UCLouvain
FUCaM MONS

LOUVAIN
School of Management

Louvain School of Management

The effect of investors sentiment on the robo-advisor's financial model

Author: Deblander Julien
Promotor: De Winne Rudy
Academic Year 2018-2019
Master degree in Business Engineering

Abstract

The goal of this master thesis is to measure the effect of investors sentiment on the robo-advisor's financial model. Consequently, two research hypotheses have been formulated. The first one aims at confirming or disproving that the Black-Litterman model without investors views offers a better risk-return trade-off to an investor than the Modern Portfolio Theory. To answer the first hypothesis, the Implied Equilibrium Excess return vector has been built in order to build the efficient frontier of the Black-Litterman model without investors views.

In the second hypothesis, we claim that the Black-Litterman model with investors views provides a better risk-return trade-off than the Modern Portfolio Theory. To this end, a sentiment analysis has been used to provide the Black-Litterman model with investors views. Since the sentiment time series showed some nonlinear dependencies with ETFs stock returns. We have used the sentiments as inputs of a Long Short Term Memory neural network to predict future returns.

The thesis highlights that the first research hypothesis is partially validated as for some level of risk an investor is better off with the Modern Portfolio Theory. However, we confirm that the Black-Litterman model with investors views offers a better risk-return trade-off to any investor.

Acknowledgements

I would like to express my gratitude to my supervisor Mr De Winne for his guidance during the writing of this thesis and for giving me the opportunity to study an interesting subject.

I am also thankful to my parents and who supported and encouraged me during my university curriculum.

Table of contents

List of tables

List of figures

List of appendices

1. Introduction

Over the past few years, Robo Advisors (RAs) have been a hot topic in the wealth management industry. In 2019, Assets under Management (AuM) in the RAs segment accounts for $980,5 million worldwide. This trend is expected to grow at an annual rate of 27% resulting in a total AuM of $2,5 trillion by 2023 (Statista, 2019). The number of users is also predicted to increase in the next years as a new generation of clients who are receptive to digital technologies arises. These clients prefer to have permanent control on their investment and relies on multiple sources of information rather than individual financial advisors (Moulliet, Stolzenbach, Majonek, & Völker, 2016). Furthermore, RAs services are offered at lower costs compared to traditional human advisors while delivering the same return to investors. Besides, it is possible to control and customise investment portfolios from multiple devices. Unlike traditional wealth management funds, RAs only require a low minimum investment and provides advanced quantitative methods to optimise the investor’s portfolio.

Beketov, Lehmann, and Wittke (2018) have found that, out of 73 RAs, 40% of them use the Modern Portfolio Theory (MPT) as their methodological framework to allocate assets. Moreover, the MPT gets the highest AuM followed by other predominant techniques such as the Full-Scale Optimisation (FSO) and the Black-Litterman model despite their relatively low occurrences among RAs. The Black-Litterman model is used as an additional component within the MPT to overcome some of its practical problems such as the input sensitivity (Best & Grauer, 1991), estimation error maximisation (Michaud, 1989) and highly concentrated portfolios (Idzorek, 2005).

In parallel, the stock price prediction using sentiment analysis is widely discussed in the literature. Due to the rising popularity of social media, people are inclined to rapidly share their thoughts and opinions (O’Connor, Balasubramanyan, Routledge, & Smith, 2010). In addition, one knows that stock movements are driven by new information and by the belief of investors regarding the future state of the market (Li, Jiang, & Li, 2015). Sentiment analysis serves as a tool to analyse the opinion shared on social networks. In finance, sentiment analysis is commonly used to capture the public mood regarding a stock price or a company. Sentiments can be retrieved from various sources of information such as newspapers, blogs, tweets, etc.

This master thesis aims at analysing the difference between the traditional Modern Portfolio Theory and the Black-Litterman model of a robo-advisor. The latter model is developed by implementing the investor's view through a sentiment analysis performed on tweets. The contributions of this thesis can be summarised as follows:

- The thesis provides a framework to build a robust RA that overcomes some limits of the Modern Portfolio Theory.
- While most research papers focus their sentiment analysis on only few stocks or one well-known index, this thesis aims at analysing a wide range of famous and less famous companies from different countries.
- The experiments show that the incorporation of views, coming from a sentiment analysis, within the Black Litterman model can significantly improve the expected return of the investor for an equal level of risk.

The remainder of this master thesis is structured as follows: Section 2 describes the concept used throughout this thesis. Section 3 details the choice of methodology and the research design. Section 4 exposes the empirical analysis by approaching the Mean Variance Optimisation, the Black-Litterman model, comparing both models and it reviews the limits of such analysis. Finally, Section 5 presents the conclusion and proposes potential future works.

2. Literature review

In this section, we explain different concepts that are useful to realise this thesis. Firstly, the Mean Variance Optimisation framework as well as its pitfalls are detailed. Over a second phase, the standard approach of the Black-Litterman model and the main concepts that are necessary to predict future returns thanks to a sentiment analysis are reviewed.

2.1. The Power of Diversification

Although the concept of diversification appeared years before the Modern Portfolio Theory (MPT), it has been mathematically framed for the first time in 1952 by Markowitz and refers to the adage “Do not put all your eggs in one basket”. It is worth noting that it exists two kinds of diversification: the naïve and the optimal one. The former is a diversification strategy where an investor chooses assets randomly hoping that it will lower the risk of the portfolio due to the varied nature of different securities (Nasdaq, 2016). Notwithstanding its random nature, this strategy turns out to decrease the risk thanks to the law of large numbers. The latter approach aims at reducing the risk exposure as long as the assets are imperfectly correlated. Diversification allows counterbalancing divergent effects undergo by manifold assets and asset classes on account of market events.

Though, diversification can only eliminate one out of two sources of uncertainty. On the one side, the market risk or systematic risk is related to conditions from the general economy such as business cycles, inflation, interest rates and exchange rates. On the other side, firm-specific risk or non-systematic risk corresponds to risks bounded to the firm one invests in. While the first risk cannot be diminished because these macroeconomic factors will affect the return of all companies, the second risk can decline thanks to diversification (Bodie, Kane, & Marcus, 2011).

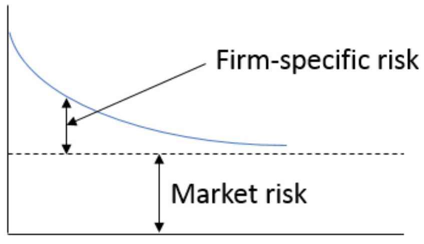


Figure 1 - Firm-specific risk and market risk

Markowitz has also shown that the portfolio risk is reduced depending on the common risk faced by stocks and how prices move altogether. Thoroughly, each security contributes to the volatility of the portfolio according to its total risk, scaled by its correlation with the portfolio. The lower the correlation between a particular security and the portfolio, the lower the portfolio volatility. It implies that the risk of a long position portfolio will be lower than the weighted average volatility of the individual stocks if assets are not perfectly correlated. Conversely, the expected return of the portfolio will equal the weighted average expected return (Berk & DeMarzo, 2014).

While the diversification reduces the volatility, it also allows the investor to potentially grow its wealth and it is known to be "the only free lunch in finance". This means that diversification can provide the investor with benefits at no additional costs.

Whereas, we know that one should invest in various types of assets such as bonds, shares, commodities, REITs, hybrids, et cetera, it remains unclear how many securities one should take care of. Evans and Archer (1968) found that the risk function has an asymptotic form as the number of securities rises. They put emphasis on the fact that firm-specific risk could be diversified by building a portfolio of ten securities. Nonetheless, Elton and Gruber (1977) have shown that portfolio risk has a decreasing marginal effect. They stated that a portfolio of 15 stocks has 32% higher risk than a portfolio of 100 stocks. The latest imminent publication on the subject declares that a well-diversified portfolio should include thirty to forty stocks (Statman, 1987). However, by taking the transaction costs (brokerage fees and taxes) into account Cornu (2018) found that the number of stocks in an optimal portfolio inflates in function of the upfront investment.

2.2. Mean-Variance Optimisation

In 1952, Markowitz published the article "Portfolio Selection" with the aim at proposing a framework to construct portfolios. It has been extended by James Tobin (1958) and by Sharpe (1964) for the theory on asset pricing. These days, Modern Portfolio Theory is seen as a mathematical framework to build portfolios of securities. The prime objective is to maximise the return for a given level of risk or similarly, minimise the risk for a given level of return. In other words, it finds the global maximum/minimum under a set of constraints. In the MPT, we assume that investors know the probability distribution of future returns based on the first and second statistical moment: the mean and variance.

From a mathematical point of view, let us consider assets $A_1, A_2, \ldots, A_n$ $(n \geq 2)$ for a single period. Respectively, μ_i and σ_i represent the expected return and the standard deviation of the return of asset A_i. Consider a multivariate vector μ as:

$$\mu = [\mu_1, \ldots, \mu_n]^T$$

And the $n \times n$ symmetric covariance matrix Σ with $\sigma_{ii} = \sigma_i{}^2$ and $\sigma_{ij} = \rho_{ij}\sigma_i\sigma_j$ for $i \neq j$ as:

$$\Sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1j} \\ \vdots & \ddots & \vdots \\ \sigma_{i1} & \cdots & \sigma_{ij} \end{pmatrix}$$

For $i \neq j$, ρ_{ij} denotes the correlation coefficient of returns of assets A_i and A_j. We denote x_i as the weight of the total fund invested in asset i. While the total return is a linear combination of returns of the underlying assets weighted by x_i, the standard deviation is a quadratic function of the assets' volatility. One can denote the expected return and the variance of the resulting portfolio $x = (x_1, \ldots, x_n)$ as follows:

$$E[x] = x_1\mu_1 + \cdots + x_n\mu_n = \mu^T x$$

And

$$Var[x] = \sum_{i,j} \rho_{ij}\sigma_i\,\sigma_j x_i x_j = x^T \Sigma x$$

Where $\rho_{ii} \equiv 1$ and $\rho_{ij} = \frac{cov(i,j)}{\sigma_i\sigma_j}$

To grasp the impact of correlation on the portfolio risk, one can take three different cases where the correlation equals 1, 0 and -1. For the sake of this example, the generalised case of N assets is reduced to 2 assets.

One can mention that when the correlation equals to 1, the covariance is the product of the volatility of each asset. It means that the standard deviation of the portfolio becomes a linear combination of the volatility of the underlying assets. Therefore, the relationship with the expected return is a straight line.

$$\sigma = \sqrt{\sigma_i{}^2 x_i{}^2 + \sigma_j{}^2 x_j{}^2 + 2x_i x_j \sigma_i \sigma_j}$$
$$= \sqrt{\left(\sigma_i x_i + \sigma_j x_j\right)^2}$$
$$= \sigma_i x_i + \sigma_j x_j$$

When the correlation equals to 0, we end up with a non-linear function.

$$\sigma = \sqrt{\sigma_i{}^2 x_i{}^2 + \sigma_j{}^2 x_j{}^2}$$

Finally, when the correlation equals to -1, we are able to find portfolios with different expected returns for the same volatility mathematically expressed by an absolute value.

$$\sigma = \sqrt{\sigma_i{}^2 x_i{}^2 + \sigma_j{}^2 x_j{}^2 - 2x_i x_j \sigma_i \sigma_j}$$
$$= \sqrt{\left(\sigma_i x_i - \sigma_j x_j\right)^2}$$
$$= \left|\sigma_i x_i - \sigma_j x_j\right|$$

One can understand that any portfolio will be found in the space between the two extreme cases where the correlation equals to 1 and -1.

The minimisation problem of the Mean-Variance Optimisation can be written as follows:

$$min_x \frac{1}{2} x^T \Sigma x$$
$$s.t. \ \mu^T x \geq \mu^*$$
$$e^T x = 1$$

Where e is defined as the unitary vector.

It is worth mentioning that Σ is positive semidefinite given that $x^T\Sigma\, x \geq 0$ for any x. However, we can assume that the Σ is positive definite because we consider that there is no redundant assets in $A_1, A_2, \dots, A_n$ $(n \geq 2)$.

The objective function corresponds to one half of the total variance of the portfolio. It has been added for optimality conditions and does not affect the optimal solution since it is a constant. The latter mathematical formulation is a strictly convex quadratic programming problem given that constraints are linear (Rardin, 2014). Since the MVO is a minimisation (or maximisation) problem under constraints, one forms the Lagrangian:

$$L = \frac{1}{2}x^T\Sigma x - \lambda_1(\mu^T x - \mu^*) - \lambda_2(e^T x - 1)$$

Where $\lambda_1, \lambda_2 \in \mathbb{R}$ and are the Lagrangian multipliers. When we assume that Σ is positive definite and symmetric, the variance is a strictly convex function and it exists a unique portfolio (solution of the Lagrangian), meaning that this portfolio weights vector x^* is the global optimum.

$$\frac{\partial L}{\partial x} = \Sigma x - \lambda_1\mu^T - \lambda_2 e^T = 0$$
$$\frac{\partial L}{\partial \lambda_1} = \mu^T x - \mu^* = 0$$
$$\frac{\partial L}{\partial \lambda_2} = e^T x - 1 = 0$$

From the first equation, the optimum portfolio weight is the form:

$$x^* = \Sigma^{-1}(\lambda_1\mu^T + \lambda_2 e^T)$$

Values of λ_1 and λ_2 are found by substituting x^* into the results of the last two partial derivatives written above:

$$\mu^* = \mu\Sigma^{-1}\Sigma x^* = \lambda_1\mu\Sigma^{-1}\mu^T + \lambda_2\mu\Sigma^{-1}e^T$$
$$1 = e\Sigma^{-1}\Sigma x^* = \lambda_1 e\Sigma^{-1}\mu^T + \lambda_2 e\Sigma^{-1}e^T$$

By writing $a = \mu\Sigma^{-1}\mu^T$, $b = e\Sigma^{-1}\mu^T$ and $c = e\Sigma^{-1}e^T$, one obtains

$$\mu^* = \lambda_1 a + \lambda_2 b$$
$$1 = \lambda_1 b + \lambda_2 c$$

Meaning that

$$\lambda_1 = \frac{c\mu^* - b}{ca - b^2}$$

$$\lambda_2 = \frac{a - b\mu^*}{ca - b^2}$$

The combination of all these portfolios make up the efficient frontier which is commonly represented in a two-dimensional graph where the x-axis corresponds to the standard deviation and the y-axis, the expected return of a portfolio. It consists of a segment among the feasible set of possible portfolios (μ, σ) where for any level of risk, it is impossible to find a higher expected return or for any given expected return there is no lower level of risk. Moreover, by removing the constraint on the expected return and one can find the global minimum variance portfolio weights:

$$x = \frac{\Sigma^{-1} e^T}{e\Sigma^{-1} e^T}$$

To measure the portfolio's risk-adjusted return, it is worth introducing the Sharpe ratio.

$$SR(x|r) = \frac{\mu(x) - r_f}{\sigma(x)}$$
$$= \frac{x^T \mu - r_f}{\sqrt{x^T \Sigma\, x}}$$

As shown by the formula above, the Sharpe ratio subtracts the portfolio expected return $(\mu(x))$ by the risk-free rate (r_f) and divides this quantity by the portfolio volatility $(\sigma(x))$. In other words, the ratio captures the average excess return earned per unit of volatility. The tangent portfolio has the highest Sharpe ratio among all portfolios available as defined by the Capital Asset Pricing Model (CAPM). In general, the greater the Sharpe ratio, the more attractive the risk-adjusted return (Hargrave, 2019).

2.3. Pitfalls of Mean-Variance Optimisation

2.3.1. Normality assumption

Mean Variance Optimisation assumes that asset class returns are normally distributed which is, most of the time, not the case. Effectively, the normality assumption does not take the possibility of extreme market moves into account such as the subprime crisis where U.S. stocks sunk by 57 percent. Another problem that arises from the normality assumption is that the variance is a symmetrical measure of risk. It means that there is no differentiation between upside and downside moves. As an example, an investment return with a positive skew could be seen riskier than it really is, leading to under-allocation to some asset classes and conversely. It has been shown by Xiong and Idzorek (2011) that encompassing skewness and kurtosis in the portfolio optimisation can have a major impact on asset allocation. Quantitative methods such as value-at-risk (VaR) and expected shortfall (ES) attempt to tackle this problem. However, they are known to be incomplete, stress tests should complement the risk analysis of the portfolio (Sironi, 2015).

2.3.2. Error estimation

MVO requires as inputs, securities expected returns, expected standard deviation and expected correlation between assets. Assume there is not any estimation error, MVO allows finding the efficient portfolio weights. Nonetheless, those statistical estimations based on historical data are subject to errors. Solutions of the MVO are subject to input-sensitivity problems since small changes in inputs that lead to large changes in the portfolio weights (Best & Grauer, 1991 ; Lummer, Riepe, & Siegel, 1994). As Michaud (1989) has underlined, the unintuitive character of many optimised portfolios lies in the fact that MV optimisers are estimation-error maximisers. MVO tends to overweight (underweight) assets with large (small) returns, negatively (positively) correlated with small (large) variances. Hence, these assets are prone to large estimation errors. In addition to that, Broadie (1993) has shown that the larger the number of securities in the portfolio, the greater the estimation error. As the number of assets rises, the likelihood that some asset has either a large negative error in the estimation of its standard deviation of return or a large positive error estimation of its expected return increases. Due to the maximisation error property of the MVO, the estimated portfolio performance is an optimistic biased predictor of the actual portfolio.

Estimation errors can cause an efficient portfolio to appear inefficient. In the figure below, the band width is proportionate to the estimation error of the inputs. Harbans and Dhingra (1980) state that the band widens as the expected return increases. This means that portfolios with low expected returns (e.g. short-term fixed-income securities) are estimated with more confidence.

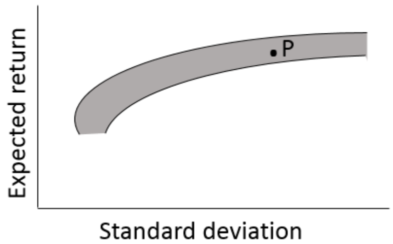


Figure 2 – Effect of estimation errors on the efficient frontier

To overcome this issue, it is advised to use constrained optimisation in order to set an upper and lower bound allocation for a single or a group of assets. This will prevent assets with favourable inputs to dominate the portfolio against other securities. Moreover, investors and economists have proposed the Black-Litterman model to solve the problems mentioned above. This model is detailed in section 2.4.

