Dynamic predictive cycle counting based on imperfect information in the spice industry

Thomas Hordijk

September, 2023

Dynamic predictive cycle counting based on imperfect information in the spice industry

Master Thesis
Industrial Engineering & Management
Production & Logistics Management
University of Twente
March, 2023

Author
Thomas Hordijk
MSC Industrial Engineering & Management

Royal Euroma B.V.
Ravensburgstraat 4
8028 PZ, Zwolle
The Netherlands

University of Twente
Drienerlolaan 5
7522N NB, Enschede
The Netherlands

Company supervisor
Mr. P. Bax
Plant manager at Euroma
Mr. J. Westerhof
Logistics manager at Euroma

University of Twente supervisor
Dr. E. Topan
Lead supervisor
Dr. I. Seyran Topan
2nd supervisor

Management summary

Euroma was founded in 1899 and is one of the largest companies in the herbs and spice industry. The facility in Zwolle is the largest of the two remaining production locations of Euroma. The facility is geared towards the production of dry herb and spice blends. The facility has a high throughput of materials, producing up to 1300 tons of mixed products weekly.

The Enterprise Resource Planner and the Mixing Control System track the inventory. The former governs all inventory, while the latter only holds the inventory in the production facility. These systems communicate and transfer inventory to keep the production process going.

Euroma experiences that these systems do not report the same inventory and that, over time, the difference between the two becomes larger. The difference between the recorded inventory levels can result in halted production due to a lack of materials, requiring a rescheduling, and possibly incurring late fees. It is also possible that a replenishment order is made, but it does not fit in the silo. This can result in a fine from the supplier or a loss of the supply contract. Euroma wants to get a better grasp of its actual inventory status. Due to the high throughput of materials, traditional counting approaches have not been implemented as the cost would be too high. Euroma seeks a smart approach to cycle counting. The main research question is therefore:

"How can inventory record inaccuracy be predicted, and how can this prediction be used to recommend counting an SKU to improve the accuracy of the inventory status at Euroma?"

Previous studies have shown that discrepancy in the inventory has six main drivers. Shrinkage, Transaction/ record errors, Misplacement, Supply yield, Incorrect product identification, and, Incorrect corrections. This research focuses on transaction/record errors, misplacement, and incorrect product identification as the main drivers for the discrepancies in the inventory record.

After identifying possible causes of inaccuracies, the search shifted towards cycle counting approaches that aim at resolving the inaccuracies. Seven possible models were identified. However, due to the process at Euroma, three were quickly disregarded. The other four, that will be used in the experimentation, are ABC Classification, Location, Transaction, and Random cycle counting.

The existing literature on predicting inaccuracy in the inventory record is limited to classification models. In the case of Euroma, a continuous value wants to be predicted; comparable studies used regression models to predict stockouts and backorders. Six regression models are identified as possible solutions: Ridge, Bayesian Ridge, Lasso, Decision Tree, Random Forest, and XGBoost regression. Aside from regression models, the application of neural networks is also investigated. Neural networks can be applied for both classification and regression.

Based on the literature review, this thesis proposes a cycle count model that uses the output of a regression model to select SKUs for counting.

The experimentation for this research is divided into two segments. The first covers the prediction model; the second expands this with the cycle counting models. Six different regression models and the neural network are investigated for the prediction model. The models are evaluated based on the R^2-score and the Mean Squared Error. The model will be used on a fast-changing dataset; computational time is an important KPI in this research.

This thesis introduces a second KPI for evaluating the performance of the cycle counting approaches. The Overall Prefect Inventory Record Accuracy (OPIRA, 2) is introduced to evaluate the performance

of the cycle counting approaches. The OPIRA is based on the Perfect Inventory Record Accuracy in the literature. This KPI shows the average Prefect Inventory Record Accuracy (PIRA, 1) of all the SKUs.

$$Perfect\ Inventory\ Record\ accuracy = \frac{Transactions\ SKU\ without\ discrepancy}{Total\ Transactions\ SKU} \quad (1)$$

$$Overall\ Perfect\ Inventory\ Record\ accuracy = \frac{\sum_{n=0}^{N} PIRA}{N} \quad (2)$$

The first experiment that is performed evaluates the effect of the encoding approach. Encoding is required as regression models cannot make use of categorical values. Two approaches were tested, one-hot encoding and integer (dummy) encoding. The performed experiments show that the integer encoding allows models to fit the data faster while not reducing the R2 score, Figure 1.

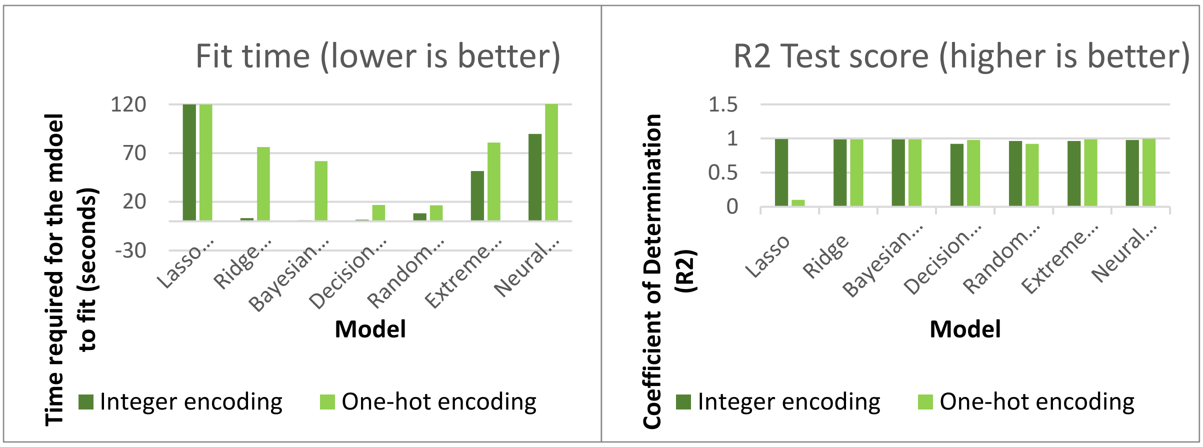


Figure 1 Results encoding experiment, fit time (left), R^2-score (Right)

Feature selection improves the model's ability to fit the data. Four feature selection methods determine which features to keep in the dataset. The four approaches did not show a consensus. The model was tested using a diminishing number of features. The features are removed based on the worst performance of all previous methods. After removing a feature, the R^2 score of the resulting model is retrieved. These scores are graphed in Figure 2 and show a decline in performance after removing the 9th worst-performing feature.

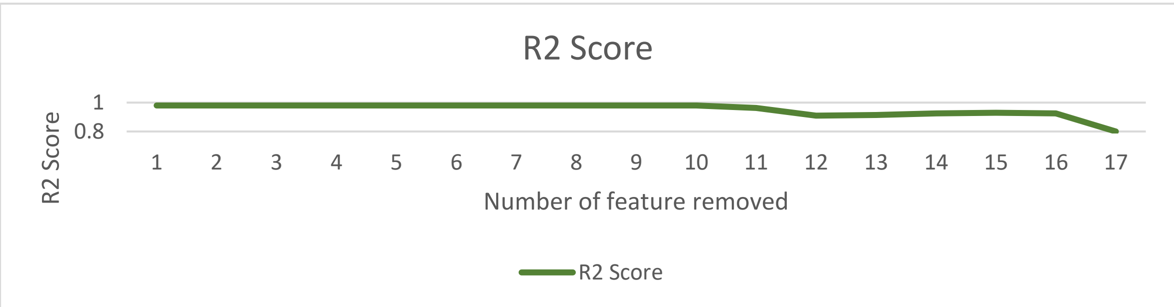


Figure 2 R^2 score feature removal

After applying the feature selection, the remaining models were tested with multiple sets of hyperparameters. The hyperparameter sets are tested using 5-fold cross-validation. The best estimator from each model was used to create a Learning Curve graph. This graph depicts the number of iterations used for training and the resulting training and validation scores. The estimators were also used to perform predictions on a third test set. The model has not "seen" this data before this. The resulting R^2-score and MSE are compared to decide which model to implement. The results showed that the best-performing model is the Random Forest regression. This model will be used in the cycle count experiments.

The second experimentation segment reviews the cycle count approaches. A Monte Carlo simulation is used for these experiments. The simulation approximates the 2022 transaction dataset based on a statistical analysis of all SKUs in the 2022 dataset. The simulation for each cycle count approach is run five times and consists of 150,000 transactions, as this closely resembles half a year of data. The results from the simulation are compared to each other, but also to half a year of transactions from the 2022 dataset.

Based on the 2022 dataset, it is concluded that the OPIRA of Euroma was around 8%. The experiment results showed that introducing any form of structured cycle counting would increase the inventory record accuracy at Euroma. The increase ranges from 2-15%. The best-performing cycle count approaches were the random forest regression and transaction-based cycle counting models. Both show an improvement of around 15%. This improvement was realized while reducing the time spent on counts over the same period by around 75%. These results stand to argue that the introduction of structured cycle counting would be beneficial to Euroma.

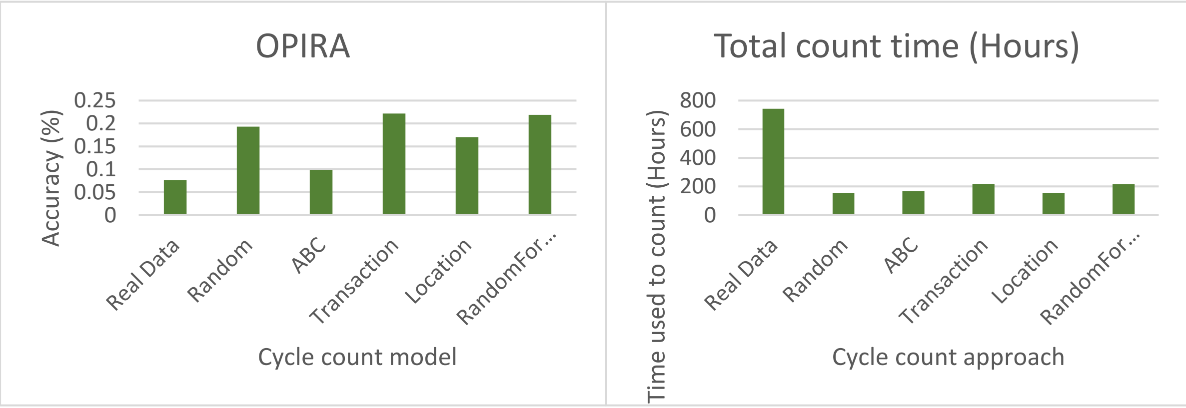


Figure 3 Summary of Overall Perfect Inventory Record Accuracy and time spent counting

The reason for the transaction and random forest regression cycle counting to perform comparably could be explained by how discrepancies are introduced in the inventory record. As there is a probability of discrepancy being introduced in every transaction, an SKU with a higher transaction count is more likely to have discrepancies.

While an improvement is realized, the level of accuracy still seems low. Literature shows that the perfect inventory accuracy in retail environments is around 30-60%. For a production environment, a higher level could be expected. The relatively low accuracy could be because the number of transactions is so large that discrepancy is introduced very quickly. The OPIRA does not account for the level of inaccuracy, so even five grams on a 60-ton silo is inaccurate.

The recommendations given to the company are to investigate further the applicability of the prediction model on real data. While the simulation, and the underlying statistical dataset, are extensive, there is a high chance of nuances in the data not being captured. This could change the model's ability to predict and the discrepancy generated. The application of the model could require re-evaluation after this implementation is realized. For future research, the recommendation is given to investigate splitting the SKUs. This split can be based on the warehouse or the type of product. The former is interesting as the different warehouses work at different magnitudes. Product types could also behave differently; further investigating and exploiting this could prove beneficial.

Preface

This thesis is the cornerstone of my master's in Industrial Engineering and Management at the University of Twente. During my time at the University I learned a lot and had many interesting experiences that I would not trade for anything. It was not the smoothest ride imaginable but definitely a fun one.

Through this preface I want to take the opportunity to express gratitude to those that have been key in the successful completion of my masters. The first thanks goes to Euroma for the opportunity to perform my thesis research with them and allowing me freedom in that process. I want to thank my friends and colleagues at Euroma for the support, contributions and insights.

Second, I would like to thank my supervisors at the University, Engin Topan and Ipek Seyran Topan. Thank you very much for the guidance and feedback during this undertaking.

The third group of people I want to thank are my fellow students, particularly those that started their graduation at the same time as I did. They will know who they are and they should know that they have been a great solace and support throughout this journey.

Last I would like to thank my family for all the support, for allowing me to go back to university after have worked for a few years. And for always being available to offer counseling words.

I hope you enjoy reading my thesis and I hope it can contribute to future research.

Thomas Hordijk

Zwolle, September 2023

Table of Contents

1 Introduction

1.1 Euroma

Euroma was founded in 1899, in Zwolle, and has been trading in herbs and spices on a global scale ever since. Initially Euroma also sold pharmaceutical products but the branches were split by Beatrice Foods after the takeover in 1969. In 1998 Euroma was again in Dutch hands and obtained a royal distinction for their 100 years anniversary in 2001. In 2018, Euroma took over one of their mayor competitors Intertaste which solidified Euroma's position as a top competitor in the European herb & spice market (Euroma, n.d.).

Production is realized at three locations in the Netherlands, namely, Schijndel, Nijkerk, and Zwolle, with the latter being the largest and the youngest. The different facilities produce products in specific categories, Zwolle produces dry products, Schijndel produces ambient liquids, and Nijkerk produces fresh liquids. The facility in Zwolle has been equipped with a number of automated systems in order to meet customer demand (Euroma, n.d.).

Euroma employs over 500 people to ensure their production can run 24/7. Aside from their employes Euroma has a variety of automation solutions that allow them to produce up to the level of their customers demand. The opening of the production facility in Zwolle was a cornerstone in Euroma's ambition to grow as a leader in the flavor market (Euroma, n.d.).

1.2 Introduction to inventory at Euroma

1.2.1 Inventory systems

Tracking of item locations and production at Euroma is realized using four different IT systems. The different programs are as follows.

An enterprise resource planner (ERP) is used to create the different orders and govern the overall inventory. The ERP keeps track of all inventory at Euroma Zwolle and all the external warehouses Euroma employes.

A warehouse management system (WMS) keeps track of the locations of every pallet inside the high-rise warehouse (EZ). This WMS is combined with a warehouse control system (WCS) that is used to operate the cranes inside of the high-rise. The cranes move the pallets between an entrance/ exit and their storage location without human intervention. The WCS keeps track of pallet locations by carrier code, but is not aware of the contents of the pallets, this information is available in the WMS.

Finally, a mixing control system (MCS) that controls the mixers used in the production of final product, this system also includes the software to control the AGVs. Aside from controlling the mixers, the MCS keeps track of the inventory in the silos, both external as internal silos.

These systems communicate with one another, and transfer inventory as needed, to realize all the different steps of day to day processes at Euroma. Figure 4 shows the communication between the different systems.

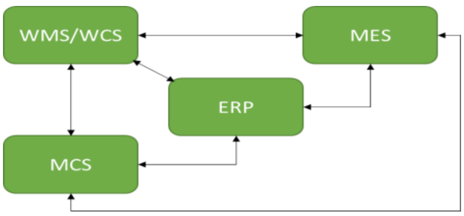


Figure 4 Communication between IT systems

1.2.2 Inventory locations

At Euroma Zwolle's facility there are 7 "warehouses" where materials are stored. The inventory is tracked by the ERP. The ERP differentiates between inventory at Euroma Zwolle's high-rise (EZ), fluid warehouse (EZVS), consumables (EZIN), outside silos (EZSI), internal silos (EZDS), miniload warehouse (EZMV), and big bags (EZBS). The ERP does not keep track of the locations inside these "warehouses".

The high-rise warehouse is the main inventory location at Euroma Zwolle and is managed with their WMS. The high-rise is an automated pallet warehouse, shown on the left of Figure 5Figure 4. The WMS records what material is on a pallet, and the location of the pallet inside the high-rise. The "miniload" warehouse is integrated in the high-rise and is also controlled and managed using the same software.

*Figure 5 Euroma Zwolle during construction**Invalid source specified.***

The MCS tracks inventory inside the 12 external (EZSI) and 32 internal silos (EZDS). The external silos have a volume of 60 m^3, the internal silos are a lot smaller, 4 of which have a volume of 7 m^3 and the remaining have a volume of 5 m^3. The MCS also tracks resources that are already inside an internal bulk container (IBC). Euroma has 60 IBCs, with a volume of 1.5 m^3, that are moved using the AGVs at the different stages of the mixing process. Each silo and each IBC is a separate inventory location in the MCS.

Figure 6 External silos

1.2.3 Inventory flows

At Euroma inbound inventory is stored in two places, the high-rise and the external silos. Inbound pallets are booked in at the expedition and stored in the high-rise. Bulk is pumped into the external silos per truck load. There exists a third location, the “MiniLoad” warehouse. This warehouse is part of the high-rise structure but contains materials in totes up to 25Kg.

Inventory that is moved inside of Euroma, is always moved between the high-rise to the location where it is, or was, needed. The high-rise stores pallet loads, either with big bags, boxes or bales. Bulk material is stored in the external silos. These silos are connected to 4 internal silos that function as a buffer. From these silos material can be injected into one of the two 10K mixers. The remaining 28 internal silos are filled using big bags.

To create a mixture the required resources are requested from the high-rise, the internal silos and the “MiniLoad” warehouse as needed. The internal silos fill the IBCs automatically while pallets from the high-rise are requested to manually deposit unit sized loads into IBCs. Any resource demand below unit sized is scooped out of a tote from the MiniLoad warehouse.

After mixing, the mixers deposit their load drop the mixture. The smaller mixers transfer the mixture to an IBC that is then used to fill bales or big bags. In the case of the two 10K mixers, the contents are deposited directly into big bags. The big bags or bales are stored on a pallet in the high-rise until needed by a customer or for packaging for consumer products.

The last production step at Euroma is the filling department. Mixtures in big bags are dropped into the filling machines that deliver consumer packaged end products. These are bundled into pallet loads, each pallet is send to the high-rise until a freight truck comes to pick it up.

The inventory flows are illustrated in **Error! Reference source not found.**. Here a differentiation is m ade between materials on pallets such as big bags or bales, bulk or loose materials, and materials in totes.

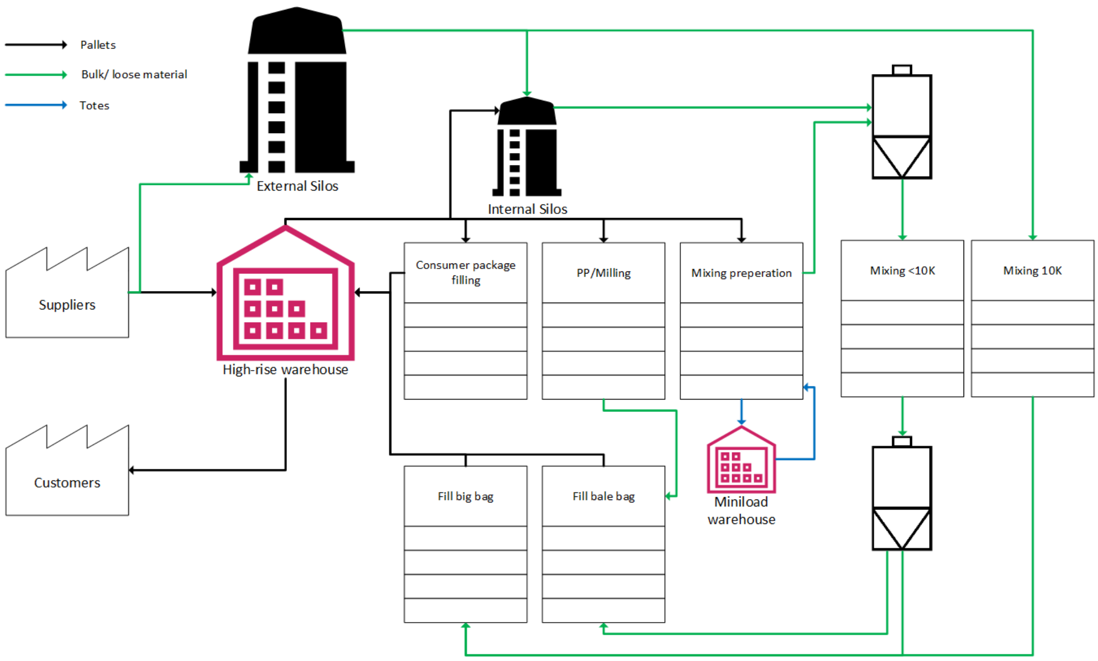


Figure 7 Inventory flows at Euroma

1.3 Problem identification

In this sections an overview of the problems is provided, including examples as illustration. This will result in a problem cluster that will be used to identify the core problem that we will solve in our research.

1.3.1 Problem background

Euroma uses three different inventory holding systems, their ERP, MCS, and WMS. The ERP is software package that governs all of Euroma's inventory, on-site and in external warehouses. When the inventory status of the ERP and MCS is compared the level of inventory for many SKUs is unequal. In Table 1 the discrepancy of inventory status of the warehouses shared by the ERP and MCS, at an arbitrary point in time, is shown.

Table 1 Inventory discrepancy between Euroma's ERP and MCS

Warehouse	Unique articles	SKUs ERP<>MCS	(%) SKUs deviating	Absolute deviation all SKUs
EZDS	28	27	96,4%	19.565,32 Kg
EZSI	10	7	70,0%	66.756,93 Kg
EZMV	58	36	62,1%	78,83 Kg
EZVS	55	20	36,4$	564,58 Kg

From Table 1 it is quickly evident that there is a lot of discrepancy, between the ERP and the MCS, in the inventory records. As both these systems aim to represent the same physical inventory of the different SKUs, the true inventory status is unknown. The inventory levels being described change at a high rate as Euroma is producing mixtures every day of the week, 24 hours per day, totaling over one million kilograms of product. During production, the MCS also corrects inventory levels. When and why this happens is not always clear and will be investigated in this research.

Different departments rely on information from either the ERP or MCS. If the inventory levels in these systems differ this can result in issues. Planners apply the ERP and the mixing department uses the MCS. It is very well possible that the planners start an order, as they see sufficient inventory, but the mixing department cannot execute it, as their system shows too little inventory.

This problem also occurs the other way around. The planner notices that the external silos do not have sufficient materials to execute a production order so they have a truck deliver a new load. However, upon arrival of the truck, the silos cannot be filled as their level is too high. Sending back a truckload costs several thousand euros and after too many occurrences results in an unbinding of the contract with the supplier.

Euroma requires insight into its inventory level to ensure that production can continue. A solution is needed that is able to provide better insight into the actual inventory status.

1.3.2 Causes of discrepancies

An analysis of the data and some preliminary literature research has shown that there are common causes for discrepancies in the inventory record that can be identified at Euroma. One of the main problem Euroma faces is transaction/record inaccuracies as a result of incomplete or incorrect communication between the ERP and MCS. Another cause that was identified is misplacement, which is typically followed by shrinkage due to expiration of the product.

1.3.3 Problem *cluster*

In Figure 8 a visual representation of the relation between problems concerning the inventory reliability is provided in a problem cluster. The problem cluster illustrates the scope of this research.

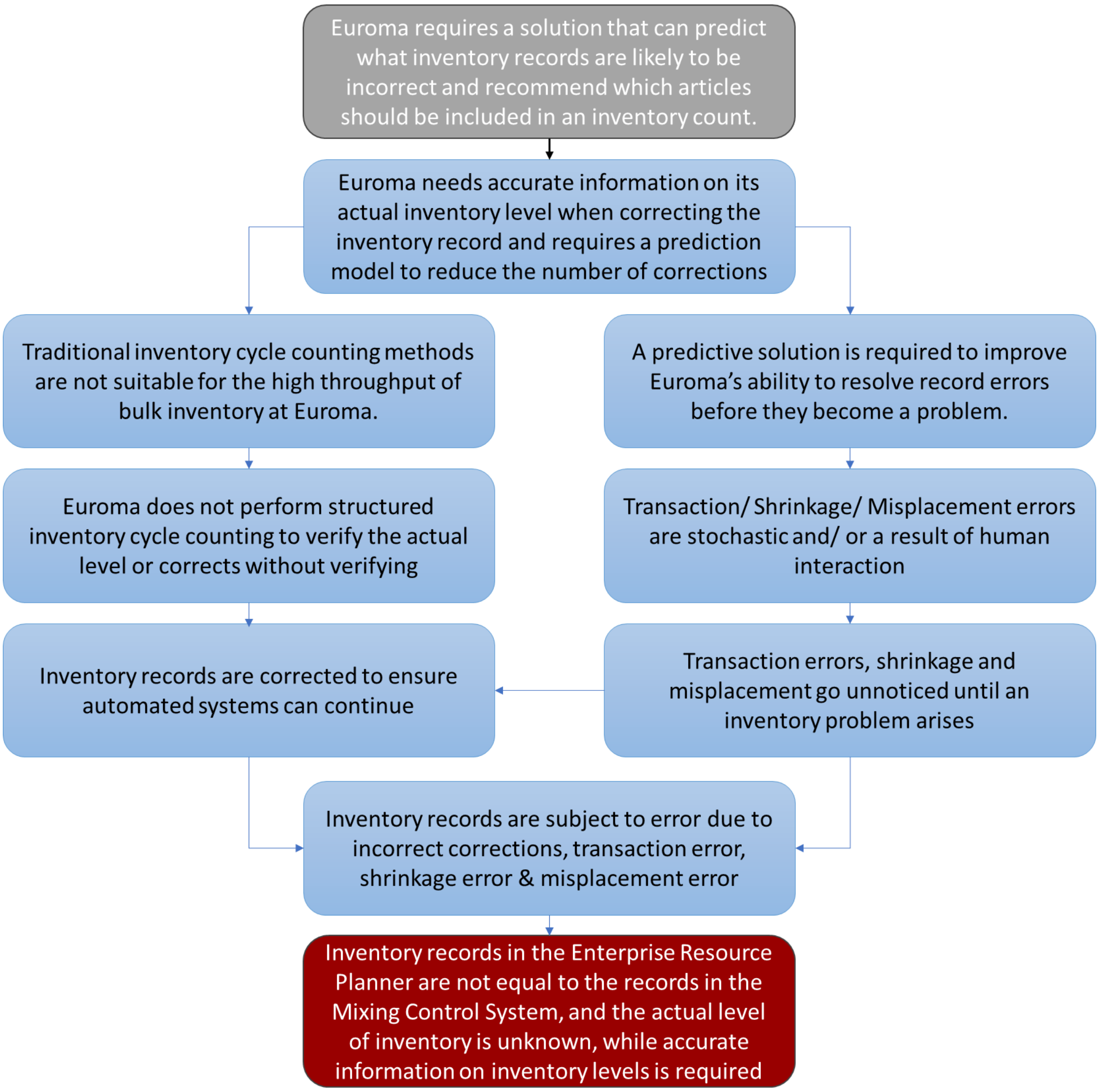


Figure 8 Problem cluster inventory reliability at Euroma

At the end of the causal chain lies the problem as it is observed by Euroma. The discrepancy between the norm and the reality as perceived by the problem owner is an action problem (Heerkens & van Winden, 2017). Therefore the action problem of this research is:

"Inventory records in the Enterprise Resource Planner are not equal to the records in the Mixing Control System, and the actual level of inventory is unknown, while accurate information on inventory levels is required"

Finding the causes of the action problem was done through an analysis of the available data and interviewing stakeholders. As the inventory reliability is a problem experienced by most departments at Euroma every department is viewed as a separate stakeholder.

1.3.4 Core problem

In this section the core problem is selected from the problem cluster, Figure 8. Solving the core problem will help solving the action problem (Heerkens & van Winden, 2017).

The core problem of this dissertation is two-fold as two main reasons for deviation in inventory records have been identified. At the start of the causal chain we find these root causes:

"Euroma requires a solution that can predict what inventory records are likely to be incorrect and recommend which SKUs should be included in an inventory count"

This research will focus on studying models applicable for prediction the inventory status and methods to differentiate SKUs and decide what to count based on the provided prediction. In doing so the problems downstream will be solved in an effort to improve the reliability of Euroma's inventory status.

1.4 Research design

This section of the report will provide an introduction to the approach in solving the core problems that were identified in section 1.3. The objective of this research will be discussed first and the practical and scientific contribution will be outlined. Following up, the scope of this research will be provided as well as the research questions that will be answered to solve the core problems. Concluding this section will be a description of the approach used to find the answers to the research questions and a method to validate the findings.

1.4.1 Objective

The main objective of this research is to increase the reliability of the inventory status at Euroma. The first step in solving this was the identification of the core problems. From this it stems that the objective of this research is finding a method that allows for predicting an expected level of inaccuracy in the inventory and providing an indication on whether an inventory count is required.

The objective with regard to the scientific contribution is to research if and to what extent the discrepancy in the inventory record and the actual level can be predicted. Furthermore, this research will look into how this prediction can be used to make the decision on when to initiate a count on specific SKUs.

The practical contribution of this research is aimed at increasing Euroma's trust in their inventory status and improving the operational performance by ensuring planned orders can be executed on schedule. Ideally, Euroma wants to have an inventory record that is 100% accurate, as this is not realistic, the objective is to achieve an accuracy of 95%. The current accuracy of the inventory is around 10%.

1.4.2 Scope

This research focuses on the relevant and known segments of Euroma's inventory status control. A crucial part of Euroma's operation is the mixing of spices, and the availability of raw materials is key for operations to continue. Inventory inaccuracy and reliability is an issue that plays in many departments of Euroma. Currently, most of the issues that Euroma experiences around its inventory status occur within the production department. The production department follows a schedule that is based off the ERP inventory status while they operate using the MCS inventory status. Therefore, this research will focus on the inventory that is shared between the ERP and the MCS. This limits the view to the EZSI, EZDS, EZMV and EZVS.

