Utrecht University

Predicting accounting liquidity at Dutch SMEs using Support Vector Regression and Multilayer Perceptrons

by

Joris Albers

Supervisors
Prof. dr. Yannis Velegrakis
V. (Vahid) Shahrivari Joghan MSc

Company Supervisor
Jeroen de Groot

Applied Data Science Master's Thesis
Department of Information and Computing Science
Utrecht University
Netherlands
July 2023

Abstract

Small to medium-sized enterprises (SMS) are critical for the global economy, but face challenges in liquidity management due to limited access to external financing and market uncertainties. This study systematically evaluates the effectiveness of multilayer perceptrons and support vector regression models in predicting liquidity for SMEs. Through multiple experiments conducted on a dataset of 496 companies, the study reveals potential for both models, although they exhibit a large number of limitations and sensitivity to data quality and size. Notably, the MLP model demonstrates a closer alignment to target values compared to SVR. While the models are not suitable for practical use, the findings highlight the importance of refining both models to improve liquidity forecasting. This research contribute to more effective decision making processes, benefitting long-term success and sustainability of SMEs as contributors of the economy.

1. Introduction

Small to medium-sized businesses (SMEs) are to great importance for the global economy. In 2021, about 22.8 million SMEs were active in the EU-27, these accounted to 99.8% of all enterprises in the non-financial business sector. These SMEs employ 64.4% in the NFBS and account to 51.8% of the added value. (European Commission, 2022) . It is impossible to overstate the significance of liquidity management for small to medium-sized businesses. Liquidity, as the foundation of financial stability, is essential to a company's ability to meet short-term obligations, maintain operational effectiveness, and invest in growth opportunities. A company that has enough liquidity can navigate through times of financial instability while avoiding the risks of insolvency, bankruptcy, and reputational harm. Since they frequently have limited access to external financing and are more vulnerable to market fluctuations and economic uncertainties, SMEs in particular face special challenges relating to liquidity management.

For SMEs, effective liquidity management is integral to their long-term success and sustainability. It helps them avoid cash flow disruptions, which can adversely affect their ability to pay suppliers, employees, and other stakeholders. In addition, a sound liquidity position enables SMEs to capitalize on strategic investment opportunities, positioning them for future growth and competitiveness. Consequently, research focused on improving liquidity forecasting and management for SMEs is of paramount importance, as it can provide valuable insights and tools to help these businesses better manage their financial resources. The accounting liquidity, the ease with which an individual or company can meet their financial obligations with the liquid assets available to them (Fernando, 2023), is an important predictor for companies. As a decline in liquidity is predictive for a greater risk of bankruptcy is small and medium sized firms (Pompe & Bilderbeek, 2005).

In this paper, we provide a systematic evaluation of liquidity forecasting models for SMEs. We test multilayer perceptrons (MLP), along with support vector regression (SVR) on their ability to predict financial data. Multiple experiments were conducted to examine how these models work on a dataset including 496 different companies. This study was done by conducting numerous experiments on the task.

We found that the models show promise in predicting liquidity for SMEs, although they exhibit certain limitations and sensitivity to the quality and size of the data used. Our results show that both the SVR and the multilayer perceptron MLP face challenges when predicting larger values. It was observed that the choice of the model does not significantly affected their predictive performance across different sectors, but the variability in their performance across different financial codes suggests further research could explore this direction. Interestingly, the study found that the MLP model showed a closer alignment to the target value compared to the SVR. While the models are not ready for practical use, these findings suggest that MLP, with further refinement has potential for improved performance in liquidity position.

This study serves as a stepping stone towards the development of more refined predictive models for liquidity management in SMEs, ultimately contributing to more effective decision-making processes in these vital contributors to the economy.

2. Related Work

The study on liquidity forecasting has been evolving over time. This has mainly been because of technological improvements. Early approaches mainly relied on financial ratios and simple statistical models to assess liquidity. As the market changed and new technologies emerged, researchers began to explore more sophisticated methods. Mramor and Valentincic (2003) proposed a model for forecasting the liquidity of very small private companies based on their financial statements and some key financial ratios, while Wisniewski (2008) developed a dynamic econometric model to assess the liquidity in small and medium enterprises. More recently, researchers have been examining the application of advanced machine learning techniques to liquidity forecasting. For instance, Wisniewski (2022) investigated the relationship between liquidity and debt recovery in small enterprises using an empirical system of interdependent equations.

While studies that focus on small enterprises have applied more econometric appliances on liquidity forecasting, other fields of financial forecasting have also been applying techniques in the field of machine learning. Cao and Tay (2001) used support vector machines to forecast financial timeseries data and Mahfoud & Mani (1996) used genetic algorithms to forecast the stock market. Weytjens et al. (2019) compared multiple methods to predict cash flow. In the beginning more "classic" methods were examined. These methods are ARIMA (Ho & Xie, 1998), which is an autoregressive integrated moving average and Facebooks' Prophet (Taylor & Letham, 2017), which tries to fit additive regression models, also known as 'curve fitting'. Later on, multi-layered perceptrons and long short-term memory networks were used. These studies concluded that neural networks are more accurate for day-to-day cash flow prediction as compared to ARIMA and Prophet.

3. Preliminary's

3.1. Support Vector Regression

Support Vector Regression (SVR) is a regression technique that builds upon the principles of Support Vector Machines (SVM). SVR aims to map data to a high-dimensional feature space using a specific kernel function. It tries to find an optimal hyperplane that can predict continuous outputs rather than performing classification, as seen in SVM. The SVR tries to find this hyperplane in a higher-dimensional feature space. In this extended space, the SVR searches for a function, the hyperplane, that for most data points, will can deviate from the true output no more than threshold ε.

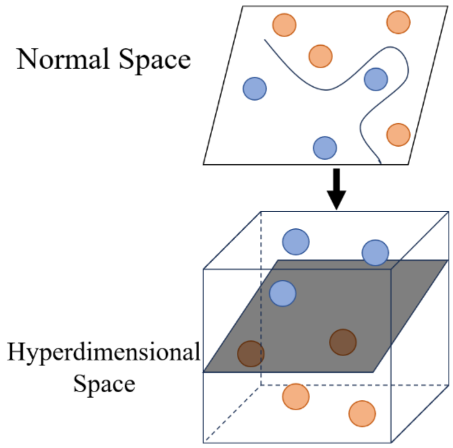


Figure 1 - Example of a SVR fitting a hyperplane in hyperdimensional feature space

In the framework of ε-insensitivity, SVR uses a cost function that disregards error that fall within a certain distance from the real value (Smola and Scholkopf, 2004). This creates an e-insensitive tube around the estimated function, within which no penalties are assigned for errors. Predictions outside this tube are penalized by the SVR, proportional to the extent of the deviation. These characteristics make the SVR's reputation as "robust", as it is less susceptible to outliers.

During training the SVR, the following objective function and constraints are (Smola and Scholkopf, 2004):

$$\text{minimize} \quad \frac{1}{2} \| w \|^2$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{cases}$$

Where x_i is a training sample with target value y_i. The prediction for the sample is the inner product plus intercept $\langle w, x_i \rangle + b$, and ε is the insensitivity parameter. This formula assumes that a function exists that approximates all pairs (x_i, y_i) with ε precision. In other words, the assumption is that the problem is *feasible*. This may not be the case, and some errors may be wanted to be allowed. The C hyperparameter controls the trade-off between allowing training errors and forcing strict margins. A smaller C value creates a wider margin, which allows more violations of the margin. This may result in a simpler model, at the potential of a higher bias error. A larger value for C aims for a larger margin violation penalty, which corresponds to a narrower margin. Therefore, the model could be more complex, leading to higher variance error, but will be more accurate with respect to the training data. The C hyperparameter controls the slack variables ξ, ξ_i^* , which allow otherwise infeasible constraints of the optimization problem. With the C parameter added, we conclude the following formula stated in Cortes & Vapnik (1995):

$$\text{minimize} \quad \frac{1}{2} \| w \|^2 + C \sum_{i=i}^{l} (\xi_i + \xi_i^*)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^* \\ \xi, \xi_i^* \geq 0 \end{cases}$$

The interplay between the C and ε parameters is a delicate balancing act. Together, these parameters define the width of the ε-insensitive tube and the severity of the penalties for violations of this tube. The ε parameter regulates the size of the ε-insensitive zone, within no penalty is given for errors. By contrast, the C parameter controls the penalty for observations that fall outside this ε-insensitive zone.

The incorporation of different kernels in SVR provides flexibility. By enabling the model to manage both linear and non-linear data relationships, the SVR can fit a wide range of data. By selecting suitable kernel functions and adjusting the parameters, it's feasible to model complex patterns and relationships in the data, which makes SVR a flexible and usable tool in machine learning, capable of tackling a broad range of regression problems (Cortes & Vapnik, 1995).

3.2. Multi-layer Perceptron

Multi-layer perceptrons are feedforward networks, which means that information in the network only flow in one direction. The network is organized in multiple layers, including an input layer, one or more hidden layers, and an output layer. Each layer consists of interconnected 'nodes', which simulate the neurons in the human brain. Every node produces an output by applying an non-linear function to its inputs from the layer below. As all neural networks, MLPs are used to estimate the true, unknown function that explains the output vector y in function of the input vectors x (Goodfellow et al., 2016).

