021, extracted from Yahoo! Finance.

Preprocessing the data for the model is the very first step of any natural language processing project. The preparation of the data consists of removing all non-alphabetic characters, such as numeric, punctuation and symbol characters, as well as removing stop words. It is also necessary to lower casing, tokenization, stemming and lemmatization.

The raw data for this work was the historical adjusted closing prices because this value considers anything that might affect the stock price after the market closes. The first step was to calculate the daily historical returns in excel.

Then, since the model cannot have numbers as inputs and it needs to be in a text format, it was necessary an adaptation of the returns calculated. The approach taken was to substitute each return into a word (string) that incorporated the name of the correspondent stock followed by interval in the return was inserted. For example, if the return from the Nice Inc. stock for a given day was -2.5%, it would be substituted by “nke32” – from now on generically called “interval word”. When doing this, all letters were lowered to follow the one of the “lower casing” preprocessing processes described before.

When the return is positive the interval word is constructed logically, with the lower extreme number smaller than the bigger extreme. But, because in the corpus cleaning the punctuations are always removed, in our treatment of data we built the interval words corresponding to negative intervals without the minus signals and thus the way to distinguish a positive from a negative interval is if that in the negative ones, the first number is bigger than second one.

Adding to this, the interval words were constructed in steps as follows:

Interval	Step
[-60; -20[	5
[-20; -10[	2
[-10; 10[	1
[10; 20[	2
[20; 60[	5

Table 1: Interval steps to build the interval words

This means that from -60 to -20 and 20 to 60 the intervals were constructed with increments of 5 as so on. The reason there were chosen different interval steps is because not all the returns happen with the same frequency, so to the returns that happen more frequently, between -10% and 10%, the intervals were done in shorter increments, of 1. Given that in the input data used there are no actual words it does not make sense to do some of the usual processes like remove stop words. Also stemming and lemmatization would not have any impact.

5. RESULTS AND DISCUSSION

5.1. RETURNS DISTANCE

From our multi-dimensional vectorial space, it is possible to measure the returns' closeness through the cosine similarity. The cosine similarity can be obtained using the function "most_similar" from the Gensim library. This function computes the cosine similarity between a simple mean of the projection weight vectors of a given return and all the vectors in the model. This requires the user to input the number of most similar returns desired to see. Our generated vector space places together all the interval words corresponding to stock returns which have some degree of correlation, the idea is to use this function to study what is the correlation between each interval word. There is also other very useful function, "wv.similarity" which providers the value of similarity between any two interval words.

Since we have in our data 788 different interval words, we picked one to see what would be the 3, 10 and 20 most similar returns. The interval word chosen is "pg67" - representing a return between 6% and 7% of the stock Procter & Gamble.

The results below show the most similar 3, 10 and 20 returns to "pg67" calculated by the model, each group in decrescent order of similarity. From these results, it is obvious that the most similar interval words correspond to similar returns which would be expectable.

The 5 most probable returns given a return of pg67 are:
1 - msft89: 0.984
2 - hon89: 0.983
3 - nke1416: 0.983
4 - mcd1012: 0.981
5 - ko67: 0.98

The 10 most probable returns given a return of pg67 are:
1 - msft89: 0.984
2 - hon89: 0.983
3 - nke1416: 0.983
4 - mcd1012: 0.981
5 - ko67: 0.98
6 - mmm89: 0.977
7 - ibm1012: 0.976
8 - axp1416: 0.975
9 - msft910: 0.974
10 - unh910: 0.974

The 20 most probable returns given a return of pg67 are:

1 - msft89: 0.984
2 - hon89: 0.983
3 - nke1416: 0.983
4 - mcd1012: 0.981
5 - ko67: 0.98
6 - mmm89: 0.977
7 - ibm1012: 0.976
8 - axp1416: 0.975
9 - msft910: 0.974
10 - unh910: 0.974
11 - cat910: 0.973
12 - ibm267: 0.972
13 - amgn910: 0.971
14 - hon910: 0.971
15 - ibm910: 0.971
16 - unh1012: 0.97
17 - ba1820: 0.97
18 - ibm289: 0.97
19 - v1214: 0.97
20 - gs1416: 0.97

6. CONCLUSION AND FUTURE WORK

Predicting stock market prices is far from being a trivial task. The uncertainty and volatility that characterize stock markets makes impossible to predict price in any timeframe. It is however the question that all investors would like to have the answer for.

In this work we address asset correlation as the first step of a successful portfolio construction although it is not a very studied one. The main objective was to be able to have a model that would calculate the correlation of the Dow Jones Index using its historical data to understand how an alternative method of calculating stock price correlation could affect the asset allocation of a portfolio and what would be the impact of a change in the stock's correlation coefficients in the final portfolio.