Comparing the inventory between the WMS and ERP does not show significant discrepancies between the two, therefore, this is left out of this research. The inventory of consumables, such as foil and packaging material, was improved on in a previous "improvement wave" at Euroma. Euroma introduced a cycle count to ensure the inventory status of the consumables stays reliable. This is therefore also excluded from the scope of this research.

1.4.3 Research questions & approach

Solving the core problem of this research will be realized by answering a number of research questions. These questions are aimed to fill the gap in knowledge required to verifiably improve the inventory reliability. The main research question is therefore as follows:

"How can inventory record inaccuracy be predicted, and how can this prediction be used to provide a recommendation to count an SKU in order to improve the accuracy of the inventory status at Euroma?"

Answering the main research question will be realized with a series of sub-research questions. These questions are structured following the MPSM method as described by Heerkens and van Winden (Heerkens & van Winden, 2017), see Table 2.

Table 2 Research approach according to MPSM phases

MPSM phases		Report chapters		
Phase	Description	Question	Section	Chapter
1	Defining the problem	-	1.3	Introduction
2	Formulating the approach	-	1.4	Introduction
3	Analyzing the problem	1	2	**Error! Reference source not found.** situation
4	Formulating solutions	2, 3	3	Literature review
5	Building model	4	4	Prediction and cycle coun Framework
6	Experiments	5 6	5	Experiments
7	Evaluating the solution	7	6	Conclusion

Improving the inventory reliability at Euroma first requires insight in the methods that are currently applied to realize this goal and how it is currently measured. Filling the knowledge gap on this subject is done through the first research question and underlying sub questions.

1. **What methods are currently applied to improve the reliability of Euroma's inventory status?**
    - **a.** *What KPIs are used to measure the reliability of the inventory records at Euroma?*
    - **b.** *How is inventory tracked throughout the facility and production process?*
    - ***c.*** *What measures or methods does Euroma apply to rectify their inventory records to represent the physical level?*

Answering the sub-research questions is realized by observing the available data and discussing the insights of the stakeholders. The answers to these questions will be discussed at length in Chapter 1.

The analysis of the current situation is followed by a literature review, this constitutes Chapter **Error! Reference source not found.**. The goal of the literature review is to provide a definition of inventory reliability, inventory record (in-)accuracy, and how these can be measured. This chapter will also be used to investigate possible methods and or models used to solve similar problems, possibly in different fields of research.

2. **What are common warehouse inventory cycle counting practices in the literature?**
    a. *What KPIs are used when measuring the reliability of inventory records?*
    b. *What approaches to inventory cycle counting are described in the literature?*
    c. *What cycle counting methods and models can be applied?*

3. **What prediction models are available in the literature for predicting discrepancies between the inventory record and the physical inventory level?**
    a. *What models are applicable for predicting potential errors in the inventory records?*
    b. *What data is available at Euroma that can be used to make predictions on errors in the inventory records or potential stockouts?*

Chapter 4 focuses on solving the inventory record accuracy problem at Euroma. Here the information gathered from the literature will be applied to construct a framework and synthesize a model able to use transaction data as input to provide a prediction on the inventory record discrepancy and use this to improve the cycle counting decision. The synthesis will be realized by answering the following question.

4. **How can the prediction and count recommendation model be constructed?**
    a. *How is data introduced to the model?*
    b. *What preprocessing steps are required to use the data for prediction models?*
    c. *What preprocessing steps are required to select SKUs for cycle counting?*
    d. *How can the prediction model's output be interpreted?*
    e. *How can the prediction be used to recommend SKUs for cycle counting?*

After gathering and synthesizing all the information into a working model, the findings will be tested. The tests will focus on what the expected improvements will be and whether the approach provides a robust and repeatable solution. Chapter 5 focuses on this experimentation phase.

5. ***Which prediction model performs best on the available data?***
    a. *How can the performance of the prediction models be evaluated and compared?*
    b. *What parameters influence the model's ability to predict the inventory discrepancy?*
    c. *How do the different prediction models compare in their ability to predict the inventory discrepancy?*
6. **Which cycle count model performs best on the available data?**
    a. *How can the performance of the cycle count models be evaluated?*
    b. *What parameters influence the selection of the cycle counting models?*
    c. *How does the performance of a random count compare to the different cycle counting approaches?*

Finally, in Chapter , the consequences of implementing the solution will be evaluated.

7. **How can the proposed solution be implemented and what are the consequences of implementing the solution in the workflow at Euroma?**
    a. *What er the requirements of the approach on the current inventory management structure?*
    b. *What are the consequences for the involved stakeholders?*

In Chapter 6 the conclusion on the research will be provided including recommendations for Euroma or future research on the subject. This chaper also provides the discussion, recommendations, future research and validation.

2 Current situation

This chapter focuses on providing an overview of the processes that take place at Euroma. In particular it will focus on providing answers to the first research question and its sub-questions. Section 2.1 will provide an overview of the stakeholders that fit in the scope of the research. In section 2.2 the processes that apply corrections to the inventory status per resource category. This will include automatic as well as manual corrections. This will be followed, in section 2.3, by an elaboration on the process flow of executing a correction. Inventory is tracked throughout the entire facility of Euroma both for keeping track of inventory levels and for track and trace on their products for end users. Tracking is done through a variety of transactions, section 2.4 goes into detail on this subject. Finally, section 2.5 provides insight in the current approach to inventory counting and where it is applied.

2.1 Stakeholders

The main stakeholder of the research is the company, Euroma. As a production company the reliability of their inventory status is important in ensuring continuous operation. The effects of not having inventory available when it is expected can be detrimental to the efficiency. The other way around, holding too much inventory increases the holding costs for Euroma unnecessarily.

Within Euroma different departments can be viewed as different stakeholders. The effects of the problem varies between departments in relation to their dependency on the inventory and the type of interaction they have with inventory. The departments that are involved are logistics, planning and mixing. While the other departments at Euroma are reliant on the inventory, they do not perform any mutations to the inventory status.

The planners are responsible for scheduling the production of mixtures such that they can be packaged and sent to customers on time. They mainly apply the ERP software and rely on its inventory status. The mixing department executes the planned production orders. They use the MCS and, consequently, are reliant on that inventory status.

Logistics is responsible for materials being available, or stored, in the proper location. Movement between to and from the high-rise is facilitated by logistics. This department uses all inventory holding systems at Euroma, the ERP, MCS and WMS.

The logistics department also employs the warehouse management specialists. Currently when inventory records and physical levels are not aligned they are tasked with resolving the issue. This usually consists of parsing the inventory transactions of the SKUs with an issue to find the problem or going out and counting the physical stock. The latter option is not used often as the production process at Euroma does not have space in the operational planning to include a sudden count.

2.2 Measuring inventory accuracy

Euroma has two different approaches to measuring the accuracy of their inventory records. The first approach constitutes creating a selection of pallets. These pallets are then retrieved and the contents are evaluated. For this approach the KPI that Euroma applies is the economic value of its inventory. The value of the retrieved selection is compared to the expected value, the deviation is then assumed to be the inaccuracy. This method only holds for the high-rise where inventory is stored per pallet.

The second approach disregards the physical inventory all together. A comparison is made between the inventory level as it is recorded in the ERP, and in the MCS. The KPI for this approach is the percentage of SKUs that deviate from the expectation and is referred to as the accuracy of the inventory record. The accuracy is calculated by dividing the number of SKUs that do not match by the total number of SKUs, with SKUs counted multiple time in the case of unique batch codes. This method is applied to the EZ, EZDS, EZMV, and EZVS warehouses.

2.3 Inventory tracking

Every change in inventory status is logged as a transaction, be it movement from one warehouse to another (ERP), or from a bulk container to an internal one (MCS). Euroma is producing round the clock, therefore, inventory transactions are constantly being performed. Every transaction falls in a category based on the movement of the inventory. The ERP and the MCS have their own sets of transaction categories.

The ERP uses five transaction types, see Appendix B. The only corrective transaction known to the ERP is the inventory correction. This is used by a warehouse specialist to mutate the current record so that the synchronization between ERP and MCS can continue.

The MCS uses a larger variety of transactions, see Appendix C. In total there are seven corrective transactions of which four are manually executed and the remaining three are executed by the system to continue operations.

2.4 Inventory flows

This section will provide insight into the inventory management and the decision around this in the current situation at Euroma. Section 2.4.1 describes the possible flows of bulk dry materials in large quantities and the subsequent section, 2.4.2, focuses on bulk dry materials in small quantities. Section 2.4.3 describes the management of fluids at Euroma.

2.4.1 Bulk dry materials, large quantities

Dry bulk material is stored in 12 silos outside the production facility. The outside silos are connected to 4 smaller internal silos inside the mixing department, material is transported using a vacuum. The volume of transferred material is monitored using level sensors in the silos.

In both the ERP and the MCS this location change is recorded, however, the ERP only records the type of warehouse, while the MCS records the exact silo or warehouse location.

The remaining 28 internal silos that are filled manually using big bags. When the level inside of the silo is low, or insufficient, for the next order, the operator receives a big bag with the required material from the high-rise. After scanning the big bag and the destination silo, the bag is positioned over the top of the internal silo. It is then connected using a portable filling aid and the contents drop into the silo. At this point the ERP communicates the weight in the big bag to the MCS, and moves it from the EZ to the EZDS in its own records. The MCS keeps track of individual silos. The internal silos deposit into a hopper before material is loaded into an IBC. Every hopper is equipped with weight sensors and the measured weight is subtracted from the internal silos content.

2.4.2 Bulk dry materials, small quantities

Dry materials used in smaller quantities, less than one unit (e.g. a bag), are stored in totes. When the material is required for a mixture operators can scoop out the required amount. A tote is initially filled with an entire bale of material, and then stored in a separate section of the high-rise. The change in location from a bag, or similar container, to a tote is stored in the MCS. The used bales are removed from the high-rise inventory.

The aforementioned totes are used to fill IBCs manually. Operators scoop from a tote, containing one material, to a tote specific to an order. Operators are granted some lee-way when measuring out materials for an order. The amount deposited is weighed and adjusted in the order.

2.4.3 Fluids

Fluids are stored in two separate warehouses, one for the bulk and a second for materials that are in use. In the bulk warehouse multiple cans are stored on one pallet, each pallet has a unique carrier code. When inventory is moved from the bulk warehouse to the "in-use" warehouse usually a single can is transferred.

The available inventory of a product remaining in a can is calculated by deducting the amount used in an order from the previous weight or volume. Similar to the totes, it is not realistic to measure out the exact amount required by the order. The extracted weight is corrected on the order.

2.5 Inventory corrections

2.5.1 Large bulk

Silos are equipped with level sensors that indicate the volume of material inside of the silo. When the level reaches the sensor, the remaining inventory is corrected based on the volume that should be in the silo and the material specific weight. As materials can suffer from bridging inside the silos a level sensor can be activated while there is still material above it. This results in an incorrect inventory record.

It is possible for the system to retrieve materials from a silo while it is, simultaneously, being filled with new materials. The track and trace then registers the resource using the code from the previous batch, or one that was digitally still in inventory. This results in an error in the inventory records in the MCS. The transaction can also not be completed between the ERP and MCS as the ERP does not allow the use of batch code that was not planned on an order. This results in an error in the ERP.

2.5.2 Small bulk

The bags used to fill the totes claim to be a certain weight, this is often not the case. The deviation from that weight is digitally corrected using scales when they fill a tote. The small deviations in actual and expected weight of the boxed materials are resolved in the MCS by booking inventory to and from "clarification storage". A box weighing less than expected is a clarification storage input. A box weighing more than expected has this extra weight compensated from the clarification storage. An issue here is that the batch codes are more than likely different. This error in the record has negative consequences for the ability to track mixtures.

A tote containing insufficient material for another recipe is discarded, and the remainder is subtracted from the inventory. If this manual operation is not applied properly or the transaction is not synced between the ERP and MCS therefore often results in an inaccurate inventory record.

Alternatively, an operator can deposit an entire product bag into an IBC. The product is not weighed when it is used this way. The weight added to the order is the expected weight in the bag. As previously stated, this is often not the correct weight also resulting in an error in the record.

2.5.3 Fluids

As fluids are known to the system as complete pallets but moved by operators as a single can. This typically leads to problems with the digital inventory as it cannot transfer the correct sized unit. This results in either the entire pallet or nothing being moved in the ERP and MCS, necessitating a manual correction.

Completely emptying a can is not realistic, a small amount always remains inside. This is corrected to zero when the can is discarded.

Fluids follow a flow that is similar to the small bulk materials as they are also measured out by hand. A key issue that was found during the data analysis was that, when preparing liquids for a mixture, operators would pour out the required amount before scanning the cans. The MCS retrieves the required amount from the cans in its inventory, potentially missing the fact that multiple batches were applied or retrieving old inventory from the “clarification storage” with incorrect batch codes. The Fluid warehouse also has two operators that perform a weekly clean-up. During this process, they look at the cans that are almost empty and discard them. Due to deviations in the weight, they have to perform many of these corrections, which increases the chances of inaccuracies persisting in the record.

2.5.4 Undesired or desired correction

Euroma differentiates between desired and undesired corrections. This section will illustrate the difference between these by providing a clear example. Desired corrections can be defined as corrections to the inventory as a result of slight deviations from the expectation. As mentioned before when an operator fills a tote using a bale the MCS initially only knows the weight claimed by the supplier of the bale, Figure 9. Typically, the filling of bales, or any type of container, involves a slight deviation from the target mean weight, the standard deviation. It is also possible a small amount of material sticks to the inside the bale. The corrections resulting from these small deviation are classified as desired corrections, they bring the inventory status closer to the actual value.

It can also happen that an operator at the mixing preparation receives a pallet where the expected number of bales is three, but the pallet only contains 2 bales, Figure 10. The missing bale is corrected in the inventory level. An entire bale going missing can result in mixtures not being completed and is not dependent on a measurement deviation. These corrections are classified as undesired corrections.

Figure 9 Desired correction of deviating weight in bale

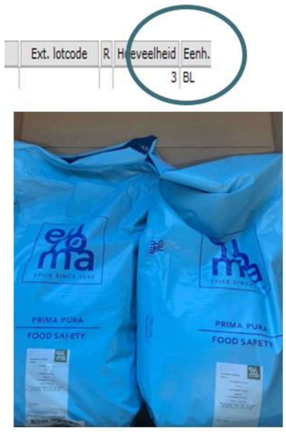


Figure 10 Example of an undesired correction

2.6 Inventory counting

Currently Euroma does not apply scheduled counting procedures over a large part of its inventory. For some of the inventory locations regular counting also is not feasible as the materials are stored in bulk. The inventory locations that are monitored are the external silo's and the pallets in the high-rise.

The measurement of the external silo's has been introduced recently and requires a person to climb on top of the silo and use a laser to measure the level of the silo. This level can be translated to a weight using the volume density of the material inside. Measuring the silos is currently done once a week.

Pallets that are received from the high-rise, either at the order pick stations or at the mixing preparation are counted by the operators at these stations. The system will ask the operator if there is an amount X of boxes, or bags, on the pallet. If the operator concludes that this is not the case they are asked to provide the correct number of boxes or bags and this amount is copied into the record.

2.7 Data analysis

For this research a dataset of 2022 transactions is used. This dataset contains all the recorded transactions, for a total of 290003 transactions. An analysis was performed on this dataset to get a better understanding of it and be able to apply that knowledge in a simulation later in this research.

2.7.1 Analysis of the transaction data

The analysis focuses on the location of transactions, the distribution of the size of the transaction. From this analysis the observation was made that the size of the transaction can be approximated with either gamma, lognormal or, normal distributions. Figure 11 shows the distribution of the transaction sizes of three different SKUs in three different warehouses. As is evident from the figure, the gamma and lognormal approximations are decent for representing the transactions. These values were gathered for each of the possible combinations of SKU and warehouse.

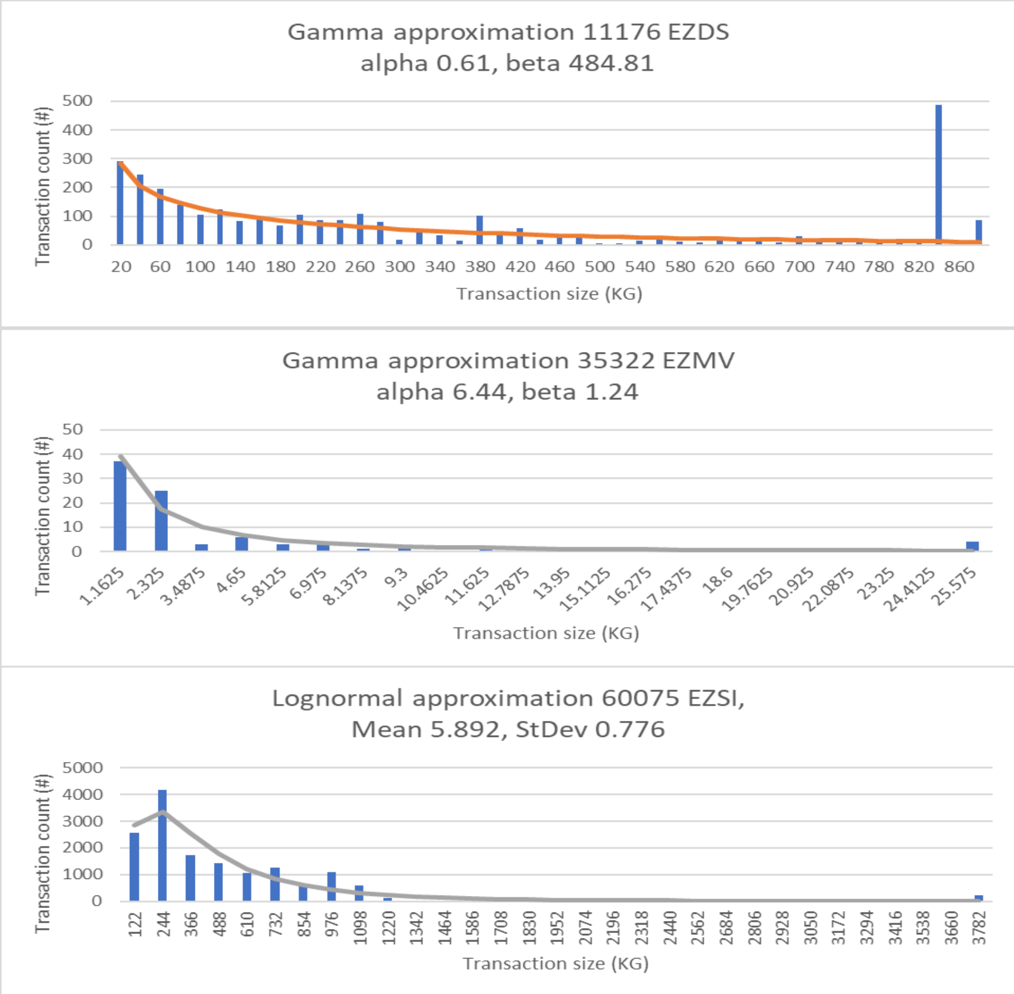


Figure 11 Examples of approximated distributions of transaction sizes

2.7.2 *Mass balance 2022*

An analysis of the transaction data from 2022 has been performed. In this analysis the transaction data of 2022 has been used. The goal of this analysis is to determine if all batches that are introduced are used to completion. For each SKU the number of batches used completely in 2022 was derived. To mitigate neglectable amounts of shrinkage a batch is considered to be used completely if the recorded used weight is within 50 grams the total weight. This turned out to be 89.56%. The total percentage of batches per SKU is shown in Figure 12.

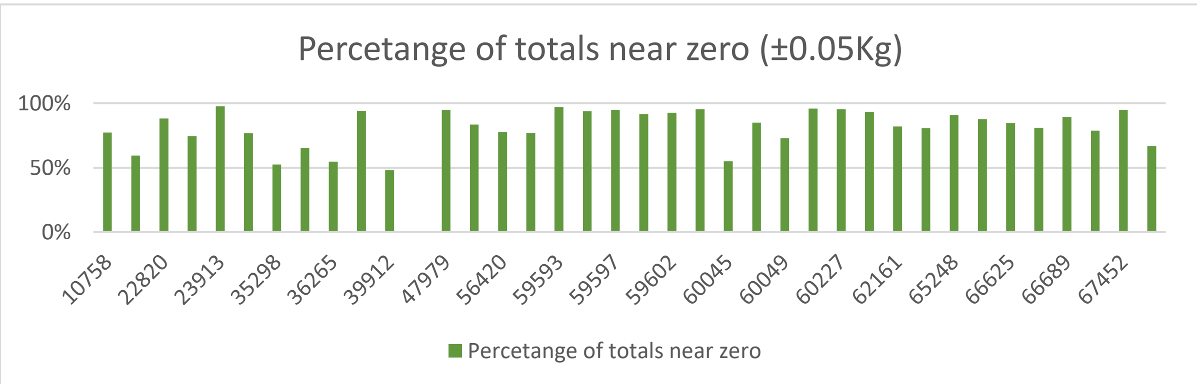


Figure 12 Percentage of batches used to within 50 grams of recorded weight

This representation of the data does not account for the batches that were carried over from one year to the next. A batch that was started in December will more than likely not be used completely by the end of the year. The number of batches that are in use at any given time is usually around two. For the entire year of 2022 it would be expected that around four batches would have inventory remains larger than 50 grams.

Figure 13, shows the number of batches that do not end within the allotted range of 50 grams. Over all SKUs, 50% of the batches were not used to completion. In this percentage, batches that are carried over to the next year, are accounted for. SKUs that have a large number of incomplete batches will be investigated in more detail to find reasons for the inaccuracies.

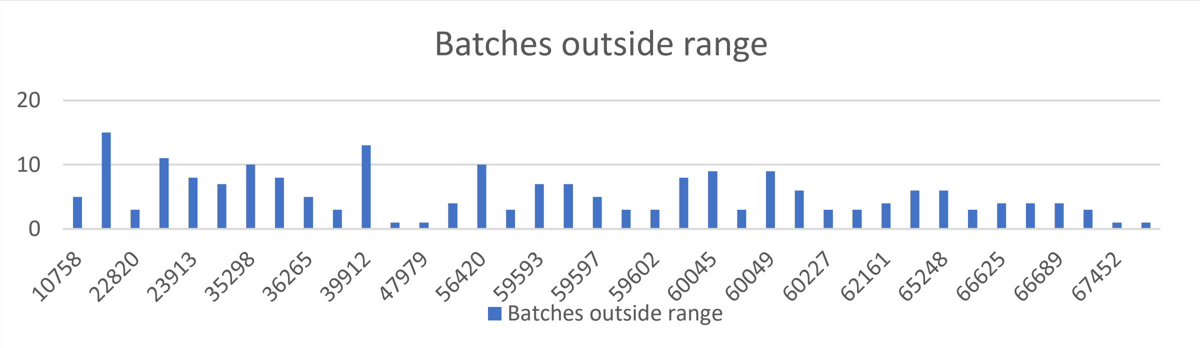


Figure 13 Number of batches outside of 50 gram range

A batch not being used completely does not directly indicate an error in the record. However, as Euroma operates with a “first in first out” principle on its in-use inventory it can be assumed that if many batches were not used completely and that there are errors in the inventory records.

2.7.3 Transactions errors

Comparing Appendix B & C it is immediately evident that one of the systems has a larger variety of transactions. From this, the assumption was drawn that not every transaction was synced properly.

Reviewing the transaction data and comparing which transactions were synced correctly showed that many of the manual transactions on the inventory are not conveyed from one system to the other. If an operator records the change of location or adds materials to an order manually this is not recorded in the ERP, resulting in inaccurate inventory records.

The MCS transactions in the range 400 are not communicated to the ERP system, this has a detrimental effect and the traceability of the products produced at Euroma, as well as the requirement of manual correction to keep the process from coming to a stop. The corrections are not verified and can lead to more issues at a later point in time.

The transactions that are logged properly are of type 100, 200, and 300. These transactions can be represented by a receival or a supply.

2.8 Conclusion

In this chapter, the main stakeholders were identified as the company Euroma, and the underlying departments in the company can be viewed as individual stakeholders. The current approaches to inventory accuracy and how this is measured are discussed. The tracking of inventory has been highlighted for the different types of material flows. This research will not split these types when performing the predictions. In the ongoing process, Euroma applies corrections, both desired and undesired. The difference between the two is discussed. Lastly, an analysis of the data is provided. This analysis shows the level of error in the inventory record and lies at the foundation of the simulation that is performed during the cycle count experiments.

3 Literature review

This chapter of the report discusses the relevant scientific literature on the subject of inventory record inaccuracy and approaches to improve the accuracy and solve incorrect corrections. Section 3.1 will provide a definition of inventory record inaccuracies and methods to measure this that are presented in literature. Section 3.2 will focus on inventory counting approaches. Prediction/ classification models will be covered in section 3.3. Both types of prediction models will be investigated, classification is of interest as the status of a record is either accurate or not, and regression is of interest as the predicted value is continuous, showing the level of the inaccuracies.

3.1 Inventory record inaccuracy

This section discusses what inaccuracy of inventory records constitutes according to the literature. Section 3.1 will define inventory record inaccuracies, 3.1.1, and methods to measure this that are presented in literature 3.1.2. This section also covers the main drivers behind IRI.

3.1.1 Definition of Inventory Record Inaccuracy

Accuracy is "the state of being exact or correct" (Oxford University, 2023). Inventory record inaccuracy, as the name suggests, refers to the deviation of the recorded inventory status from the physical stock (DeHoratius & Raman, 2008) (Iglehart & Morey, 1972) (Schrady, 1970). Inventory record inaccuracy is an issue experienced by every firm, for retailers it is even considered the norm (Chuang & Oliva, 2015).

3.1.2 Measurement of IRI

The most common method of calculating the inventory record inaccuracy is absolute deviation (Kang & Gershwin, 2005) (DeHoratius & Raman, 2008). Comparing the inventory record to the result of a count and determining the total percentage of SKUs, where the record and the physical match perfectly, gives the accuracy. Kang and Gershwin defined this as "*perfect inventory accuracy"* (Kang & Gershwin, 2005). Kang and Gershwin studied 500 stores and showed that the majority of the stores had a "*perfect inventory accuracy"* of 51%. The reason for using the absolute deviation is because it has the property of capturing the uncertainty of the inventory management process in a single measure as it reflects the mean and the spread if the discrepancy distribution (DeHoratius & Raman, 2008).

$$Inventory\ accuracy = \frac{Accurate\ SKUs}{Total\ SKUs} \tag{3}$$

Literature also proposes measuring the inventory accuracy with a predefined tolerance. Kang and Gershwin proposed allowing a deviation of five units between the record and the physical count. The average accuracy rose to 76%, showing that in most cases the record was only off by a small amount, 20% of the SKUs deviated more than six units (Kang & Gershwin, 2005). Similar results were presented by Raman et al. in a study of a single retailer with 370,000 SKUs where accuracy amounted to 65% (Raman, DeHoratius, & Ton, 2001). Schrady defined an inventory record to be accurate when the recorded and actual inventory level are within 1% difference (Schrady, 1970). Brooks & Wilson mention a case where tolerances differ per SKU, ranging 0-5%. It was also noted that fine tuning these tolerances has a low return on investment as companies using 5% are not less productive compared to those that did tune their tolerances (Brooks & Wilson, 1995)

$$Inventory\ accuracy = \frac{(1-\%)\ Accurate\ SKUs}{Total\ SKUs} \tag{4}$$

Financial measurement of the inventory is also common practice, especially for periodic financial statements. However, this measurement is not useful on the factory floor as the dollar value might be correct but the underlying records are not. Brooks & Wilson noted that a fiscal accuracy is commonly around 97%, while the item-by-item accuracy is between 30-60% (Brooks & Wilson, 1995).

This thesis introduces two measures as derivatives of the those present above. The first measure is the *perfect inventory record accuracy (PIRA),* the second is the overall perfect inventory record accuracy (OPIRA). The equations are denoted below (3 & 4)

The perfect inventory record accuracy aims to capture the number of transactions for which the inventory record was accurate. This KPI aims to capture the accuracy of an SKU over time.

$$Perfect\ Inventory\ Record\ accuracy = \frac{Transactions\ SKU\ without\ discrepancy}{Total\ Transactions\ SKU} \tag{5}$$

The overall perfect inventory record accuracy (4) combines the PIRA for all SKUs. This provides an indication of the accuracy of the entire inventory over time by taking the average of the PIRA for all SKUs.

$$Overall\ Perfect\ Inventory\ Record\ accuracy = \frac{\sum_{n=0}^{N} PIRA}{N} \tag{6}$$

3.1.3 Origins of IRI

Literature describes the six main reasons for inventory record inaccuracy; shrinkage, transaction, or record, errors, misplacement, supply yield, incorrect product identification, and incorrect corrections (Chuang & Oliva, 2015) (Khader, Rekik, Botta-Genoulaz, & Campagne, 2014) (Kang & Gershwin, 2005) (Rinehart, 1960).

Shrinkage refers to the loss of product and is generally the result of spoilage or theft (Chuang & Oliva, 2015). While spoilage is generally noticed and the inventory status corrected accordingly, theft typically goes unnoticed until a problem arises.

Transaction/ record errors occur when the status of inventory is changed. Typically this happens at the outbound and inbound sides of a firm (Kang & Gershwin, 2005). In the case of a retailer that means either an error at the supplier or the cashier. In a production environment there are typically multiple "embedded" in- & outbounds between the different production processes as well as at expedition increasing the opportunity for transaction errors. Transaction errors mainly affect the level of the inventory record and not the physical inventory (Khader, Rekik, Botta-Genoulaz, & Campagne, 2014).