2.3.3. Static inputs

The MVO also suffers from its inability to capture time-varying data (Swensen, 2009). It only takes static data as input but in the real world, correlations between asset classes do evolve over time. Correlation between assets tends to be higher in period of financial crisis, implying that assets become increasingly correlated with the market, leading to an increase in market risk. Since the MVO does not differentiate the firm-specific and market risk, investors should be concerned about the limitation of the beta exposure. However, one can accede that in the long term, correlation is markedly lower than in the short term (Cronqvist & Siegel, 2014). Consequently, investors can use the MVO to benefit from diversification and stand a better chance of making accurate capital market assumptions.

2.3.4. Time horizon

While the MVO framework is set over a one-year period, many investment objectives can go beyond this timeframe. As Swensen (2009) has written, not taking a longer time span into consideration might lead to suboptimal investment decisions. In point of fact, stock returns exhibit a mean-reverting behaviour, meaning that periods of underperformance in the long run are more likely to be followed by periods of outperformance. Irrespectively, bond returns display mean-averting behaviour, indicating that once bond returns have deviated from their long-run average, there is an increasing probability that they will continue to drift further. Hence, it displays that the relative risk exposure of each asset class is dependent of the holding period. It leads to a wrong risk-return management given that the long risk-return dynamic cannot be represented (Sironi, 2015).

2.4. Black-Litterman Model

The Black-Litterman asset allocation model, created by Fischer Black and Robert Litterman, is a sophisticated portfolio construction method that overcomes the problem of unintuitive, highly concentrated portfolios, input sensitivity, and estimation error maximisation (Idzorek, 2004). This model is based on a Bayesian approach that combines subjective views of an investor regarding the expected return of various assets and the market equilibrium of expected returns. These elements are known in the Bayesian model as the prior distribution and are used to build the new return vector, that is to say the posterior distribution. Recall that the Bayes theorem is written as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where $P(A)$ represents the a priori probability of A (prior distribution). $P(B|A)$ is the sampling distribution while $P(A|B)$ is the conditional probability of A knowing B. Therefore, this probability is the posterior distribution because A depends on B.

The Black-Litterman model allows mitigating the input sensitivity thanks to the integration of the investor's view. Michaud (1989) and Lee (2000) also agree that the model largely mitigates the problem of estimation error by spreading the errors throughout the vector of expected returns. In the MVO, the expected returns vector is essential in the model. However, it has been shown that forecasting techniques such as historical returns (Black & Litterman, 1992), equal mean returns for all assets (He & Litterman, 1999) and risk-adjusted equal mean returns (Litterman, 2003) produce extreme portfolios. This is why the Black-Litterman model introduces the Implied Equilibrium Excess (IEE) return vector (Π) computed using reverse optimisation. In contrast to mean-variance optimisation, reverse optimisation uses portfolio weights as inputs and produces the IEE return vector. The reason to use these weights is that they are sometimes easier to predict and also easier to interpret. Moreover, the IEE is the market-neutral starting point of the Black-Litterman model (Idzorek, 2004). It is the prior equilibrium distribution which follows $N \sim (\Pi, \tau\Sigma)$. Starting from the quadratic utility function:

$$U = w_{mkt}^{T}\Pi - \frac{\lambda}{2} w_{mkt}^{T}\Sigma w_{mkt}$$

And taking the gradient of the utility function with respect to the market weights:

$$\nabla U = \Pi - \lambda\Sigma w_{mkt} = 0$$

The solution to the problem is

$$w_{mkt} = (\lambda\Sigma)^{-1}\Pi$$

Since the weights are already known, one can rearrange the equation to compute the IEE return vector

$$\Pi = \lambda\Sigma w_{mkt}$$

Where:

λ is the risk aversion coefficient of the market ($\lambda = \frac{E(r)-r_f}{\sigma_m^2}$)

Σ is the covariance matrix of excess returns (*n x n* matrix)

w_{mkt} is the market capitalisation weight of the assets (*n x 1* vector)

In the reverse optimisation, the risk aversion coefficient is used as a scaling factor, it characterises the risk-return trade-off. This coefficient can be interpreted as the rate at which an investor agrees to take a smaller expected return for less risk.

As stated before, the Black-Litterman model assumes that prior equilibrium returns are normally distributed.

$$E(r_{eq}) \sim N(\Pi, \tau\Sigma)$$

Where τ is a constant of proportionality which indicates the confidence level of the CAPM estimation of Π. It is assumed to be close to zero since the uncertainty in the mean is much smaller than the volatility of the return itself (Black & Litterman, 1992). He and Litterman (1999) consider it has the ratio of the sampling variance to the distribution variance (i.e. 1/t). The value is usually set between 0.025 and 0.050.

Aside from the market expected return from the market, the model also requires views that have to be specified relative to the expected return vector computed previously. It is also possible to express a level of certainty in the view. These can be expressed either in relative or absolute terms:

- Absolute view:

"I forecast that the Microsoft stock will return 2%."

- Relative view:

"I expect that domestic bonds will outperform foreign ones by 3%."

To translate it into a mathematical expression, three parameters are needed:

- P, a *k x n* matrix of the asset weights of the views. For relative views, row sum must be 0 while for absolute views it should be equal to 1.
- Q, a *k x 1* matrix of the investor expected return for each view.
- Ω, a *k x k* matrix which represents the covariance between the views. Since views are assumed to be independent of each other, this matrix is diagonal. This assumption will not affect the expressiveness of the view as long as the *k* views are not self-contradictory. The inverse is known as the confidence in the investor's view.

Some adjustments can be made to these parameters in order to simplify the model following two theorems (Xing, Cambria & Malandri & Vercellis, 2019).

Theorem 1 (Compatibility of independent views): *Any set of independent views are compatible.*

Theorem 2 (Universality of absolute view matrix): *Any set of independent relative and absolute views can be expressed with a non-singular absolute view matrix.*

Following these two theorems, we end up with a simpler definition of market views which reduces the complexity of the model. Market views on n assets can be represented by three matrices:

- $P_{n,n}$ an identity matrix
- $Q_{n,1} \in \mathbb{R}^n$
- $\Omega_{n,n}$ a nonnegative diagonal matrix

In the most original form of the model, the confidence matrix Ω is set manually according to the investor's experience, however, it is possible to derive the confidence matrix from the covariance matrix (He & Litterman, 1999)

$$\widehat{\Omega} = diag(P(\tau\Sigma)\mathrm{P}')$$

$P(\tau\Sigma)\mathrm{P}'$ can be understood as a covariance matrix of the expected returns in the views and it is therefore a diagonal matrix. Moreover, it is also assumed that the variance of an absolute view on a specific asset is proportional to the volatility of this asset. One can imagine that if an asset underwent high volatility in the past, an investor will be less confident while investing in this asset. The uncertainty of the views results is a random, unknown and independent vector with the form $\epsilon \sim N(0, \Omega)$. Hence, investor's views have the form:

$$Q + \epsilon = \begin{pmatrix} Q_1 \\ \vdots \\ Q_n \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

The fact that the mean error vector equals to zero means that investors do not have a bias against a set of assets. However, as one could expect, if an investor is totally confident in his expressed views the error term will be 0. The Black-Litterman model also assumes that the expected return from market views held by investors is normally distributed

$$E(r_{views}) \sim N(\mathrm{Q}, \Omega)$$

Knowing that the Implied Equilibrium Excess return vector as well as the expected returns from investor's views have a normal distribution, it implies that the posterior distribution of the portfolio returns also have normal distribution $r_{BL} \sim N\left(\overline{\mu}, \overline{\Sigma}\right)$. The mean and the variance of r_{BL} can be found with the following equations.

$$\overline{\mu} = \left[(\tau\Sigma)^{-1} + P'\widehat{\Omega}^{-1}P\right]^{-1}\left[(\tau\Sigma)^{-1}\Pi + P'\widehat{\Omega}^{-1}Q\right]$$

$$\overline{\Sigma} = \left[(\tau\Sigma)^{-1} + P'\widehat{\Omega}^{-1}P\right]^{-1}$$

To sum up, the optimal portfolio produced by the Black-Litterman model is the market equilibrium portfolio plus a weighted sum of investor's views. It means that investor's views can only affect the optimal portfolio when they have returns that vary from those engendered by a combination of the equilibrium portfolio and all other views (Litterman, 2003).

Nevertheless, it is worth mentioning that the model suffers from some limitations. On the one side, it might be difficult for an individual investor to estimate the market portfolio as public markets do not represent entirely the asset universe since some sector might be largely privately held (real estate, commodities, etc.). On the other side, due to the limited availability of the market capitalisation data of illiquid assets, it is hard for individual and institutional investor to estimate their market capitalisation (Walters, 2014). For example, one can wonder whether state-owned oil assets must be included in the calculation of the asset class weights of the market portfolio if an institution invest in natural resources.

2.5. Sentiment Analysis

Two main techniques can be used to perform sentiment analysis: the machine learning and lexicon-based approach. It has been shown that supervised machine learning techniques give better results than unsupervised lexicon-based approach. Nevertheless, supervised methods imply a large number of labelled data to train the model while unsupervised methods do not require such data. For the purpose of this thesis, a lexicon-based technique is used as the retrieved data are not prelabelled.

The 'sentimentr' package[1] is employed so as to get polarity values. It has been built in order to provide a higher accuracy than a simple dictionary lookup method. As a matter of fact, the common lexicon-based method for sentiment analysis is called "Bag of words"[2] meaning that each sentence provided on input has to be tokenized. Practically speaking, the sentence should be decomposed so as to get a data frame containing one word per row. To reduce the number of rows, it is common to remove tokens (i.e. words) that will not be useful in the analysis. Consequently, punctuation, URL addresses, usernames, duplicated messages and stopwords are removed and the remaining words are transformed to lower case. Stopwords are words that fail to add any insight to a sentence, that is to say: the, a, he, etc. In some cases, it can be interesting to delete messages that are not written in the language of interest. For instance, an English-based dictionary will fail to return a polarity value to a sentence written in Spanish. Hence, this can be harmful while drawing a conclusion from an analysis since this sentence should be considered as noise in the model. The next step is to match the words of our data frame with the ones of a dictionary in order to get a specific value for each word. By subtracting positive values by negative values, one is able to get a sentiment score for a sentence.

The sentiment() function[3] of the 'sentimentr' package goes beyond the simplistic "Bag of words" technique by taking valence shifters into account. Indeed, as mentioned before, words that do not add any information on the sentiment should be removed after the tokenizing process. However, words such as "not", "however", "doesn't" are considered as stopwords by

[1] https://github.com/trinker/sentimentr
[2] Datacamp interactive courses "Text mining: Bag of Words" and "Sentiment Analysis in R" have been followed so as to better understand the implication of such technique before choosing the SentimentR package.
[3] See the complete model on appendix A.

most stopword dictionaries. However, these words do add information on the sentiment when it is attached to another word. Valence shifters attempt to fill this gap.

Valence shifters are words aiming at reducing or putting emphasis on the meaning of polarised words and also include negators and amplifiers. Negators are generally adverbs that negate the sentence meaning while adversative conjunctions are conjunctive relations of units that express the opposition of their meanings. Amplifiers are generally adverbs or adjectives that intensify the sentence meaning whereas de-amplifiers decrease the polarity of a polarised word.

In R, the sentiment() function takes on the following arguments.

sentiment(text.var, polarity_dt, valence_shifters_dt, hyphen = "", amplifier.weight = 0.8, n.before = 5, n.after = 2, question.weight = 1, adversative.weight = 0.25, neutral.nonverb.like = FALSE, missing_value = 0)

The *text.var* argument concerns sentences one would like to get the sentiment. *Polarity_dt* is the subjective dictionary used so as to match tokens of *text.var* with the tokens of the subjective dictionary. Even if dictionaries from the 'lexicon' package in R are suitable for the *polarity_dt* argument, the function also make other dictionaries suitable for this argument by adding the *as_key* function inside. *Valence_shifters_dt* is the second subjective dictionary used in the sentiment analysis in order to inverse, increase or decrease the polarity of a sentence when it is matched with a word of *text.var*. *Hyphen* is used in order to replace a hyphen with the desired character, by default the hyphen is replaced by a space. *Amplifier.weight* enables the user to increase the sentiment importance by multiplying the token's sentiment value by a multiplier ranging from 0 to 1. The default value is 0.8. An amplifier becomes a de-amplifier in case of there are an odd number of negators in the sentence Then, *n.before* and *n.after* correspond to $n ** b$ and $n ** a$, respectively and determine the number of words one should take into account before and after the polarised word ($p ** w$) in order to make the polarised cluster $c_{i,j,l}$. By default, the number of words before is five and after is two.

$$c_{i,j,l} = \{p ** w_{i,j,k-n**b}, \dots, w_{i,j,k}, \dots, w_{i,j,k-n**a}\}$$

Question.weight allows polarising the weight of questions with a value from 0 to 1. This argument has been designed since it is supposed that opinion tasks such as course evaluations are more likely to contain polarised sentences. However, in a dialogue, a question is less likely to be polarised and is used to gain information. *Adversative.weight* is the weight to give to adversative conjunctions or contrasting conjunction that overrule the previous clause. Before a polarised word it will up-weight the cluster while an adversative conjunction after a polarised word will down weight the cluster. It stems from the belief that adversative conjunctions (e.g. "but", "however", etc.) amplify the current clause and/or down weight the prior one. If *neutral.nonverb.like* is set to TRUE, the polarity of the word "like" will be neutralised as long as a linking verb: "'s", "was", "is", "has", "am", "are", "'re", "had", "been" has been put before. It comes from the fact that the verb "like" is more likely to be preceded by a noun or pronoun and not by one of the linking verbs mentioned above. If the verb is followed by one of those linking verbs, the meaning of the sentence changes. Finally, *missing_value* replaces NA with the value mentioned in the argument.

Moreover, it is worth mentioning that the punctuation of sentences must not be removed before using the sentiment() function as it is supposed to have an influence on the meaning of the sentence. The function will delete the punctuation by itself except commas, colons and semicolons.

2.6. Information Theory

At first glance, one can look whether there exists causality between two time series thanks to a Granger causality analysis. Even if this tool might be great to spot whether a time series is able to predict another one, its mathematical formulation is based on linear regression modelling. Consequently, the model can only give information about linear features of signals. Therefore, the concepts of Shannon and transfer entropy are explained in the next sections. The advantage of these techniques lie in the sensitivity to nonlinear signal properties which can be worth using with financial time series since they are known to be nonlinear by nature.

2.6.1. Shannon Entropy

Initially developed by Shannon (1948) in the field of thermodynamics where information theory uses probability theory to analyse the information of communication. Shannon information theory has been used in other fields such as computer sciences, statistics, biology, economics, etc. In its definition, information means a decrease of uncertainty meaning that we can measure the information by computing its decrease of uncertainty.

$$H_J = -\sum_j p(j)\log(p(j))$$

Where p(j) is the probability distribution of the discrete random variable J.

While talking about the relationship between two different time series, one often mentions the mutual information which is known as the measurement of information in the information theory. It can be seen as the information a random variable contains about the information of another random variable. However, the mutual entropy fails to determine the direction of the relationship. It is impossible to know if X causes Y or the reverse. The transfer entropy is used to solve this issue.

2.6.2. Transfer entropy

In transfer entropy, the concept of Shannon entropy is combined with the Kullback-Leibler distance (Kullback & Leibler, 1951) and assumes that the underlying processes evolve over time according to a Markov process (Schreiber, 2000). As a result, this technique is able to distinguish the driving factors from the response factors and detect the asymmetry of the interaction between two processes.

If we take two random variables I and J and denote their marginal probability distribution as $p(i)$ and $p(j)$ and their joint probability distribution $p(i,j)$ whose dynamic structures represent stationary Markov processes of order k (process I) and l (process J). The Markov properties implies that the probability to observe I at time $t+1$ is

$$h_I(k) = -\sum_i p(i_{t+1}, i_t^{(k)})\, log(p(i_{t+1}|i_t^{(k)}))$$

Where $i_t^{(k)} = (i_t, \dots, i_{t-k+1})$. $h_J(l)$ can be derived in the same way for J. In the bivariate case, the flow of information from process J to I is measured by quantifying the deviation from the generalised Markov property $p(i_{t+1}|i_t^{(k)}) = p(i_{t+1}|i_t^{(k)}, j_t^{(l)})$ relying on the Kullback-Leibler distance. Therefore, the Shannon transfer entropy is given by:

$$T_{J\to I}(k,l) = \sum_{i,j} p(i_{t+1}, i_t^{(k)}, j_t^{(l)}) * log\left(\frac{p\left(i_{t+1}, i_t^{(k)}, j_t^{(l)}\right)}{p(i_{t+1}, i_t^{(k)})}\right)$$

2.7. Neural Network

This section aims at explaining the basic principle of neural networks. Firstly, it will be explained how a traditional Artificial Neural Network works. Secondly, it deep dives into a type of Recurrent Neural Network (RNN) which is the Long Short Term Memory (LSTM) neural network which turns out to be useful in time series forecasting.

2.7.1. Artificial Neural Network

Artificial Neural Network (ANN) is a statistical bio-inspired model derived from the Machine Learning field. The usefulness of this model comes with the rise of deep learning due to its ability to handle a large amount of data and has become specifically popular in stock price forecasting throughout the years. In this section, we will illustrate the functioning of this model in order to get a better insight about this nonlinear mathematical model.

The aim of the ANN is to reproduce the perceptive capability of living organisms. In fact, through its structure, this model tends to mimic the human brain functioning with the different nodes (called neurons) linked between each other through some specific activation functions. The analogy with the human brain could be pushed even further since, as for the brain, the information in the ANN is transferred between the neurons through the activation functions.

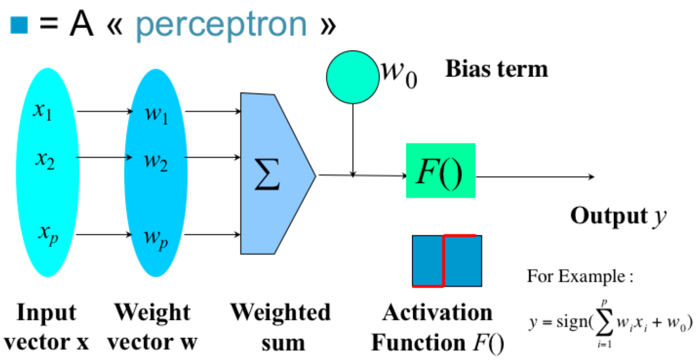


Figure 3 - Process of an Artificial Neural Network

Regarding the ANN process, we firstly have an initial layer, called the input layer, including all the values for the features. Then, these values are multiplied by weights and added to form the hidden layer. Finally, the model applies a specific function, called the Activation Function, in order to transform the weighted inputs into the final outputs.