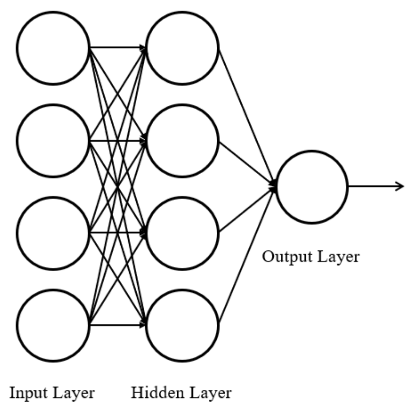


Figure 2 - Example of a Multi-layer Perceptron with 4 input nodes, 4 hidden nodes and a single output node

These networks are called *neural* because they are closely inspired to neuroscience. The dimensionality of the hidden layers determines the width of the model. Each element of the vector may be interpreted as resembling a neuron.

Backpropagation is a fundamental algorithm in training neural networks, including MLPs. At its core, backpropagation is an application of the chain rule from calculus used to compute gradients efficiently. During the forward pass of network training, inputs are propagated through the network's layers to generate an output. This output is then compared with the expected output, and the difference forms a 'loss' or 'error.' Backpropagation comes into play in the backward pass, where this loss is propagated back through the network. Starting from the output layer and moving toward the input layer, the algorithm calculates the gradient of the loss function with respect to the network parameters (weights and biases). This gradient, also called the gradient descent, is calculated as followed (LeCun et al., 2015):

$$\Delta_w(t) = -\epsilon \frac{dE}{dw_{(t)}} + \alpha \Delta_w(t-1)$$

In which $\Delta_w(t)$ is the current gradient iteration, ϵ is the bias, dE is the error and $dw_{(t)}$ is the weight factor. The learning rate is depicted by α and $\Delta_w(t-1)$ is the weight of the previous iteration.

These gradients indicate the direction in which the parameters should be adjusted to minimize the loss and improve the accuracy of the network. As the backpropagation is performed iteratively, the network's parameters are updated in each iteration, which allows the model to learn complex patterns in the data over time. This iterative learning process is a key aspect of the supervised learning algorithms used in MLPs.

Instead of a layer of only nodes, it is also possible to add dropout layers to a MLP. Dropout layers are a regularization technique used in neural networks to mitigate overfitting and improve generalization performance (Srivastava et al., 2014). The main idea is to randomly "drop out" a certain percentage of the nodes or units in a layer during training, preventing them from contributing to the forward pass and backpropagation. One of the key advantages of dropout layers is that there is no need for complex regularization technique or extensive hyperparameter tuning.

4. Experiments

In this section we will experimentally show which patterns can be found in predicting the liquidity of companies in the Netherlands. Proving the impact and precision of MLPs and LSTM in predicting liquidity predictions.

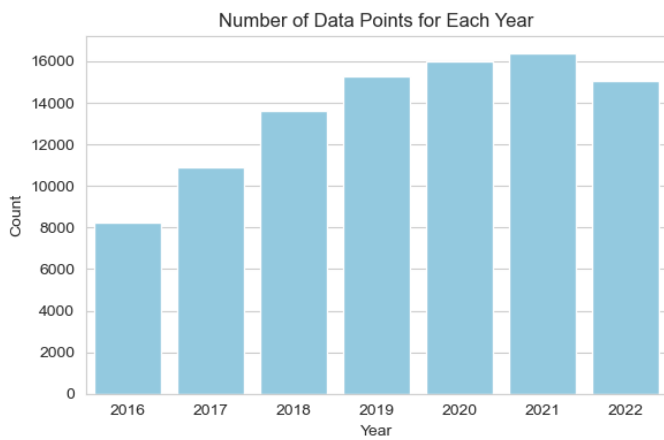


Figure 3 - Number of Data Points for Each Year

All experiments were conducted on financial data of Dutch SMEs. The timeframe of the dataset is from January 2016 to December 2022. The dataset consists all transactions within the timeframe. To make the task more manageable, all transactions were grouped in monthly sums. This also eases the prediction process, as otherwise the models would also need to predict the amount of transactions in a given month. After grouping *107863* rows of data remained, with each row resembling a financial month per reference classification (RCSFI code), per company (division). The dataset consists of financial data of 496 companies in the Netherlands. Outliers outside of the 97.5% quantile were left out as most these months were deemed hard to predict, although there will be an experiment where outliers are kept in the dataset.. All data has been normalized with normal scaling to enhance the predictability of the models. The division and RCSFI codes are one-hot encoded to improve prediction. The models will be evaluated on their precision in predicting the net amount for each RCSFI code per division. Summarizing these amounts would result in the liquidity position for a company. Each experiment will be evaluated through the mean absolute error (MAE) and root-mean-square error (RMSE). The mean absolute error is used to make the results easily interpretable and to not overly penalize outliers. The root mean-square error will be used to give a higher weight to outliers. To give a better overview of the performance of the models, both metrics are used.

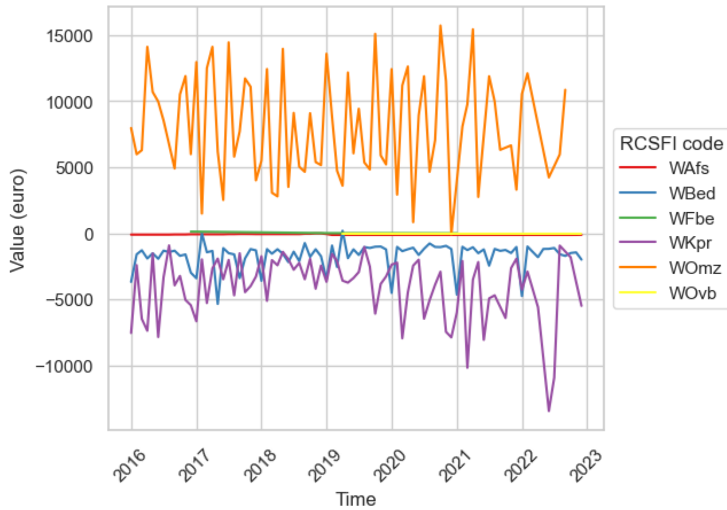


Figure 4 - Timeline of values for a single division. Note that some RCSFI codes have little datapoints at this particular company as these RCSFI codes are not used frequently by the company

4.1. Performance experiments

4.1.1. Choosing models

The models for all experiments were optimized on the dataset, of which all data before 2022-01-01 has been made the training set and data after is the test data. For building the MLP, an exploratory research method was applied. Multiple experiments with different complexities were conducted before determining the most effective one. The most optimal model was determined by the lowest RMSE, while keeping the model as small as possible to prevent overfitting. For this reason, dropout layers are also introduced. As result, the model that was used had an input layer of 1000, two hidden layers with 800 and 500 nodes and a output layer of a single value. A dropout of 10% was applied after every layer. The model was trained using TensorFlow's' RMSProp with a learning rate of 0,001.

The hyperparameters for the SVR were found using a grid search over the hyperparameters using a linear kernel. Non-linear kernels such as radial basis function (RBF) and polynomial were deemed to computationally expensive, slowing down the process. Therefore, a linear kernel was used to train the model in a reasonable timeframe e. Using grid search, in which the trained model was compared to test data. In the end, the hyperparameters were set on C = 0.1, ε = 0.1 and 10000 maximum iterations.

4.1.2. Assessing the impact of window size on accuracy

In this experiment, the point of the train-test split is gradually changed, such that the train set involves more data, and the test set is at a later timepoint. The object is to see how the model performs over time. The model stability can be observed, by seeing whether the performance deteriorates, stabilizes, or improves, also the robustness and predictive power can be observed. With this analysis, we may also see how the model handles temporal dynamics, as the model must deal with new trends and data.

Experiment	Train-test split
2.1	2017-01-01
2.2	2018-01-01
2.3	2019-01-01
2.4	2020-01-01
2.5	2021-01-01
2.6	2022-01-01

Table 1 - Train-test splits for different experiments

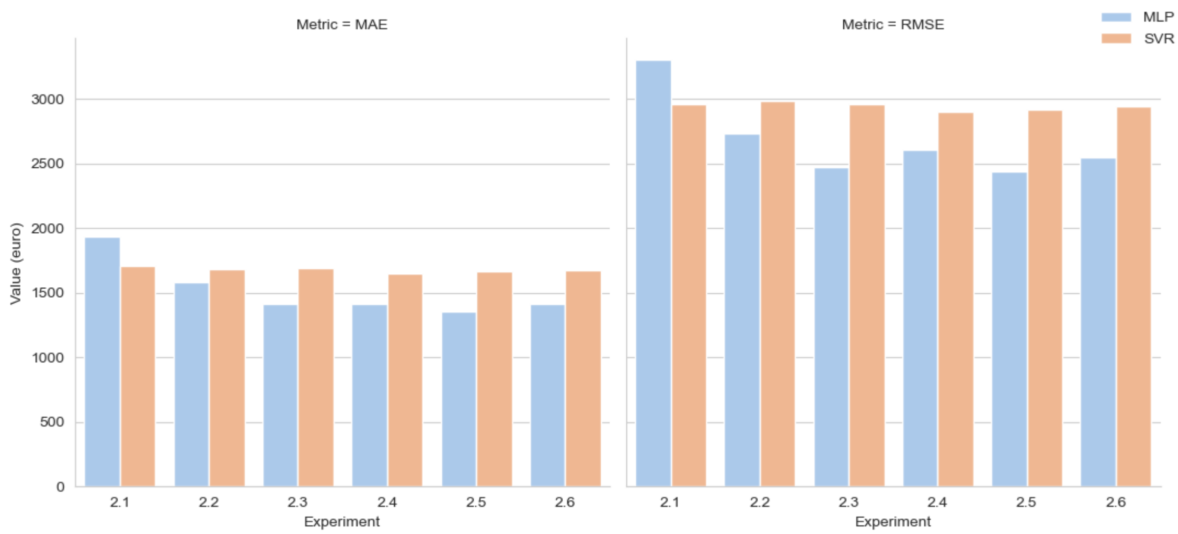


Figure 5 - Results for experiments 2.1 to 2.6

For each experiment, the training set spans from the beginning of the dataset, which is January 2016, up to the point where the train-test split is made. Subsequently, the test set is a full year for each individual experiment.