Misplacement of inventory usually results in temporary discrepancy in the inventory status (Khader, Rekik, Botta-Genoulaz, & Campagne, 2014). When inventory is stored in the wrong location and it is forgotten about, it can become "lost". It can be expected that the stock is later on found and the discrepancy is resolved. However, until that point in time the inventory is unavailable (Kang & Gershwin, 2005).

Supply yield can have an effect on the inventory records when the yield of a production or supply system is below expectations. If the error on the physical quantity could not be detected by the inventory systems the records may become inaccurate (Khader, Rekik, Botta-Genoulaz, & Campagne, 2014).

Incorrect product identification can occur when a barcode or other identifier is misplaced or missing. If a mutation to the inventory for that item is introduced, it will result in the wrong inventory record being changed (Kang & Gershwin, 2005).

Incorrect corrections, as it suggest, are corrections made to the inventory record that have an adverse effect on the accuracy. A study investigating the causes of discrepancies in supply operations for a US government agency determined that 80% of the discrepancies found in the inventory record were the result of activities aimed to resolve these exact discrepancies (Rinehart, 1960).

3.2 Inventory counting

In this section an overview of inventory counting will be provided. Section 3.2.1 will focus on KPIs that provide insight into the performance of the inventory counting. Section 3.2.2 will look different approaches to selecting a set of SKUs to include in a count.

3.2.1 Inventory counting KPIs

Counting is aimed at finding the cause of inventory record errors, mediate their effects and prevent them from occurring, while also providing a correct overview of the current inventory (Rossetti, Collins, & Kurgund, 2001). Accuracy of the inventory record is a good KPI for the cycle count as well as the discrepancy on the record (Gumrukcu, Rossetti, & Buyurgan, 2008). Gumrukcu et al. mention a number of other performance indicators in three categories Table 3 provides an overview of all measures.

Table 3 Performance measures for cycle counting Gumrukcu et al. (2008)

Performance	**System**	**Cost (annual average)**
Accuracy	Fill rate	Holding costs
Discrepancy	Probability of lost sales	Asset costs
	Probability of backorders	Cost of lost sales
	Inventory	Transportation costs
		Cost of cycle counting

3.2.2 Approaches to cycle counting

A common method that is utilized in many production, retail and storage environments is counting of inventory at pre-specified locations (DeHoratius & Raman, 2008). A person physically goes by these locations and counts the units of inventory that they find on a predefined time interval. Two forms of counting can be distinguished, either including the entire set of SKUs (wall-to-wall counting), or a specific set (cycle counting) (Rossetti, Collins, & Kurgund, 2001).

Selecting which SKUs to include in a cycle count can done using one of six approaches that follow; random sample, ABC, opportunity based, transaction based, and location based (Rossetti, Collins, & Kurgund, 2001) (Brooks & Wilson, 1995).

Random sample cycle counting selects a sample of SKUs at random to perform a count on. Every SKU has the same probability of being included in the count. Random sample is generally considered as the best measure of inventory record accuracy under a stable and sufficiently large sample.

Two variations on this counting technique are mentioned: constant population and diminishing population. Constant population consists of the sample being drawn from the same group every time, allowing for SKUs to be counted multiple times, or being left unchecked. Diminishing population solves this issue by moving the counted SKUs from the sample set to a separate set. This second set becomes the samples set when the initial set is depleted. This allows for an SKU to be counted in the last sample and the first of the next cycle. This is prevented by including a timing restriction (Brooks & Wilson, 1995).

ABC cycle counting is based around the ABC classification common to inventory management. Typically, SKUs are categorized based on the total annual usage dollars, however, Rossetti et al. also propose frequency of issue, length of lead-time and criticality of equipment usage (Rossetti, Collins, & Kurgund, 2001). Regardless of the categorization method the counting approach remains the same. SKUs in of type A are counted most often, and the least amount of time is spend on SKUs of type C.

An issue with ABC cycle counting is the fact that the differentiation is typically based on economic value, while A & C class items can be equally important for production (Rossetti, Collins, & Kurgund, 2001). Brooks & Wilson identified an issue with ABC cycle counting in situations with a large variety of SKUs. Counting all SKUs within one year requires a number of dedicated counters, with increasing variety of SKUs the required number of counters increases as well (Brooks & Wilson, 1995).

Opportunity based cycle counting focuses on counting SKUs at key events in the process, such as the moment of reorder, stowing, when the level drops below a threshold, or when it is issued. Using planned data a decision can be made on whether a SKU can be counted or if it should be ignored. An argument against this approach lies in the fact that an item could be at a critical level but is not counted as it is not used often (Rossetti, Collins, & Kurgund, 2001).

Transaction based cycle counting counts SKUs based on the number of transactions since the last count. SKUs that experience a higher number of transactions are more likely to have an incorrect record. For this reason this approach focusses on counting the SKUs with the most transactions since the last count (Rossetti, Collins, & Kurgund, 2001).

Process control cycle counting is, according to Brooks & Wilson, controversial in theory but effective in practice and is bound to two prerequisites. First prerequisite, “Inventory records must have piece count by multiple location capability”, meaning all SKUs can be stored in any available location and the amount stored there can be recorded. Second prerequisite, “An inventory record listing of all quantities in all locations for all parts is available to the cycle counter”, meaning the counter has information on the inventory status of an SKU per location. With the prerequisites fulfilled, cycle counting samples are selected based on location, ease of counting and obvious errors. A supervisor assigns a counter to a location, the counter knows the inventory records in that location and decides what to count based on ease of counting or obvious errors (Brooks & Wilson, 1995).

Location based cycle counting has similarities with process control cycle counting excluding the knowledge on the inventory records and the discretion of the counter in choosing which SKUs to count. Characteristics of the SKUs are not included in the decision to count (Rossetti, Collins, & Kurgund, 2001).

3.3 Inventory record error

This section will cover models from literature used to predict errors in inventory record and, or, potential stockout events. Section 3.3.1 will focus on identifying errors in inventory records and the models that could be applied to do this. Section 3.3.2 will provide insight in the different types of models that were encountered in section 3.3.1. Finally, insight into the evaluation of the model performance is provided in section **Error! Reference source not found.**3.

3.3.1 Identification of inventory record error

Sheppard and Brown (1993) looked at eight drivers for inventory record error as a means to predict inaccuracies in the records, see Table 4. Using the eight features they constructed a discriminant model which was tested using two experiments. The first experiment started with a baseline count, followed by three months of not counting. This was followed by the second experiment. This started with a count that was compared with the perpetual records since the first count and was also followed by three months of not counting. The model was shown to be 75% and 74% accurate in predicting record error in the first and second experiment respectively (Sheppard & Brown, 1993).

Table 4 Applied features in Sheppard & Brown (1993) and Wijffels et al. (2016)

Features Sheppard & Brown (1993)	**Features Wijffels et al. (2016)**
Transaction frequency	Inventory position
Quantity in the transaction	Quantity in
On-hand inventory	Quantity out
Weight counting	Weight
Total dollar value	Unit value
Unit value	Recommended retail price
Number of places used	Rate of sale
Difficulty to track	Is pickable

As was also stated by Sheppard & Brown (1993), it is often assumed that the frequency of transactions on a SKU is directly correlated with the error in the record. This assumption holds in some, but not in all cases. Wijffels et al. avoid this assumption by applying data mining to classify items to be accurate or inaccurate based on a classification model trained on historical data. Wijffels et al. applied both a logistic regression model and a neural network. A random and an inventory-based policy were also investigated. The features that were used are presented in Table 4. The performance of the neural network and the logistic regression were very comparable and able to detect 75% of the inaccuracies by reviewing half the records. Both approaches outperformed the random approach, and also the inventory-based approach when the number of observations increases (Wijffels, Giannikas, Woodall, McFarlane, & Lu, 2016).

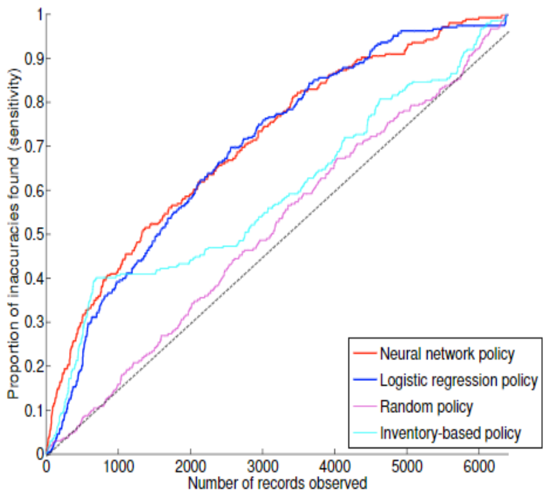


Figure 14 Experiment results Wijffels et al. (2016)

Both the models applied by Sheppard and Brown and Wijffels et al. are referred to as a supervised classification models, meaning the model is trained on a dataset in which inaccurate records have been labeled, *accurate* or *inaccurate*. The model correlates certain item properties and historical data to later on infer which records fall in a particular class when provided with new data.

Regression algorithms have not been widely applied in the literature to predict the inaccuracy of the inventory record. Kurian et al experimented with a variety of regression models to predict stockout events on inventory records (Kurian, Maneesh, & Pillai, 2020). They reported relatively high sensitivity but lacking levels of specificity, the models were not very accurate in their predictions. De Santis et al. used regression models to predict material backorders. They reported very good results, averaging around 95% accuracy in prediction, using any of the regression models selected for their experiments (De Santis, De Aguiar, & Goliatt, 2018). While these studies did not focus on prediction the error in their inventory record both the stockout probability and the backorders can be regarded as outcomes form inventory error.

3.3.2 Machine learning models

3.3.2.1 Classification

Some well-known supervised classification models are logistic regression, decision (classification) tree, random forest, support vector machine, k-nearest neighbor, naïve bayes, and neural network.

Logistic regression finds a hyperplane as a means to separate the data in different classes. The models aims to maximize the conditional likelihood of data points adhering to the same class. A sigmoid function is used to determine the probability of a class. Depending on a predefined threshold the data falls in one of the categories. Logistic regression is widely used for binary classification problems (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995).

Decision trees are among the most easily visualized forms of machine learning. Every node in the tree constitutes a question that needs to be asked. The branches that can be chosen as the answers to this question lead towards the next node. The tree ends in one of the classes that the model aims to determine (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995). An example decision tree is provided in Figure 15, the tree is trained on three classes.

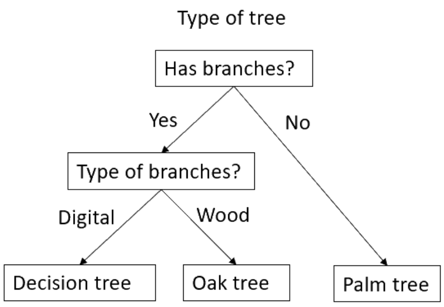


Figure 15 Example decision tree

Support vector machine (SVM), similar to logistic regression, aims to construct boundaries between the data points of different classes. Unlike logistic regression a SVM is able to create an arbitrary number of boundaries, allowing for more detailed classification and making it better suited for big data. SVMs can construct linear and nonlinear boundaries. Linear boundaries are constructed using the straight line equation, either side of the line represents a domain (Suthaharan, 2016):

$$wx' + \gamma = 0 \tag{5}$$

Nonlinear support vector machines classify data in n-dimensional space, n being the number of features in the data. The observations can be plotted in this n-dimensional space in the form of scatterplot. The advantage of nonlinear SVMs over linear is the ability to separate classes through the use of extra dimensions (Suthaharan, 2016).

K-nearest neighbor (knn) is based on the idea that observations that are close to each other are likely of the same class. K indicates the number of neighbors that the model aims to find in order to classify an observation. If, for instance, K = 5, the class of an observation will be set to equal the most frequent class of the 5 closest observations (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995). Figure 16 provides a visual example where the black dot represents a new observation. The five neighbors closest to this observation consist of three green and two red observations. From this, the knn model would conclude the new observation to be green.

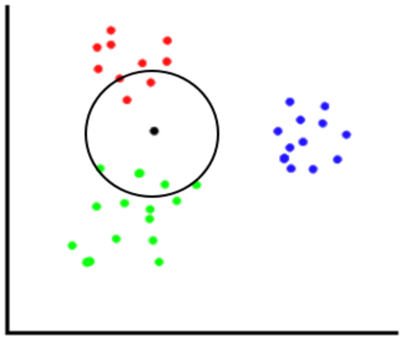


Figure 16 Example of k-nearest neighbour, black dot is a new observation

Naïve bayes is a classification model that assumes independence between the features of the data that it is presented, without needing to prove whether independence holds, hence the naivety. To obtain the classifier the assumption is made that the joint distribution of classes and attributes can be formulated as (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995):

$$P(A_j, x_1, \dots, x_n) = \pi_i \prod_{j=1}^{p} f(x_j|A_i) \qquad \forall i \tag{7}$$

The assumption of independence eases the process of finding the probabilities $\{\pi_i, f(x_i|A_i), \quad \forall i, j\}$. The result of the naïve bayes classifier is a probability of the observation being in a certain class, for every class. This provides more insight compared to other classifiers that only provide the most probable class (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995).

Neural network aims to emulate the working of the human brain. The network consists of layers of nodes, neurons. Each neuron is the weighted sum of its inputs (2) and depending on an activation function, or a threshold, the neuron fires or stays quiet (Fulkerson, Michie, Spiegelhalter, & Taylor, 1995).

$$y_k \begin{cases} 1 \; if \; \sum_j w_{kj}x_j - U_k \geq 0 \\ 0 \qquad\qquad otherwise \end{cases} \tag{8}$$

3.3.2.2 Regression

Linear regression is the continuous counterpart to logistic regression. Relationships between variables are modeled using linear predictor functions. The conditional mean of the response is assumed to be a affine function of the independent variables. The underlying formula for a linear regression model is given below (4).

$$Y_i = X_i\beta + \epsilon_i \tag{9}$$

$$\vec{\hat{\beta}} = (X^T X)^{-1} X^T Y \tag{10}$$

When training the model the value for X_i and Y_i can be observed and the model is fit by estimating the value for β. Y, X, β, and ϵ are vectors, the dimensions are provided in Table 5. The error coefficient, ϵ_i, is independent and identically distributed, with mean = 0 and $\mathrm{VAR}(\epsilon_i) = \sigma^2$ (Freedman, 2009).

Table 5 Linear regression variables (Freedman, 2009)

Y	n * 1	Vector of observable random variables
X	n * p	Matrix of observable random variables
B	p * 1	Vector of parameters
ϵ	n * 1	Random vector

Ridge regression is a multiple-regression model that provides increased stability when the dataset consists of highly correlated independent variables. The model decreases its error by shrinking the sum of squares of the regression coefficients in order to reduce overfitting of the model. In ridge regression the underlying function is similar to that in linear regression, (4). However, the function used to find the estimator β has been changed to include the ridge parameter λ and the identity matrix I (6) (Hilt & Seegrist, 1977).

$$\vec{\beta} = (X^T X + \lambda I)^{-1} X^T Y \tag{11}$$

Lasso regression is a method that performs feature selection and regularization as a means to improve the prediction accuracy and interpretability of a model. “Lasso” is an abbreviation for *least absolute shrinkage and selection operator*. Lasso selects a subset of the known independent variables to be used in training the model. Lasso regression was originally designed as an improvement to linear regression but can be extended to a variety of models. The underlying function for lasso regression aims at finding the value for an α & β that minimizes error in the prediction. The formula is shown below (7). Here x_{ij} represent the value of feature j in the i^{th} record and y_i represents the response variable (Tibshirani, 1995):

$$\left(\hat{\alpha}, \hat{\beta}\right) = \arg\min\left\{\sum_{i=1}^{N}\left(y_i - \alpha - \sum_j \beta_j x_{ij}\right)^2\right\} \quad subject\ to \sum_j |\beta_j| \leq t \tag{12}$$

Regression Trees are a form of supervised machine learning that aims at constructing trees to predict the response variable in the dataset. This type of model is similar to the decision tree in section 3.3.2.1, but focused on predicting a continuous value. Regression trees provide automatic feature selection, similar to lasso regression. The models tend to be computationally efficient and this allows for the addressing of large problems. As the model has roots in classification approaches it tends is able to form predictions using both numerical and nominal features from the available data. While regression trees have many advantages, they tend to be less accurate due to the piecewise constant approximations they provide. Regression trees are also unstable when small changes to the data occur. The underlying function of the regression tree is the sum of the products of the constants at each leave and a value P, which represent the logical assertion of the conjunction conditionals from the root to the current leave. As an example if l = 2 and from l = 0 to 1 X_2 >= 3 and from l = 1 to 2 is X_1 >=2, then P at leaf l=2 can be represented by X_2>=3 ^ X_1 >=2 (Torgo, 2017).

$$Y = \sum_{l \in L} K_l * I(P_l) \tag{13}$$

Random Forest is an ensemble approach to the tree based machine learning. Instead of constructing a single tree the model constructs an entire forest. The model output is provided based on the consensus of the trees in the forest. An added benefit of the random forest as a result of the Strong Law of Large Numbers is that overfitting becomes less of a problem (Breiman, 2001).

Extreme Gradient Boost, or XGBoost, is "scalable machine learning system for tree boosting" according to Chen & Guestrin. Gradient boosting is an ensemble machine learning approach that sequentially trains weak models, combines models and improve the underlying loss functions in order to attain a good prediction rate. The boost approach trains a set of trees each with a predictor function f_t. It then selects the function f_t that best improves the overall model prediction by minimizing the following objective, the first order gradient:

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y_1}^{t-1} + f_{t(x_i)}\right) + \Omega(f_t) \tag{14}$$

The constructed trees that failed to further improve the model output are retrained, combined or discarded until the model reaches its stop parameters (Chen & Guesterin, 2016).

3.3.2.3 Neural networks

Neural networks describes a form of machine learning in which models are constructed to mimic how the brain operates. Neural networks are built up in layers, the input layer, the output layer and a user defined number of hidden layers. Each layer consists of a set of nodes that process the incoming data. Based on the input, and an internally stored weight, a node can "fire" or stay quiet. The weights are estimated when training the model by adjusting them to provide the expected output given an input. When a node "fires" it conveys information to the next layer of the network where another set of nodes repeats a similar process, or if the output layer is reached the model will provide a result, usually in combination with a level of certainty. Neural networks are most commonly applied to classification problem but can also be applied for regression problems. (Jain, Mao, & Mohiuddin, 1996)

The formula that represents the working of a neuron is provided below (8). The output of a neuron, y, is calculated using the weight of input j, w_j, and the observed input j, x_j, which need to be higher than the threshold u. θ is a unit step function at 0. (Jain, Mao, & Mohiuddin, 1996)

$$y = \theta\left(\sum_{j=1}^{n} w_j x_j - u\right) \tag{15}$$

3.3.3 Performance metrics machine learning models

This section will provide details on how the model performance is validated and quantified. The performance of the models is evaluated by a selection of performance metrics. The underlying calculation, meaning and interpretation of the values will be discussed in sections 3.3.3.1 and 3.3.3.2.

3.3.3.1 Classification metrics

Performance of classification models is typically evaluated with one of the following four indicators: receiver operating characteristic (ROC), area under ROC curve (AUC), sensitivity and specificity. These are calculated using four measurements presented in Table 6.

Table 6 Classification model performance measures

Measurement	Description
True Positives	Number of observations predicted to be positive that are positive
False Positives	Number of observations predicted to be positive that are negative
False Negatives	Number of observations predicted to be negative that are positive
True Negatives	Number of observations predicted to be negative that are negative

Sensitivity

The sensitivity, or true positive rate, of a classification model indicates the models ability to predict true positives for every class that is has been trained on. Sensitivity is calculated using equation (4) (Tan, 2009).

$$True\ Positive\ Rate = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{16}$$

Specificity

The specificity of a classification model indicates the models ability to correctly identify a negative observation. Specificity is equal to 1 – the false positive rate. The equation for this value is shown below, equation (5) (Tan, 2009)

$$False\ Positive\ Rate = \frac{False\ Positives}{False\ Positives + True\ Negatives} \tag{17}$$

Receiver Operating Characteristic (ROC)

Analyzing the performance of a classifier can be done graphically using the ROC analysis. The axis of the resulting graph is the false positive rate on the x-axis and the true positive rate on the y-axis. These rates are calculated using equations (4) and (5) (Tan, 2009). Figure 17 gives an idea of how the ROC curve should look. The blue line shows the best performance, the red line shows the worst. The black dotted line represents how the model would have performed had it been completely random. The closer the curve is to that 50/50, the less predictive the model is.

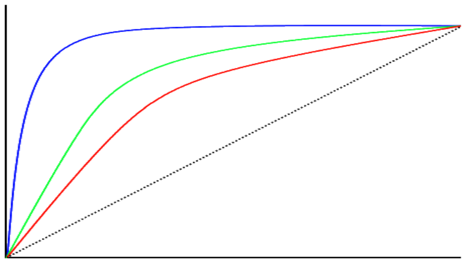


Figure 17 Schematic ROC curve

Area Under ROC Curve (AUC)

The area underneath the curve of the ROC is called the AUC and represents the degree of separability, the degree at which the model is able to distinguish between classes. A higher AUC typically indicates the model is able to distinguish well between classes and provides accurate predictions. An AUC of 0.8

indicates that the models chance of distinguishing one class correctly from the another is 80% (Hanley & McNeil, 1982).

The python package sklearn that also hosts the machine learning algorithms has combined the ROC and AUC into a single metric. This metric will be used to evaluate the performance.

3.3.3.2 Regression metrics

Evaluating the performance of the regression models, obviously, requires different metrics compared to the classification models. In many researches the metrics used for evaluating the regression models performance are the following:

- Mean Squared Error (MSE) or the rooted variant Root Mean Square Error (RMSE)
- Mean Absolute Error (MAD) or the percentage variant Mean Absolute Percentage Error (MAPE)
- Coefficient of determination (R^2) or Symmetric Mean Absolute Percentage Error (SMAPE)

MSE & RMSE

The mean squared error (11) or its rooted variant (12) provide an indication on the outliers in the model predictions. In equation 11 and 12 the X_i represents the prediction provided by the regression models, Y_i represents the expected value. Both MSE & RMSE metrics provide information on the average error of the models predictions compared to the expected output. Typically the preference is given to the RMSE. Taking the root provides a response in the same range as the dependent variable (Richard F. Gunst, 1977).

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2 \tag{18}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2} \tag{19}$$

MAE & MAPE

The mean average error and mean average percentage error are also commonly applied to measure the accuracy of a forecast or prediction. Both metrics indicate the average error experienced in the models prediction. In the case of the MAPE this error is made proportional to the actual value. A downside of that the MAPE introduces comes into play when the dependent variable can be 0. A value for Y_i of 0, as is evident from equation 14, results in a division by 0 and the MAPE becoming undefined (Chicco, Warrens, & Jurman, 2021).

$$\mathrm{MAE} = \mathrm{mean}(|e_i|) \tag{20}$$

$$\mathrm{MAPE} = \mathrm{mean}\left(\frac{|e_i|}{y_i}\right) * 100\% \tag{21}$$

R^2 score & SMAPE

The R^2 score is also known as the coefficient of determination and refers to the degree of which the dependent variable is determined by the independent variables, in terms of proportion of variance.

The formula for the R^2 score is depicted below (15). X_i represents the predicted values while Y_I represents the actual value of the dependent variable. An R^2 score close to 1 indicates that the independent variables are determinate for the dependent variable. A score of 0.3, or lower, indicates a poor relation between independent and dependent variables .

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(X_i - Y_i)^2}{\sum_{i=1}^{m}(\bar{Y} - Y_i)^2} \tag{22}$$

While the R^2 score provides a good indication of the model performance it is not a say it all value. A model with a low R^2 score can still perform decent while a model with a high score can perform poor due to a bias on the data.

3.4 Conclusion

The literature review identified the definition of inaccuracies as the deviation of an SKUs recorded inventory compared to the actual level. The overall inventory accuracy is found by dividing the number of accurate SKUs by the total number of SKUs. Literature showed that the measurement of inventory accuracy can also be based on the economic value, this is typically used for periodic financial statements. This approach is less useful in production environments. This thesis introduces two KPIs Perfect Inventory Record Accuracy, and Overall Perfect Inventory Record Accuracy based of the inventory record accuracy found in literature.

Form literature the main drivers of inaccuracies in the record are found. These are shrinkage, transaction/record, misplacement, supply yield, incorrect product identification, and incorrect corrections. In the case of Euroma the first three apply.

Rectifying the inventory record is realized through counting. The literature shows six approaches to cycle counting. Due to limitations at Euroma the cycle count approaches that will be tested with are ABC Classification, transaction-based, location based cycle count, and radom cycle counting. This thesis proposes a fifth approach using a prediction model.

Identification of inventory record inaccuracies using prediction models has been done before. Previous studies focused on the classification of inaccurate records. As the decision to cycle count is best supported by a continuous value this thesis investigates the use of regression models. Regression models have been applied for prediction stockouts and backorders, but no previous research on inaccuracies was found. Regression models used in the prediction of stockouts and backorders are: Ridge, Bayesian Ridge, Lasso, Decision Tree, Random Forest and XGBoost regression.

Neural networks can also be used to predict continuous values and can be used to perform predictions on more intricate datasets. For this reason neural networks will also be investigated.

4 Prediction and cycle counting framework

This chapter discusses the framework of the model and the sequence of operations. The model consists of two segments, namely, the prediction and the counting. The first section of this chapter discusses the general framework. Sections 4.2 and 4.3 explain the prediction and count model more in-depth, respectively.

Chapter 4 focusses on answering the fifth research question:

"How can the prediction and count recommendation model be constructed?"

Closing the knowledge gap represented by this question will be realized by comparing the results of several model alternatives.

The prediction models will be coded in Python using the Scikit-learn and XGBoost library. For the cycle counting mixed integer linear programming will be applied. For this Python offers the MIP package.

4.1 Model framework

Figure 18 shows the general framework of the model. The model consists of two segments, a prediction model and a cycle counting model. The prediction model uses transaction data to predict discrepancies in the inventory levels of SKUs at Euroma. This prediction is used as the input of the cycle counting model. The cycle counting model will provide a list of SKUs to include in the next cycle count.

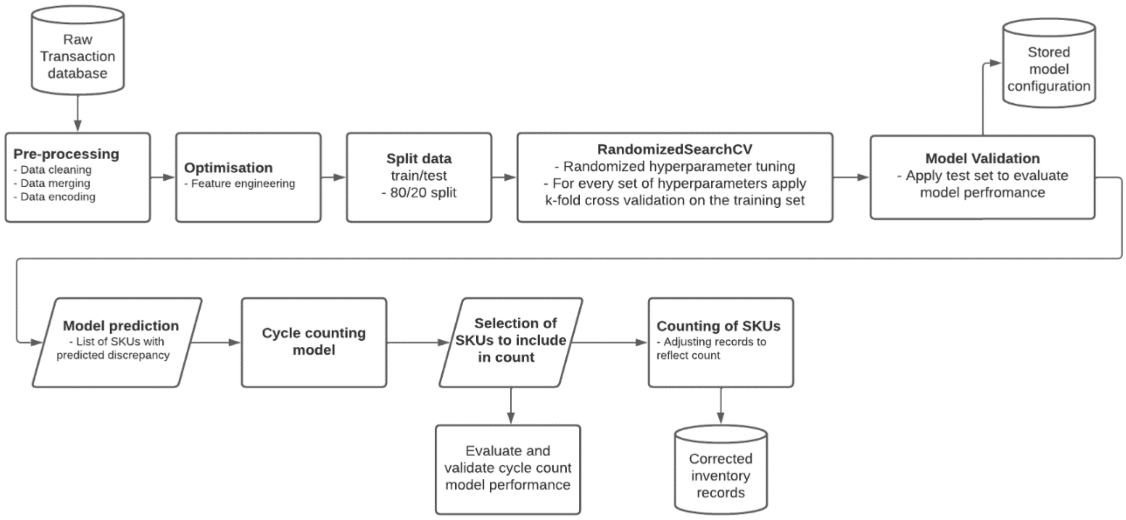


Figure 18 Framework for the prediction and cycle count model

Figure 18 shows the model starting with raw transaction data from the ERP and MCS. This data is processed to ensure that it can be applied to the model. After cleaning the data and merging the data from the ERP and MCS the features that are used for the prediction.

Data is split into a train and test segment, following common practice in machine learning. The former is used to tune the model parameters and the latter is used to validate the performance of the model on unseen data. After evaluating the model performance the configuration is stored. The configuration allows for skipping the training of the model and going straight to a prediction.

The model outputs a prediction of the discrepancy in the inventory record for each of the SKUs. This information is then used in combination with a cycle counting approach to provide a list of SKUs to include in the count and correct the records.

4.2 Prediction model

In this section the focus will be on the prediction model and the different steps that are required to receive a useful output. First an outline of the prediction model is provided, which is followed by a more in depth explanation of the data processing and preparations. Finally the training of the model will be discussed. The goal of the prediction model is to predict the discrepancy in the recorded inventory level and the actual inventory level. In order to perform this prediction the data from two systems, the ERP and MCS, are combined. This section of the report aims to answer the following sub-questions:

4.2.1 Data collection

This section answers the first sub-question of the fourth research question:

How is data introduced to the model?

The data that will be used to make a prediction on the inventory record error is logged by the ERP and MCS into a "Data warehouse" server. This server contains transaction data from both the systems and the data is already structured neatly for use in dashboards that Euroma applies to visualize their data. From this server an export is provided for both the ERP and the MCS transactions.

After importing the data, the model will have two datasets that contain information on the same transactions. Training the model or applying the data for inference will require a merge of the two datasets. The method for merging the data is described below in section 4.2.2.

