Is it possible to predict the transaction costs of orders in stock portfolios?

Martijn Huig

(2556214)

Master Thesis Business Analytics

Is it possible to predict the transaction costs of orders in stock portfolios?

Martijn Huig

(2556214)

Vrije Universiteit Amsterdam

Faculty of Science

Business Analytics

De Boelelaan 1081a

1081 HV Amsterdam

Host organisation:

PGGM

Noordweg Noord 150
3704 JG Zeist

Mentor PGGM: Lukas ten Berge

First supervisor: Jan Bouwe van den Berg

Second supervisor: Mark-Jan Boes

Preface

Transaction costs have a big influence on the return of the stock portfolios for sizeable investment companies. These costs are variable and depend on the price movement of a stock. This makes transaction costs extremely hard to predict. In this thesis, machine learning algorithms are used to find if it is possible to predict the transaction costs of orders in stock portfolios.

This research was conducted between November 2020 and April 2021 at the risk control department at PGGM. The research was supervised by Lukas ten Berge, who is a senior transaction cost analyst at PGGM.

During this research, I have had daily meetings with Lukas ten Berge in which we mostly talked about transaction costs, what steps I needed to take to improve my results, and PGGM's weekly transactions. I want to specifically thank Lukas for the time and effort he took during my internship. I also would like to thank Betim Lushtaku. Betim is a junior performance analyst at PGGM who was also present during the daily meetings with Lukas and helped me with my research.

Furthermore, I would like to thank the supervisors at the VU who reviewed my research. A special thank you goes out to my first supervisor, Jan Bouwe van den Berg. Jan Bouwe is a mathematics professor at the Faculty of Science at the Vrije Universiteit Amsterdam. I had a meeting with Jan Bouwe once every two weeks where we discussed the proceedings of this research. These meetings all went very smoothly and it was always nice to see Jan Bouwe again. Next to this, Jan Bouwe helped me with my action plan and gave me some feedback on my research.

My second supervisor at the VU is Mark-Jan Boes. Mark-Jan is a professor at the Department of Finance at the Vrije Universiteit Amsterdam. I contacted Mark-Jan if he wanted to be my second supervisor because of his expertise in finance, and he immediately responded positively. Besides that, Mark-Jan gave me feedback on the financial background in this research.

Summary

The objective of this research is to find out if it is possible to predict the transaction costs of orders in stock portfolios.

Transaction costs are the difference between the average price that is actually paid for a stock and the price of this stock when the order entered the market. These costs consist of implicit and explicit costs. Explicit costs are fixed costs known before an order is executed, like fees and taxes, while implicit costs are influenced by the size of the order, volatility, liquidity, participation ratio, and momentum in the market. These are five known factors that influence a stock's price movement during the interval of the order. The price movement during the interval of an order influences the implicit costs. Implicit costs are variable costs, and during this research, these costs are measured using the market impact on arrival. The market impact on arrival is the difference between the average price that is paid for a stock and the price this order had when it was executed.

Because the market impact is the variable part of transaction costs, this part needs to be predicted. The data set consisted of 18768 orders that were executed between November 2016 and December 2019 and forty-four variables. Machine learning algorithms are used to form models with these variables to predict the market impact. During this research, four machine learning algorithms were used: multiple linear regression, artificial neural network, random forest, and gradient boosting tree. These four machine learning algorithms were optimised and compared.

The best performing machine learning algorithm is the random forest. The random forest has a MAE of 66,05 and a R-Squared of 0,1377. This means that only 13,77% of the variance in the market impact is explained by the random forest model. However, this is an improvement compared to the existing Bloomberg model and an improvement on simply taking the average market impact as a prediction. This shows that it is possible to predict the market impact to some extent. Transaction costs also consist of fixed costs, which are known before an order is executed. Therefore, this part can be accurately predicted. This makes it somewhat possible to predict the total transaction costs, which makes it a useful prediction.

Contents

1 Introduction

Transaction costs are receiving more and more attention in the financial world. These are the costs made when a transaction is executed. This research focuses on the transaction costs in PGGM's stock portfolios.

PGGM is a pension administration organisation with a large capital base. It holds the pensions of 4.4 million Dutch workers in the health and care sector and has assets worth 252 billion euros (PGGM.nl, 2020) in total. To ensure that these people get the pension they are promised, PGGM invests this money. In total PGGM invests around 32 billion euros in the stock market. According to MSCI World, the total market capitalisation currently is 44465 billion euros.[1] This makes PGGM a company that can influence the stock market worldwide.

Transaction costs are the difference between the price that is paid for a stock and the price this stock had when an order was executed. Transaction costs influence the price that is paid for a stock, and influence the return when a stock is sold. These costs are one of the essential parameters that affect the investment performance (Park, Lee, and Son, 2016). Still, there are a substantial amount of companies that do not include the transaction costs of a certain stock in their decision process when they form their stock portfolios. This raises the question: why do companies not consider these costs?

Transaction costs consist of implicit and explicit costs. Explicit costs are costs like fees and taxes, while implicit costs depend on the fluctuations of a stock's price during the time interval of an order. In this research, these implicit costs are measured with the market impact on arrival, and these costs form the most significant part of transaction costs.

Because PGGM has such a large capital base they mostly execute sizeable orders. These sizeable orders cannot be executed at once and have to be spread out over time. During this period, the price of a stock fluctuates, which often results in higher transaction costs. As a result, PGGM has to consider the influence of transaction costs on their investment performance.

Trading at PGGM is done by two different departments. The first side is the portfolio managers. These portfolio managers decide which stocks are bought and sold and determine the contents of PGGM's portfolio. The portfolio managers buy the stocks that they expect to go up in the future.

The traders are on the other side. These traders have to buy and sell stocks according to the portfolio managers' preference. Traders have to make sure that they buy and sell these stocks at the right time to make sure that the transaction costs are as low as possible. There are a few basic steps and variables that they consider during this process, which will be discussed in this research.

PGGM spends every year around 185 million euros on transaction costs with their transactions at the stock market. Around 25 million euros of these costs are fixed costs, and around 160 million euros of these costs are variable costs. The latter shows the significance of transaction costs, and the gravity of influence these costs can have on the return of PGGM's investments.

PGGM currently uses Bloomberg's prediction as an evaluation for the market impact value.

[1] The MSCI World Index captures large and mid-cap representation across 23 Developed Markets (DM) countries. With 1,586 constituents, the index covers approximately 85% of the free float-adjusted market capitalisation in each country (MSCI, 2021).

However, they do not know what Bloomberg's model looks like and PGGM does not have a model itself. More importantly, Bloomberg's prediction is often quite different from the actual market impact.

Bloomberg is one of the largest financial institutions in the world, and even for them, it is challenging to accurately predict these variable costs. Predicting these costs more precisely can help PGGM significantly when the portfolio managers make their investment decisions, or when the traders execute their trades. Because of this, the following research question is formulated:

Is it possible to predict the transaction costs of orders in stock portfolios?

This thesis is structured as follows. In *Section 2* an explanation of the operation of the stock market is given, and existing portfolio strategies are shown. In *Section 3* the liquidity in the stock market is discussed. Liquidity is connected with the transaction costs, and, therefore, clarification is given on the liquidity in the market. In *Section 4*, a comprehensive description of the variables that determine the transaction costs is given. This section describes what transaction costs exactly are, and how they can be calculated. *Section 5* contains an explanation of the data set used in this research. In *Section 6* the currently used model is evaluated. *Section 7* describes four machine learning algorithms used in this research to make models that predict the market impact. These four models are then optimised in *Section 8*, and the importance of the variables is stated. The results of our four models are then considered and compared in *Section 9*. Finally, the discussion and conclusion are in *Sections 10* and *11*.

The crucial sections in this thesis are *Sections 4, 7, 8, 9, 10 and 11*. In these sections transaction costs are comprehensively explained, the four machine learning algorithms that are used to get results are explained, and the optimal set of hyperparameters and variables is given, after which the results of our models are discussed and compared, and, finally, a conclusion for this thesis is given.

2 Portfolio strategies

In this section, an explanation of the most famous stock models is given. These stock models help to get a better view of the price fluctuations in the stock market, which is necessary to understand transaction costs. The description in this section is loosely based on the following four papers: H. Markowitz, 1959, E. Fama and Macbeth, 1973, E. F. Fama and French, 1992, Jegadeesh and Titman, 1993.

Most stock models started with the Modern Portfolio Theory (MPT) (H. Markowitz, 1959). This theory assumes that all investors are risk-averse, which means that investors want to be compensated for bearing extra risk. Therefore, if an investor can choose between two portfolios with the same expected return and different levels of risk, the investor will always choose the portfolio with the least amount of risk. A portfolio that has a higher risk should thus always have a higher expected return (Mangram, 2013). The MPT maximises a portfolio's expected return, given a certain level of risk or minimises risk, given a certain expected return level.

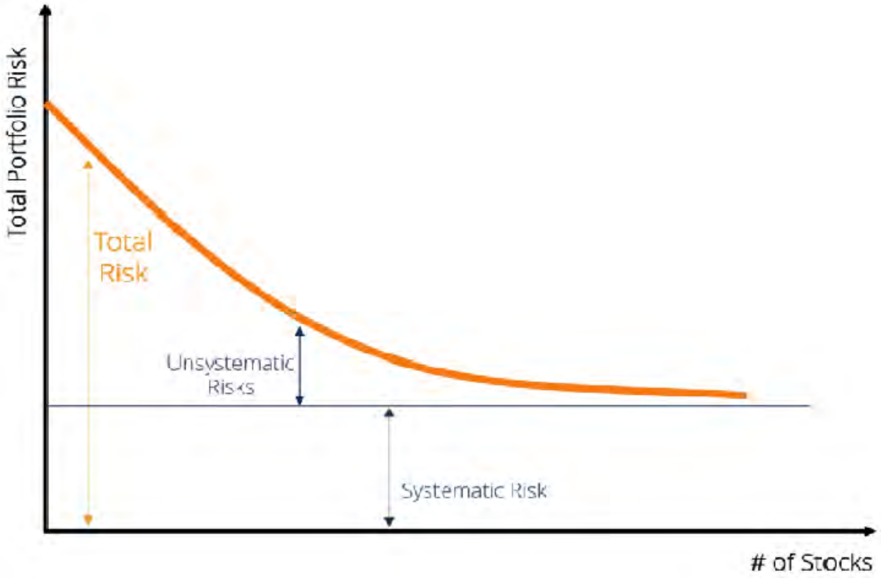
This theory focuses on the optimal combination of assets. There are two types of risk traders are confronted with: systematic risk and idiosyncratic risk.

Definition 1. Systematic risk is caused by the fluctuations of the stock price, and it is another word for volatility. All investments or securities are subject to systematic risk, and, therefore, this risk cannot be avoided.

Definition 2. Idiosyncratic risk (unsystematic risk) is the risk that exists within the company of the stock. This risk can be diversified away by having a lot of different stocks in your portfolio.

Informed traders will always look for the best combination of assets because choosing just one stock to invest in will yield a significant idiosyncratic risk. This relates to the well-known proverb that says: "Do not put all your eggs in one basket". If the basket is dropped, all eggs are broken; while if the eggs are placed in more than one basket, the risk that all eggs will be broken simultaneously is minuscule (Fabozzi, Gupta, and H. M. Markowitz, 2002). By diversifying their portfolio in many assets, traders can diversify this idiosyncratic risk away such that the portfolio becomes less risky.[2] This becomes clear in figure 1:

Figure 1: The relationship between systematic risk and idiosyncratic risk based on the number of stocks in a portfolio (CorporateFinanceInstitute, 2015)



It is possible to measure the portfolio's return by taking the weights and the returns of all stocks included in the portfolio. The expected return of a portfolio that consists of two stocks is therefore calculated in the following way:

$$E(r_p) = x_a * E(r_a) + x_b * E(r_b)$$

In this formula, $E(r_p)$ is the expected return of the portfolio. The calculation of this expected return consists of the following four factors: x_a is the fraction of equity invested in stock a and $E(r_a)$ is the expected return of stock a, while x_b is the fraction of equity invested in stock b and $E(r_b)$ is the expected return of stock b.

[2] A sizeable pension administration organisation like PGGM also diversifies its capital by investing in other financial instruments, like bonds, options, etcetera.

This portfolio's risk is determined by the individual risk of these stocks plus the correlation that these stocks have. The correlation measures the statistical relationship between two variables. In this case, the correlation measures the statistical relationship between the stock prices of two companies.

The expected risk of a portfolio that consists of two stocks can be calculated in the following manner:

$$\begin{aligned}\sigma^2\left(\underline{r}_p\right) = {} & x_A^2 \cdot \sigma^2\left(\underline{r}_A\right) \\ & + 2 \cdot x_A \cdot x_B \cdot \sigma\left(\underline{r}_A\right) \cdot \sigma\left(\underline{r}_B\right) \cdot \rho\left(\underline{r}_A, \underline{r}_B\right) \\ & + x_B^2 \cdot \sigma^2\left(\underline{r}_B\right)\end{aligned}$$

In this formula, $\sigma^2(r_p)$ is the expected risk of the portfolio. This is based on the following 5 factors: x_a and x_b still give the fraction of equity invested in these stocks, $\sigma^2(r_a)$ and $\sigma^2(r_b)$ are the expected variances of stocks a and b, and $\rho\left(\underline{r}_A, \underline{r}_B\right)$ measures the expected correlation between stocks a and b.

As already stated above, variance is a measure of the variability or spread in a set of data. Mathematically, it is the average squared deviation from the mean score. The variance of a stock A is thus calculated in the following way:

$$\operatorname{Var}(A) = \sum_{i=1}^{N} \left(A_i - \bar{A}\right)^2 / N$$

In this formula, A_i is the return of stock A on day i, $\bar{A}$ is the average of stock A's returns in period $t-1$, and N are the number of days in this period.

The correlation measures the statistical relationship between the stock prices of two companies. Companies that are very much alike often have positive and negative returns on the same day.

It is possible to calculate the correlation of two stocks by looking at the returns of both stocks. Covariance is a measure that captures the correlation between the stock prices of two stocks. The covariance of two stocks can be calculated in the following way:

$$\operatorname{Cov}(A, B) = \sum_{i=1}^{N} \frac{\left(A_i - \bar{A}\right)\left(B_i - \bar{B}\right)}{N}$$

In reality, a portfolio consists of more than two stocks. Therefore, a variance-covariance matrix must be formed with all stocks the portfolio manager has an interest in. In this matrix, the covariances between all stocks are calculated.

The Capital Asset Pricing Model (CAPM) is an example of an approximation model for stock prices. This model tries to find the relation between risk and expected return for stocks (E. Fama and Macbeth, 1973). Because of this, the CAPM is an expansion on the MPT that assumed that risk and expected return are positively related and that it is not possible to have a higher return in a portfolio without having a higher risk. The empirical translation of the CAPM is the market model. This is given by:

$$r_{i,t} = \alpha + \beta r_{m,t} + \epsilon_{i,t}$$

In this formula $r_{i,t}$ is the return of asset i in period t, α is some constant, β is the market risk, $r_{m,t}$ is the return of the market in period t and i, t is the residual of asset i in period t. The residual is the difference between the observed value and the estimated value of the quantity of interest.

Fama and Macbeth empirically tested the CAPM in their paper in 1973. Based on the CAPM equation, they tested three implications:

1. The relation between risk and return is linear.
2. Beta is a complete measure of the risk of an asset i in its portfolio m.
3. Higher risk should be associated with higher returns.

Fama and Macbeth tested these hypotheses in practice by estimating the following formula:

$$\widetilde{R}_{it} = \widetilde{\gamma}_{0t} + \widetilde{\gamma}_{1t}\beta_i + \widetilde{\gamma}_{2t}\beta_i^2 + \widetilde{\gamma}_{3t}s_i + \widetilde{\eta}_{it}$$

In this formula, s_i is some measure of the risk of asset i that is not deterministically related to β_i, the coefficient γ_2 is related to the first hypothesis that states that the relationship between risk and return is linear, which means that γ_2 should be zero. The coefficient γ_3 is related to the second hypothesis that beta is a complete measure of risk, while $\widetilde{\eta}_{it}$ is the error term. If beta is a complete measure of risk, there should not be any other factors that influence the return and γ_3 should be zero. The coefficient γ_1 is related to the third hypothesis that higher risk should be associated with higher returns, which means that this value should be positive. According to their research γ_1 is indeed the only significant coefficient (E. Fama and Macbeth, 1973).

2.1 Factor models

Factor models are models that describe the expected return of assets. According to the CAPM and the regression that Fama and Macbeth made, beta should be the only risk factor. However, in reality, this is not the case.

In 1992 Fama and French wrote a paper about the cross-section of stock returns. In this paper, they checked if beta is in fact related to stock returns and if there are company characteristics other than beta that can predict stock returns. The three company characteristics that Fama and Macbeth tested were: beta, book-to-market and size. As explained more elaborately in the previous section, beta is the market risk. Book-to-market is the book value of a company divided by its market value, while the size is the number of outstanding stocks times the price of the stock.

It becomes clear that there is indeed a relationship between size and return. The smaller sized portfolios perform better compared to the sizeable companies. However, there is no clear positive relation between beta and size. Furthermore, it becomes clear that portfolios with a high book-to-market have higher average monthly returns compared to the portfolios with a low book-to-market. However, the beta is not significant, which would mean that it does not influence the return of an asset. Therefore, the CAPM is rejected according to this model (E. F. Fama and French, 1992).

Fama and French try to save the β by suggesting that a sampling error causes the variation in the β. They believe that it is too early to throw the β away completely. Fama and French come up with their three-factor model as an alternative to the CAPM model. This adjusted three-factor model looks as follows:

$$r_t = \alpha + \beta_1 RMRF_t + \beta_2 SMB_t + \beta_3 HML_t + \epsilon_t$$

In this function $RMRF_t$ is the β, SMB_t captures the size, and HML_t captures the book-to-market of the stock. If the three factors capture the variation of all stock returns, the intercept α should be zero. The reason for this is that the model then explains all return.

In 2015 Fama and French came up with two extra factors that explain return, namely profitability and investments (E. F. Fama and French, 2015). The other three factors stay the same. However, the added value of these two extra factors is questioned. For this reason, these two extra factors are not taken into consideration for the rest of this research.

2.2 Momentum

Jagadeesh and Titman (1993) check in their paper if past returns are related to future returns. Their paper includes the past returns of one to four quarters, called the lookback period. They keep the portfolio for one to four quarters as well and call this the holding period. Therefore, there are sixteen combinations between the lookback periods and the holding periods in total.

A portfolio that consists of stocks that have increased in the last period (winners' portfolio) significantly outperform a portfolio that consists of stocks that have decreased in the last period (losers portfolio) in fifteen out of the sixteen combinations. The difference between the winners and losers portfolio is the most for a holding period of three months and a lookback period of twelve months (Jegadeesh and Titman, 1993).

Therefore, it becomes clear that the factor momentum influences the returns as well. This means that the three factor model becomes the four factor model and looks as follows:

$$r_t = \alpha + \beta_1 RMRF_t + \beta_2 SMB_t + \beta_3 HML_t + \beta_4 MOM_t + \epsilon_t$$

2.3 Why these factors?

So far, the factors risk, size, book-to-market, momentum, profitability, and investments have been mentioned. However, in the academic literature, one can find hundreds more. These are all empirical/statistical findings, typically without much economic explanation. It is important to find an economic reason to know why the stock price moves the way it does. There are three main options for how factors can predict stock returns:

1. Data mining - This says that the statistical results exist because of coincidence and state that there is no real effect.
2. Risk factor - An economic risk factor is causing the expected return.
3. Characteristic - Investors want to buy the stock because of other reasons.

For investors, the statistical results of a factor must not be caused by data mining. Data mining means that the factor does not influence the return of a stock in reality. Investors can invest in risk factors if they are willing to face the risk. Investors should invest in characteristics because they can have a higher expected return without bearing extra risk. The six factors explained in this section all have an economic theory why these factors influence the stock's returns. For this reason, these are the six factors used to explain a stocks return in this research.

So what can be the economic reason that these six factors determine the return of a stock? The first factor in our model is risk, which is measured by beta. The reasoning behind this was already mentioned at the beginning of this section. Here, it was stated that when two portfolios have the same return, an investor will always choose the portfolio with the least amount of risk.

Therefore, higher risk should always be related to a higher return. The reasoning behind this is clear and seems logical. However, when looking at Fama and French's results, it does not seem to be a significant factor that explains return.

The second factor in our model is size. Size is measured by the number of outstanding stocks times the price of the stock. In *Section 2.1* it became clear that smaller companies have a higher expected return. A reason behind this might be that sizeable firms (like PGGM) are limited in how much they can invest in small-cap stocks because of the limited liquidity there is for these stocks. In the next section, liquidity will be explained more elaborately.

As a result, they mainly invest in large-cap stocks, although small-cap stocks might be more attractive concerning risk/return. Because of this, large-cap stocks are overpriced, and small-cap stocks are under-priced. Expected returns of small-cap stocks are, therefore, more extensive than those of large-cap stocks. This means that there is a misallocation of capital that is affecting market prices.

The third factor in our model is book-to-market. Book-to-market is the book value of a company divided by its market value. "A company's book value is calculated by looking at its historical cost or accounting value. A firm's market value is determined by its share price in the stock market, and the number of shares it has outstanding, which is its market capitalisation." (investopedia.com, 2020)

Companies with a higher book-to-market ratio have a higher expected return. The reason for this is that companies with a low book-to-market value invest much money in their company to expand. Therefore, these stocks do not have much profit and will not turn out much dividend. This means that companies with low book-to-market have a lower expected return.

This can also be shown by looking at the following formula that determine the book-to-market value:

$$\frac{B_t}{M_t} = \frac{B_t}{\sum_{\tau=1}^{\infty} E\left(Y_{t+\tau} - dB_{t+\tau}\right)/(1+r)^{\tau}}$$

In this formula the book value is given by B_t, while the total market capitalisation of a company is given by M_t. The total market capitalisation of a company is equal to the total sum of expected earnings, $\sum_{\tau=1}^{\infty} E\left(Y_{t+\tau}\right)$, minus the change in book value, $\sum_{\tau=1}^{\infty} E\left(dB_{t+\tau}\right)$. The change in book value is determined by the investments that the company made. Because if a company makes more investments, this company has less money available to pay out dividend. This means that the market value decreases. This value is divided by the discount rate r.

This formula clarifies that when book-to-market increases while expected earnings and book value stay constant, the discount rate r has to increase. The discount rate is equal to the required return and an increase in the discount rate means that the return has to increase. Therefore, a higher book-to-market ratio has a positive influence on the return.

The fourth factor in our model is momentum. Momentum is measured by how well the stock performed over the last period T. Momentum is a significant factor for the return of a stock (Jegadeesh and Titman, 1993). A reason behind this is that a stock that is increasing in value, most likely means that the company is performing well.[3]

Therefore, it is very likely that this company knows what it is doing and will continue with this work. Because of this, likely, the returns of a company that performed well in the last period are higher than the returns of a company that did not perform so well last period.

3 Liquidity

The description in this section is loosely based on the paper that is written by De Jong and Rindi, 2009. Liquidity is the ease with which financial instruments can be traded. The liquidity of a stock is measured by the trader's ability to quickly trade this stock in the quantity he prefers at a low cost. Because large investment firms make sizeable trades that can significantly move a stock price, especially when liquidity is low, liquidity is one of an exchange's essential characteristics.

Because of this, liquidity has a significant influence on transaction costs. There are multiple dimensions of liquidity and multiple measures to calculate it. First, a quick look is taken at the market to see how liquidity can be measured here.

3.1 Markets

Two main market mechanisms exist in equity exchanges: order-driven markets and quote driven markets. There is a direct interaction between traders in an order-driven market, while in a quote driven market, there is an intermediate. In an order-driven market, there is liquidity from a continuous flow of orders from market participants (De Jong and Rindi, 2009). However, when everyone wants to sell, and no one wants to buy a specific stock for whatever reason, the liquidity of a stock dries up.

The traders who still own this stock cannot sell the stock because no one wants to buy the stock. Stocks with low liquidity will experience such a dry up earlier than stocks with high liquidity. Because of this risk, sizeable investment companies do not prefer stocks with low liquidity.

This phenomenon causes liquidity to attract liquidity. Exchanges that have high liquidity attract sizeable investment funds. These sizeable investment funds trade significant volumes and thereby increase the liquidity of such an exchange.

A popular way to measure the liquidity of a stock is to look at its bid-ask spread. The bid-ask spread is the difference between the highest price that a buyer wants to pay for an asset (the bid) and the lowest price for which a seller wants to sell his asset (the ask). The bid-ask spread is a measure of the supply and demand of an asset. When the liquidity of a stock is low, the bid-ask spread is called high or wide, and when the liquidity of a stock is high, the bid-ask spread is said to be low or narrow.

[3] There can be other reasons why a stock is increasing value. If all of a sudden a lot of people decide to buy a certain stock for whatever reason, this stock's price will increase. This is what became clear with GameStop's stock. Many small investors decided to buy this stock via the internet forum *Reddit*. Because of this sudden increase in demand for GameStop's stock, the stock price increased from 20$ to 480$ between the 12th of January and the 28th of January. After this sudden increase, the price dropped down again.

Another quick way to measure a stock's liquidity is by looking at the quoted volume of this stock. The quoted volume is the volume of all the buy and sell orders currently in the market. When the quoted volume is low, there will not be many buy and sell orders in the market. Both these indicators will be explained more elaborately later in this research.

A third method to measure liquidity is to look at the turnover or volume in the market. The volume is the amount of trading activity there is in the market.

3.1.1 Order-driven market

In an order-driven market, a trader can enter two types of orders: market orders and limit orders.

