The Business Impacts of Artificial Intelligence Predictions: Designing a Forecast Evaluation Model for the Food Supply Chain

MSc in Management of Information Systems and Digital Innovation

The London School of Economics and Political Science

Course: MG4D7 MSc MISDI Dissertation

Supervisor: Dr. Kari Koskinen

Candidate nr: 19301

Word count: 8764

August 2019

Abstract

The present study followed Hevner´s (2007) design science cycles in the attempt to design an evaluation model artefact for Artificial Intelligence enabled forecast models in the food supply chain. Through industry analysis and practitioner research, the relevance cycle identified the need for an evaluation model artefact to that could evaluate model performances in the context of food product demand forecasting. The rigor cycle identified the state of the art within forecast evaluation, relevant concepts from the fields concerned with measuring business performance, as well as several design methods from the entrepreneurship and information systems fields that would be useful in designing such a model. The design cycle was split in two phases; The problem design phase applied composite conceptual models drawn from various information systems design methods to capture assumptions about the food supply chain order process and conceptualise the role of forecasts in this context. It then tested conceptual models for the evaluation model itself. During the artefact design phase a composite artefact capturing the evaluation model was iteratively designed and tested. This artefact successfully met the requirements set out in the relevance cycle, both enabling the the validation of hypotheses about the food supply chain order decision process and the evaluation of forecasting models in this specific business context. Outcomes from the study include requirements for a forecast evaluation model in a supply chain context, as well as insights into how to develop evaluation models for Artificial Intelligence enabled prediction systems.

Keywords: food industry, supply chain, forecast, prediction, artificial intelligence, machine learning, information systems, design science.

Acknowledgements

I would like to thank my dissertation supervisor Dr. Kari Koskinen for steady and reliable advice in the process, as well as general solid conversations. I would also like the thank Dr. Carsten Sørensen for advice on the direction of the dissertation and for various reflections about digital life.

I would also like to thank Are Traasdahl and Dag Liodden for the opportunity to work with Crisp for the dissertation, and Dag for throwing me in the deep end with the challenge that became the topic of this dissertation, for engagement and support through the process. Thank you to all the members of the Crisp team for welcoming me during my time with the organization, I found working with you truly inspiring. I would especially like to thank Sam Mourad for mentorship and close collaboration.

Contents

1.0 Introduction

Improved technological capabilities is making it possible to develop new types of information systems incorporating Artificial Intelligence (AI) (Rzepka & Berger, 2018). The prominence and future promise of these AI systems is so significant that they are predicted to potentially disrupt entire industries, and have a considerable impact on business and society (Agrawal, Gans, Goldfarb, 2018a; Brynjolfsson & McAfee, 2016; McKinsey, 2017). Despite this, the evaluation mechanisms used in most AI models fail to capture these impacts, rather relying on purely statistical metrics (Hyndman & Athanasopoulos, 2019; Sanders & Graham, 2009). The present study applies Hevner´s (2007) Design Science cycles to address this problem, iteratively designing an AI model evaluation artefact. This research was conducted in collaboration with the startup company Crisp (hereafter "host organization"); Crisp aim to address sustainability challenges in the food industry by facilitating data-driven decision making through AI driven forecasting services (Crisp, 2019). The AI model artefact was developed in the context of this host organization, addressing the task of evaluating forecasts that form the basis of order- and production decisions in the retail food supply chains of Norway and the US.

After framing the industry and organization specific problem of AI forecast evaluation during the relevance cycle, the present study identifies potential evaluation model elements and applicable information system design techniques during the rigor cycle. The main design cycle conducted in two phases: First the problem is designed and tested using conceptual tools from Information systems development, then these insights are used as basis for iterative design of a forecast evaluation artifact. This artifact is successfully validated and tested in dialogue with both internal and external stakeholders. The artifact result is a Minimum Viable Product consisting of a validated conceptual model capturing requirements for an evaluation IT artifact, and a functional implementation of these requirements through a software package used in successful model evaluation. Outcomes relevant for academia and industry are highlighted during the discussion. These include insights on how to develop an evaluation framework for a specific Artificial Intelligence application, linking the present study to previous research.

2. Methodology

2.1 Embedded Design Science

This dissertation applies a Design Science approach as prescribed by March & Smith (1995). The approach is defined as "designing artifacts to attain goals", were the aim is to create innovative and valuable models through the activities of building and evaluating (March & Smith, 1995 p.253). As such, it is well suited studies of IT artifacts, as it allows the researcher to build the artefact or a related construct, and evaluate this based on its context (March & Smith, 1995). It is also a method well suited to 'wicked' problems, such as the design and introduction of IT artefacts in traditionally oriented industries (Pries-Heje & Baskerville, 2008). The researcher spent one month embedded as a Data science professional with the host organization. During this time, the researcher carried out Hevner´s (2007) design science cycles to design an AI evaluation model. These consist of the relevance cycle, the rigor cycle and the central design cycle.

The embeddedness allowed all three cycles to draw on qualitative methods for participation and observation research to support the main design activities (Eisenhardt, 1989 p.538; Marshall & Rossman, 1995 p.79; Yin, 2009). It also allowed the relevance cycle to reveal an issue of high importance to industry, both linking the present research to the long tradition of socio-technical participatory design (e.g. Bjerknes, Ehn & Kyng, 1987; Greenbaum & Kyng, 1991; Ehn, 1988; Greenbaum, 1995).

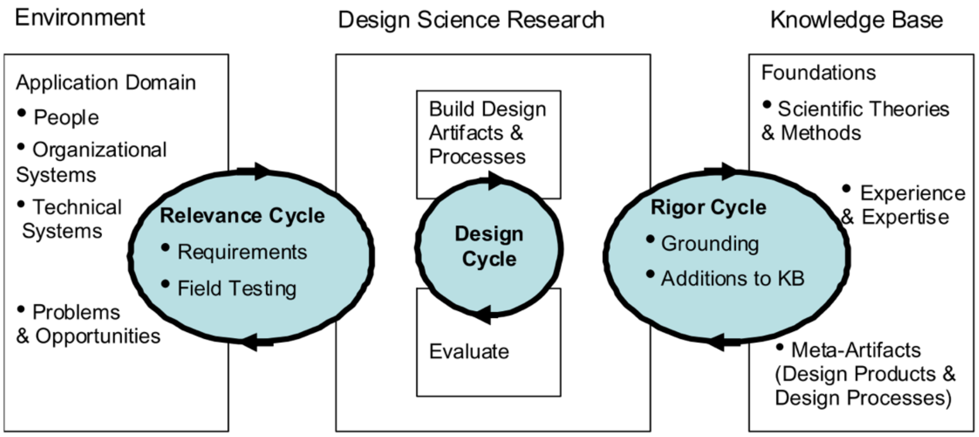


Figure 1: Hevner´s (2007) Three Design Science Cycles

2.2 The Relevance Cycle

The relevance cycle connects the design science activities to the context environment and begins with identifying the relevant challenges and opportunities (Hevner, 2007). The relevance cycle was carried out by collecting primary data through embedded practice and documentation review at the host organisation. Participant observations also supplemented by secondary industry research and domain specific review academic literature was also conducted to further understand and frame the problem. Based on this problem definition, artefact acceptance criteria by which the artifact was evaluated were identified (Hevner, 2007).