Regarding the activation function, there exists several possibilities such as using a multinomial logistic function or a sigmoid function. Nevertheless, it is important to know that the outputs are obtained by activating the weighted inputs in the hidden layer through the above formula and a bias term:

$$y = F\left(\sum_{i=1}^{p} w_i x_i + w_0\right)$$

Finally, most of the Artificial Neural Networks use the backpropagation algorithm to optimise the outputs. In fact, there is a cost function characterised by:

$$C = -\frac{1}{2}\sum_{k=1}^{n}\sum_{i=1}^{q}\left[\hat{y}_i^{(L)}(k) - y_i(k)\right]^2$$

After having propagated the activation function and reached the output layer, the algorithm will work backward in order to optimise the cost function in terms of the weights for each artificial neuron. However, it is worth mentioning that this algorithm is characterised by two main drawbacks. In fact, through the gradient-based optimisation, the cost function could have many local maxima. As a result, the algorithm could finally reach one of this local maximum without having reached the global one. The second disadvantage is due to the fact that the final results could vary if we start with different initial values for the weights.

2.7.2. Recurrent Neural Network

Some parallelisms have been made between the functioning of the human brain and the principle behind the ANN. However, unlike traditional neural networks, the human brain has the capability to understand elements given previous events. Recurrent Neural Networks address this issue thanks to loops that allow the information to persist in the network. This kind of neural network can be thought of as multiple identical networks, each passing a message to the successor.

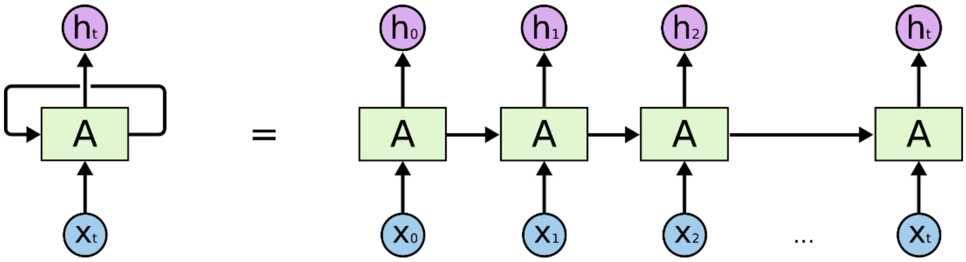


Figure 4 - Unrolled Recurrent Neural Network

In *Figure 4*, A is a module of neural networks, x(t) is the input while h(t) is the output. Recurrent neural networks turn out to be useful in a variety of problems such as speech recognition, language modelling, translation, etc. Indeed, in language modelling, it might be able to predict the next word in a sentence based on previous ones. For instance, in a short sentence as "Clouds are in the *sky*", the last word might be easy to predict for the model. However, in some cases, it can be much harder when some context is needed (Bengio, et al., 1994).

In the sentence, "I grew up in Spain… I speak fluent *Spanish*" the model knows that the last word to predict is supposed to be a language. Nevertheless, it is difficult for the network to predict the language "Spanish" if there is a large gap between the word to predict and in this case the word "Spain" which helps to suggest which language the person speaks. Luckily, Long Short Term Memory (LSTM) neural networks are able to learn long-term dependencies.

2.7.3. Long Short Term Memory neural network

Firstly, introduced by Hochreiter & Schmidhuber (1997), LSTM is designed to avoid long-term dependency problems. While LSTM has the basic structure of an RNN, it does contain four neural network layers interacting between each other where the standard RNN has only one layer.

The most important thing in the LSTM is the cell state C_{t-1}. The model has the capability to add or remove information to the cell state based on new information the model has at its disposal. Denoted by a yellow box[4], the sigmoid layer called the forget gate layer f_t looks at the information in h_{t-1} and in x_t and outputs a value between 0 and 1 since the Y values of a sigmoid function range between 0 and 1. In this case, the higher the output value, the more past information is kept in the cell state. The formula of the forget gate is the following where σ is the sigmoid function, W_f the weights for the forget gate neurons and b_f is the bias of the gate.

$$f_t = \sigma\left(W_f \ * \ [h_{t-1}, x_t] + b_f\right)$$

Once the information one wants to keep is stored in the cell state, it needs to decide what new information will be recorded in the cell state. Firstly, another sigmoid function called the input gate layer i_t whose values will be updated while a tanh layer creates a vector of new values $\tilde{C}_t$ to be added to the cell state. See the corresponding formulas below:

$$i_t = \sigma(W_i \ * \ [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = tanh(W_C \ * \ [h_{t-1}, x_t] + b_C)$$

To update the cell state from C_{t-1} to C_t, one simply multiplies the old state by the forget gate in order to delete the things it has been decided to forget. Then, this state is added to the new values multiplied by the weight one wants to dedicate to each new value.

$$C_t = f_t \ * \ C_{t-1} + i_t \ * \ \widetilde{C}_t$$

[4] See graphs on appendix B.

Finally, the output is based on the current cell state C_t. First of all, a sigmoid layer decides which parts of the cell state will be output. Afterwards, a tanh layer is multiplied by the output provided by the sigmoid function previously uses in order to return values between -1 and 1.

$$o_t = \sigma\,(W_o * \,[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t \, * \, tanh(C_t)$$

Once the model has output the values provided by the sigmoid function, it remains to check the accuracy of the model. To this end, one used three widely known performance measures, the MAE, RMSE and MSE.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \qquad RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2} \qquad MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

3. Methodology

In this section, we discuss the methodology followed throughout this thesis. As suggested by Jacquemin (2017), one must justify whether the methodology is qualitative or quantitative. It is also worth stating the goal of the work as well as the related hypotheses. Over a second phase, the design of the research is detailed and broken down into three sections: the sample composition and size, the data collection method, and the data analysis method.

3.1. Choice of Methodology

The methodology of this master thesis is quantitative and follows a counterfactual approach. As a matter of fact, this thesis treats concepts related to the portfolio management and the sentiment analysis. Therefore, one must deal with a bunch of financial data such as stock prices, volumes as well as non-financial data, that is to say, messages used for the sentiment analysis.

The goal is to answer the following question: "What is the impact of integrating investors sentiment into a Robo Advisor advanced financial model?". The advanced financial model refers to the Black-Litterman model as it is known to overcome some problems that lie in the most widely used financial model for RA, the Modern Portfolio Theory. Various sentiment analyses have already been done on well-known indices or different sets of famous companies. However, this thesis aims for covering a broader number of well-known and less famous companies that make up the ETFs analysed and also proposes a framework to build a Robo Advisor. Hereunder, one finds the two research hypotheses.

1st research hypothesis:
"Omitting the integration of investors sentiment into the Robo Advisor advanced financial model provides a higher risk-adjusted return than the Modern Portfolio Theory."

2nd research hypothesis:
"The integration of investors sentiment into the Robo Advisor advanced financial model improves the risk-adjusted return compared to the Modern Portfolio Theory."

3.2. Research Design

This part will cover the three elements needed to design the research: the sample composition and size, the data collection method and the data analysis method (Jacquemin, 2017).

3.2.1. Sample composition and size

The sample composition has been determined by the method Betterment has to rate funds. Betterment is an online financial advisor proposing investment and automated portfolio management. To score funds, Betterment (2018) uses the Total Annual Cost of Ownership (*Figure 6*). This scoring method takes ETFs transactional, liquidity and holding costs into account. It can be seen as the sum of the Cost-to-Trade and the Cost-to-Hold.

$$\underbrace{\textbf{Liquidity (Volume) + Bid-Ask Spread}}_{\text{Cost-to-Trade}} + \underbrace{\textbf{Expense Ratio + Tracking Error}}_{\text{Cost-to-Hold}} \textbf{ = TACO}$$

Figure 5 - Total Annual Cost of Ownership

On the one side, the volume helps to measure whether it will be difficult to find a buyer or a seller in the future. The availability of counterparts is important because it will allow the RA to trade without impacting market prices. Betterment measures this liquidity by gauging the average volume for each ETF as a percentage of Betterment's normal trading activity. Moreover, the bid-ask spread makes up the second part of the Cost-to-Trade. The way an investor care about the bid-ask spread may depend on his investment style. While buy-and-hold investors do not care a lot about the spread, active traders make numerous transactions and are therefore highly concerned about that. Minimising this cost is beneficial to building an efficient portfolio (Betterment, 2018). It is worth mentioning that there is no commission in the transactions operated by Betterment. However, this parameter should be added to other wealth management firms.

On the other side, the expense ratio represents the percentage of the share price paid by the shareholders to the fund every year. Consequently, the higher the fee, the lower the return net-of-fees that the investor gets. Finally, the tracking error represents the performance of the ETF compared to the benchmark index. It can be caused by the difference of weights and trades regarding the fund's holding. To tackle this issue, ETF issuers can enhance their operational systems, but this procedure is costly and may impact the management fee charged to the investor. To sum up, expenses and tracking error are inversely correlated. It is important to find the right balance between these two parameters.

In their computations, Betterment also takes the tax loss harvesting principle into account. For this reason, they always provide an alternative to the primary tickers called the alternate tickers. However, this aspect will not be considered as it is not directly related to the purpose of this work.

In the end, the sample of this thesis is made up of twelves ETFs divided equitably between stocks and bonds from January 1st, 2018 to December 31st, 2018. *Table 1* summarises the ETFs, its asset classes as well as indices they track. More precise information about the composition of each ETF are explained in section 4.1.

Asset Class	**ETF**	**Index**
US Total Stock Market	VTI	CRSP US Total Market TR USD
US Large-Cap Value Stocks	VTV	CRSP US Large Cap Value TR USD
US Mid-Cap Value Stocks	VOE	CRSP US Mid Cap Value TR USD
US Small-Cap Value Stocks	VBR	CRSP US Small Cap Value TR USD
International Developed Stocks	VEA	FTSE Dvlp ex US All Cap (US RIC) NR USD
Emerging Market Stocks	VWO	FTSE EMs AC China A Incl (US RIC) NR USD
Short-Term Treasuries	SHV	ICE U.S. Treasury Short Bond TR USD
Inflation Protected Bonds	VTIP	BBgBarc US TIPS 0-5 Year TR USD
US Municipal Bonds	MUB	S&P National AMT Free Muni TR USD
US High Quality Bonds	AGG	BBgBarc US Agg Bond TR USD
International Developed Bonds	BNDX	BBgBarc Gbl Agg x USD Fl Aj RIC TR HUSD
Emerging Market Bonds	EMB	JPM EMBI Global Core TR USD

Table 1 - Primary tickers proposed by Betterment

3.2.2. Data collection method

From Lambin & de Moerloose (2016), we can say that the data collection methods are a published extern secondary data collection and a primary data collection.

On the one side, it has been chosen to capture the daily prices, volume and market capitalisation data of twelve ETFs from 1st January 2018 to 31st December 2018 from Thomson Reuters. However, one has to keep in mind that daily pricing for weekends and other holidays are missing when the market is closed. Therefore, it is wise to approximate the missing values by a concave function since the stock data usually follows such function (Mittal & Goel, 2012). The risk-free rate used along this thesis is the 10-year Treasury Note of the US government on 31st December 2018 and is taken from the U.S. Department of the Treasury. As the risk-free rate is annualised, it is converted from an annualised rate to a daily risk-free rate.

To get market-related data, it has been decided to retrieve messages from Twitter. Since July 2012, Twitter allows users to filter financial information related to listed companies thanks to the cashtag ($) instead of the hashtag (#) followed by the stock name. Consequently, this terminology will be used to retrieve our tweets in order to avoid the collection of irrelevant messages.

While the Twitter's official API (Application Programming Interface) gives access to historical tweets, the retrieval period cannot exceed one week. Therefore, it was necessary to find an alternative to overcome this constraint. The Get Old Tweets Python script is an open-source project[5] which allows bypassing the time constraint set by the Twitter API. Through calls to a JSON provider, the script schematically scrolls down a Twitter web page so as to get the desired tweets for the desired time frame.

To be able to use this Python script, it has been necessary to install an old version of Python (Python 2.7) as it appeared not to work properly with the lastest version (Python 3.7). After having installed PIP, a tool that is able to manage and install Python packages, I had to install some other packages thanks to the Windows Command Line (lxml and pyquery). As it can be seen in the file path of *Figure 5*, the Get Old Tweets (GOT) script had to be stored into the root of the Python27 folder as the file "Exporter.py" has to be read by the Python program. Finally, through the Command Line, researches are performed on a specific term one wants to look for[6]. In the example (*Figure 5*), the script retrieves all tweets from 1st January 2018 to 31st December 2018 containing the string "$AAPL". The output is a Comma Separated Values (.csv) file named "Apple.csv" filled with ten columns[7] providing general information on each tweet. However, for the sake of this thesis, only the date and the content of the tweet have been kept.

[5] https://github.com/Jefferson-Henrique/GetOldTweets-python
[6] This process is time consuming and subject to failures since it is largely experimental. Therefore, the script has been run on two computers at a time.
[7] ID, permalink, username, text, date, retweets, favorites, mentions, hashtags and geo

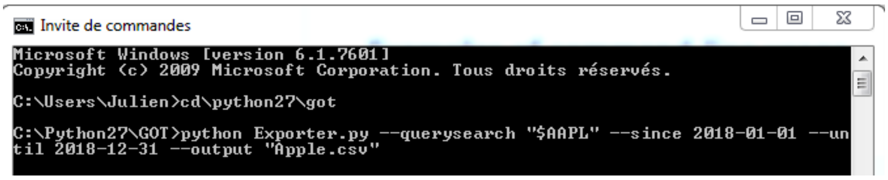


Figure 6 - Command line to retrieve tweets

The choice of keywords is made with an assumption. In the model, one focusses on twelve ETFs that are supposed to represent the global market and therefore gives a wide range of possibilities to invest. While some ETFs might be largely mentioned on Twitter, analysing the underlying assets of each ETF might be more interesting in order to increase the number of tweets. To this end, tweets of the largest companies[8] making up the ETFs have been chosen. However, it is worth mentioning that it is not possible to retrieve tweets for all ETFs. While half of ETFs are made up of listed companies, the second half only contains bonds. Unfortunately, it is not possible to look for this kind of financial instrument on Twitter as it is the case for stocks.

To support this idea, it has been found that the Reuters Social Media Monitor neither manages to predict the sentiment of bond ETFs. Therefore, these ETFs will be removed from our sentiment analysis. Instead, investor's views regarding the return of bond ETFs will be based on their past prices and volumes.

To make sure that the underlying assets chosen are representative of the index, we intend to verify the correlation between the ETFs daily return and the one of its underlying assets. Measuring this correlation is essential since it determines whether this small number of companies is able to correctly mimic the index. If it fails to do so, bad results can be expected while analysing the causality between investors sentiment and the return of ETF's underlying assets. To verify the correlation, a vector of weighted returns wr_i is built as follows:

$$wr_i = \begin{pmatrix} r_{11} & \cdots & r_{1j} \\ \vdots & \ddots & \vdots \\ r_{i1} & \cdots & r_{ij} \end{pmatrix} * \begin{pmatrix} w_1 \\ \vdots \\ w_j \end{pmatrix} = \begin{pmatrix} wr_1 \\ \vdots \\ wr_i \end{pmatrix}$$

Where:

r_{ij} = stock price return of company j on day i

w_j = average weight of company j in 2018

[8] See all company names, RIC (Reuters Instrument Code), cashtag and average weight for 2018 on appendix C.

Results in *Table 2* show that the first four ETFs have a strong correlation with the chosen underlying assets, meaning that this number of companies is sufficient to represent the ETF. However, the VEA and VWO show a lower correlation suggesting that the number of companies to represent the ETF is not sufficient. The cause of this rather lower correlation compared to other ETFs consist in the removal of some elements making up the ETF. Indeed, in the VEA, the USD cash has an average weight of 2.72%. However, this instrument cannot be analysed with tweets as no cashtag is provided for this instrument. Hence, it should be removed from our analysis. The same phenomenon is mentioned in the VWO where the Vanguard Market Liquidity Fund accounts for, on average, 2.05%. Nonetheless, this instrument can neither be analysed with tweets since it is a cash management vehicle for the Vanguard funds (Vanguard, 2019) and therefore should be deleted for the purpose of the analysis.

Moreover, for some companies in the VOE (KeyCorp, Motorola Solutions Inc.), VBR (PerkinElmer Inc.) and VWO (Taiwan Semiconductor Manufacturing Co Ltd.), the Python script faced some issues while retrieving tweets. As a matter of fact, it turns out that it was unable to extract any tweets in 2018 while some were effectively posted on Twitter. Therefore, they have been deleted from the analysis as well.

ETF	**Correlation**	**Number of companies**
VTI	96.55%	11
VTV	98.74%	24
VOE	92.42%	12
VBR	90.38%	9
VEA	43.51%	10
VWO	61.23%	10

Table 2 - Correlation between the ETF and its underlying assets

After several trials, it has been chosen not to increase the number of companies for the VEA and VWO as the rise in correlation was not significant compared to largest amount of data that should have been collected.

While some companies such as PsychSignal and Reuters already provide sentiment data for many companies, these data are expensive to acquire and do not necessarily analyse all the companies we have included in this thesis. Therefore, it was essential to find a technique to retrieve sentiments on our own.

3.2.3. Data analysis method

The data analysis method is a comparative method between the Modern Portfolio Theory and the Black-Litterman model. In fact, as stated in our research hypotheses, we want to compare the risk-return trade-off of the MPT with BL one without views. Also, we make comparisons between the complete BL model (i.e. with investor's views) and the MPT.

In the first part, we aim at building the Markowitz's Mean Variance Optimisation framework. We draw the efficient frontier based on the ETF's mean expected returns. On this frontier, we plot the minimum variance portfolio, the tangency portfolio and an efficient portfolio if an investor wish to get a daily return of 0.01%.

In the second part, we construct the Implied Equilibrium Excess return vector of the Black-Litterman model. At this stage, the BL model does not include any investor's view. As for the MPT, we draw the efficient frontier and plot the minimum variance portfolio and same efficient portfolio as proposed before. By drawing both efficient frontier on the same graph, one can approve or disprove the first research hypothesis.