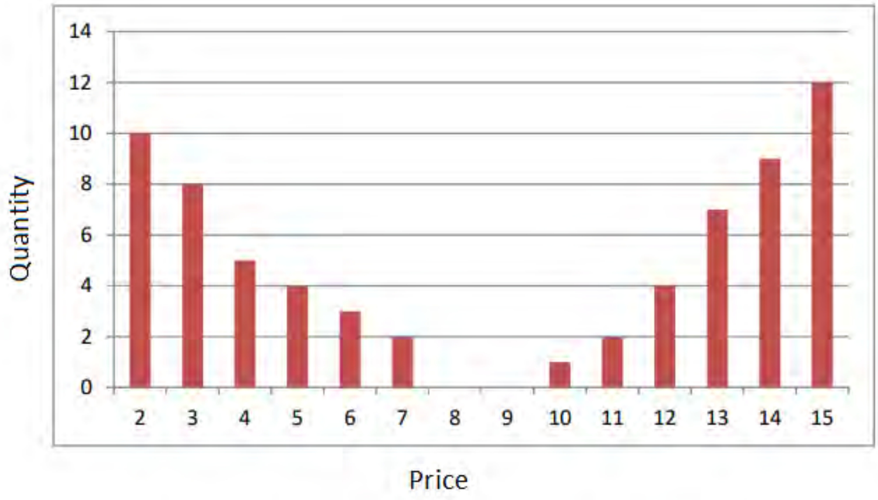
Definition 3. A limit order is an order where a trader chooses both the quantity and a maximum price (buy-order) or minimum price (sell-order) (De Jong and Rindi, 2009).

Definition 4. A market order is an order executed at the best price available at that moment (De Jong and Rindi, 2009).

Market orders are immediately executed, while limit orders are not. A market order buys (sells) the available stocks, for the lowest price available. A limit order will buy (sell) several stocks x for a price y. When a stock price is still too high (low), the order will stand until the price drops (goes up) to y. If this does not happen the limit order will not be executed. As long as there is sufficient supply and demand a market order will always be executed.

So with a market order, traders buy their stocks for an uncertain price at a particular time, while with a limit order traders buy stocks for a specific price at an uncertain time. In figure 2, an example of the order-driven market is given.

Figure 2: Example of an order driven market with buy-orders on the left side and sell-orders on the right side



Let us assume that an investor wants to buy five stocks in the order-driven market of the example given above in figure 2. He has two options to buy these five stocks. The first option is to execute a market order and the second option is to execute a limit order.

By executing a market order, he will buy the five cheapest stocks available on the market. This means that he will buy one stock for ten euros, two stocks for eleven euros and two stocks for twelve euros. In total, he will then thus pay fifty-six euros. After the market order is executed the limit sell orders of ten and eleven euros are gone, while the limit order of twelve euros has decreased from four to two. There are now fewer orders in the market, which implies that the quoted volume has decreased. If the quoted volume has decreased, this means that liquidity has decreased as well. The decrease in liquidity also becomes visible in the bid-ask spread that has decreased from 7-10 to 7-12.

Now assume that this investor executes a limit order of five stocks. He still wants to buy his stocks as quickly as possible, so he decides that he wants to have the buy-order with the highest price in the quote driven market at that moment. The investor then has to enter a limit order for five stocks for the price of eight euros. He now has to wait until someone wants to sell his stocks for a price of eight euros. When this happens, he will pay forty euros for these stocks. Until then, there are more orders in the market because of the buy-order that he entered into the market, which means that quoted volume has increased. Therefore, liquidity has increased as well. This is reflected by the bid-ask spread that has decreased from 7-10 to 8-10. This example demonstrates that a market order decreases liquidity, while a limit order increases liquidity.

These examples show that stocks with low liquidity exhibit a high bid-ask spread. These stocks will often have a low quoted volume which can quickly dry up. When a trader makes a sizeable buy- or sell-order for such a stock, the bid-ask spread can increase significantly.

Furthermore, for stocks that show high volatility, the price will fluctuate more compared to stocks with low volatility. Because of this, there will be more disagreement between traders about stocks with high volatility. Because of this high disagreement, the difference for which traders want to buy and sell increases and, therefore, the bid-ask spread increases. So both volatility as well as liquidity influence the bid-ask spread.

3.1.2 Quote-driven market

In a quote driven market, trades are made based on prices quoted by designated liquidity providers. These liquidity providers offer to buy and sell stocks themselves. Besides, they will buy and sell these stocks on the order-driven market. The liquidity providers will then hold these stocks until a trader wants to buy (some of) them for the quoted ask price. Therefore, the liquidity providers take positions themselves and carry a certain amount of risk. When the price of a stock that they hold all of a sudden decreases, they will lose money. Liquidity providers want to be compensated for running this risk.

The liquidity provider creates this profit with the difference between the prices for which he buys and sells his assets. This is the difference between the quoted sell- and buy-order, the bid-ask spread. As stated above, the bid-ask spread is also a measure of the market liquidity of a stock. When a liquidity provider is uncertain about the odds of selling a stock, he requires a higher return. Therefore, the bid-ask spread will increase.

There is more price uncertainty for stocks that have low liquidity. These stocks usually show low quoted volumes, and this can quickly dry up. Furthermore, stocks with high volatility have a higher probability of decreasing in value. Because of this, stocks that have low liquidity or high volatility (or both) will have a higher bid-ask spread compared to stocks with high liquidity and low volatility.

These are two examples of inventory control costs. The liquidity provider requires a reward for the risk that he cannot sell his stock or the risk for selling the stock for a lower price than anticipated.

Next to the inventory control costs, a liquidity provider has order processing costs. These are the costs of providing the service that traders can immediately buy and sell stocks to and from him.

Finally, there are adverse selection costs. These adverse selection costs exist because there are informed traders in the market. These informed traders have more information about the actual value of a particular stock than the liquidity provider. When this actual value is outside the bid-ask spread, these informed traders can take advantage of this by either buying, (when the actual value is above the bid-ask spread) or selling (when the actual value is below the bid-ask spread). Because of this, informed traders can take advantage of the position that the liquidity provider has when his bid-ask spread is not large enough.

These three kinds of costs given above influence the bid-ask spread's width and thus the return that the liquidity provider demands.

Let us look at the example of an order-driven market given in figure 2. Assume now that this stock is offered on a quote drive market. The liquidity provider offers a bid-ask spread of 6-11, meaning that a trader can buy the stock for eleven euros and sell the stock for six euros in this market. This seems to be a sizeable bid-ask spread and not very realistic, as the liquidity provider will make a revenue of $\frac{11-6}{6} = 83.33\%$.

However, the liquidity provider needs this revenue to guarantee liquidity for this stock to the traders. An investor that wants to buy five stocks can now buy all these five stocks for eleven euros. In total, he will thus pay fifty-five euros for these five stocks, which is one euro less than he would have paid in the order-driven market.

This difference would only increase when a trader wants to buy more stocks. When more traders decide to buy this stock, the liquidity provider can choose to decrease the bid-ask spread and pay more for the stock. The liquidity provider is now more confident of selling his stocks, which means that he might offer a bid-ask spread of 9-11 or 10-11 when the demand is high enough.

Of course, there is the possibility that this stock's fundamental value goes down when the stock has not performed well lately. In this case, there will be more sellers than buyers for this stock. The liquidity provider has to change his bid-ask spread to reflect this. Because a liquidity provider holds a lot of these stocks, he will most likely make a loss. To be compensated for this risk, he requires a return. Thus, this also means that the bid-ask spread of a risky stock will be higher than that of a less risky stock. The reason for this is that risky stocks have a higher probability of decreasing in value.

In a quote driven market, limit orders are not possible. Besides, a quote driven market is less transparent than an order-driven market. Another drawback is that the liquidity provider requires a profit, which makes it harder for traders to gain money themselves. However, the advantage of the quote-driven market is that the liquidity provider guarantees to trade a particular stock for a specific price. The liquidity provider thus guarantees liquidity.

3.2 Liquidity Dimensions

There are four different dimensions of liquidity:

1. The ability to convert stocks to cash without affecting the price (price impact dimension).

2. The ability to quickly convert stocks to cash (speed dimension).

3. The cost of converting a stock to cash (cost dimension).

4. The ability of a stock to quickly revert to its previous levels (price, bid-ask spread, etc.) after a sizeable trade is made. This is called the resiliency of a stock.

Because there are four dimensions of liquidity, there are also four ways of measuring the liquidity for these dimensions:

1. For the price impact dimension, it is possible to measure the illiquidity by dividing the absolute return on a particular day by the volume. This is called Amihud's ratio. The intuition behind this is that if a stock is very liquid and there is high volume on a specific day, the price will not be affected that much. The reason for this is that the order book will absorb the volume. If a stock is not very liquid, the price will be affected by large volumes on a specific day.

   Another way to measure the price impact dimension is to look at the quoted volume. By looking at figure 2 given in *Section 3.1*, it becomes clear that when the quoted volume is low, the price impact of a sell or buy order is high. In the example given earlier, a buy order of 5 stocks had a significant price impact, while this would not have been the case had the quoted volume been higher.

2. For the speed dimension, it is possible to measure the liquidity by looking at a fraction of time with zero returns. By counting the fraction of time for which there was no return, one knows the amount of time for which there was no trade.

   Another way to measure the speed dimension is to look at the traded volume or turnover for a specific period (Speed $= \frac{\text{Volume}}{\text{Time}}$). This is a measure of how much trading activity there is in the market.

3. For the cost dimension, it is possible to look at the bid-ask spread as a fraction of the price. How the bid-ask spread measures the cost dimension of liquidity is explained in the subsection above. Besides, by measuring this bid-ask spread as a fraction of the price, it is possible to see the actual effect of the bid-ask spread on the price.

4. Resiliency is measured by looking at how quickly the bid-ask spread goes back to its previous levels after a significant transaction is executed.

A liquid market is a market where traders can execute sizeable orders with minimal price impact almost instantly. When a sizeable order does have a price impact, prices will quickly mean-revert to their fundamental value in a liquid market.

Because of this phenomenon, liquidity is a risk factor. Investors would rather want a stock with high liquidity than a stock with low liquidity. This is especially the case for sizeable investments companies that execute significant volumes. Because of this, stocks with low liquidity will have higher expected returns.

4 Variables that determine transaction costs

In *Section 2*, a lot is explained about the factors that predict a stock's value. However, buying a stock also brings along costs. These costs are called transaction costs. The models described in *Section 2* do not consider these transaction costs but look at the development of the stock price.

These models do not consider transaction costs because there is much uncertainty about these costs, as transaction costs are not constant over time, and there is no uniform method to determine them. However, transaction costs can influence investors' returns massively, and an investor should most certainly not forget these costs.

Investors still have to bear transaction costs in the most liquid market. These transaction costs might be why the factor models explained before do not always work for sizeable investment companies, because transaction costs reduce the returns that a trader makes from his investments.

"Though a trading cost may be a small fraction of the value of the single transaction, over the long term horizon such expenses can significantly lower the return attained by application of the investment strategy, especially when a large number of purchases or sales is required" (Kociński, 2014).

Therefore, an investor must understand transaction costs, how to measure them, and how to trade to reduce these costs. These insights can significantly increase the net returns of an investor. Besides, exchanges are interested in these transaction costs to determine how liquid their market is. In this section, the theory behind these transaction costs is explained more elaborately.

Firstly, transaction costs are not known beforehand, but only after a transaction is executed. Therefore, it is hard for companies to consider these transaction costs when they make their investment decisions. Companies know that there will be transaction costs when they make a transaction, but they do not know the magnitude. Because of this, companies are more reluctant in changing their portfolios.

Secondly, it is good to consider the cases of buying and selling assets separately. In the case of buying stocks, transaction costs are the difference between the amount spent on these stocks and the market value these stocks had right before the purchase. In the case of selling stocks, the transaction costs are the difference between the stocks' market value right before the stock is sold and the amount of money that is obtained from these stocks.

Furthermore, sizeable investment companies will avoid small-sized companies and low liquidity companies when choosing their portfolios. The reason for this is that sizeable investment companies make significant transactions, and these significant transactions influence the price of the stock of small-sized companies and low liquidity companies more than the price of the stock of sizeable and high liquidity companies. Therefore, these companies expect a relation between a company's size and liquidity, and the transaction costs included when buying this stock.

Four significant sources determine the total transaction costs of a stock. These four forms of transaction costs will be explained in the following subsections.

4.1 Commissions & taxes

An order is always executed via a brokerage or a liquidity provider. These institutions will ask for a commission for this service. The commissions that these liquidity providers set are often negotiable. Because financial institutions execute many trades at a liquidity provider, these companies can negotiate low fees.

These financial institutions make sizeable orders at a liquidity provider. These sizeable orders are often split up into multiple smaller orders. Otherwise, the market impact will be enormous, which will hurt the return of such a financial institution. A financial institution can choose to execute these smaller orders themselves by researching the market and buying stocks when liquidity is high and the price is low.

Another option is to let the liquidity provider execute this order. A liquidity provider will ask for a different form of commission for this, called research commission. These are costs that a liquidity provider asks for researching about one or more stocks.

Some countries have taxes for trading in stocks. These taxes have to be paid when a stock in such a country is traded. Therefore, these costs are extra transaction costs.

4.2 Bid-ask spread

As stated in *Section 3.1*, the bid-ask spread is the difference between the quoted bid and sell order. Therefore, it is the difference between the price for which a trader can buy a stock and sell a stock. Let us look at figure 2 given earlier. The bid-ask spread for this example is three. The highest sell order is namely ten, while the highest buy order is seven. This means that if a trader buys this stock and then immediately sells it, he loses three euros. Before making a profit, the stock has to increase in value by three euros. For this reason, the bid-ask spread is also considered as a part of transaction costs.

Besides, a liquidity provider generates his profit via this bid-ask spread. As explained more elaborately in *Section 3*, a trader pays this bid-ask spread for the liquidity that the liquidity provider supplies. When buying at asking prices or selling at bidding prices, traders pay the bid-ask spread. As such, the market spread is an evident and essential part of the trading cost.

In *Section 3* it was stated that stocks with less market liquidity or higher volatility have a more significant spread. As already explained in *Section 2* investors require extra return for bearing more risk. A higher bid-ask spread means more transaction costs. So on one side, stocks with higher volatility will have higher returns, because investors require an extra return for the extra risk they bear for holding these stocks. On the other side, stocks with higher volatility will have a higher bid-ask spread, which means that they will have higher transaction costs. Therefore, higher volatility causes a higher expected return on the one hand and a lower expected return on the other hand. This is something investment managers of sizeable funds certainly have to take into consideration.

4.3 Market impact

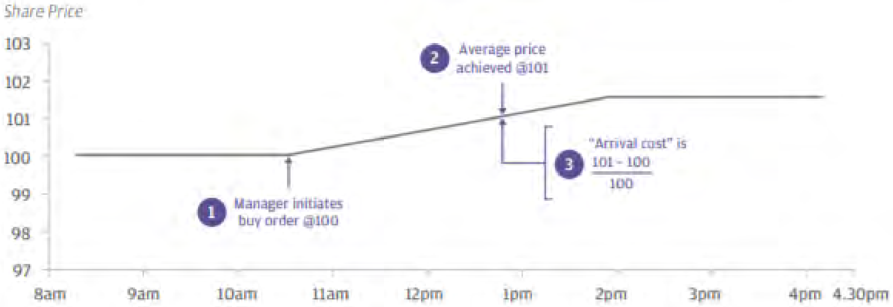
The price change measures the market impact that this transaction causes. In other words, it is the difference between the price of the stock after the transaction and the price of the stock had the transaction not taken place. A buy-order will drive a stock up, while a sell-order will drive the price of a stock down. This is because executing a buy-order means that the stock demand increases while executing a sell-order means that the stock supply increases.

Because it is impossible to know the price of the stock had the transaction not taken place, the market impact of a transaction is calculated by executing the following three steps:

1. Looking at the asset's price at the exact time that the order is executed (arrival price).
2. Looking at the average price paid for the whole order (execution price).[4]
3. Calculating the difference between the average execution price and the arrival price (arrival cost).

These arrival costs are measured by "the difference between the price at which an asset is valued immediately before an order and the price at which it is actually traded" (jpmorgen.com, 2018). In figure 3 below is shown how these arrival costs are measured.

Figure 3: Example of how the arrival costs of a buy-order are calculated (jpmorgen.com, 2018)



In this figure, a manager initiates a buy order at 100 euros for a particular stock. The order is initiated around 10:45 AM and continues until 2 PM that day. After the order is completed, the stock price is around 101.5 euros. The order started at 100 euros, and the average price paid for this order is 101 euros. The arrival cost is then calculated in the following manner: $\frac{101-100}{100} = 1\%$.

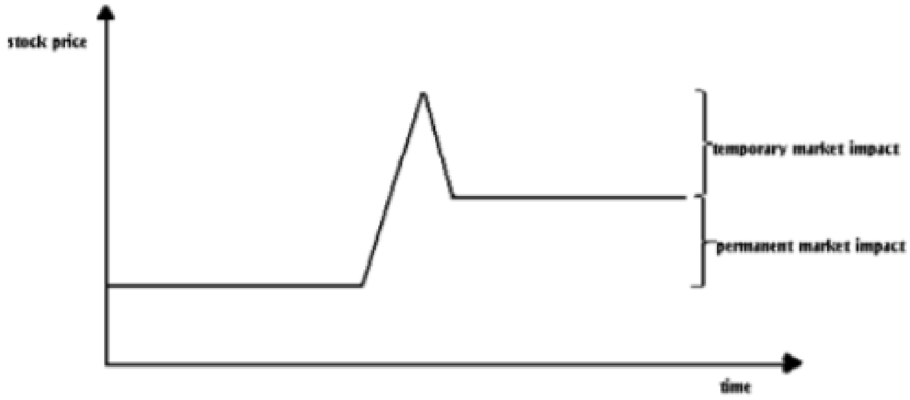
There are two primary forms of market impact: temporary impact and permanent impact. These two effects combined form the total market impact.

Temporary market impact is the temporary price movement of an asset away from its equilibrium (fundamental value). After the trade is executed, the price will go back to its equilibrium. Therefore, the temporary price impact only affects this transaction. The temporary impact is affected by the way that the order is executed. An order executed over a short period will have a more significant temporary market impact than a trade executed over a more extended period. Permanent market impact means a change in fundamental value caused by an order. When this happens the equilibrium shifts.

The market impact of a buy-order is visible in figure 4 given below. In this figure, there is a distinction between a temporary market impact and a permanent market impact.

[4] If the order is split into multiple transactions, all transactions will be taken into consideration.

Figure 4: Market impact of a buy-order (Kociński, 2014)



In this figure, it is clear that the price of the stock suddenly increases. This price increase happens after the buy-order is executed. After some time, the stock price mean-reverts to its fundamental value. However, this fundamental value is higher than the value the stock had before the transaction was made. There has thus been a permanent price shift caused by the permanent market impact of the order.

As stated, the temporary market impact of an order only affects that specific transaction. This is money that a trader loses when making the transaction. Permanent market impact changes the fundamental value of a stock, which means that the price of a stock has increased (or decreased) permanently. This money is, therefore, not necessarily lost. Let us look once more at figure 4. If the trader that executed this buy-order decides to sell the stock immediately after he bought it, he can sell it for a higher price than for which he bought the stock because of the permanent market impact. A trader, therefore, prefers a permanent market impact over a temporary market impact.

Sizeable investment companies have to realize this as well. A considerable buy- or sell-order can influence the value of stock permanently, which influences their return as well. The market impact for small and low liquidity assets will be higher than the market impact for sizeable and high liquidity stocks. Because of this, sizeable investment companies will not buy small and low liquidity stocks. Therefore, these stocks are undervalued. This is why small size stocks have a higher expected return than sizeable size stocks, as explained in *Section 2.1*.

Next to this, sizeable investment companies have to consider the extra transaction costs for high volatility stocks. Therefore, the three main stock factors that might influence the transaction costs are size, liquidity, and volatility. Next to these three factors, the transaction's market impact is hugely dependent on how the order is carried out.

In the previous section, market orders and limit orders were explained. A market order is immediately executed, while a limit order is executed over a longer period. As explained in *Section 3.1*, a limit order is only executed when there is a counterparty that wants to sell (buy) this stock for the price of that buy-order (sell-order). A limit order, therefore, provides liquidity, while a market order takes liquidity. "Market impact pertains to the costs incurred by extracting liquidity from the market in order to acquire or dispose of a position." (Ferraris, 2008)

Because a market order is executed at once and this order takes liquidity away from the market, while a limit order is executed over a more extended period and provides liquidity, the market impact of a market order is higher than that of a limit order.

Another option to not create a considerable price impact is to split the market order into smaller transactions, called placements, executed over a more extended period. This will reduce the impact the order has on the market. However, this brings in the risk that prices increase (decrease when you want to sell) and that the opportunity of taking advantage of the pre-transaction price, on which you based your decision to buy or sell, is gone.

It becomes clear that there is an interaction between the first two dimensions of liquidity explained in *Section 3.2*. An order can be executed at once, which is good for the speed dimension and bad for the price impact dimension. On the other hand, it is possible to execute the order over a more extended period, which is good for the price impact dimension and bad for the speed dimension.

It can also occur that an investor gains a profit by delaying his order. This is the case when the price of a stock decreases (increases when you want to sell), such that it is possible to buy this stock for less. If this happens, the order has had a negative market impact. It is then thus possible for a trader to have negative transaction costs.

As stated at the beginning of this subsection, a buy-order (sell-order) will always increase the demand (supply) and will therefore drive the price of a stock up (down). Therefore, an order can never hurt the market by itself. However, if there are more sell orders than buy orders, the price can go down. If this happened when a buy-order was being executed, this order can have a negative market impact and therefore have negative transaction costs.[5]

By doing the opposite of most traders in the market, a trader provides liquidity to these other traders. He is then rewarded for providing liquidity by receiving a negative market impact. This trader then hopes that the number of buy orders increases again after his buy order is executed, because this means that the stock's price will increase in value once again.

As explained in *Section 3.1*, liquidity can be measured by looking at the quoted volume of a stock. The quoted volume are all the buy and sell orders that are currently in the market. Two variables combine the size of the order and the market's liquidity to show the relation between these two components. These variables are the participation rate and the average daily volume in percentages.

Definition 5. The participation rate shows how big a trader's order is compared to the rest of the market. This rate is determined by dividing the size of the trader's order by the size of all orders that were executed during the trader's execution. When making an order at a brokerage, a trader can choose the participation rate for which his order is executed.

Definition 6. The average daily volume in percentages is the size of the order divided by average daily trading volume. The average daily trading volume is the average number of stocks traded in a day. This is, therefore, a good measure of liquidity.

[5] The order itself did not hurt the market. This is not possible as the demand (supply) always increases when a buy order (sell order) is submitted. However, it is plausible there were more sell orders (buy orders) than buy orders (sell orders) during the time that the buy order (sell order) was executed, and that the price moved down (up) in this period. This makes it possible for a trader to have a negative market impact on his order.

One can imagine that getting a good participation rate is a crucial aspect of minimizing the market impact. Getting the perfect participation rate "involves matching the speed of trading a stock to the speed at which its price moves" (Yegerman, 2020).

A lower participation rate means that the order is spread over a more extended period. This order is then a less significant part of the total number of trades executed during this time. Because of this, the impact of the order on the market will be lower as well. The impact of an order on the market, namely, exponentially increases in combination with the participation rate.

However, by spreading his order over a more extended period, the trader is more at risk for fluctuations in the stock price. The duration of an order can be estimated by executing the following equation: $\frac{\text{ADV}}{\text{participation\%}}$

This shows how much the average daily volume, participation rate, and duration are intertwined. A trader has to think about how the stock price will move in the near future when he decides the participation rate. Because market impact is affected by other traders and how the stock price moves, this is the most significant question mark for transaction costs. If an investment company can accurately predict the market impact, this can make a difference of millions of euros. The main focus of this research will, therefore, be on the market impact.

4.4 Opportunity costs

Missed trade opportunity costs are the costs that arise when an order is not executed. These costs are calculated by looking at the price movement between the time that the broker receives the order and the time that the order was rejected. The non-execution results in opportunity costs due to lost profits.

4.5 Measuring the transaction costs

The total transaction costs can be measured as the difference between the value of a paper portfolio and the value of an actual portfolio. This paper value will assume that a trader paid the decision time mid-point of the best bid- and ask prices, which means that the paper value assumes that the trader will pay (receive) the middle point between the lowest buy-order and the highest sell-order, at the time the trader made his decision.

The value of the actual portfolio is the amount that a trader paid for the portfolio. This actual portfolio includes the commissions, bid-ask spread, price impact, opportunity costs, and other factors that influence the price.

Transaction costs consist of two parts, explicit costs and implicit costs. Explicit costs consist of the commissions and taxes that were explained above in *Section 4.1*, while implicit costs consist of the bid-ask spread, market impact and opportunity costs that were explained in *Sections 4.2, 4.3 and 4.4*. How implicit and explicit transaction costs are measured is explained in the subsections below.

4.5.1 Explicit transaction costs

Explicit costs are costs that are not as hard to measure as implicit transaction costs. These explicit transaction costs represent a part of the order that is visible. These costs can be determined before an order is executed.

However, these explicit costs are harder to calculate than it looks at first glance. Brokerage commissions are namely often paid for bundled services and not just for the execution of an order. Explicit transaction costs can be split into four sections. These four sections are shown below:

1. Costs for buying and selling stocks (broker commission).
2. Costs for the research the asset manager has to do (research commission). These costs only apply to the orders that the broker executes. When an investor executes his order, these costs do not apply.
3. Taxes and levies that are needed to execute an order. This is also called stamp duty. The cost of these taxes and levies differ per country in which the stock is traded. Many countries do not have this stamp duty.
4. Costs for borrowing money or the admin fee required for lending (stocks lending). These are costs that are required when a trader makes a short sale.