4.2.2 Data pre-processing

This section discusses the different pre-processing steps are performed. The raw data is extensive but misses key features or has categorical features that will not work for a regression model. Data wrangling, encoding and feature selection are used to structure the dataset in a form that is applicable for the prediction model.

4.2.2.1 *Merging data*

Part of the preprocessing that is required is a merge of the datasets. Both the ERP and the MCS contain transactions for all SKUs at Euroma. The ERP contains transactions between different warehouses and the MCS contains transactions between storage locations. As there are five warehouses that, combined, contain over 140 storage locations. It is therefore obvious that the MCS dataset is much larger than the ERP dataset.

The data is merged by looping through the MCS transactions and filtering the ERP data. First the ERP data is filtered using the SKU number in the ERP transaction. The next filter that is applied separates the records on the warehouse where they occur. The third filter applies the weight in the transaction and, finally, the fourth filter uses the included timestamp. If this does not return a record or if more than one record is returned the dataset is filtered further. The time stamp is split by year, month, date, hour and minute. This is then used to create a five minute range around the transaction to match it.

When merging the data the columns in the ERP dataset are copied over to the MCS dataset to expand the dataset. The main goal of the merge is to copy over the inventory position at the time of transaction, the transaction type in the ERP. All available fields are copied over to prevent introducing a bias in the model.

Table 7 Dataset merge

ERP Transaction count	MCS Transaction count	Realized merges	Percentage of ERP merged
143.982	290.020	115.755	80.4%

Not all transactions from the ERP dataset are merged into the MCS data, as Table 7 depicts. Typically this can be accounted to the timestamp of the transaction, or transactions simply not existing in both systems. Randomly selecting a record to merge the record with does not constitute good form, therefore these transactions are omitted. The missing features are extrapolated later on, section 4.2.2.3 expands on this process.

4.2.2.2 *Categorical feature encoding*

The dataset contains many categorical features that need to be transformed for the use in regression models. The SKU number, transaction type, and warehouse are all categories. The transaction types for either of the systems can be found in Appendix B & C. The types are differentiated through a numerical code. This code should not be interpreted as a quantity by the model. Similarly, the locations are indicated using a code, in the ERP this is one of the warehouses as stated in section 1.2.2 the MCS has code for every location in the production environment.

One-Hot Encoding

Categorical features (numeric) can have an adverse impact on model performance in the case of regression algorithms. This can be resolved by applying one-hot encoding to create a dichotomous column for each SKU number and transaction type. For every possible value in the categorical feature column a column is created. This column can have either a one or zero value, indicating if the record in the dataset corresponds to the feature represented by the column, see Figure 19.

SKU number
10758
10758
23889
11176
22820
11176

SKU_Num_10758	SKU _Num_11176	SKU _Num_22820	SKU _Num_23889
1	0	0	0
1	0	0	0
0	0	0	1
0	1	0	0
0	0	1	0
0	1	0	0

Figure 19 Example of One-Hot Encoding

Integer encoding

Integer encoding can be applied to encode categorical data into a number format. Each category is then represented by an integer value ranging from 1 to N. N is the number of unique labels in a categorical feature, see Figure 20. A benefit of this approach is that the dataset does not increase in size, a downside is the fact that it is not immediately evident what the value is of a feature.

SKU number
10758
10758
23889
11176
22820
11176

SKU number
1
1
4
2
3
2

Figure 20 Example of integer encoding

4.2.2.3 Feature extrapolation

As stated in section 4.2.2.1, the main goal of merging the data is, mainly, to attain the inventory position after transaction value from the ERP database. It was also stated that there is a possibility of transaction not being matched. To ensure that the information on the inventory position is filled out for all datapoints it is extrapolated. The extrapolation of the features is described in the flowchart in Figure 21.

The datasets filtered based on the SKU number and warehouse to ensure that the extrapolation of the inventory position is only applied to one SKU in one warehouse. The data is parsed and the program tries to identify the first transaction where the merge step did not manage to include the required data from the ERP dataset.

When an inventory position is found the models works its way back to the first transaction where the inventory position is not available. The inventory position is calculated by applying the recorded transactions in reverse to the inventory level to fill the field in every transaction.

A similar approach is used to find the discrepancy at the time of transaction. Only instead of finding the first transaction with an inventory position, the first count instance is sought. The discrepancy of previous records can be extrapolated from the recorded discrepancy at the time of the count.

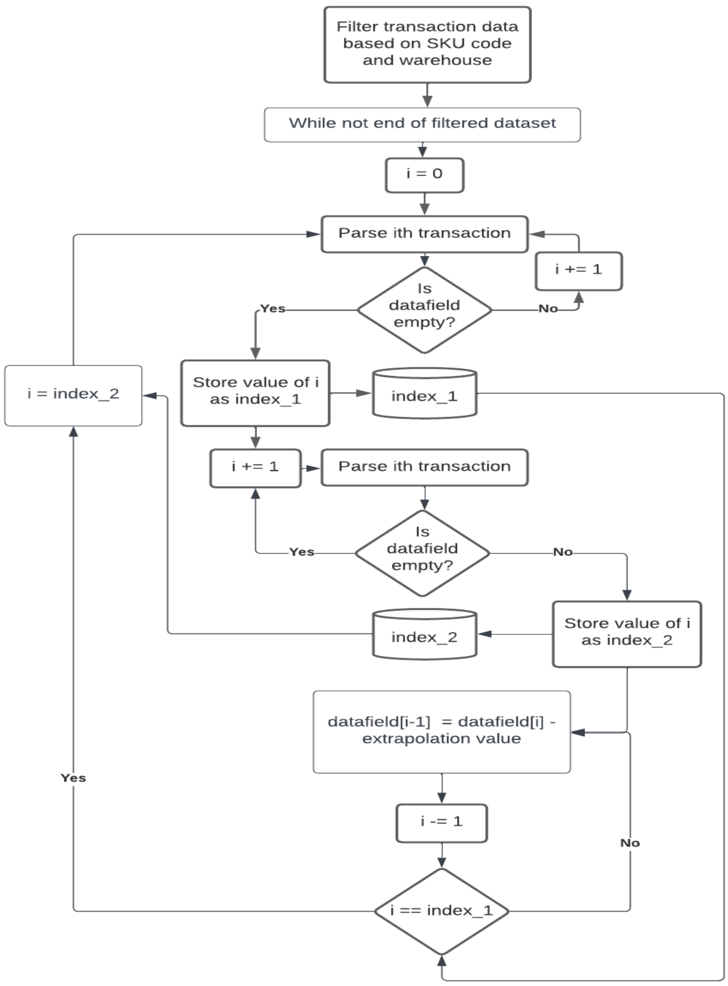


Figure 21 Feature extrapolation flowchart

4.2.2.4 Feature selection

The last step in the data preprocessing is feature selection. Feature selection constitutes filtering of the explanatory variables to find those that have been proven useful and correlated to the response variable. The features that will be selected in the final model are the results of experiments. There are a number of approaches to feature selection, commonly models are used to select features, mainly Lasso regression is used for this approach.

Forward feature selection has also proven to be a useful approach. Features are added one after the other based on the highest correlation to the response variable until a drop off in performance is experienced.

Backward feature selection, or elimination, is the reversed version of forward feature selection. This approach starts with the full set of features available and eliminates these based on their significance. Typically a significance of 0.05 is applied.

The final method that will be applied is a correlation heatmap. This is a visual representation of the degree of correlation between the features in the dataset. These experiments are described in section 5.1.2

4.2.3 Model training

After the transaction data has been cleaned and prepared it can be used for model training and prediction. The aim of this research is to determine if a prediction model can provide an indication of the actual inventory level using transaction records from two systems.

The literature review identified a number of machine learning models with potential for predicting inaccuracy in the inventory record. The possible models can be split in two segments, classication and regression models. Sheppard & Brown and Wijfels et al. applied classification models to determine the state of a record being correct or incorrect. While DeSantis et al. and Kurian et al. Applied regression models to predict backorders, and stockouts, respectively. For this research the interest lies with the actual level of the inaccuracy. This is a continuous value that can change with every transaction executed on that SKU. As the value being predicted is continuous the focus will lie with regression models as potential solutions. The models will be trained and evaluated using different hyperparameters to find the setup that can be applied best on the available data. A set of hyperparameters has been defined for each of the machine learning algorithms and are provided in Appendix E.

To validate the models ability to learn from the provided data learning curves will be plotted. Here the trainset is incrementally increased and the models ability to fit to the data during training and predict during testing is compared. An example of such plot is provided in Figure 22. The blue line represents the models ability to predict the response variable in the training set given a certain number of datapoints in the set. The green line represents the models ability to predict the response variable when it is provided with unseen data. Ideally the lines converge.

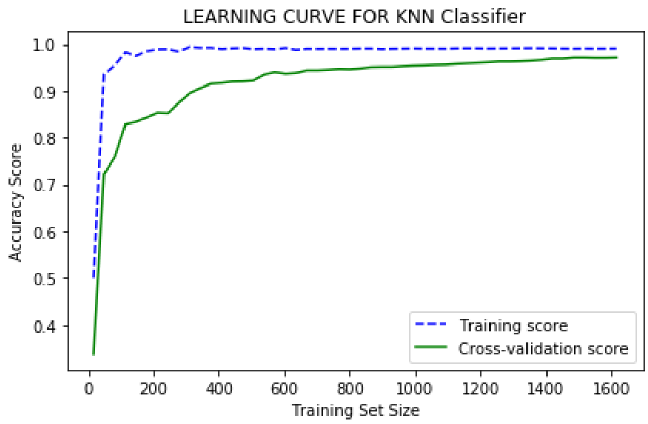


Figure 22 Example of learning curve plot

As previously mentioned the models will be trained and tested using transaction data of a selection of SKUs over the entirety of 2022. Training the models and evaluating the output is part of the experimentation phase of this research. The is will be discussed at length in Chapter 5.

4.2.4 Evaluation

After the model has been trained the configuration is used to perform predictions on the test dataset. The resulting prediction can then be compared against the original data to evaluate the model performance. The metrics that will be used as an indication of the performance are the R^2-score, the max absolute error (MAE) and, root mean-squared error (RMSE), these were discussed in Section 3.3.3.2.

Computational time is also an important performance indicator for this model. The transaction data being used for the prediction is generated quickly, with an average transaction rate of one transaction every two minutes. To keep up with the current data a low computational time is desired.

4.3 Sci-kit Learn machine learning model implementations

As stated in section 4.1 the full model can be divided into two segments, the prediction model and cycle counting model. For the prediction model the python package sci-kit learn and xgboost are utilized. The different models that will be experimented with can be called as a function in python after importing the proper libraries. This allows for easy testing with different parameters.

4.3.1 Prediction model alternatives and parameters

In this section the model alternatives that will be tested for the prediction will be discussed. Each of the modes discussed is aimed at predicting a continuous response variable. In the case of this research the variable being predicted is the discrepancy in the inventory record at the time of transaction.

For each of the implemented models the underlying parameters are explained. The parameters can be found in the documentation of the scikit learn package for python. Appendix E shows these parameters and the corresponding set of values that will be used for experimentation can be found in Appendix F.

Hyper parameter tuning and validation

Hyper parameter tuning is realized using the *RandomizedSearchCV* function available in the sci-kit learn package in python. The model is run 50 times, for every run a different set of hyperparameters is randomly selected. For every run, and set of hyperparameters, k-fold cross validation is applied to train the model. The performance of the different sets of hyperparameters is recorded and in the end the optimal configuration is returned.

K-fold cross validation

Validation of the performance of the prediction model is realized by splitting the dataset. The dataset was split in two segments, train and test. This was done so that the model could be tested on data it had not been trained on. Further improving the performance of the model is done by introducing k-fold cross validation. Here the trainset is subdivided into K sets, of which one will be used for validation and the other, K-1 sets, are used for training. After completing K folds of training the model is tested on completely new data. A visual representation of K-fold cross validation is shown in Figure 23.

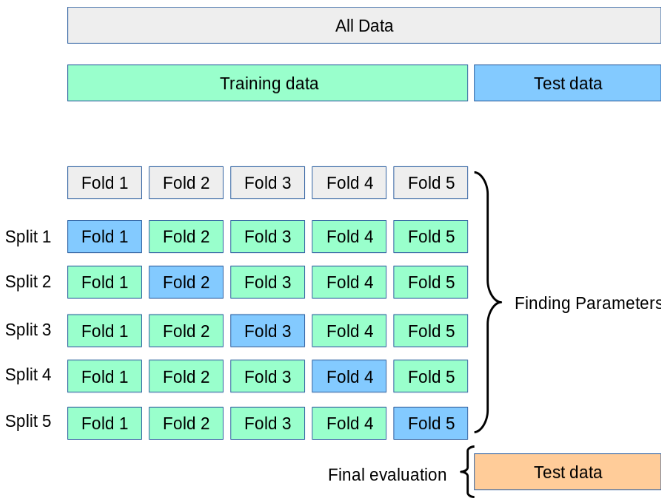


Figure 23 K-fold cross validation training with K = 5,

4.4 Cycle count model

This section first explains the setup of the cycle counting model. The implementation of the model will be discussed in depth as well as any changes that are required for the data to be used. The assumptions and restrictions to the model are discussed and finally, the output and interpretation will be explained.

The following paragraph describe the design of the model. The assumptions that are made to ease the modeling of a realistic situation are explained and argued. This is followed by the restrictions that are applied to the model to better approach a realistic situation are described. This information is then combined in the construction of the mathematical model that will be applied.

4.4.1 General mathematical cycle count model

This section describes the mathematical model for cycle counting is provided. The cycle count models described in the literature were all implemented as algorithms. The desire is to provide the used code as close as possible to the description in this thesis. For this reason the decision is made to use mixed integer programming (MIP). As no MIP model was found in literature this thesis first presents a general cycle counting MIP model. First the parameters and indices that are used in the model are introduced. In the case of indices, all values included in the set are also provided. This is followed by the decision variable or the model. Next the objective function is given and, finally, the constraints on model are provided. The assumptions and constraints will be discussed in more detail in sections 4.4.3 & 4.4.3

Sets	Definition
I	Set of all SKUs
J	Set of warehouses J ϵ {EZSI, EZDS, EZMV}

Parameter	Definition
S_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j has not been counted in current cycle} \\ 0 & \text{If already counted in current cycle} \end{cases}$
$\mathrm{E}[t_j]$	Expected time required to count a SKU warehouse j, $j \in J$
σ_{t_j}	Deviation in time required to count a SKU in warehouse j, $j \in J$
D_{ij}	Discrepancy of SKU i in warehouse j at the time of counting, $i \in I$,$j \in J$
AT	Available time for a cycle count
NS	Maximum number of SKUs to include in count
N	Number of items
W	Number of warehouses
Decision variable	**Definition**
$\mathrm{X_{ij}}$	$\begin{cases} 1 & \text{If SKU i in warehouse j is included in the count} \\ 0 & \text{Otherwise} \end{cases}$

Objective function

Traditional Appoach

$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \tag{4.1}$$

Constraints

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq NS \tag{4.2}$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq Y_{ij} \tag{4.3}$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} (\mathrm{X}_{ij} * E[t_j]) \leq AT \tag{4.4}$$

$$X_{ij} \leq S_{ij} \qquad \forall i,j \tag{4.5}$$

$$X_{ij} \in \{0,1\} \tag{4.6}$$

4.4.2 Assumptions

This section covers the assumptions that are made during the construction of the cycle count model. These assumptions serve to eliminate or model uncertainties in the process outside the scope of this research. Assumptions were made on two aspects of the process. The time required to count and the result of a count.

The time required to count a SKUs depends on many factors. For the purposes of testing and evaluating the performance of the model the assumption will be made that the time required to count an item in a particular location is normally distributed with some mean and standard deviation. Table 8 shows the expectation of the count time of an SKU in a warehouse. The values are derived from speaking with operators, no quantified measurements were recorded in the past and travel time between locations is not included.

Table 8 Expectation and deviation of count time per SKU in a warehouse

Warehouse	Expected count time
EZDS	5 minutes
EZSI	10 minutes
EZMV	20 minutes

The second assumption that is made is that travel times are zero. As mentioned in section 4.4.1 the travel times between locations have not been documented. The travel time is also very dependent on the chosen path and whether the counter encounters obstacles. Including travel time would reduce the number of SKUs that can be included in a cycle count and, obviously, impacts the model performance negatively. For the sake of simplicity travel time is excluded from the model.

Thirdly the assumption is made that the inventory does not change during the count. In a realistic situation a possibility exists of materials being extracted from the silos, or required for production in the MiniLoad warehouse, at the same time as when the counter is present. This would either delay the count or invalidate the result. This was excluded from the model. This assumption also ensures that, when performing a cycle count on inventory in the MiniLoad warehouse, the production process is not interrupted and all bins with the desired SKU are available to the counter.

The fourth assumption that is made pertains to the performance of the model. The cycle counting models does not take the possible error of the prediction model into account. This allows for comparison of the different count approaches without also evaluating the prediction model performance.

4.4.3 Constraints

In this section the constraints to the cycle count model are discussed. The constraints are included in the model to better approximate the performance under realistic circumstances.

Ideally every SKU is counted in every cycle as this would guarantee an accurate inventory record every time the count is finished. Due to the number of SKUs and time that would be required to count everything, it is not realistic to count everything. For this reason a number of constraints is introduced into the model that limits the time that can be allocated to the cycle count and the number of items that are included in a count.

The constraints are noted in mathematical format in section 4.4.5. The general model the constraints will be briefly elaborated on in this section.

The first constraint (4.2) limits the number of SKUs that can be included in the count. If this constraint is not included in the model would select as many SKUs as are available or until another constraint comes into play.

Constrain (4.3) ensures that the population from which SKUs are selected for counting is diminishes with every count until the completion of the cycle. Initial testing with the cycle counting models showed that this is a requirement as the model selected the same SKUs for counting over the entire experiment.

The third constraint (4.4) limits the time that can be spend on counting SKUs. Constraint (6.4) ensures that the model can only decide to count an SKU in a particular warehouse if this SKU is indeed stored in that warehouse. Without this constraint the model would not take location of the product into account.

The last constraint (4.5) ensures that the value of X is an integer, either zero or one. This ensures that, from a modeling perspective, only all available inventory of the SKU in a location can be counted and not a fraction.

Additional restrictions will be apply based on the cycle count approach being experimented with. These restrictions will be discussed in section 4.4.5 where the implementation of the different approaches is also discussed. The mathematical formulation of the added restrictions will also be available in that section.

4.4.4 Cycle counting model alternatives and parameters

In this section an overview of the different cycle count model alternatives is provided. For details on reasoning behind a particular model see the literature review.

The objective of the cycle count model is to maximize the total correction of the inventory record in the current cycle. This total correction can be calculated by summing the resolved discrepancy of all SKUs included in the count.

ABC cycle counting as it was also described in the literature separates the SKUs into categories of importance. This can be based on transaction frequency, economic value or another quantifying metric. For this research the classes will be determined by the total annual euro value of the SKU. Per class a set number of SKUs is counted each time. Higher class SKUs (A) are counted more often than lower class SKUs (C). Table 9 shows the combinations of the number SKUs to include in a count per class.

Table 9 ABC classes with # of SKUs to include in count

Class	**Number of SKUs of class to include in count**			
A	1	5	10	15
B	1	2	5	10
C	1	1	2	5

Transaction based cycle counting focuses on counting SKUs based on the number of transactions since the last count, with higher transaction counts getting priority. At any given point in time the SKUs can be sorted by the number of transaction since the last time it was included in a count. For SKUs with the same transaction count the SKUs will be sorted, in descending order, based on the predicted discrepancy. The number of SKUs that will be included in the cycle count is constrained by the time allowed for counting.

Location based cycle counting entails counting all SKUs in an area. In the case of Euroma the areas can be defined as the different warehouses. Location based cycle counting would work for the external and internal silos but is less applicable to the mini-load warehouse at Euroma. Retrieving all the inventory from the automated mini load warehouse would have a very large impact on the ongoing process. The number of SKUs that can be included in the count will be restricted by the time allowed to the counter. The models objective is to maximize the total correction over an area, the resulting output would therefore be a warehouse and a list of SKUs to include.

Random sample serves a control approach. By randomly selecting the SKUs to include in the count. The model will randomly select SKUs to include in the cycle count until the constraints are met. By comparing the performance of the other models to the random sample a statement can be made about the quality of the control decision.

4.4.5 Changes to general model

This section will elaborate on any additional sets, parameters, and constraints, and changes to the decision variable that might be necessary for a particular approach cycle counting approach. For the full model descriptions of the ABC-, Transaction based-, Location based-, random-, and predictive cycle counting see Appendices I, J, K, L, and M respectively.

From the literature three approaches were identified as promising cycle counting approaches. Namely, ABC cycle counting, transaction-based cycle counting, and location based cycle counting. Aside from these four the model will be run using the random sample approach. As a base line a random sample approach is included.

Random cycle counting

This approach to cycle counting there are no constraints as to what warehouse the SKUs are in, or what the discrepancy is at the time of counting. The model chooses a predefined number of SKUs to include from all available SKUs. The performance of the random approach serves as a base line to compare the other approaches with.

Random number generators are never completely random and rely on a seed value in the generation of their number sequence. In the experiments performed in chapter 5, the run number will be used as the seed to increase the variability in the chosen SKUs.

ABC cycle counting

ABC cycle counting divides the SKUs into a predefined number of categories. The ABC split is found in many aspects of inventory management. Typically the SKUs in category "A" represent 20% of the volume but 80% of the total value. For "B" items this ratio comes down to 30% of the volume and 15% of the value, "C" items cover the remaining 50% of the volume and 5% of the value. The combination with cycle counting is made by assigning a count frequency per class, with "A" class SKUs having a higher frequency. While it is common to use three classes, a,b, and c, the approach is not limited to three classes. In the case of ABC cycle counting additional sets, constraints will be introduced. These are noted below.

Sets	**Definition**
$\mathrm{Classes}_c$	Classes used in the abc approach, $c \in \{a, b, c\}$
Parameter	
NS_x	Maximum number of SKUs to include in count from category x, $x \in \{a, b, c\}$
S_{ijx}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j is in category x} \\ 0 & \text{Otherwise} \end{cases}$
Decision variable	
X_{ijc}	$\begin{cases} 1 & \text{If SKU i in warehouse j is of type c and is included in the count} \\ 0 & \text{Otherwise} \end{cases}$
Constraint	

$$\sum_{i=0}^{N} \sum_{j=0}^{W} X_{ijc} \leq NS_x \qquad \forall\, c \qquad (4.7)$$

$$X_{ijc} \leq S_{ijx} \qquad \forall i,j,x \qquad (4.8)$$

$$X_{ijc}, S_{ijx} \in \{0,1\} \qquad \forall\, i,j,x \qquad (4.9)$$

For this thesis the classification is kept to a minimum as the data available to create the distinction in the classes is also limited. The number of classes is kept to a, b, and c.

Transaction based cycle counting

For transaction based cycle counting the required change to the mathematical model is not a constraint but the objective function of the model. As was described in the literature transaction based cycle counting aims at correcting the items with the most transactions since the last time they were counted. First the set RC_{ij} is introduced containing the transaction count of each SKU since their last count. The objective of this model is to maximize the combined transaction count of the SKUs included in the cycle count, as is represented by the objective function (4.10). The transaction based cycle counting approach does not require any additional constraints.

Sets	**Definition**
RC_{ij}	Set that contains the transaction count since the last correction for all SKUs i in warehouse j, $i \in I,\ j \in J$
Objective	
	$\max \sum_{i=0}^{N} \sum_{j=0}^{W} RC_{ij} \times \mathrm{X}_{ij}$ (4.10)

Location based cycle counting

Location based cycle counting requires a constraint on the number of warehouses to include in a cycle count. As only three warehouses are available the options that will be tested with are either one or two warehouses. When including all three warehouses, location based cycle counting is the same as counting based on the highest error.

The modification to the general model that are required for location based cycle counting are the addition of the parameter W_j, indicating if SKUs from warehouse j are included in the count, and NW, the number of warehouses to include in a cycle count. To limit the number of warehouses that can be included constraint (4.11) is introduced. Constraint (4.12) ensures that SKUs can only be selected from warehouse j, if warehouse j is included in the count.

Parameter	**Definition**
W_j	$\begin{cases} 1 & \text{If SKUs from warehouse j are included in the count} \\ 0 & \text{Otherwise} \end{cases}$
NW	Number of warehouses to include in the cycle count, $\mathrm{NW} = 1$ or 2
Constraint	
	$\sum_{j=0}^{J} W_j \leq NW$ (4.11)
	$X_{ij} \leq W_j \quad \forall\, i,j$ (4.12)
	$W_j \in \{0,1\} \quad \forall\, i,j$ (4.13)

Predictive cycle counting

This research is aimed at evaluating the benefit of adding a prediction to a cycle counting model. This also constitutes the novelty of this research. While traditional cycle counting models simply select SKUs with the hope of finding inaccuracies, the predictive cycle counting model tries to make an informed decision on what to include in the count. This informed decision is based on the prediction that is based on historical data. The change to the general model noted in section 4.1 is the addition of the prediction to the objective function. The altered function is provided below.

Objective function

$Discrepancy$ $Approach$	$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij} * D_{ij}$$	(4.1.2)

4.4.6 Warehouse simulation

This section discusses the experimental setup for the cycle counting models. The simulation used to test the different approaches is outlined, as well as how the simulation is constructed.

Euroma is operated 24/7; therefore, performing many real-time tests is undesirable and would impact the factory's production capacity. For this reason, a simulation is constructed that best mimics the transactions at Euroma.

Euroma has a large dataset containing the historical transactions used in their production process. Counting the inventory has not been a structured process for the warehouses considered in the scope of this research. For this reason, knowledge of the occurrence and the severity of incorrections is based on assumptions grounded in the literature and speculation resulting from the transaction data analysis. The cycle count models will be tested using a simulated transaction dataset based on the 2022 data. The data for 2022 was analyzed to find the probability of an SKU being used in a transaction, the correlated warehouse probability, and the consequent packaging used.

The probability of a transaction being for a particular SKU was derived from the number of transactions divided by the total number of transactions. A similar approach is used to find the probability of the transaction being from a warehouse and type.

The Excel solver was used to approximate the transaction size for each combination of SKU and warehouse. As stated in section 2.7, a gamma or lognormal distribution is common. The other data is generated using the SKU, warehouse, transaction type, and size.

The simulation is presented using the flowchart in Figure 24. The warehouse is simulated on a transaction level; there are no underlying checks on whether there is space for inventory to be stored or available in the location where it originated. The simulation consists of a predefined number of runs, each with a set number of iterations. Each iteration is a transaction. If the prediction model is selected, each transaction is used for a prediction. The resulting predicted error is stored in a different column in the dataset.

The count interval used in the simulations is one day, with a cycle of one week using a diminishing population. The cycle count methods were introduced as MIP models to reduce the bias that a programmed algorithm might introduce, and to stay as close to the way the models are presented in this thesis. A transaction is generated for each of the selected SKUs that rectifies the discrepancy to zero and adjusts the inventory level accordingly.

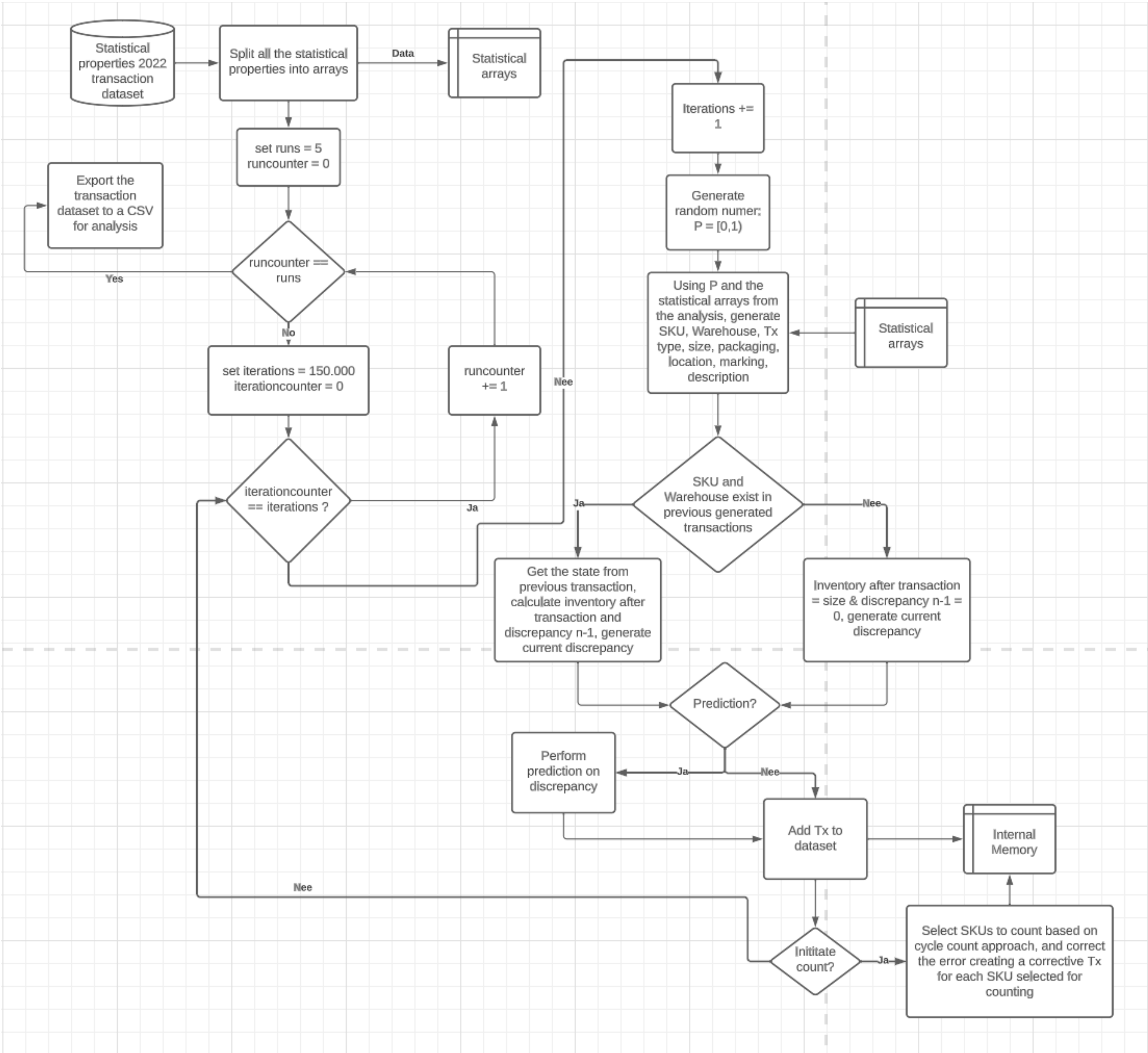


Figure 24 Flowchart warehouse simulation

5 Results

This report chapter discusses the results of the experiments performed on the designed framework. The experiments aim to answer the fifth and sixth research questions and determine which alternatives in the select models are best suited for implementation at Euroma. The prediction models aim to predict the discrepancy in the inventory record. This prediction is then used in a cycle count model to make an "informed" decision on what SKUs to include in a count.