2.3 The Rigor Cycle: Review of literature and grounding knowledge

the Rigor cycle provides the contextual knowledge to the design project by applying existing methodologies and foundations in an appropriate manner. This ensures the research outcome is innovative (Hevner, 2007; Hevner et al. 2004). At this stage, literature that could contribute to a potential solution of the problem was reviewed, providing further academic grounding for the study (Hevner et al. 2004). Additional domain knowledge from related fields were also reviewed, providing elements for initial hypotheses the evaluation model. Design science can be draw on a variety of idea sources instead of being grounded in any specific theory (Hevner, 2007). The second part of the Rigor cycle therefore consulted the state-of-the-art application domain knowledge in the field of Information systems and entrepreneurship, exploring applicable tools and design techniques (Hevner, 2007).

2.4 The Design Cycle

The design cycled forms the central part of the design science project, and includes the rapid iteration between artifact construction, evaluation and feedback for refinement (Hevner, 2007).

In this cycle, the researcher generates design alternatives which are then evaluated against the requirements specified earlier in the research, iterating until the design is satisfactory (Simon, 1996). The requirements for the design were identified in the relevance cycle, while Methods and theories for design and evaluation where found during the rigor cycle (Hevner, 2007) .Creative insights were also allowed to emerge during the design process (Csikszentmihalyi, 1996). The artefact design had two main phases; an initial problem design phase allowed for in-depth exploration of the problem domain coupled with some early solution concepts, while a second Artifact design phase allowed for iterations on the solution artifact.

The Problem Design Phase

The first iteration of the artifact was derived from the relevance and rigor cycles, following Hevner ´s (2007) design cycle methodology. An initial creative design workshop was conducted with the data scientist, from which four scenarios were produced. The scenarios were multi layered in that they incorporated elements from several of the information systems design tools identified in the Rigor Cycle. The scenarios were the following*:* (1) The current status of the Food Supply Chain Order Decision process, (2) The current status of the order decision process, (3) Forecast Business Value Impact, (4) Order Decision as a Markov Decision / Reinforcement Learning Process.

The Artifact Design Phase

The second design phase entailed design of the a Minimum Viable Product artifact, capturing hypotheses and enabling testing of these (Ries, 2011). Two full iterations were conducted, and the artifact took the form of a composite artifact in three levels of abstraction. The main artifact was the evaluation model requirement, which was kept implementation non-specific. To practically test the concept as a Minimum Viable Product, this was implemented in two lower levels of abstraction; a software package in the programming language Python (Appendix 1), which in turn was implemented through client specific scenario models in the data science environment Jupyter Notebooks. The actual client data is confidential, but Appendix 2 includes an example notebook with simple simulated data and forecasts.

Testing

Artifacts were tested both against internal client documentation (Marshall & Rossman, 1995 p.85) and in dialogue with the following stakeholders:

Internal at host organization:

- Chief Technical Officer
- Data Scientist
- Engineer
- Engineering Manager
- Head of Products

External clients (anonymized):

- Co-founder representative of Client 1
- Representative from Client 2
- Representative from Client 2
- Representative from Client 3

2.5 Design Science Process

Figure 2 shows how the present study executed the design science process. In the relevance cycle, environment participation, industry research and academic literature were used to defined general and specific problem themes along with model acceptance criteria. The rigor cycle identified potential model elements and design tools. The design cycle then iterated between artifact creation and tests in the environment in two phases, problem phase and artifact design phase. This resulted in outcomes with academic insights towards problem themes and evaluation model implementation to “improve the environment” (Hevner, 2009).