Figure 5 illustrates a decline in error for the MLP as the time-test split is pushed back, notably after the transition from experiment 2.1 to 2.2. This could be attributed to the model's complexity, which require larger quantities of training data for more accurate predictions. The continuous improvement observed across all experiments suggest that both models are generalizing effectively and that the predictive power enhances over time. On the other hand, the SVR shows no signs of improving and stays on the same level throughout all experiments. Interestingly, the MLP model starts of performing worse compared to the SVR, but performs better when the dataset includes more data.

4.1.3. Testing the consistency and temporal dependence

In the sliding window analysis, a 'window' of consecutive data points is defined, which predicts the next points. This continually trains on a new set and predicts the subsequent points. This process continues over the whole data range. The sliding window analysis is conducted to assess the temporal dependence of the model, the robustness and generalization capability.

Experiment	Train year	Test year
3.1	2016	2017
3.2	2017	2018
3.3	2018	2019
3.4	2019	2020
3.5	2020	2021
3.6	2021	2022

Table 2 - Distribution for the train and test data

For each experiment, both the training set as the test set consist of data for one year, with the test set being the consecutive year of the training set.

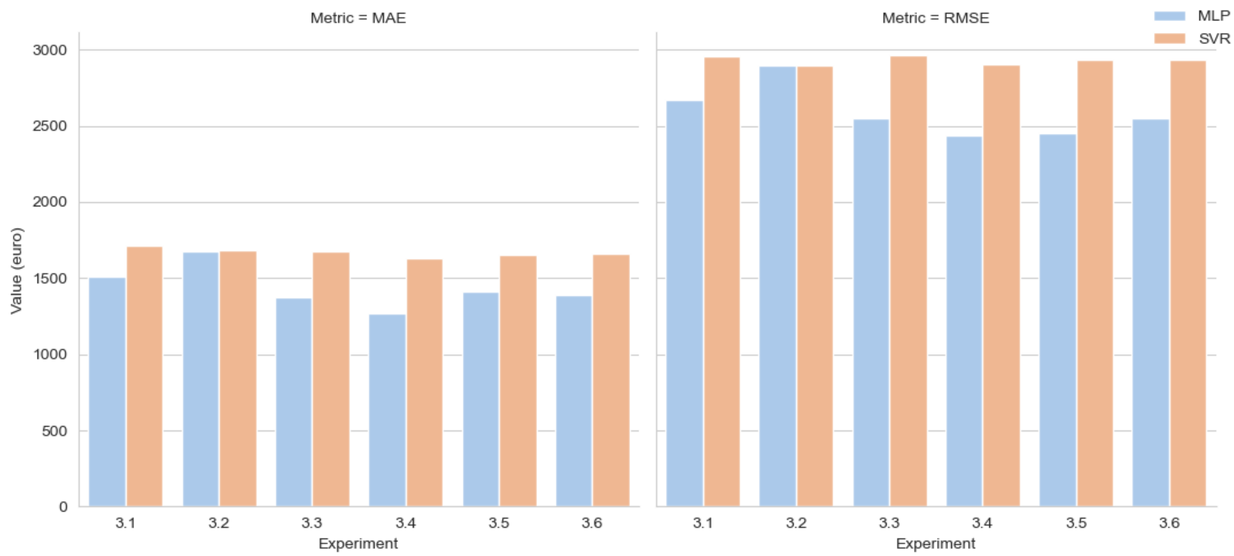


Figure 6 – Results for experiments 3.1 to 3.6

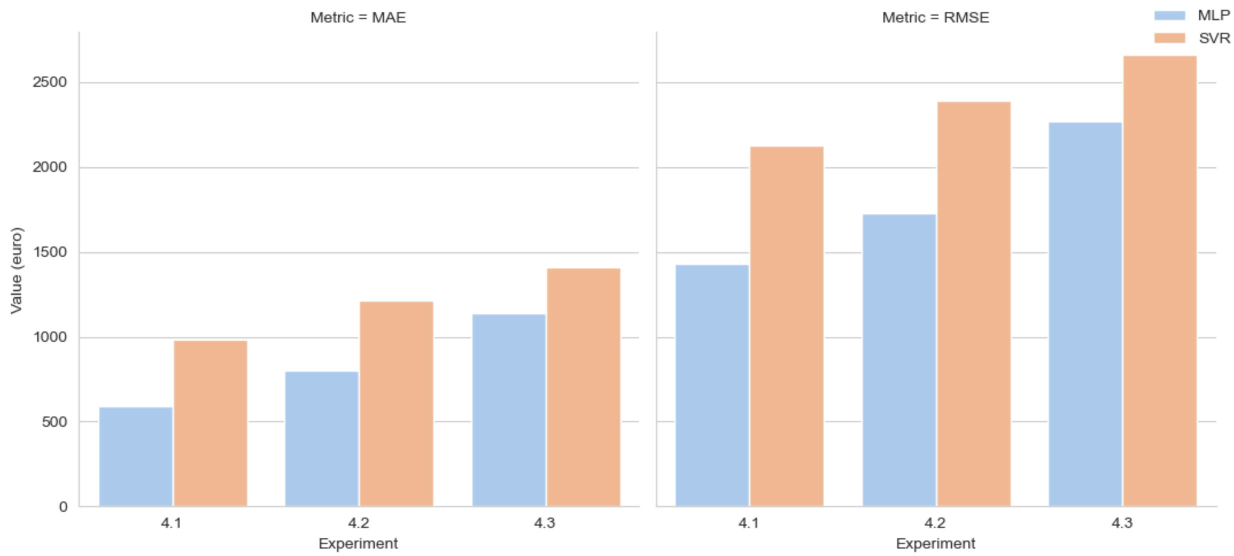
In Figure 6 is visible that the error moves around a little bit for the MLP and stays the same for the SVR. The fluctuating error for the could be attributed to the amount of data points, as it looks to resemble the plot in Figure 3. This observation suggests that the MLP, given its complexity, may require a larger quantity of data points to perform optimally. This theory is also supported by higher error rate compared to the experiments in section 2, which indicate that the model might have too little data to its structure. Nonetheless, the small fluctuations for the MLP implies that the model remains reasonably robust. On the other hand, the SVR consistently reaches the same error for all experiments. This consistency could be interpreted as a sign of the model's stability and reliability.

Despite the stability for both models, the error rate is quite high across all experiments. This suggests that while the model is stable, it may not be as efficient or accurate as required. The models may struggle to capture the underlying patterns, or the models may not be the best given the complexity and characteristics of the dataset.

4.1.4. Assessing impact of data volume on robustness

In this experiment the number of divisions in the data will be decreased to examine how the model handles a diminishing availability of data. Observing the results of this may provide valuable insight in the efficiency and scalability of the model. If the model continues to perform reasonably well with reduced data, it may indicate a high level of robustness and the potential to deliver useful insights even in data-scarce situations. Decreasing the data size may also help identifying at which point the models begin to significantly lose predictive accuracy, which can help at better understanding the minimum data requirements for the model.

The data was reduced through randomly sampling divisions from the dataset. In experiment 4.1 the full dataset is utilized, without exclusions. For experiment 4.2 half of the divisions are randomly selected and omitted, reducing the dataset to 50% of its original size. Finally, in experiment 4.3 the dataset is reduced by randomly excluding 75% of the divisions.



In the experiments, it has been observed that a substantial reduction in data leads to an increase in both MAE and RMSE. This suggests that the predictive performance of the models under study ss significantly influenced by the size of the input data. Due to the learning capability of MLP being strongly influenced by the richness of data, a reduced dataset may not adequately represent the diversity of the input space. Thus, the network may fail to learn and generalize effectively. In the case of SVR, the reduced dataset may not provide the sufficient variability and density required to define a optimal hyperplane, causing the model to underfit.

4.1.5. Testing impact of external datasets on model performance

To examine whether the performance of the models can be improved by adding external data to the datasets. As external information might impact the performance of companies, both a dataset with consumer confidence (experiment 5.1) and one with the customer price index (experiment 5.2) are tested. Both datasets are gathered from the website of the Dutch Central Agency for Statistics (CBS). The datasets are joined to the original dataset on the transaction date.

The external datasets did not appear to contribute to the predictive accuracy of the models. This lack of improvement could come from several factors, such as irrelevance of the added variables to the target output, or the external data not being complimentary to the original dataset. This does not necessarily mean that adding external dataset does not have impact, it might be that these external datasets, in combination with added variables from within the company might have more impact. The results are visible in appendix.