In the third part, we need to construct investor's views with the help of our tweets retrieved thanks to the Python script. As a first step of the sentiment analysis, we would like to clean our tweets in order to only keep the most relevant tweets. Indeed, amongst the tweets collected many of them can be considered as noise since they do not provide the investor's sentiment. Over a second phase, we choose a subjective dictionary that is suitable for social media data. Then, we use the sentiment() function from the 'sentimentr' package to obtain a daily sentiment for each company. Next, we want to test whether there exists any nonlinear dependency between the ETF return and sentiment time series with lags ranging from one to four days. As each ETF is made up of numerous companies, a home-made algorithm and the Shannon transfer entropy test, decide which combination of companies best exhibits a nonlinear dependency for each lag. Lastly, we use the Long Short Term Memory neural network and we keep, for each ETF, the company combination that has the smallest MAE, RMSE and MSE. This neural network provides the expected returns of investor's views.

In the last part, we draw the efficient frontier of the BL model including the investor's views as well as the dots previously plotted on the other efficient frontiers and we compare the results with the Markowitz's MVO. From that point, we can approve or disconfirm the second research hypothesis.

4. Empirical Analysis

In this section, we firstly describe the twelve ETFs selected and perform an exploratory data analysis. Then, we compute the Markowitz Mean Variance Optimisation framework followed by the calculation of the Black-Litterman model. Lastly, we compare both models and impose the limits of this thesis.

4.1. Data Description

As explained previously, the ETFs have been chosen following the method proposed by Betterment. This portfolio of twelve-asset classes aims at proposing the most diversified portfolio as possible.

On the one side, the investor's portfolio is made up of six stock ETFs. The VTI tracks the CRSP US Total Market which nearly represents 100% of the U.S. investable equity market with roughly 4.000 constituents (CRSP, NA). Moreover, the VTV, VOE and VBR tracks the CRSP US Large, Medium and Small Cap Value respectively. The CRSP US Large Cap Value includes the top 85% of U.S. investable market capitalisation while the Medium and Small Cap value comprises the top 70-85% and the bottom 2-15% of the U.S. investable market capitalisation, accordingly (CRSP, NA). Besides, the VEA tracks the FTSE Developed All ex US Index which intends for representing the performance of large, medium and small capitalisations in developed economies excluding the United States of America. This index is designed to capture 98% of the world's investable market capitalisation (FTSE, 2019). Finally, the VWO follows the FTSE Emerging Markets All Cap China A inclusion index. It is dedicated to show the performance of large, medium and small capitalisation stocks in emerging markets (FTSE, 2019).

On the other side, the investor is counterbalanced by less risky investments since the portfolio is provided with six bond ETFs. The SHV follows the U.S. Treasury Short Bond Index, it has been created in order to assess the U.S. Treasury market and is designed to include fixed rate securities, denominated in U.S. dollars with a term maturity ranging from one month to one year (ICE, 2019). In addition to that, the VTIP tracks the Barclays Capital U.S. Treasury Inflation Protected Securities Index. It includes all the investment grade US TIPS that still have at least one year remaining to maturity (ETFdb, 2019). Besides, S&P National AMT-Free

Municipal Bond Index, tracked by the MUB, is designed to measure the performance of the investment grade tax-exempt U.S. municipal bond market (S&P, NA). Furthermore, the AGG which follows the Barclays Capital U.S. Aggregate Bond Index, covers a wide range of public, investment grade, taxable, fixed income securities in the USA. It includes but is not limited to government, corporate and international dollar denominated bonds with maturities greater than one year (ETFdb, 2019). Regarding international developed bonds, BNDX aims at tracking the Barclays Global Aggregate ex-USD Float-Adjusted Index. It has been created to measure the performance of global non-U.S. dollar-denominated government, government agency, corporate and securitised investment grade fixed income investments (ETFdb, 2019). In the end, EMB focusses on the emerging bond market by chasing the J.P. Morgan Emerging Markets Bond Index Global Core. It allows tracking liquid, fixed and floating-rate debt instruments issued by sovereign entities in U.S. dollars (J.P. Morgan, 2017).

Table 3 reports the descriptive statistics of daily excess returns for the year 2018 for each stock ETF. Firstly, it can be seen from this table that all annualised mean returns are negative. VEA has the lowest value with a negative return of -19.95% while VTI has the highest with -9.23%. The lowest standard deviation can be observed in international developed stocks (VEA) whereas the emerging market stocks (VWO) is more than three times as high as the latter. The skewness values indicate that the distribution of all ETFs has an asymmetric tail towards positive values except for VWO. From the kurtosis values, one can mention that VTV has the fattest tails while VWO has the thinnest. Moreover, the hypothesis of normal distributed returns is rejected for all instruments at the threshold of 1%.

	VTI	**VTV**	**VOE**	**VBR**	**VEA**	**VWO**
Annualised mean return (%)	-9.23	-9.89	-16.62	-15.92	-19.95	-19.58
SD	20.97	16.43	15.05	17.25	12.58	39.71
Median	0.000322	0.000375	0.000119	-0.000132	-0.0000543	-0.000368
Min	-0.036	-0.0366	-0.0334	-0.0362	-0.0276	-0.0334
Max	0.024	0.0206	0.0204	0.0218	0.0197	0.0359
Skewness	-0.794	-0.976	-0.828	-0.732	-0.581	0.223
Kurtosis	2.977	3.19	2.704	2.723	1.467	0.87
Jarque Bera stat	175.945***	215.87***	155.43***	147.83***	54.351***	15.079***

Table 3 - Descriptive statistics of excess returns – Stocks

Figure 7 exhibits the cumulative return of stock ETFs. As from March 2018, the graph shows that ETFs focused on the U.S. market follow the same trend while ETFs making up the international developed stocks excluding the U.S. and the emerging market stocks follow the same trend as well. However, one can observe that VWO gives a larger negative cumulative return as from March 2018 than VEA even if the annualised mean return is lower for VEA (-19.95%) than VWO (-19.58%).

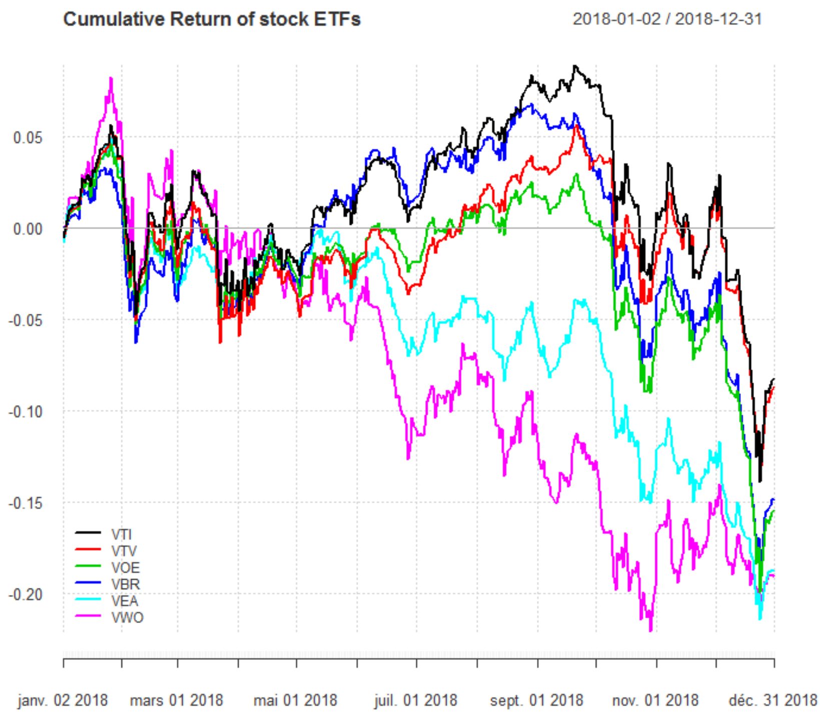


Figure 7 - Cumulative return for stock ETFs in 2018

Table 4 also reports descriptive statistics of daily excess return for the same year but for bonds instead of stock ETFs. As expected, the annualised mean return is lower for bond ETFs ranging from -2.38% to -12.84%. The latter value may seem quite high compared to others. This is because EMB corresponds to emerging market bonds. Unlike stocks ETFs, investors can enjoy a lower volatility (i.e. standard deviation) in the bond market. Quite trivially, the lowest volatility can be found in Inflation Protected Bonds (VTIP). Moreover, in general, one can mention that the kurtosis is far higher for bonds than stock ETFs. Finally, as for stocks, the hypothesis normal returns can be rejected since the p-value of the Jarque Bera test is lower than 1%.

	SHV	VTIP	MUB	AGG	BNDX	EMB
Annualised mean return (%)	-2.38	-4.19	-3.78	-4.80	-2.51	-12.84
SD	0.00025	0.00076	0.00139	0.0014	0.00112	0.00305
Median	-0.000045	-0.000068	-0.0000848	-0.0000676	-0.0000676	-0.000414
Min	-0.00178	-0.0067	-0.0077	-0.0054	-0.0087	-0.0114
Max	0.00084	0.0022	0.0108	0.0036	0.0036	0.0126
Skewness	-3.316	-2.461	0.527	-0.291	-1.725	0.174
Kurtosis	17.401	18.409	13.664	0.687	11.555	2.817
Jarque Bera stat	5324.8***	5576.8***	2887.1***	12.799***	2235.5***	124.98***

Table 4 - Descriptive statistics of excess returns – Bonds

Figure 8 showcases the cumulative return of bond ETFs. Interestingly, fixed income securities bounded to the U.S. market does not seem to follow the same trend as we have observed for the stock market. Furthermore, the BNDX and the EMB evolve in opposite directions, the international developed bonds return a positive cumulative return from time to time while the emerging bond market gives the largest negative cumulative return among these six ETFs.

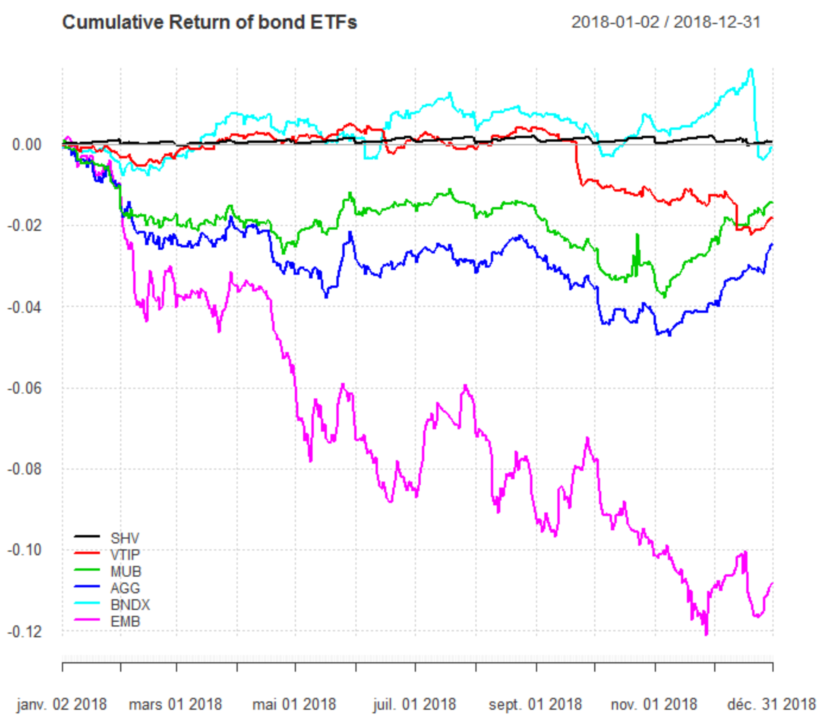


Figure 8 - Cumulative return for bond ETFs in 2018

Looking at the pairwise correlation between ETFs (*Figure 9*), one mentions that there is a strong correlation between stock ETFs while the correlation is weaker between bonds ETFs. A negative correlation can even be observed between stock and bonds ETFs. It suggests that it could be worth investing in both types of assets in order to diversify the investor's portfolio. Indeed, both asset types move in opposite directions. Therefore, an investor can hedge against a possible bull market thanks to bonds which are known to be safer than stocks.

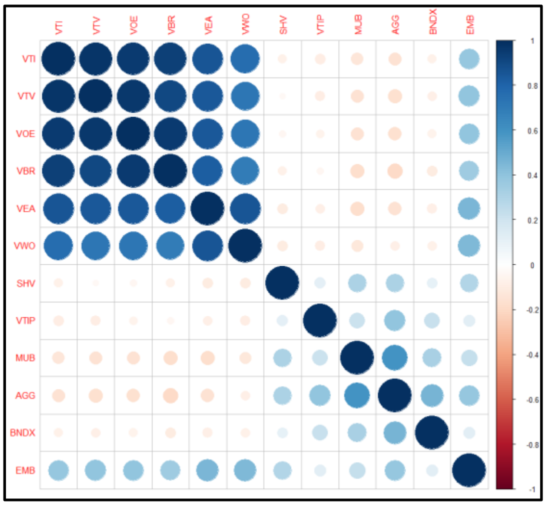


Figure 9 - Correlation between ETFs

4.2. Mean Variance Optimisation

The first step of the analysis is to build the Mean-Variance Optimisation (MVO) portfolio. To do so, it has been decided to design from scratch the various functions that are needed to get the MVO portfolio. Thanks to the creation of self-made functions, it is possible to enable or disable the short sell constraint depending on the aversion of the investor toward the risk. Moreover, it is also allowed to set a minimum and maximum allocation on assets. Nonetheless, the function solve.QP() from the quadprog package in R is used in order to find the optimal solutions of this quadratic problem given the aforementioned constraints. The sole use of this function has been chosen since it allows a better customisation of the needs unlike other packages that allows to directly find the optimal weights given some input parameters. R functions that have been used to build the minimum variance portfolio, the efficient portfolio and the efficient frontier are briefly described hereunder[9].

[9] Further information on the R code of these functions as well as on other manipulations realised for this master thesis can be found on a dedicated GitHub: https://github.com/judeblander/Master_Thesis

Minimum variance portfolio:

Min.var.portfolio(covmat, mean.returns, Rfdaily, short = "no", max.allocation, min.allocation, names)

Efficient portfolio:

Eff.portfolio(covmat, mean.returns, Rfdaily, short = "no", target return, max.allocation, min.allocation, names)

Efficient frontier:

Eff.frontier(covmat, mean.returns, Rfdaily, short = "no", max.allocation, min.allocation, risk.premium.up = 5, risk.increment = 0.005, names)

One mentions that each function presented above requires the variance-covariance matrix (*covmat*), the vector of mean returns (*mean.returns*), the daily risk free rate (*Rfdaily*), a minimum (*min.allocation*) and a maximum allocation (*max.allocation*) for all assets as well as their names (*names*). However, the efficient portfolio function takes as an argument, the target return which is the return an investor expects to get in the future. Finally, *risk.premium.up* and *risk.increment* are parameters that are necessary to build the efficient frontier. The division of *risk.premium.up by risk.increment* makes up the number of portfolios generated to build the efficient frontier.

By looking at the annualised mean return of 2018, one can be scared about the large negative return investors faced during that period. Therefore, an investor who wants to invest in 2019 maybe would like to short sell if he expects that the return of these ETFs will continue to worsen. Recall that being short or long depends on the risk aversion of the investor. In a short selling, investors expect that the price of a stock will go down in order to make profit whereas it is the reverse for long selling.

In order to gauge the market and see the expected excess return an investor can get for the lowest level of risk, it can be interesting to check the weights of the minimum variance portfolio. By allowing short selling up to -10% and setting the maximum allocation to 50% of the total wealth of the investor, one gets the weights[10] of *Table 5*.

[10] For the sake of this paper, it is not considered that the investor can borrow or lend at the risk-free rate meaning that the sum of the weights equals to 1.

ETF	Weights
VTI	-0.00083
VTV	0.01431
VOE	-0.02089
VBR	0.00706
VEA	0.01202
VWO	0.0033
SHV	0.5
VTIP	0.34693
MUB	0.06663
AGG	-0.04597
BNDX	0.13857
EMB	-0.02112

Table 5 - Weights of the minimum variance portfolio

With these weights, the investor can only expect a negative return of -0.00874% per day with a standard deviation of 0.00037. This means that he is obliged to take on more risk if he wants to keep, at least, his current wealth. It is worth mentioning that the model has reached the weight limit for the SHV as it exhibits the highest annualised mean return. In this portfolio, the weights are largely skewed towards bonds with 98,50% whereas only 1.50% is invested in stocks.

If an investor does not want to lose his wealth in 2019, one has to build a portfolio constrained by a positive expected return. In the present case, it has been decided to set a daily expected return of 0.01% meaning that the investor could expect to get an annualised return of 3.72% for 2019. With the same constraints on minimum and maximum allocation, one obtains the weights depicted in *Table 6*.

ETF	Weights
VTI	0.3111
VTV	-0.1
VOE	-0.1
VBR	-0.1
VEA	-0.1
VWO	-0.1
SHV	0.5
VTIP	0.1013
MUB	0.2875
AGG	-0.1
BNDX	0.5
EMB	-0.1

Table 6 - Weights of an efficient portfolio

From *Table 6*, one sees that the short sale is applied on all stock ETFs except for the VTI as the latter provides the highest annualised expected return (-9.23%). However, only two out of six bond ETFs have been short sell (AGG and EMB) due to their lowest expected return. A weight of 0.5 has been set on the SHV and BNDX since it delivers the highest return among all assets. For this expected return, the investor will suffer from a standard deviation which is more than four times greater than the minimum variance portfolio one (0.0019). Nevertheless, the Sharpe ratio has largely improved from -0.43 to 0.014 suggesting that this portfolio has a better risk-adjusted return. Moreover, 118% of the portfolio has been invested in bonds and -18% in stocks. Recall that the MVO assumes that the returns are normally distributed; however the Jarque Bera test rejected this hypothesis. This implies that an investor cannot entirely rely on these weights to make an investment decision.