5.1 Prediction Experiment results

5.1.1 Encoding

As stated in section 4.2.2.2, the dataset with transaction data contains many categorical features. Regression models construct a function that can only take continuous values. Using the categorical features therefore requires a form of encoding. In the previous chapter, two approaches are discussed, namely, one-hot encoding and integer encoding.

Experiments were performed using both encoding methods. Each model is run for N = 10 iterations and K = 5-fold cross-validation. All experiments were performed using the same dataset, with the key difference being the encoding approach for the categorical features. Figure 25 shows the difference in the computational time and the model performance; for the exact values view Appendix D.

The fit time and score presented in the table are averages over the 10 runs with randomly selected sets of hyperparameters. Each run consists of 5 folds used for cross-validation. The R^2-score and MSE are the results of predicting the test set. The scorer used for refitting the models is the R^2-score. After fitting the model using the randomized search algorithm, the best performing estimator is tested on the separated test dataset.

Figure 25 Performance comparison of encoding approach

An issue experienced with applying a one-hot encoded dataset and performing a model fit is the size of the dataset. One-hot encoding creates a dichotomous column for every value a categorical feature can take, see section 4.2.2.2. Applying this to the available dataset increases the size fivefold. The effects of the additional columns are also immediately apparent when fitting the lasso models. As lasso performs feature selection during a fit, the time required is significantly higher with the increased dataset size, leading to a single fold in the cross-validation taking over an hour to compute. As the performance using integer encoding is better in all aspects, this approach will be used from this point forward.

These initial results also conclude that ridge and Bayesian ridge regression underperform compared to the other models. The MSE is low compared to the other models, and the R2 score is close to 1, meaning a very accurate prediction. Even though the KPIs indicate a good performance, the models show a tendency to overfit the data. Looking at the performance between the folds in the cross-validation the R2-score attained on the test set starts at very high and then decreases, indicating an overfit on the data.

5.1.2 Feature selection

Feature selection constitutes a step in machine learning where the dataset is pruned from features proven not to improve the model performance. There are several approaches to feature selection, and four of these will be applied to find the set of features that best explains the desired response variable. All four approaches were performed on the full dataset to prevent introducing a bias. The four approaches that were tested are:

Table 10 Feature selection approaches

Approach	Benefit
Lasso regression	Performs model selection during data fit
Feedforward feature selection	Adds features to the model based on best performance
Backward feature selection	Removed features from the model based on their level of significance
Correlation heatmap	Graphical representation of the correlation between features

Lasso regression features

This approach to feature selection is similar to that described in the section above. The main difference is that Lasso regression has the ability to set the importance of a feature to zero, effectively performing feature selection during the model fit. The feature importance according to the Lasso regression model is shown in Figure 26.

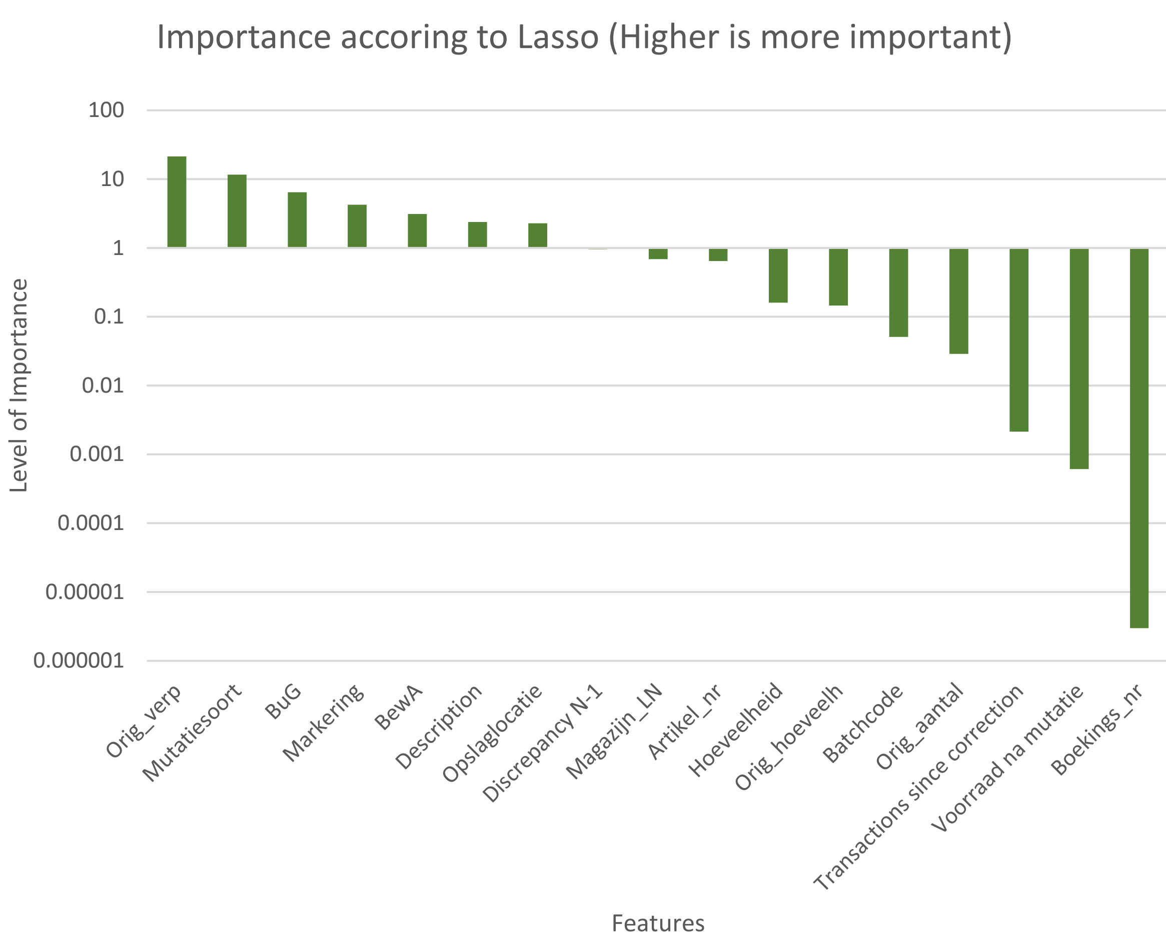


Figure 26 Feature importance according to Lasso Regression Model

Feed forward feature selection

The sklearn package in python provides the SelectKBest function which does the selection of the best N features. Applying this software library with an incrementing number for N shows the sequence in which the features are added based on their correlation. The order in which the features were added is shown in Table 11. A thing of note is that this approach to feature selection requires a single datatype. For this reason categorical features were converted to a numerical substitute, i.e., the warehouses EZ, EZDS, EZSI and EZMV are encoded as 1,2,3 and 4. Features with only zero values were removed.

Table 11 Forward feature selection

Feature	Added as	Feature	Added as
Discrepancy N-1	1st	Voorraad na mutatie	10th
Transactions since correction	2nd	BewA	11th
Magazijn_LN	3rd	Description	12th
Boekings_nr	4th	Markering	13th
Orig_verp	5th	Artikel_nr	14th
Class	6th	Orig_hoeveelh	15th
Batchcode	7th	Opslaglocatie	16th
Mutatiesoort	8th	Orig_aantal	17th
BuG	9th	Hoeveelheid	18th

Backward feature elimination

The “statsmodel” library available in for python allows for a linear model to be fit to the data. From this fit the p-value of each of the included features can be extracted. Backward feature elimination applies this by removing features with a p-value above a chosen level of significance. Typically, 0.05 is chosen as the boundary level of significance. Applying this to the available data shows that with a significance level of 0.05, only the feature describing the class of the item is removed.

Correlation heatmap

A correlation matrix is constructed using the associations function in the dython package for python 3. This heatmap, Figure 27, summarizes the correlation of categorical features while also evaluating continuous features; see Appendix G & H for a larger version.

The interest lies in the last row or column. This depicts the correlation between exploratory features and the desired response variable or vice versa. The correlation heatmap shows a high correlation between the current discrepancy and the discrepancy in previous periods. This is expected as the discrepancy is a continuous variable, highly dependent on its previous state.

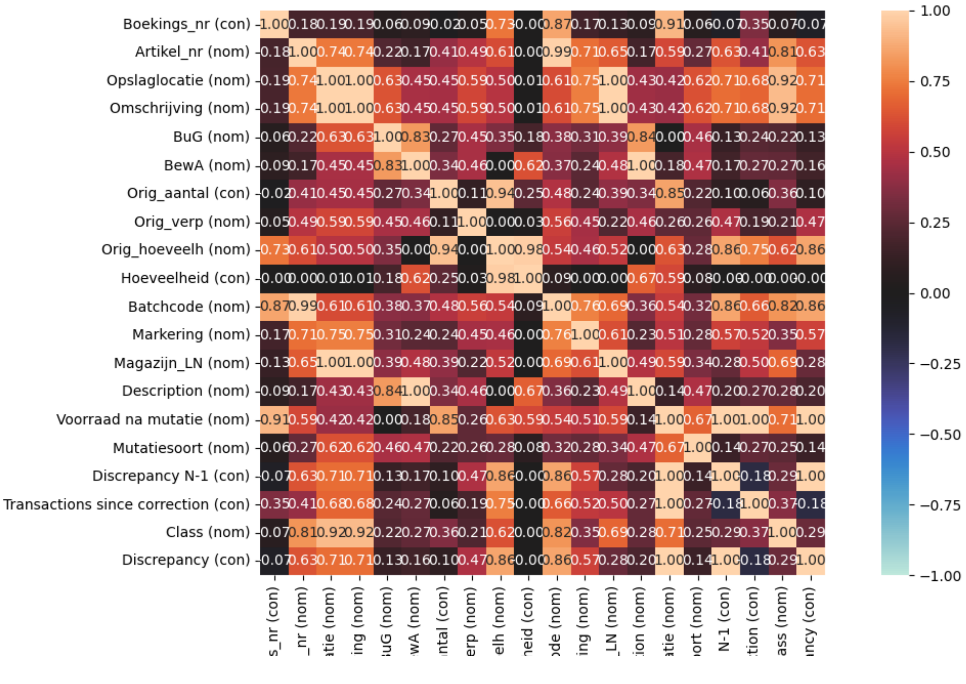


Figure 27 Dython correlation heatmap

Due to limitations in the dython package, some of the numerical and categorical features are not identified correctly, and correlation is not found correctly. For this reason, the correlation of the inventory position after a transaction is not copied from the heatmap shown in Appendix G & H. In the Appendix, two correlation heatmaps are shown; inventory after transaction is identified as a categorical feature while it is continuous, while in H, the SKU number is identified as a continuous feature while it is a category. Adjusting the code to properly represent the data discarded after several failed attempts. The values in the two matrixes are combined based on what the type would have been. This approach fails to capture that the correlation between features changes when the type differs.

Conclusion on feature selection

A comparison of the deduced feature importance is provided in Figure 28. The level of importance is not provided as a fractional value when using lasso regression or, SelectBestK, as it is in the dython heatmap. The importance values have been changed to be represented as fractional values. In the case of the lasso feature importance values, everything is scaled with the highest resulting value being 1. For the SelectBestK approach only a sequence is provided, the fractional value is found by dividing 1 by the position in the sequence that the features were added. Again, the most important feature is represented by 1.

From Figure 28 it is concluded that the previous discrepancy is an important feature. This was to be expected as the discrepancy builds on the previous value until it is corrected. The other features have varying correlations according to the different approaches. Contrary to what would be expected the different approaches do not have a general consensus behind the importance of the different features, some similarities are visible.

While the batch code shows correlation with the response variable under the dython and Lasso approach, this is a value that is non-recurring. Training a model using this is counter intuitive and for this reason the feature is omitted. Aside from the batch code some features show very little correlation for any of the three approaches. This is motivation to also remove these features.

Using the information from the four approaches there are no clear features that are not correlated with the response variable. The resulting importance's are all scaled so that the maximum value is one. In the case of the backward feature selection a significance level of 0.05 was used.

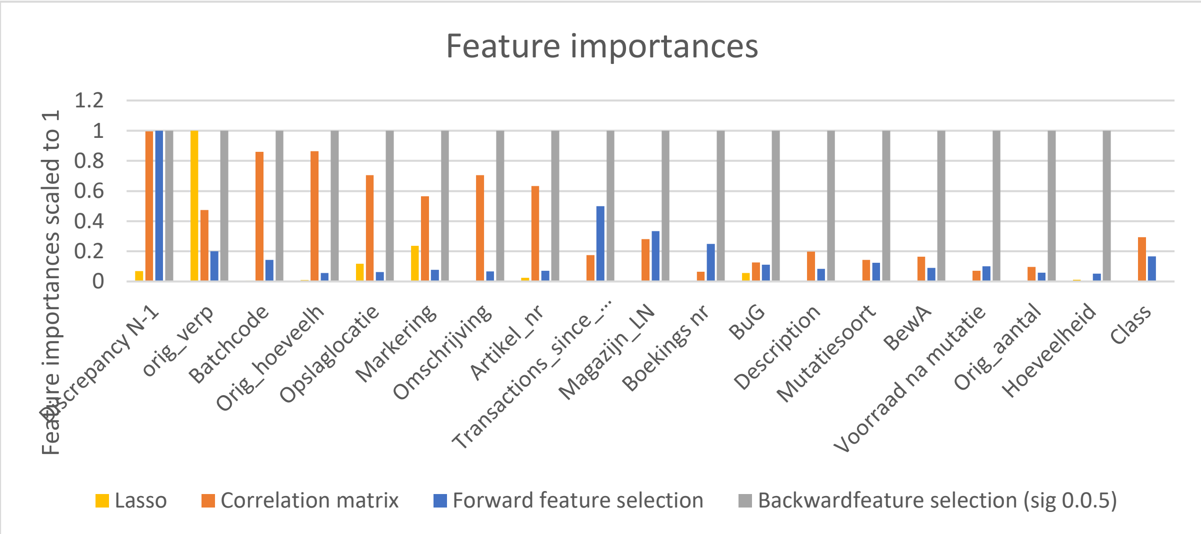


Figure 28 Feature importance of the three approaches

Ensuring that useful features are note removed is realized by reviewing the R^2 score of the model after removing a feature. The features are removed from the model in the reverse sequence shown in Figure 28. Starting with "class", after removing a feature the model is trained again and the R^2-score is again calculated. This is repeated until all features are removed. The model used for this experiment is the Random Forest Regression. Figure 29 shows that the R^2-score start dropping when the 10^{th} feature, the warehouse (Magazijn_LN) is dropped.

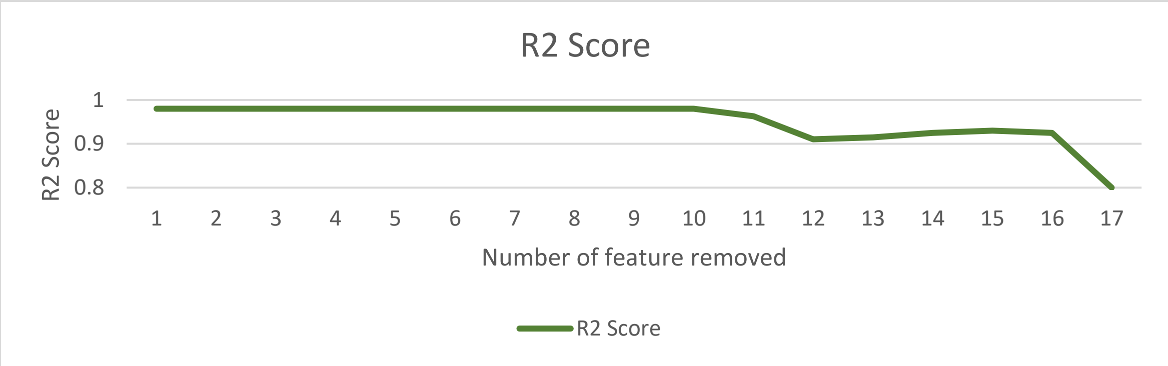


Figure 29 R2 score depending on number of remaining features

5.1.3 Hyperparameter search and cross-validation

Finding the optimal values for the hyperparameters of the different machine learning models is realized by applying the RandomizedSearchCV algorithm provided by the Scikit-learn package. Each model has an accompanying set of hyperparameters. The RandomizedSearchCV selects a set of hyperparameters at random and attempts to fit the model. For every set of hyperparameters the model is fitted using K-fold cross-validation. For each of the models the set of hyperparameter is denoted in Appendix F. The optimal values resulting from RandomizedSearchCV can also be found in Appendix F.

The models that were included in the hyperparameter search are Lasso, DecisionTree, RandomForest, XGBoost and Neural Networks. The hyperparameter search for each model was performed 50 times to ensure that a large portion of the possible permutations of the hyperparameter set are covered.

Table 12 Regression model experimental results

Regression Model	**Fit time (sec) (lower is better)**	**Best R^2-score train set (higher is better)**	**R^2-score test set (higher is better)**	**MSE test set (kg) (lower is better)**
Lasso Regression	395.2	0.99	0.99	203332.3
Decision Tree	1.5	0.79	0.92	1549620.9
Random Forest	8.1	0.99	0.96	661237.1
Extreme Gradient Boost	51.7	0.97	0.96	798846.3
Neural Network	15.3	0.92	0.89	1423168.8

Learning curves best models

As discussed in Section 4.2.3 the concept of the learning curve has been introduced. Learning Curve plots visually show the effect of increasing the number of iterations in training on the performance of the model. Below, in Figures 30-36**Error! Reference source not found.**, the learning curve plots for all r egression models are provided. The plots for ridge, Bayesian ridge, lasso, and the decision tree model show a negative impact on the performance when increasing the number of samples. This indicates that the model is overfitting on the data and “learning” the response variable as opposed to predicting it. The RandomForest, XGBoost, and Neural Network do not show this same characteristic. What is apparent is loss in accuracy on the training dataset displayed by the Neural Network. This behavior in a learning curve is also indicative of overfitting. Comparing the ensemble models to the Neural Network model shows the Neural Network model less affected by the variability in the data. The Neural Network shows this variability at the beginning of the training.

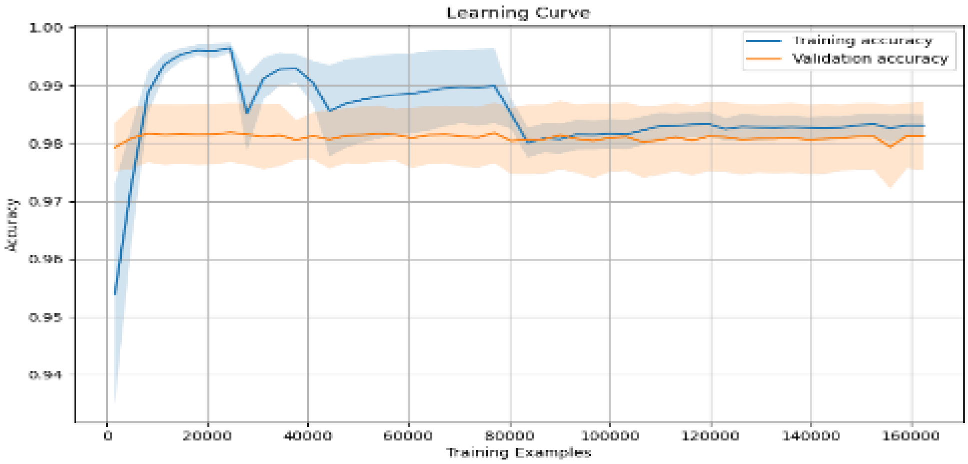


Figure 30 Learning Curve Ridge Regression

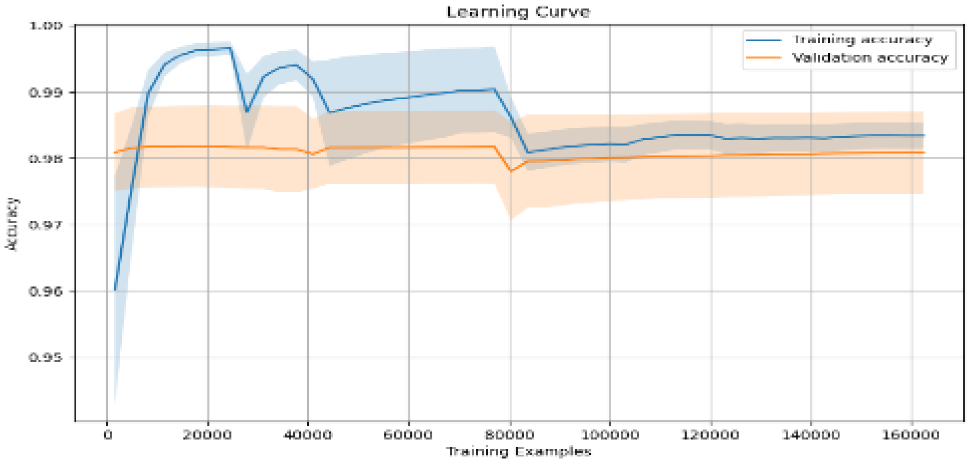


Figure 31 Learning Curve Bayesian Ridge Regression

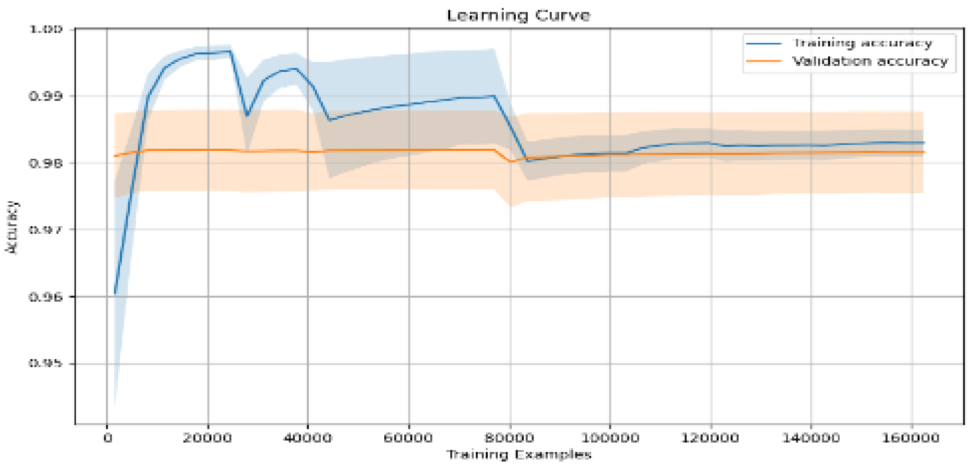


Figure 32 Learning Curve Lasso Regression

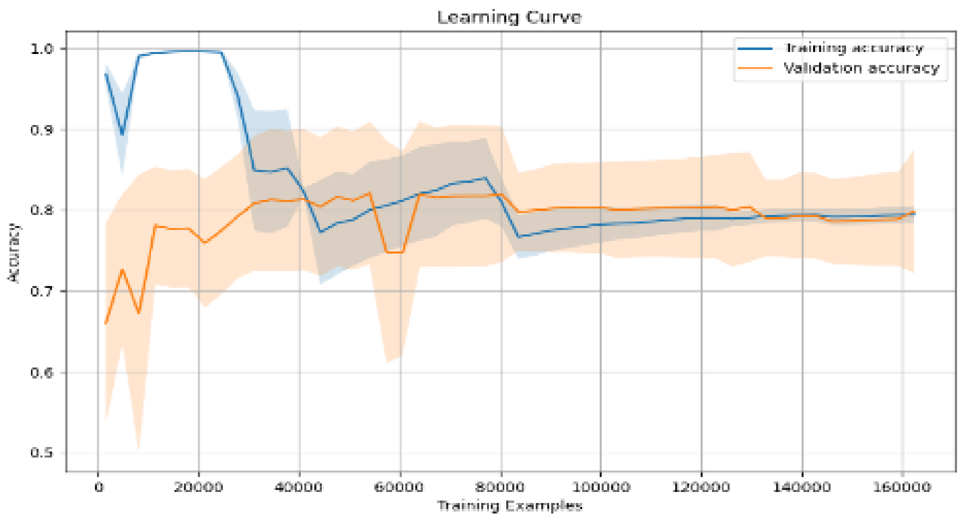


Figure 33 Learning Curve Decision Tree Regression

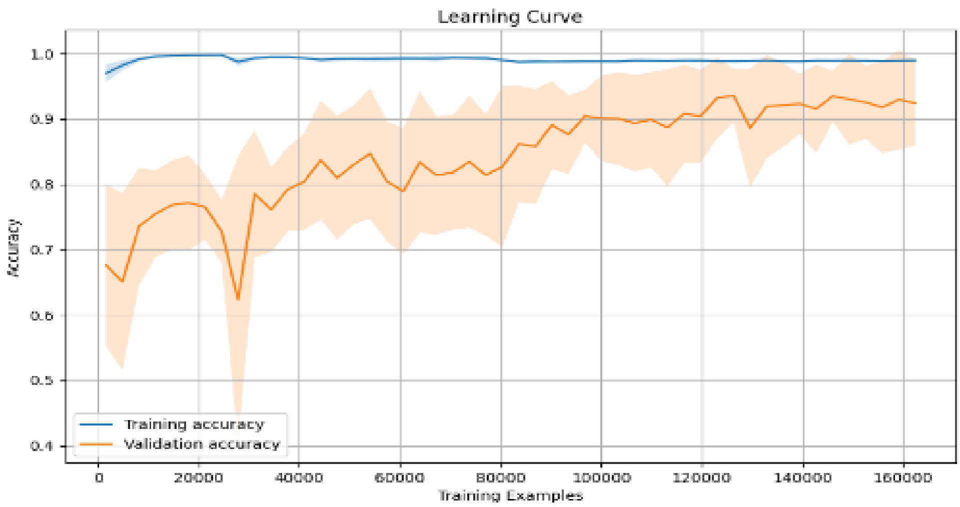


Figure 34 Learning curve Random Forest Regression

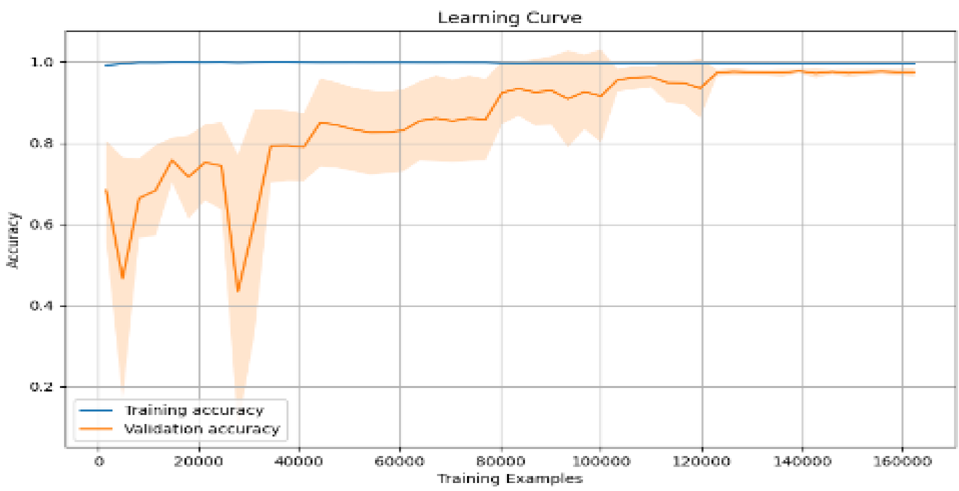


Figure 35 Learning Curve XGBoost Regression

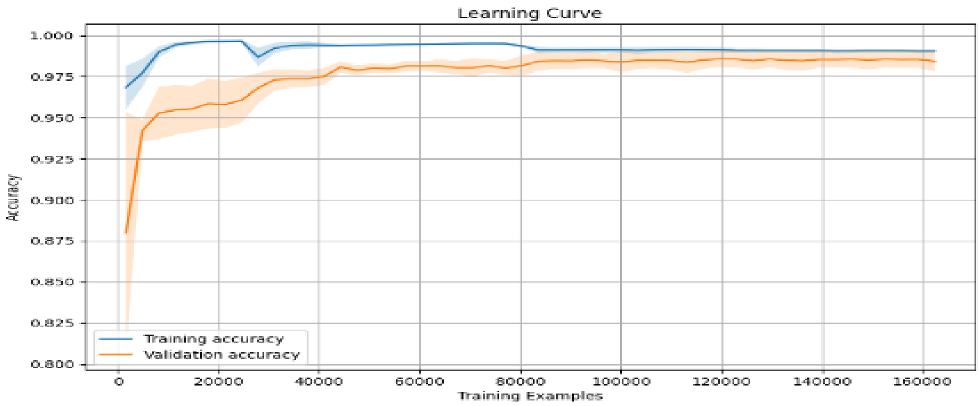


Figure 36 Learning Curve Neural Network

5.1.4 Conclusion prediction model experiments

The experiments for the prediction model have provided a good insight into the usability of the different models. Section 5.1.1 concluded that the encoding approach mainly impacted the time required for a model to fit to the data. This was expected as the dataset increases significantly when applying one-hot encoding compared to integer, or dummy, encoding. The performance of the prediction did not immediately suffer from the added features that result from the former encoding approach. However, as ideally processing time is reduced as much as possible integer encoding is the better option.