4.1.6. Testing resilience to outlier data

In the data preparation for all experiments, outliers that were beyond the 95^{th} percentile were excluded to preserve data homogeneity and potentially enhance the performance of both the SVR and MLP. This decision was primarily driven by the concern that the presence of extreme values might distort the learning process, skewing the generalizability and reliability of the models. To examine the robustness of the models in real-world scenarios, where outliers often exist, we reintroduce these data point into the analysis and evaluate the impact on the model.

Model	MAE	RMSE
MLP	1382	2525
MLP + outliers	6657	35125
SVR	1678	2939
SVR + outliers	7972	56584

Table 3 - Results for experiment 6, with and without outliers

The introduction of outliers back into the dataset has resulted in extreme increase in both MAE and RMSE, with RMSE experiencing an even more pronounced escalation. This suggests a significant deterioration in the model's performance when dealing with more irregular and complex data including outliers. The increase could be due to the model trying to adjust its predictive parameters to accommodate extreme values and thus overfitting. The more dramatic increase in RMSE is of even greater concern, as the RMSE places more weight on larger errors by squaring the individual prediction errors before averaging. Therefore, a large increase in RMSE suggests that the outliers are causing even more substantial errors. The results underline the sensitivity of both models. It may be beneficial to explore techniques that are more resilient to outliers or apply a more advanced outlier detection mechanism.

4.1.7. Robustness against noise

This section further investigates the robustness and resilience of the models. Additional experimental process involving the addition of Gaussian mixture noise to the dataset. Gaussian Mixture Models (GMMs) represent a probabilistic model that assumes all data points are generated from a mixture of finite number of Gaussian distributions. By adding noise generated from GMMs, we aim to examine the stability and performance of our models under circumstances of increased uncertainty and variability.

Experiment	Noise
default	0
7.1	0,01
7.2	0,1
7.3	1
7.4	10

Table 4 - Division for experiments in 7

The noise addition serves to simulate real-world scenarios where data may be contaminated with various type of noise and disturbances, and to assess how well the models can handle such conditions. The underlying hypothesis is that a robust model should be able to maintain a reasonable performance level, even in the presence of such noise.

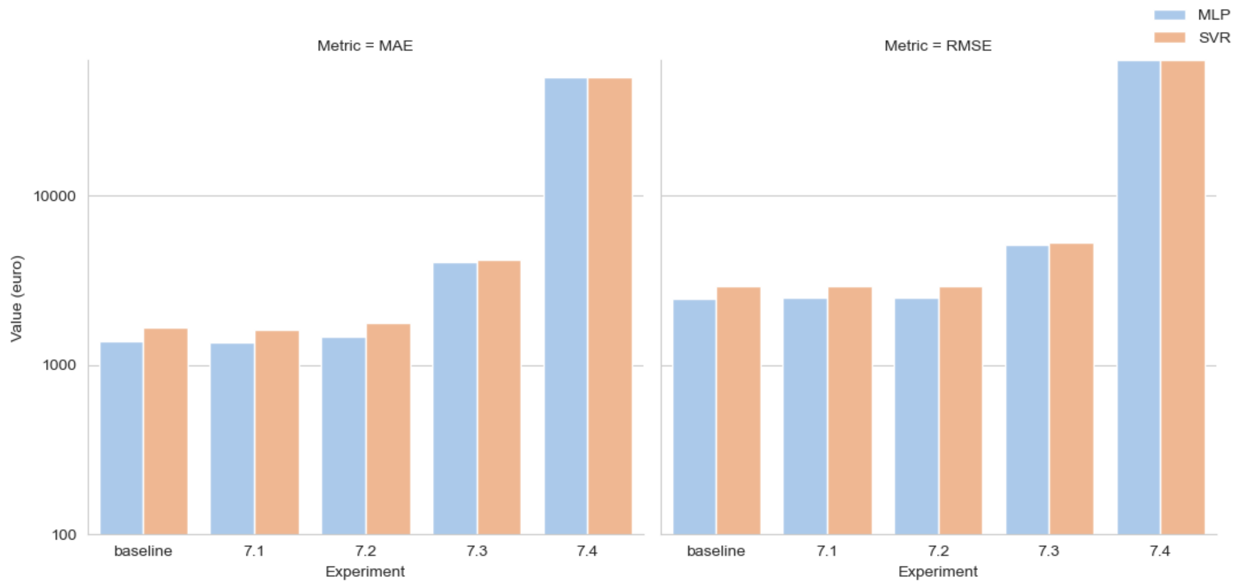


Figure 7 - Results for experiment 7 on a logarithmic scale

It is observed that the error metrics increase significantly with each subsequent experiment, indicating a degradation in the model performance. As much as that the increase in errors necessitated the use of a logarithmic scale on the plot for a clearer visualization of results. This trend underscores a substantial decrease in the model's prediction accuracy as the level of noise increases. It suggests that that while the SVR and MLP may handle a certain degree of noise, their performance deteriorates as the noise level becomes more extreme. These findings provide valuable insights into the sensitivity of noise, which can be crucial in future model selection and the development of strategies for noise handling in real-world applications.

4.2. Practical Application

Following the extensive evaluation of the SVR and MLP under various experimental conditions, the next step involves the exploration of their potential practical applications. As the goal of forecasting is not merely attaining a high accuracy on a given dataset, but the successful implementation of these models in real-world scenarios to solve practical problems. The performance, robustness, and resilience of the SVR and MLP models have been assessed in relation to varying data volumes, outlier presence, addition of external dataset and Gaussian noise. While these experiments have provided valuable insights into the characteristics of and limitations of the models, there is still room to examine their usability in real-world scenarios. All of these statistics are based on the results of the default model, which has its train-test split on January 1st 2022.

4.2.1. Sector-specific analysis

At first the results for companies will be divided per sector, to see if some sectors can benefit more from using the models than others. The characteristics of the financial landscape for each sector can be quite different. Factors as market trends, regulatory environment and sector volatility can influence financial data differently, leading to unique data patterns within each sector. By conducting analysis that respects these specific dynamics, we can better assess the performance and suitability of the models for real-world scenarios.

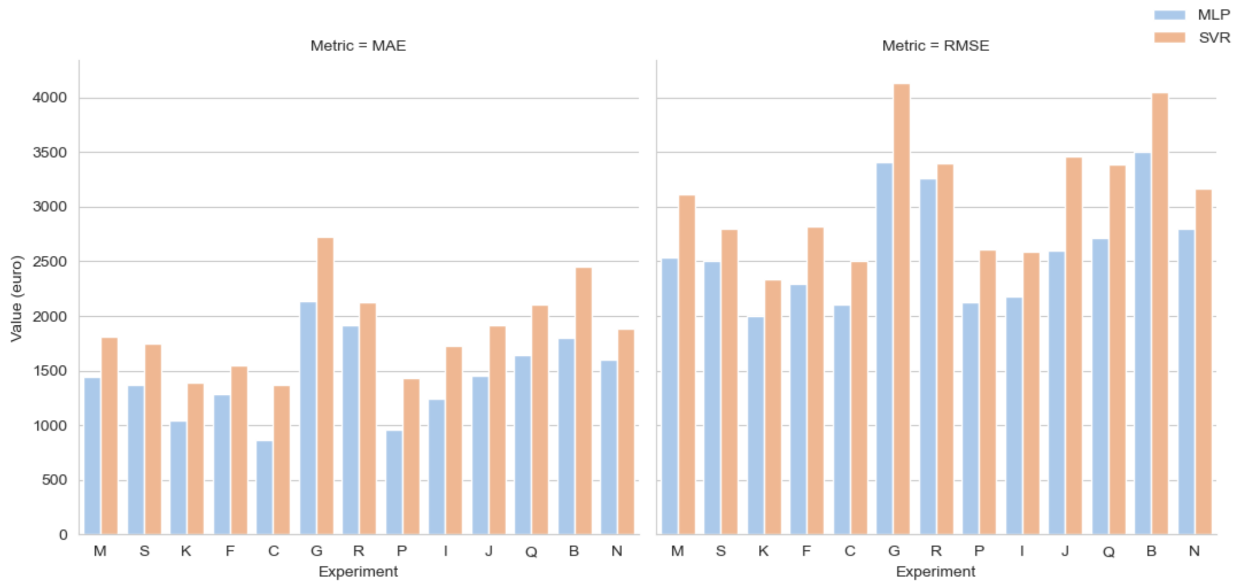


Figure 8 - Errors scores across sectors using the baseline model. Note that sectors have varying data occurrences. Excluding sector L, E and T due to insufficient data occurrences.

As illustrated in Figure 8, the error scores marked variations across different sectors. This heterogeneity can potentially be attributed to the different financial characteristics inherent to each sector. For instance, some sectors may have exhibited a more predictable trajectory throughout the financial year than others. Other sectors may have more substantial fluctuations, anomalies, or unique patterns. However, does not fully reconcile with the observed discrepancy between the MAE and RMSE. Under the assumption that sectors with more extreme values would contribute to more extreme errors in the RMSE compared to the MAE. Yet, this expected pattern is not conclusively evident in the results.

It is also visible that there are almost no differences between the MLP and SVR when comparing both with each other. The difference between error margins is the same for all sectors. This suggests that there are no sectors which may be more beneficial to one model than the other.