These explicit costs are precise and easily measured. To calculate the total sum of explicit costs, the costs of the points given above need to be added up.

4.5.2 Implicit transaction costs

Implicit costs are costs that are hard to measure. These costs include the bid-ask spread, the market impact and the opportunity costs. From these three costs, market impact is the most complicated to measure. To know an order's exact impact on the market, one needs to estimate the price if the transaction had not taken place, which is impossible to measure. As a replacement, the price that the stock had before the transaction is taken.

Six different benchmarks can be used to calculate the implicit transaction costs. These six different forms will all result in a different result for the implicit costs. The general form of the formula that can calculate these implicit transaction costs is given by:

$$\text{Implicit Transaction Costs} = x_j \cdot d_j \cdot (p_j - b_j)$$

In this formula, x_j is the size of the order, d_j is the direction of the trade, p_j is the total price paid for this order, and b_j is the benchmark that is used to calculate the implicit transaction costs. In this formula, d_j has a value of one when the order was a buy order and a value of negative one when the trade was a sell order.

Many benchmarks b_j can be used to measure the implicit transaction costs. The six most important ones are discussed below:

1. The time-weighted average price (TWAP). This is the average transaction price in a given day. It is calculated by dividing the sum of all transactions by the number of transactions made that day. A benefit of this benchmark is that it is easy to measure and calculate. However, a con for this benchmark is that it does not include the size of the orders that day.
2. The volume-weighted average price (VWAP). This is the trade-size weighted average price. It is calculated by multiplying each transaction's price by its size and then dividing this by the total volume that was traded for that day. The VWAP is an expansion of the TWAP

because it also includes the size of the made trades. This makes sure that more significant trades have a bigger influence on the benchmark. VWAP is an attractive benchmark for a trader. This benchmark allows a trader to see whether they paid a higher or lower price than the average order of that stock done that day.

3. The decision-time bid and ask price midpoint. This is the midpoint of the bid and ask at the time the order entered the market. Sizeable orders are often split into multiple smaller placements. The decision-time bid and ask price midpoint is then the sum of the difference between the midpoint of the bid and ask of all individual placements and the midpoint of the bid and ask at the time the order entered the market.

4. The price of the last transaction of the day. This is called the closing price. A benefit of this benchmark is that closing prices are easily obtained.

5. The average of the lowest, highest, opening and closing prices (LHOC). The benefit of this benchmark is that it is not hard to calculate. However, a drawback of this method is that it depends for fifty per cent on the opening and closing prices, while these prices are by no means relevant in all cases.

6. Midpoint of the bid and ask price at the time of the trade. When this midpoint is calculated before the execution of an order, this is called the one-way effective spread. When this midpoint is calculated after the execution of an order, this is called the realized spread. The benefit of this benchmark is that it is simple to interpret. To calculate this value, a trader has to take half the value of the bid-ask spread. Therefore, this is an indication of the price that a trader pays because of the bid-ask spread. However, a con of this benchmark is that this method does not indicate whether the trade was well-timed.

The exact implicit transaction costs are only known after the completion of an order. It is then possible to calculate the exact difference between the amount that the trader paid for the stocks and the stock price when the trader made his decision.

The six benchmarks that are given above are the main methods to estimate the implicit costs. None of these methods is right or wrong, but a trader must choose one of these methods when calculating the transaction costs. All benchmarks have their benefits and cons. Different companies and even different departments in the same companies use different benchmarks. These different benchmarks will all give different estimates for the transaction costs. This means that the benchmark the trader chooses to calculate his implicit transaction costs will influence his trading strategy.

The third benchmark, the decision-time bid and ask price midpoint, seems to be the most comprehensive method out of these six. This is the benchmark that PGGM uses most. The decision-time bid and ask price midpoint takes the difference between the price of a stock when a trader made his decision and the stock's price after completion of these orders. This benchmarks thus captures the difference between the price that a trader was expecting to pay and the price that he paid, and it gives a good indication of whether the order was well-timed.

If this benchmark is combined with the opportunity costs for the part of the order that is not completed, the total transaction costs become more clear. The method that combines the decision-time bid and ask price midpoint and the opportunity cost is called the implementation shortfall.

4.5.3 Implementation shortfall

An essential method that captures most of the transaction costs is the implementation shortfall. This method adds the opportunity costs to the implicit transaction costs that use the decision-time bid and ask price midpoint as a benchmark.

There are four main components that the implementation shortfall captures. These four components are given below:

1. First of all, the implementation shortfall captures the cost there is due to delay between deciding to trade and the order's arrival in the market (delay cost). To calculate the delay cost, the decision-time bid and ask price midpoint are required when the order arrived in the market and for the time that the trader decided to send the order to the market. The delay cost is then thus calculated in the following manner:

$$\text{Delay Cost} = x_j \cdot d_j \cdot ((o_0 - m_0) - (o_d - m_d))$$

2. Secondly, the implementation shortfall captures the cost due to change in the midpoint between arrival time and execution time (Change in midpoint cost). This change in the midpoint captures the implicit transaction costs and uses the third benchmark discussed in *Section 4.5.2*. To calculate the change in midpoint cost, the decision-time bid and ask price midpoint are required when the order arrived in the market and the bid and ask price midpoint after completion of the order. The change in midpoint cost is then thus calculated in the following manner:

$$\text{Change In Midpoint Cost} = x_j \cdot d_j \cdot ((o_j - m_j) - (o_0 - m_0))$$

3. Thirdly, the implementation shortfall captures the effective spread. The effective spread is the midpoint of the bid and ask price at the time of the trade. The effective spread is calculated in the following manner:

$$\text{Effective Spread Cost} = x_j \cdot d_j \cdot (p_j - (o_j - m_j))$$

4. Finally, the implementation shortfall captures the loss that is caused when a part of the order is not executed (opportunity cost).

$$\text{Opportunity Cost} = (x_j - y_j) \cdot d_j \cdot (p_n - (o_0 - m_0))$$

In these formulas x_j is the size of the trade, d_j is the direction of the trade, o_0 is the price of the order at the time that the order arrived in the market, o_d is the price of the order at the time that the order was made, o_j is the price of the order at the time that the order was executed, m_0 is the midpoint in the bid-ask spread at the time that the order arrived in the market, m_d is the midpoint in the bid-ask spread at the time that the order was made, m_j is the midpoint in the bid-ask spread at the time that the order was executed, p_j is the total price that was paid for this trade, y_j is the total quantity that is traded, and p_n is the price of the last order that day.

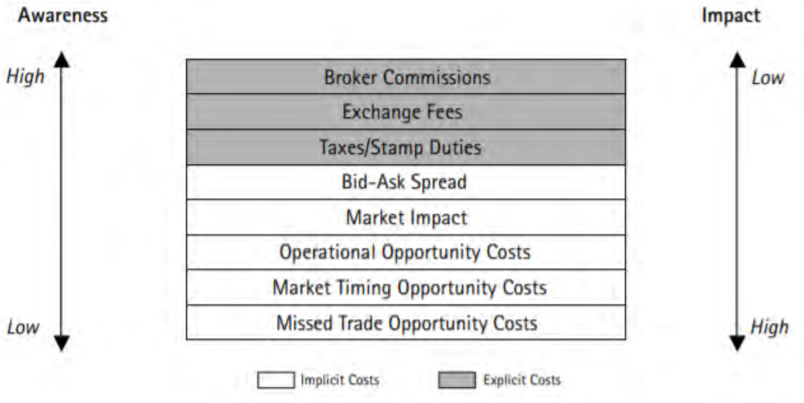
The implementation shortfall is then given by the sum of these four components. For the implementation shortfall, this means that the effective spread cost (sixth benchmark) and the opportunity cost are added to the third benchmark explained in *Section 4.5.2*. To calculate the opportunity costs, one needs to know which part of the order was not executed.

4.6 Total transaction costs

After the implementation shortfall is calculated, the total transaction costs can be calculated by adding the explicit transaction costs. The total transaction costs is then given by adding the explicit transaction costs together with the implementation shortfall.

Figure 5 below shows the essential sources of transaction costs. In this figure, the explicit costs are in grey, and the implicit costs are in white.

Figure 5: Sources of transaction costs (Giraud and d'Hondt, 2008)



The left side of figure 5 shows the average companies' awareness of these different sources. This figure shows the explicit costs at the top, and the implicit costs follow. The right side of figure 5 shows the extent to which these sources impact the total transaction costs. The implicit costs are now at the top, while the explicit costs follow.

In this figure, it becomes clear that the sources with the highest impact have the lowest awareness. A reason for this is that the implicit costs are easier to see and calculate for traders. This makes them more aware of these costs. The explicit costs are harder to calculate and more vague. However, these costs have a higher impact on transaction costs and are, therefore, more critical. A side note to this figure is that it is from 2008, and in the meantime, the awareness for implicit costs has increased.

The main focus of this research will be on the market impact. In this section, it became clear that the market impact is calculated by taking the difference between the average execution price and the arrival price. As was visible in figure 3, the average execution price is dependent on how the price of a stock moves.

Therefore, market impact is for the most significant part captured by the change in midpoint cost. The market impact fulfils the most significant part of the variable transaction costs. As these costs are variable, a trader can do something about them and find ways to lower these costs.

Because of this it is very important for PGGM to get information about why the market impact had its value. Understanding this can help PGGM lower their transaction costs. This is also the reason that the main focus of this research will be on the market impact.

5 Data

Bloomberg and FactSet provide all data in this research. Bloomberg is a sizeable company that provides financial services to many clients. PGGM uses Bloomberg to do financial analyses and estimate the market impact their orders will encounter.

Furthermore, Bloomberg can provide relevant stock variables and can give the market impact of an order. In this case, all these relevant variables are used to understand why the market impact has its value, and these variables are used to predict the market impact.

FactSet is a financial data and software company that PGGM uses to get financial data. In this case, only the financial data of the orders that PGGM has made is relevant. Factset will be used to find variables of stocks that are not available in Bloomberg but can help predict and understand the market impact.

There were two data sets provided to us, one data set that contained PGGM's orders that were executed between November 2016 and December 2017 and one data set that contained PGGM's orders between January 2018 and December 2019. Both data sets combined contain 18768 orders. There are forty-four variables available in our data set. The description of these variables is given in tables 1 and 2 below.

Table 1: Description of the variables in the data set (part 1)

Variable	**Description**
OrigOrdIdPx	A unique ID for the in order as provided by external feed
Account Groups Firm	User defined groups of accounts shared at firm level
ISIN	ISIN of the security being traded
SecName	Name of the security
MarketCap	Total market capitalization (small, medium, large)
Side	This shows whether is is a buy (B) - or a sell-order (S)
Size	Total size of the order
Value	Total value of the order ?
Trader	Trader of the order (at PGGM)
Brkr	Broker who executed the order
Executed Venue Groups Global	Global grouping of executed venues??
Country	Country of the security
Type	Type of order that was executed (market, limit, etc.)
Reason	Aim reason code at order level
ReasonDesc	Displays the descriptive text defined for REASON CODE
PrevCloseDate	The date when the Previous Close Date benchmark price was taken
Duration	The time from order arrivel to the last fill of the order
MktImp	Estimated market impact
% Adv	The average daily volume in percentages.
Part %	The participation rate of the order.
10dAbsVol	Last 10 days of stock volatility absolute
5-day Average Bid-Ask Spread Ordinal	Ordinal of the bid/ask spread
OrderLifeMomentum	The percentage change in price of the underlying security during the interval of the order (order arrival to last fill) relative to the side
Fill Ratio	The percentage of the order that was executed
IC/Arrival bp	Difference between the average execution price and the mid price at the time that the order is received (given in basis points)
IC/Arrival sum	Difference between the average execution price and the mid price at the time that the order is received (given in a real value)
IC/MktImp Arrival bp	Difference between the expected market impact and the real market impact (given in basis points)
IC/MktImp Arrival sum	Difference between the expected market impact and the real market impact (given in a real value)
IC/Prev Close bp	Difference between the average execution price and the price of the stock at the end of the previous day (given in basis points)
IC/Prev Close sum	Difference between the average execution price and the price of the stock at the end of the previous day (given in a real value)
IC/Day Close bp	Difference between the average execution price and the price of the stock at the end of the day (given in basis points)
IC/Day Close sum	Difference between the average execution price and the price of the stock at the end of the day (given in a real value)

Table 2: Description of the variables in the data set (part 2)

Comm bp	Commissions paid to the brokerage (given in basis points)
Comm sum	Commissions paid to the brokerage (given in a real value)
CommFeeTax bp	All commissions paid including fees and taxes (given in bp)
CommFeeTax sum	All commissions paid including fees and taxes (given in a real value)
PrOp Px	Opening price the day before the order is received (for example, if the order is received on Tuesday, it would be Monday's open)
PrCl Px	Closing price the day before the order is received (for example, if the order is received on Tuesday, it would be Monday's close)
LMOAP	The percentage change between the open price of the first day of the order and the order arrival price.
LM2dOpArr	The percentage change between the open 2 days before the order and the order arrival price
LM3dOpArr	The percentage change between the open 3 days before the order and the order arrival price
LM4dOpArr	The percentage change between the open 4 days before the order and the order arrival price
LMWkArr	The percentage change between the open 1 week before the order and the order arrival price
LMMoArr	The percentage change between the open 1 month before the order and the order arrival price

In tables 1 and 2 all variables in our data set are described. In total, there are forty-four variables in this data set. Some of these variables need a bit more explanation, which is given here.

First of all, there is the user-defined group of accounts shared at firm level. This variable gives information about the type of stocks in which PGGM invests. There are three main types: LRE, BOA and DMAE. LRE stands for listed real estate, BOA stands for invest for solution shares,[6] and DMAE stands for developed markets alternative equities.

ISIN is an abbreviation of International Securities Identification Number. This is a unique code that consists of twelve characters. The ISIN identifies the security.

The estimated market impact is an estimate that Bloomberg gives before the execution of an order. This variable gives the market impact Bloomberg expects an order will have, according to their model. This model is not known by PGGM, because Bloomberg wants to keep this information to itself.

However, PGGM assumes that Bloomberg bases this model on the following four components: risk, momentum, participation rate and liquidity. These four components were all given in section 4. The estimated market impact is given as an expected cost of the order. So when a trader executes a buy order for a security that has a price of one hundred euros but due to the size of his order he pays one euro extra, the market impact will be minus one per cent.

As explained in section 4.3, a buy order will always increase the price of a stock, while a sell

[6] This is translated from Dutch (Beleggen voor Oplossen Aandelen).

order will always decrease a stock price. The value of the Bloomberg expected market impact variable will, therefore, always be negative. However, in this same section, it became clear that this was not always true. It is, therefore, sensible to compare the Bloomberg market impact estimate with the actual market impact. This comparison is made in the next section.

The average daily volume in percentages is given by dividing the order's size by the average daily volume. A sizeable investment company often executes significant orders that are higher than the average daily volume. Therefore, the average daily volume may be above 100%. The variable % ADV is thus a display of $\frac{Size}{ADV}$. This means that when % ADV is mentioned in the rest of this research, this is about $\frac{Size}{ADV}$.

The order's participation rate is a measure of the liquidity of this stock and a measure of how fast this order was executed. This is measured by the total number of traded stocks in this order, divided by how many other stocks were traded during the execution of this order. Definitions of the participation rate and the average daily volume were already given in *Section 4.3*.

The last 10 days of stock volatility absolute gives information about how volatile a stock is based on the last ten trading days before the trade was executed.

The ordinal of the bid-ask spread gives information on how large the average bid-ask spread was in the five days before the order was executed, given in basis points. This variable consists of two values and can have the following values: 0-5, 5-10, 10-20, 20-50, 50-100, 100-200, 200+. Therefore, this variable gives information about the value between which the bid-ask spread lied five days before the order was executed.

The percentage change in the price of the underlying security during the interval of the order, relative to the side (momentum), gives information on how the security price moved during the time that the trade was executed. Therefore, this variable gives information about the change in midpoint cost during the execution of the order.

As explained in section 4.3, it is profitable to buy a stock when the stock's price decreases during the time that the order is executed.

In this case, a positive value means that the stock was moving in a favourable direction during the trade execution. This variable is a big part of the market impact, and the momentum variable will, therefore, have a high correlation with the actual market impact variable.

The difference between the average execution price and the mid-price when the order is received gives the actual market impact of an order that uses the midpoint of the bid and ask price as a benchmark to calculate the arrival price. This benchmark is equal to the third benchmark that was discussed in *Section 4.5.2*.

There are also other variables in the data set that indicate the market impact, but this is the most prominently used variable by PGGM to indicate the market impact. The value in this variable is given in basis points and a real value. A basis point is 0.01%.

This variable gives the order's actual market impact, and because of this, this variable is essential in the data set. This is the variable that needs to be predicted. It is possible to have a positive value in the actual market impact. A positive market impact can occur when the price of a stock moved favourably during the order's execution for this stock.

Furthermore, there is a difference between the expected market impact and the real market impact. This difference should be the same as $IC/Arrivalbp - 100 \cdot MktImp$. The market impact has to be multiplied by one hundred because it is given in percentages and not in basis points.

This column's value is around the same value as is given by the calculation above, but not exactly. This is probably due to rounding errors. The expected market impact is given with two decimals, just as the actual market impact and the difference. However, the expected market impact is given in percentages, and when this is multiplied by one hundred to transform the expected market impact to basis points, there are no decimals left.

The difference between the average execution price and the stock price at the end of the previous day is another benchmark to calculate an order's total market impact. This difference is given in basis points and a real value. As explained in *Section 4.5.2*, multiple benchmarks can be used to calculate the implicit transaction costs. One benchmark is the last transaction of the previous day (closing price).

The percentage change between the open price of the first day of the order and the order arrival price is the percentage difference in the price of the underlying stock between the opening price on that day and the price at which the order was executed. Positive numbers mean the stock was moving in a favourable direction (buying a stock that was falling, selling a stock that was rising). This is the same for the variables with two days, three days, four days, one week, and one month.

A disadvantage of this variable is the fact that it has a lot of zeroes, which will make it harder to use this variable in a model. One explanation for this is that this variable will have a value of zero when it is bought when the market opens because then the price has not moved yet. Only 4755 values in this variable contained applicable information. All other values were zero. Because there are many missing variables, there might be more information in the other price difference variables.

6 The difference between the actual market impact and the estimated market impact

The critical aspect of this research is to predict the change in midpoint cost during the execution of an order and to find a way to lower this change in midpoint cost for PGGM. The change in midpoint cost during the execution of an order captures the price difference this stock experiences during the order's execution. This is the most significant part of the market impact.

PGGM currently uses Bloomberg's market impact prediction as an estimation for the market impact when they execute an order and use this as a benchmark to see how well trades are executed. PGGM states that an order had under-performance if the actual market impact is higher than the predicted market impact and that an order had out-performance if the actual market impact is lower than the predicted market impact. Therefore, PGGM uses the predicted market impact to evaluate their trades.

However, it becomes clear that the actual market impact significantly differs from Bloomberg's estimated market impact. It seems weird that a sizeable company like Bloomberg is not able to accurately predict the market impact. This shows us how hard it is to predict the market impact.

The first necessary step in our research is explaining the difference between the actual market impact and the estimated market impact that Bloomberg gives. Explaining this difference is of significant importance. By understanding why the market impact is different from its expected value PGGM will be able to change its order strategy to lower its market impact.

Namely, if the market impact is under-estimated, PGGM will trade too fast with a higher market impact as a result. However, if the market impact is over-estimated, PGGM will trade too slowly, which increases the risk that the price moves against them (Ferraris, 2008).

This also means that by creating a model that predicts the market impact closer to the actual market impact (compared to the Bloomberg model), it is possible to help PGGM.

6.1 First Bloomberg market impact model

PGGM has no information about the current model Bloomberg uses to predict the market impact. Therefore, the historic Bloomberg models are evaluated. One of the first known Bloomberg models that was used to predict the market impact looks as follows (Ferraris, 2008):

$$\text{Market Impact} = \frac{S}{2P} + \sqrt{\frac{\sigma^2/3}{250}} \cdot \sqrt{\frac{V}{0.3 \cdot EDV}}$$

In this model $\frac{S}{P}$ is a measure of the bid-ask spread, σ^2 is a measure of the volatility and $\frac{V}{EDV}$ is a measure of the relative order size.

6.2 Second Bloomberg market impact model

So how is it possible that such a sizeable company like Bloomberg (or any other company) cannot correctly predict the market impact? In 2012 Bloomberg said that they have introduced "a new highly accurate pre-trade cost model with predictive power (R^2) of up to 26%" (Rashkovich and Verma, 2012).

The R-squared is a measure that gives information about how much of a dependent variable's variance is explained by the model's independent variables. If the R-squared has a value of one, the model completely explains the dependent variable's variance. This means that the model works well and accurately predicts the market impact. A model that has a R-squared value of twenty-six per cent is usually considered substandard. However, Bloomberg states that they are delighted with their new model. This shows once more how hard it is to predict the market impact.

In their new model, Bloomberg looks at the predictive power it can get from a transaction cost model. Their model splits the job of a trader into two tasks: Liquidity sourcing and momentum management.

Momentum is the stock price movement during the execution of the order. A trader can manage momentum by assessing the preference of trading conditions and making decisions about the order's speed. A trader can source liquidity by finding the other side of the order (sellers when buying) based on the desired execution speed.

In their research, Bloomberg states that roughly 83% of the market impact is accounted for by momentum management and that liquidity sourcing is responsible for the remaining 17%. The current transaction cost models only take liquidity sourcing into account and do not consider momentum management. The reason for this is that it is too hard to capture the price movement during the execution of an order. Because transaction cost models only take liquidity sourcing into account, these models are trying to estimate a weak signal. As a result, it is expected that a market impact model has a predictive power of no more than 17%.

Bloomberg splits the transaction costs into three components with their new method: instant impact, temporary impact, and market impact. The immediate impact is the impact that directly occurs when an order is executed. This is the price paid because of the bid-ask spread.

In their research, Bloomberg finds out that more aggressive strategies, with higher participation rate, have a higher instant impact. As stated in *Section 4.5.3*, the implementation shortfall is calculated by subtracting the arrival price from the average execution price. The arrival price is in this case measured by taking the midpoint between the bid and ask price at the moment that the broker received the order.

If the execution strategy is not aggressive, traders can stay on their side of the spread (bid or ask) and not come to the midpoint. If this happens, the instant impact is negative. In their research, Bloomberg states that orders with a participation rate up to 5% on average only cross a quarter of the spread, while trades with a participation rate higher than 30% fully cross the spread. The instant impact can then be expressed with the following formula:

$$\text{Instant Impact} = \lambda \cdot \text{Bid-Ask Spread \%}$$

The value of λ is given in table 3. This means that with a participation rate of 13.33%, the instant impact factor is zero. To ensure this factor's stability, Bloomberg has used the bid-ask spread over the last five trading days.

The second component that Bloomberg investigates is the temporary impact. The duration of the order is added to the model to capture the temporary impact. Because, if the participation rate is the only used component, then "the temporary impact does not scale as the size grows" (Rashkovich and Verma, 2012).

As explained in *Section 3.1*, how much the price of a stock moves depends on the volatility of this stock. If the stock volatility is very low, and the stock price barely moves, then the costs would always be lowest with a maximum duration and a minimum participation rate. This means that volatility has to be added as a component to measure the temporary impact. The temporary impact is then expressed with the following formula:

$$\text{Temporary Impact} = \alpha \cdot \sigma \cdot (\frac{\text{Participation \%}}{100})^{\beta_1} \cdot (\text{T})^{\beta_2}$$

In this model a, β_1 and β_2 are all constants for which the values are given in table 3, σ is the volatility over the last thirty trading days, and T is the duration of the order in hours.

The third component that Bloomberg investigates is the permanent impact. In their research, Bloomberg finds out that the only factors that determine the permanent market impact are the size divided by the average daily volume and stock volatility.

The permanent impact is only dependent on how many stocks were exchanged and not on the order's aggressiveness. The aggressiveness only affects the temporary impact, and this disappears over time. The permanent impact can then be expressed with the following formula:

$$\text{Permanent Impact } = \gamma \cdot \sigma \cdot (\frac{\text{Size}}{\text{ADV}})^{\eta}$$

In this formula, γ and η are constants for which the values are given in table 3, and σ is the volatility over the last thirty trading days.

The new Bloomberg model for transaction costs is then calculated by combining these three components, such that.

$$\begin{aligned}\text{Market} &= \text{Instant impact} + \text{Temporary impact} + \text{Permanent impact}\\ &= \lambda \cdot \text{ Bid-Ask Spread } \% + \alpha \cdot \sigma \cdot (\frac{\text{Participation } \%}{100})^{\beta_1} \cdot (\text{ T})^{\beta_2}\\ &+ \gamma \cdot \sigma \cdot (\frac{\text{Size}}{\text{ADV}})^{\eta}\end{aligned}$$

A process called model calibration is used to get the parameters for this model.

Definition 7. Model calibration is a technique that adjusts the parameters such that the objective function is optimal.

The realized coefficients and confidence intervals for the Bloomberg model are shown in table 3. In this table the parameters α, β_1, β_2, γ, η are found with model calibration (Rashkovich and Verma, 2012), and the optimizations are performed using the Levenberg-Marquardt method (Levenberg, 1944).

Table 3: Coefficients of the Bloomberg model (Rashkovich and Verma, 2012)

λ	-0.25 + 3 $\cdot$(max (min$(0.3, \frac{\text{participation\%}}{100}), 0.05) - 0.05)$
α	0.023 $\pm$0.0014
β_1	0.76 $\pm$0.06
β_2	0.19 $\pm$0.04
γ	0.030 $\pm$0.0017
η	0.81 $\pm$0.08

An advantage of this model is that a trader has only two unknown variables before he enters an order in the market, participation rate and duration. In *Section 4.3* it was stated that the participation rate and duration are intertwined, and that the duration of an order can be estimated by calculating: $\frac{\text{ADV}}{\text{Part\%}}$.