When applying and comparing the different feature selection approaches it was concluded that there is no general consensus between the four methods. Most of the available features seem to have some degree of correlation, or importance, according to one of the methods. This is most likely due to the varying estimators being used in the four methods. However, reviewing the R^2 plot created using the Random Forest Regression, Figure 29, the conclusion is drawn that the following features can be removed from the dataset: Class, Size, Original amount, Inventory after transaction, type, description, BuG, and booking number.

From the learning curves it is concluded that the ensemble methods and the neural network have a good ability to fit to the data after about 80.000 transactions. This in combination with the speed and performance of the R2 and MSE have led to the decision of using the Random Forest Regression model for the cycle counting experiments.

5.2 Cycle counting experiment results

This section will elaborate on the experiments performed using the different cycle count approaches. A comparison will be made between the performance of the selected approaches on the situation at Euroma. The performance of the cycle count models will be evaluated in their ability to improve the accuracy in the inventory. Aside from the traditional approaches of these cycle counting models, the models will be tested with the use of the prediction model output. The resulting accuracy and or improvement will be evaluated. Initial experiments were conducted with a count every single day as lowering the count frequency would decrease the performance.

This section of the report will be used to answer the 6th question of this research:

"Which cycle count model performs best on the available data?"

5.2.1 Experiment results

Below the results of the cycle count experiments are presented. An explanation for the perceived behavior of the approach and the benefits and drawbacks. The performance of each of the models will be provided. First random cycle counting is discussed, this will be used to make some comparisons when discussing the other approaches. A comparison between the approaches is done in section 5.2.2. Followed by a conclusion on the cycle count experiments in section 5.2.3.

Random cycle count

Random cycle counting was introduced to serve as a baseline. The model show what would happen if the decision on what to count was not supported by the data, it is random. The random cycle count approach is also the least restricted approach that is experimented with.

Due to the limited restrictions the model is relatively free in the decision on what to count. Partly due to the selected settings the distribution of the counts per SKU and warehouse is very uniform, averaging about 30 counts over the full simulated period. This is to be expected when random selecting from a population.

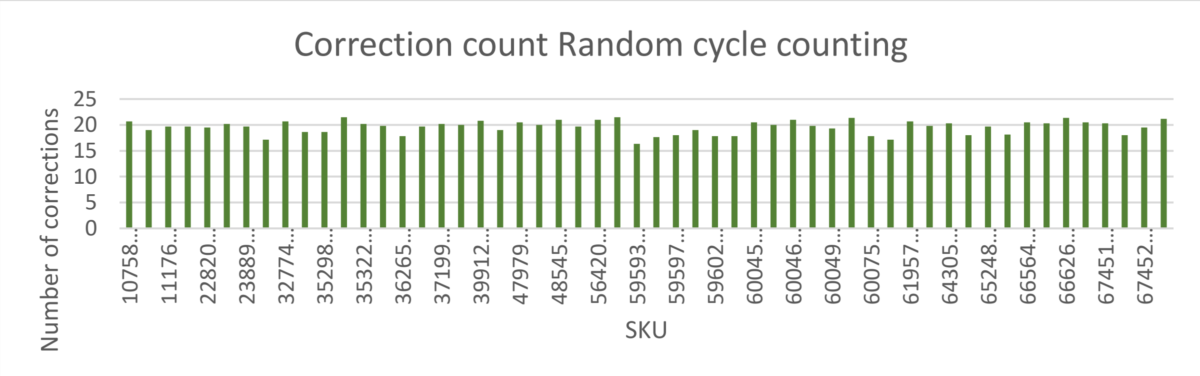


Figure 37 Number of corrections per SKU under random counting

As the decision to count is not based on anything the expectation is also that the model has a number of counts where no discrepancy is registered. This is also the case in the performed simulation. Figure 38 shows the distribution of the zero corrections per SKU and location combination, the Y-axis represent the number of counts where zero discrepancy was encountered. The average number of zero corrections over all SKUs, over 5 runs is 106.6, which constitutes 8.4% of all counts.

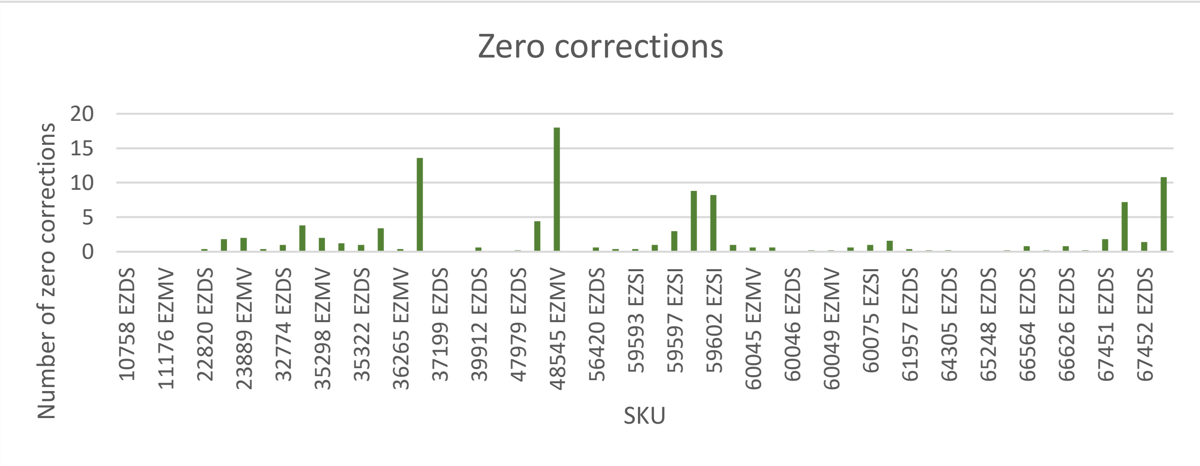


Figure 38 Zero corrections under random cycle counting, lower is better

The OPIRA averaged to 19.31% over all 5 runs of the simulation. Figure 39 shows the perfect inventory record accuracy averaged over 5 runs per SKU and location combination. The Y-axis represents the percentage of time where the SKUs did not have any discrepancy in the recorded inventory.

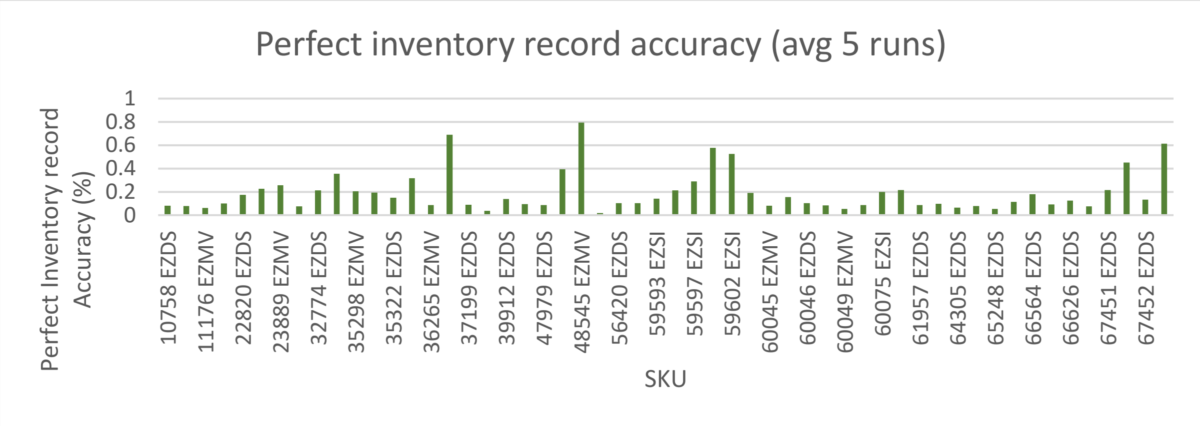


Figure 39 Perfect inventory record accuracy per SKU and warehouse, higher is better

As expected, the random cycle counting approach has a very uniform distribution when regarding what to include in a count. The level of zero corrections is relatively high, and the perfect inventory record accuracy is not.

ABC cycle count

As stated in the literature, ABC cycle counting is aimed at dividing the time spent counting an SKU based on the contribution to the annual dollar value of the inventory. The model is provided with a predefined separation of the SKUs in the three available classes. Each class has set number of counts that can be performed on an SKU of that set in the current cycle.

The model's objective is to select the maximum number of SKUs. This objective is constrained by the maximum number of SKUs that can be counted per count and a time constraint limiting the time that can be spend on counting. The model is not overly constraint and there is no benefit or cost in choosing the same SKUs every time a count is initiated. The result is that the same SKUs are counted until they have reached the allowed number of counts in the cycle. This was resolved by improving the objective function to also take into account the number of remaining counts in the cycle, prioritizing the SKUs that have not been counted. This did improve the variation in SKUs counted but did not result in every SKU being counted at least once. The total number of items counted under the ABC cycle counting is halved compared to random cycle counting.

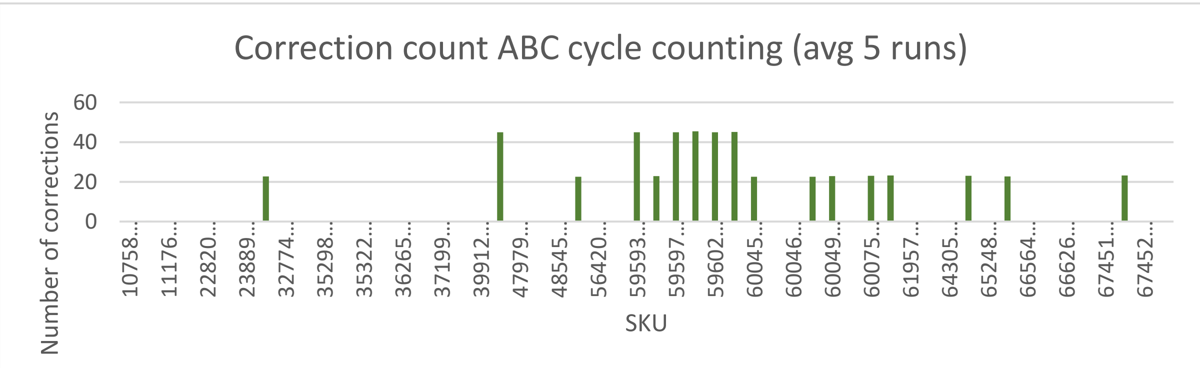


Figure 40 Average count over 5 simulation runs per SKU and Warehouse

Even after applying this addition to the objective the ABC approaches fails to include all SKUs in its cycle. Another consequence of only counting a select set of SKUs results in some SKUs not being counted at all, and also some counts not finding an error and therefore not executing an actual correction. On average 126.4 counts per run did not encounter any discrepancies. Which is approximately 25% of all counts.

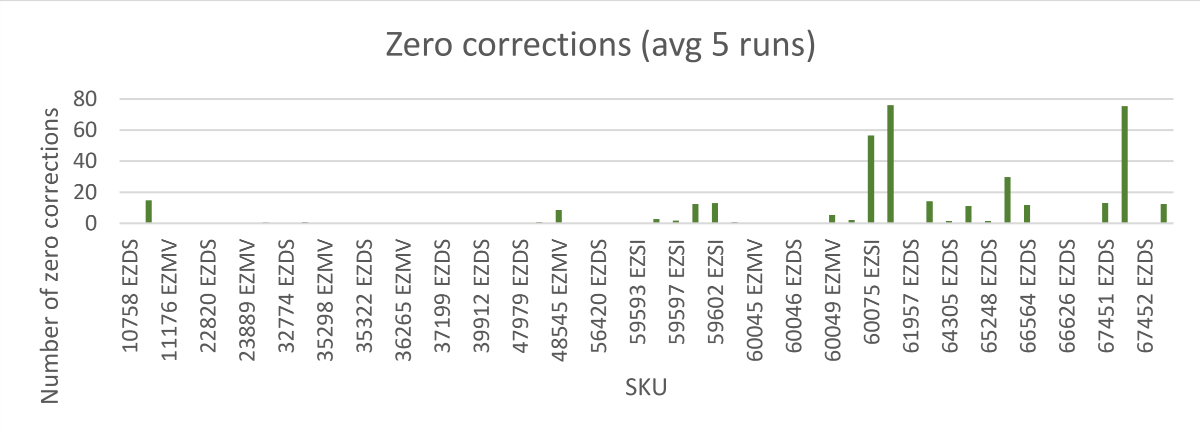


Figure 41 Average number of counts resulting in zero discrepancy, lower is better

Due to the model not being varied in the decision what to include in the count the accuracy of the inventory record was low. The OPIRA averaged to 9.91% over the 5 simulation runs. Only a very slight improvement over the historic data. The perfect inventory record accuracy per SKU is depicted in Figure 42. As expected the items with higher count rates show a higher level of accuracy.

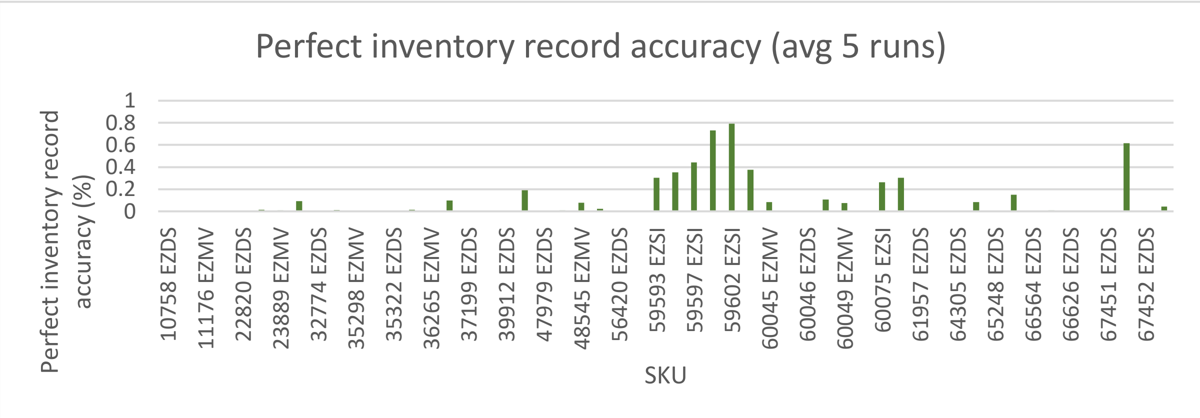


Figure 42 Perfect inventory record accuracy averaged over 5 runs, higher is better

Location cycle count

Location cycle counting, as discussed, is slightly altered from what is presented in the literature. Instead of only counting each location one per cycle the model was allowed to select one of the warehouses. The reason for this is that the locations are not directly connected to an SKU, while the warehouses are. However, only counting a warehouse once per cycle would result in excluding a very large amount of SKUs and doing nothing for periods of time.

Location based cycle counting shows a, somewhat, even distribution of counts over all the available SKUs and locations. The main limitation on the model in this approach was the time constraint and the number of counts that can be performed on an SKU each cycle.

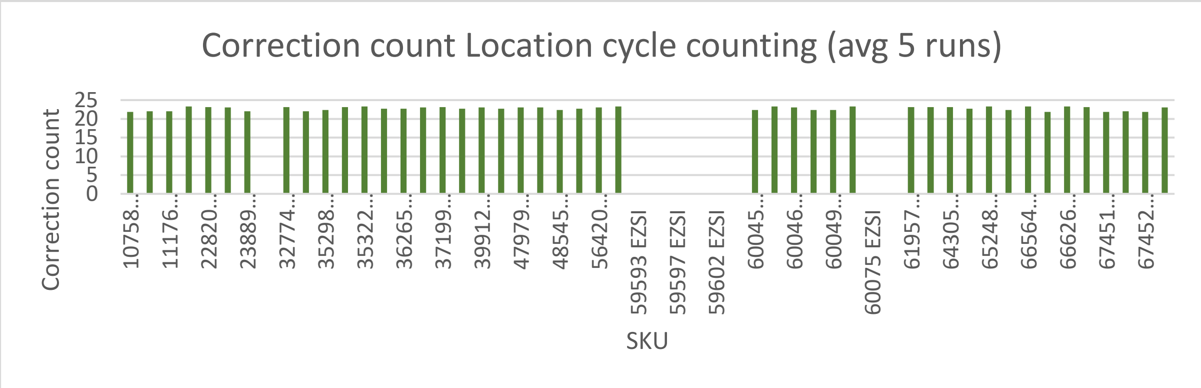


Figure 43 Distribution of counts per SKU averaged over 5 runs

The number of corrections not finding any discrepancy in the record is shown in Figure 44. The total number of zero corrections averaged over the 5 simulation runs is 77.2. This is significant improvement, close to 30%, over the random cycle counting.

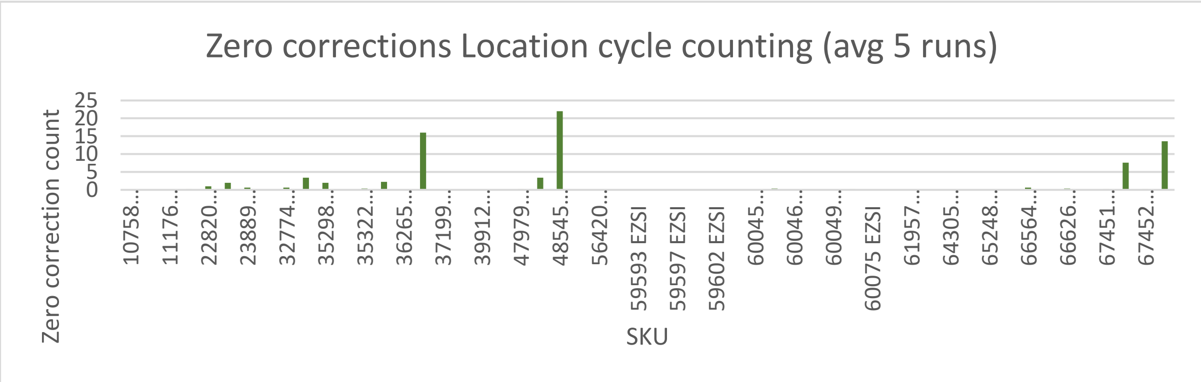


Figure 44 Average number of corrections not encountering an error, lower is better

The perfect inventory record accuracy average over 5 simulation runs is presented in Figure 45. Comparing the result with the random cycle counting a very similar distribution is observed. The OPIRA averaged to 17.03%. Showing a slight decrease compared to the random approach, however this was achieved with a reduced number of counts and significantly less counts that turned out to be fruitless.

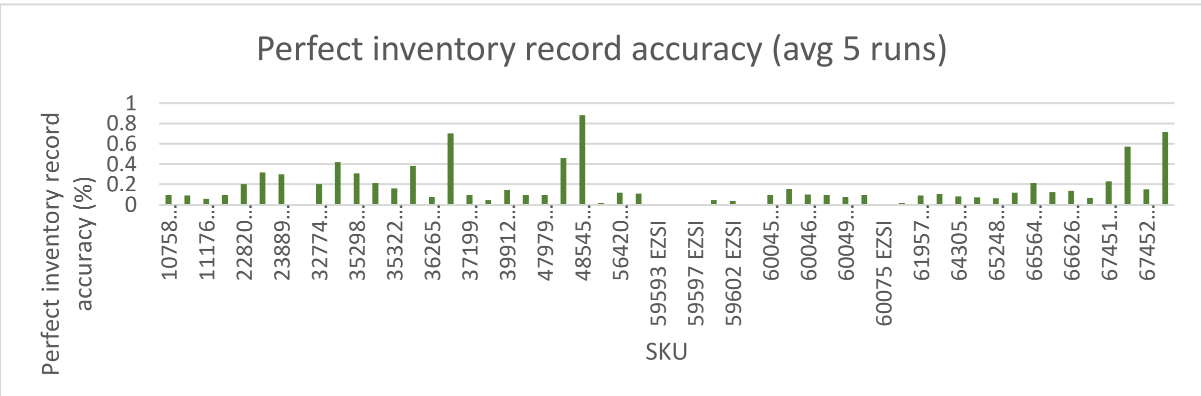


Figure 45 Average perfect inventory record accuracy over 5 runs, higher is better

Transaction cycle count

Transaction based cycle counting is based in the idea that articles that are used more are more prone to error in the inventory record. Following that logic, counting items that are used more would resolve more discrepancies. The objective function of the model is therefore also aimed at maximizing the sum of the transactions since the last correction for the selected SKUs. Figure 46 shows the distribution of the counts, performed per SKU and warehouse, being relatively uniform.

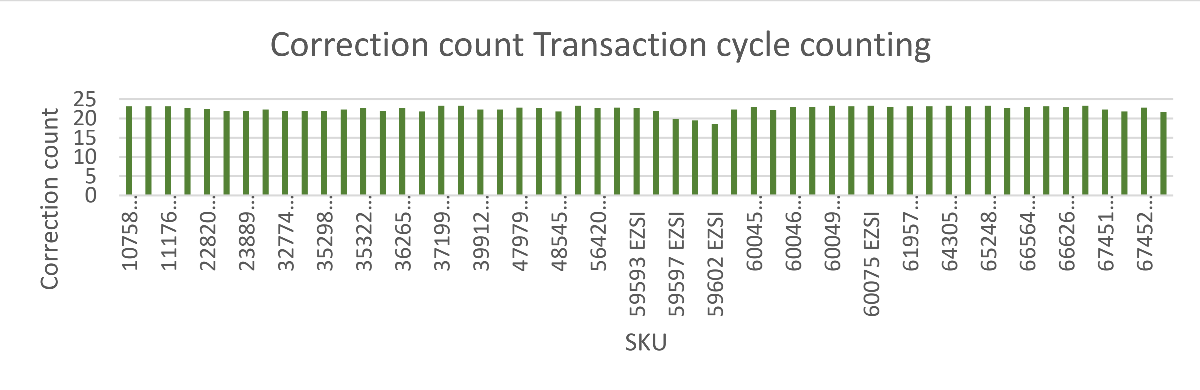


Figure 46 Correction count per SKU and location under Transaction based cycle counting

The theory that targeting SKUs with a high usage is effective is also evident when looking at the number of corrections that did not result in finding a discrepancy. The total number of counts that showed no difference in the record and the simulated level is lower compared to random counting, see Figure 47. The same distribution does show up when comparing the output of the transaction counting and random counting. This is likely due to the statistical properties of the simulation, see Section 2.7.

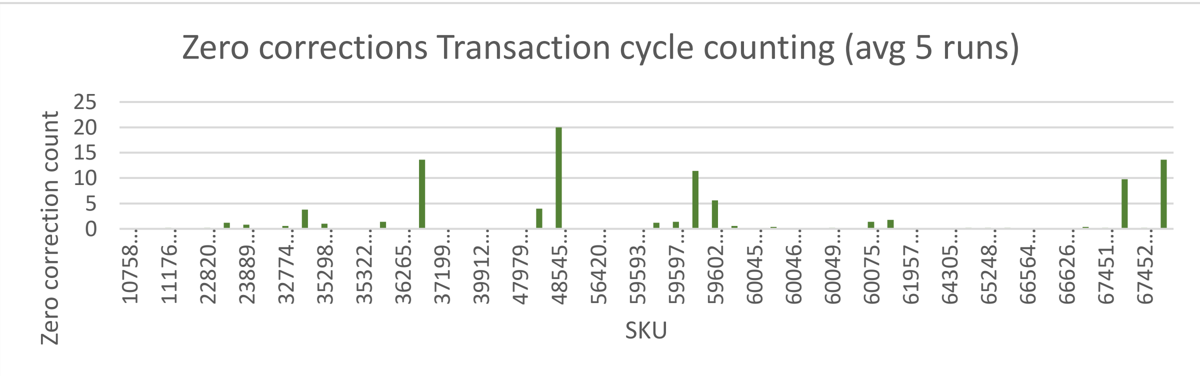


Figure 47 Distribution of zero corrections under transaction cycle counting

When looking at the perfect inventory record accuracy there is still a lot of discrepancy in the record. The OPIRA averaged over the 5 runs of the simulation comes down to 22.15%. The perfect inventory record accuracy per SKU and warehouse is shown in Figure 48.

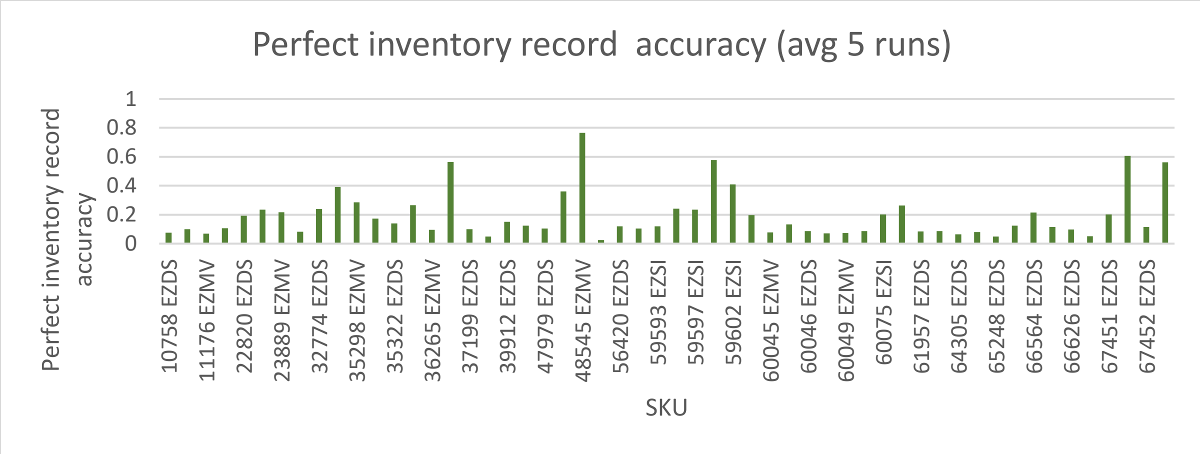


Figure 48 Perfect inventory record accuracy per SKU and warehouse under transaction cycle counting

Random Forest Regression cycle count

The predictive cycle count approach is similar to the random cycle count approach in regard to the constraints that are enforced. The key difference between the two is that the prediction model maximizes the predicted discrepancy for its count selection. The idea behind it is that the decision to count an SKU is an "informed" decision. For the prediction model a RandomForestRegerssion estimator that was trained on the available historical data from 2022 was used. As the simulated data closely represents the historical data.

As is evident from the count distribution over the available SKUs and warehouses, the selection on what to include in a count is less uniform compared to the random approach. This is expected behavior as the model will aim its efforts towards SKUs that it predicts to have a high discrepancy, which should correlate with a high discrepancy in the past.

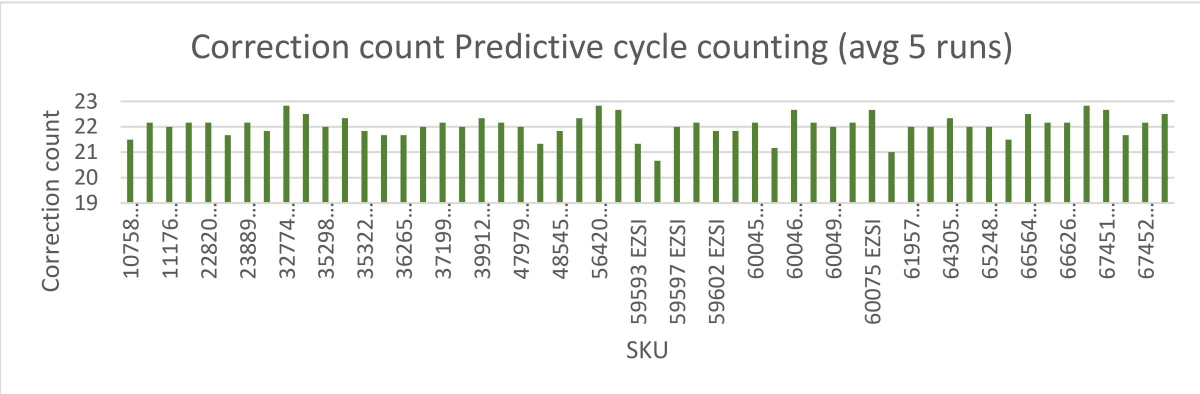


Figure 49 Count distribution under predictive cycle counting

While the prediction model is aimed at maximizing the resolved predicted discrepancy the number of zero corrections is still relatively high, averaging around 113 counts. This indicates that the model predicts high discrepancy for SKUs of which the record is in fact accurate.