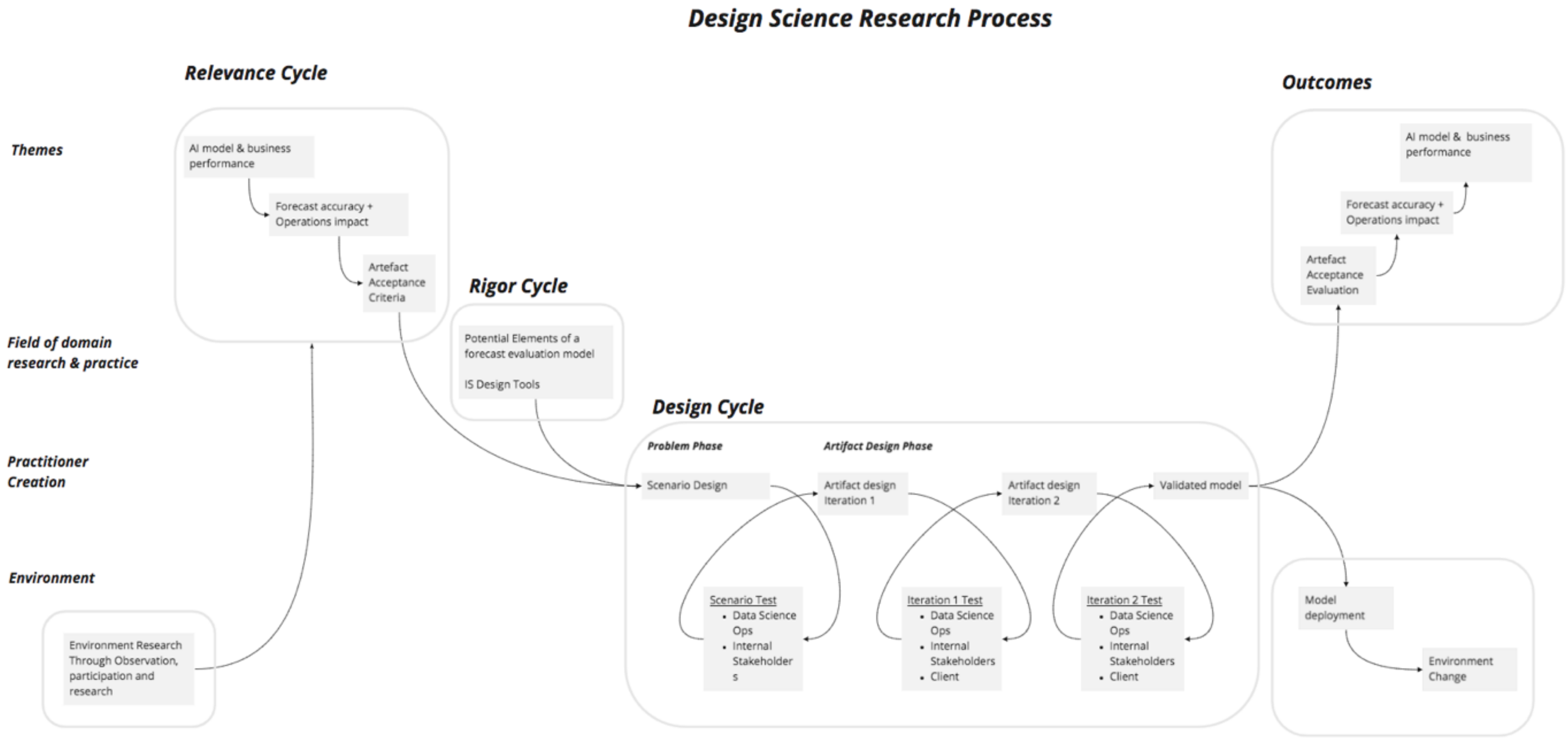

Design Science Research Process
Relevance Cycle
Outcomes
Themes
AI model & business performance
Forecast accuracy + Operations impact
Artefact Acceptance Criteria
Rigor Cycle
Potential Elements of a forecast evaluation model
IS Design Tools
Field of domain research & practice
Design Cycle
Problem Phase
Artifact Design Phase
Practitioner Creation
Scenario Design
Artifact design Iteration 1
Artifact design Iteration 2
Validated model
Environment
Scenario Test
Data Science Ops
Internal Stakeholders
Iteration 1 Test
Data Science Ops
Internal Stakeholders
Client
Iteration 2 Test
Data Science Ops
Internal Stakeholders
Client
Environment Research Through Observation, participation and research
AI model & business performance
Forecast accuracy + Operations impact
Artefact Acceptance Evaluation
Model deployment
Environment Change

Figure 2: The Implemented Design Science research process.

3. Relevance Cycle

This phase begins with identifying the challenges and opportunities in a specific environment application environment (Hevner, 2007). The host organization was serving the food supply chain industries of Norway and the US. The following section therefore reviews the challenge of waste in the food industry and how the host organization was addressing this challenge, framing the problem of AI forecast evaluation.

3.1 Food Industry Waste and Operational challenges

The global food industry faces significant sustainability challenges, and the UN estimates that "approximately one third of the global food production is wasted or lost annually" (Gustavson et.al. 2011; International Journal of Production Economics, 2014; UN Environment, 2019). In the US markets food waste is estimated "between 30-40 percent", amounting to $161.6 billion in losses (U.S Department of Agriculture, 2019; Buzby et. al., 2014), representing 2.5 percent of yearly US energy consumption (Webber 2012). In the significantly smaller Norwegian market, food waste still accounts for a NOK 22 billion (approx. $2.48 billion) annual loss (Stensgård, Prestrud, Hanssen & Callewaert, 2018). 66% of the losses are in food groups where freshness is an important criteria for consumption (Buzby et. al., 2014), and Spoilage often occurs due to over-ordering and overstocking which is a result of forecasting difficulties (U.S Department of Agriculture, 2019; Buzby, Wells & Hyman, 2014; Gustavson et. al., 2014; Ilbery et al., 2004). These uncertainties can cause the phenomenon known as the "bullwhip effect" (Figure 3), where larger order size fluctuations as orders are relayed through the supply chain (Ivanov et. al., 2017).

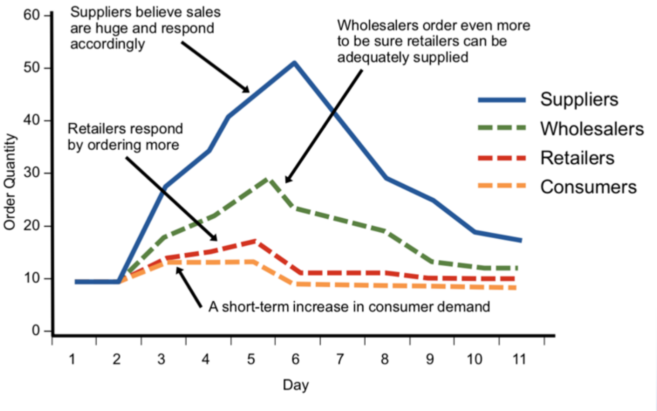


Figure 3: The Bullwhip Effect Illustrated. (Ivanov et. al., 2017)

Underlying factors of these operational inefficiencies also include lack of information sharing between organisations and inefficient decision making systems (Zhong et. al., 2017). The host organizations findings at firm level confirmed this, with reports including widespread use of legacy IT, unsophisticated forecasting techniques and manual overriding of forecasts which resulting in worsened outcomes. Addressing such challenges through digitization and data-driven decision making have been associated with higher productivity rates across industries (Brynjolfsson & McElheran, 2016), as they enable organisations to bypass human biases (e.g. Kahneman & Tversky, 1984; Shiller, 2003). The food industry has been slow to adopt digital technologies (Demartini, Pinna, Tonelli, Terzi, Sansone & Testa, 2018), but early efforts have contributed to food waste reduction and operational improvements(e.g. Li et. al., 2014; Jagtap & Rahimifard, 2019).

3.2 Forecasting models

The host organisation had therefore chosen to first enter the food supply chain information services market by offering a forecast solution, implementing forecasting techniques ranging from well-established industry standards for time series forecasting such as ARMA and Holt-Winters to newer state-of-the art models , such as Facebook´s Prophet (Figure 4) and Amazon Forecast´s DeepAR Plus (Figure5).

The Facebook Prophet model is composed of several sub-functions, conceptually similar to a Generalized Additive Model (GAM) (Taylor & Letham, 2017). DeepARPlus on the other hand, is based on the Recurrent Neural Networks using LSTM-Cells. Such techniques have been found able to produce accuracies above those produced by traditional forecasting techniques (Salinas et. al., 2019; Makridakis, Spiliotis & Assimakopoulos, 2018). However, in comparing these and other models, the data science team encountered the challenge of identifying which models produced better forecasts for the clients.

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t.$$

Figure 4: Facebook Prophet can be decomposed into three functions, trend, seasonality and holidays, plus an error term (Taylor & Letham, 2017).

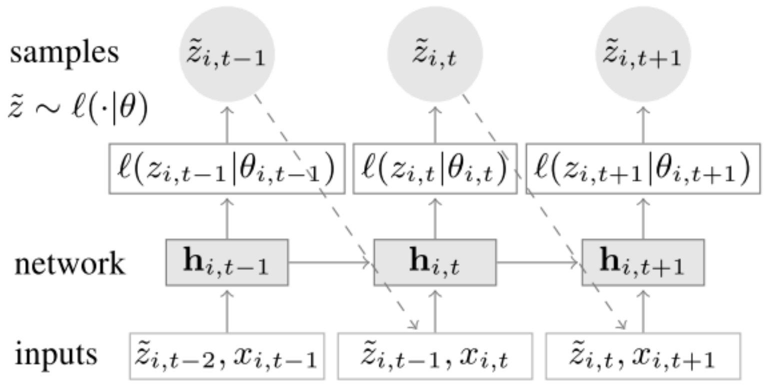

samples
$\tilde{z} \sim \ell(\cdot|\theta)$
$\tilde{z}_{i,t-1}$
$\tilde{z}_{i,t}$
$\tilde{z}_{i,t+1}$
$\ell(z_{i,t-1}|\theta_{i,t-1})$
$\ell(z_{i,t}|\theta_{i,t})$
$\ell(z_{i,t+1}|\theta_{i,t+1})$
network
$\mathbf{h}_{i,t-1}$
$\mathbf{h}_{i,t}$
$\mathbf{h}_{i,t+1}$
inputs
$\tilde{z}_{i,t-2}, x_{i,t-1}$
$\tilde{z}_{i,t-1}, x_{i,t}$
$\tilde{z}_{i,t}, x_{i,t+1}$

Figure 5: DeepArPlus uses a Recurrent Neural Net formulation (Salinas et. al., 2019).