If duration is replaced by $\frac{\text{ADV}}{\text{Part\%}}$ there is only one variable that remains unknown. The trader can then choose the value of the participation rate such that the market impact is minimized.

Bloomberg tested how much of the market impact their new model could explain with different participation rates and a different values for $\frac{\text{size}}{\text{ADV}}$. The R-squared for these different values is shown in the following table:

Table 4: $R^2(\%)$ of the Bloomberg market impact model with different participation and $\frac{size}{ADV}$ rates (Rashkovich and Verma, 2012)

		Participation %						
	$R^2(\%)$	1-5	5-10	10-15	15-25	25-35	35-50	50-100
$\frac{Size}{ADV}$	1-5	3.3	4.0	6.0	7.6	10.4	15.9	19.2
	5-10		4.6	6.3	9.5	12.7	14.6	23.5
	10-20			7.0	12.0	14.8	18.8	24.3
	20-50				11.6	14.7	18.5	23.5

The R-squared is a measure that gives information about how much of the variance of a dependent variable is explained by the independent variables of the model. Because of this, the R-squared is a good measure to see how much a certain model matches the actual values. If the R-squared has a value of one, this means that the variance of the dependent variable is completely explained by the model. This means that the model works well.

In table 4 it becomes clear that the R-squared increases if the participation rate and $\frac{Size}{ADV}$ increase. A reason for this is that large and aggressive trades have a higher impact on the market. This means that they are easier to measure and, therefore, a market impact model is better able to predict the market impact.

At the beginning of this subsection, it was stated that only 17% of the implementation shortfall could be accounted for via liquidity sourcing. In table 4 it becomes clear that when the participation rate is more than 35% the R-squared is more than 17% in most cases.

Bloomberg states their model can have an R-squared of more than 17% because if the participation rate is above 35%, this order creates momentum itself. Their model captures this momentum, and for this reason, the R-squared can be higher than 17% for high participation rates.

6.3 Difference between the actual market impact and the estimated market impact in numbers

From the data, it becomes clear that the Bloomberg model is not correct for the orders that PGGM made. This is something that Bloomberg knows. In *Section 6.2* they stated that a model for the market impact could only have a predictive power of maximum of 17%.

The average market impact of an order by PGGM is -11,40 basis points, and the average expected market impact is -13,88 basis points. Therefore, the difference between the market impact as estimated by Bloomberg and the actual market impact is 2,38 basis points.

This is quite a big difference compared to the expected market impact, namely $\frac{-11{,}40-(-13{,}88)}{-13{,}88} \cdot 100\% = -21,75\%$. This means that PGGM pays on average 21,75% less on market impact than it expects to according to Bloomberg's model.

At the beginning of this section, it was stated that an order had under-performance if the actual market impact is higher than the predicted market impact and that an order had out-performance if the actual market impact is lower than the predicted market impact. The calculation above shows that PGGM's traders on average reach out-performance.

However, this total difference does not give us that much information. The reason for this is that positive and negative values cancel each other out. A huge over-estimation of the market impact will lower this difference, while this can still be bad for PGGM.

The root mean square error (RMSE) is calculated to get a better view of the difference between the actual market impact and the market impact that Bloomberg estimates. The RMSE is a measure that gives an indication of the accuracy of a model, and it is calculated in the following manner (Barnston, 1992):

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}$$

With this calculation, the RMSE of a particular model can be calculated. The RMSE is calculated by squaring the difference of the predicted market impact, $\hat{y}_i$, and the actual market impact, y_i, for all n observations. This difference is squared such that a positive and a negative difference do not cancel each other out. It is namely essential to know how well the Bloomberg model predicts the market impacts for all stocks.

Of course, it is desirable for PGGM that the actual market impact is lower compared to the predicted market impact. This means that the order had lower costs than expected. However, if PGGM had known this information before the trade was executed, they might have chosen to execute this trade faster to have fewer opportunity costs or to possess this stock earlier.

Another reason that the difference is squared is that more significant differences have a higher weight on the RMSE compared to less significant differences. After the difference is squared for each observation, these squared differences are summed up and divided by the number of observations n. In our case, the number of observations is determined by the size of the test set. After this is done the root of this number is taken. Because the root is taken after the squared differences are summed up and divided by n, more significant differences have a higher weight on the RMSE.

In our case, it is not desirable that more significant differences have a higher weight compared to smaller differences. The goal of this research is to find a model that can predict the market impact accurately, and significant differences should not have a heavier weight when determining the accuracy of such a model. The reason for this is that the market impact is very variable and large differences often occur for created market impact models. These significant errors are not particularly less desirable compared to the small errors.

Because of this, another measure is used to indicate the accuracy of the model, called the Mean Absolute Error (MAE). This measure looks at the absolute difference between the predicted value and the actual value, and it is calculated in the following manner:

$$MAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{n}$$

With this calculation, the MAE of a particular model can be calculated. The MAE is calculated by taking the absolute difference between the predicted market impact, $\hat{y}_i$, and the actual market impact, y_i, for all n observations. This difference is then summed up and divided by n.

The MAE between the actual market impact and the estimated market impact is 76,49. There is thus quite a big difference between the actual market impact and the estimated market impact. This difference shows that the Bloomberg estimate is not quite an accurate predictor to estimate the market impact for the orders of PGGM.

As stated in *Section 6.2*, another method to see how much the Bloomberg market impact estimate matches the actual market impact is the R-squared. The R-squared shows the amount of variance in the dependent variable that is accounted for or explained by the independent variables.

Just like the RMSE and the MAE, the R-squared is a measure that is often used to show the accuracy of the model. The R-squared is calculated in the following manner:

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}} = 1 - \frac{\text{Sum of Squares Regression Error}}{\text{Sum of Squares Total Error}} = 1 - \frac{\sum_{i=1}^{n} \left((y_i - \hat{y}_i)^2 \right)}{\sum_{i=1}^{n} \left((y_i - \frac{\sum_{i=1}^{n} y_i}{n})^2 \right)}$$

The R-squared thus divides the unexplained variation by the total variation. This is done by dividing the sum of squares of the regression error by the sum of squares of the total error. The sum of squares of the regression error is the squared difference between the actual market impact and the predicted market impact. The sum of squares for the total error is the squared difference between the actual market impact and the average of the actual market impact. This value is then subtracted from one to get the value for the R-squared.

If the R-squared has a value of one, the model completely explains the dependent variable's variance. This is only the case when the predicted market impact and actual market impact are equal for all observations. The R-squared is -0.0259 in our data set.

A negative R-squared occurs when the average difference between the actual market impact and the predicted market impact is more substantial than the average difference between the actual market impact and the average of the actual market impact. The chosen model fits worse than a horizontal line of the average, which shows once more that the Bloomberg estimate is not an accurate predictor to estimate the market impact for the orders of PGGM.

As stated above, PGGM has no information about the model Bloomberg currently uses to predict the market impact. It is possible to look at the models that are given above with our data, and see how these models compare to the current Bloomberg model. By using the variables in our data set, the model from 2008 looks as follows:

$$\text{Market Impact} = -100 \cdot \left(0.5 \cdot \frac{\text{Bid-Ask Spread}}{100} + \sqrt{\frac{(\text{10dAbsVol}^2)/3}{250}} \cdot \sqrt{\frac{\%ADV}{0,3}} \right)$$

In this equation, the market impact is multiplied by minus one hundred because in our data set the market impact costs are given by a negative number in basis points. The bid-ask spread is divided by one hundred because in the data set it is given in basis points, while it needs to be in percentages for this calculation. Furthermore, % ADV replaces $\frac{V}{EDV}$ because in our data set % ADV represents volume divided by the average daily volume.

The MAE for this whole data set between the old Bloomberg model and the actual market impact is 76,54. This is slightly higher than is the case with the current Bloomberg model, although this is not substantial. The R-squared decreases to a value of -0,0311.

To see how much the current Bloomberg model and this model are alike, the MAE and the R-squared are calculated between these two models. This resulted in a MAE of 13,81, which is reasonably low. This shows that the new Bloomberg model has not adjusted that much and that the bid-ask spread, volatility and average daily volume are still essential variables in the Bloomberg model. This is as well visible in the high R-squared of 0,4334.

Let us take a look at the Bloomberg model from 2012. By using the variables from our data set and inserting the parameter values that are given in table 3, the model looks as follows:

$$\begin{aligned}\text{Market Impact} &= -100 \cdot (\lambda \cdot \frac{\text{Bid-Ask Spread \%}}{100} + \alpha \cdot \text{10dAbsVol} \cdot (\frac{\text{Participation \%}}{100})^{\beta_1} \cdot (\text{T})^{\beta_2} \\ &+ \gamma \cdot \text{10dAbsVol} \cdot \text{ADV}^{\eta}) \\ &= -100 \cdot (-0.25 + 3 \cdot (\max(\min(0.3, \frac{\text{Participation \%}}{100}), 0.05) - 0.05) \cdot \frac{\text{Bid-Ask Spread \%}}{100} \\ &+ 0.023 \cdot \text{10dAbsVol} \cdot (\frac{\text{Participation \%}}{100})^{0.76} \cdot (\text{T})^{0.19} + 0.03 \cdot \text{10dAbsVol} \cdot \%\text{ADV}^{0.81})\end{aligned}$$

In this equation, the market impact is multiplied by minus one hundred because in our data set the market impact costs are given by a negative number in basis points. The bid-ask spread is divided by one hundred because in the data set it is given in basis points, while it needs to be in percentages for this calculation. Furthermore, % ADV replaces $\frac{Size}{ADV}$ again. The MAE between the old Bloomberg model and the actual market impact is 83,04, which is substantially higher than is the case with the current Bloomberg model. The R-squared has decreased to a value of -0,5674. It is thus clear that with this formula Bloomberg does not even come close to their stated R-squared values.

To see how much the current Bloomberg model and this model are alike, the MAE is calculated between these two models. This resulted in a value of 28,87, which is a higher difference than between the 2008 Bloomberg model and the current model. Therefore, it seems like Bloomberg has dismounted from this model, and their current model looks more like the 2008 model. The high R-squared and low MAE for this model compared to the actual market impact show that this is justified.

An explanation for the fact that such a massive company like Bloomberg does not provide an accurate estimator for the market impact is that Bloomberg might have one model that explains the market impact of all stocks, and PGGM invests in specific stocks for which this model is not correct. Therefore, it is crucial to develop a model that can correctly explain the market impact for the stocks that PGGM invests in.

In section 5 it was already stated that the Bloomberg estimated market impact is always negative. This is also visible in their old models shown above. However, in reality, the market impact can also be positive. PGGM needs to know which trades have a positive market impact and for what reason these trades have a positive market impact. If PGGM can trade their orders to have a positive market impact more often, this will diminish their cost massively.

7 Modelling the market impact with machine learning

Machine learning is a form of artificial intelligence (AI) based on building a system that can learn from past data to improve performance. Machine learning algorithms build a model based on a sample set of data, called the training set, and use this model to predict another set of data, called the test set. A machine learning model is thus the output of a machine learning algorithm run on the data.

The training set has access to all variables and learns the relation between the input variables and the output variable (in our case market impact). This is what needs to happen in this research, as it is essential to make an accurate prediction of the market impact (output variable) with the available data (input variables). The test set does not have access to the output variable but does have access to the input variables. The test set then uses the model created by the algorithm with the training set to predict the output variable with the available input variables. The results of this test set can then be compared to the actual values of the data set.

Usually, around eighty per cent of the data set is used to make the training set, to ensure there is enough data to train the model. The remaining twenty per cent of the data set is used to test the newly trained model.

Machine learning can be used to solve classification problems and regression problems. Different algorithms are better for different situations. In our case, the market impact has to be estimated. Therefore, it is only possible for us to use machine learning algorithms that solve regression problems. These are the algorithms that will be tested, and from these algorithms, the algorithm that works best will be chosen to predict the market impact.

There are two significant types of machine learning algorithms, parametric models and non-parametric models. Parametric models use a pre-selected set of input variables to train the model, which means that the number of parameters doesn't change when the algorithm is run. Non-parametric models can select the input variables themselves from the complete data set.[7]

Because the training and test set are randomly sampled, they differ every time they are sampled. The model is created based on the training set, and if the training set is different, the model will be different as well. This means that the result of the model can change every time the algorithm is run. Because of this, the model makes different predictions and can have a different performance.

Therefore, a small difference in performance between two machine learning algorithms does not necessarily mean that one algorithm outperforms another. This is something to keep in mind when using machine learning algorithms to train a model.

It is possible to use the same train and test set every time an algorithm is run. However, it is still possible that one algorithm works better on a particular training set, and another algorithm works better on another training set.

There are three leading causes of a difference between actual values and machine learning algorithms' predicted values. These three leading causes are noise, variance and bias.

Definition 8. Noise is a deformation in the data. This is something that can't be explained by the model.

Let us look at a model with a certain set of n input variables $x_1, ..., x_n$ that predict the value of a certain output variable y. These variables cannot predict the output variable y exactly, because there is noise in the data. The model then looks as follows:

$$y = f(x_1, ..., x_n) + \epsilon$$

In this formula $f(x_1, ..., x_n)$ is an underlying function of the n input variables $x_1, ..., x_n$ and ϵ

[7] A non-parametric model can thus choose the input variables that this algorithm finds important when predicting the output variable from the complete data set (apart from the output variable), while a parametric model uses all pre-selected variables as input variables to predict the output variable.

is a measure of the noise in the data.

A machine learning algorithm's objective is to learn a function $g(x_1, ..., x_n)$ to predict the output variable y. The algorithm works well if the difference between y and $g(x_1, ..., x_n)$ is minimal. In *Section 6* it was explained that the mean absolute error is used to calculate this difference.

Because the noise function ϵ also influences the output variable y, the machine learning algorithm will likely learn about the noise as well. The model then looks as follows:

$$y = g(x_1, ..., x_n + \epsilon)$$

If this happens, the machine learning algorithm has overfitted. Overfitting happens when more variables are used to predict the output variable than can be explained by the data (Everitt and Skrondal, 2002). When this happens, variables that do not influence the output variable are used to predict the noise in the data (Burnham and Anderson, 2004).

Overfitting will cause that the created model works well on this particular data set, but not on another data set. The model will then work well on the training set, but not on the test set. This is also why the data is always split in a train and a test set for machine learning algorithms. When overfitting occurs, the variance of the model is high, and the bias is low.

Definition 9. Variance is a measure of how sensitive the machine learning algorithm is to the data.

High variance means that a small change in the input data that is created by noise gets picked up by the machine learning algorithm. This causes the model to overfit and results in poor accuracy of the model.

Definition 10. Bias is the inability of a machine learning algorithm to learn the relation between the input variables x and an output variable y. The machine learning algorithm is then not able to learn from the training set.

Let us now assume that a machine learning algorithm is trained such that an underlying function $g(x_1, ..., x_n)$ is found that tries to predict the output variable y, while there is noise in the data. The expected bias can then be mathematically represented with the following formula:

$$\text{Bias [g(x)]} = g(x) - f(x)$$

In *Section 6* it became clear that the market impact that Bloomberg gives is far from correct. They state it is only possible to have a market impact with a predictive power of maximum 17%.

However, later in their research, they create a model with a predictive power higher than 17%. Their explanation for this is that if the participation rate and the Size/ADV is high, their model also captures a part of the momentum.

In this section, multiple machine learning algorithms will be implemented to make a model that predicts the actual market impact more accurately. The programming language *Python* is used to implement these machine learning algorithms.

7.1 Parametric models

Parametric models use a particular set of input variables $A \subseteq X$ to determine the output variables Y. A parametric model assumes that these input variables A capture everything that there is to know about the data. The remaining variables $B = X \; / \; A = A^C \subseteq X$ then do not influence the output variable, such that:

$$M(Y|X) = M(Y|A, B) = M(Y|A)$$

In this equation, M is the model that is used. The input variables A need to be correct because the model is made based on these input variables. It is not possible to know or see which input variables are most important, so it is essential that the correct input variables are selected and that these input variables can explain the output variables.

In our case, there is only one output variable: market impact. The input variables are the variables that influence this market impact. Based on this research, it is clear that there are at least six main variables that should influence the market impact. These six variables are:

1. $\%$ Average Daily Volume (ADV) - This input variable will capture the size of the trade and the stock's liquidity.
2. $\%$ Participation - This input variable will capture how this order was carried out. Fast trades will have a much higher participation rate than slow trades.
3. Volatility - This input variable will capture a stock's volatility over the last ten days.
4. Duration - This input variable will capture the time it took for executing the complete order.
5. Bid-Ask spread - This input variable will capture the spread between the buy- and the sell orders.
6. Momentum - This input variable will capture the price change when the order was executed.

These six input variables were chosen based on the analysis done in *Section 4*. During the research done in this section, it became clear that all these six variables influence the market impact.

There is data available for all these six input variables. There is, however, one problem whit one of these variables. Although the momentum variable is available in the Bloomberg data, this is not a variable that the traders know beforehand. Momentum only becomes visible after an order is executed. Therefore, the momentum variable cannot be used to predict the market impact.

In *Section 2.2* it became clear that the price momentum of the past influences the price momentum of the future. Thus, it might be possible to use the stock's price difference before the order was executed as a momentum indicator. In the data set is information available about how the price has moved on the day of the order, one day before the order, two days before the order, three days before the order, four days before the order, one week before the order and one month before the order. These are the variables that will be used as momentum indicators to predict the market impact.[8]

[8] One side note to this is that this momentum strategy was measured over a more extended period (Jegadeesh and Titman, 1993). An order most often takes place over a couple of hours/days, so it remains to be seen how accurately this price difference can be predicted. The search for more variables that can explain the market impact goes on during this research.

Because our model needs to develop a numeric outcome to predict transaction costs, the only parametric models applicable are the models that can solve regression problems. The two main parametric machine learning algorithms that can solve regression problems are multiple linear regression and neural network.

7.1.1 Multiple linear regression

An example of a simple parametric machine learning algorithm is a multiple linear regression model. A multiple linear regression model is a model that can only be used to solve regression problems.

There are four assumptions that the multiple linear regression model needs to follow (Kenton, 2020):

1. There needs to be a linear relationship between the independent variables and the dependent variable.
2. The independent variables need to be not too highly correlated with each other.
3. Observations are selected independently and randomly from the population.
4. Residuals should be normally distributed with a mean of zero and variance σ^2.

The most unlikely assumption of these four assumptions is that there is a linear relationship between the independent variables $A \subseteq X$ and the dependent variable Y. Based on this research, it is not likely that this relation is linear. However, the multiple linear regression model is chosen because it is a relatively simple model, and this model will give a better insight into the data.

Another assumption that might not be satisfied is that the independent variables are not too highly correlated. Both ADV and participation rate are related to the size of the order, and the duration of an order can be calculated by dividing the ADV with the participation rate. Furthermore, in *Section 4.2* it was stated that bid-ask spread and volatility are correlated. It is not clear if this correlation is too high for the model to work. If this correlation is too high, one of the variables will be insignificant. If this is the case, this independent variable will be removed, and the model will be rerun.

Multiple linear regression is a model that predicts the value of one dependent variable based on two or more independent variables. This makes the multiple linear regression model applicable to our data.

As explained in this section's beginning, the data needs to be split into a test set and a training set. In our case, eighty per cent of the data is used for the training set to train the model. After this is done, this model is tested on the remaining twenty per cent of the data. This way, there is enough training data to develop a good model and enough testing data to test this model.

The predicted variable is the market impact, and the variables used to predict the market impact are ADV, participation, volatility, bid-ask spread, and the available price differences that are used as momentum indicators.

In the table 5 below, the multiple linear regression machine learning algorithm's main advantages and drawbacks are shown. Multiple linear regression is a simple algorithm that works well when there is a linear relation between the input variables and output variables. However, it is not very likely that this is the case with our data set.

Table 5: Advantages and drawbacks of a multiple linear regression model

Advantages	Drawbacks
Easy to interpret	Assumes a linear relation between the input variables and the output variable
Not much data required	Requires pre-selection of predictive input variables
Gives a clear model as output	

7.1.2 Artificial neural network

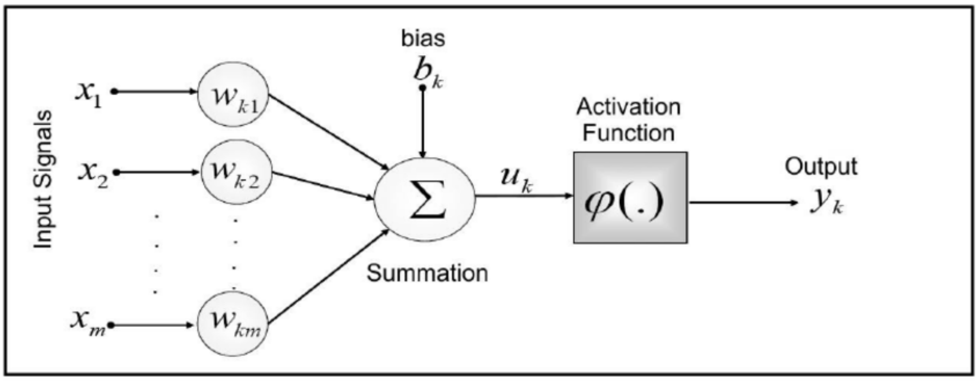
An artificial neural network (ANN) is based on the neural network of the human brain. Like is done in the human brain, each connection transmits a signal from one neuron to another. An artificial neural network can be used to solve a classification or a regression problem.

An ANN can find patterns in the data or find relations between input and output variables. The connection between neurons is called an edge. All edges in an artificial neural network have a weight based on how important they are. The artificial neural network algorithm adjusts this weight as it learns. The weight increases the strength of a signal at an edge when this edge has a lot of influence on the output variables, and the weight decreases the strength of a signal at an edge when this edge does not have a lot of influence on the output variables.

The neurons are divided into layers. Signals move from the input layer to the output layer. During this process, they move through one or multiple hidden layers. Each of these layers can perform a different transformation on its input. Each neuron gives a specific output based on the input it receives. This neuron transforms its input variables to an output variable based on the activation function it has.

The different layers in a neural network can perform different transformations on their inputs, which means that the neurons in different layers can have different activation functions. However, the activation functions in a neural network are usually the same. These different layers with non-linear activation functions make it possible for the network to see non-linear behaviour in the model. The manner in which a single neuron with input weights works, is shown in figure 7 below:

Figure 6: Example of a neuron in a neural network (Veronez et al., 2011)



In this figure, the input signals are multiplied by the weights and then summed up with a bias. This summation is then put into an activation function, such that the neuron can generate an output. There are three primary types of activation functions. These functions are the binary step function, the linear activation function and the non-linear activation functions.

The binary step function will return a value of one as output if the input value of a neuron is above a specific benchmark value. If this neuron's input value is below this benchmark value, the binary step function will return zero as an output value for this neuron. Because the binary step function returns zero or one, which means a no or a yes, this function is not applicable to predict the market impact.

The linear activation function can return all real numbers $\mathbb{R}$ as an output value. This linear activation function returns the input variables multiplied by their weights, summed up with a bias.

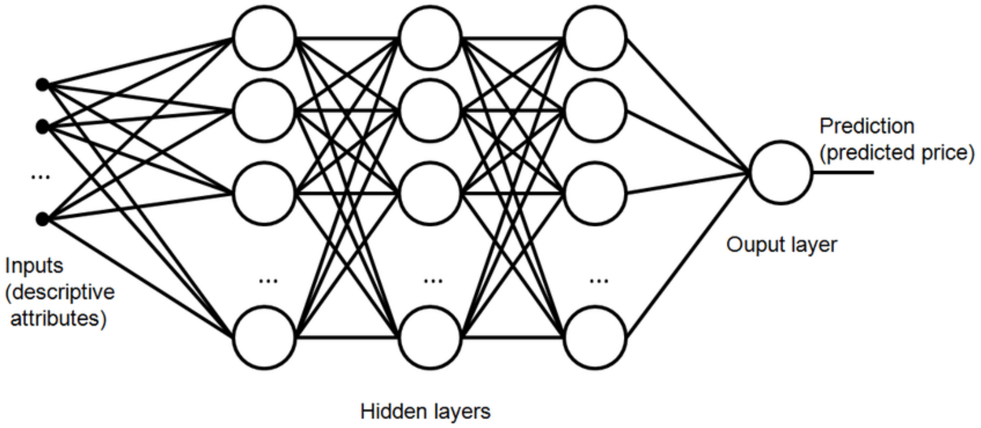
Nonlinear activation functions can establish complex relations between the input values of neurons and the output value. The output of a nonlinear activation function depends on what kind of activation function is used. The four most important and well-known nonlinear activation functions are the Sigmoid function, the Hyperbolic Tangent function, the Rectified Linear Unit (ReLU) function and the Swish function. The formulas of these four functions are given below:

1. Sigmoid function: $\sigma(x) = \frac{1}{1+e^{-x}}$
2. Hyperbolic Tangent function: $\sigma(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
3. ReLU function: $\sigma(x) = \max(0, x)$
4. Swish function: $\sigma(x) = \frac{x}{1+e^{-x}}$

In this formula, the variable x is given by the input variables multiplied by their weights and summed up with a bias. The Sigmoid function returns values between zero and one, while the Hyperbolic Tangent function returns values between minus one and one. Because of this, these two functions are mostly used for classification problems. However, these activation functions can also be used for regression problems. This will be explained further down this subsection. The ReLU function can return all positive real numbers $\mathbb{R}_+$ as output, while the Swish function can return all real numbers greater or equal than minus one $\mathbb{R}_{\geq -1}$ as an output value.