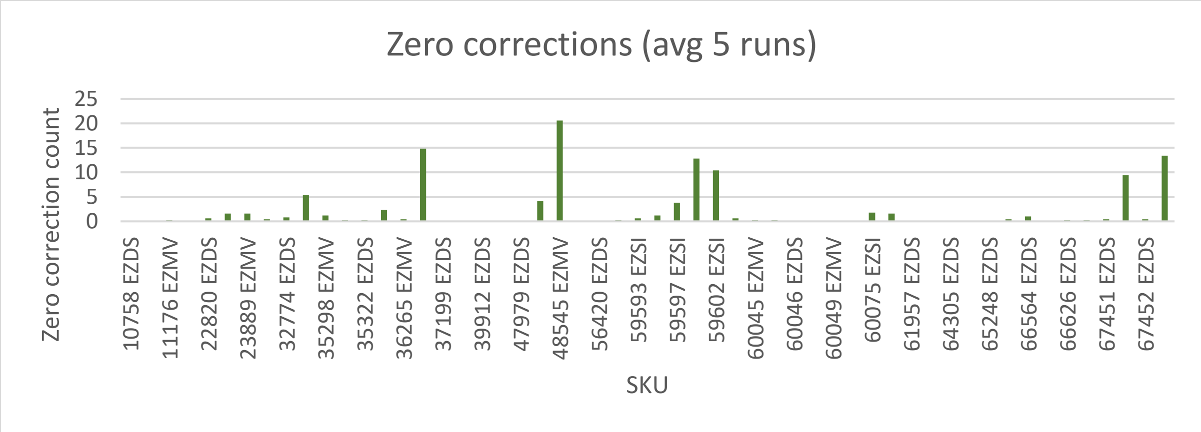


Figure 50 Distribution of zero corrections under predictive cycle counting

The PIRA averaged over 5 runs is shown in Figure 51. The distribution of the PIRA is similar to that encountered under random cycle counting. However, the OPIRA increased to 21.9%. This indicates that the model selected more inaccurate SKUs compared to the random approach.

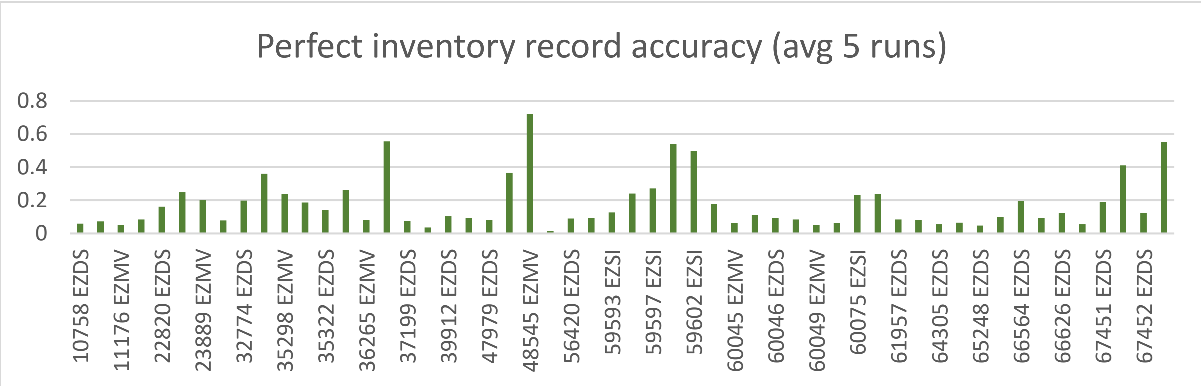


Figure 51 Prefect inventory record accuracy under predictive cycle counting

Real Data

For the sake of comparison the same graphs provided for the simulated cycle counting approaches will be provided for the approach currently applied. This approach can best be described as reactive cycle counting.

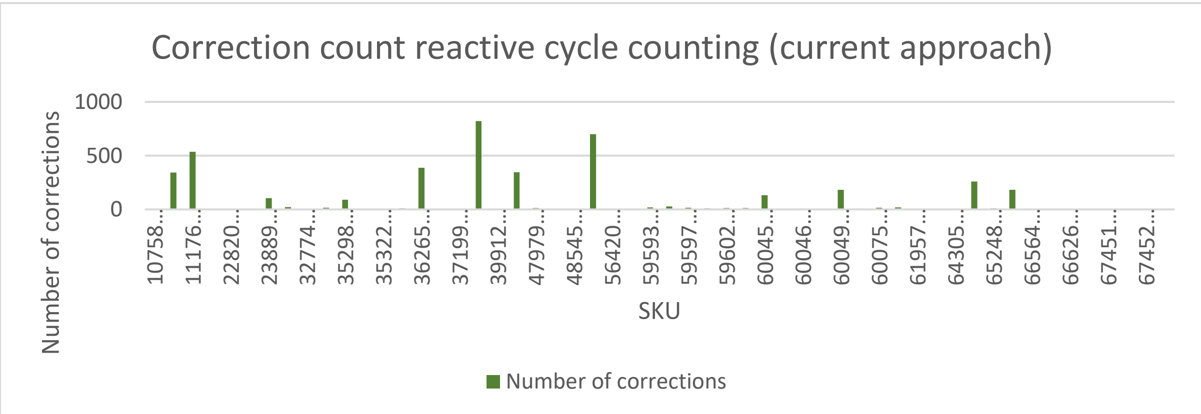


Figure 52 Correction count for reactive cycle counting

The reactive nature of the cycle counting approach is immediately apparent when comparing the number zero corrections. Over half a year of transaction, only two corrections were noted that did not change the inventory record.

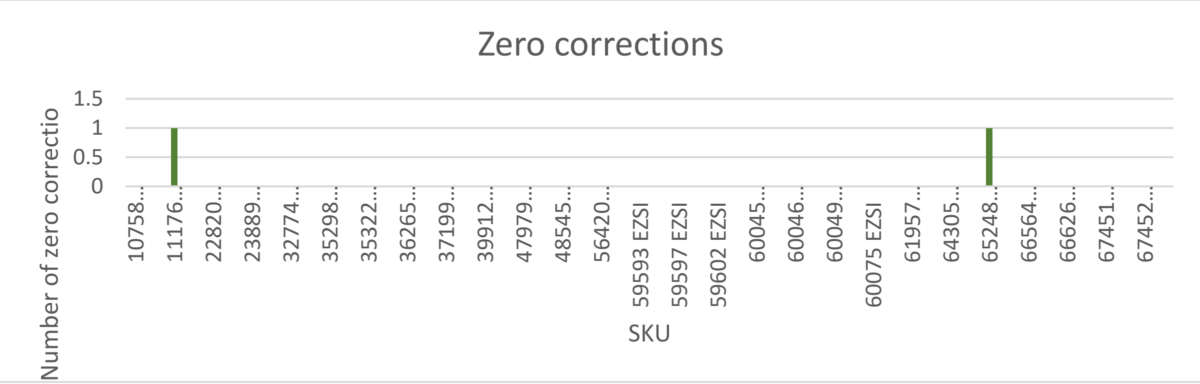


Figure 53 Zero corrections on the real data

The reactive approach shows a very poor performance with respect to the PIRA KPI. This is because an item can have an error in the record for a very long time before it becomes an actual problem, triggering a reaction.

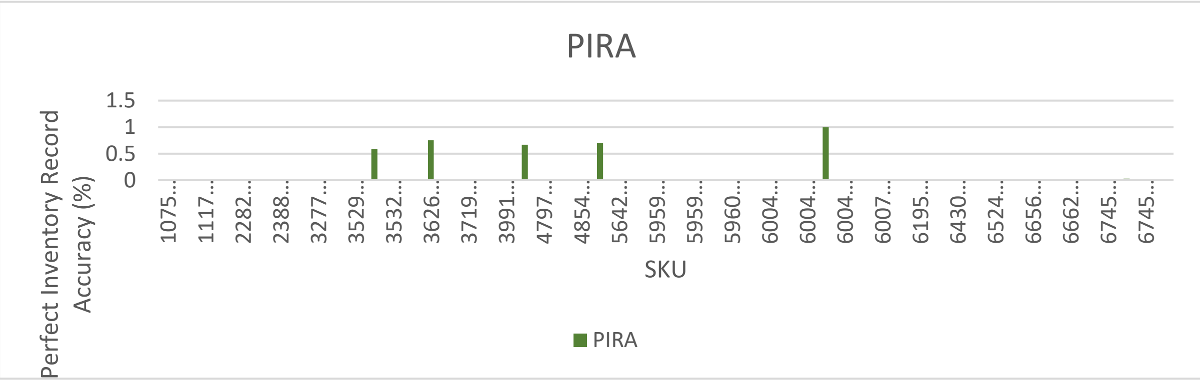


Figure 54 Perfect inventory record accuracy for the real data

5.2.2 Conclusion cycle counting

The performance of the predictive and traditional cycle counting approach will need to be compared to evaluate the benefit, if any, of including the prediction model. The comparison will be based on the perfect inventory record accuracy, see (1) section 3.1.2 . This is a measurement that can only be used in offline testing, or after a complete count, as the information on the record accuracy is required. The reason for using the perfect inventory record accuracy does not allow any deviation and is a harsh scorer. However, implementing a 5% deviation, as also suggested in the literature, does not provide any beneficial insights in simulation.

Comparing the different approaches and the real time data it is evident that a more targeted cycle count approach improves the overall perfect inventory record accuracy. Where the real time data showed an accuracy of around 7.68% even random counting improved this to around 18%. ABC cycle counting performed relatively poorly, however, this could also be due to the settings of the model being too restrictive for this approach. Transaction based and Random Forest Regression based cycle counting managed a comparable performance when it comes to the overall perfect inventory record accuracy, both being around 22%. Location based cycle counting shows a OPIRA close to that of the random cycle counting, strangely the location based approach completely disregards the EZSI warehouse.

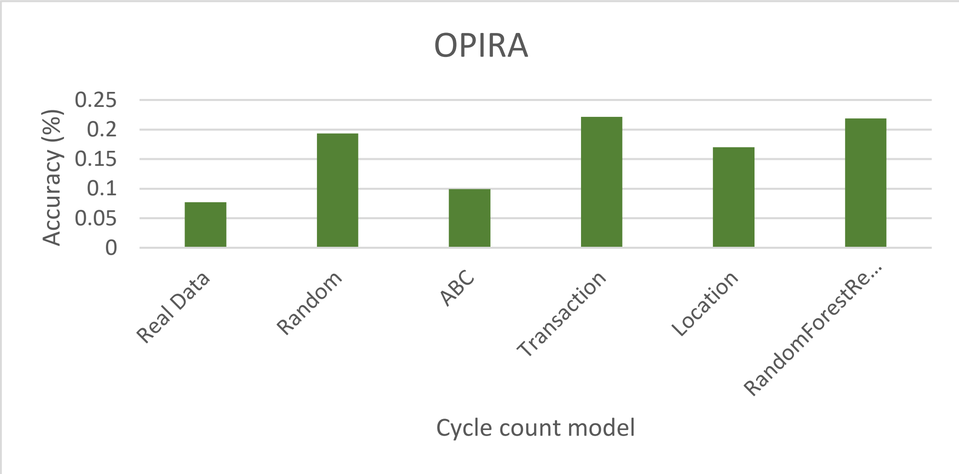


Figure 55 Summary of Overall Perfect Inventory Record Accuracy3

Figure 56 provides an overview of the average number of counts not encountering any discrepancy. Comparing the historical data to the results from the simulation, shows that the introduction of a structured cycle counting approach would reduce the level of effort not finding errors. Comparing the cycle count approaches, shows that location based counting has a higher tendency of finding discrepancies in the inventory record. ABC cycle counting performs worst of all models, this is likely due to the model having a very limited selection of SKUs that it includes in the counts. As A items can be counted more often than B or C items, the model tends to select these SKUs more often. The Random Forest Regression based counting also shows, comparatively, poor performance in this regard. This would indicate that the model is predicting errors where there are none.

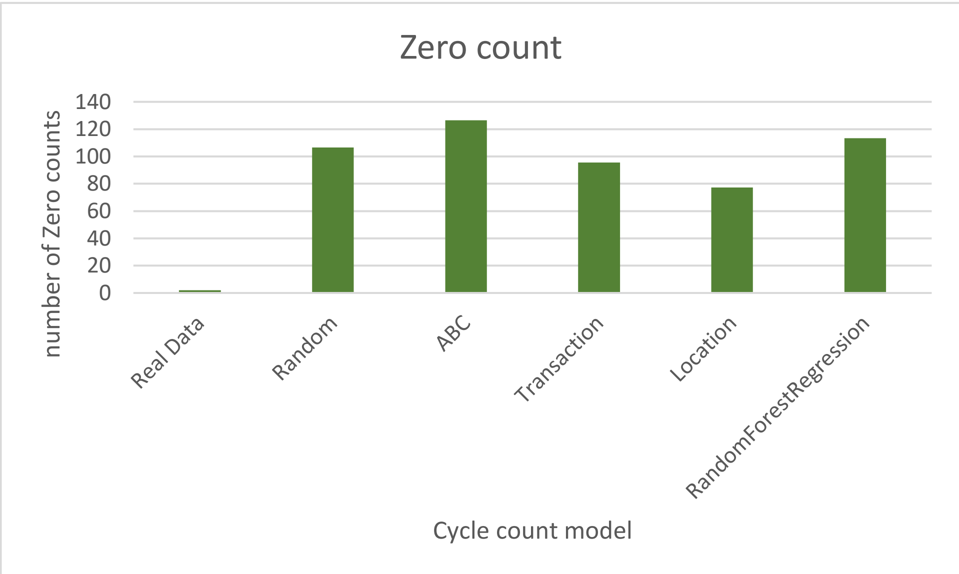


Figure 56 Summary of zero counts

Introducing cycle counting, regardless of the chosen approach, shows an improvement of the inventory record. While this increase is marginal in the case of ABC, an increase of at least 10% is achieved in any of the other approaches.

The efficiency of the performed counts also shows an improvement compared to the real data. The cycle counting approaches also select SKUs of which the records are still correct. However, the number of counts, as well as those not returning an error is reduced while the accuracy of the records increases.

The performance of the predictive cycle counting approach on the simulated data is not a convincing benefit over more established approaches. The simulation even shows that the predictive approach has a tendency of counting more SKUs where no error is present. This seems to affect SKUs in the EZMV warehouse the most.

6 Conclusion

6.1 Conclusion

This chapter will provide a conclusion on the research performed and presented in this thesis. The chapter is divided in three segments. Section 6.1 will discuss the prediction model. The results from the experiments will be discussed and the resulting decision will be explained. Section 6.2 covers the cycle counting models and experiments. The final section, 6.3 will discuss the applicability and possible improvements of combining the prediction model with the cycle count model.

In the experimentation and testing with the prediction model it quickly became evident that the simple regression models were not up to the task. When reviewing the performance of Bayesian Ridge and Ridge regression it was evident that the models were unable to learn from the data. A guess was made that typically closely resembled the feature "discrepancy n-1".

The decision tree regressor did show more promising performance compared to the previously mentioned models, however, as it is still a simple tree based regression model had difficulty coping with the large dataset.

The ensemble methods that were experimented with, *XGBoost* and *RandomForestRegression*, are also tree based regression algorithms. Both models performed comparable on the data, this is most likely because the models are very similar.

The final prediction approach that was tested was the neural network. Neural networks have gained a lot of popularity as they are very versatile in their application. The neural network showed the comparable performance when comparing the R2 score. Comparing the MSE showed that the model was predicting with a greater error than both ensemble approaches. This in combination with the longer training time that was required supported the decision not to continue with this variant.

The final decision came down to the *RandomForestRegressor* as this model showed the most promising results regarding the selected KPIs. This is the model that was used for the prediction on the simulated data.

From the simulation runs with the cycle counting approaches it is concluded that the introduction of any structured cycle counting method increases the accuracy in the inventory. Applying structured cycle counting instead of reactive counting to resolve the reported issues is almost guaranteed to increase the accuracy of the inventory record. An interesting addition to that is the fact that the models were able to increase the OPIRA while reducing the number of counts that are performed.

Comparing the results from the simulated cycle counting approaches and the historic data shows an increase in the OPIRA of 10%, aside from the ABC cycle counting. This does not allow for the conclusion that ABC is an invalid approach. Of the other four approaches random cycle counting showed the lowest OPIRA score. This is expected as it constitutes the result of simply counting something. The difference between the random and other three approaches is limited. Only showing an increase of 1%-5%.

Prediction based counting showed performance comparable to transaction based counting. The prediction based counting had a larger number of counts that did not resolve any discrepancies. Most of these counts were performed on the EZMV warehouse. This points towards the conclusion that the model has difficulty with predicting the error in that warehouse. As transaction in EZDS and EZSI are magnitudes larger compared to EZMV it is possible that the model overfits on the larger values.

The achieved, simulated, accuracy on the inventory record does not match the goal that was set out at the start of this research. The accuracy that was attained during simulation was 22%, an increase in of ~15% compared to the 2022 data. This does fall in line with the literature as presented by Brooks & Wilson. However, is not as high as Euroma would have wanted it to be

While the goal was not achieved an improvement was realized. This does point towards the conclusion that Euroma should implement a form of cycle counting. The results also show that there is some merit in applying the prediction model in assisting the decision to count an SKU. However, basing the count decision on transaction count showed similar performance. Transaction based cycle counting is easier to implement and requires less speculation on the possible error.

6.1.1 Impact

The impact of the proposed solution is hard to quantify as the time and costs are grounded in assumptions. It was also not recorded for the real time data what the costs were for performing counts. Or the additional costs due to unexpected stockouts.

Simulation showed that by introducing structured counting the number of counts that were performed were severely reduced. This was also due to the limitations on the cycle count models. Even though the number of counts was reduced the accuracy was increased significantly. In the case of the random forest regression cycle counting model an increase of 15% or 3 times the accuracy of the real time data was achieved.

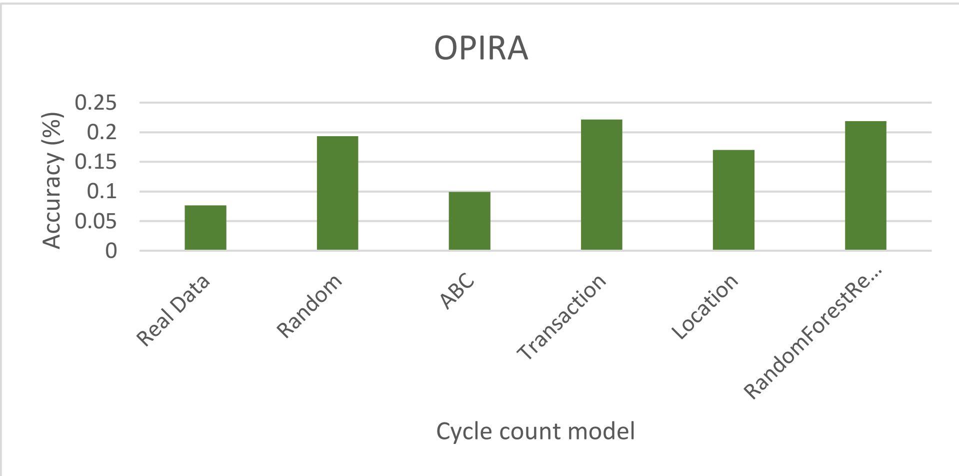


Figure 57 Summary of Overall Perfect Inventory Record Accuracy

Aside from increasing the accuracy the time spent on counting was also reduced. Even though the number of zero counts increased this shows that the time spent on counting is used more efficiently. On average the structured approaches realized a decrease of 75% compared to the real data. The hourly wage of the person counting the items is unknown but the cost reduction is obviously significant. A conservative salary would be around 15 euros, meaning a reduction of almost €10.000,- is realized. The increase OPIRA also indicates that the chances of stockout or backorders is reduced as a better understanding of the inventory position is gained. Following this logic the number of standstills is also reduced.

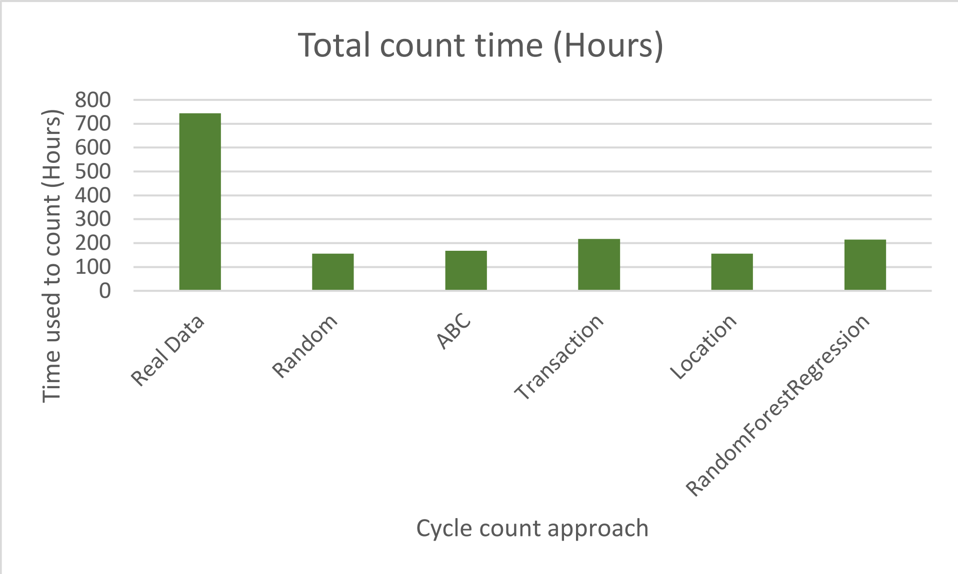


Figure 58 Time used for counting over half a year

6.1.2 Implementation

A setup for an implementation has also been provided to Euroma. Currently Euroma exports the transaction data of the ERP and MCS to CSV files. An algorithm has been devised that retrieves these rows and performs the same merge on the data as is described in the prediction testing. The algorithm then uses previous count moments to infer the discrepancy over the historical data. Using the combined and enriched dataset the prediction model is trained, all data of an SKU after the last correction is stored in a different dataset. The data after the last correction of an SKU is used to perform predictions after training the model. The predicted discrepancy at the last transaction of each SKU is provided in an excel file to the warehouse management specialist. Cycle counting has not been programmed. The Random Forest Regression model used in the simulation simply selects SKUs with the highest discrepancy. These can be found by sorting the provided file based on size. The diminishing population will have to be guarded manually. This algorithm can be expanded with its own database that would allow for the cycle counting method to also be implemented. However this was outside the scope of the research.

6.1.3 Discussion

Testing the cycle count models was done through a Monte Carlo simulation. During this simulation transactions are generated based on the statistical properties that were found in the historical data, see Section 4.4.6 for a more in depth explanation. While much effort was spend on creating a dataset that can be used to generate the required features, it is well possible that this still lacks to incorporate al nuances of the real data. As described in Section 2.7 an approximation was made for the transaction sizes. However, this this approximation was over all transactions of an SKU in a particular warehouse. If the transaction type also influences the size of the transaction this is not carried over into the simulation. Expanding the dataset to include this, possible, nuance would require a dataset that is 14 times larger than the current. Formulating this dataset would take up a lot of time while it does not guarantee any improvements.

The prediction model showed promising results when experimenting with the historical data. The model showed a good ability to fit to the data and the performance metrics indicated a decent prediction being made with R2 scores close to 1. However, when testing the prediction model on the simulated data it is found that the model predicts errors where there is none. Resulting in counts being performed on SKUs without any discrepancy. The possible reasons for this to happen are two-fold. Either the simulation does not properly represent the historical data or the model does overfit when training on the historical data, resulting in random predictions being made when applying it for inference.

The experimentations performed with the simulation and different cycle count approaches were limited. It could therefore well be possible that higher performance can be attained with one of the approaches when the parameters are better configured. For the experiments all approaches were given the same base parameters. In the case of ABC cycle counting approach it is can be assumed that increased testing with the model parameters, the division of the classes or the inclusion frequency of each class can improve the models performance. Comparing the data shows a lower level of executed counts and increasing this will likely increase the accuracy in the inventory record. The decision was made not to do further testing with individual model parameters as this is not the aim of this research.

6.2 Recommendations

6.2.1 Prediction model

While the prediction models showed good performance on the test data set, in the simulation the performance was lower than expected. As was also covered in the section 7.1 and 7.2 there are more than one reasons for why this might happen. To better be able to evaluate the performance of the prediction and compare this to the transaction based approach, a real time test would be required. By testing the performance on real data instead of a simulation it can be reviewed if transaction count actually shows the same as the prediction

Currently the model does not retrain on the discrepancy that it encounters. This decision was made because retraining the model on the simulated data is testing whether it can "learn a trick". By testing the model on actual data it is less likely to overfit and the actual performance can be reviewed. It is also possible to retrain the model every time a count has been performed and correct the "prediction" before retraining.

The model is currently training on three of the warehouses at Euroma. These warehouses are quite different in regard to how product is handled, the quantities in which it is stored and the level of use. Due to these differences it might be interesting to see if there is a difference in performance when splitting the data further and training on specific warehouses.

6.2.2 Cycle counting

The results that were returned from the cycle count experiments showed that any form of cycle counting is likely to improve the inventory accuracy. For this reason the recommendation is made that cycle counting is in fact implemented at Euroma. It is also recommended to perform a number of experiments on the real data to determine what works best.

Re-evaluating the ABC classification would also be interesting. The current classification only takes into account the yearly financial value when making the distinction between the SKUs. This fails to capture SKUs of low financial value and use but high importance in the process.

After having performed cycle counting in a structured format, the data that is available describing the discrepancies is likely to be more extensive and more accurate. Using this data to train a model could result in significant improvements over the model presented in this thesis that was based on unstructured and, possibly, incorrect counts.

6.3 Future research

This thesis has shown that there is a potential for the application of machine learning models in the prediction of inventory record inaccuracies to improve the cycle count decision in a production environment. This research has mainly focused on the automated parts of Euroma. The silos, both internal and external, are automated silos. The 'miniload' is not, however, this part of the warehouse did not show very large discrepancies. For future research the application of the machine learning models on the fluid warehouses would be an interesting field to explore. This warehouse has not automated parts and error is therefore more user dependent. Exploring the ability to predict discrepancies in an area that is intuitively more stochastic would be interesting.

Euroma also has a large part of the inventory of which use is not recorded but later on inferred on an order. Items such as pallet sheets and plastic wrap fall into these categories. Similar to the fluids, these products and their use is completely manual and dependent on operators. It would be interesting to see if errors can predicted here as well.

6.4 Contribution

Prediction of inventory record inaccuracies has been performed in the past. Past researchers investigate the application of various categorization models and neural networks to predict if a record was erroneous. These researches focused on a yes or no prediction. The contribution of this thesis is the investigation of applying machine learning models to predict discrepancies in inventory records continuously, instead of categorically. From the literature review no previous study was identified that aimed to predict the same information. Similar studies predicted backorder, or stock-out but not the error in the record.

The use of the prediction as an input for a cycle count model was also not previously investigated. While the assumption can be made that previous studies aimed at this application of their prediction, this was not found in any experiments.

6.5 Validation

The validation of the prediction model results is done through the use of K-fold cross-validation. This method ensures that the models perform well when presented with new data. Learning curves were applied to validate that the size of the dataset is sufficient. The learning curves showed that the required number of transactions was around 80.000, the training set is almost twice the required size for the ensemble models and neural network to fit to the data.

The cycle count simulation was based of the 2022 transaction data. This dataset was analyzed and the properties of each SKU were determined and stored in a table that is accessed during the simulation. During the simulation the transaction are generated based on random value generated in python. The transactions are based of the 2022 dataset, and the prediction model is trained on the 2022 dataset. Each run consisted of 150.000 transactions, constituting half a year of data and 5 runs were performed to ensure that the law of large numbers can be applied.

References

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Brooks, R. B., & Wilson, L. W. (1995). *Inventory Record Accuracy Unleashing the Power of Cycle Counting.* Wiley & Sons Ltd.

Chen, T., & Guesterin, C. (2016). XGBoost: A scalable tree boosting system. *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785-794). Association for Computing Machinery.

Chicco, D., Warrens, M., & Jurman, G. (2021). The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis. *PeerJ Computer Science*, 1-24.

Chuang, H. H.-C., & Oliva, R. (2015, Juli 31). Inventory record inaccuracy: Causes and labor effects. *Journal of Operations Management*, pp. 63-78.

De Santis, R., De Aguiar, E., & Goliatt, L. (2018). Predicting material backorders in inventory management using machine learning. *IEEE Lating American Conference on cCmputational Intelligence* (pp. 1-6). Arequipa: nstitute of Electrical and Electronics Engineers Inc.

DeHoratius, N., & Raman, A. (2008). Inventory Record Inaccuracy: An Emperical Analysis. *Management Science*, pp. 627-641.

Euroma. (n.d.). *Historie*. From Euroma: https://www.euroma.com/nl/about/Historie

Freedman, D. A. (2009). *Statistical Models - Theory and Practice.* New York: Cambridge University Press, New York.

Fulkerson, B., Michie, D., Spiegelhalter, D., & Taylor, C. (1995). *Machine learning, Neural and Statistical Classification.* Ellis Horwood.

Gumrukcu, S., Rossetti, M., & Buyurgan, N. (2008). Quantifying the costs of cycle counting in a two-echelon supply chain with multiple items. *International Journal of Production Economics*, Elsevier.

Hanley, J., & McNeil, B. (1982). The Meaning and Use of the Area under a Receiver Operating Charateristic (ROC) Curve. *Radiology*, 29-36.

Heerkens, H., & van Winden, A. (2017). Cause and Effect. In H. Heerkens, A. van Winden, & J.-W. Tjooitink, *Solving Managerial Problems Systematically* (pp. 42-43).

Hilt, D. E., & Seegrist, D. W. (1977). *Ridge: A Computer Program For Calculating Ridge Regression Estimates.* Upper Darby: Forest Service U.S. Department of Argriculture.

Iglehart, D. L., & Morey, R. C. (1972, April). Inventory Ssytems with Imperfect Asset Information. *Management Science*, pp. 388-394.

Jain, A., Mao, J., & Mohiuddin, K. (1996). *Artificial Neural Networks: A tutorial.* IEEE.

Kang, Y., & Gershwin, S. B. (2005, Februari 23). Information inaccuracy in inventory systems: stock loss and stockout. *IEE Transaction*, pp. 843-859.