4.2.2. RCSFI analysis

In this part, the error metrics for each RCSFI code individually will be examined. Since different RCSFI code may have distinct underlying characteristics, understanding the distribution of errors across these codes can provide valuable inside. For instance, one RCSFI code may yield higher errors compared to others, indicating potential issues or challenges associated with that particular code. By focusing on understanding these factors contributing to the errors, it becomes possible to develop targeted strategies or interventions to address those specific challenges and improve the accuracy of predictions.

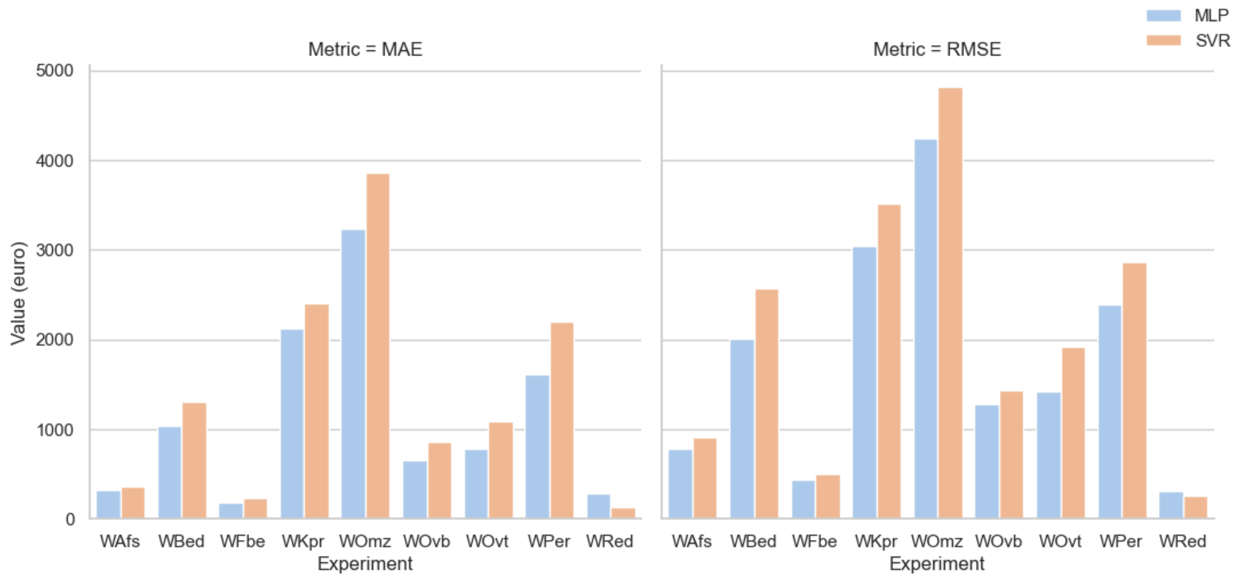


Figure 9 - Error scores across different RCSFI classifications. Excluding WBel due to insufficient data occurrences

The results in Figure 9 show that there is a big discrepancy in the error scores for different RCSFI codes. RCSFI codes like Wafs, WFbe and WRed perform exceptionally well as compared to other codes. It is not exceptional then that these codes are respectably amortizations, financial incomes and expenses and shares. These are all codes that resemble a very stable source of income and outcome. The values for these predictions will not fluctuate as much as other RCSFI codes, such as WOmz, which relates to the net sales, but also WKpr, which is the cost of sales. Both codes are more prone to substantial variations in amounts, making them more challenging to predict. Both models are not able to correctly predict these amount with the current information. There may be more data, such as indicators for predicting sales, needed to correctly predict the values corresponding to these codes.

Overall, the results reinforce the understanding that certain RCSFI codes, such as amortizations, financial incomes and expenses, and share, have more consistent and predictable patterns, resulting in lower error rates. In contrast, codes like net sales and cost of sales are more susceptible to significant fluctuations which opens challenges for precise predictions.

This analysis underscores the importance of considering the inherit characteristics and volatility of each RCSFI code when evaluating the performance of predictive models. By recognizing these distinctions, further steps can be taken to improving the accuracy of predictions for codes that are prone to larger fluctuations.

4.2.3. Prediction quality

In this subsection, the focus is to compare the predicted verses actual values for both SVR and MLP. The predicted vs. actual plot, is a commonly used graphical representation for assessing the performance of predictive models. By plotting predicted values against the actual ones, we can visually inspect the extent to which the model's predictions align with the reality.

In an ideal scenario, the scatter plot would show a homogeneous distribution along the diagonal line, without a noticeable pattern. This would indicate an accurate prediction irrespective of the actual output value. However, patterns in the residual or systemic deviations from the diagonal line would signal issues. Systemic deviations from the diagonal line might imply a model bias, where the model consistently underpredicts or overpredicts the target variable. By analysing these plots, we aim to obtain qualitative insights about the behaviour of the models and identify any specific areas of strength and weakness. These insights will further enrich our understanding of these models' performance and may contribute to more informed decision-making in future studies and appliances

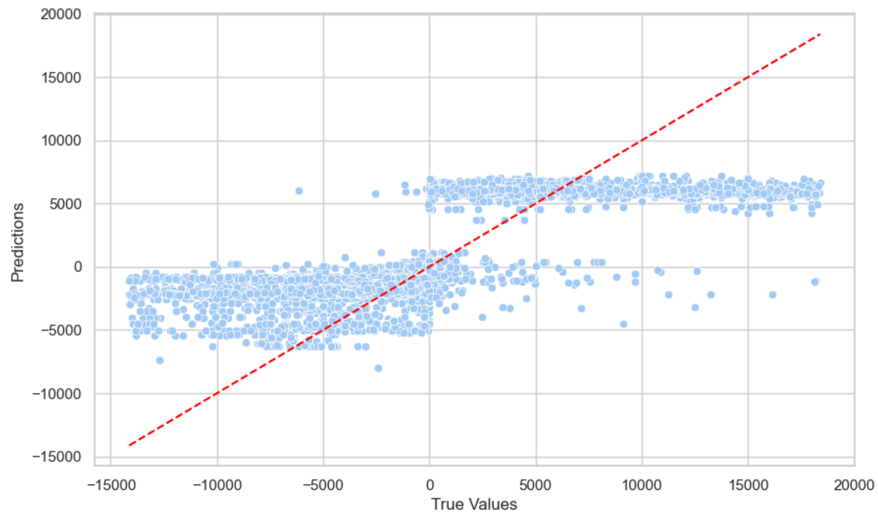


Figure 10 - Predicted vs. Actual plot for the SVR

As depicted in Figure 10, the predictive performance for the SVR exhibits certain limitations. It appears that the model struggles to predict values out of a specific range, leading to a narrow band of predictions primarily in the positive value range. Altough, this band widens for negative values, it remains relatively constrained. Furthermore, a siginificant portion of predicted values deviate considerably from the optimal red diagonal line, resulting in a plot that does not hold any resemblence to the expected optimal line. This discrepancy coulg potentially be attributed to the use of a linear kernal. Due to linear kernels assuming a linear relationship between the input features and the target variable, it might not capture the inherent non-linear patterns present in the dat. As a result, the model's ability to accurately predict values beyond the target range is limited. To improve this predictive performance, it may be worth to consider using a different kernel, such as RBF. These kernels are more capable of capturing complex relationships between the features and the target variable.

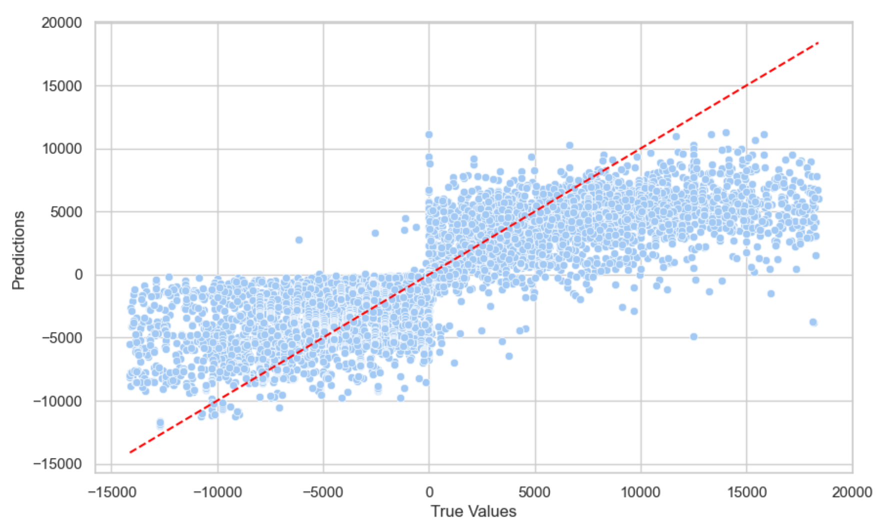


Figure 11 - Predicted vs. Actual plot for the MLP

In comparison to the results obtained from the SVR, Figure 11 shows more resemblance of the target line. While the model still faces challenges in accurately predicting larger positive values, there is a wider band of positive value predictions compared to the SVR model. Although the results still do not meet the requirements for practical use, the increased similarity to the target line suggests improved performance of this model. This is further supported by most error scores, which indicate better predictive accuracy. Despite remaining gaps between the predicted values and the target line, the closer alignment indicates that this model captures more of the underlying patterns in the data. However, further enhancements may still be needed to achieve a sufficient level of prediction. Exploring different model architectures, such as more complex neural networks or ensemble methods, could potentially lead to better results.