The best weights in the neural network have to be determined in order to have the best predicting power. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight is adjusted such that the error between the output value and the target value is minimal. In this case, the target value is the actual value of the market impact in the test set, and the output value is the market impact as predicted by the artificial neural network. The weight increases or decreases the strength of the signal at a connection. This network of connected neurons consists of multiple input variables, one or more hidden layers and one or more output variables. The artificial neural network then looks as follows:

Figure 7: Example of a neural network with multiple input variables, three hidden layers and one output variable (Bazan - Krzywoszańska and Bereta, 2018)



There are two major types of artificial neural networks. A shallow neural network is a neural network with one hidden layer, and a deep neural network (DNN) is a neural network with two or more hidden layers. In general, DNN's are more accurate.

In every connection a signal can be transmitted from one neuron to another neuron. The activation function of the receiving neuron then processes these input signals and give a new signal to the connected neurons, as was visible in figure 7. These signals travel from the first layer (the input layer) to the last layer (the output layer), possibly after passing the layers multiple times. A signal can pass these layers multiple times via a process called backpropagation.

Definition 11. Backpropagation is the technique that changes the weights in a neural network, by going backwards in this neural network, such that the difference between the output value and the actual value is minimal.

Backpropagation is a technique that changes the neurons' weights repeatedly until the error of the output function is minimal. This error is calculated by taking the sum of squares between the actual output and the desired output. Therefore, the error function E looks as follows:

$$E = \frac{1}{2}\sum_{i \in I}(o_i - a_i)^2$$

In this function, o_i is the output value of neuron i, and a_i is the actual output value neuron i should have. The neural network is run over and over again until this error is minimal.

Backpropagation is done via a mathematical technique called gradient descent. This technique looks at the derivatives of the function's parameters and searches which step, via which parameter is the best step to minimize the function's error.

The weights are updated step by step, starting at the output layer until it reaches the input layer. This process is repeated until the error function E is minimal. This technique finds new weights for all neurons and makes sure that the error decreases step by step. By using the gradient descent algorithm, the update function for the weights looks as follows:

$$w_{ij} = -\alpha \frac{\delta E}{\delta w_{ij}}$$

In this function w_{ij} is given by the output weight j of neuron i, α is a constant and E is the error function E given above.

The number of hidden layers and neurons per hidden layer can be chosen freely in this neural network. However, one rule of thumb for hidden layers "is that it should never be more than twice as large as the input layer" (Berry and Linoff, 1997).

The batch size is a hyperparameter that determines the number of samples of the training set that are propagated through the artificial neural network. After each propagation, the weights of the neural network are updated. A batch size of one hundred means that the test set is split into one hundred smaller sets. These smaller sets are all run through the algorithm. A higher batch size will mean that the estimate of the gradient will be more accurate. However, this also requires a higher computation time.

The number of epochs is a hyperparameter that determines the number of times the artificial neural network works through the entire training data set. One hundred epochs mean that the artificial algorithm runs one hundred times through the training set. For every run, the test set is then split into one hundred batch sizes.

At the beginning of this section, activation functions were discussed. The five activation functions that can be used to solve regression problems were: the linear activation function, the Sigmoid function, the Hyperbolic Tangent function, the ReLu function and the Swish function. There are, however, two problems with the linear activation function.

First of all, it is not possible to use backpropagation. As stated above, backpropagation uses a technique called gradient descent. This technique takes the derivative of the error function and the weight. If the linear activation function is used, the derivative is a constant and has no relation to the input. This means that it is not possible to go back in the neural network to understand which weights in the input neurons can provide a better prediction.

Another problem with the linear activation function is that the number of hidden layers does not influence the output. Above it was mentioned that more hidden layers mean more accurate results to some extent. However, with a linear activation function, the last layer will always be a linear function of the first layer.

A linear combination and composition of multiple linear functions is still a linear function. Thus, a linear activation function changes a neural network with multiple layers into a neural network with just one layer. For this reason, the linear activation function is not used as an activation function in the neural network to predict the market impact.

There are still four non-linear activation functions that can be used. The first one is the Sigmoid function. As stated above, the Sigmoid function returns values between zero and one. This means that there are two options to use this function.

The first option is to normalize the output data such that the values of the market impact are between zero and one. Another option is to use the linear activation function in the output layer and the Sigmoid function in the hidden layers.

There is one more problem with this function: there is almost no distinction between low and significantly low values and between high and significantly high values. Because the market impact fluctuates a lot and significantly low and significantly high values often occur, the Sigmoid function might not be the optimal activation function to use to predict the market impact.

The second non-linear activation function that was discussed was the Hyperbolic Tangent function. This function returns values between minus one and one. Because of this, there are still two options to use this activation function. These two options were given above; normalize the output data or use the linear activation function in the output layer and the Hyperbolic Tangent function in the hidden layers.

Like is the case with the Sigmoid function, there is almost no distinction between significantly low and significantly high values with the Hyperbolic Tangent function. Therefore, the Hyperbolic Tangent function is not the optimal activation function to use for our problem.

The third non-linear activation function that was discussed was the ReLU function. The ReLU function is the most used activation function in neural networks (Nair and Hinton, 2010).

This function might look like a linear function, but the ReLU function has a derivative that is not a constant and because of this, the ReLU function supports backpropagation. As becomes clear by looking at the ReLU function, this function is bounded by zero for negative values and unbounded for positive values.

A drawback of this function is that when the inputs are negative the function's derivative becomes zero. Some of the input variables in our data set can have negative input variables. The artificial neural network cannot perform backpropagation for the negative values with the ReLU function, and cannot learn from the negative values in this variable.

The last non-linear activation function that was discussed was the Swish function. The Swish function is the Sigmoid function multiplied by x. Because of this, the values of the Swish function do not lie between zero and one, and this function can be used to model the market impact. The Swish function is bounded for negative values and unbounded for positive values, just like the ReLU function.

However, the Swish function is smooth and non-monotonic, unlike the ReLU function (Ramachandran, Zoph, and Le, 2017). Therefore, the Swish function can still perform backpropagation when input values are negative. Therefore, it looks like the Swish function is an improvement on the ReLU function.

Based on this theory, the two most useful activation functions to use for our problem are the ReLU function and the Swish function. These are the two activation functions used to predict the market impact in our artificial neural network.

In an artificial neural network, it is not possible to see which weights are used. It is also not possible to see which input variables have the most effect on the output variable. Because of this, a neural network can be called a black box model:

Definition 12. A black box model is a method that one can give input variables to and receive output from. However, it is not possible to see how the model came up with this output.

In table 6 below the main advantages and drawbacks of the artificial neural network algorithm are shown. An artificial neural network is a complex algorithm that does not give an explicit relation between the input variables and the output variables. As stated above, a neural network is a Black Box model, which makes it impossible to see the relation between the input variables and the output variable. However, an artificial neural network has a high predictive power compared to the multiple linear regression, and this algorithm is likely to give a more accurate output.

Table 6: Advantages and drawbacks of a neural network

Advantages	Drawbacks
High predictive power	Difficult to interpret
Does not assume a relation between the input variables and the output variable (flexible)	Requires pre-selection of predictive input variables
	It is a Black Box model
	Requires a large data set

7.2 Non-parametric models

The benefit of parametric models is that they are easy to understand, and their results are easily interpreted. Besides, these models are fast and do not require much data. However, because the input variables $A \subseteq X$ of a parametric model need to be chosen before the model is run, they are constrained. It is possible that other variables $B = X \;/\; A = A^C \subseteq X$ in the data set influence the output variables, while they are not used by the parametric model. The model then looks as follows:

$$M(Y|X) = M(Y|A, B) \neq M(Y|A)$$

Non-parametric models do not make strong assumptions about the form of the function. For these models, it is unnecessary to choose the input variables before the model is run because they can consider all available data. This means that such a model cannot ignore variables that influence the market impact.

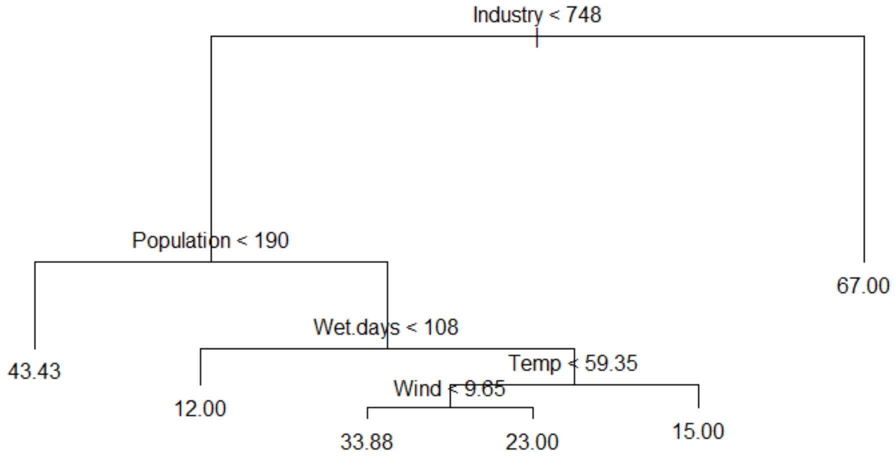
A drawback of non-parametric models is that they require a considerable amount of data to work correctly. However, the data set provided by PGGM is big enough to implement such non-parametric models. The two main non-parametric machine learning algorithms that can solve regression problems are the random forest and the gradient boosting tree.

7.2.1 Random forest

A random forest consists of multiple decision trees. There are two types of decision trees: classification trees and regression trees. Classification trees can solve classification problems, while regression trees can solve regression problems. A term that captures both these trees is the Classification and Regression Tree (CART) analysis. In our case, the regression tree is needed to estimate the market impact.

A regression tree consists of multiple nodes (conditions). At each node the tree is split into multiple edges (branches). The last edge is called the leaf (decision), and this leaf determines the model's output value. An example of such a regression tree is visible in the figure below:

Figure 8: Example of a regression tree for a pollution data set(Vala, 2019)



This regression tree determines the pollution there is based on the variables industry, population, wet days, temperature, and wind. The variable for which the regression tree splits first is the most important variable.

As the regression tree is a non-parametric model, only a small selection for the input variables is needed. The regression tree takes all available variables in the data as input variables, apart from the output variable. This regression tree then has to decide on which variable it splits at first, which is done by repeatedly testing out different splits on all variables. The squared error function is then used to determine how much accuracy is lost during a split:

$$\text{Costs} = \sum (Y - \text{Prediction})^2$$

The first variable in the tree is chosen such that the least amount of accuracy is lost, i.e. the variable for which the costs of the split have the lowest value. The second variable is chosen in the same manner, and so on. This process is repeated until all variables are in the tree.

Our data set has many variables, which means that the tree consists of many splits and is sizeable. This can lead to overfitting.

As stated at the beginning of this section, a random forest consists of multiple decision trees. The reason for this is that a random forest is created via a process called bagging predictors.

Definition 13. "Bagging predictors is a method for generating multiple versions of a predictor and using these to get an aggregated predictor. The multiple versions are formed by making bootstrap replicates of the learning set and using these as new learning sets." (Breiman, 1996)

A random forest consists of multiple aggregated decision trees. All these decision trees are trained on a random subset of the variables of the training set. This is done to make sure that not all decision trees in the random forest are alike. A process called bootstrapping is used for this problem.

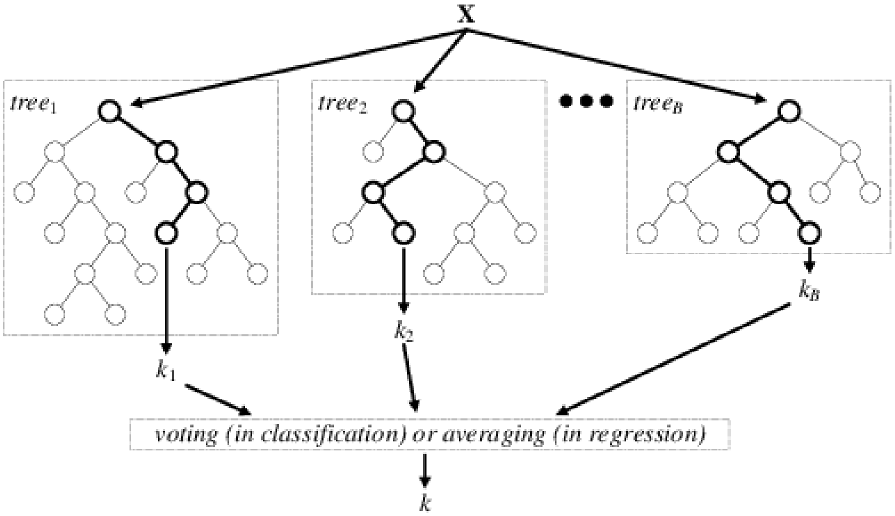
Definition 14. Bootstrapping is a technique that makes multiple smaller samples from a large sample, with replacement. In this case, "with replacement" means that this observation is put back in the data set after a random observation (order) is put in the sample. After the observation is put in the sample, the next random observation will be put in the sample. This process is repeated until the sample size reaches a given value N. It is thus possible that an observation is put multiple times in a specific sample.

With all these randomly created samples, different regression trees are formed. Because the samples are created with replacement, an observation can enter a sample more than once.

Thus, by using the bootstrap technique, multiple regression trees are created from the training set. These regression trees all contain a random subset of the orders that are in the training set. Because the regression trees are created with replacement, the different regression trees can have matching orders and variables.

When the output of the market impact has to be estimated for a value in the test set, all input variables are inserted in all created regression trees. The expected market impact is then the average outcome of all these trees. A visualization of how this works is shown in the figure below:

Figure 9: Example of a random forest with B trees (Verikas et al., 2016)



In figure 9 it is clear to see that a different path is taken in each regression tree. The expected market impact is then calculated by taking the average of the outcomes of all regression trees. Therefore, the market impact is calculated with the following formula:

$$\text{Market Impact} = \frac{1}{B}\sum_{n=1}^{B} f_n(x)$$

In this formula, B is the number of trees in the random forest, $f_n(x)$ is the n-th decision tree, and x are all input variables that are used to predict the market impact.

Therefore, a random forest is a machine learning algorithm that uses many weak learners (decision trees) and converts these into a strong learner (random forest).

Definition 15. A weak learner is a machine learning algorithm that is just slightly better than chance.

Definition 16. A strong learner is an algorithm that can achieve good performance and can make an accurate prediction.

In comparison to a regular decision tree, a random forest decreases the variance of the model (without increasing the bias). When a random forest is used, the variance of the expected market impact can be shown with the following formula:

$$\sigma^2 = \frac{\sum_{n=1}^{B}(f_n(x) - \text{Market Impact})^2}{B-1}$$

By looking at this formula, it becomes clear that if there are more trees used in the random forest, the expected market impact's variance decreases. However, overfitting is not solved by just taking many trees in a random forest. The trees in the random forest are all solved with the same training set. Because there are multiple small samples created from this training set, these small samples have overlap in observations and variables. This means that the trees will be correlated.

Therefore, there can still be more variables in the data set used to predict the output variable than can be explained by the data. This problem can be solved by decreasing the size of the trees in the random forest.

It is possible to set a minimum number of inputs for an edge or a leave to decrease the size of the tree. When a minimum number of inputs is determined, this means that at least a certain amount of the training data must satisfy the condition before a new edge is created. If this amount is not high enough, the split is not accepted. If there are no other splits that can then be accepted, this is a final node (leave). The predictive value of this node is the average of the samples that are in the split.

Other options to make the tree smaller is to set a maximum depth for a tree. The maximum depth of a decision tree is given by the largest distance between a leaf in the tree and its root. Thus, a decision tree's maximum depth displays how many splits can be present in the decision tree. If there are fewer splits, the tree automatically becomes smaller. Another manner to make the tree smaller is to use a technique called pruning.

Definition 17. Pruning is a technique that removes the edges in the tree with the lowest importance.

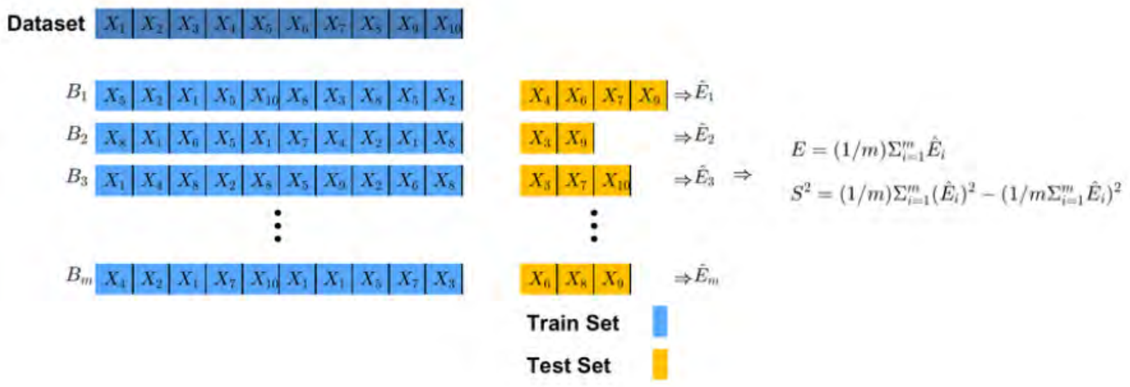
These techniques make the tree easier to interpret and reduce the tree's complexity such that there is no overfitting.

Another advantage of the random forest is that it can also use data in the training set to test the model. Namely, it is possible to use the created Out of Bag (OOB) samples to test the model.

Definition 18. An Out of Bag (OOB) sample is a sample that contains all the observations that are not used when a sample is created with the bootstrapping technique. It is clear that with the bootstrapping technique multiple small samples were created. This means that for every sample that is created by the bootstrapping technique, there are observations that are not in this sample. All these observations together form an OOB sample.

Because there were many samples created with bootstrapping, there will be many OOB samples as well. In total, there will be as many OOB samples as there are trees in the random forest. Figure 10 is a display of how these OOB samples are exactly created.

Figure 10: Example of how OOB sampling works (Tohka and Gils, 2020)



In this figure, the bootstrapping samples are in blue, while the OOB samples are in yellow. Because these OOB samples are not used to train the model, it is possible to use these samples to test the model. In this case, it is unnecessary to split the data into a training and a test set, which means that the complete data set can be used to train the model. Because of this, OOB sampling is particularly profitable to use for small data sets.

In total, there are N observations in the data set. Let us say that each bootstrap sample X_i contains $n \in N$ observations that are randomly drawn from the data set. Above it was already stated that every sample was randomly drawn from the full data set ('with replacement'). The probability that a single random draw will not draw a specific observation j in a bootstrap sample is then given by:

$$\frac{n-1}{n}$$

The probability that a specific observation j will not be drawn by a bootstrap sample X_i, is given by:

$$P(j \not\subseteq X_i) = (\frac{n-1}{n})^n = (1 - \frac{1}{n})^n$$

It is possible to find a numerical solution to the formula given above if n approaches infinity ($n \to \infty$). Of course, this can never occur in reality, but it will give us a better view of the lowest possible probability that a specific observation j is not drawn by a bootstrap sample X_i.

Namely, if there are more observations $n \in N$ drawn by the bootstrap sample X_i, the probability that a specific observation j is drawn increases. To see what happens to the formula given above if N approaches infinity, the following calculation is done:

$$\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} = 0.3679$$

If n approaches infinity, there is still quite a lot of data that is not used in a single tree. The calculations above state that there will still be 36.79% of the data set that is not being used for every tree. This data can then be used to test the model. It remains a random draw so the OOB samples will differ in size. This was already visible in figure 10.

As stated above, OOB sampling makes it possible to use the complete data set to train the model, and this is particularly useful when there is a small data set. The data set that PGGM provided is quite sizeable, which means that it is not necessary to use OOB sampling in our case.

In table 7, the random forest machine learning algorithm's main advantages and drawbacks are shown. As stated above, a random forest consists of multiple trees. Therefore it is hard to give a relation between the input variables and the output variable. It is, however, possible to see which variables had the most predictive power. Furthermore, it is not necessary to make a pre-selection for the input variables.

Table 7: Advantages and drawbacks of a random forest

Advantages	Drawbacks
High predictive power	Difficult to interpret
Does not assume a relation between the input variables and the output variable (very flexible)	Sensitive to overfitting
Gives a ranking of how important the input variables are	Doesn't give the significance for each variable
Can use the complete data set as input variables	Requires a lot of data
Can use OOB samples to test the model when the data set is too small	

7.2.2 Gradient boosting tree

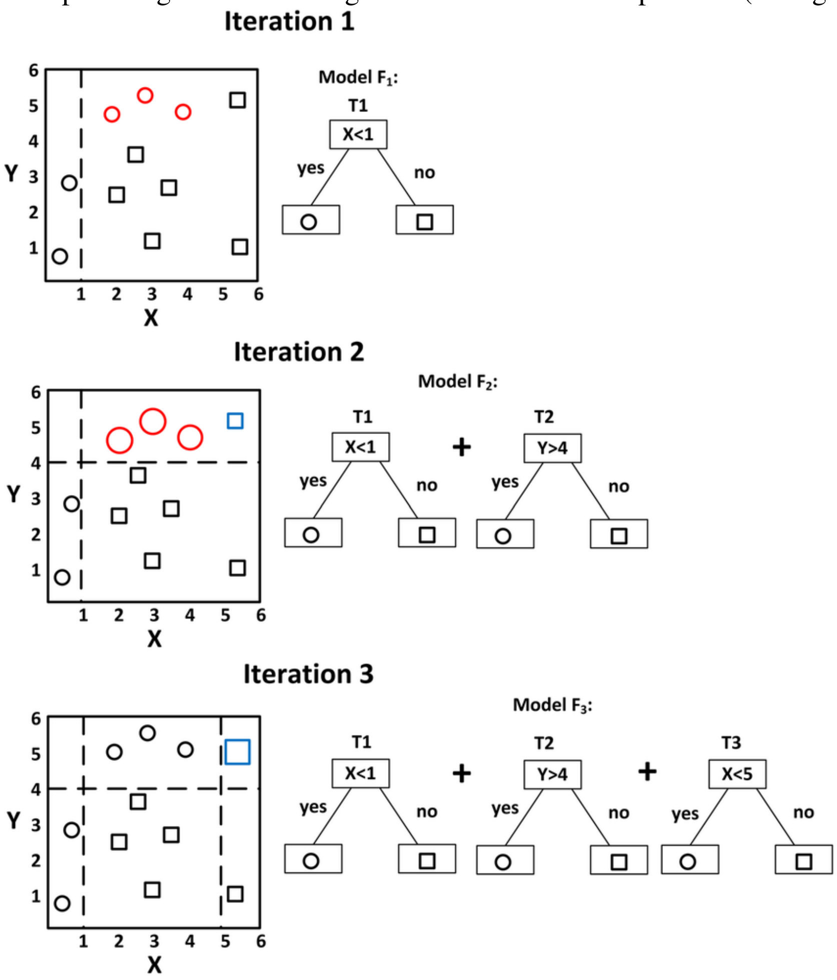
Gradient boosting is not a specific machine learning algorithm. However, it is a concept that can be applied to several machine learning algorithms.

Definition 19. Gradient boosting is an iterative process where a weak learner is fitted on the data set. Hereafter, a weak learner is used to make a new model that gives extra weight to the previous model's residuals. This process is repeated until the model can explain the complete data set. The final prediction is the weighted average of all previously generated models' predictions.

Boosting is a technique that uses many weak learners and converts these into a strong learner. This technique is similar to the bootstrapping technique discussed in the previous subsection.

The weak learner that is most often used with the gradient boosting technique is the decision tree. This is the same weak learner as is used in the random forest. A gradient boosting tree is a machine learning technique that can be used to solve a classification or a regression problem. An example of a gradient boosting tree for a classification problem can be seen in the figure below:

Figure 11: Example of a gradient boosting tree for a classification problem (Zhang et al., 2018).



In iteration one of figure 11 a first distinction is made between the squares and the circles. After this iteration, there are three circles incorrectly classified as a square. After the second iteration, this problem is solved. However, after this iteration, there is one square incorrectly classified as a circle. This is solved after iteration three. There have now thus been three different decision trees created. The final model F_3 is then a combination of these trees.

This example with two possible classifications and two variables is used to give a simple display of the gradient boosting tree. In our case, there are many more variables, and a regression needs to be executed to have a reasonable estimation of the market impact, which makes our problem a lot trickier.

Because a gradient boosting tree builds learns from the decision trees it has created and builds new decision trees with this information, a gradient boosting tree is more sensitive to overfitting if the data is noisy. This is the most significant disadvantage of the gradient boosting tree.

In table 8, the gradient boosting tree machine learning algorithm's main advantages and drawbacks are given. Because this algorithm uses multiple decision trees, the advantages and drawbacks given in this table are very much like the advantages and drawbacks of the random forest machine learning algorithm, which is given in table 7.

Table 8: Advantages and drawbacks of a gradient boosting tree

Advantages	Drawbacks
High predictive power	Difficult to interpret
Does not assume a relation between the input variables and the output variable (very flexible)	Very sensitive to overfitting
Gives a ranking of how important the input variables are	Requires a lot of data
Can use the complete data set as input variables	
Learns from previous decision trees	

8 Preparing the models

In this section, all variables of the data set will be evaluated and the optimal hyperparameters will be given per different technique.

8.1 Preparing the variables

The data set that PGGM provided was not immediately ready to be used. In this data set, there are among other things, N.A.'s (not available) and missing values. Some variables have more N.A.'s or missing values than others. When a model for the market impact is made with a particular set of variables, all the values in these variables must be available.

Furthermore, it might be the case that there is some false information in the data. In our case, the data is provided by Bloomberg. As explained in *Section 5* this is a sizeable financial institution. Because of this, the data is quite reliable. However, there are still some values in the data set that cannot be true. For example, the participation rate for some orders is higher than one hundred per cent.