3.3 Defining a 'good' Artificial Intelligence model

Time series forecasting is an application of statistical machine learning, which is part of the Artificial Intelligence field (James et. al., 2017; Russell & Norvig, 2016). The question of what entails a "good" forecasting model therefore generalizes to what makes an AI model "good". In Artificial Intelligence, performance measures are designed to evaluate what is a "good" outcome. It is recommended that these performance measures are designed "according to what one actually wants in the environment" (Russell & Norvig, 2016 p.43). Performance measures should therefore reflect how the AI model impacts its application domain, as the purpose of machine learning in AI is to improve future task performance (Géron, 2017; Goodfellow et. al., 2016; James et. al., 2017; Russell & Norvig, 2016 p.693). A performance measure should correspond to "the desired behavior of the system" in the application domain (Goodfellow et. al., 2016 p.101). Demand forecasting is concerned with predicting future demand, providing information basis for improved operational decisions outcomes (Ivanov et. al., 2017; Taha, 2013). A forecasts performance measure for should therefore capture whether the predictions provide a good information basis for improved operational outcomes.

Evaluation Challenge in Data Science Operations

A common practice in data science is to use statistical measures for model accuracy and / or goodness. These facilitate model selection and tuning (technical adjustments) towards attaining desirable model prediction accuracies, enabling the improvement of the resulting prediction products (Géron, 2017; Goodfellow et. al., 2016; James et. al., 2017). A standard in forecasting is the statistical metric Mean Absolute Percentage Error (MAPE) (Hyndman & Athanasopoulos, 2019; Ivanov et. al.,2017 p.308), which initially used in the host company´s data science operations.

Much of the waste in the food industry comes from the perishable foods (Buzby et. al., 2014), and Client interviews also indicated that indicated that demand accuracies on short timeframes were of high importance. However, because MAPE fails to signal many critical business considerations, such as the assymmetric cost of over-vs under-production, or how lead-time and shelf-life decide what period forecast errors should be evaluated over. MAPE also failed to reflect whether the forecast was able to capture short term trends such as daily demands. This led to the need to identify what constituted a “good” forecast.

$$p_t = 100 e_t / y_t.$$

$$\text{Mean absolute percentage error: MAPE} = \text{mean}(|p_t|).$$

Figure 6: *The mathematical formulation of MAPE.* For each timestep, the error amount is divided by the correct target value and put into percentage terms. The mean of the absolute values of the percentage errors is then taken (Hyndman, & Athanasopoulos, 2019).

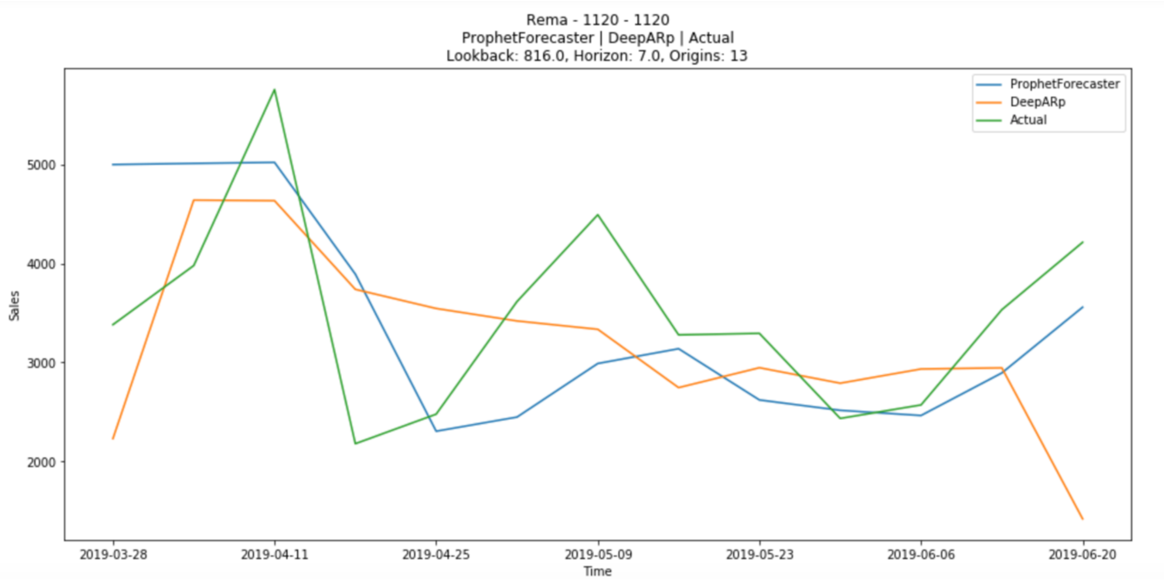


Figure 7:Forecasts from Facebook Prophet and Amazon DeepARPlus compared to Actual demand.

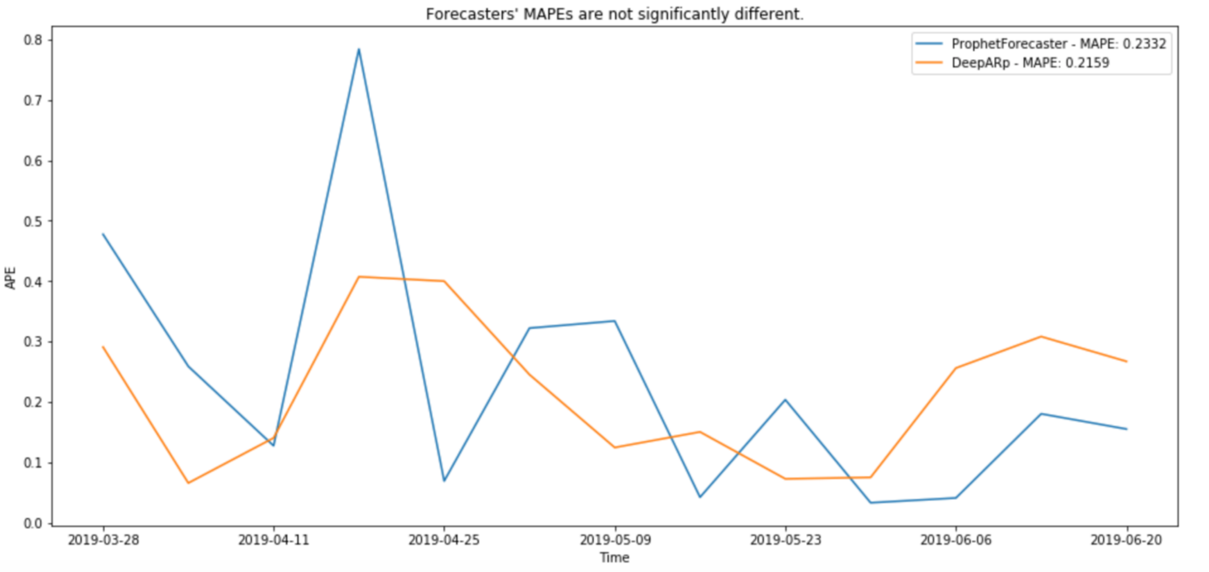


Figure 8: Mean Average Percentage Error Amazon DeepAR and Facebook Prophet compared to actual demand.