Khader, S., Rekik, Y., Botta-Genoulaz, V., & Campagne, J.-P. (2014, September). Inventory Management subject to Multiplicative Inaccuracies. *International Journal of Production Research*, pp. 5055-5069.

Kurian, D., Maneesh, C., & Pillai, V. (2020). Supply chain inventory stockout prediction using machine learning classifiers. *Internatianal Journal Business and Data Analytics*, 218-232.

Oxford University. (2023). *Oxford English Dictionary.* Oxford: Oxford University Press.

Raman, A., DeHoratius, N., & Ton, Z. (2001, April 1). Execution: The missing link in retail operation. *California Management Review*, pp. 136-152.

Richard F. Gunst, R. L. (1977). Biased Estimation in Regression: An Evaluation Using Mean Squared Error. *Journal of the American Statiscal Association*, 616-628.

Rinehart, R. (1960, August 1). Effects and Causes of Discrepancies in Supply Operations. *Operations Research*, pp. 437-587.

Rossetti, M. D., Collins, T., & Kurgund, R. (2001). *Inventory Cycle Counting - A Review.* University of Arkansas.

Schrady, D. (1970, March). Operational definitions of inventory record accuracy. *Naval research logistics quarterly*, pp. 133-142.

Sheppard, G., & Brown, K. (1993). *Predicting inventory record-keeping errors with discriminant analysis: A field experiment.* Seattle: Elsevier.

Suthaharan, S. (2016). *Machine Learning Models and Algorithms for Big Data Classification.* Springer.

Tan, P.-N. (2009). Receiver Operating Characteristics. In L. Liu, & M. Tamer Ozsu, *Encyclopedia of Database Systems* (pp. 2349-2352). New York: Springer. From Springer Link.

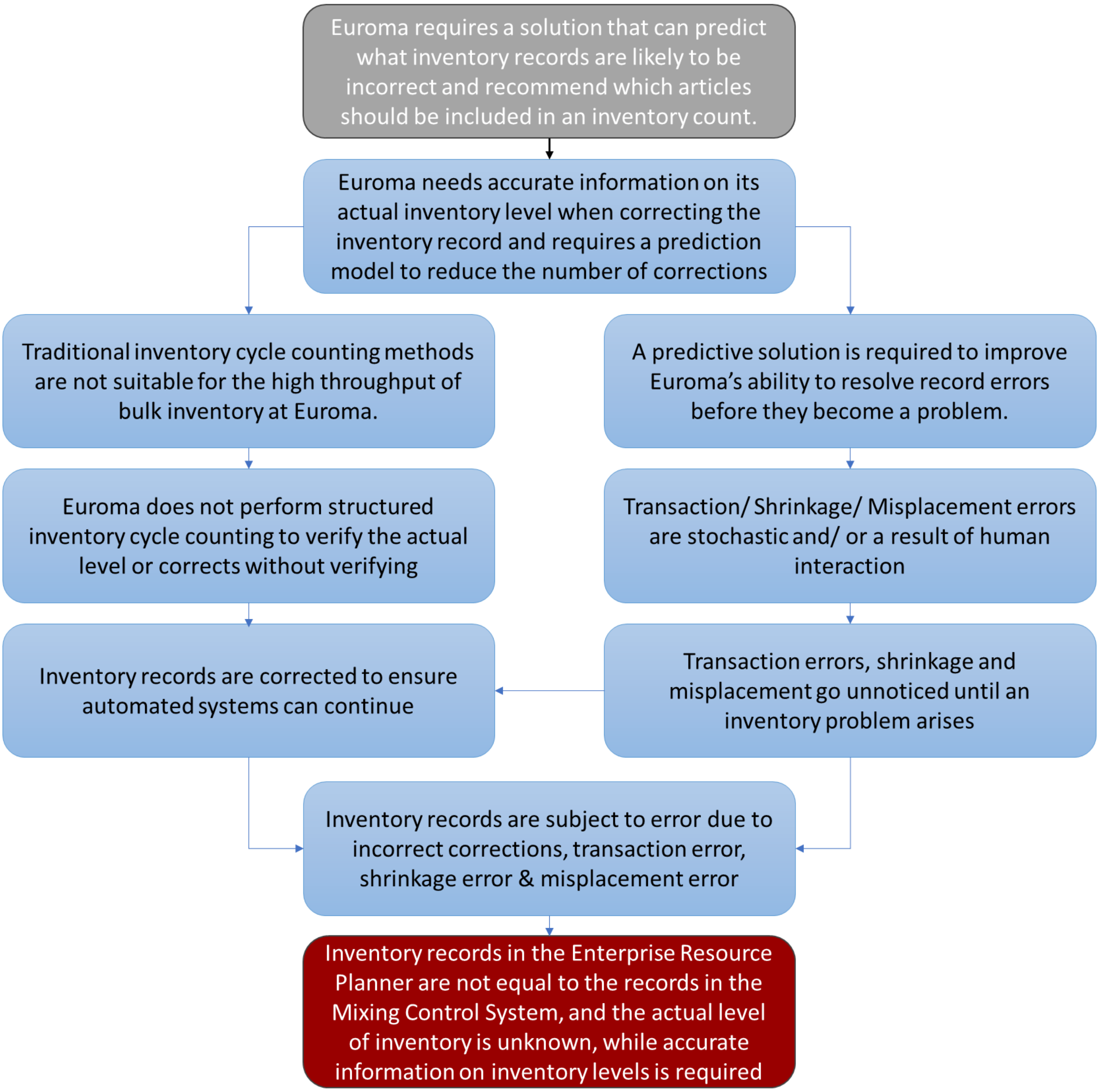
Tibshirani, R. (1995). *Regression Shrinkage and Selection via the Lasso.* Toronto: University of Toronto.

Torgo, L. (2017). Regression Trees. In C. Sammut, & G. I. Webb, *Encyclopedia of Machine Learning and Data Mining* (pp. 1080-1083). Springer.

Wijffels, L., Giannikas, V., Woodall, P., McFarlane, D., & Lu, W. (2016). An enhanced cycle counting approach utilising historical inventory data. *IFAC-PapersOnLine* (pp. 1347-1352). Elsevier B.V.

Appendix

A. Problem cluster / causal chain



B. Transaction types Enterprise Resource Planner

Type	***Meaning***	***Purpose***
On order	Inventory that has been ordered from a supplier but has yet to arrive	Regular transaction
Reservation	Inventory that has been reserved for an order in the future	Regular transaction
Receiving	Inventory that has been received at the new location for storage or use	Regular transaction
Supplying	Inventory that is being supplied to a new location for storage or use	Regular transaction
Inventory correction	Inventory status that has been changed, either by the system or a person, with the intention of consolidating missing inventory	Corrective transaction, manual

C. Transaction types Mixing Control System

Code	***Type***	***Meaning***	***Purpose***
100	Addition	Inventory is added to a location	Regular transaction
104	Manual addition	Inventory is added to a location by an operator	Corrective transaction, manual
108	Assign addition	Assign an addition to a tote	Regular transaction
200	Subtraction	Remove inventory from one location	Regular transaction
202	Reserved subtraction	A reservation for contents of a tote has been made so that it cannot be assigned to a different order	Regular transaction
300	Storage to Storage	Move inventory from one storage location to another, always in pairs	Regular transaction
328	Manual Storage to Storage	Move inventory from one location to another, transaction is created manually	Corrective transaction, manual
402	Negative inventory resolved by receival	A negative inventory has been resolved by adding inventory to the location	Corrective transaction, automated
408	Reserved/planned quantity breakdown	Contents of a tote are being split over two totes	Regular transaction
414	Inventory correction due to empty notification	Delete inventory remaining in system as the container has indicated its empty	Corrective transaction, automated
422	Inventory correction due to manual empty notification	Operator manually indicates container to be empty, remainder is removed from inventory.	Corrective transaction, manual
900	Automatic container emptying	Container sensor indicates that it is empty, remaining inventory is destroyed	Corrective transaction, automated
902	Remove warehouse inventory manually	Manually remove inventory from a storage location	Corrective transaction, manual

D. Encoding comparison

Approach	Regression Model	Fit time (sec)	Best R^2-score train set	R^2-score test set	MSE test set (kg)
One-hot encoding	*Lasso*	No fit	Nan	Nan	Nan
	Ridge	*76.090*	*0.797958*	*0.989490*	*212169.3607*
	Bayesian Ridge	*61.863*	*0.994267*	*0.989489*	*212184.5304*
	Decision Tree	*16.612*	*0.985383*	*0.980063*	*402487.5672*
	Random Forest	*16.317*	*0.997989*	*0.921087*	*1593143.746*
	Extreme Gradient Boost	*106.135*	*0.99987*	*0.98145*	*854236.125*
	Neural Network	No fit	Nan	Nan	Nan
Integer encoding	*Lasso Regression*	703.83	0.999645	0.990505	203332.3886
	Ridge Regression	3.1387	0.994179	0.989266	216685.7697
	Bayesian Ridge	0.8541	0.983589	0.989264	216738.342
	Decision Tree	1.5418	0.797958	0.922019	1549620.906
	Random Forest	8.0839	0.999376	0.966724	661237.1312
	Extreme Gradient Boost	51.702	0.973090	0.962697	798846.3086
	Neural Network	96.917	0.353092	0.978040	0.004505428

E. Hyperparameters for machine learning alternatives

Lasso Regression

Model	Datatype & default value	Description
Alpha	Float, default = 1.0	Constant that is multiplied with L1 term to control regularization strength. Alpha ϵ [0, inf)
Fit intercept	Boolean, default = False	Same as linear regression
Precompute	Boolean or array-like of shape (n, n), default = False	Indicate whether to use precomputed Gram matrix, or parse matrix as argument
Copy X	Boolean, default = True	Same as linear regression
Max iter	Integer, default = 1000	Same as linear regression
Tol	Float, default = $1e^{-4}$	Tolerance of the solver, this is used a stopping criteria. When the updated coefficients or the gap between the prediction and expectation is lower than the tolerance the model stops running
Warm start	Boolean, default = False	Indicate whether to reuse the previous solution as initialization
Positive	Boolean, default = False	Same as linear regression
Random state	Integer, RandomState instance, default = None	Seed number for the pseudo random number generator used to select the feature to update only used when "selection" is set to random
Selection	{'Cyclic', 'Random'}, default = 'Cyclic'	Based on the selected option the features are updated one by one or at random

Ridge Regression

Model	Datatype & default value	Description
Alpha	Float, ndarray of shape (n_targets), default = 1.0	The Alpha constant in ridge regression is aimed at controlling the regularization strength. Alpha ϵ [0, inf).
Fit intercept	Boolean, default = False	Same as linear regression
Precompute	Boolean or array-like of shape (n, n), default = False	Indicate whether to use precomputed Gram matrix, or parse matrix as argument
Max iter	Integer, default = None	Maximum number of iterations for conjugation. Default value differs per solver option.
Tol	Float, default = $1e^{-4}$	Same as lasso regression
Solver	{'auto', 'svd', 'cholesky', 'lsqr', 'sparse_cg', 'sag', 'saga', 'lbfgs'}, default='auto')	Indicates the solver to be used

Bayesian Ridge Regression

Model	Datatype & default value	Description
N_iter	Integer, default = 300	Maximum number of iterations, >=1
Tol	Float, default = $1e^{-3}$	Stopping criteria for algorithm, if validation and train results are within tolerance, stop
Alpha_1	Float, default = $1e^{-6}$	Shape parameter for the Gamma distribution prior over the alpha parameter
Alpha_2	Float, default = $1e^{-6}$	Inverse scale parameter for the Gamma distribution prior over the alpha parameter
Lambda_1	Float, default = $1e^{-6}$	Shape parameter for the Gamma distribution prior of the lambda parameter

Lambda_2	Float, default = $1e^{-6}$	Inverse scale parameter for the Gamma distribution prior of the lambda parameter
Alpha_init	Float, default = None	Initial value for alpha (precision of the noise). If not set, alpha_init is 1/Var(y), where y is the response variable
Lambda_init	Float, default = None	Initial value for lambda (precision of the weights). If not set, lambda_init is 1
Compute score	Boolean, default = False	If set to true, calculate the log marginal likelihood at each iteration of the optimization
Fit Intercept	Boolean, default = True	Indicate whether to calculate intercept for the data, required if data is not centered
Copy_X	Boolean, default = True	If true, X will be copied, else it might be overwritten
Verbose	Boolean, default = False	If true verbose mode is activated during model fitting and console provides more detailed output

F. Hyperparameter search spaces

Lasso Regression

Hyperparameters	**Search Space**	**Best option**
Alpha	[0.01, 0.02, 0.03, … 4.98, 4.99, 5.00]	0.1
Number of iterations	[100, 200, 300, 400, 500, 600, 700, 800, 900, 1000]	10000
Tolerance	[1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1]	1e-4
Selection	['cyclic', 'random']	Random

Decision Tree Regressor

Hyperparameters	**Search Space**	**Best option**
Criterion	['squared_error', 'friedman_mse', 'absolute_error', 'poisson']	Poisson
Splitter	['best', 'random']	Random
Max_depth	[1, 5, 10, 15, 20, 25]	20
Min_samples_split	[1, 2, 3, 4, 5]	4
Min_samples_leaves	[1, 2, 3]	3

Random Forrest Regression

Hyperparameters	**Search Space**	**Best option**
Criterion	['squared_error', 'friedman_mse', 'absolute_error', 'poisson']	Friedman_mse
Splitter	['best', 'random']	Random
Max_depth	[1, 5, 10, 15, 20, 25]	15
Min_samples_split	[1, 2, 3, 4, 5]	3
Min_samples_leaves	[1, 2, 3]	3
n_estimators	[5, 10, 15, 20, 25, 30]	20

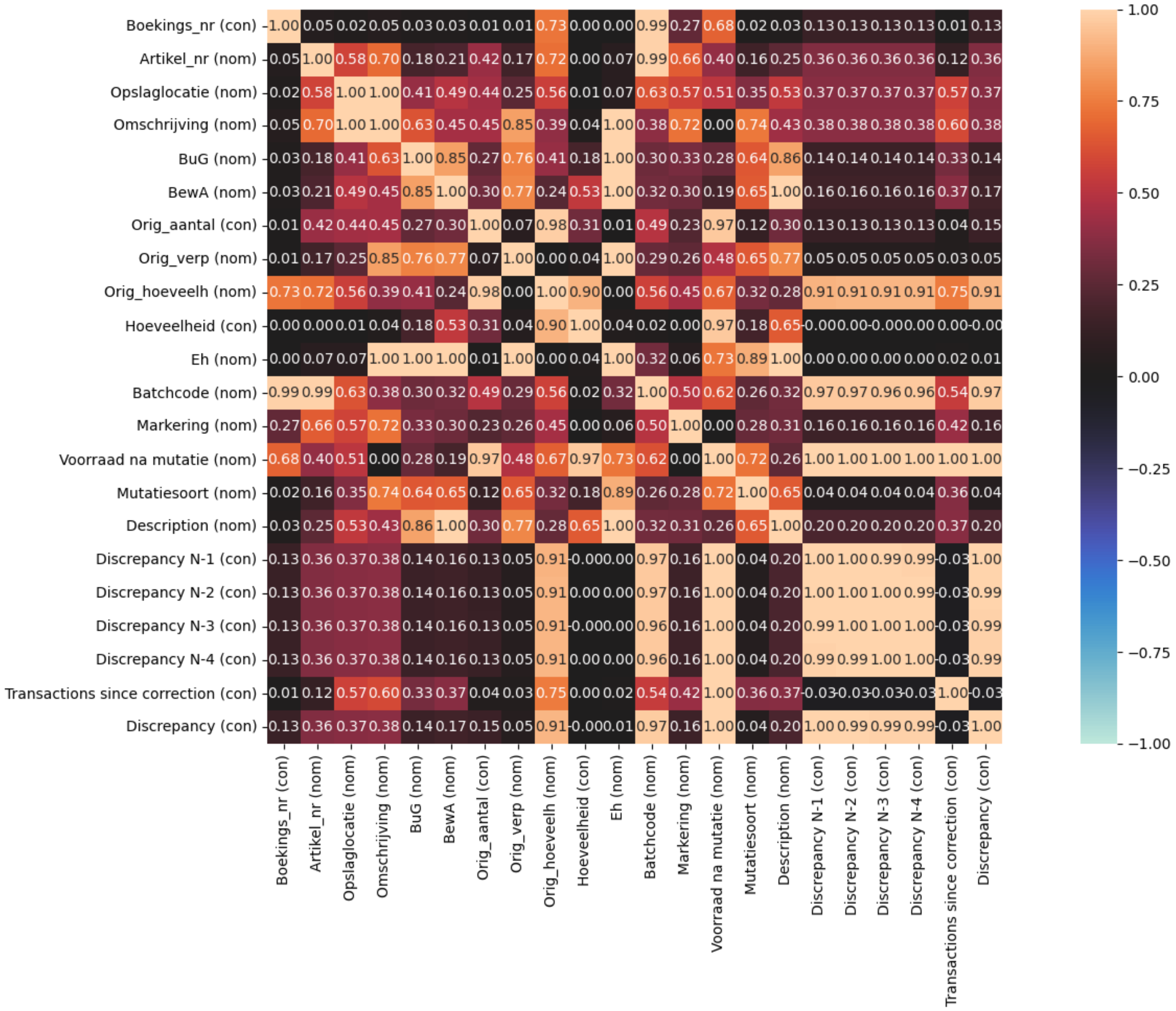
Extreme Gradient Boosting (GBTree)

Hyperparameters	Search Space	Best option
objective	["reg:squarederror"]	reg:squarederror
booster	["gbtree"]	Gbtree
eta	[0.01, 0.05, 0.1, 0.3]	0.3
gamma	[0, 0.1, 0.3, 1]	1
max_depth	[3, 5, 7, 9]	5
min_child_weight	[1, 3, 5]	3
Subsample	[0.6, 0.7, 0.8, 0.9]	0.8
colsample_bytree	[0.6, 0.7, 0.8, 0.9]	0.9
colsample_bylevel	[0.6, 0.7, 0.8, 0.9]	0.9
max_delta_step	[0, 1, 2]	0
lambda	[0, 1, 5]	1
alpha	[0, 1, 5]	0
n_estimators	[100, 200, 300]	100

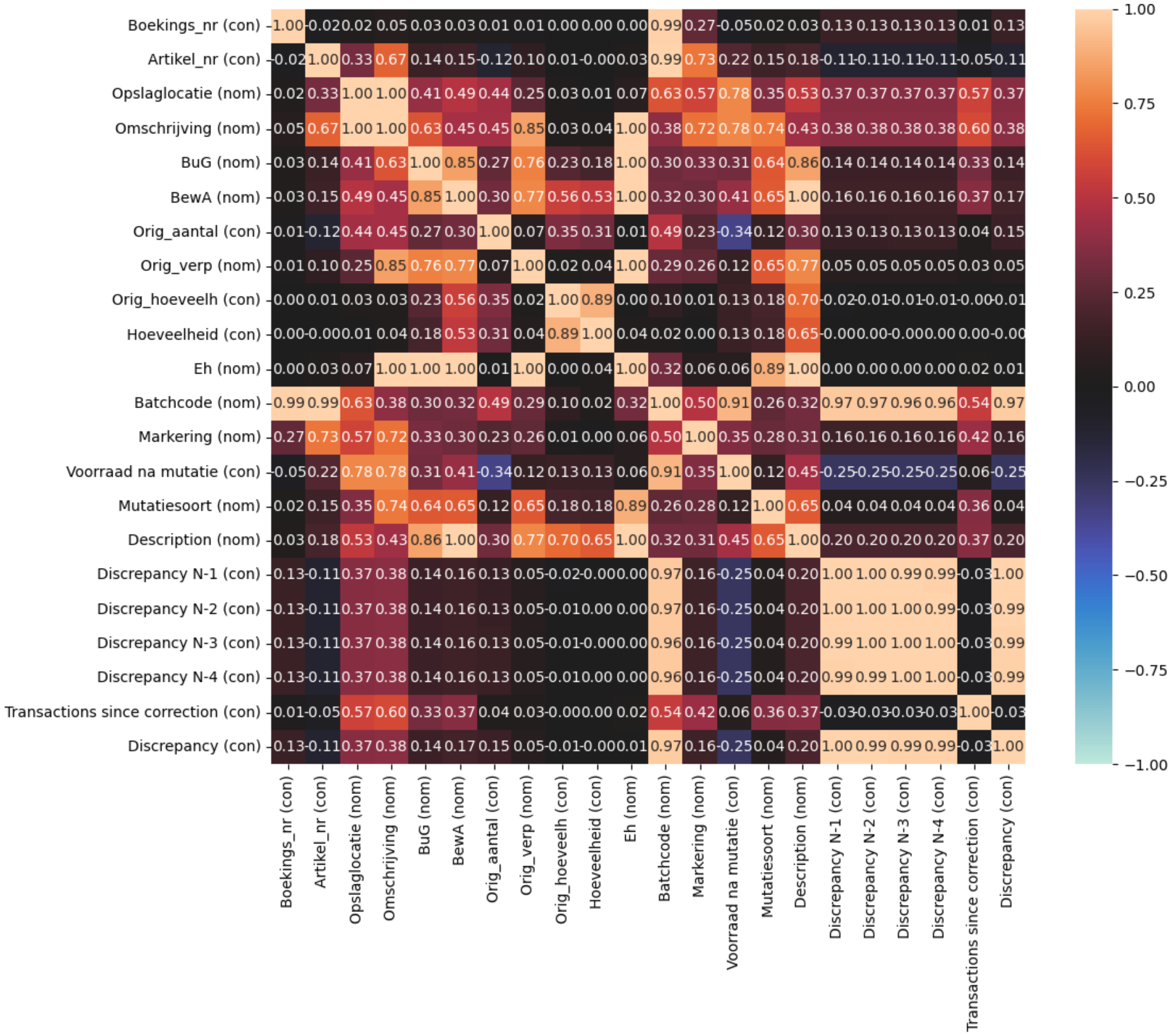
Neural Network

Hyperparameters	Search Space	Best option
hidden_layer_sizes	[20, 50, 100, 150]	20
activation	["identity", "logistic", "tanh", "relu"]	“relu”
solver	["lbfgs", "sgd", "adam"]	“lbfgs”
alpha	[1e-05, 0.0001, 0.001, 0.01]	0.01
batch_size	[50, 100, 150, 200]	100
learning_rate	["constant", "invscaling", "adaptive"]	“adaptive”
learning_rate_init	[0.0001, 0.001, 0.01, 0.1]	0.001
power_t	[0.1, 0.25, 0.5, 0.75, 0.9]	0.1
max_iter	[100, 200, 300]	100
tol	[1e-05, 0.0001]	0.0001
momentum	[0.1, 0.25, 0.5]	0.25
validation_fraction	[0.001, 0.01, 0.1]	0.01
beta_1	[0.95, 0.9, 0.75]	0.95
beta_2	[0.999, 0.95, 0.9]	0.95
epsilon	[1e-10, 1e-09, 1e-08]	1e-10
n_iter_no_change	[5, 10, 20, 30]	20
max_fun	[5000, 10000, 15000]	5000

G. Correlation heatmap, Inventory after transaction misidentified as categorical feature



H. Correlation heatmap, SKU number misidentified as continuous feature



I. Random cycle counting model

Sets	**Definition**
I	Set of all SKUs
J	Set of warehouses J ϵ {EZSI, EZDS, EZMV}
Parameter	**Definition**
S_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j has not been counted in current cycl} \\ 0 & \text{If already counted in current cycle} \end{cases}$
$\mathrm{E}[t_j]$	Expected time required to count a SKU warehouse j, $j \in J$
σ_{t_j}	Deviation in time required to count a SKU in warehouse j, $j \in J$
D_{ij}	Discrepancy of SKU i in warehouse j at the time of counting, $i \in I$,$j \in J$
AT	Available time for a cycle count
NS	Maximum number of SKUs to include in count
N	Number of items
W	Number of warehouses
Decision variable	**Definition**
$\mathrm{X_{ij}}$	$\begin{cases} 1 & \text{If SKU i in warehouse j is included in the count} \\ 0 & \text{Otherwise} \end{cases}$

Objective function

Traditional Appoach

$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \qquad (4.1.1)$$

Discrepancy Approach

$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} * D_{ij} \qquad (4.1.2)$$

Constraints

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq NS \qquad (4.2)$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq Y_{ij} \qquad (4.3)$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} (\mathrm{X}_{ij} * E[t_j]) \leq AT \qquad (4.4)$$

$$X_{ij} \leq S_{ij} \qquad \forall i,j \qquad (4.5)$$

$$X_{ij} \in \{0,1\} \qquad (4.6)$$

J. ABC Cycle counting model

Sets	**Definition**
I	Set of all SKUs
J	Set of warehouses J ϵ {EZSI, EZDS, EZMV}
Classes_c	Classes used in the abc approach, $c \in \{a, b, c\}$
Parameter	**Definition**
S_{ijc}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j is in category c} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j has not been counted in current cycle} \\ 0 & \text{If already counted in current cycle} \end{cases}$
$E[t_j]$	Expected time required to count a SKU warehouse j, $j \in J$
σ_{t_j}	Deviation in time required to count a SKU in warehouse j, $j \in J$
D_{ij}	Discrepancy of SKU i in warehouse j at the time of counting, $i \in I$,$j \in J$
AT	Available time for a cycle count
NS	Maximum number of SKUs to include in count
NS_c	Maximum number of SKUs to include in count from category c, $c \in \{a, b, c\}$
N	Number of items
W	Number of warehouses
Decision variable	**Definition**
X_{ijc}	$\begin{cases} 1 & \text{If SKU i in warehouse j is of type c and is included in the count} \\ 0 & \text{Otherwise} \end{cases}$
Objective function	
Traditional Appoach	$\max \sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij}$
Discrepancy Approach	$\max \sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij} * D_{ij}$
Constraints	
	$\sum_{i=0}^{N} \sum_{j=0}^{W} X_{ijc} \leq NS_c \quad \forall\, c$
	$\sum_{i=0}^{N} \sum_{j=0}^{W} X_{ijc} \leq Y_{ij} \quad \forall\, c$
	$\sum_{i=0}^{N} \sum_{j=0}^{W} (X_{ijc} * E[t_j]) \leq AT \quad \forall\, c$
	$X_{ijc} \leq S_{ijc} \quad \forall i, j, c$
	$X_{ijc}, S_{ijc} \in \{0,1\} \quad \forall\, i, j, c$

K. Transaction based cycle counting model

Sets	Definition
I	Set of all SKUs
J	Set of warehouses J ϵ {EZSI, EZDS, EZMV}

Parameter	Definition
RC_{ij}	Set that contains the transaction count since the last correction for all SKUs i in warehouse j, $i \in I,\ j \in J$
S_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j has not been counted in current cycle} \\ 0 & \text{If already counted in current cycle} \end{cases}$
$E[t_j]$	Expected time required to count a SKU warehouse j, $j \in J$
σ_{t_j}	Deviation in time required to count a SKU in warehouse j, $j \in J$
D_{ij}	Discrepancy of SKU i in warehouse j at the time of counting, $i \in I, j \in J$
AT	Available time for a cycle count
NS	Maximum number of SKUs to include in count
N	Number of items
W	Number of warehouses

Decision variable	Definition
X_{ij}	$\begin{cases} 1 & \text{If SKU i in warehouse j is included in the count} \\ 0 & \text{Otherwise} \end{cases}$

Objective function	
Traditional Appoach	$\max \sum_{i=0}^{N} \sum_{j=0}^{W} RC_{ij} \times X_{ij}$
Discrepancy Approach	$\max \sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij} * D_{ij}$

Constraints

$$\sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij} \leq NS$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} X_{ij} \leq Y_{ij}$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} (X_{ij} * E[t_j]) \leq AT$$

$$X_{ij} \leq S_{ij} \qquad \forall i, j$$

$$X_{ij} \in \{0,1\}$$

L. Location based cycle counting model

Sets	Definition
I	Set of all SKUs
J	Set of warehouses J ϵ {EZSI, EZDS, EZMV}

Parameter	Definition
S_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j} \\ 0 & \text{Otherwise} \end{cases}$
Y_{ij}	$\begin{cases} 1 & \text{If SKU i is stored in warehouse j has not been counted in current cycle} \\ 0 & \text{If already counted in current cycle} \end{cases}$
W_j	$\begin{cases} 1 & \text{If SKUs from warehouse j are included in the count} \\ 0 & \text{Otherwise} \end{cases}$
$\mathrm{E}[t_j]$	Expected time required to count a SKU warehouse j, $j \in J$
D_{ij}	Discrepancy of SKU i in warehouse j at the time of counting, $i \in I$,$j \in J$
AT	Available time for a cycle count
NS	Maximum number of SKUs to include in count
NW	Number of warehouses to include in the cycle count, NW $=$ 1 or 2
N	Number of items
W	Number of warehouses

Decision variable	Definition
$\mathrm{X_{ij}}$	$\begin{cases} 1 & \text{If SKU i in warehouse j is included in the count} \\ 0 & \text{Otherwise} \end{cases}$

Objective function

Traditional Appoach

$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij}$$

Discrepancy Approach

$$\max \sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} * D_{ij}$$

Constraints

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq NS$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} \mathrm{X}_{ij} \leq Y_{ij}$$

$$\sum_{i=0}^{N} \sum_{j=0}^{W} (\mathrm{X}_{ij} * E[t_j]) \leq AT$$

$$\sum_{j=0}^{J} W_j \leq NW$$

$$X_{ij} \leq W_j \qquad \forall\, i, j$$

$$X_{ij} \leq S_{ij} \qquad \forall i, j$$

$$X_{ij}, W_j \in \{0,1\}$$