In *Section 4.3*, it became clear that the participation is calculated by dividing the trader's size by the size of all orders that were executed during the trader's execution. This means that the participation rate cannot be higher than one hundred per cent. Bloomberg stated that this sometimes occurred because of timing differences. Therefore, the orders in the data set for which the participation rate is higher than one hundred per cent have to be removed. In total there were 107 observations removed because of this.

Next to this, the data needs to be made applicable. As explained in the previous subsection, the bid-ask spread variable can have the following values: 0-5, 5-10, 10-20, 20-50, 50-100, 100-200, 200+. These values give information about the value between which the bid-ask spread lay before the order was executed.

To create our models, the programming language *Python* will be used. The input variables of these models need to be numeric in order for *Python* to be able to use them. For this reason, the average of the bid-ask spreads shown above is calculated, and the 200+ is changed to just 200. Therefore, the bid-ask spread now has the following values: 2.5, 7.5, 15, 35, 75, 150, 200. By creating dummy variables, it is also possible to use the values in this variable. However, there will be more information in the numerical values.

Definition 20. A dummy variable is a variable that has a value of zero or one. These values represent a 'no' (zero) or a 'yes' (one).

When dummy variables are created, one variable is split into multiple dummy variables. The number of dummy variables depends on the number of possible answers there are in the variable. There are seven possible answers for the bid-ask spread, which means this variable would create seven dummy variables. Thus, the number of variables in the data set increases if variables are transformed into dummy variables.

8.2 Removing outliers

For a machine learning algorithm to perform optimally, there must be no outliers in the data.

Definition 21. An outlier is an unlikely observation in the data.

An outlier can occur due to a measurement or input error, corruption of the data, or an actual outlier observation. Our data set is provided by Bloomberg. In *Section 8.1* it became clear that some values in the data set cannot be true. Therefore, there may be more incorrect values in the data set.

The outliers are removed no matter if they are an actual observation. The reason for this is that outliers will make it hard for the machine learning algorithm to make an accurate model.

There can be several reasons that such actual outliers occurred. For example, a sudden stock price drop because of a huge scandal at a company can influence the market impact significantly.

If this observation is in the training set the machine learning algorithm will find patterns in the data that explain this market impact. However, it is impossible to explain the significance of the market impact with the data in this case.

If this observation is in the test set the market impact prediction of the machine learning model will probably be far away from the actual market impact. This observation can then increase the MAE significantly.

Several methods can be used to remove outliers. One of these methods uses the z-scores. The z-score shows how many standard deviations away from the mean an observation is. Therefore, the z-scores are calculated with the following formula: $z = \frac{x-\mu}{\sigma}$

In this equation, x is a specific observation, μ is the average of that variable, and σ is the standard deviation of that variable. Because all values in our data set are expected to be true values, not too much of these observations must be removed.

Therefore, only the extreme outliers will be removed. Observations that have a z-score higher than ± 4 can then be considered as extreme outliers. If the data had a normal distribution with a z-score higher than ± 4, this would mean that 0.006% of the observations are removed. There were 102 observations removed from the data set because of extreme outliers.

8.3 Removing unusable variables

As stated in *Section 7*, non-parametric models can use the complete data set as input variables. Therefore, it is not necessary to select or remove variables beforehand. However, there are still several reasons why it can be useful to remove some variables before the model is run. Some of these reasons are given below:

1. To reduce the computation time of the model. More variables mean a higher dimension of the model, which causes the model to take more time to run. There will be multiple variables in the data set that do not have any predictive power, and removing these variables will not change the model's outcome. It can, therefore, be useful to remove these redundant variables. Moreover, there may be highly correlated variables in the data set. Highly correlated variables provide the same information, and it is enough to keep one of these two (or more) correlated variables. The extra variables do not improve the model's performance, and for this reason, these variables can be removed as well.

2. To reduce the probability that the model overfits. In *Section 7* it was explained that overfitting causes variables that do not influence the output variable are used to predict the noise in the data. Removing these variables that do not influence the output variable will decrease the chance of overfitting.

3. To reduce the complexity of the model. Having many input variables makes the model quite complex and hard to follow for people who have less knowledge about the subject. Removing some variables means that the model will become a lot more clear to these people.

Above there are three reasons why it can be useful to remove variables from the data set. There are many variables in the data set that cannot be used to predict the market impact. Therefore, there is no information on these variables, which means that they can be removed.

In *Section 5* it became clear that there are a lot of variables in our data set. Some of these variables are not known before an order is entered into the market. These are output variables, and because they are not known before an order is entered into the market, they cannot be used to predict the market impact. Examples of these variables are the fill ratio and the momentum variable.

The fill ratio is the part of the order that is executed. When an order is entered into the market by PGGM, it is always the intention that this order is completed, and that the fill ratio is one hundred per cent. Due to several circumstances, it can occur that this does not happen. However, because the fill ratio is not known before an order is executed, this variable is an output variable and cannot be used to predict the market impact.

The momentum is the price movement of the stock during the execution of the order. In *Section 7.1*, it was stated that this is not a variable that the trader knows before making an order and that momentum only becomes visible after the order is executed. Because of this, the momentum variable is an output variable and cannot be used to predict the market impact.

Another output variable in our data set is duration. Duration is an output variable in our data set because traders do not exactly know how long an order will be when they enter it in the market. However, they do have an indication by looking at the % ADV and participation rate and can influence this duration by increasing their participation rate. Therefore, duration is kept as an input variable.

Furthermore, all other variables that use different benchmarks to indicate the market impact cannot be used to predict the market impact.

8.4 Removing redundant variables

In the subsection above, three reasons were given why it is useful to remove variables. In this section, the unusable variables were removed. Next to these variables that cannot be used to predict the market, some variables do not influence the market impact. These redundant variables can be removed, as well.

Another distinction can be made within these redundant variables. First of all, there are the redundant variables for which it is known beforehand that they do not influence the market impact. Secondly, there are the redundant variables for which it is known that they do not influence the market impact after the models are run.

8.4.1 Known redundant variables

Variables for which it is known beforehand that they do not contain any information about the market impact are the order ID, the ISIN, the security, the executed venue groups globally, the reason, the reasondesc and the previous close date.

All these variables can be removed before running the model to reduce the computation time of the model.

8.4.2 Other redundant variables

First of all, some variables have a high correlation with each other. If this correlation is high enough, these variables carry the same information. Because these variables carry much of the same information, only one of these variables is kept. In *Section 8.3* several reasons were given to remove these variables.

Besides, correlated variables make it harder to find the relationship between the input variables and the output variable. The reason for this is that the correlated variables compete with each other to try to explain the output variable.

Variables that have a correlation higher than ± 0.7 are considered to be strongly correlated. Therefore, these variables were removed. Variables that have a correlation higher than $|0.7|$ are the three days price difference, the four days price difference, the side sell orders, the market capitalization mid cap, and the type market orders.

It seems logical that the price difference variables are correlated. The price difference of a stock over the last week is comparable to the price difference of a stock over the last four days, because four of the five days are the same. Therefore, the difference only exists because of the fifth day. The correlation of 0.9155 reflects this. The price difference in the past four days was removed because this variable has a strong correlation with the two-day and three-day price difference as well, while the price difference in the past week only has a strong correlation with the three-day price difference.

The three-day price difference itself is removed because it has a strong correlation between the two-day price difference and the price difference in the past week. After removing this variable there is no strong correlation between the price difference variables anymore.

Side is one of the variables that was split into dummy variables. This variable can have two possible answers; buy and sell. Therefore, this variable was split into two dummy variables; a variable that returned a value of one when the order was a sell order and a zero otherwise, and a variable that returned a value of one when the order was a buy order and a zero otherwise. Because there are only two possible possible answers, these variables have a perfect negative correlation. The same goes for the type variable that can have two possible answers; market and limit.

The market capitalisation is another variable that was split into dummy variables. This variable can have four possible answers; micro, small, mid, and large. As explained in *Section 2.1*, sizeable companies like PGGM do not trade much in micro and small cap stocks. Therefore, almost all stocks are either mid cap or large cap. This means that if the market capitalisation mid cap variable has a value of zero, the market capitalisation large cap variable will have a value of one in most cases. This is visible in the high negative correlation these variables have, -0.9915.

In *Section 7.2.1* it was explained that it is possible to see the importance of variables in a random forest. Several methods can be used for this. These methods will be discussed in the next subsection.

The software machine learning library *Scikit learn* is used to build the random forest. With this library, it is possible to see the importance of each variable. A method called Recursive Feature Elimination with Cross-Validation (RFECV) in *Scikit learn* is used to remove the redundant variables. This method uses the importance of each variable to remove the redundant variables.

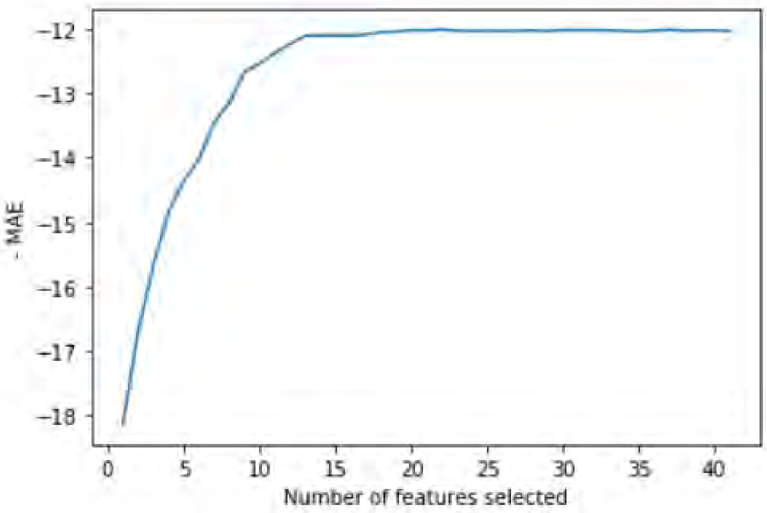
RFECV is a variable selection algorithm that works in the following manner: Let us say that there are M variables in the data set. Firstly, the model is run on the training set with all available variables V, and the MAE of this model is saved.

After this is done, a subset of variables $S_i \subseteq V$, where $i \in M$, is created excluding the variable with the lowest importance. The importance of variables is determined by looking at the Gini importance of the variables. The way this Gini importance works will be explained later in this section.

A variable is important if the MAE of the model increases after the variable is removed. This process is repeated M times until the model is run with just one variable. The model performed at its best for the subset S_i with the lowest MAE. This is the optimal set of variables. If multiple subsets have the same MAE, the subset with the least amount of variables is the optimal set of variables.

Because the process is repeated M times, the random forest is run M times with a different subset of variables $S_i \subseteq V$. Therefore, RFECV has a high computation time. This is also why it is essential to remove unusable variables, known redundant variables, and highly correlated variables before the RFECV is used. RFECV is a technique that maximises the objective function. Therefore, the negative MAE error is used and maximised. The result of the RFECV is shown in figure 12 below:

Figure 12: Negative MAE per number of variables using RFECV



In this figure, an optimum of the MAE is reached after the model is run with the twenty-two most important variables. The MAE is 12,01 in this case. However, the MAE stays around the same value after the random forest is run with the thirteen most important variables. The MAE of the model with thirteen variables is 12,11, which is only a slight increase compared to the model that was run with twenty-two variables.

The in *Section 8.3* stated benefits of removing variables can therefore be more advantageous for the model than the small increase of the MAE. This already becomes visible when the model is run on the test set with thirteen variables and with twenty-two variables.

The MAE of the model with thirteen variables is 66,05, while the R-squared is 0,1377. The MAE of the model with twenty-two variables is 66,28, while the R-squared is 0,1332. The model with thirteen variables has thus a lower MAE compared to the model with twenty-two variables. Therefore, our machine learning algorithms are run with the thirteen most important variables according to the RFECV.

The MAE for the test and training set show that the MAE for the test set is much higher than for the training set. It is a standard occurrence that the MAE for the test set is higher than for the training set, the model is namely built based on the training set. However, such a significant difference is not ordinary.

One reason for this significant difference might be overfitting. In *Section 7* it was mentioned that overfitting causes variables that do not influence the output variables are used to predict the noise in the data. However, all unusable and redundant variables have been removed. This shows once more how difficult it is to predict the market impact.

It is not without reason that Bloomberg's predicted market impact often substantially deviated from the actual market impact. The market impact of an order depends on the stock's price movement during the time interval of the order. The stock's price movement is almost impossible to predict and is extremely variable. Therefore, this might even be described as noise. The training set uses the variables in the data set to predict the market impact and thus the stock's price movement. Because the stock's price movement is almost impossible to predict and very variable the created model does not accurately predict the market impact in the test set. This explains the difference in the MAE between the training set and the test set.

Before the RFECV was run there were still forty-one variables. With the RFECV technique, the capitalization micro and small cap are removed. This means that the large cap variable is the only remaining variable that shows the capitalization of a company. There is now only a distinction whether a company is large cap or not.

Furthermore, the type limit variable is removed. This means that there is no information left in the data whether the order was a market order or a limit order and that this variable does not influence the market impact. Besides this, the countries Austria, Belgium, Denmark, Finland, France, Great Britain, Ireland, Israel, Italy, New Zealand, Norway, Poland, Portugal, Spain, Sweden, and Switzerland have been removed.

After this first selection process, there were still twenty-two variables left. However, by looking at figure 12 it became clear to us that the model would improve if it had thirteen variables. This became more clear after comparing the model with thirteen and twenty-two variables by looking at their MAE's and R-squared in the test set.

The random forest worked better with thirteen variables. Therefore, nine more variables were removed. The most unexpected variable that was removed during this process was the bid-ask spread. According to our theory, this variable should influence the market impact. This was also visible in both Bloomberg models that were discussed in *Sections 6.1 and 6.2*. The bid-ask spread was part of both these models that predicted the market impact. An explanation for this fact can be that there is no exact notation of the bid-ask spread in our data set, but only information about the value between which the bid-ask spread lied five days before the order was executed. This makes it harder for the random forest to use this variable to predict the market impact, and, therefore, there is not enough information in this variable and it is removed.

Furthermore, during this second selection process, the large cap variable was removed, which means there is no distinction whether a company is large cap or not anymore. Besides this, the countries Australia, Canada, Germany, Japan, Netherlands, Singapore, and South Korea were removed. The final selection of important variables is given below:

1. Size
2. Value
3. Duration
4. %Average Daily Volume
5. Part %
6. 10 day absolute volatility
7. Price difference on the day of the order
8. Price difference 2 days before the order
9. Price difference 1 week before the order
10. Price difference 1 month before the order
11. Side: Buy
12. Country: Hong Kong
13. Country: USA

These are the thirteen variables that are used by our machine learning algorithms to predict the market impact. In *Section 7.1* it was stated that the variables for parametric models need to be selected before the model is run, and in this section, six variables were selected that would be used to run these parametric variables.

However, after running the RFECV, it becomes clear that more variables can be used to predict the market impact. Therefore, the non-parametric models will be run with the variables that are given above.

8.5 Hyperparameter optimisation

In *Section 7*, four machine learning algorithms have been discussed. Apart from the multiple linear regression, all these algorithms have multiple hyperparameters. Good hyperparameter values must be chosen because only then it is possible to effectively use these four machine learning algorithms and compare them.

Definition 22. A hyperparameter is a variable for which a value has to be (or can be) inserted before the algorithm is run.

The number of hyperparameters and the sort of hyperparameters differ per algorithm. These different hyperparameters determine, among other things, the constraints, learning rates or weights of an algorithm. For instance, examples of hyperparameters are the number of hidden layers in a neural network or the number of trees in a random forest. Because the hyperparameter values have to be inserted into the algorithm, they cannot be trained by the training set. However, these parameters still influence the accuracy of the model.

Although the machine learning algorithm does not change, different hyperparameters can find different data patterns. Therefore these hyperparameters need to be tuned in order for the algorithm to work optimally. "The goal of the optimisation procedure is to find a vector that results in the best performance of the model after learning, such as maximum accuracy or minimum error" (Brownlee, 2020).

Definition 23. Hyperparameter optimisation is the technique that finds the set of hyperparameters that minimise the loss function and, as a result of this, optimise the model.

In our case, the loss function is given by the mean absolute error (MAE). The algorithm has to be run for each set of hyperparameters that is tested to find the minimum value for the MAE. This means that the algorithm first is trained on eighty per cent of the data set for each set of parameters and then tested on the remaining twenty per cent of the data set to find the different values of the MAE's.

By optimising the hyperparameters, it is possible to find the best solution for a machine learning model. It is not fair to compare two machine learning models if one model has a better selection of hyperparameters. Therefore, hyperparameter optimisation makes it possible to compare the different machine learning models with each other accurately. This optimisation procedure requires a search space that needs to be defined.

Definition 24. The search space is the volume that needs to be searched. This search space consists of multiple dimensions. Each dimension represents a hyperparameter that needs to be determined, and in this dimension the values that the hyperparameter may take on are given (Brownlee, 2020).

Several techniques can be used to optimise the hyperparameters. The three most famous techniques will be discussed below.

8.5.1 Grid search

Grid search is a technique that defines the search space as a grid of hyperparameter values and evaluates all positions in the grid. Therefore, this technique runs the machine learning algorithm for every possible combination from a selected grid of hyperparameter values. This grid of hyperparameter values has to be selected before the grid search is applied.

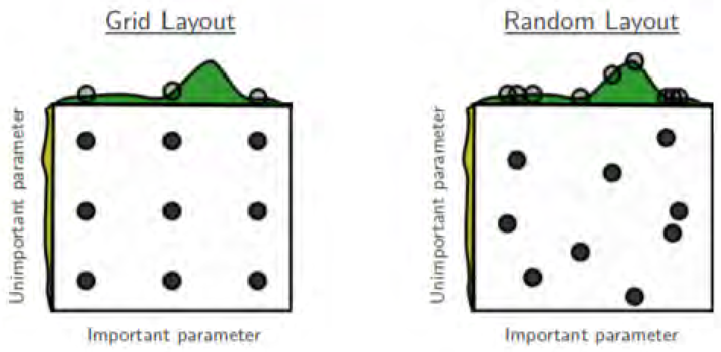
The grid search will then remember which combination of hyperparameters had the lowest value for the loss function. Because the model is run for every possible combination from a grid of hyperparameter values, this technique will always find the optimal combination of hyperparameters from that grid of hyperparameter values, such that the loss function in the training set is minimised.

One has to remember that this hyperparameters solution is not necessarily the optimal combination of hyperparameters for the model, but it is the optimal combinations of hyperparameter values from the pre-selected grid. Besides, the grid search is dependent on the train and test set that are formed. Therefore, the grid search can give different outcomes if it is run on a different training and test set (although it is from the same data set) or if the grid of hyperparameter values is different.

8.5.2 Random search

The random search technique differs from a grid search technique because a random search does not check every possible combination of hyperparameters from a selected grid. Instead of this, every hyperparameter is given a statistical distribution from which values are randomly sampled. The number of iterations that will be run can be chosen as well. The values for the hyperparameters can then be set by sampling from the given statistical distributions. The number of iterations that are done in a random search can be set based on time or resources. Figure 13 below shows a grid search and a random search.

Figure 13: Difference in performance between a grid search and a random search (Bergstra and Bengio, 2012)



In this figure, there are two hyperparameters. One of these hyperparameters is important and has a significant influence on the model's outcome, while the other hyperparameter is unimportant and does not have a significant influence on the outcome of the model. Nine combinations of hyperparameter values are tested for both the grid search and the random search. The green on top of the square shows the performance of the model.

The random search performs better in this case. This is because the grid search does not come close to the optimal value for the important hyperparameter, while the random search does come close to this optimal value. The reason for this is that the random search has nine different values for both hyperparameters, while the grid search only has three.

Thus, the random search has a higher chance of finding the optimal value for the important hyperparameter. This means that the model is likely to work better with a random search. In reality, a machine learning model always has more important hyperparameters and less important hyperparameters. Because the random search has more different values for the hyperparameters, this technique has a higher probability of coming close to the hyperparameters' optimal values. Because of this, the random search is likely to perform better compared to the grid search.

8.5.3 Bayesian optimisation

The two previously explained techniques test multiple sets of a combination of hyperparameters and compare these sets' loss function. The set that has the lowest value for the loss function is then the optimal set of hyperparameters.

With these two techniques it is not possible to use information from one iteration into the next iteration. However, Bayesian optimisation is a technique that can do this.

Bayesian optimisation is an approach that uses Bayes Theorem to direct the search in order to find the minimum (or maximum) of an objective function. Bayes Theorem is a technique that can be used for calculating the conditional probability of an event:

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

This calculation gives the probability that event A occurs if it is known that event B has occurred. In this case, the interest is in optimising a quantity and not in calculating a probability. Because of this, it is possible to remove the normalising value of P(B). The conditional probability is then described as a proportional quantity, such that: $P(A|B) = P(B|A) * P(A)$.

As stated above, Bayesian optimisation is a technique that can use information from one iteration into the next iteration. Therefore, Bayesian optimisation keeps track of past evaluations, and these past evaluations are used to form a probabilistic model mapping hyperparameters to a probability of a score on the objective function:

$$P(\text{Score}|\text{Hyperparameters})$$

This model is called a surrogate model for the objective function.

Definition 25. A surrogate model is a method used when an outcome of interest cannot be easily measured. As a replacement, a surrogate model approximates the output. A single evaluation of the surrogate model is generally much faster than a single evaluation of the original simulation.

Because of this, the surrogate model is much easier to optimise compared to the objective function. The objective function is in our case the MAE between the estimated value and the actual value in the train set. The surrogate model can be interpreted as an approximation of this objective function (Kraus, 2019).

The Bayesian optimisation technique selects a set of hyperparameters that performs best on the surrogate model and evaluates this set of hyperparameters on the actual objective function. Therefore, Bayesian optimisation consists of the following five steps (Koehrsen, 2018):

1. Build a surrogate probability model of the objective function.
2. Find the hyperparameters that perform best on the surrogate.
3. Apply these hyperparameters to the true objective function.
4. Update the surrogate model incorporating the new results.
5. Repeat steps 2–4 until the maximum number of iterations or time is reached.

These steps ensure that the surrogate probability model will become more like the objective function as the number of iterations increases. This is done based on a selection function.

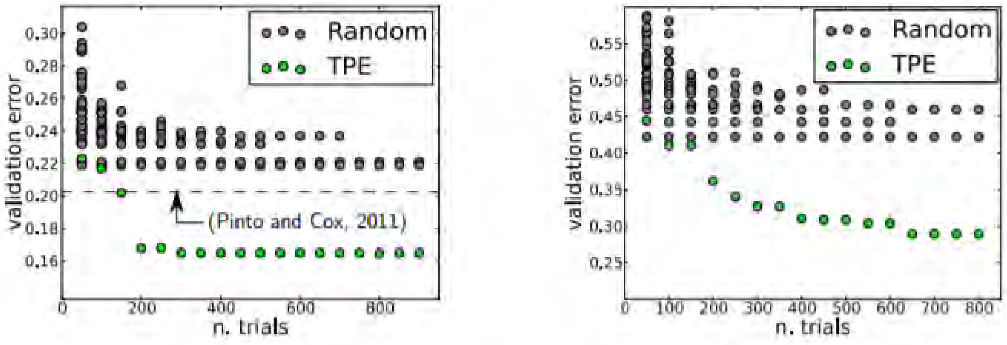
Definition 26. "The selection function is the criteria by which the next set of hyperparameters is chosen from the surrogate function" (Koehrsen, 2018).

Apart from the selection function, it is important to determine the surrogate function. There are several different forms of the surrogate function, but in this research, the focus will be on the Tree-Structured Parzen Estimator (TPE), described in (Bergstra, Daniel Yamins, and D. Cox, 2013). Hyperparameter optimisation will find a better set of hyperparameters after each iteration, and, therefore, the machine learning algorithm improves after each iteration. The search space from which these hyperparameters are sampled has to be defined before the Bayesian optimisation begins. These distributions have to be set manually, just as was done with the random search.

Bayesian optimisation is an efficient technique because it chooses the set of hyperparameters in an informed way. Therefore, this technique spends a little bit more time selecting the hyperparameters. However, the machine learning algorithm does not have to be run as much to get the same performance as the random search. This will decrease the running time massively.

Next to this, Bayesian optimisation improves after each iteration. Because of this, this technique has a higher probability of finding the optimal hyperparameters. This is visible in figure 14:

Figure 14: Optimisation of the test set on two different data sets with a random search and a Tree-Structured Parzen Estimator (TPE) (Bergstra, Daniel Yamins, and D. Cox, 2013)



The results of a random search and a Tree-Structured Parzen Estimator (TPE) on two data sets are visible in this figure. The idea of the TPE is similar to the Bayesian optimisation, where Bayesian optimisation tries to figure out $P(\text{Score}|\text{Hyperparameters})$, and a TPE models $P(\text{Hyperparameters}|\text{Score})$ and $P(\text{Hyperparameters})$ (Bissuel, 2019).

By using the Bayesian Theorem and describing the conditional probability $P(\text{Score}|\text{Hyperparameters})$ as a proportional quantity, it is clear that:

$$P(\text{Score}|\text{Hyperparameters}) = P(\text{Hyperparameters}|\text{Score}) \cdot P(\text{Hyperparameters})$$

Therefore, the idea of the TPE is similar to the Bayesian optimisation. The grey dots in figure 14 denote the lowest error observed among N random trials (as N increases to the right), while the green dots represent the lowest error observed within the first N suggestions of the TPE algorithm.

In this figure, it becomes clear that the validation error of the TPE is lower than the validation error of the random search for most values of N. "The TPE has discovered the best-known model configuration in the search space within 750 trials, but our 2000-trial random search has not come close." (Bergstra, Daniel Yamins, and D. Cox, 2013)

Therefore, it is clear that the TPE outperforms the 2000-trial random search (D. Cox and Pinto, 2011) on the left or the 15,000-trial random searches on the right (Pinto et al., 2011).