The results obtained can only help us understand what the highest probable returns would be given other stock return. Besides the setbacks, this opens ways to new studies. The main one would be to, using these probabilities, get to the correlation coefficients between stocks. After this the answer to the questions made in the beginning of the work would be more easily addressed. Although the purpose and focus of this work has been asset correlation, there is also a possibility to use this model to go even further and try to predict stock price.

7. BIBLIOGRAPHY

Biró, T. S., & Rosenfeld, R. (2007) https://doi.org/10.1016/j.physa.2007.10.067

Chhatwani, M. N. (2019) Reflections on theories of market efficiency https://doi.org/10.1002/pa.1947

Radzimski, M., & Sánchez-Cervantes, J. L., & Cuadrado, J. L. L., & García-Crespo, A. (2014) Predicting stocks returns correlations based on unstructured data sources

Choi, H. K. (2018) Stock Price Correlation Coefficient Prediction with ARIMA-LSTM Hybrid Model arXiv:1808.01560v5

Ma, Y., & Han, R., & Wang, W. (2020) Prediction-Based Portfolio Optimization Models Using Deep Neural Networks," in *IEEE Access*, vol. 8, pp. 115393-115405 10.1109/ACCESS.2020.3003819

Wang, H., & Wang, J., & Cao, L., & Li, Y., & Sun, Q., & Wang, J. (2021) A Stock Closing Price Prediction Model Based on CNN-BiSLSTM https://doi.org/10.1155/2021/5360828

Moews, B., & Ibikunle, G. (2020) Predictive intraday correlations in stable and volatile market environments: Evidence from deep learning

Gu, Y., & Shibukawa, T., & Kondo, Y., & Nagao, S., & Kamijo, S. (2020) Prediction of Stock Performance Using Deep Neural Networks https://doi.org/10.3390/app10228142

Ta, V., & Liu, C., & Tadesse, D. A. (2020) Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading https://doi.org/10.3390/app10020437

Kohli, P. P. S. & Zargar, S. & Arora, S. & Gupta, P. (2019) Stock Prediction Using Machine Learning Algorithms https://doi.org/10.1007/978-981-13-1819-1_38

Khan, W., & Malik, U., & Ghazanfar, M. A., & Azam, M. A., & Alyoubi, K. H., & Alfakeeh, A. S. (2019) Predicting stock market trends using machine learning algorithms via public sentiment and political situation analysis https://doi.org/10.1007/s00500-019-04347

Wang, W., & Li, W., & Zhang, N., & Liu, K. (2019) Portfolio formation with preselection using deep learning from long-term financial data. Expert Systems with Applications, 143. 113042. https://doi.org/10.1016/j.eswa.2019.113042

Li, C., & Song, D., & Tao, D. (2019) Multi-task Recurrent Neural Networks and Higher-order Markov Random Fields for Stock Price Movement Prediction https://doi.org/10.1145/3292500.3330983

Lee, T. K., & Cho, J. H., & Kwon, D. S., & Sohn, S. Y. (2018) Global stock market investment strategies based on financial network indicators using machine learning techniques https://doi.org/10.1016/j.eswa.2018.09.005

Fischer, T., & Krauss, C. (2017) Deep learning with long short-term memory networks for financial market predictions https://doi.org/10.1016/j.ejor.2017.11.054

Zhang, Y., & Li, X., & Guo, S. (2018) Portfolio selection problems with Markowitz's mean-variance framework: a review of literature. Fuzzy Optim Decis Making 17, 125-158 https://doi.org/10.1007/s10700-017-9266-z

Moghaddama, A. H., & Moghaddamb, M. H., & Esfandyari, M. (2016) Stock market index prediction using artificial neural network http://dx.doi.org/10.1016/j.jefas.2016.07.002

Rout, A. K., & Dash, P.K., & Dash, R., & Bisoi, R. (2015) Forecasting financial time series using a low complexity recurrent neural network and evolutionary learning approach http://dx.doi.org/10.1016/j.jksuci.2015.06.002

Krauss C., & Doa X. A., & Huck, N. (2016) Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500 http://dx.doi.org/10.1016/j.ejor.2016.10.031

Patel, M. B., & Yalamalle, S. R. (2014) Stock Price Prediction Using Artificial Neural Network

Ticknor, J. L. (2013) A Bayesian regularized artificial neural network for stock market forecasting, v. 40 (14), 2013, pp. 5501-5506. http://dx.doi.org/10.1016/j.eswa.2013.04.013