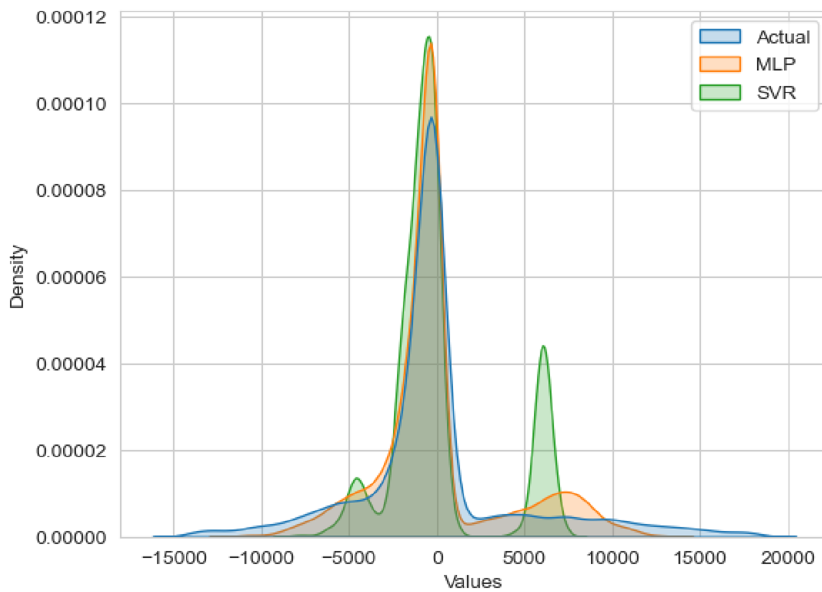


Figure 12 - Density plot for the actual values and predicted values by the MLP and SVR

Our findings earlier are also supported by the density plot in Figure 12**Fout! Verwijzingsbron niet gevonden.**. In this plot is visible that for the values around 0, both models are able to grasp how the actual data behaves, but around that the models predict different values. Especially the pronounced clusters around the negative values suggest that both models tend to predict considerably different values compared to the actual ones within these regions of output space. This observed behaviour implies that while the SVR and MLP models demonstrate satisfactory performance for values around 0, they may struggle to accurately predict values outside this range.

5. Discussion

The experimental analysis conducted on both models provide valuable insights into their performance and applicability in predicting cash flow and liquidity position.

In the experiments section, we found that both models do not work well enough to be applied by accountants in a real world scenario. In respect to this, the MLP performs better over a number of tasks when comparing it to the SVR with the MLP performing better over a number of tasks when compared to the SVR. Both models were found to be sensitive to the size and quality of the data. This suggests that both models are reliant on robust, clean and representative datasets to ensure optimal performance. Similarly, the reintroduction of outliers and noise significantly increased the error metrics, pointing to difficulties to maintaining predictive accuracy under these conditions.

Additionally, the predicted vs. actual plots revealed that while MLP managed to capture more of the underlying patters than the SVR, which did not hold much resemblance to the target data. Both models struggled with accurate predictions for larger values, especially in the positive range. This suggests a limitation in their ability to effectively manage the complexities of financial data and indicates a potential area for further exploration.

Interestingly, the performances of the two models were not significantly different across various sectors. This suggests that the decision to utilize one model over another should not influenced by the sector under investigation. However, variations in model performance across different RCSFI codes indicate that the specific financial aspects being predicted should influence model selection.

Despite the limitations observed, our findings offer a substantive contribution to the existing body of knowledge on machine learning applications in liquidity forecasting. The sensitivity of both SVR and MLP models to size and quality of data underscores the importance of comprehensive data management practices in machine learning projects. Data preprocessing techniques, such as outlier and noise handling must be implemented to enhance model performance.

The potential for further refinement and explorations is evident from the experiments, as performance lacks to ensure real-world usability. Future work could consider using a SVR with a different kernel, such as RBF, which can capture more complex relationships between variables. For MLP, experimenting with more complex architectures may yield better performance.

Although the inclusion of external datasets did not lead to improvement in present study, it should not deter future research into adding more predictors. The inclusion of extra internal data would be recommended for future investigation, as this might better help in predicting future course of a company as opposed to the external data added earlier.

6. Conclusion

While the present study has contributed to a deeper understanding of the capabilities and constraints of SVR and MLP models in predicting financial prediction in Dutch SMEs, it is not recommended to apply the current model to real-world scenarios. As the findings have demonstrated that both the SVR and MLP models possess significant sensitivity to the size and quality of the data used. Our findings also highlight that a sufficient volume of data is crucial for achieving optimal model performance. Furthermore, our study illuminated the models' limitations in handling complexities of financial data. Especially with values that deviated further from most predictions. This is also supported by the larger error scores for RCSFI codes that show bigger variation in the amount of money that gets handled, like the WOmz code for net sales. This variability suggests that the specific financial aspects being predicted can influence the choice of the model, this requires further investigation.

In summary, while this research has provided valuable insights into the predictive capacities of SVR and MLP models in the context of predicting liquidity position. It is important to note that the selection and application of machine learning models should be guided by the unique requirements of the task and data at hand. By building on the findings on this study, we can continue to advance our understanding of machine learning applications in financial prediction, and develop more refined predictive models.

7. References

Cao, L., & Tay, F. E. H. (2001). Financial Forecasting Using Support Vector Machines. *Neural Computing and Applications*, *10*(2), 184–192. https://doi.org/10.1007/s005210170010

European Commission (2022). *Annual Report on European SMEs 2021/2022.* Luxembourg: Publications

Fernando, J. (2023). Working Capital: Formula, Components, and Limitations. *Investopedia*. https://www.investopedia.com/terms/w/workingcapital.asp

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. In *MIT Press eBooks*. https://dl.acm.org/citation.cfm?id=3086952

Ho, S. L., & Xie, M. (1998). The use of ARIMA models for reliability forecasting and analysis. *Computers & Industrial Engineering*, *35*(1–2), 213–216. https://doi.org/10.1016/s0360-8352(98)00066-7

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature, 521*(7553), 436–444. https://doi.org/10.1038/nature14539

Mahfoud, S., & Mani, G. K. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, *10*(6), 543–566. https://doi.org/10.1080/088395196118425

Mramor, D., & Valentincic, A. (2003). Forecasting the liquidity of very small private companies. *Journal of Business Venturing*, *18*(6), 745–771. https://doi.org/10.1016/s0883-9026(03)00002-8

Pompe, P. P. M., & Bilderbeek, J. (2005). The prediction of bankruptcy of small- and medium-sized industrial firms. *Journal of Business Venturing*, *20*(6), 847–868. https://doi.org/10.1016/j.jbusvent.2004.07.003

Smola, A. J., & Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*, *14*(3), 199–222. https://doi.org/10.1023/b:stco.0000035301.49549.88

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958. https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf

Taylor, S. J. & Letham, B. (2018) Forecasting at Scale, *The American Statistician*, 72:1, 37-45, https://doi.org/10.1080/00031305.2017.1380080

Weytjens, H., Lohmann, E., & Kleinsteuber, M. (2019). Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electronic Commerce Research*, *21*(2), 371–391. https://doi.org/10.1007/s10660-019-09362-7

Wiśniewski, Jerzy. (2008). Econometric Modelling of Monthly Liquidity of Small Enterprise. *Dynamic Econometric Models*. 8. 95-102.

Wiśniewski, J. R. (2022). Financial Liquidity and Debt Recovery Efficiency Forecasting in a Small Industrial Enterprise. *Risks*, *10*(3), 66. https://doi.org/10.3390/risks10030066

I. Appendix I

This part of the appendix includes all test results from the Performance Experiments section in 4.1. The results include the test results of both the MLP and SVR. The first error scores are the error scores on the normalized dataset, the second pair of error scores are transformed back to normal values.

Experiment 2:

Experiment 2.1: *Train the model with data from Jan. 2016 until Dec. 2016 and test with data from Jan. 2017 until Dec. 2017*

```
MLP:
MAE: 0.4261371312309283
RMSE: 1.5669248707308419
MAE: 1931.9658480803876
RMSE: 3304.75239364593


SVR:
MAE: 0.37764570633949696
RMSE: 0.6525514493716149
MAE: 1712.1216008201568
RMSE: 2958.4539513107998
```

Experiment 2.2: *Train the model with data from Jan. 2016 until Dec. 2017 and test with data from Jan. 2018 until Dec. 2018*

```
MLP:
MAE: 0.34896906749978635
RMSE: 1.1906280521306645
MAE: 1583.820661986129
RMSE: 2729.6179261016846


SVR:
MAE: 0.3716744203709538
RMSE: 0.6376755579486715
MAE: 1686.8705098343223
RMSE: 2894.1353899796786
```

Experiment 2.3: *Train the model with data from Jan. 2016 until Dec. 2018 and test with data from Jan. 2019 until Dec. 2019*

```
MLP:
MAE: 0.31324259202880633
RMSE: 1.3081016886859567
MAE: 1417.0317679459545
RMSE: 2476.055596268678

SVR:
MAE: 0.37366744152900094
RMSE: 0.6542566111834799
MAE: 1690.3787991934228
RMSE: 2959.6945889941962
```

Experiment 2.4: *Train the model with data from Jan. 2016 until Dec. 2019 and test with data from Jan. 2020 until Dec. 2020*

```
MLP:
MAE: 0.312128323038391
RMSE: 1.388591753980912
```

```
MAE: 1410.12309498697
RMSE: 2608.6769098116342

SVR:
MAE: 0.3646159895057975
RMSE: 0.6420233272008189
MAE: 1647.2501658893584
RMSE: 2900.5119431811727
```