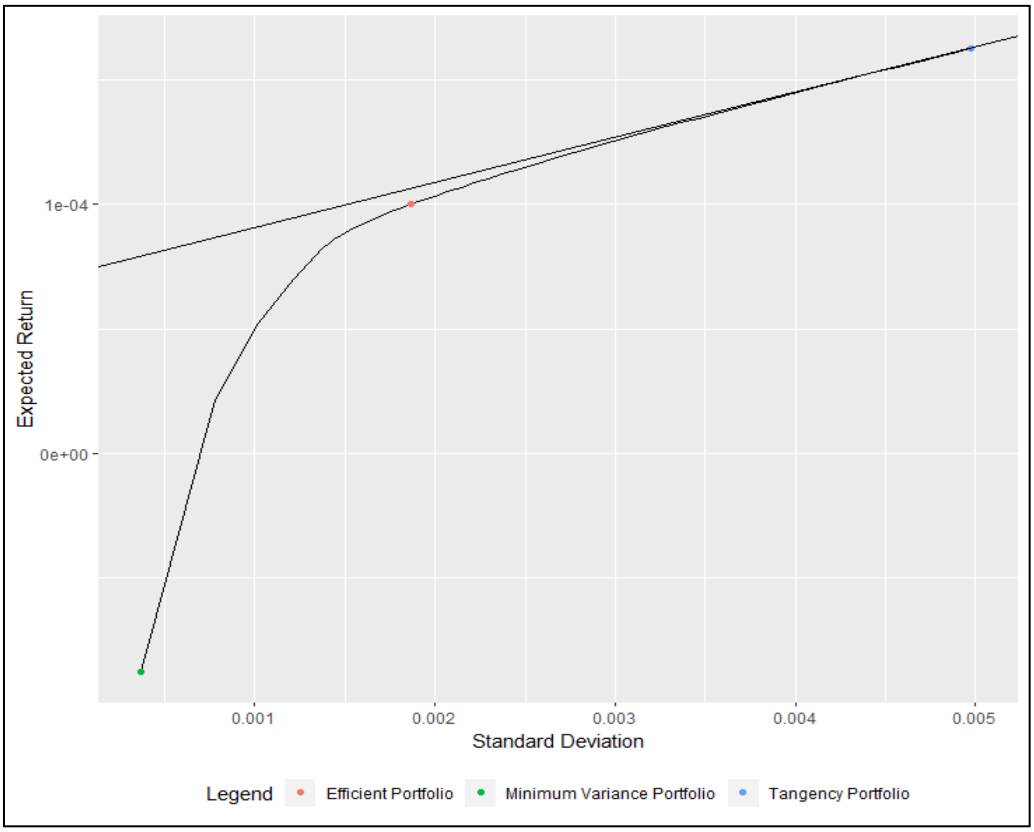


Figure 10 - Efficient frontier of the Mean Variance Optimisation

In this section, we have observed that an investor who is risk averse is unlikely to get a positive return in 2019 based on the mean returns of 2018. To get a daily return of 0.01%, he is obliged to short sell a large part of his assets, engendering a large discrepancy between the investments in bonds (118%) and stocks (-18%).

4.3. Black-Litterman Model

In this section, the Black-Litterman model is broken down into three parts. Firstly, one builds the vector of Implied Equilibrium Excess return thanks to market data. Secondly, the investor's views are formed with the help of a sentiment analysis. Lastly, one compares the IEE returns vector with the expected returns from investor's views to check whether views add information on the model.

4.3.1. Implied Equilibrium Excess returns

To compute the Implied Excess Equilibrium returns, the risk-free rate is needed as for the MVO. Moreover, the formula requires two additional parameters, that is to say, the market capitalisation weights of the global market portfolio and the risk-aversion coefficient. ETF's market capitalisation weights for 2018 (*Table 7*) have been found by computing the average of daily market capitalisation of each ETF.

ETF	**Market capitalisation weights (%)**
VTI	26.15
VTV	9.40
VOE	2.07
VBR	3.07
VEA	15.21
VWO	19.81
SHV	2.05
VTIP	1.01
MUB	2.54
AGG	13.70
BNDX	2.03
EMB	2.96

Table 7 - Market capitalisation weights

Furthermore, the implied risk-aversion coefficient is estimated by dividing the expected excess return by the variance of the market portfolio. The chosen market portfolio is the S&P1200 as it provides efficient exposure to the global market equity representing roughly 70% of the global market capitalisation. It ends up with a risk-aversion coefficient of -9.284495. The IEE returns vector is shown in *Table 8*.

ETF	IEE
VTI	-0.000457
VTV	-0.000413
VOE	-0.000398
VBR	-0.000402
VEA	-0.000379
VWO	-0.000520
SHV	0.000001
VTIP	0.0000035
MUB	0.0000095
AGG	0.0000081
BNDX	0.0000039
EMB	-0.0000807

Table 8 - Implied Equilibrium Excess return

Figure 11 shows how the Implied Equilibrium Excess returns vector performs compared to the mean returns vector of the MVO. Graphically, one can already understand that our first research hypothesis is partially true. Firstly, the minimum variance portfolio gives a largely better return with the IEE than with the MVO mean return for the same level of risk where it is reverse for a return included between 0.005% and 0.01%. Indeed, there is a small difference in terms of risk between both models for our investor who would like to get a daily return of 0.01% as we have assumed in the previous section. It implies that the risk-adjusted return of the BL model without views is 0.01431 whereas the MVO has a risk-adjusted return of 0.01463. Nonetheless, beyond this level of return, the Black-Litterman model without views is able to provide a far better return compared to the other model. Indeed, the Sharpe ratio of the tangency portfolio goes from 0.018 for the MVO to 0.04 for the BL model. It is worth mentioning that the efficient frontier of the BL model goes beyond the MVO one. It is explained by the fact that the IEE is balanced with positive and negative returns while the MVO mean excess return vector only has negative values. All constraints remaining equal for both models, the BL model proposes other efficient portfolios that could allow the investor to get a higher return.

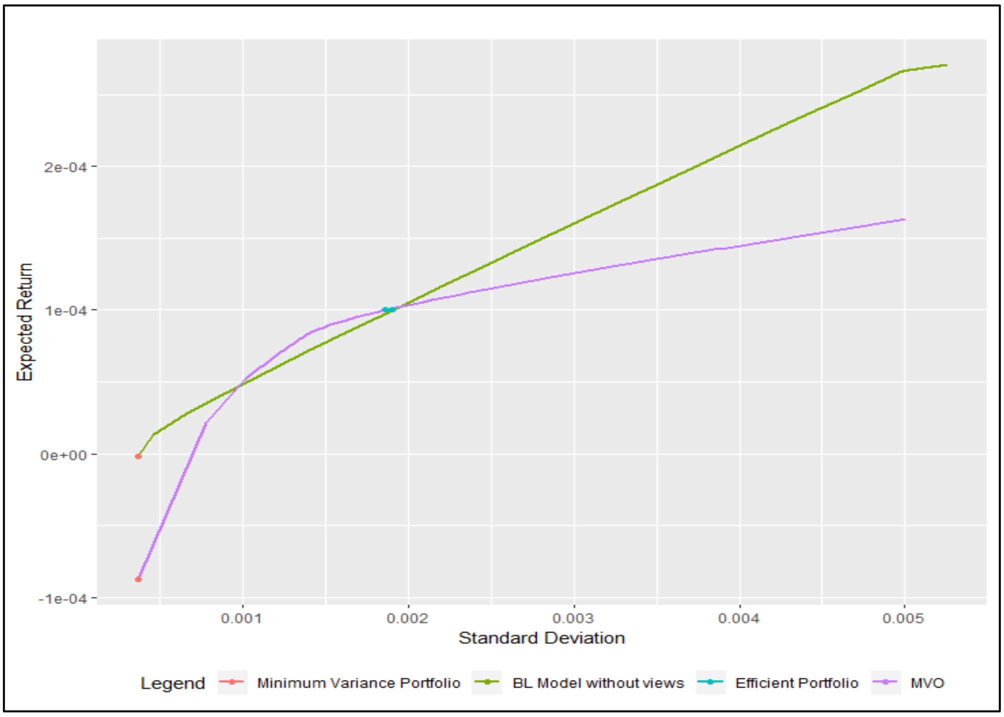


Figure 11 - Comparison between the MVO and BL model without views

4.3.2. Sentiment analysis for investors views

In order to get the investor's views needed for the second part of the Black-Litterman model, the sentiment of investors retrieved from tweets is added to predict future returns. The prediction of returns is realised thanks to a Long Short Term Memory neural network which turns out to be highly accurate in time series forecasting.

4.3.2.1.Cleaning data

The Python script that has been used to only retrieve a certain type of tweets allowed to narrow down the research about the investor's sentiment. Nonetheless, it appeared that among these messages, a vast majority of them does not clearly state an investor's sentiment. Instead, these tweets contain irrelevant information for us such as links, pictures, etc. Furthermore, some messages are considered as advertising and therefore they should be removed from the analysis. Firstly, it was decided to exclude tweets that do not contain words such as "I'm", "I believe", "I feel", "I don't", "I didn't", "makes me", "my guess" and "my sentiment". It allows keeping tweets that showcase the sentiment of someone regarding a particular stock. This is more relevant for the analysis and drastically decreases the number of tweets at the same time that in turn, reduce the processing time. Indeed, through the data collected 66 companies, an amount of 2.476.186 tweets has been gathered but only 28.533 remain after this cleaning process meaning that roughly 1% of the tweets collected are relevant.

Secondly, it can be expected that the remaining tweets still include characters that are not useful in a sentiment analysis. To this end, each message is filtered by removing usernames (@xxx), cashtags ($xxx) and URLs. Finally, every word is transformed into lower case in order to avoid a mismatch with words indexed in the subjective dictionary. Recall that the punctuation is not removed as the sentiment() function takes it into account while building clusters inside a sentence.

4.3.2.2.Choice of the subjective dictionary

Although many different dictionaries can be used in a sentiment analysis, it is essential to choose one that contains a specific vocabulary for the financial industry. In sentiment analysis, dictionaries for financial-related documents (Loughran & McDonald, 2011) and news (Huang, Zang & Zheng, 2013) are available but only few focus on social media data.

To this end, Chen, Huang and Chen (2018) have built a market sentiment dictionary (NTUSD-Fin) based on 330.000 labelled posts from the social media StockTwits which is known as a platform dedicated to share ideas between investors, traders and entrepreneurs. The NTUSD-Fin dictionary gathers a total of 8.331 words, 112 hashtags and 115 emojis with market sentiment values ranging from -3.112 to 1.275. Comparing *Table 9* and *Table 10,* one can mention that the number of words, hashtags and emojis has been drastically reduced while building the dictionary. This is due to the fact that tokens appearing fewer than ten times across all messages have been removed.

Hereunder, the distribution of the words shows that the dictionary provides more positive words (bullish) than negative words (bearish). As a matter of fact, 80% of words of the dictionary are considered positive while the remaining 20% are labelled as negative. It can be explained by the fact that the number of posts retrieved on StockTwits already shown this trend as one sees in *Table 10*. The main reason is that in the history the bull market is much longer and profitable than the bear market implying that people tend to bet in time of bullish market.

	Bullish	Bearish
Word	6.670	1.661
Hashtag	97	15
Emoji	103	12

Table 9 - Distribution of the NTUSD-Fin dictionary

	Bullish	Bearish
Post	289.416	45.382
User	12.452	5.834
Word	69.114	25.956
Hashtag	2.507	715
Emoji	427	174

Table 10 - Distribution of the original dataset

However, it has been proved that the NTUSD-Fin dictionary outperforms the Loughran and McDonald dictionary. It was tested on the SemEval-2017 Task 5 dataset (Cortis et al., 2017) which was made up of posts from Twitter and StockTwits. To remain consistent, all messages contain at least one cashtag, which is necessary to spot a financial instrument.

Some could argue that the number of words in the dictionary is not sufficient to get a sense of the meaning of a sentence. However, the Zipf's law states that given a large corpus of natural language occurrences, the frequency of any word is inversely proportional to its rank in a frequency table. *Figure 12* exhibits the Zipf's law regarding the words used in our sentiment analysis.

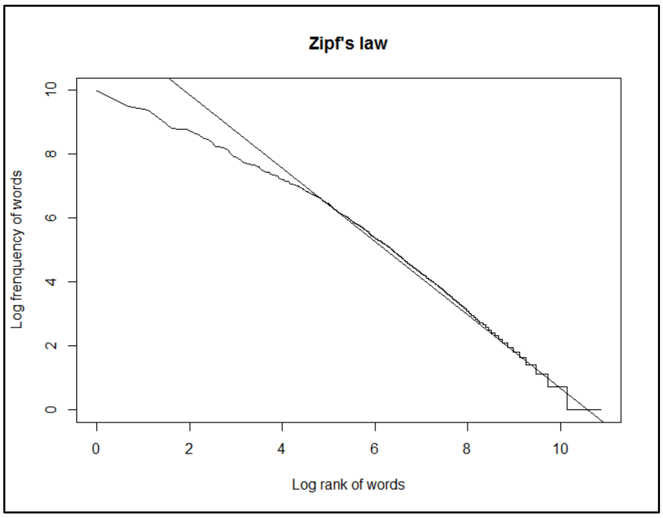


Figure 12 - Zipf's law of words from tweets

From a technical point of view, one can find the NTUSD-Fin dictionary on GitHub[11]. As the files have been updated in the JSON format, it has been necessary to convert it, thanks to the 'rjson' package in R, in order to only keep the data we were interested in, that is to say, tokens and market sentiments. Moreover, to make this data frame readable by the sentiment() function, we used the as_key function in the *polarity_dt* argument as explained previously.

[11] https://github.com/thtang/NLP2018SPRING/tree/master/project1/NTUSD-Fin

Regarding valence shifters dictionary, it has been decided to select the one proposed by the 'lexicon' package called hash_valence_shifters. The package regroups a large variety of hash tables, dictionaries and word list that are largely used in text analysis. The aforementioned data frame is composed of 140 words labelled either as negators, amplifiers, deamplifiers or adversative conjunctions.

4.3.2.3.Result analysis

As expected from the analysis of the dictionary, the logarithmic probability density function (*Figure 13*) shows that the distribution of sentiment is left-skewed (-0.985), the distribution is longer on the left-hand side. Mathematically, from *Table 11*, one sees that the logarithmic mean (-1.277) is less than the logarithmic median (-1.086). It means that a larger part of sentiment scores are positive in accordance with the large number of positive words present in the NTUSD dictionary. Indeed, 20107 tweets have been labelled as positive while 8086 messages are categorised as negative. In accordance with the skewness of the dictionary towards positive words, 71% of our tweets are labelled as positive. It is worth mentioning that these results have been obtained by letting all the parameters of the sentiment() function to default.

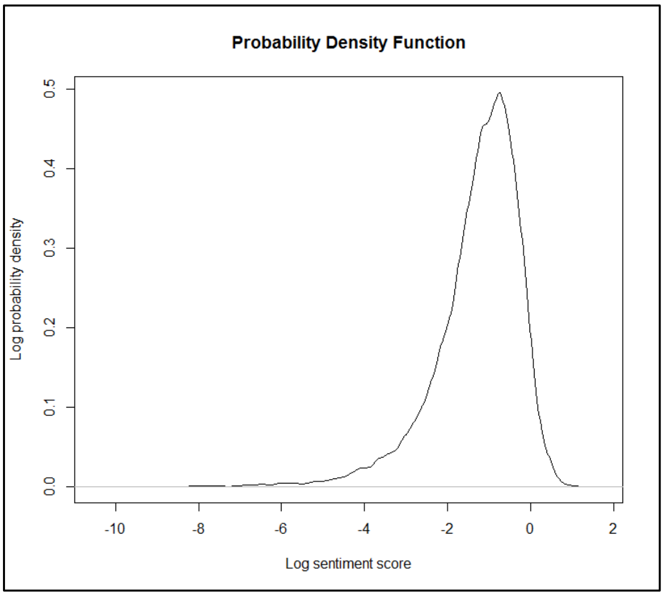


Figure 13 - Log probability Density Function of sentiments

	Log Value
Minimum	-10.172
Maximum	1.417
Mean	-1.277
Median	-1.086
SD	1.136
Skewness	-0.985
Kurtosis	1.206

Table 11 - Descriptive statistics of sentiments

In order to verify how the sentiment() function performed with the Fin-NTUSD dictionary, 130 tweets have been labelled manually either as positive (+1), negative (-1) or neutral (0). In this process, it has been considered that sentiment values that are encompassed between -0.1 and 0.1 are neutral.

The accuracy of the model is computed by summing the numbers on the diagonal of the confusion matrix (*Table 12*) and dividing them by the sample size, that is to say, 130. In the end, an accuracy of 67,69% is reached. One could better appreciate the accuracy of the model by extending the labelled sample size. However, this task is highly time consuming and can be subjected to some human errors.

		Predicted label		
		-1	**0**	**1**
True label	**-1**	10	0	7
	0	5	23	23
	1	4	3	55

Table 12 - Confusion matrix

Table 13 exhibits the precision, recall and F1-score for each class. Firstly, the precision is the number of correct predictions divided by the number of total predictions made (Lanaro, 2016). Generally speaking, when the level of precision is high, it means that the model is likely to be true while predict a class. In the table below, one sees that class 0 reaches the highest prediction with 88.46% suggesting that, most of the time, the model classifies messages correctly with no sentiment.

Secondly, the recall is the number of correct predictions divided by the total number of elements present in the class (Lanaro, 2016). The higher the recall, the better the model manages to recover instances of the class. In *Table 13*, the highest recall reaches 88.71% for class 1.

Lastly, the F1-score is the harmonic mean of the precision and recall. Again, one mentions that class 1 attains the highest F1-score meaning that this class is the one that is the least subject to errors. However, it is more difficult for the model to correctly predict elements of class -1 where the F1-score is 55.55%. This result is in accordance with what was suggested previously. As a matter of fact, the NTUSD-Fin dictionary possesses more positive words than negative words. It might be tricky to match a negative word from a tweet with the NTUSD-Fin dictionary ones since the number of those words are limited.

	Precision	Recall	F1-score
-1	0.5263	0.5882	0.5555
0	0.8846	0.4339	0.5822
1	0.6471	0.8871	0.7483

Table 13 - Evaluation measures

4.3.2.4.Transfer entropy test

In order to uncover which combination of companies for a given ETF shows the greatest causality between the sentiment and the ETF return, it has been decided to test all possibilities thanks to the creation of a home-made algorithm.

To put it into perspective, let us take the example of the VTI. In the analysis, it is composed of 11 companies meaning that 2048 possibilities (2^{11}) must be checked so as to find the one that best causes the ETF return. To do so, a *11 x 2048* matrix containing all possible combinations of binary values (0 and 1) is pasted next to the vector of average weight for 2018 for each company. Thanks to multiple "for" loops, binary variables act as a pointer to determine which companies one should consider in our causality analysis. Once these companies are selected, it sums the weights w_j to rescale the weights of each company in the subset. Once this is done, the algorithm computes the weighted sentiment of companies present in the subset. All weighted sentiments for each day are summed to end up with a global sentiment that is in line with the weight each company has in the ETF.

Then, a left join is executed between the VTI return and the sentiment for each day to make sure the return and the sentiment are matched the same day. While some NA values may appear due to the left join, it will not be considered for the causality analysis.