Veeresh, K. V. (2021) Applications of Machine Learning in Finance using LSTM Algorithm DOI:10.15680/IJIRSET.2021.1005124

Huang, C. (2011) A hybrid stock selection model using genetic algorithms and support vector regression https://doi.org/10.1016/j.asoc.2011.10.009

Freitas, F., & Souza, A., & Almeida, A. (2006) A Prediction-Based Portfolio Optimization Model https://doi.org/10.1016/j.neucom.2008.08.019

Dionisio, A., & Menezes, R., & Mendes, D. A. (2005) An econophysics approach to analyse uncertainty in financial markets: an application to the Portuguese stock market https://doi.org/10.1140/epjb/e2006-00113-2

Hamida, S., & Iqbal, Z. (2002) Using neural networks for forecasting volatility of S&P 500 Index futures prices https://doi.org/10.1016/S0148-2963(03)00043-2

Mikolov, T., & Chen, K., & Corrado, G., & Dean, J. (2013) Efficient Estimation of Word Representations in Vector Space arXiv:1301.3781

Ma, Y., & Han, R., & Wang, W. (2020) Portfolio optimization with return prediction using deep learning and machine learning https://doi.org/10.1016/j.eswa.2020.113973

Rasekhschaffe, C., & Jones, R. (2019) Machine Learning for Stock Selection, Financial Analysts Journal, 75:3, 70-88, https://doi.org/10.1080/0015198X.2019.1596678

Mikolov, T., & Sutskever, I., & Chen, K., & Corrado, G., & Dean, J. (2002) "Distributed representations of words and phrases and their compositionality", Conference and Workshop on Neural Information Processing Systems

Du, X., & Tanaka-Ishii, K. (2020) Stock Embeddings Acquired from News Articles and Price History, and an Application to Portfolio Optimization 10.18653/v1/2020.acl-main.307

Mamum, Al, & Syeed, Abu, & Yasmeen, F. (2015). Are investors rational, irrational or normal? https://doi.org/10.18533/jefs.v3i04.161

Tse, Chi K., & Liu, J., & Lau, F. C.M. (2010). A network perspective of the stock market. Journal of Empirical Finance, v. 17 (4) pp. 659-667. https://doi.org/10.1016/j.jempfin.2010.04.008

Lyócsa, Š., & Výrost, T., & Baumöhl, E. (2012). Stock market networks: The dynamic conditional correlation approach. Physica A: Statistical Mechanics and its Applications, v. 391 (16) pp. 4147-4158. https://doi.org/10.1016/j.physa.2012.03.038

Namaki, , & Shirazi, A.H., & Raei, R., & Jafari, G.R. (2011). Network analysis of a financial market based on genuine correlation and threshold method. Physica A: Statistical Mechanics and its Applications, v. 390, pp. 3835-3841. https://doi.org/10.1016/j.physa.2011.06.033.

Guresen, E., & Kayakutlu, G., & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. Expert Systems with Applications, v.38 (8), pp. 10389-10397. https://doi.org/10.1016/j.eswa.2011.02.068.

Moghaddam, A. H., & Moghaddam, M.H., & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. Journal of Economics, Finance and Administrative Science, v. 21 (41), pp. 89-93. https://doi.org/10.1016/j.jefas.2016.07.002

Pang, X., & Zhou, Y., & Wang, P., & Lin, W., & Chang, V. (2018). An innovative neural network approach for stock market prediction. https://doi.org/10.1007/s11227-017-2228-y

Vijh, M., & Chandola, D., & Tikkiwal, V.A., & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, v. 167, pp. 599-606. https://doi.org/10.1016/j.procs.2020.03.326

Li, H., & Shen, Y., & Zhu, Y. (2018). Stock Price Correlation Coefficient Prediction with ARIMA-LSTM. Proceedings of The 10th Asian Conference on Machine Learning, PMLR 95:454-469. http://proceedings.mlr.press/v95/li18c.html

Mantegna, R.N., (1999). Hierarchical structure in financial markets, Eur. Phys. J. B. 11

Beyhaghi, M. & Hawley, J. (2013). Modern portfolio theory and risk management: assumptions and unintended consequences, Journal of Sustainable Finance & Investment, 3:1, 17-37 10.1080/20430795.2012.738600

Cruz, J. (2014). Emergent Behavior In Multiplicative Critical Processes and Applications to Economy

Diksha, P. Modern Portfolio Theory: Basis and Strategies, Financial Economics. [Web log post] Retrieved from https://www.economicsdiscussion.net/portfolio-management/theories-portfolio-management/modern-portfolio-theory-basis-and-strategies-financial-economics/29790