Experiment 2.5: *Train the model with data from Jan. 2016 until Dec. 2020 and test with data from Jan. 2021 until Dec. 2021*

```
MLP:
MAE: 0.30307166078463227
RMSE: 1.29119268079975
MAE: 1358.670474605443
RMSE: 2436.434423834808


SVR:
MAE: 0.371015000114837
RMSE: 0.6514271978305008
MAE: 1663.2605238505494
RMSE: 2920.3486165753143
```

Experiment 2.6: *Train the model with data from Jan. 2016 until Dec. 2021 and test with data from Jan. 2022 until Dec. 2022*

```
MLP:
MAE: 0.3153261273252727
RMSE: 1.3050354793526038
MAE: 1412.1913509656033
RMSE: 2549.9648113607363


SVR:
MAE: 0.37488476347848304
RMSE: 0.656363578634987
MAE: 1678.9253285882999
RMSE: 2939.5311420715766
```

Experiment 3:

Experiment 3.1: *Train the model with data from Jan. 2016 until Dec. 2016 and test with data from Jan. 2017 until Dec. 2017.*

```
MLP:
MAE: 0.332895640643987
RMSE: 1.247669320434794
MAE: 1509.2394991402027
RMSE: 2669.5962996508747

SVR:
MAE: 0.3776300956738865
RMSE: 0.6525677847310537
```

```
MAE: 1712.0508271893534
RMSE: 2958.528010465406
```

Experiment 3.2: *Train the model with data from Jan. 2017 until Dec. 2017 and test with data from Jan. 2018 until Dec. 2018.*

```
MLP:
MAE: 0.3685140025444335
RMSE: 1.1707813946073924
MAE: 1673.8913097050174
RMSE: 2895.6511962368913

SVR:
MAE: 0.3698783227125679
RMSE: 0.6377124257140993
MAE: 1680.0884122094715
RMSE: 2896.6640945780587
```

Experiment 3.3: *Train the model on data from Jan. 2018 until Dec. 2018 and test with data from Jan. 2019 until Dec. 2019.*

```
MLP:
MAE: 0.30565069801774786
RMSE: 1.2862451635326102
MAE: 1376.284098314289
RMSE: 2551.612831190164

SVR:
MAE: 0.3719927516657564
RMSE: 0.6582817191650068
MAE: 1675.0091335098664
RMSE: 2964.111228206668
```

Experiment 3.4: *Train the model on data from Jan. 2019 until Dec. 2019 and test with data from Jan. 2020 until Dec. 2020.*

```
MLP:
MAE: 0.28083866303122257
RMSE: 1.219699173268476
MAE: 1265.1413147143908
RMSE: 2436.4645164852222

SVR:
MAE: 0.3625556037104745
RMSE: 0.6446240640450516
MAE: 1633.2654076676858
RMSE: 2903.9467987252738
```

Experiment 3.5: *Train the model on data from Jan. 2020 until Dec. 2020 and test with data from Jan. 2021 until Dec. 2021.*

```
MLP:
MAE: 0.32192027282162994
```

```
RMSE: 1.4037233569313752
MAE: 1408.7750481764715
RMSE: 2449.747122008863

SVR:
MAE: 0.3770686208324786
RMSE: 0.6707253595388329
MAE: 1650.1131219543822
RMSE: 2935.202389843805
```

Experiment 3.6: *Train the model on data from Jan. 2021 until Dec. 2021and test with data from Jan. 2022 until Dec. 2022.*

```
MLP:
MAE: 0.3115044011088801
RMSE: 1.2970320315145056
MAE: 1389.2636832881535
RMSE: 2549.3323177710977

SVR:
MAE: 0.3724641962972156
RMSE: 0.6580858231631739
MAE: 1661.1353865044707
RMSE: 2934.9657204123696
```

Experiment 4:

Experiment 4.1: *Decrease the number of divisions with 50%*

```
MLP:
MAE: 0.17892276859256087
RMSE: 0.38543577954279507
MAE: 801.8303629306724
RMSE: 1727.3045438645695

SVR:
MAE: 0.2703112237771535
RMSE: 1.048672700989415
MAE: 1211.3815812783469
RMSE: 2391.419293227503
```

Experiment 4.2: *Decrease the number of divisions with 75%*

```
MLP:
MAE: 0.25553055510703765
RMSE: 0.5095596525570809
MAE: 1138.170982278992
RMSE: 2269.6542476806662
```

```
SVR:
MAE: 0.31652614637148596
RMSE: 1.15809788463156
MAE: 1409.8543927879955
RMSE: 2662.480189054445
```

Experiment 5:

Experiment 5.1: *Adding the consumer confidence dataset.*

```
MLP:
MAE: 0.31227519413397636
RMSE: 0.5792622028582972
MAE: 1398.5277162263903
RMSE: 2594.231822167411


SVR:
MAE: 0.3749076590712107
RMSE: 1.2509457418028274
MAE: 1679.0278667394557
RMSE: 2939.635254754767
```

Experiment 5.2: *Adding the CPI dataset.*

```
MLP:
MAE: 0.3113486441149564
RMSE: 0.5649565158725685
MAE: 1394.3781518353105
RMSE: 2530.1636488039667

SVR:
MAE: 0.37494136011433205
RMSE: 1.2508923085458525
MAE: 1679.1787972130533
RMSE: 2939.800145899893
```

Experiment 6

Experiment 6.1: *Adding the existing outliers to the data.*

```
MLP:
MAE: 0.10668778424861194
RMSE: 0.5628795891373318
MAE: 6657.720355462023
RMSE: 35125.81063008761

SVR:
MAE: 0.12775012374272068
RMSE: 0.9202554497972566
MAE: 7972.089837448926
RMSE: 56584.0568284587
```

Experiment 7:

Experiment 7.1: *Adding Gaussian noise to test and training data (standard deviation = 0.01)*

```
MLP:
MAE: 0.3064581062340936
RMSE: 1.322533529332814
MAE: 1372.4758259011858
RMSE: 2505.589008580252


SVR:
MAE: 0.37546735768297596
RMSE: 0.6560515619380072
MAE: 1681.534482817818
RMSE: 2938.1337720354054
```

Experiment 7.2: *Adding Gaussian noise to test and training data (standard deviation = 0.1)*

```
MLP:
MAE: 0.33139964903858393
RMSE: 1.3977950209640406
MAE: 1484.1767874805262
RMSE: 2529.8952004472553


SVR:
MAE: 0.39703264519980447
RMSE: 0.6594425280645599
MAE: 1778.114848192872
RMSE: 2953.320249248906
```

conf

Experiment 7.3: *Adding Gaussian noise to test and training data (standard deviation = 1)*

```
MLP:
MAE: 0.9122532098539292
RMSE: 1.637575959942448
MAE: 4085.535526990174
RMSE: 5187.794237160231


SVR:
MAE: 0.9349895654252746
RMSE: 1.1870752599707721
MAE: 4187.360533921406
RMSE: 5316.328949762349
```

Experiment 7.4: *Adding Gaussian noise to test and training data (standard deviation = 10)*

```
MLP:
MAE: 11.270403150721926
RMSE: 14.162616394301208
MAE: 50474.61821998911
RMSE: 63332.40887175743


SVR:
```

```
MAE: 11.264887358167114
RMSE: 14.132072343332684
MAE: 50449.91568564092
RMSE: 63290.63358699366
```

II. Appendix II

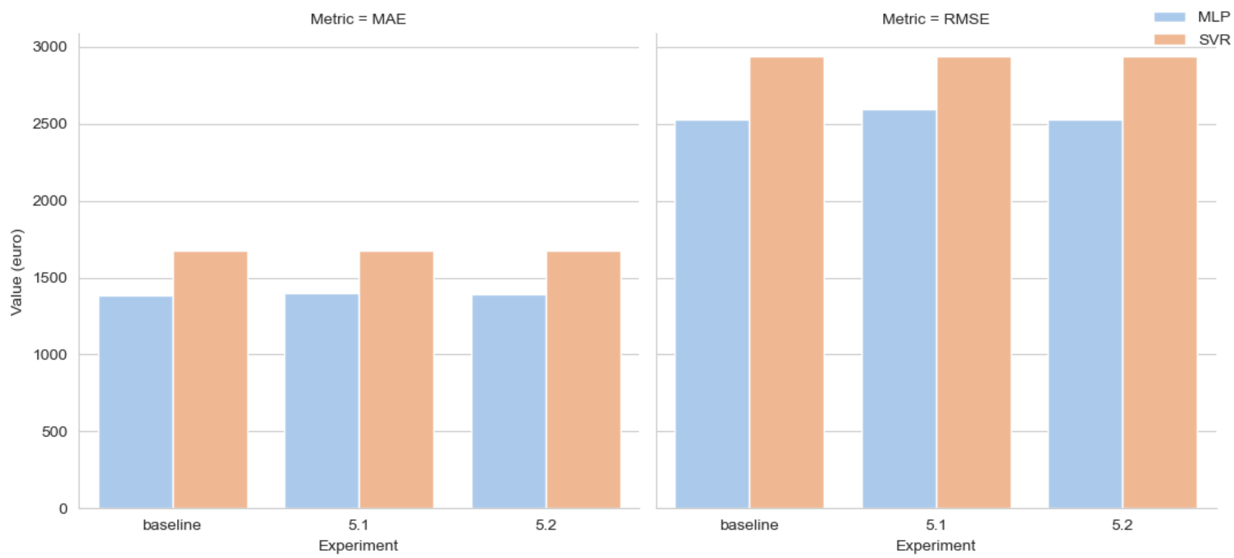


Figure 13 - The plot for section 4.1.5 in which external datasets were added to assess model performance. Experiment 5.1 added the consumer index, 5.2 added the customer confidence interval