8.5.4 Optimal hyperparameters

In the previous subsections, it becomes clear that the Bayesian optimisation technique is the best technique for optimising the hyperparameters of a machine learning algorithm. Therefore, this technique is used to find the optimal values for the hyperparameters in our machine learning algorithms.

The *Python* library *Hyperopt* is used to execute the Bayesian optimisation (Bergstra, Dan Yamins, D. D. Cox, et al., 2013). This library uses the TPE that was explained in the previous subsection to find the optimal hyperparameters.

The optimal hyperparameters have to be found for the neural network, the random forest, and the gradient boosting tree. The neural network is run with two different activation functions, which means that the Bayesian optimisation is done twice for the neural network. For every algorithm, the Bayesian optimisation is iterated 500 times in total to make sure that the results for the hyperparameter values are (almost) optimal. The results of this Bayesian optimisation technique is given in table 9.

Table 9: Optimal hyperparameters

Artificial neural network ReLU function	**Artificial neural network Swish function**	**Random forest**	**Gradient boosting tree**
Batch size: 225	Batch size: 225	Trees: 327	Trees: 376
Epochs: 141	Epochs: 52	Depth: 28	Depth: 12
Hidden layers: 2	Hidden layers: 3	Min leaf samples: 2	Min leaf samples: 3
Units: 90	Units: 191	Min split samples: 2	Min split samples: 6
Optimiser: 'adam'	Optimiser: 'adam'		Learning rate: 0,043

These are the hyperparameters that are used to run our algorithms and to achieve the results that are shown in *Section 9*.

8.6 Variable importance

Although it is nice to have a model that predicts the market impact well, for PGGM the most important aspect is to see how the model works, and to know which variables influence market impact most significantly. Therefore, PGGM is likely to prefer a model that is explainable over a model with a higher accuracy that is not explainable.

In *Section 7.2.1* it was stated that an advantage of a random forest is the fact that it can find the importance of the variables in the model. There are multiple ways in which the importance of variables can be found. These ways are explained in the subsections below.

8.6.1 Gini importance

The software machine learning library *Scikit learn* is used to build the random forest. With this library, it is possible to see the importance of each variable. This is done by looking at the Gini importance (mean decrease impurity) of each variable.

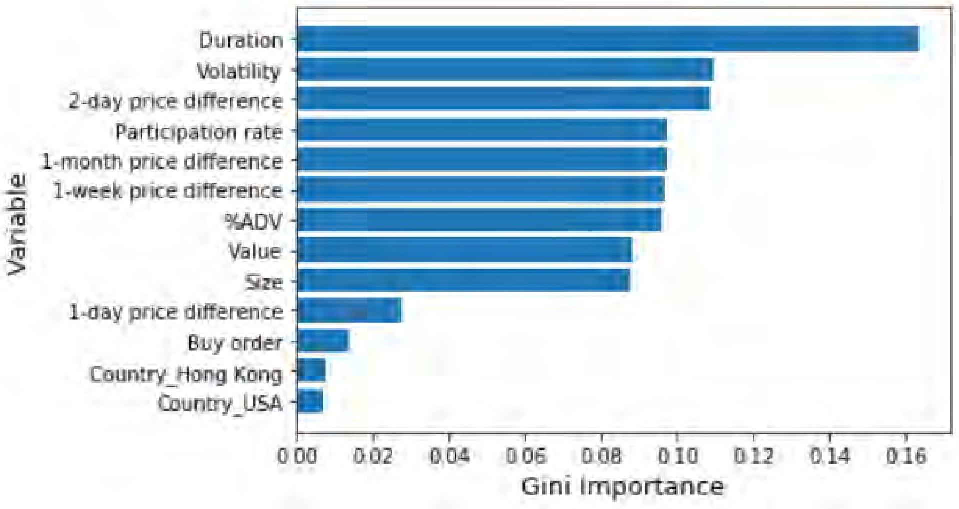
As stated in *Section 7.2.1*, the squared error function is used to determine how much accuracy is lost during a split. Therefore, the impurity reduction for a variable is calculated by looking at the MSE of a node in a tree before the split and the MSE of the nodes in this tree after the split.

After this is done, the importance of each variable is normalized. This means that every variable's importance is divided by the importance of all variables summed up together. Therefore, the importance of variables is given in percentages. If the importance's of the variables is summed up, they result in one (as it is normalized).

The variable importance than shows how much each variable contributes to the total impurity reduction of a tree. In *Section 7.2.1*, it was further stated that a tree chooses to split on the variables for which the costs of the split have the lowest value and that this process is continued until all variables are in the tree. This means that variables that tend to split nodes closer to a tree's root will have higher variable importance. The average of the variable importance for each tree is taken to determine the variable importance of the random forest.

As stated before, the Gini importance is a build-in feature of the *Skicit* library. Because of this, all values are computed during the time that the random forest is trained. Therefore, this method's most significant advantage is that there is no extra computation time necessary to determine the Gini importance for the variables. This is also the measure of variable importance that the RFECV uses, which was explained in *Section 8.4.2*. The results of the Gini importance of the variables in our data set are shown in figure 15.

Figure 15: All input variables ordered by Gini importance



According to the Gini importance, duration has the biggest effect on the market impact. In this research it became clear that a stock's price movement during the time interval of an order has the biggest influence on the market impact. A longer time interval means that a stock has more time to fluctuate, which means that it has a higher impact on the market impact.

Volatility is another variable that influences a stock's price movement during the time interval of an order. A higher volatility means that a stock fluctuates more, which means that it has a higher impact on the market impact.

The momentum indicators have a high Gini importance as well. The only momentum indicator that negatively stands out is the 1-day price difference. However, as stated in *Section 5*, this variable has a lot of zeroes as observations. Because if an order is executed when the market opens, the price of a stock has not moved yet. This happened in 13757 orders of the 18424 usable orders, which are a lot more orders than just the orders that are executed when the market opens. This indicates that there are more causes that this variable has zeroes as observations. Because this variable has a lot of zeroes as observations, there is not much information in this variable and it has a low Gini importance.

What further stands out is the participation rate. This variable is seen as one of the most important variables that influence the market impact, but it is only on the fourth place with the Gini importance. In *Section 4.3* it became clear that participation rate, duration and % ADV are connected. Duration can in fact be estimated by dividing % ADV by the participation rate. % ADV is the size of the order divided by the average daily volume of the stock, and this is something that cannot be altered by the traders. However, participation rate is something that can be altered by a trader. A higher participation rate will always shorten the duration time. Because duration has a significantly higher effect on the market impact compared to the participation rate, a first thought for PGGM could be to trade more aggressively and increase their participation rate. However, one needs to keep in mind that these variables are dependent on one another. Therefore, more investigation is needed on these variables to be able to make a good conclusion.

The rest of the variables do not really stand out. Value and size are variables that do influence the market impact, but not as much as the previously mentioned variables. Furthermore, it was expected that both countries and the buy order would have the lowest Gini importance. The reason for this is that there is no clear explanation on why these variables influence the market impact.

8.6.2 Permutation importance

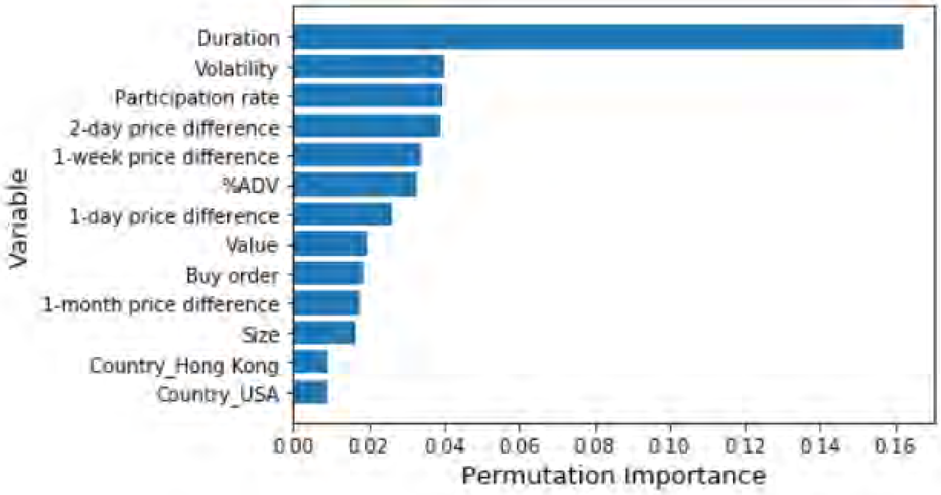
Permutation importance is another method that looks at the importance of the variables in a random forest. This method first evaluates the performance of the model. After this is done, this method shuffles the values of a variable $i \in M$ and reevaluates the model's performance. The decrease in performance is an indication of the importance of this variable. The values of this variable are then returned to their original places, and the values of the next variable $j \in M, j \neq i$ are randomly shuffled.

This process is repeated M times until the values of all variables are randomly shuffled. The performance of the random forest has then been reevaluated for all variables. Therefore, the observed mean decrease in performance of the random forest is an indication for the variable importance.

An advantage of this method is that it does not give correlated features high importance. If one correlated variable's value is randomly shuffled, the other correlated variable will still contain much information about this randomly shuffled correlated variable. The mean decrease in performance will not be high in this case. In *Section 8.4.2* it was explained that the variables that had a correlation higher than $|0.7|$ were already removed from the data set, such that there is no real worry about correlated features in our case.

A drawback of this method is that it takes a lot of computation power if the number of variables is large. Remember that each variable has to be randomly shuffled, and every time this is done, the model has to be reevaluated. The results of the permutation importance of the variables in our data set are shown in figure 16.

Figure 16: All input variables ordered by permutation importance



In this figure it becomes clear that with the permutation importance duration stands out more compared to the other variables. The difference between the participation rate and duration has increased, although participation rate now has the third highest ranking.

The variable one month price difference has decreased the most in permutation importance compared to the Gini importance. This variable went from the fourth most important variable to the ninth most important variable. One reason for this can be that this variable has a moderate correlation with the one week price difference variable. The information of the one week price difference is of course also visible in the one month price difference. However, the one week price difference has not decreased that much in permutation importance compared to Gini importance. Apparently the most important information in the one month price difference variable is the most recent price difference and this information is already given by the other momentum indicators.

The other values in this figure are quite similar compared to the values that were visible in figure 15.

8.6.3 Shapley values

Shapley values are values that are used in game theory to estimate how a player contributes to the outcome (Shapley, 1953). Shapley values look at the marginal contribution of players after all possible combinations have been considered. The main idea behind these values is to look at a team's performance with a particular player and subtract this from a team's performance without a particular player. This is the impact of this player. By looking at a player's marginal contribution, it is possible to determine the payout each player deserves. The way this works can be shown via an example. In this case, the glove game is used.

In this game, there are three players, N = 1, 2, 3. Player one and two both have a left glove, and player three has a right glove. The goal of this game is to form a pair of gloves. This is the case when player three and player one and/or two are in the game. The combination of players is given by $S \in \{\{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$. The value function V(S) looks as follows:

$$V(S) = \begin{cases} 1 & \text{if } S \in \{\{1,3\}, \{2,3\}, \{1,2,3\}\} \\ 0 & \text{otherwise.} \end{cases}$$

The marginal contribution of a player $i \in N$ is determined by: $V(S) - V(S - i)$. The Shapley value of player i can then be calculated via the following formula:

$$Sh_i = \sum_S \frac{(s-1)\cdot!(n-s)!}{n!}[V(S) - V(S-i)]$$

In this formula, s is the number of players in subset S, and n is the game's total number. The Shapley value of player 1 is therefore calculated in the following manner:

$$\begin{aligned} Sh_1 = w_1 \cdot (V(\{1\}) - V(\{\emptyset\})) + w_2 \cdot (V(\{1,2\}) - V(\{2\})) \\ + w_3 \cdot (V(\{1,3\}) - V(\{3\})) + w_4 \cdot (V(\{1,2,3\}) - V(\{2,3\}) \end{aligned}$$

The weights w_1, w_2, w_3 and w_4 in this formula are given by the first part of the Shapley formula: $\frac{(s-1)!\cdot(n-s)!}{n!}$. This gives the following weights: $w_1 = \frac{(1-1)!\cdot(3-1)!}{3!} = \frac{1}{3}$; $w_2 = w_3 = \frac{(2-1)!\cdot(3-2)!}{3!} = \frac{1}{6}$; and $w_4 = \frac{(3-1)!\cdot(3-3)!}{3!} = \frac{1}{3}$

The calculations for the value functions are given in table 10 below for player 1.

Table 10: Example of Shapley values for player 1

S	V(S)- V(S-i)
$\{1\}$	V($\{1\}$) - V($\{\emptyset\}$) = 0 - 0 = 0
$\{1, 2\}$	V($\{1, 2\}$) - V($\{2\}$) = 0 - 0 = 0
$\{1, 3\}$	V($\{1, 3\}$) - V($\{3\}$) = 1 - 0 = 1
$\{1, 2, 3\}$	V($\{1, 2, 3\}$) - V($\{2, 3\}$) = 1 - 1 = 0

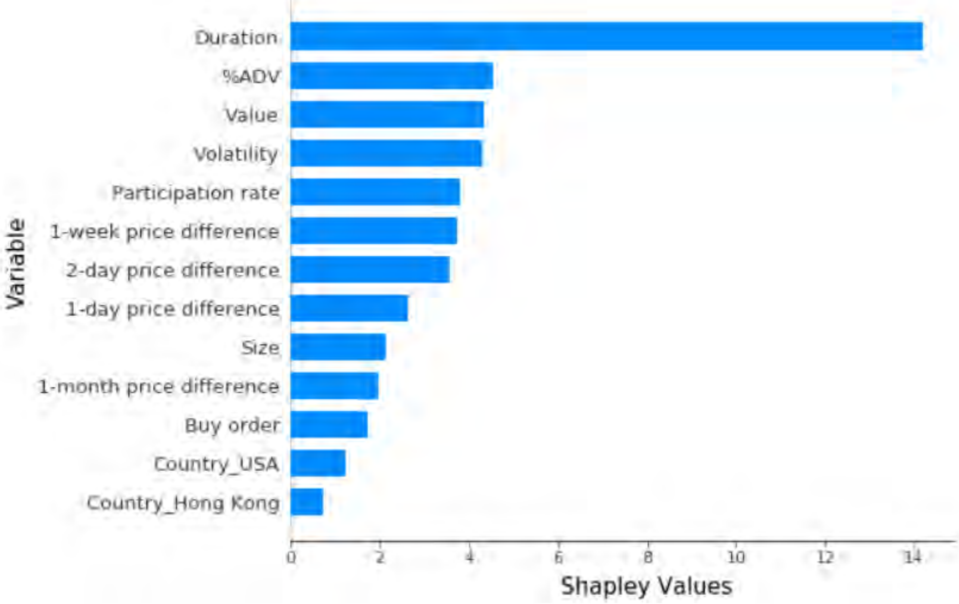
If the values of $w_j, j \in (1, 2, 3, 4)$ and $V(S) - V(S-1)$ are entered in the formula for Sh_1, this results in $Sh_1 = \frac{1}{6}$. When similar tables as table 10 are created for player 2 and 3, this gives us the results $Sh_2 = \frac{1}{6}$ and $Sh_3 = \frac{2}{3}$.

These Shapley values can also be used to determine the importance of variables in a random forest. Instead of estimating how a player contributes to the outcome, the Shapley value will estimate the marginal contribution of a variable to the model's prediction. In this case, the players of the game are the values of the variables.

The Shapley values are calculated by looking at how much each variable contributed to the prediction compared to the model's average prediction. This means that the Shapley values concur to every variable's contribution towards pushing the prediction away from the average value. The sum of all Shapley values is then the difference between predictions and the model's average value.

The average of the Shapley values gives a good indication about the contribution of a variable to the prediction of the model. This means that Shapley values are a good indicator for the importance of variables. The results of the Shapley values of the variables in our data set are shown in figure 17.

Figure 17: All input variables ordered by Shapley values



In this figure with the Shapley values duration stands out from the other variables once more. Therefore, it is plausible to say that duration is the most important variable for the market impact. Just a was the case with the permutation importance, the one month price difference variable is the ninth most important variable. This shows once more that more recent price differences have the most impact on a stock's price fluctuation, and hereby the market impact.

Furthermore, the variables % ADV and value have increased in importance and are now the most important variables besides duration. The other relative differences in this figure are quite similar compared to the relative differences that were visible in figure 15 and 16.

9 Results

In this section results of our four machine learning algorithms are represented. The predictions of the market impact that these algorithms return are compared with the actual market impact. This comparison is done via the MAE and the R-squared that were discussed in *Section 6*.

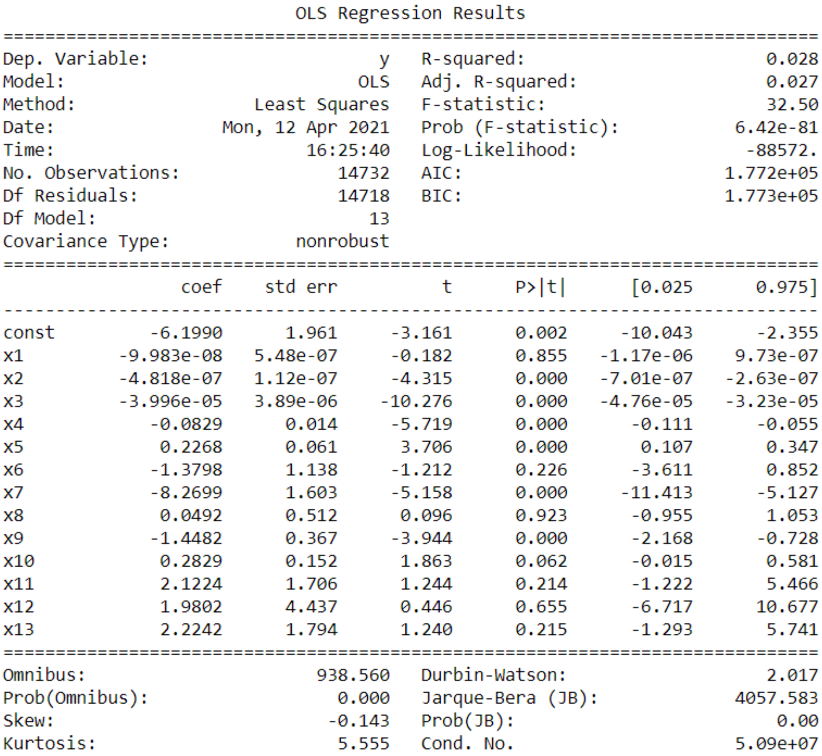
9.1 Multiple linear regression

The multiple linear regression model is implemented in *Python* via the library *Sklearn*. At first, the multiple linear regression is run with the output variable market impact and all thirteen input variables that were selected in *Section 8.4.2* by the RFECV. Therefore, the model looks as follows:

$$\begin{aligned}\text{Market Impact} = {} & x_1 \cdot \text{Size} + x_2 \cdot \text{Value} + x_3 \cdot \text{Duration}\\ & + x_4 \cdot \text{ADV} + x_5 \cdot \text{Participation} + x_6 \cdot \text{Volatility}\\ & + x_7 \cdot \text{1-Day Price Difference} + x_8 \cdot \text{2-Day Price Difference}\\ & + x_9 \cdot \text{1-Week Price Difference} + x_{10} \cdot \text{1-Month Price Difference}\\ & + x_{11} \cdot \text{Buy-order} + x_{12} \cdot \text{Hong Kong} + x_{13} \cdot \text{USA}\end{aligned}$$

The multiple linear regression model then gives estimations of the unknown coefficients $x_1, ..., x_{13}$. This can be made visible via the ordinary least squares (OLS) method. A summary of the OLS regression results is shown in the figure below:

Figure 18: Multiple linear regression model of the market impact with the thirteen input variables selected by RFECV



```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.028
Model:                            OLS   Adj. R-squared:                  0.027
Method:                 Least Squares   F-statistic:                     32.50
Date:                Mon, 12 Apr 2021   Prob (F-statistic):           6.42e-81
Time:                        16:25:40   Log-Likelihood:                -88572.
No. Observations:               14732   AIC:                         1.772e+05
Df Residuals:                   14718   BIC:                         1.773e+05
Df Model:                          13
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -6.1990      1.961     -3.161      0.002     -10.043      -2.355
x1         -9.983e-08   5.48e-07     -0.182      0.855   -1.17e-06    9.73e-07
x2         -4.818e-07   1.12e-07     -4.315      0.000   -7.01e-07   -2.63e-07
x3         -3.996e-05   3.89e-06    -10.276      0.000   -4.76e-05   -3.23e-05
x4            -0.0829      0.014     -5.719      0.000      -0.111      -0.055
x5             0.2268      0.061      3.706      0.000       0.107       0.347
x6            -1.3798      1.138     -1.212      0.226      -3.611       0.852
x7            -8.2699      1.603     -5.158      0.000     -11.413      -5.127
x8             0.0492      0.512      0.096      0.923      -0.955       1.053
x9            -1.4482      0.367     -3.944      0.000      -2.168      -0.728
x10            0.2829      0.152      1.863      0.062      -0.015       0.581
x11            2.1224      1.706      1.244      0.214      -1.222       5.466
x12            1.9802      4.437      0.446      0.655      -6.717      10.677
x13            2.2242      1.794      1.240      0.215      -1.293       5.741
==============================================================================
Omnibus:                      938.560   Durbin-Watson:                   2.017
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4057.583
Skew:                          -0.143   Prob(JB):                         0.00
Kurtosis:                       5.555   Cond. No.                     5.09e+07
```

The most important values in this figure of the OLS regression results are the p-values of the variables and the R-squared of the model. In figure 18 it becomes clear that the R-squared for the multiple linear regression is low and has a value of 0,028.

A variable is significant when the p-value is below 0.05. This means that there is less than a five per cent probability that the relationship between this input variable and the output variable is caused by chance. The p-values are visible in the fourth column of figure 18.

By looking at the p-values of the variables it becomes clear that there are six significant variables, x_2, x_3, x_4, x_5, x_7, and x_9. These six variables represent value, duration, ADV, participation rate, volatility, and one week price difference.

The machine learning algorithm has added a constant, apart from the input variables. A constant gives information about the market impact that is not given by the input variables in the model. In figure 18 it is visible that this constant is significantly different from zero. If the model would perfectly predict the market impact, this constant would have a value of zero. After inserting the values of the unknown coefficients the market impact model looks as follows:

$$\begin{aligned}\text{Market Impact} = {} & -6.1990 - 9.983 \cdot 10^{-8} \cdot \text{Size} - 4.818 \cdot 10^{-7} \cdot \text{Value} - 3.996 \cdot 10^{-5} \cdot \text{Duration} \\ & - 0.0829 \cdot \text{ADV} + 0.2268 \cdot \text{Participation} - 1.3798 \cdot \text{Volatility} \\ & - 8.2699 \cdot \text{1-Day Price Difference} + 0.0492 \cdot \text{2-Day Price Difference} \\ & - 1.4482 \cdot \text{1-Week Price Difference} + 0.2829 \cdot \text{1-Month Price Difference} \\ & + 2.1224 \cdot \text{Buy-order} + 1.9802 \cdot \text{Hong Kong} + 2.2242 \cdot \text{USA}\end{aligned}$$

From this model, it becomes clear that the participation rate variable has a positive coefficient, which means that the participation rate has a positive influence on the market impact. A higher participation rate would then thus give a more positive market impact. This does not concur with the theory of this research.

This model gives a positive coefficient value for some momentum indicators and a negative value for others. In *Section 5* it became clear that positive numbers in the price difference variables mean that the stock was moving in a favourable direction. Because of this, it would be logical that these momentum indicators have positive coefficients. The negative coefficients have a higher absolute value compared to the positive coefficients, so in total, the momentum indicators will have a positive effect on the market impact in this model, which concurs with the theory of this research. There is no logical explanation for the fact that some momentum indicators have a positive coefficient and some momentum indicators have a negative coefficient, apart from the fact that the model does not work well because these variables are correlated.

Furthermore, the low coefficient for size, value, and duration stands out. The reason for this is that these variables have high values, which causes that the coefficients of these variables are a bit lower. This does not mean that these variables have a less significant effect on the market impact.

The MAE of this market impact model compared to the actual market impact is 75.45, which is slightly lower than the MAE of the Bloomberg expected market impact compared to the actual market impact.

It is possible to remove the insignificant variables to try to improve the model. However, the MAE and the R-squared did not decrease after removing these variables. Therefore, our multiple linear regression model is not updated, and it is still the equation above.

9.2 Artificial neural network

An artificial neural network is another example of a parametric machine learning model. Just as was the case with the multiple linear regression model, the artificial neural network is run with all thirteen selected variables in *Section 8.4.2* by the RFECV. These are the variables that influence the market impact according to the RFECV.

The *Python* library *Tensorflow* is used to build the artificial neural network. In *Section 7.1.2* it was discussed that there are two activation functions applicable when predicting the market impact with an artificial neural network, the ReLU function and the Swish function.

9.2.1 ReLU function

At first, the artificial neural network is run with the ReLU function as an activation function. The prediction of the artificial neural network with the ReLU function as activation function has a MAE of 73,49, and a R-squared of 0,0519. This is only a slight improvement compared to the simple multiple linear regression model.

As stated in *Section 7.1.2*, the ReLU function is bounded by zero for negative values and unbounded for positive values. Because of this, the artificial neural network cannot perform back-propagation with negative values, and cannot learn from negative values in the input variables. Therefore, it seems like the ReLU function is not the best activation function to predict the market impact.

9.2.2 Swish function

After the artificial neural network was run with the ReLU function as an activation function, the Swish function is used as an activation function. As stated in *Section 7.1.2*, the Swish function is, just like the ReLU function, bounded for negative values and unbounded for positive values. The Swish function is however smooth and non-monotonic and can, therefore, still perform backpropagation when input variables are negative.