The analysis is executed 2048 times for lag periods ranging from 1 to 4 days and p-values are stored in a vector for each lag. However, an if condition is set previous to the transfer entropy test. As it has been assumed that the tweet's sentiment can be detected in the stock price up to 4 days after its post, a minimum of 91 days (one fourth of the 364 VTI returns) of sentiment is required to test the Shannon transfer entropy. Among each p-value vector, the lowest is retrieved and for completeness other p-values for this company combination is shown.

Table 14 exhibits company combinations[12] which demonstrate the lowest p-value among the set of all possible combinations. One can mention that at least one out of the four models has a significant p-value at a level of 5% (*) or below 0.1% (***). Unsurprisingly, widely covered ETF such as the VTI and VTV have many significant p-values for various lags while it appears

[12] See each significant company combinations on appendix D.

not to be the case for less known ETFs like the VWO where the lowest p-value for $\boldsymbol{M_{2,VWO}}$ and $\boldsymbol{M_{4,VWO}}$ are 0.18 and 0.34, respectively. Shannon transfer entropy test has also shown that the same company combination can exhibit the lowest p-value for different lags. This is the case for $\boldsymbol{M_{3,VOE}}$ and $\boldsymbol{M_{4,VOE}}$ as well as for $\boldsymbol{M_{1,VEA}}$ and $\boldsymbol{M_{2,VEA}}$.

Lag	$\boldsymbol{M_{1,VTI}}$	$\boldsymbol{M_{2,VTI}}$	$\boldsymbol{M_{3,VTI}}$	$\boldsymbol{M_{4,VTI}}$
1	0.01*	0***	0.05	0.17
2	0.01*	0***	0.02*	0.15
3	0.03*	0.19	0.02*	0.06
4	0.07	0.53	0.04*	0.01*
Lag	$\boldsymbol{M_{1,VTV}}$	$\boldsymbol{M_{2,VTV}}$	$\boldsymbol{M_{3,VTV}}$	$\boldsymbol{M_{4,VTV}}$
1	0.01*	0.17	0.30	0.09
2	0.01*	0***	0.03	0.26
3	0.09	0.06	0.02*	0.09
4	0.58	0.55	0.08	0.02*
Lag	$\boldsymbol{M_{1,VOE}}$	$\boldsymbol{M_{2,VOE}}$	$\boldsymbol{M_{3,VOE}}$	$\boldsymbol{M_{4,VOE}}$
1	0.03*	0.97	0.31	0.31
2	0.17	0.03*	0.46	0.46
3	0.19	0.37	0***	0***
4	0.20	0.53	0***	0***
Lag	$\boldsymbol{M_{1,VBR}}$	$\boldsymbol{M_{2,VBR}}$	$\boldsymbol{M_{3,VBR}}$	$\boldsymbol{M_{4,VBR}}$
1	0.15	0.24	0.15	0.30
2	0.15	0***	0***	0.15
3	0.71	0.07	0***	0.44
4	0.81	0.61	0.42	0.44
Lag	$\boldsymbol{M_{1,VEA}}$	$\boldsymbol{M_{2,VEA}}$	$\boldsymbol{M_{3,VEA}}$	$\boldsymbol{M_{4,VEA}}$
1	0***	0***	0***	0***
2	0***	0***	0***	0***
3	0.01*	0.01*	0***	0.01*
4	0.02*	0.02*	0.02*	0***
Lag	$\boldsymbol{M_{1,VWO}}$	$\boldsymbol{M_{2,VWO}}$	$\boldsymbol{M_{3,VWO}}$	$\boldsymbol{M_{4,VWO}}$
1	0***	0.33	0.01*	0.49
2	0.52	0.18	0.45	0.35
3	0.33	0.63	0.29	0.32
4	0.50	0.55	0.7	0.34

Table 14 - p-value for the Shannon entropy test

4.3.2.5.Predictions

To be able to predict the ETF return given the sentiment expressed on Twitter towards its underlying assets, the LSTM neural network is used since it is known to reach a high accuracy in time series forecasting.

In R, the commonly used packages for LSTM are Keras and the TensorFlow backend. Keras is a model-level library providing high-level building blocks for fast experimentation in deep learning. Indeed, it does not handle low-level operations such as tensor products, convolutions, etc. Instead, it relies on well-optimised tensor manipulation libraries to act as a back-end engine. In this case, the tensor manipulation framework TensorFlow has been used since it turns out to be straightforward to use in combination with Keras. It is worth mentioning that both packages need to be installed in Python in order to be used in R afterwards.

Machine learning systems use tensors as their data structure. In fact, a tensor can be interpreted as a container for data which is, most of the time, numerical. Matrices are 2D tensors with a sample axis and a feature axis. In our case, the data is stored in a 3D tensor since the time dimension must be considered.

Recall that the size of the dataset is 364 data points, in machine learning a rule of thumb is to use roughly 70% of the dataset as the training set and the remaining part as the test set. Therefore, 252 data points make up the training set and 84 constitutes the test set. It is also essential to prepare the data in a way that it can be fed into the LSTM. To this end, it is wise to scale the data in the range [0,1]. However, it can be expected to have missing values in the sentiment time series since minimum of 91 days of sentiment is expected for 2018. To overcome this obstacle, it is safe to input missing values to a value that is out of the range [0,1]. The network will learn from exposure to the data that this value means missing value and will ignore it (Chollet & Allaire, 2018). In such circumstances, missing values have been arbitrarily replaced by 2.

Our network consists of two LSTM stacked layers and the output of each LSTM layer is made through a tanh activation function. This means that the output value will be in the range [-1,1]. However, as one faces a regression problem, the final output of this neural network is generated through a dense layer whose activation function is linear. Consequently, the network is free to learn to predict values in any range. As recommended by Jozefowicz, Zaremba and Sutskever (2015), a bias of one is added to the forget gate at initialisation to reach better results.

It has been chosen to only build a model with two layers since the dataset is small, this means that we could face overfitting. In machine learning, overfitting is worse because the model reaches a high accuracy on the training set, but it fails to get such good results on the test set. Building a small network is therefore useful to mitigate this problem (Chollet & Allaire, 2018). Moreover, to further avoid overfitting, a dropout layer is added after each LSTM layer. Developed by Geoff Hinton, this regularisation technique consists of randomly dropping out (setting to zero) several output features. The dropout rate is the fraction of output features set to zero and usually range between 0.2 and 0.5. In our case, a dropout rate of 0.5 is applied.

Once the model is built, the learning process must be configured by stating the loss function, the optimizer and the metric one wants to use during the training. The loss function is an objective function that needs to be minimised during training and therefore represents a measure of success. The optimizer determines how the network is updated based on the loss function. While the basic optimiser is the Stochastic Gradient Descent, it was decided to use the Adam optimiser since it is widely mentioned in the literature and take advantages of other optimizers. It will not be detailed further as it is out of the scope of this thesis[13]. Regarding the loss function and the metric, Chollet and Allaire (2018) mentions that the Mean Square Error (MSE) and Mean Absolute Error (MAE) are respectively widely used in regression problems.

Before measuring the accuracy of the test set, it is common to validate the model on a part of the training set by a K-fold cross validation. This method consists in splitting the data into a K number of partitions, each of them having the same size. The model is trained on K-1 partitions and is tested on the remaining one. K-fold cross-validation is advised when the size of the dataset is small. As a matter of fact, if only a sample part of the dataset is chosen as validation set, the validation score might change a lot depending on which data points one chooses possibly leading to high variance. Cross-validation is also used to check whether there is still overfitting in the model when the data are processed too many times. For this analysis, it has been chosen to split the training set of 252 data points into 3 partitions of 84 data points. Therefore, the model is trained 3 times on 168 data points and tested on the 84 remaining points.

[13] The reader can refer to "Adam: A Method for Stochastic Optimization" (Kingma & Ba, 2014) for more details

In this context, the MAE is chosen to measure the error on the validation set. Since the data is processed 100 times in the model, we have 100 MAE for each fold. To get the optimal number of processes (i.e. to avoid overfitting), an average is made between each MAE of the 3 folds. This means that one ends up with an average of 100 MAE. The lowest MAE is retrieved, and its position in the vector gives the optimal number of processes.

$$Average\ MAE = \frac{\sum_{j=1}^{3} MAE_{i,j}}{3}\ \forall i :\ 1, \dots, 100$$

To determine which company combination best predicts the ETF return, the sentiment of each company combination is used as a feature in a LSTM neural network. Recall that each ETF has four different company combinations meaning that one combination among the four will be used to predict the future return. To ascertain which one works the best, MAE, RMSE and MSE are employed as performance measures. Hence, the model with the lowest error is kept. The prediction on models with no significant p-values on the Shannon transfer entropy test is not undertaken.

VTI	$\boldsymbol{M_{1,VTI}}$	$\boldsymbol{M_{2,VTI}}$	$\boldsymbol{M_{3,VTI}}$	$\boldsymbol{M_{4,VTI}}$
MAE	0.13228	**0.13206**	0.13282	0.13715
RMSE	**0.18800**	0.18923	0.18906	0.19296
MSE	**0.03534**	0.03581	0.03574	0.03723
VTV	$\boldsymbol{M_{1,VTV}}$	$\boldsymbol{M_{2,VTV}}$	$\boldsymbol{M_{3,VTV}}$	$\boldsymbol{M_{4,VTV}}$
MAE	**0.12151**	0.12277	0.12423	0.12350
RMSE	**0.17226**	0.17377	0.17380	0.17344
MSE	**0.02967**	0.03020	0.03021	0.03008
VOE	$\boldsymbol{M_{1,VOE}}$	$\boldsymbol{M_{2,VOE}}$	$\boldsymbol{M_{3,VOE} = M_{4,VOE}}$	
MAE	0.13800	**0.13288**		
RMSE	0.18736	**0.18077**		
MSE	0.03510	**0.03268**		
VBR	$\boldsymbol{M_{1,VBR}}$	$\boldsymbol{M_{2,VBR}}$	$\boldsymbol{M_{3,VBR}}$	$\boldsymbol{M_{4,VBR}}$
MAE		**0.13363**	0.13712	
RMSE		**0.01842**	0.18666	
MSE		**0.03393**	0.03484	
VEA	$\boldsymbol{M_{1,VEA} = M_{2,VEA}}$		$\boldsymbol{M_{3,VEA}}$	$\boldsymbol{M_{4,VEA}}$
MAE	0.14138		0.14209	0.14027
RMSE	**0.18530**		0.18677	0.18765
MSE	**0.03434**		0.03488	0.03521
VWO	$\boldsymbol{M_{1,VWO}}$	$\boldsymbol{M_{2,VWO}}$	$\boldsymbol{M_{3,VWO}}$	$\boldsymbol{M_{4,VWO}}$
MAE	**0.13097**		0.13206	
RMSE	**0.16789**		0.16896	
MSE	**0.02819**		0.02855	

Table 15 – Results: return predictions in function of the sentiment

From the *Table 15*, one can observe that for the VTI, VTV, VEA and VWO, the sentiment helps to predict the return with a lag of 1 day. However, for the VOE and VBR, the return can be best predicted 2 days after the publication of the tweets. This can be explained by the fact that companies which make up the VOE and VBR are less famous on Twitter than the others. One could also suppose that fewer people follow the stock price of those companies and therefore the fluctuation on the stock market will be less sensitive to tweets.

4.3.3. Returns from views

To provide the views for the future expected return, the model which has the smallest MAE, RMSE and MSE is chosen for each ETF. However, as outlined earlier, it was not possible to get sentiments from bond ETFs. Therefore, it has been decided to use the price and volume of 2018 for these ETFs in order to predict future returns.

	Investor's view	**IEE**
VTI	0.00055	-0.000457
VTV	0.00174	-0.000413
VOE	-0.00024	-0.000398
VBR	0.00044	-0.000402
VEA	-0.00076	-0.000379
VWO	-0.0007	-0.000520
SHV	0.000038	0.000001
VTIP	0.000095	0.0000035
MUB	-0.000106	0.0000095
AGG	-0.000033	0.0000081
BNDX	-0.000002	0.0000039
EMB	-0.000125	-0.0000807

Table 16 - Returns from views

From *Table 16*, one can see that the stocks ETFs are shared between positive and negative values whereas it is expected that four out of six bond ETFs will give a negative return in 2019. These results counterbalance those found for the Implied Equilibrium Excess return vector, where five out of six bond ETFs are forecast to give a positive return.

Before computing the combined return vector, one has to remember that three parameters remain to be computed. Firstly, the P matrix, which expresses whether investor's views are formulated in absolute or relative terms. In the theoretical framework, it has been determined that the P matrix can be viewed as an identity matrix, hence all views are absolute views. In this case, the asset weights matrix is *12 x 12* matrix. Furthermore, the covariance matrix of views can be simplified as well. As recommended, the τ has been set to 0.025. *Figure 14* shows the Ω matrix.

	VTI	VTV	VOE	VBR	VEA	VWO	SHV	VTIP	MUB	AGG	BNDX	EMB
VTI	1.8067e-06	0	0	0	0	0	0	0	0	0	0	0
VTV	0	1.5449e-06	0	0	0	0	0	0	0	0	0	0
VOE	0	0	1.4566e-06	0	0	0	0	0	0	0	0	0
VBR	0	0	0	1.5955e-06	0	0	0	0	0	0	0	0
VEA	0	0	0	0	1.2858e-06	0	0	0	0	0	0	0
VWO	0	0	0	0	0	2.6044e-06	0	0	0	0	0	0
SHV	0	0	0	0	0	0	1.6e-09	0	0	0	0	0
VTIP	0	0	0	0	0	0	0	1.54e-08	0	0	0	0
MUB	0	0	0	0	0	0	0	0	4.82e-08	0	0	0
AGG	0	0	0	0	0	0	0	0	0	4.9e-08	0	0
BNDX	0	0	0	0	0	0	0	0	0	0	3.13e-08	0
EMB	0	0	0	0	0	0	0	0	0	0	0	2.33e-07

Figure 14 - Covariance matrix of views

In section 3.3., we have built the Black-Litterman model thanks to market data for the IEE return vector and a sentiment analysis for the investor's views. We have firstly seen that the IEE return vector widely varies from the mean expected return vector of the Mean Variance Optimisation. Most of the time, the Black-Litterman model without views is able to provide an investor with a higher return for the same level of risk borne in the MVO, which partially confirms the first research hypothesis. Over a second phase, results of the sentiment analysis show that most of our tweets are positive. However, after a manual control, one has mentioned that the accuracy of the model is 67,69%. Therefore, we could expect that the number of positive tweets is not as high as claimed by the sentiment() function. From that point, we have performed a transfer entropy test to check whether it exists nonlinear dependencies between sentiments and the ETF returns for several company combinations. By retaining models presenting the lowest p-values for the transfer entropy test for stock ETFs, it has been possible to predict investor's expected return thanks to a LSTM neural network. Returns of bond ETFs have been predicted with the help of the price and volume variable in 2018. Finally, one computes the simplified P and Ω matrix so as to get all parameters that are necessary to compute the expected return of the Black-Litterman model.

4.4. Comparison of the Mean Variance Optimisation and Black-Litterman model

From the results of *Table 17*, it is worth comparing the expected return of the Black Litterman model with the MVO one. From the table below, one observes that only half of the daily expected returns of the BL model are negative compared to the MVO where all expected returns are negative. Moreover, the difference between both returns is always positive ranging from 0.000071 for the BNDX to 0.000766 for the VOE. Values also suggest that on average the daily expected return changes by 0.037% between both models.

Expected Return	**Black Litterman Model**	**Mean Variance Optimisation**	**Difference**
VTI	0.000272	-0.000270	0.000542
VTV	0.000333	-0.000290	0.000623
VOE	0.000263	-0.000503	0.000766
VBR	0.000263	-0.000480	0.000743
VEA	-0.000072	-0.000615	0.000543
VWO	-0.000258	-0.000602	0.000344
SHV	0.000017	-0.000071	0.000088
VTIP	0.000042	-0.000122	0.000164
MUB	-0.000037	-0.000111	0.000073
AGG	-0.000015	-0.000140	0.000125
BNDX	-0.000004	-0.000075	0.000071
EMB	-0.000033	-0.000382	0.000349

Table 17 - Comparison of expected returns between the MVO and BL model

The combined return vector of the BL model is employed in order to draw the efficient frontier. In *Figure 15*, the MVO efficient frontier is compared with BL one. One can firstly mention that the BL efficient frontier is largely above the other one. On the left-hand side of the graph, one notes that with the Black Litterman model, an investor who wants to carry the minimum level of risk is able to get a positive and higher expected return than the MVO for the same level of risk. Moreover, as previously seen in *Figure 6*, the BL model can propose more efficient portfolio than the MVO. For example, thanks to investor's views, the model expects that in 2019, it will be possible to get a daily return of 0.04% whereas the MVO is unable to do so.

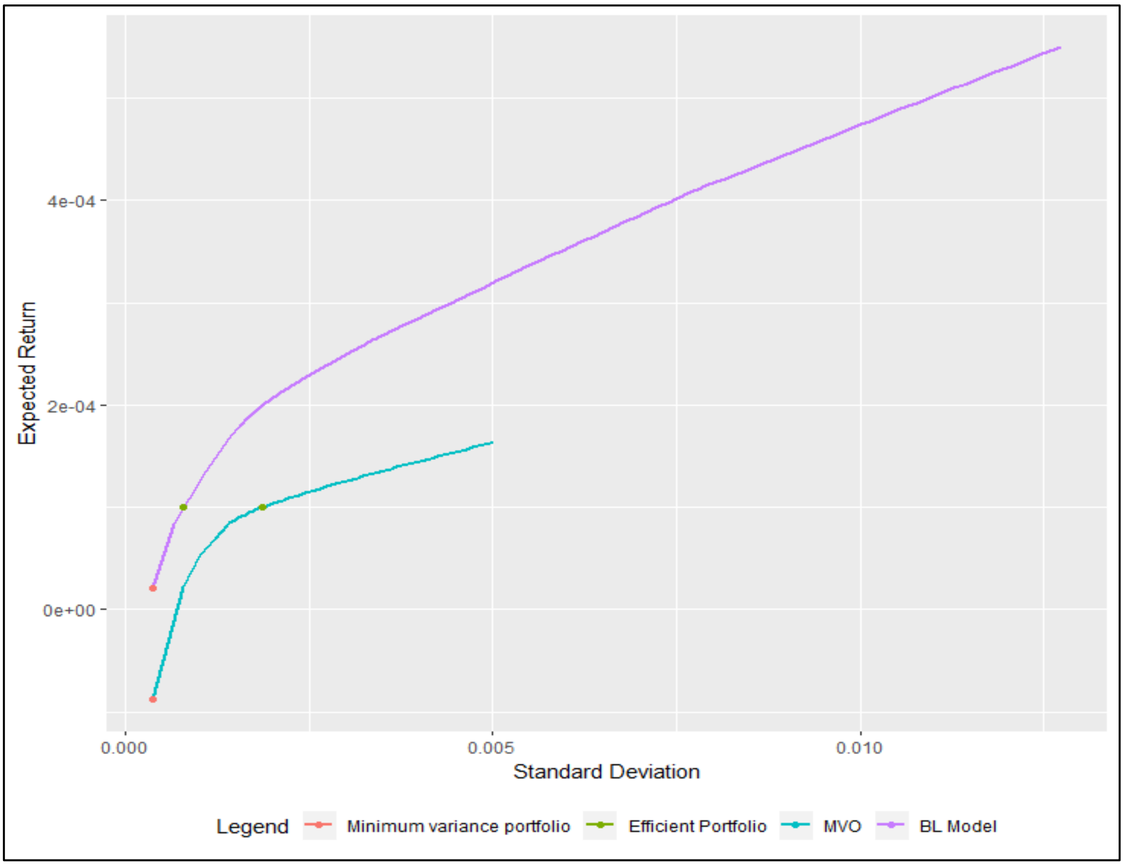


Figure 15 - Comparison between the MVO and BL model

Finally, our investor who would like to get a daily return of 0.01% will improve his risk-return trade-off going from 0.0146 to 0.0346. Hereunder, *Table 18* compares weights generated by both financial models. The BL model aims for better diversifying the investor's portfolio weights. It reduces short sell made in all stock ETFs except for VTI where the BL model proposes a negative weight unlike the MVO. It is worth mentioning that the weight limit of 50% has been reached for SHV in both models as it remains the most profitable asset among all. The BL model also managed to reduce more than twice the standard deviation going from 0.0019 for the MVO to 0.0008. These observations allow us to confirm our second research hypothesis stating that investors sentiment improve the risk-adjusted return of the robo-advisor advanced financial model.