Fabozzi, F., & Gupta, F., & Markowitz (2002). The Legacy of Modern Portfolio Theory

Hayes, A. (2020) Correlation. Retrieved fromhttps://www.investopedia.com/terms/c/correlation.asp

Hayes, A. (2020) Variance. Retrieved from https://www.investopedia.com/terms/v/variance.asp

Kochkodin, B. (2020). Everything We've Learned About Modern Economic Theory Is Wrong [Web log post] Retrieved from https://www.bloomberg.com/news/articles/2020-12-11/everything-we-ve-learned-about-modern-economic-theory-is-wrong

Leote, F., Silva, C., & Teixeira, N. A aplicação da teoria de markowitz na euronext.

Markowitz, H. (1952). Portfolio Selection. USA: Journal of Finance.

Marling, H., & Emanuelsson, S. (2012). The Markowitz Portfolio Theory.

Mangram, M. (2013). A simplified perspective of the Markowitz portfolio theory. Global Journal of Business Research, v. 7 (1) pp. 59-70. https://ssrn.com/abstract=2147880

Parkinson, C. (2020). Maximizing Returns for Investors Using Modern Portfolio Theory and the Efficient Frontier. Undergraduate Honors Capstone Projects. 494. https://digitalcommons.usu.edu/honors/494

Peters, O. (2019). The ergodicity problem in economics. Nat. Phys. 15, 1216–1221. DOI: 10.1038/s41567-019-0732-0

Săvoiu, G. & Andronache, C. (2013). Econophysics

Vaclavik, M., Jablonsky, J. (2011). Revisions of modern portfolio theory optimization model. DOI 10.1007/s10100-011-0227-2

Wilford, D. (2012). True Markowitz or assumptions we break and why it matters.

Chopra, V. K., & Ziemba, W. T. (2013) The Effect of Errors in Means, Variances, and Covariances on Optimal Portfolio Choice https://doi.org/10.1142/9789814417358_0021

Chatterjee, A., & Yarlagadda, S., & Chakrabarti, B. (2005) Econophysics of Wealth Distributions http://dx.doi.org/10.1007/88-470-0389-X

Morin, F., & Bengio, Y. (2005) Hierarchical probabilistic neural network language model

Mnih, A., & Hinton, G. E. (2008) A Scalable Hierarchical Distributed Language Model, Neural Information Processing Systems

Bernardo, M. (2020) Construction of Geometries Based on Automatic Text Interpretation

8. APPENDICES

8.1. DATA PREPARATION

```
import pandas as pd

file = pd.read_csv('Returns.csv')

min = -60
max = 40

def calculate_step(lower):
    if lower in range(-20, -10) or lower in range(10, 20):
        step = 2
    elif lower in range(-10, 10):
        step = 1
    else:
        step = 5
    return step
with open('intervals.txt', "w") as my_output_file:
    for i in range(len(file)):
        rows = []
        for column in file.columns[1:]:

            lower = min
            step = calculate_step(lower)
            upper = lower + step

            while(True):
                if file[column].values[i] >= lower and
file[column].values[i] < upper:
                    new_value =
f'{(column.lower())}{abs(lower)}{abs(upper)}'
                    file.loc[i, column] = new_value
                    rows.append(new_value)
                    break
                else:
                    lower = lower + step
                    step = calculate_step(lower)
                    upper = lower + step
        my_output_file.write(" ".join(rows)+'\n')
print("Done!")
```

8.2. MODEL IMPLEMENTATION

```
from gensim.models import Word2Vec

path_to_file = 'intervals.txt'
data = []

with open(path_to_file,'r') as data_file:
  for line in data_file:
        phrase = line.split()
        data.append(phrase)

# Creating Word2Vec
model = Word2Vec(sentences=data, window=30, min_count=1, workers=4, sg =
1)

model.save('vectors.kv')
```

8.3. RESULTS VERIFICATION

```
from gensim.models import KeyedVectors

model = KeyedVectors.load('vectors.kv')

stock = "pg67"
top = 3

print(f"The {top} most probable returns given a return of {stock} are:")
words = model.wv.most_similar(stock, topn=top)
i=0
for word in words:
  i+=1
  print("" + str(i) + " - " + word[0] + ": " + str(round(word[1],3)))


stock1 = 'aapl87'
stock2 = 'cat89'

similarity = model.wv.similarity(stock1, stock2)
print(similarity)
print(f"The similiarity between {stock1} and {stock2} is {similarity}")
```