The prediction of the artificial neural network with the Swish function as activation function has a MAE of 73,05, and a R-squared of 0,0582. This is only a slight improvement compared to the artificial neural network with the ReLU function.

9.3 Random forest

A random forest is an example of a non-parametric machine learning model, which means that the input variables do not need to be selected before the algorithm is run. These models do not make strong assumptions about the form of the function and can select the most important variables themselves. The RFECV used this property to find the thirteen most important variables in the data set. These variables were used by the non-parametric models as well.

The *Python* library *Sklearn* is used to build the random forest. The prediction of the random forest has a MAE of 66,05 and a R-squared of 0,1377. This is a huge improvement compared to the multiple linear regression model.

Although the random forest is not a black box model it is not possible to show the exact relationship between the variables. In *Sections 8.6.1, 8.6.2, and 8.6.3* three methods were explained that show the importance of variables in this random forest. The importance of the variables for these methods are given in tables 15, 16, and 17.

Furthermore, it is possible to show one of the decision trees in the random forest. However, this would not give us any information as there are 327 different decision trees in this random forest.

9.4 Gradient boosting tree

A gradient boosting tree is another example of a non-parametric machine learning model. Just as was the case with the previous algorithms, the Gradient boosting tree is run with all thirteen variables that were selected in *Section 8.4.2* by the RFECV.

The *Python* library *Sklearn* is used to build the gradient boosting tree. The prediction of the gradient has a MAE of 67,05 and a R-squared of 0,0971. Therefore, the gradient boosting tree is not an improvement compared to the random forest. Just like is the case with the random forest, it is not possible to show the exact relationship between the variables.

The most significant difference compared to the decision tree is that a gradient boosting tree is that this algorithm learns from the decision trees it has created, and builds new decision trees with this information, while a random forest builds these trees randomly. Because of this, one would expect that the gradient boosting tree is a better model compared to the random forest.

However, in *Section 7.2.2* it became clear that the gradient boosting tree is more sensitive to overfitting if the data is noisy. In this research, it became clear that the most significant part of the market impact is a stock's price movement during the time interval of an order. In *Section 8.4.2* it was stated that this might even be described as noise because it is almost impossible to predict a stock's price movement. This is the reason that the random forest outperforms the gradient boosting tree.

9.5 Looking at the subsets in the data

It remains difficult to accurately predict the market impact. The machine learning algorithm that performed best was the random forest. However, the random forest model still had a MAE of 66,05, and a R-squared of 0,1377. This difference is still too high to say that the random forest model can accurately predict the market impact.

In *Section 6* it already became clear that it would be hard to accurately predict the market impact. However, after looking at table 4 it became clear that the R-squared in Bloomberg's model increased if the participation rate and % ADV increased.

The reason for this is that large and aggressive trades have a higher impact on the market, and this makes it is easier for a market impact model to accurately predict the market impact. If the participation rate is sufficient an order creates momentum itself, which makes it easier to predict the stock's price movement during the time interval of an order.

The two variables that most influence the fluctuations of the stock price during the time interval of the order are the volatility of the stock and the duration of the order. The stock's price fluctuation is the most significant part of the market impact. Therefore, it is helpful to take a deeper look into the two variables that have the most influence on the fluctuations of the stock price.

In total, four variables will be looked into to understand the results that are given in this section. These four variables are the participation rate, % ADV, volatility, and duration. These four variables are split into four subsets to see if it is easier to predict the market impact for some of these subsections. These subsections are all approximately the same size to be able to make a fair comparison between the four subsets.[9]

The first subset then consists of the smallest quarter of values in that variable, the second subset consists of the second smallest quarter of values in that variable, the third subset consists of the second-largest quarter of values in that variable, and the fourth subset consists of the largest quarter of values in that variable.

For each subset, the random forest algorithm is run. This means that the random forest will create a different model for every subset in the data.

9.5.1 Participation rate

The first variable that is divided into four subsets is the participation rate. The prediction of the random forest model for the subset that contains the smallest quarter of values for the participation rate has a MAE of 72,11 and a R-squared of 0,1181, the second quarter has a MAE of 65,16 and a R-squared of 0,1452, the third quarter has a MAE of 66,34 and a R-squared of 0,1373, and the fourth quarter has a MAE of 58,63 and a R-squared of 0,1632. On average, there is indeed a decrease in the MAE when the participation rate increases, and an increase in the R-squared. Therefore, it seems correct to assume that aggressive orders with a high participation rate create momentum themselves, and these orders are easier to predict because of this.

[9] All subsets are equally split, which means that they all contain $\frac{18424}{4} = 4606$ orders.

The MAE of Bloomberg's prediction decreases from 84,01 to 70,50 from the first to the last subset, while the R-squared decreases from -0,0034 to -0,1391. Therefore, their theory is correct for their model as well. This confirms that aggressive trades with a high participation rate are easier to predict than non-aggressive trades with a low participation rate.

9.5.2 % ADV

The second variable that is divided into four subsets is the % ADV. The prediction of the random forest model for the subset that contains the smallest quarter of values for the % ADV has a MAE of 64,72 and a R-squared of 0,0794, the second quarter has a MAE of 68,38 and a R-squared of 0,1267, the third quarter has a MAE of 71,57 and a R-squared of 0,1047, and the fourth quarter has a MAE of 77,16 and a R-squared of 0,1108. On average, there is thus an increase in the MAE when the % ADV increases. The R-squared fluctuates a bit but has on average a slight increase when the % ADV increases.

A higher MAE does not necessarily mean that the R-squared is lower for these different subsets. The reason for this is that these subsets all have a different average market impact. The smallest quarter of values for the % ADV consist of the orders for which PGGM's share is the smallest, and because of this, the average market impact in this subset is lower. The reason behind this is that large trades have a higher impact on the market. If the average market impact diminishes the MAE of the random forest will automatically be lower as well. Therefore, there is more information in looking at the R-squared for these different subsets.

The R-squared has a slight increase when the % ADV increases. This concurs with table 4 where it became clear that the R-squared of Bloomberg's model increased when % ADV increased. Because significant orders have a higher impact on the market, the market impact is easier to measure, and a market impact model should be able to predict the market impact more accurately with a higher % ADV. This is indeed visible in the increase of the R-squared. However, this difference is not as significant as was visible in table 4.

The MAE of Bloomberg's prediction increases from 70,66 to 88,27 from the first to the last subset, while the R-squared decreases from -0,0087 to -0,0870. Therefore, their theory is not correct for their model. Our random forest model already showed that the difference is not as significant as was visible in table 4. The R-squared and MAE of Bloomberg's prediction confirm that this is indeed the case and that it might be the case that it is not easier to predict large orders.

9.5.3 Volatility

The third variable that is divided into four subsets is volatility. The prediction of the random forest model for the subset that contains the smallest quarter of values for the volatility has a MAE of 45,98 and a R-squared of 0,1625, the second quarter has a MAE of 70,83 and a R-squared of 0,1584, the third quarter has a MAE of 73,79 and a R-squared of 0,0623, and the fourth quarter has a MAE of 84,50 and a R-squared of 0,0523. On average, there is indeed a decrease in the MAE when the volatility increases, and an increase in the R-squared when volatility increases.

The MAE of Bloomberg's prediction increases from 54,04 to 102,60 from the first to the last subset, while the R-squared decreases from -0,0068 to -0,0535. The accuracy of Bloomberg's prediction decreases thus as well when volatility increases.

9.5.4 Duration

The fourth variable that is divided into four subsets is the duration. The prediction of the random forest model for the subset that contains the smallest quarter of values for duration has a MAE of 41,74 and a R-squared of 0,1687, the second quarter has a MAE of 61,58 and a R-squared of 0,1652, the third quarter has a MAE of 70,90 and a R-squared of 0,1368, and the fourth quarter has a MAE of 86,59 and a R-squared of 0,0521. On average, there is indeed a decrease in the MAE when the duration increases, and an increase in the R-squared when the duration increases.

The MAE of Bloomberg's prediction increases from 51,73 to 105,90 from the first to the last subset, while the R-squared increases from -0,0309 to -0,1023. Therefore, the accuracy of Bloomberg's prediction decreases when the duration increases. Duration and volatility both influence a stock's price movement during the time interval of an order. The significant decrease in performance for the random forest and Bloomberg's prediction once more shows how significant the influence of a stock's price movement is on the market impact.

10 Discussion

This research aims to see if it is possible to predict the transaction costs of orders in stock portfolios. Transaction costs are a relatively unknown concept in the financial world, and it is avoided at most financial courses in the universities. This meant that it was necessary to do extensive research related to transaction costs.

During this research, it became clear that the most substantial unknown part of transaction costs is the market impact of an order. Other parts like fees and taxes are well known before an order is executed.

After looking at the data and comparing Bloomberg's market impact prediction with the actual market impact, it became clear that this would not be an easy task. Bloomberg's market impact prediction is currently used by PGGM as a benchmark for the realized market impact; they state that an order has had out-performance if the actual market impact is lower than the predicted market impact, and an order has had under-performance if the actual market impact is higher than the predicted market impact. However, their prediction was often substantially deviated from the actual market impact and had a negative R-squared.

Later in the research, it became clear that the market impact was hugely dependant on a stock's price movement during the time interval of the order. Because of this, the market impact is quite variable and difficult to predict. Bloomberg itself stated that roughly 83% of the market impact is caused by the stock's price movement. At the beginning of this research, it became clear that the price momentum of the past correlates with the price momentum of the future. Therefore, momentum variables were used to predict the stock price movement, and, hereby, the market impact. This is a new strategy for predicting the market impact.

An advantage was that there was enough data available for this research. The data set that was used during this research consisted of 18768 orders and forty-four variables. At first, the unusable orders were removed, and after splitting some of the variables into dummy variables a selection for the most essential variables was done using RFECV. In total 18424 orders and thirteen variables were used to predict the market impact. Four of those variables were momentum indicators. These variables had thus indeed influence on the market impact.

Four different machine learning models were used in this research. The performances of these models were evaluated to get an answer to our research question:

Is it possible to predict the transaction costs of orders in stock portfolios?

Our best model was the random forest model. Although this model was an improvement on Bloomberg's model, the MAE was still 66,05 and the R-squared 0,1377. These values show that the random forest model can predict the market impact to some extent and show that the model is an improvement on just taking the average. However, this model cannot be used to predict the market impact accurately. Therefore, a portfolio manager will not be able to use this model to help him with his trading decisions.

After dividing the data into smaller subsets it became easier for our random forest model to predict the market impact in some of these subsets. In the sixteen subsets that were created, it became clear that our random forest models outperformed Bloomberg's model for all these subsets. The best prediction for the random forest was visible in the subset that contained the smallest quarter of values for the duration. Besides the fact that the price of a stock fluctuates less when there is a short duration, it may be expected that if the duration is smaller any method will be better since one needs to predict less far in the future. However, even for this subset the values of the MAE and R-squared show that the random forest model cannot accurately predict the market.

Further research is needed to see if the rules for these subsets could be made tighter, such that more subsets can be created from the same variables. Another option can be to combine some of the variables that were used to make these subsets, and to make different subsets with this combination of variables. This would make it easier to predict the market impact for some of these subsets. However, because the MAE was still high and the R-squared was still low for every subset, it is still not likely that our random forest model can then accurately predict the market impact. This is something that has to be investigated further. If there is more data available the rules for these subsets can be made very tight.

One more thing that needs to be taken into consideration is the fact that the RFECV technique uses the random forest algorithm to make a selection for the most important variables. This means that some of the removed variables might have been useful to one of the other algorithms, but were still removed because the variable was not used by the random forest algorithm. This might also have been one of the reasons that the random forest performed best.

Some variables carried not as much information as if they were complete. Firstly, there was the bid-ask spread. The bid-ask spread influenced the market impact according to the theory that was discussed in *Section 4*, the first Bloomberg (Ferraris, 2008) model that was discussed in *Section 6.1*, and the second Bloomberg model (Rashkovich and Verma, 2012) that was discussed in *Section 6.2*. However, the bid-ask spread was removed by the RFECV because this variable did not influence the prediction of the market impact in the random forest. In our data set, the bid-ask spread variable gave the values between which the bid-ask spread lied before the order was executed, and not the exact bid-ask spread value. Therefore, there is less information in this variable. This could have been the reason that the bid-ask spread did not influence the prediction of the market impact. If this variable would have had the exact bid-ask spread value this variable might have been useful to predict the market impact, which only could have increased the accuracy of our model. Therefore, it is useful for PGGM to note the precise bid-ask spread of stocks at order execution time.

Secondly, there was a price difference one day before the order. This variable only contained information in 4667 orders of the 18424 usable orders. This variable contained the most recent information on a stock's price movement, which makes it unfortunate that there was not much information in the variable. Besides that, it would be helpful to have more recent information on the price differences. One could, for example, make the variables, besides the already existing momentum indicators, price difference one hour before the order, two hours before the order, and four hours before the order. This would give more information on the most recent price movement of a stock. If a stock is then bought in the opening hours of the market, the price difference of the last hours in the previous day is also included. Adding these variables can only help with the model's prediction. This should be investigated further in future research.

During this research, the focus has been on predicting the transactions costs of orders in stock portfolios. PGGM pays every year around 185 million on transaction costs with their stock orders. These costs naturally hurt the return of PGGM's investments. However, one needs to keep in mind that an order with high (low) transaction costs is not necessarily an order that is badly (well) executed.

Transaction costs are determined by looking at the stock's price before an order is executed and the average execution price paid for this stock. A buy-order has relatively high transaction costs if the price of that stock increased significantly during the time interval of that order. However, this order is still well-executed if the price continued to increase after the order was completed. Namely, it would not have been possible to have a much lower average execution price. Of course, it might have been profitable if the order was executed quicker, but this is not necessarily the case since the price might have increased more during the time interval of that order due to the higher participation rate. If the price decreased after the order was completed it is correct to say that the order could have been executed in a better way. The average execution price would then have been lower if the participation rate was lower and the order was spread over a more extended period.

In our data set, there was no information on a stock's price movement after an order is executed. Information on this would be useful for the traders to see which trades are executed well, and which are not. This can help them with their decision process in the future, which will help to lower PGGM's transaction costs.

In this research, no investigation was done about which steps PGGM could take to lower their transaction costs. However, in this research three techniques were given that show the importance of variables. With all three techniques duration was the most important variable that determines the market impact. At a first sight, this is the variable that should be lowered. If the size of the orders then stays the same, the participation rate has to increase. More research should be done into the connection between the participation rate, the duration and the market impact of an order. This research should then also include the price difference shortly after the order is completely executed, as stated above. This information will help to make visible how transaction costs can be lower.

One simple suggestion to lower the transaction costs would be to accept a small deviation from the benchmark. PGGM then has to re-balance their portfolios less often, which means that they have to trade less often. This will automatically mean that PGGM will have lower transaction costs. However, more research needs to be done to investigate the consequences this tactic will bring.

The RFECV and Bayesian optimisation are two techniques that were used to optimise our models. However, these techniques are dependent on one another. The RFECV is dependent on the hyperparameters that the model uses, while the Bayesian optimisation is dependent on the set of variables that the model uses. Therefore, both techniques were applied three times. Our random forest could have been optimised further if these techniques were applied more often, but this would have taken too much time. Besides, the added value that this would have brought is not significant.

In this research, machine learning techniques were used to predict the market impact. There may be other techniques that can be used for this problem that works better. Another option is to set the restriction for outliers less strict, such that more outliers are removed. This will make it easier for the machine learning algorithms to accurately predict the market impact.

A downside for this is that more actual orders are removed from the data set. As unusual as these values may be, these orders all carry information about the market impact. Therefore, removing these variables might improve our models, but this is not necessarily desirable.

11 Conclusion

Although our random forest model outperforms Bloomberg's model on our test set data, this does not necessarily mean that our model can predict the market impact of all stocks more accurately compared to Bloomberg's model. It is plausible that PGGM invests in stocks that have specific characteristics. Our random forest model learns from these characteristics and can predict the market impact of these stocks, unlike Bloomberg's model that is not based on a specific set of stocks.

However, it seems not correct for PGGM to use Bloomberg's model as an evaluation of over-performance or under-performance for their orders. The negative R-squared showed that Bloomberg's model performed worse than a horizontal line of the average actual market impact. Our random forest model performed better, and with the suggestions given in *Section 10* the random forest can only improve. Therefore, this model is a better evaluation of over-performance or under-performance for PGGM's orders. This only indicates how much more further research is needed on the subject, as the present models lack performance. As stated in *Section 10*, a better evaluation of over-performance or under-performance can be created by looking at a stock's price movement after an order is executed.

The model that performed best in our research was the random forest model. However, the prediction of this model still had a MAE of 66,05 and a R-squared of 0,1377. The high MAE and low R-squared show that it is not possible to accurately predict the market impact of all orders.

After dividing the data into different subsets it became clear that the random forest could predict the market impact of orders in these subsets more accurately. The best prediction for the random forest was visible in the subset that contained the smallest quarter of values for the duration. The prediction of the random forest model then had a MAE of 41,74 and a R-squared of 0,1687. Although this is an improvement, these values show that it is still not possible to accurately predict the market impact of orders in stock portfolios. Even if the rules for the subsets are made tighter, or multiple variables are combined to make smaller subsets, it is not possible to predict the market impact of orders in stock portfolios in such a way that portfolio managers can consider these costs when they make their trading decisions.

However, the positive R-squared does show that it is possible to predict the transaction costs of orders in stock portfolios to some extent. Our random forest is an improvement compared to taking the average of the actual market impact. As stated, transaction costs also consist of implicit costs that are known before an order is executed. Therefore, this part can be accurately predicted. This makes it somewhat possible to predict the total transaction costs.

References

Barnston, Anthony G (1992). "Correspondence among the correlation, RMSE, and Heidke forecast verification measures; refinement of the Heidke score". In: *Weather and Forecasting* 7.4, pp. 699–709.

Bazan - Krzywoszańska, Anna and Bereta, Michał (Dec. 2018). "The use of urban indicators in forecasting a real estate value with the use of deep neural network". In: *Reports on Geodesy and Geoinformatics* 106. DOI: `10.2478/rgg-2018-0011`.

Bergstra, James and Bengio, Yoshua (2012). "Random search for hyper-parameter optimization." In: *Journal of machine learning research* 13.2.

Bergstra, James, Yamins, Dan, Cox, David D, et al. (2013). "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms". In: *Proceedings of the 12th Python in science conference*. Vol. 13. Citeseer, p. 20.

Bergstra, James, Yamins, Daniel, and Cox, David (2013). "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures". In: *International conference on machine learning*. PMLR, pp. 115–123.

Berry, Michael and Linoff, Gordon (1997). *Data mining techniques: For marketing, sales and marketing support*.

Bissuel, Aloïs (2019). *Hyper-parameter optimization algorithms: a short review*. Available at `https://medium.com/criteo-engineering/hyper-parameter-optimization-algorithms-2fe447525903`.

Breiman, Leo (1996). "Bagging predictors". In: *Machine learning* 24.2, pp. 123–140.

Brownlee, Jason (2020). *Hyperparameter Optimization With Random Search and Grid Search*. Available at `https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/`.

Burnham, Kenneth P and Anderson, David R (2004). "Multimodel inference: understanding AIC and BIC in model selection". In: *Sociological methods & research* 33.2, pp. 261–304.

CorporateFinanceInstitute (2015). *Systematic Risk*. Available at `https://corporatefinanceinstitute.com/resources/knowledge/finance/systematic-risk/`.

Cox, David and Pinto, Nicolas (2011). "Beyond simple features: A large-scale feature search approach to unconstrained face recognition". In: *Face and Gesture 2011*. IEEE, pp. 8–15.

De Jong, Frank and Rindi, Barbara (2009). *The microstructure of financial markets*. Cambridge University Press.

Everitt, Brian and Skrondal, Anders (2002). *The Cambridge dictionary of statistics*. Vol. 106. Cambridge University Press Cambridge.

Fabozzi, Frank J, Gupta, Francis, and Markowitz, Harry M (2002). "The legacy of modern portfolio theory". In: *The Journal of Investing* 11.3, pp. 7–22.

Fama, E. and Macbeth, J.D. (1973). "Risk, Return, and Equilibrium: Empirical Tests". In: *jstor.org*.

Fama, Eugene F and French, Kenneth R (1992). "The cross-section of expected stock returns". In: *the Journal of Finance* 47.2, pp. 427–465.

— (2015). "A five-factor asset pricing model". In: *Journal of financial economics* 116.1, pp. 1–22.

Ferraris, Andrew (2008). *Equity Market Impact Models*. Available at `http://dbquant.com/Presentations/Berlin200812.pdf`.

Giraud, Jean-René and d'Hondt, Catherine (2008). "Cash Equity Transaction Cost Analysis". In:

investopedia.com (2020). *Book-to-Market ratio definition*. Available at `https://www.investopedia.com/terms/b/booktomarketratio.asp`.

Jegadeesh, Narasimhan and Titman, Sheridan (1993). "Returns to buying winners and selling losers: Implications for stock market efficiency". In: *The Journal of finance* 48.1, pp. 65–91.

jpmorgen.com (2018). *Transaction costs explained*. Available at `https://am.jpmorgan.com/blob-gim/1383537981326/83456/JPM50934_MiFID\%20II\%20Transaction\%20Costs\%20Guide_A5_FINAL.pdf`.

Kenton, Will (2020). *Multiple Linear Regression (MLR)*. Available at `https://www.investopedia.com/terms/m/mlr.asp#:~:text=Key\%20Takeaways-,Multiple\%20linear\%20regression\%20(MLR)\%2C\%20also\%20known\%20simply\%20as\%20multiple,uses\%20just\%20one\%20explanatory\%20variable.`.

Kociński, Marek (2014). "Transaction costs and market impact in investment management". In: *e-Finanse: Financial Internet Quarterly* 10.4, pp. 28–35.

Koehrsen, Will (2018). *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning*. Available at `https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f`.

Kraus, Mike (2019). *Using Bayesian Optimization to reduce the time spent on hyperparameter tuning*. Available at `https://medium.com/vantageai/bringing-back-the-time-spent-on-hyperparameter-tuning-with-bayesian-optimisation-2e21a3198afb`.

Levenberg, Kenneth (1944). "A method for the solution of certain non-linear problems in least squares". In: *Quarterly of applied mathematics* 2.2, pp. 164–168.

Mangram, Myles E (2013). "A simplified perspective of the Markowitz portfolio theory". In: *Global journal of business research* 7.1, pp. 59–70.

Markowitz, Harry (1959). *Portfolio selection*.

MSCI (2021). *MSCI World Index (USD)*. Available at https://www.msci.com/documents/10199/178e6643-6ae6-47b9-82be-e1fc565ededb.

Nair, Vinod and Hinton, Geoffrey E (2010). “Rectified linear units improve restricted boltzmann machines”. In: *ICML*.

Park, Saerom, Lee, Jaewook, and Son, Youngdoo (2016). “Predicting market impact costs using nonparametric machine learning models”. In: *PLoS One* 11.2, e0150243.

PGGM.nl (2020). *PGGM numbers*. Available at https://www.pggm.nl/werken-bij/impact4/.

Pinto, Nicolas, Stone, Zak, Zickler, Todd, and Cox, David (2011). “Scaling up biologically-inspired computer vision: A case study in unconstrained face recognition on facebook”. In: *CVPR 2011 WORKSHOPS*. IEEE, pp. 35–42.

Ramachandran, Prajit, Zoph, Barret, and Le, Quoc V (2017). “Swish: a self-gated activation function”. In: *arXiv preprint arXiv:1710.05941* 7.

Rashkovich, Vlad and Verma, Arun (2012). “Trade cost: Handicapping on PAR”. In: *The Journal of Trading* 7.4, pp. 47–54.

Shapley, Lloyd S (1953). “A value for n-person games”. In: *Contributions to the Theory of Games* 2.28, pp. 307–317.

Tohka, Jussi and Gils, Mark (Aug. 2020). *Evaluation of machine learning algorithms for Health and Wellness applications: a tutorial.*

Vala, Kushal (2019). *Tree-Based Methods: Regression Trees*. https://towardsdatascience.com/tree-based-methods-regression-trees-4ee5d8db9fe9.

Verikas, Antanas, Vaiciukynas, Evaldas, Gelzinis, Adas, Parker, James, and Olsson, M. Charlotte (Apr. 2016). “Electromyographic Patterns during Golf Swing: Activation Sequence Profiling and Prediction of Shot Effectiveness”. In: *Sensors* 16, p. 592. DOI: 10.3390/s16040592.

Veronez, Mauricio, Souza, Sérgio, Matsuoka, Marcelo, Reinhardt, Alessandro, and Macedonio da Silva, Reginaldo (Dec. 2011). “Regional Mapping of the Geoid Using GNSS (GPS) Measurements and an Artificial Neural Network”. In: *Remote Sensing* 3. DOI: 10.3390/rs3040668.

Yegerman, Henry (2020). *Participation Rates and Trading Costs*. Available at https://insights.issgovernance.com/posts/participation-rates-and-trading-costs/.

Zhang, Zhongxing, Mayer, Geert, Dauvilliers, Yves, Plazzi, Giuseppe, Pizza, Fabio, Fronczek, Rolf, Santamaria, Joan, Partinen, Markku, Overeem, Sebastiaan, Peraita-Adrados, Maria, Silva, Antonio, Sonka, Karel, Río, Rafael, Heinzer, Raphael, Wierzbicka, Aleksandra, Young, Peter, Högl, Birgit, Bassetti, Claudio, Manconi, Mauro, and Khatami, Ramin (July 2018). “Exploring the clinical features of narcolepsy type 1 versus narcolepsy type 2 from European Narcolepsy Network database with machine learning”. In: *Scientific Reports* 8. DOI: 10.1038/s41598-018-28840-w.