	MVO	BL
VTI	0.3111	-0.0226
VTV	-0.1	0.1949
VOE	-0.1	-0.0647
VBR	-0.1	0.0367
VEA	-0.1	-0.0854
VWO	-0.1	-0.0339
SHV	0.5	0.5
VTIP	0.1013	0.5
MUB	0.2875	-0.1
AGG	-0.1	0.0111
BNDX	0.5	0.0986
EMB	-0.1	-0.0349

Table 18 - Weights comparison between the MVO and BL model

Finally, the BL model is able to better balance the weights between stock and bond ETFs. *Table 19* presents these results:

	MVO	BL
Stock ETFs	-18.89%	2.51%
Bond ETFs	118.19%	97.49%

Table 19 - Weights repartition between the MVO and BL model

To better illustrate the discrepancy between the weight allocation of both financial models. One can look at *Figure 16* where the weight of each ETF is shown as the expected return increases. It can be seen that for most ETFs, it exhibits different trends when the expected return rises.

In section 4.4., the Mean Variance Optimisation and the Black Litterman model have been evaluated. It has been found that the latter model outperforms the former on various aspects. First of all, one notices that the BL model only proposes portfolios with positive expected daily returns along its efficient frontier. Secondly, this model is also able to propose a better return for the same level of risk borne in the MVO framework. Furthermore, as the investor's views provide more optimistic values for 2019, the BL model offers portfolios that can deliver a higher return. Finally, *Table 19* shows that the weights between stocks and bond ETFs are better balanced for our investor even if it is still largely skewed towards bond ETFs.

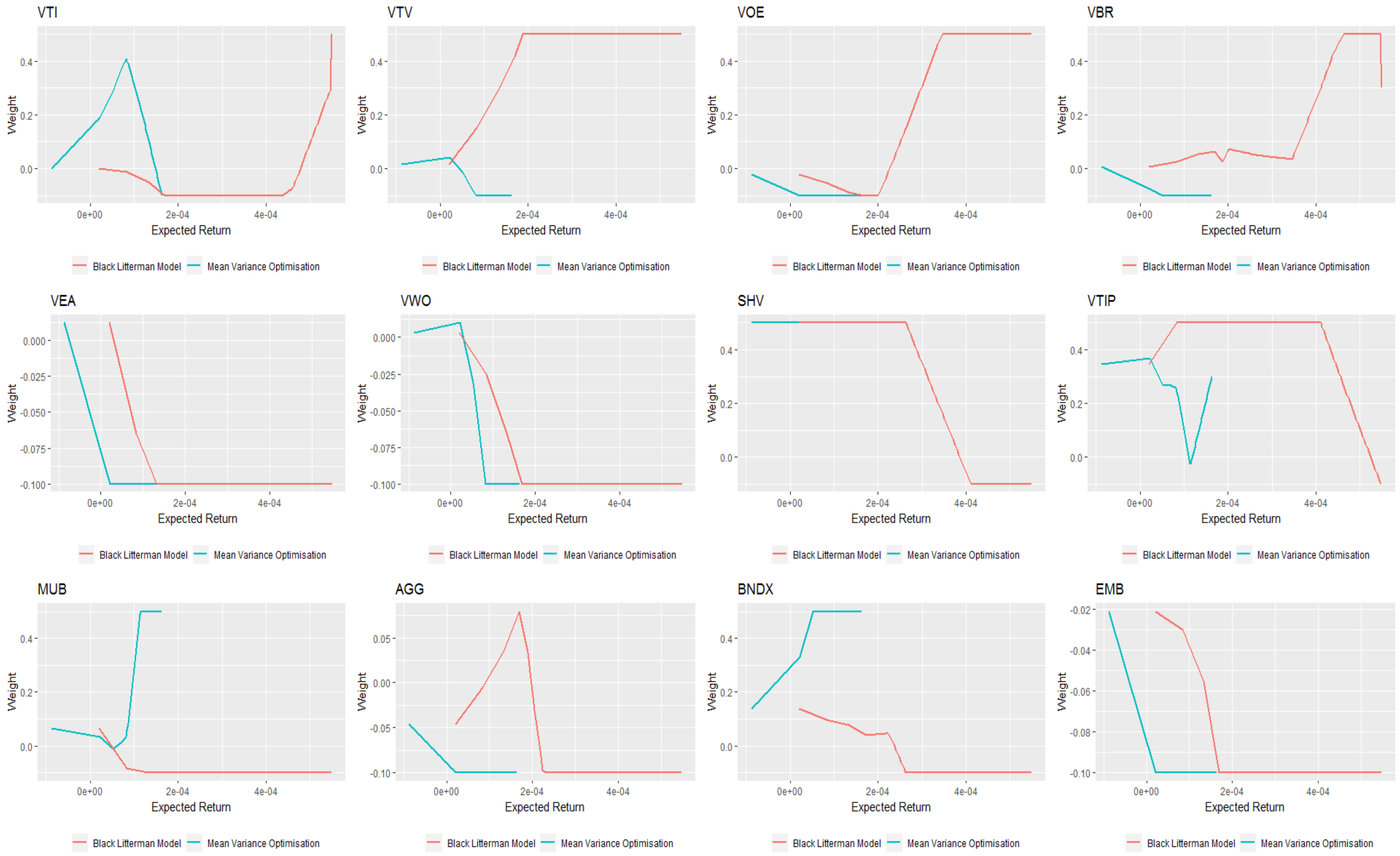


Figure 16 - Weights and expected returns evolution

4.5. Limits

Throughout this thesis, a variety of concepts have been discussed in order to provide the most relevant analysis regarding the implementation of sentiment analysis in the Black Litterman model. However, this work faces some limits that are reviewed in this section.

For the sake of efficiency, it has been decided to only retrieve tweets for the year 2018. As it was focused on analysing tweets of many companies, the task would had been highly time consuming if one wants to focus on a greater time frame. To support this argument, the Python script used already took several days to extract tweets that were necessary for the sentiment analysis. Moreover, the sentiment() function could have taken some days to return the sentiment score as this task is resource-demanding. This limited time period has also been paramount in the determination of the portfolio weights for both financial models. Indeed, in order to better approximate the vector of expected returns, it is advised to capture as many historical data as possible as long as it is relevant and fits the current state of the market. Besides, the small dataset has an impact on the prediction capability of the LSTM. As previously stated, the deep learning algorithm has been trained on roughly 70% of the dataset whereas the remaining part was the test set. In most cases, machine learning models are trained on a bigger set of data in order to learn how hypothetical booms and busts are caused on stock markets.

Regarding the tweets, it turned out that it is sometimes difficult to find a tall enough number of relevant tweets for the sentiment analysis. It has been uncovered that less known companies possess a lower number of relevant tweets. In the end, some of the companies analysed only had few tweets that were suitable for the sentiment analysis. Consequently, one fails to provide a sentiment score for each day and for each company. This shortcoming can lead to irrelevant results while executing the LSTM on the test set. In addition, tweets are a random source of information. It is unsure whether the quality and quantity of tweets will be relevant for any other sentiment analysis so as to get a relevant sentiment score for each day. This is particularly true for less known companies.

5. Conclusion and Future Works

In this thesis, one wanted to determine whether investors sentiment has an effect on the risk-return trade-off of the robo-advisor's financial model.

In the first place, we have built the Markowitz's Mean Variance Optimisation framework as it turned out to be the most widely used financial model for robo advisors worldwide. As the chosen historical data provides on average a negative return for 2018, a risk-averse investor could not expect to get a positive return for 2019. The only way to get one is to largely short sell his portfolio. In this thesis, we have assumed the investor would like to get a daily return of 0.01%.

Over a second phase, we have constructed the Implied Equilibrium Excess return vector that is part of the Black-Litterman model. As this vector is known to be the neutral starting point for any investor who would like to invest in financial markets, it has been decided to compare it with the MVO. Through this analysis, we have seen that the BL model is able to provide only positive and higher expected returns unlike the MVO. It also proposes new efficient portfolio that the MVO was unable to produce. However, it has been mentioned that MVO efficient frontier outperforms the BL one in terms of risk-return trade-off. In this case, our investor was slightly better off with the MVO. This first analysis allows us partially validating the first research hypothesis.

Subsequently, we have undertaken a sentiment analysis so as to provide the Black-Litterman model with investors views which make up the second part of the model. For this analysis, we have gone beyond the simplistic "Bag of words" technique by including valence shifters. It allowed us to better capture the investor sentiment by checking for negators, amplifiers, de-amplifiers and adversative conjunctions. With the help of the NTUSD-Fin dictionary as a subjective dictionary, the model has reached an accuracy of 67.69%. Then, the Shannon transfer entropy test has been used so as to detect whether it exists nonlinear dependencies between sentiments and ETF returns of various company combinations. By selecting the company combinations that provide the lowest p-values for different lags from the Shannon entropy test, we perform the prediction of future returns thanks to the Long Short Term Memory neural network which has demonstrated to be highly effective in time series forecasting.

Lastly, we compare the MVO with the complete Black-Litterman model in order to confirm or disprove the second research hypothesis stating that the integration of investors sentiment in the robo-advisor's financial model improve the risk-return trade-off. By taking a closer look at the efficient frontiers, we clearly see that the complete BL model largely does better than the classical Markowitz's Mean Variance Optimisation framework. As previously mentioned in the first efficient frontier comparison, the BL model still provides positive and higher returns than the MVO. However, this is true for any point along the efficient frontier unlike before. One also remarks that the complete BL model provides a new set of efficient portfolios that allow any investor to expect a higher return in 2019. Finally, our investor would opt for the latter model as it improves his risk-adjusted return.

Propositions for future works mainly lie in the enhancement of the sentiment analysis. As a matter of fact, it could be interesting to further extend the sources of sentiments, such as newspapers, financial reports, etc. as in this thesis we have focused on messages coming from Twitter. Furthermore, in order to improve the accuracy of the sentiment analysis model, it could be worth fine-tuning the sentiment() function and creating an augmented subjective dictionary that could increase the number of words treated by the model. Eventually, other sentiment analysis techniques such as supervised learning techniques could be used as they are known to outperform unsupervised lexicon-based techniques. All in all, the purpose of these suggestions is to improve the prediction of future returns so as to provide investors with a more reliable robo-advisor.

6. Bibliography

6.1. Books

Berk, J. & DeMarzo, P. (2014). Corporate Finance (3^{rd} ed.). Boston: Pearson Education.

Bodie, Z., Kane, A. & Marcus, A. (2011). Investment and Portfolio Management (10^{th} ed.). New York: McGraw-Hill Education.

Chollet, F. & Allaire, J.J. (2018). Deep learning with R. Manning Publications.

Lambin, J-J. & de Moerloose, C. (2016). Marketing stratégique et opérationnel : La démarche marketing dans l'économie numérique (4th ed.). Malakoff, France: Dunod

Rardin, R. (2014). Optimization in operations research. Hans, Great Britain: Pearson Education Limited.

Sironi, P. (2015). Modern portfolio management: From Markowitz to probabilistic scenario optimisation. Goal-based and long-term portfolio choice. United Kingdom: Risk Books

6.2. Scientific, business articles and university courses

Beketov, M., Lehmann, K. & Wittke, M. (2018). Robo advisors: Quantitative methods inside the robots. *Journal of Asset Management, 19*(6), 363-370. doi: 10.1057/s41260-018-0092-9

Bengio, Y. & Lecun, Y. (1997). Convolutional networks for images, speech and time series. In M. Arbib, The handbook of brain theory and neural networks (pp. 255-258). United States of America: The MIT Press

Best, M.J. & Grauer, R.R. (1991). On the sensitivity of mean-variance-efficient portfolios to changes in asset means: Some analytical and computational results. *The Review of Financial Studies, 4*(2), 315–342. doi: 10.1093/rfs/4.2.315

Black, F. & Litterman R. (1992). Global portfolio optimization. Financial Analysts Journal, 48(5), 28-43. Doi: 10.2469/faj.v48.n5.28

Broadie, M. (1993). Computing efficient frontiers using estimated parameters. *Annals of Operations Research, 45*(1), 21-58. doi: 10.1007/BF02282040

Chen, C-C., Huang, H-H. & Chen, H-H. (2018). NTUSD-Fin: A market sentiment dictionary for financial social media data applications. *Proceedings of the 1st Financial Narrative Processing Workshop*.

Cornu, M. (2018). Valeur de portefeuille et diversification : quelle est la taille de portefeuille optimale en tenant compte des coûts de transaction ? (Master's thesis). Louvain School of Management. Louvain-la-Neuve.

Cortis, K., Freitas, A., Daudert, T., Huerlimann, M., Zarrouk, M., Handschuh, S., & Davis, B. (2017). Semeval-2017 task 5: Fine-Grained Sentiment Analysis on Financial Microblogs and News. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)* (pp. 519-535). Vancouver: Association for Computational Linguistics.

Cronqvist, H. & Siegel, S. (2014). The genetics of investment biases. *Journal of Financial Economics, 113*(2), 215-234. doi: 10.1016/j.jfineco.2014.04.004

Elton, E. J., & Gruber, M. J. (1977). Risk reduction and portfolio size: An analytical solution. *The Journal of Business*, *50*(4), 415–437. doi: 10.1086/295964

Evans, J. L., & Archer, S. H. (1968). Diversification and the reduction of dispersion: an empirical analysis. *The Journal of Finance*, *23*(5), 761–767. doi: 10.1111/j.1540-6261.1968.tb00315.x

Harbans, L. & Dhingra, S. (1980). Effects of estimation risk on efficient portfolios: A Monte Carlo simulation study. *Journal of Business Finance & Accounting, 7*(2), 277-295. doi: 10.1111/j.1468-5957.1980.tb00743.x

He, G. & Litterman, R. (1999) The intuition behind Black-Litterman model portfolios. Retrieved from Goldman Sachs Quantitative Resources Group, Investment Management Division Web site: https://ssrn.com/abstract=334304

Horchreiter, S. & Schmidhuber J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Huang, A., Zang, A. & Zheng, R. (2013). Large sample evidence on the informativeness of text in analyst reports. *The Accounting Review, 89*(6), 2151-2180. doi: 10.2139/ssrn.1888724

Idzorek, T. (2004). A step-by-step guide to the Black-Litterman model – Incorporating user-specified confidence levels. In S. Satchell, Forecasting expected returns in the financial markets (pp. 17-38). Great Britain: Elsevier

Jacquemin, A. (2017). Séminaire d'accompagnement au mémoire : Penser sa méthodologie. Unpublished document, Université Catholique de Louvain, Louvain-la-Neuve.

Jozefowicz, R., Zaremba, W. & Sutskever, I. (2015). An empirical exploration of recurrent network architectures. *32nd International Conference on Machine Learning*, 3, 2332-2340.

Kingma, D. P. & Ba, J. (2014). *Adam: a method for stochastic optimization*. Paper presented at 3rd International Conference for Learning Representations, San Diego. Retrieved from https://arxiv.org/pdf/1412.6980.pdf

Kullback, S. & Leibler, A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics, 22*(1), 79-86. doi: 10.1214/aoms/1177729694

Lee, C. & Swaminathan, B. (2000). Price momentum and trading volume. *The Journal of Finance, 55*(5), 2017-2069. doi: 10.1111/0022-1082.00280

Li, Q., Jiang, L. & Li, P. (2015). Tensor-based learning for predicting stock movements. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence* (pp. 1787-1790). Menlo Park, United States of America: The AAAI Press

Litterman, B. (2003). Beyond equilibrium, the Black-Litterman approach. In *Modern investment management: An equilibrium approach* (pp. 81-87). New York: John Wiley & Sons

Loughran, T. & McDonald B. (2011). When is a liability not a liability? Textual analysis, dictionaries, and 10-ks. *The Journal of Finance, 66*(1), 35-65. doi: 10.1111/j.1540-6261.2010.01625.x

Lummer, S. L., Riepe, M.W. & Siegel, L. B. (1994). Taming your optimizer: A guide through the pitfalls of mean-variance optimization. In J. Lederman & R. A. Klein, *Global Asset Allocation: Techniques for Optimizing Portfolio Management*. New York: John Wiley & Sons

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77-91.

Michaud, R.O. (1989). The Markowitz optimization enigma: Is optimized optimal? *Financial Analysts Journal, 45*(1), 31–42.

Mittal, A. & Goel A. (2012). Stock prediction using twitter sentiment analysis. *Stanford University Working Paper*.

Moulliet, D., Stolzenbach, J., Majonek, A. & Völker, T. (2016). The expansion of robo-advisory in wealth management. *Deloitte White Paper*.