The ambiguity of MAPE is illustrated in figure 7 and figure 8. These models are experimental models from internal in R&D, and do not reflect the performance level of production ready forecasts. The MAPE value of the DeepARPlus model is slightly lower than that of the Prophet model, though not statistically significant using students t-test with 5% threshold (figure 8). However, looking at the forecasts in figure 7 we can observe that following the DeepARplus (yellow) would result in cycling between over- and underproduction, while following the Prophet model (blue) would lead to some overproduction in mid-April, followed mainly by underproduction. Whether the deficits in periods of underproduction could be covered by surplus from overproduction would depend on the shelf-life of products, but we cannot tell from the metric whether it would be possible to store these products, and how storing or discarding the products would impact the business. It is also not possible to tell whether the potential deficit is acceptable or not, and whether running a deficit or having excess inventory impacts the business most. This illustrates how MAPE is not able to capture important information about the business value of forecasts, such as the asymmetric impacts of over- and underproduction(Sanders & Graham, 2009). Although MAPE values the two forecasts similarly, they might have significantly different business impacts.

3.4 The need for a new measure

Statistical concepts of accuracy are often adopted to ensure the model can generalize from the training data to new observations, and state-of-the-art practice include several evaluation metrics for time series forecasting (Géron, 2017; Hyndman & Athanasopoulos, 2019). Scale dependent measures include metrics such as Mean Absolute error and Root Mean Squared Error, Mean Absolute Percentage Error Mean Absolute Scaled Error , (sMAPE2) , with the most common being Mean Absolute Percentage Error (MAPE) (Géron, 2017 p.37-39; Hyndman & Athanasopoulos, 2019; Ivanov et. al.,2017 p.307).

However, the challenge of these mean based metrics is that they fail to consider any practical implications of forecasting accuracy for specific application domains such as ordering and production in the food supply chain (Ivanov et. al, 2017 p.308; Kerkkänen et. al. (2009). Suggested alternative measures also consider only technical performance indicators (e.g. De Gooijer, & Hyndman, 2006; Hyndman & Koehler, 2006; Sungil, & Heeyoung, 2016). More complex loss functions have been suggested for classification task (e.g Zhang & Chandrasekar, 2017), However, summary and average statistics are still widely used even in state-of-the art forecasting research (Baecke, P., De Baets, S., & Vanderheyden, K.,2017; Kilimci, 2019; Lai et. al., 2018). These findings suggests a lack of links between AI model evaluation and business performance measures (Figure 9).

Research has also argued adequate "adequate justification" is important for knowledge system acceptance (Gregor & Benbasat, 1999), and the host organizations findings suggested clients would sometimes override their systems when they lacked trust in the recommendations. A relevant forecast evaluation was therefore also relevant for client acceptance of the host company´s solution.

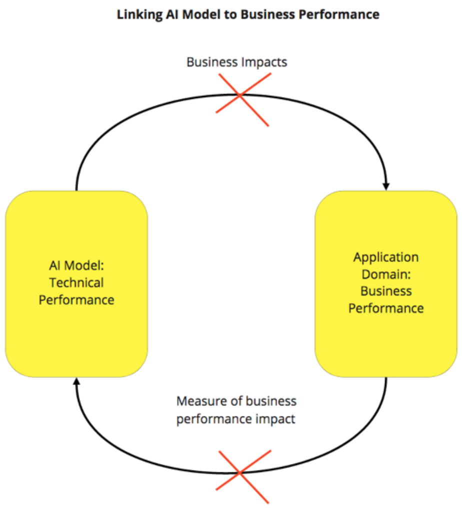

Linking AI Model to Business Performance
Business Impacts
AI Model: Technical Performance
Application Domain: Business Performance
Measure of business performance impact

Figure 9: The lack of links between AI model evaluation and business performance.

This research therefore attempts exploring how to develop an AI performance measure that links model task performance to business outcomes. Task and domain specific, this becomes a question of how to link technical performance of forecast models to business performance in the food supply chain. This will be answered through the design of an evaluation model artefact, aiming to meet acceptance criteria developed from the above:

Relevance: Overall Acceptance Criteria for the Artefact		
Number	**Criteria**	**Stakeholder group**
1	Enable the relative scale comparison of forecast models in order to select best performer for client use	Data Science Operations
2	Providing an absolute scale business value measure of forecast performance in a specific product and organizational context thus linking technical forecast performance to business outcomes	Data Science Operations, Sales, Product Development
3	Capture hypotheses about what product factors affect the value of a forecast	Data Science Operations, Product Development, Sales
4	Capture hypotheses about client how clients value of forecasts	Product Development, Sales
5	Enable communication of forecast value to potential clients	Sales
6	Capture hypotheses about the clients operational processes to enable validation of the captured hypotheses against internal research, internal stakeholders and direct client feedback	Product Development
7	Enable clients to understand how the performance of the host company´s forecast would impact their organization	Client
8	Support system acceptance in use by enabling users to understand forecasts	Client, Product Development

Figure 10: Artifact Acceptance Criteria

4.0 Rigor Cycle

The rigor cycle explores the literature as well as the state-of-the art within the domain (Hevner, 2007). As Information systems being socio-technical phenomenon (Mathiassen & Sørensen, 2008), the rigor cycle explores both technical elements that can form part of an evaluation model, and information systems design methods which allow the design cycle to be informed by the socio-technical context of the forecast models.

4.1 Quantifying costs of forecast errors

A fundamental feature of operational outcomes is that different quantitative decision errors business performance impacts depending on when they are made and on whether they are above or below the optimal decision point (Ivanov et al., 2017; Taha, 2013). Some authors recognize the shortcomings of purely statistical metrics, and argue that time series forecasts should rather be evaluated on the total cost of forecast errors in to specific organizational context (Kerkkänen, Korpela & Huiskonen, 2009; Lee, Cooper & Adam, 1993; Russell & Norvig, 2016; Sanders & Graman, 2009). Russel & Norvig, (2016 p.710) suggest designing domain specific loss functions to better capture the actual impacts of prediction errors, Mentzer & Moon, (2005) suggest sales forecasts should be evaluated based on its impact on business performance.