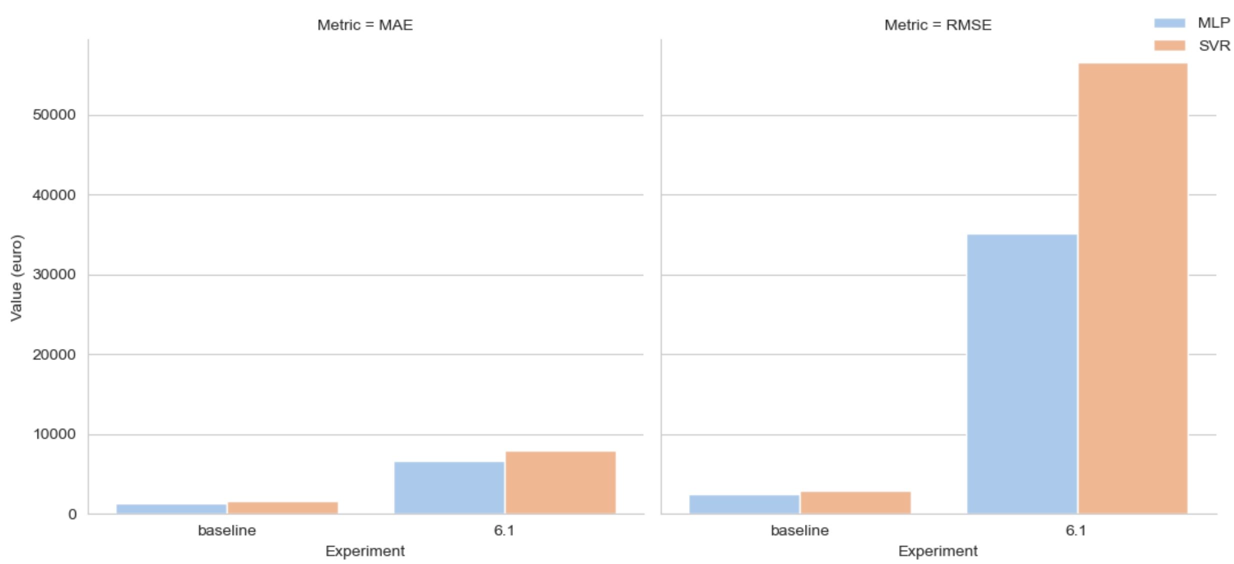


Figure 14 - The plotted results for section 4.1.6 in which outliers got added back into the dataset to assess the model resilience to outliers.

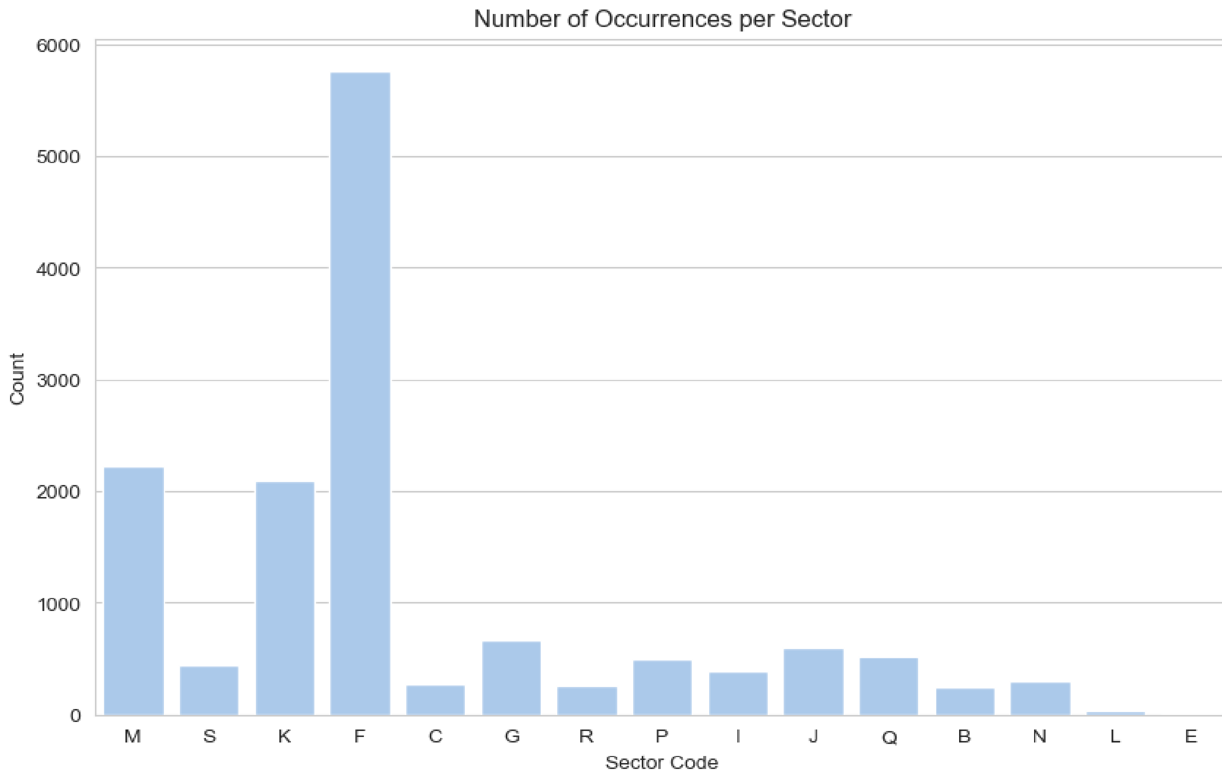


Figure 15 - Barplot to visualize the number of occurrences for each sector in the test dataset for the baseline model

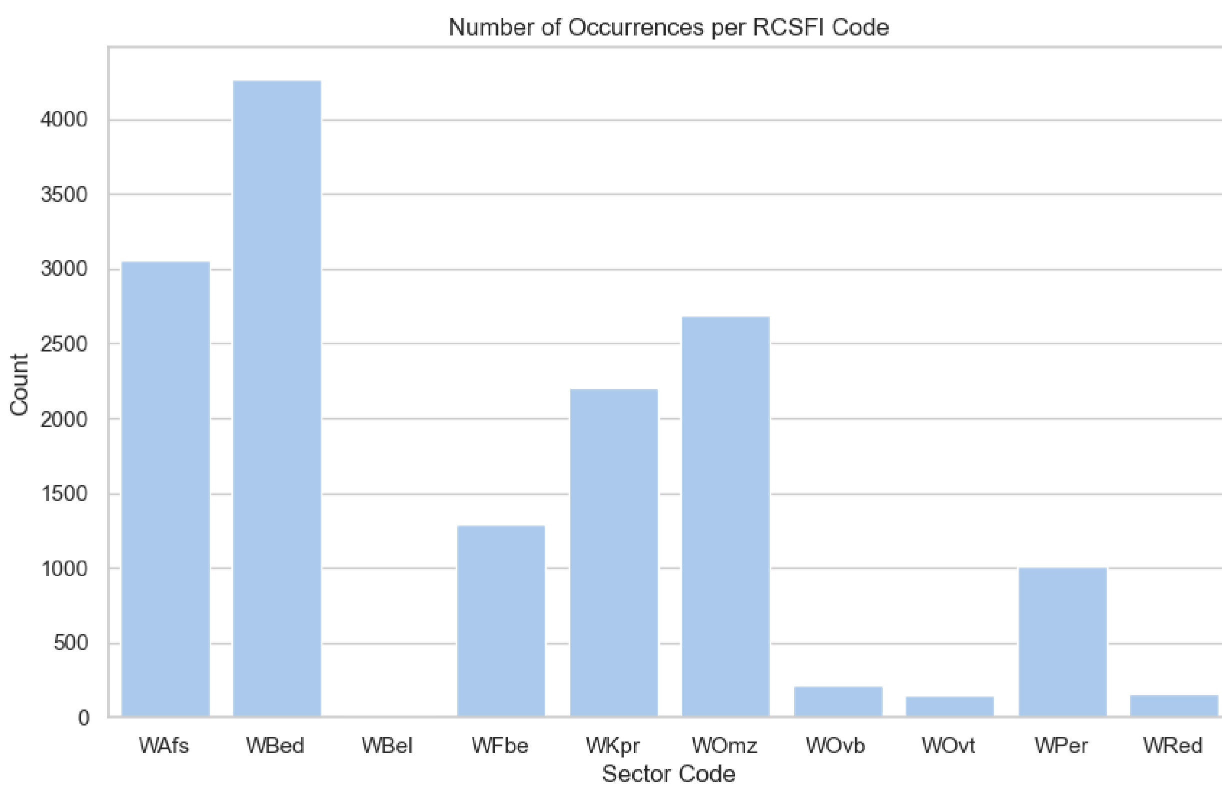


Figure 16 - Barplot to visualize the number of occurrences for each RCSFI code in the test dataset