O'Connor, B., Balasubramanyan, R., Routledge, B.R., Smith, N.A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media* (pp. 122-129). Menlo Park, United States of America: The AAAI Press

Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters, 85*(2), 461-464. doi: 10.1103/PhysRevLett.85.461

Shannon, C.E. (1948). A mathematical theory of communication. The Bell System Technical Journal, 27(3), 379-423. doi: 10.1002/j.1538-7305.1948.tb01338.x

Sharpe, W. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *The Journal of Finance, 19*(3), 425-442. doi: 10.1111/j.1540-6261.1964.tb02865.x

Statman, M. (1987). How many stocks make a diversified portfolio? *Journal of Financial and Quantitative Analysis, 22*(3), 353-363. doi: 10.2307/2330969

Swensen, D. (2009). Pioneering portfolio management: An unconventional approach to institutional investment. New York: Free Press.

Tobin, J. (1958). Liquidity preference as behaviour towards risk. *The Review of Economic Studies, 25*(2), 65-86. doi: 10.2307/2296205

Walters, J. (2014). The Black-Litterman model in detail. Retrieved from Boston University, Metropolitan College, Department of Computer Science Web site: https://ssrn.com/abstract=1314585

Xing, F., Cambria, E. & Welsch, R. (2018). Intelligent asset allocation via market sentiment views. *IEEE Computational Intelligence Magazine, 13*(4), 25-34. doi: 10.1109/MCI.2018.2866727

Xing, F., Cambria, E., Malandri, L., Vercellis, C. (2019). Discovering bayesian market views for intelligent asset allocation. *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics), 11053*, 120-135.

Xiong, J. & Idzorek, T. (2011). The impact of skewness and fat tails on the asset allocation decision. *Financial Analysts Journal, 67*(2), 23-35. doi: 10.2469/faj.v67.n2.5

6.3. Web articles

Betterment (2018). *ETF Selection for Portfolio Construction: A Methodology*. Retrieved from https://www.betterment.com/resources/research/etf-portfolio-selection-methodology/

Colah’s blog. (2015). *Understanding LSTM Networks*. Retrieved from https://colah.github.io/posts/2015-08-Understanding-LSTMs/

CRSP. (n.d.). *CRSP U.S. Total Market Index*. Retrieved July 14, 2019 from http://www.crsp.com/products/investment-products/crsp-us-total-market-index

ETFdb.com. (2019). *Barclays Capital U.S. Aggregate bond Index – ETF Tracker*. Retrieved July 14, 2019 from https://etfdb.com/index/barclays-capital-us-aggregate-bond-index/

ETFdb.com. (2019). *Barclays Capital U.S. Treasury Inflation Protected Securities (TIPS) Index (Series-L) – ETF Tracker*. Retrieved July 14, 2019 from https://etfdb.com/index/barclays-capital-us-treasury-inflation-protected-securities-tips-index-series-l/

ETFdb.com. (2019). *Barclays Global Aggregate ex-USD Float-Adjusted Index (Hedged) – ETF Tracker*. Retrieved July 14, 2019 from https://etfdb.com/index/barclays-global-aggregate-ex-usd-float-adjusted-index-hedged/

FTSE Russel. (n.d.). *FTSE Benchmarks - AMERICAS*. Retrieved July 14, 2019 from https://research.ftserussell.com/Vanguard/Home/Indices

Hargrave, M. (2019). *Sharpe ratio*. Retrieved July 26, 2019 from https://www.investopedia.com/terms/s/sharperatio.asp

ICE. (2019). *U.S. Treasury Short Bond Index*. Retrieved July 14, 2019 from https://www.theice.com/publicdocs/data/IDCOTS-factsheet.pdf

J.P. Morgan. (2017). *Emerging Markets Bond Index Global CORE (EMBIG CORE) – Product Overview*. Retrieved from https://www.jpmorgan.com/jpmpdf/1320723155889.pdf

Lanaro, G. (2016). Evaluation measures for multiclass problems. Retrieved from http://gabrielelanaro.github.io/blog/2016/02/03/multiclass-evaluation-measures.html

Nasdaq. (2011). *Naïve diversification*. Retrieved from https://www.nasdaq.com/investing/glossary/n/naive-diversification#ixzz3xc7TsXgB

S&P Dow Jones Indices. (2019). *S&P National AMT-Free Municipal Bond Index*. Retrieved July 14, 2019 from https://us.spindices.com/indices/fixed-income/sp-national-amt-free-municipal-bond-index

Statista. (2019). *Robo-Advisors – Worldwide*. Retrieved July 14, 2019 from https://www.statista.com/outlook/337/100/robo-advisors/worldwide

Vanguard. (2019). *Market Liquidity Fund*. Retrieved July 14, 2019 from https://institutional.vanguard.com/web/cf/product-details/fund/1142

Appendices

Appendix A - Mathematical model of the sentiment function

$$\delta_{i**j} = \frac{c'_{i**j}}{\sqrt{w_{i,j**n}}}$$

Where:

$$c'_{i,j} = \sum((1 + w_{amp} + w_{deamp}) * w^{p}_{i,j,k} * (-1)^{2+w_{neg}})$$

$$w_{amp} = \sum(w_{neg} * (z * w^{a}_{i,j,k})$$

$$w_{deamp} = \max(w_{deamp'}, -1)$$

$$w_{deamp'} = \sum(z(-w_{neg} * w^{a}_{i,j,k} + w^{d}_{i,j,k}))$$

$$w_b = 1 + z_2 * w_{b'}$$

$$w_{b'} = \sum(|w_{adversative\ conjuction}|, \ldots, w^{p}_{i,j,k}, w^{p}_{i,j,k}, \ldots, |w_{adversative\ conjuction}| * -1)$$

$$w_{neg} = \left(\sum w^{n}_{i,j,k}\right) mod\ 2$$

i = paragraph number

j = sentence number

k = word number

Lower bound cluster:

$$max\left\{p ** w_{i,j,k-n**b}, 1, max\{c ** w_{i,j,k} < p ** w_{i,j,k}\}\right\}$$

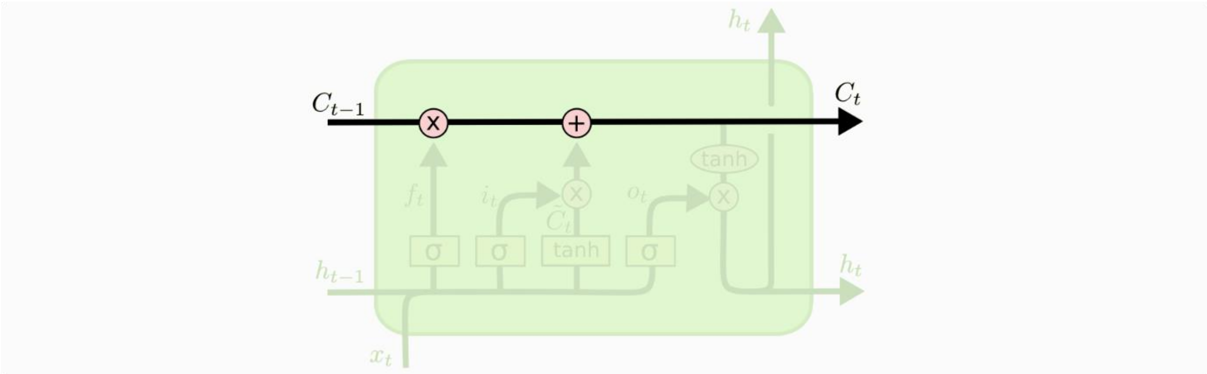
Upper bound cluster:

$$min\left\{p ** w_{i,j,k\,+\,n**a}, w_{i,j**n}, min\{c ** w_{i,j,k} > p ** w_{i,j,k}\}\right\}$$

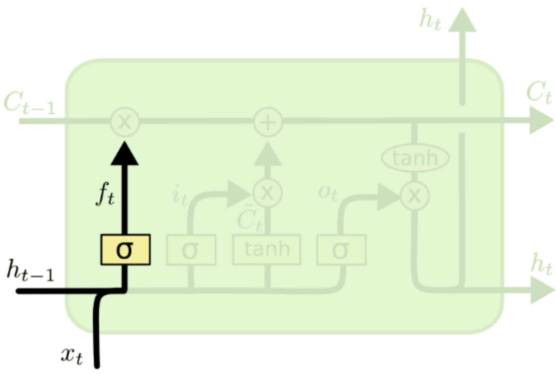
Where $w_{i,j**n} = number\ of\ words\ in\ the\ sentence$

Appendix B - Long Short Term Memory neural network

Cell state:

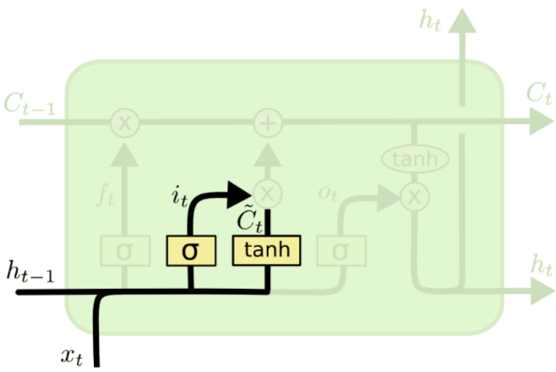


Forget gate layer:



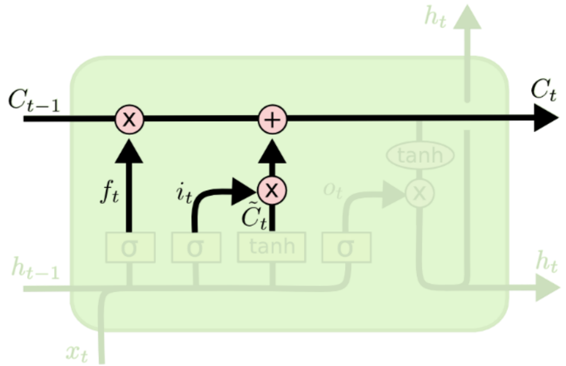
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Input gate layer:



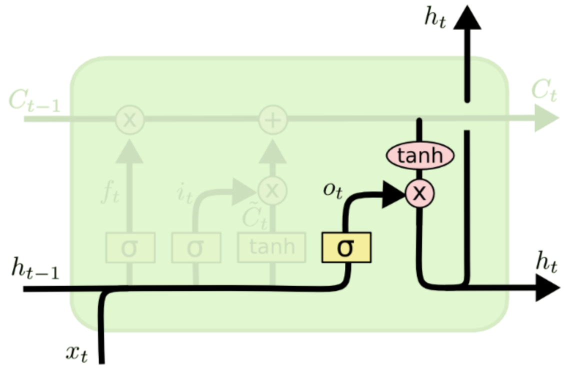
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

New cell state:



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output layer:



$$o_t = \sigma\left(W_o \ [h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Appendix C - Company codes

VTI			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
Apple Inc.	AAPL.OQ	$AAPL	2.96%
Microsoft Corp.	MSFT.OQ	$MSFT	2.79%
Amazon.com Inc.	AZMN.OQ	$AMZN	2.43%
Facebook Inc.	FB.OQ	$FB	1.43%
Berkshire Hathaway Inc.	BRKb.N	$BRK.B	1.37%
JPMorgan Chase & Co	JPM.N	$JPM	1.34%
Johnson & Johnson	JNJ.N	$JNJ	1.26%
Exxon Mobil Corp.	XOM.N	$XOM	1.20%
Alphabet Inc.	GOOGL.OQ	$GOOGL	2.39%
Bank of America Corp.	BAC.N	$BAC	0.92%
Wells Fargo & Co	WFC.N	$PFE	0.86%

VTV			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
Microsoft Corp.	AAPL.OQ	$AAPL	5.93%
Berkshire Hathaway Inc.	BRKb.N	$BRK.B	3.15%
JPMorgan Chase & Co	JPM.N	$JPM	2.98%
Johnson & Johnson	JNJ.N	$JNJ	2.81%
Exxon Mobil Corp.	XOM.N	$XOM	2.67%
Bank of America Corp.	BAC.N	$BAC	2.24%
Wells Fargo & Co	WFC.N	$WFC	1.91%
Chevron Corp.	CVX.N	$CVX	1.81%
AT&T Inc.	T.N	$T	1.76%
UnitedHealth Group Inc.	UNH.N	$UNH	1.90%
Intel Corp.	INTC.OQ	$INTC	1.82%
Pfizer Inc.	PFE.N	$PFE	1.86%
Verizon Communications Inc.	VZ.N	$VZ	1.72%
Procter & Gamble Co	PG.N	$PG	1.65%
Citigroup Inc.	C.N	$C	1.39%
Cisco Systems Inc.	CSCO.OQ	$CSCO	1.68%
Dow DuPont Inc.	DD.N	$DWDP	1.18%
Pepsico Inc.	PEP.OQ	$PEP	1.25%
Oracle Corp.	ORCL.N	$ORCL	1.11%
Merck & Co Inc.	MRK.N	$MRK	1.39%
Walmart Inc.	WMT.N	$WMT	1.09%
IBM Corp.	IBM.N	$IBM	1.02%
General Electric Co	GE.N	$GE	0.85%
Amgen Inc.	AMGN.OQ	$AMGN	1.01%

VOE			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
FreePort - McMoran Inc.	FCX.N	$FCX	1.01%
Western Digital Corp.	WDC.OQ	$WDC	0.95%
M&T Bank	MTB.N	$MTB	1.12%
KeyCorp	**KEY.N**	**$KEY**	**1.00%**
WEC Energy Group Inc.	WEC.N	$WEC	1.01%
Eversource Energy	ES.N	$ES	0.95%
DTE Energy Co	DTE.N	$DTE	0.95%
Willis Tower Watson Plc.	WLTW.OQ	$WLTW	0.97%
Royal Caribbean Cruises Ltd.	RCL.N	$RCL	0.94%
Citizens Financial Group Inc.	FCG.N	$FCG	0.93%
Motorola Solutions Inc.	**MSI.N**	**$MSI**	**0.92%**
Newmont Goldcorp Corp.	NEM.N	$NEM	0.93%
NetApp Inc.	NTAP.OQ	$NTAP	0.91%
Regions Financial Corp.	RF.N	$RF	0.98%

VBR			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
Spirit Aerosystems Holdings Inc.	SPR.N	$SPR	0.50%
IDEX Corp.	IEX.N	$IDEX	0.55%
On Semiconductor Corp.	ON.OQ	$ON	0.47%
Leidos Holdings Inc.	LDOS.N	$LDOS	0.50%
Atmos Energy Corp.	ATO.N	$ATO	0.52%
East West Bancorp Inc.	EWBC.OQ	$EWBC	0.46%
NRG Energy Inc.	NRG.N	$NRG	0.48%
Wellcare Health Plans Inc.	WCG.N	$WCG	0.44%
UGI Corp.	UGI.N	$UGI	0.46%
PerkinElmer Inc.	**PKI.N**	**$PKI**	**0.46%**

VEA			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
Nestle SA	NESN.S	$NSRGY	1.28%
HSBC Holdings Plc.	HSBA.L	$HSBC	0.95%
Samsung Electronics Co Ltd.	005930.KS	$SSNLF	0.92%
Novartis Ag.	NOVN.S	$NVS	0.94%
Toyota Motor Corp.	7203.T	$TM	0.83%
Royal Dutch Shell Plc.	RDSa.AS	$RDS.A	0.77%
British American Tobacco Plc.	BATS.L	$BTI	0.57%
BP Plc.	BP.L	$BP	0.71%
Total SA	TOTF.PA	$TOT	0.74%
Roche Holding Ag	ROG.S	$RHHBY	0.85%

VWO			
Companies	**RIC**	**Cashtag**	**Mean weight 2018**
Tencent Holdings Ltd.	0700.HK	$TCEHY	4.80%
Naspers Ltd.	NPNJn.J	$NPSNY	1.82%
Taiwan Semiconductor Manufacturing Co Ltd.	**2330.TW**	**$TSM**	**3.49%**
Alibaba Group Holding Ltd.	BABA.N	$BABA	2.88%
China Construction Bank Corp.	0939.HK	$CICHY	1.56%
Ping An Insurance Group Co of China Ltd.	2318.HK	$PNGAY	0.95%
China Mobile Ltd.	0941.HK	$CHL	0.92%
Sberbank Rossii Pao	SBER.MM	$SBRCY	0.67%
Baidu Inc.	BIDU.OQ	$BIDU	0.97%
Reliance Industries Ltd.	RELI.NS	$RS	0.89%

Appendix D - Company combinations

$M_{1,VTI}$	Microsoft, Facebook, Berkshire Hathaway
$M_{2,VTI}$	Microsoft, Berkshire Hathaway, JPMorgan
$M_{3,VTI}$	Microsoft, Facebook, Wells Fargo
$M_{4,VTI}$	Berkshire Hathaway, ExxonMobil

$M_{1,VTV}$	Microsoft, Berkshire Hathaway, JPMorgan
$M_{2,VTV}$	Microsoft
$M_{3,VTV}$	Berkshire Hathaway, Johnson & Johnson
$M_{4,VTV}$	Berkshire Hathaway, Intel

$M_{1,VOE}$	FreePort-McMoran, M&T Bank, Royal Caribbean
$M_{2,VOE}$	Western Digital, M&T Bank, Citizens Financial, Regions Financial, Newmont, Wec
$M_{3,VOE}$	FreePort-McMoran, Royal Caribbean, Citizens Financial, WEC
$M_{4,VOE}$	FreePort-McMoran, Royal Caribbean, Citizens Financial, WEC

$M_{1,VBR}$	ON Semiconductor, Leidos, NRG, Wellcare, UGI Corp
$M_{2,VBR}$	Spirit Aero, ON Semiconductor, Leidos, East West Bancorp, Wellcare, UGI Corp
$M_{3,VBR}$	Spirit Aero, ON Semiconductor, Leidos, Atmos, East West Bancorp, UGI Corp
$M_{4,VBR}$	Spirit Aero, IDEX, ON Semiconductor, Leidos, Atmos, NRG
$M_{1,VEA}$	HSBC, Novartis, Toyota, Roche
$M_{2,VEA}$	HSBC, Novartis, Toyota, Roche
$M_{3,VEA}$	Nestle, HSBC, Novartis, Toyota, Roche, Shell
$M_{4,VEA}$	Nestle, HSBC, Novartis, Toyota, Roche

$M_{1,VWO}$	Alibaba, China Mobile
$M_{2,VWO}$	Tencent, China Mobile, Baidu, Reliance
$M_{3,VWO}$	Tencent, Alibaba
$M_{4,VWO}$	Tencent, China Mobile, Reliance

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Louvain School of Management
Chaussée de Binche 151, 7000 Mons, Belgique | www.uclouvain.be/lsm