Dellino et. al., (2018) developed a decision support system for fresh food supply management where forecasts are combined with an algorithm that considers attributes such as outdating, shortage, freshness and stock. Their approach thus entailed two part model consisting of a forecasting model and an optimal decision model. Kerkkänen et. al, (2009) assess the impact of forecast errors by defining the role of the forecast in inventory management and production planning, using operational metrics, while Sanders & Graham (2009) simulate a warehouse environment in which specific cost drivers are captured and modelled. Total forecast error impacts are also decomposed into relevant subcategories. Categorisations include planning impacts, capacity impacts and inventory impacts (Kerkkänen et. al, (2009) and costs caused by either overforecasting and underforecasting (Sanders & Graham, 2009) . An artefact implementation of the evaluation model therefore model these outcomes using time series cross validation (Hyndman & Athanasopoulos, 2019; Géron, 2017).

4.2 Overforecast Loss

In standard operations, overforecasting will likely lead to excess orders or production, causing a build-up in inventory (Taha, 2017; Ivanov et. al., 2017). In the food industry, this is further complicated by the perishable nature of food products. This results in product value impairment either through product expiry or transformation (e.g. freezing) at the end of product shelf life (Gustavson et. al., 2014). The resulting inventory costs has been identified as error cost (Sanders & Graman, 2009), and standard inventory principles from management accounting and operations research can be applied to the handling of this inventory (Taha, 2017; Weetman, 2010), such as the assumption of first-in-first-out inventory costing (e.g. Weetman, 2010). The costs associated with inventory can be direct inventory storage costs, as well as the cost of obsolescence when inventory expires (Kerkkänen et. al, 2009). The capital bound up in inventory also accrues Cost of Capital , which can be measured using corporate finance practice (e.g. Brealy, Myers, Allen, 2011; Pennings et. al., (2017). The inventory level itself can also be a performance measure, providing an indication of the above cost levels (Kerkkänen et. al., 2009).

4.3 Underforecast Loss

Underforecasting errors can lead to deficits in orders and production (Taha, 2017; Ivanov et. al., 2017). Lost revenue due to lost sales has also been proposed as a cost measure related (Pennings et. al., 2017). This is a domain specific application of the Opportunity Cost concept in Microeconomics (e.g. Frank & Cartwright, 2013). Inability to deliver on orders creates the risk of contract loss, because the loss of goodwill with downstream organisations (Ivanov et. al., 2017). This however, is a qualitative psychological concept which can be hard to quantify. A standard measure of quality within the supply chain domain is also the statistical Service level, capturing the percentage of successful delivery. High service levels have been reported as successful outcomes of forecasting (Ivanov et. al.,2017; Kerkkänen et. al., 2009). Whether a missed order brings a company below a minimum acceptable service level could therefore potentially be used as a proxy for the goodwill loss incurred by such delivery failure.

4.4 Decision process models

With uncertainty associated with several aspects of the dynamic order and production processes, established tools for modelling decision under uncertainty are relevant. Watson (2017) emphasizes the use of systems to support cognitive decision making, emphasizing the information form AI systems as an input for human decision makers. In Markov Decision Processes (MDP) and Reinforcement learning (RL) , the concept entails an action being is based on a set of inputs. After the action is taken, the environment returns the outcome of that action, and the decision maker updates the estimation of how valuable that action was (Taha, 2017; Russell & Norvig, 2016; Sutton & Barto, 2018). As such, the order process can be seen as a MDP / RL model with the human estimating the value function. Agrawal et. al., (2018a, b) introduce a model is conceptually similar to a MDP, with AI predictions acting as inputs to the judgement by a human decision maker.

Rigor: Potential Elements of a forecast evaluation model artefact

Element	**Field of origin**	**Reference**
Evaluation through cross validation set of time series training data	Statistics, Forecasting, Machine Learning	Hyndman & Athanasopoulos, (2019; Géron, (2017), Goodfellow et. al., (2016), James et. al., (2017).
Forecast Error cost decomposed to cost of Overforecast and cost of Underforecast	Operations Research	Sanders & Graham, (2009);
Operational environment simulation	Operations Research	Sanders & Graman, (2009);
Inventory storage cost	Operations Research	Kerkkänen et. al, (2009);
Inventory /stock investment cost of working capital	Operations Research, Corporate Finance	Pennings, van Dalen, and van der Laan (2017); Brealy, Myers, Allen, (2011)
Inventory Levels	Operation Research	Taha, (2017); Kerkkänen et. al., 2009)
Inventory used on a First-in-First-Out basis	Management Accounting	e.g. Weetman, (2010)
Lost revenue due to lost sales aka. Opportunity cost	Operations Research, Microeconomics	Pennings, van Dalen, and van der Laan (2017); Frank & Cartwright, (2013).
Service Level	Operations Research	Pennings, van Dalen, and van der Laan (2017); Ivanov et. al., (2017)

Figure 11: Elements for the evaluation model

4.5 State -of- the-art: Designing Information Systems

To develop systems and techniques for a specific domain, it is important to thoroughly understand the domain dynamics (Kerkkänen et. al, 2009). Especially in the development of advanced knowledge systems, formal approaches should be complemented by semi-formal development approaches from socio-technical domains (Whitley, 1991). The following section therefore explores how tools from the domains of technology entrepreneurship and information systems can inform the artefact design process.

4.6 Two-Phase Iterative Development

Highly iterative development practices are used to tackle the complexities in system development (Cadin & Guérin, 20016; Sapsed & Tschang, 2014; Tschang, 2005), and are also an integral part of entrepreneurship and information systems methodologies (Blank, 2013; Checkland & Poulter, 2006; Knapp, 2016; Kautz, Madsen & Nørbjerg, 2007M Ries, 2011). Common practice in several of these methodologies is an initial stage in which the assumptions about a problem context are tested before iteration starts on a business or IT artefact design to further probe hypotheses about both problem and solution design (Blank, 2013; Checkland & Poulter, 2006; Knapp, 2016; Ries, 2011). In the Google Design Sprint methodology, ideas are iterated and tested internally with the project team before tested with clients towards the end of the process (Knapp, 2016). This informed the approach to internal and external testing in the design cycle. Also adopted was the Sprint combination of collaborative sessions and individual concept refinement (Knapp, 2016). In addition to drawing on the above design ideas, the research also allowed for creative insights stemming from the “flow” of the creative design process (Csikszentmihalyi, 1996).

Conceptual models are also used to validate hypotheses about problem- solution fit (Checkland & Poulter, 2006; Knapp, 2006). The Design cycle was therefore structured in two main phases. The first phase included the problem design and initial conceptual models probing problem – solution fit. The second entailed designing the main artefact as a minimum viable product, a prototype capturing hypotheses about the problem and solution with the aim of validating these (Ries, 2011).

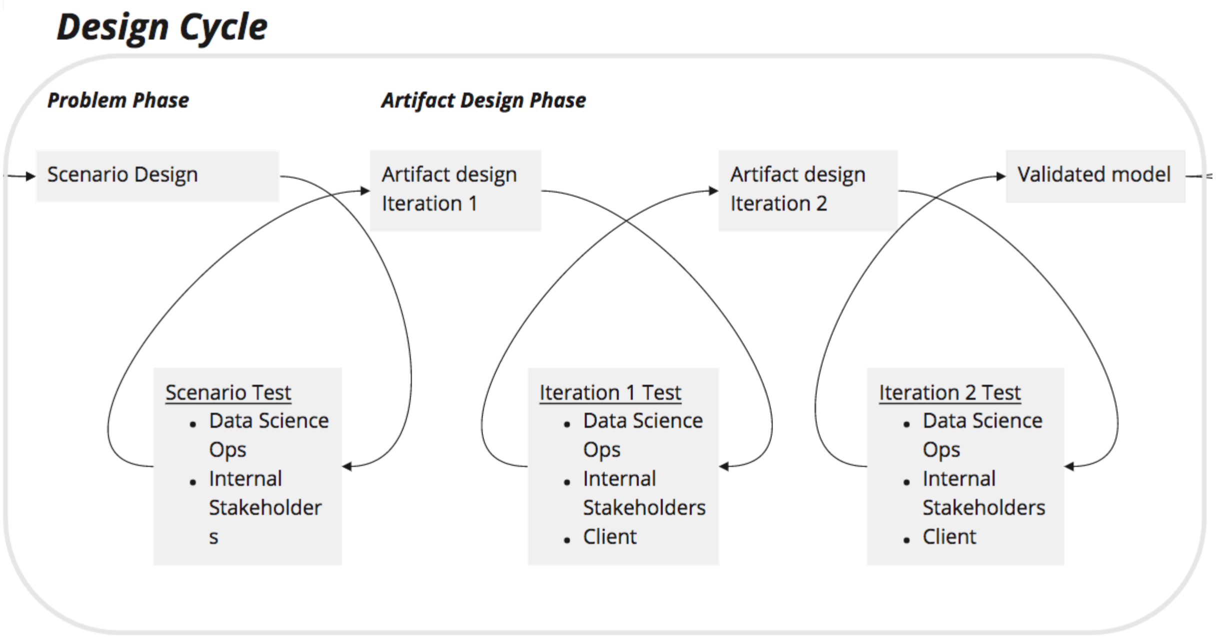


Figure 12: The Design Cycle in two phases, each cycle iterating between design and test activities.

4.7 Problem & Solution Concept Tools

Specific conceptual tools explored that were found potentially useful to the research during the rigor cycle were the Rich Picture from Soft systems Methodology (SSM) (Checkland & Poulter, 2006), empathy maps from Design Thinking (IBM, 2019), class diagrams and process flow diagrams from the Unified Modelling Language (Stevens, 2006). The iterative Lean Startup methodology also the concept of a Minimum Viable Product, which is defined as a version of the product that allows the developer to test fundamental business hypotheses. By getting feedback on actual design solutions rather than asking customers for feature specifications, It attempts to remove confirmation bias (Ries, 2011 p.94).

5.0 Design Cycle

The following section details the execution and findings of the design cycles two phases, the problem design and artifact design.

5.1 Design Cycle: Problem Design Phase

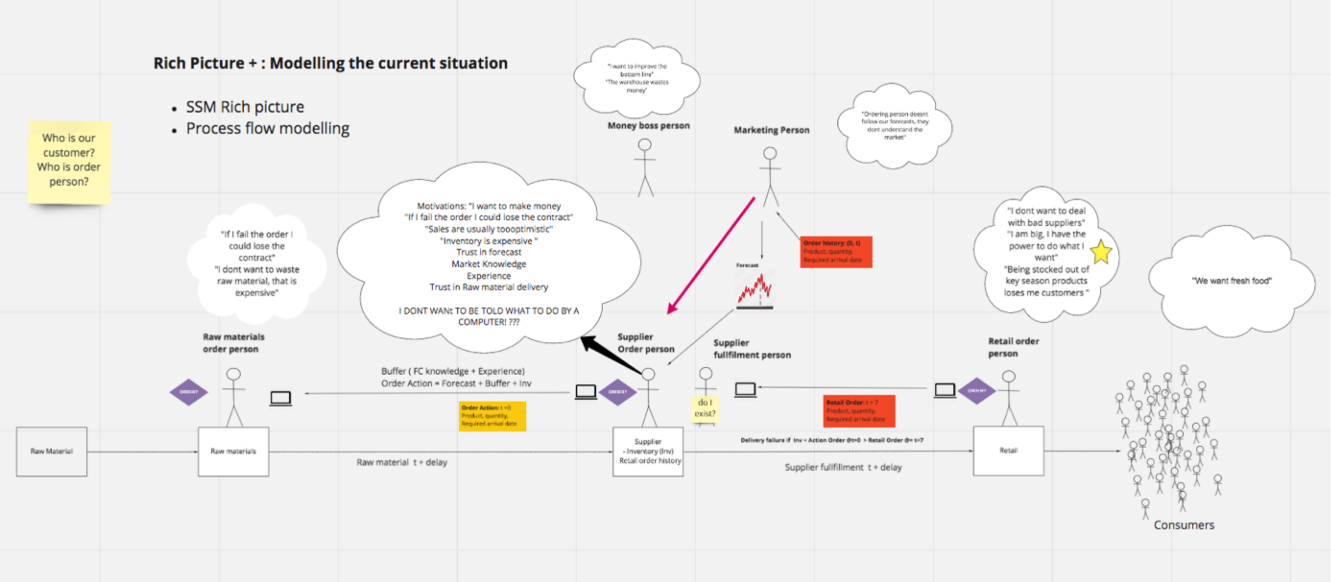
Hevner´s (2007) design cycle method prescribes that the first artefact version should be derived from the relevance and rigor cycles. In collaboration with the Data Scientist, an initial design session was held with the goal of developing artefacts to address the following : (1) Understand customer problems under status quo, (2) Understand how the host organisation´s services adds value to the food supply chain decision makers (3) Understand users interactions with the forecasting system (4) Link customer problem solving to model evaluation metrics. It drew on techniques explored in the rigor cycle including Soft Systems Methodology, Google Design Sprint, Design Thinking, Process Flow Modelling, and Reinforcement Learning. After the initial session (Appendix 3), the composite model that emerged from the workshop were broken apart and refined into individual scenario designs by the author. The session resulted in the following artefact models, which were tested through a discussion session involving the Data Scientist and senior leadership.

Problem Design Phase Scenarios

- *Scenario 1: The current status of the food supply chain and order process*
- *Scenario 2: The current status of the order decision Process*
- *Scenario 3: Process flow of a forecast evaluation Programme*
- *Scenario 4: The Order decision Process as a Markov Decision Process / Reinforcement Learning Case*

5.1.1 Scenario 1: The current status of the Food Supply Chain Order Decision process quo

Drawing on SSM techniques to capture hypotheses about the status quo of the food supply chain order decision processes. It along the guidelines for Rich picture modelling, it captured people, structure, processes and conflict as proposed by the SSM framework (Checkland & Poulter, 2006). Key actors are placed in different points of the supply chain, with communication facilitated through computerized communications technology. Drawing on insights from the relevance cycle, the model aims to capture both information and material flows along the supply chain(Taha, 2016; Ivanov et. al., 2017). Notes borrowing mathematical vocabulary from the more formalized field of operations research were used to precisely capture entity associations and timeframe hypotheses about these flows (Taha, 2016; Ivanov et. al., 2017).



Rich Picture + : Modelling the current situation
SSM Rich picture
Process flow modelling
Who is our customer? Who is order person?
"I want to improve the bottom line" "The warehouse wastes money"
Money boss person
Marketing Person
"Ordering person doesnt follow our forecasts, they dont understand the market"
"If I fail the order I could lose the contract" "I dont want to waste raw material, that is expensive"
Motivations: "I want to make money "If I fail the order I could lose the contract" "Sales are usually toooptimistic" "Inventory is expensive " Trust in forecast Market Knowledge Experience Trust in Raw material delivery I DONT WANt TO BE TOLD WHAT TO DO BY A COMPUTER! ???
Order history: (0, t) Product, quantity, Required arrival date
Forecast
"I dont want to deal with bad suppliers" "I am big, I have the power to do what I want" "Being stocked out of key season products loses me customers "
"We want fresh food"
Raw materials order person
Supplier Order person
Supplier fullfilment person
Retail order person
Buffer (FC knowledge + Experience)
Order Action = Forecast + Buffer + Inv
Order Action: t =0 Product, quantity, Required arrival date
do I exist?
Retail Order: t = 7 Product, quantity, Required arrival date
Delivery failure if Inv + Action Order @t=0 > Retail Order @= t=7
Raw Material
Raw materials
Raw material t + delay
Supplier - Inventory (Inv) Retail order history
Supplier fullfillment t + delay
Retail
Consumers

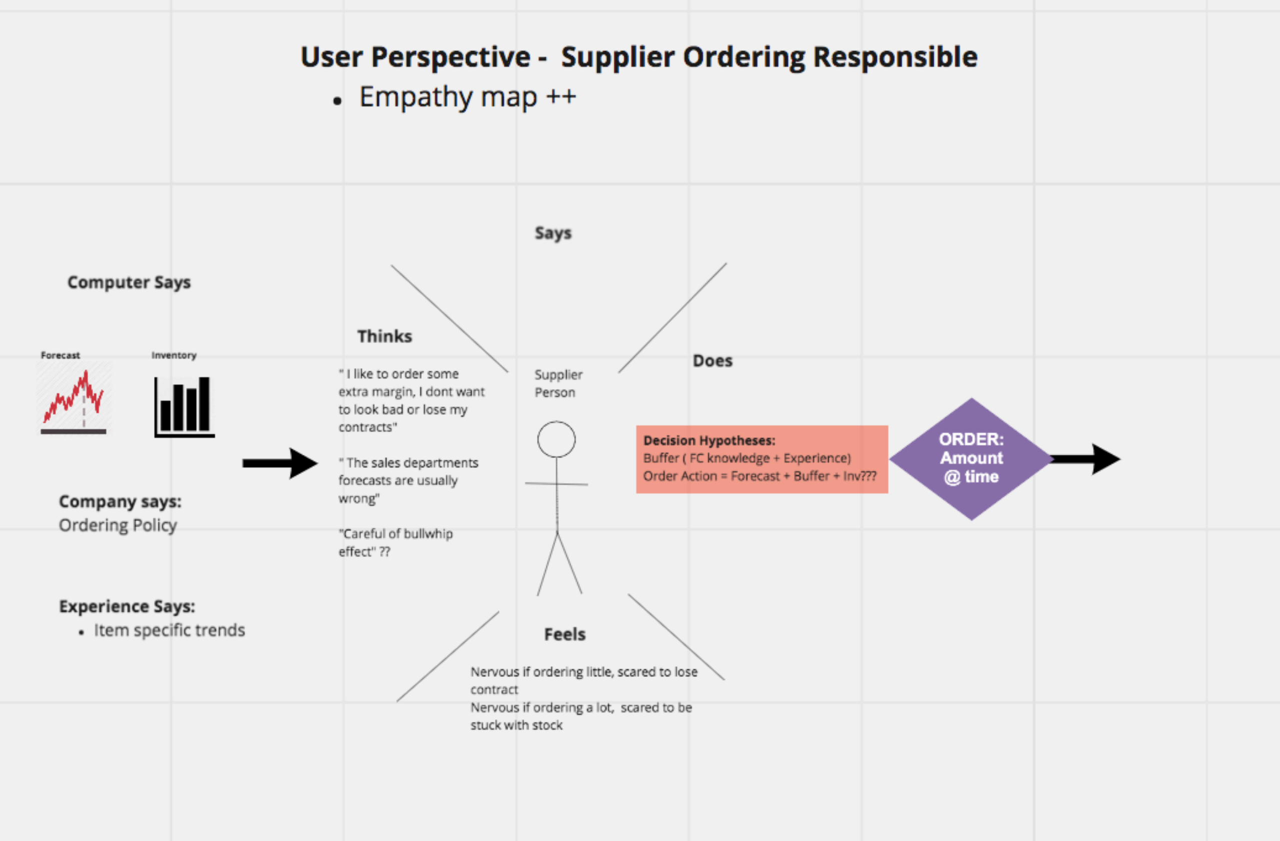
Figure 13: The current Status of the Food Supply Chain Order Decision process.

5.1.2 Scenario 1: Test

Scenario 1 (Figure 13) was tested against both the direct insights of the internal stakeholders through a workshop meeting, and against internal documentation capturing the company´s industry insights. During the workshop, feedback suggested that the model captured the overall business context of the forecast solution. Key aspects of the information flows were also validated, such as the time delays between order flows and delivery. Also validated were the different interest of the order person and money boss person, where the order professional might care mostly about operational efficiency and inventory, while the money boss person would care mostly about overall business performance outcomes. Overall, Scenario 1 was found to be sufficiently accurate to use as basis for initial artefact design along with the additional insights from the test.

5.1.3 Scenario 2: The current status of the order decision Process

Figure 14 shows the model attempting to capture the current status of the order decision process. It combines ideas form Design Thinking, Process flow models in Software Engineering and AI decision processes by Agrawal et. al. (2018a,b). It assumes several the forecast prediction is combined with inputs such as inventory levels, company policies and the order person´s context specific experience. The judgement element suggested by Agrawal et. al., (2018a,b) is decomposed using an empathy map from Design thinking. The “Does” section of the map is used to represent the action element, in which main hypotheses about the order persons internal decision rules are captured by a models section inspired by a Process flow chart (Stevens, 2006). As prescribed empathy map (IBM, 2019) judgement section attempts to capture hypotheses about how the decision maker values different outcomes of their actions, which are hypotheses that feed into the criteria for the evaluation artefact.



User Perspective - Supplier Ordering Responsible

- Empathy map ++

" I like to order some extra margin, I dont want to look bad or lose my contracts"

" The sales departments forecasts are usually wrong"

"Careful of bullwhip effect" ??

Decision Hypotheses:
Buffer (FC knowledge + Experience)
Order Action = Forecast + Buffer + Inv???

Nervous if ordering little, scared to lose contract
Nervous if ordering a lot, scared to be stuck with stock

Figure 14: Current status of the order decision process

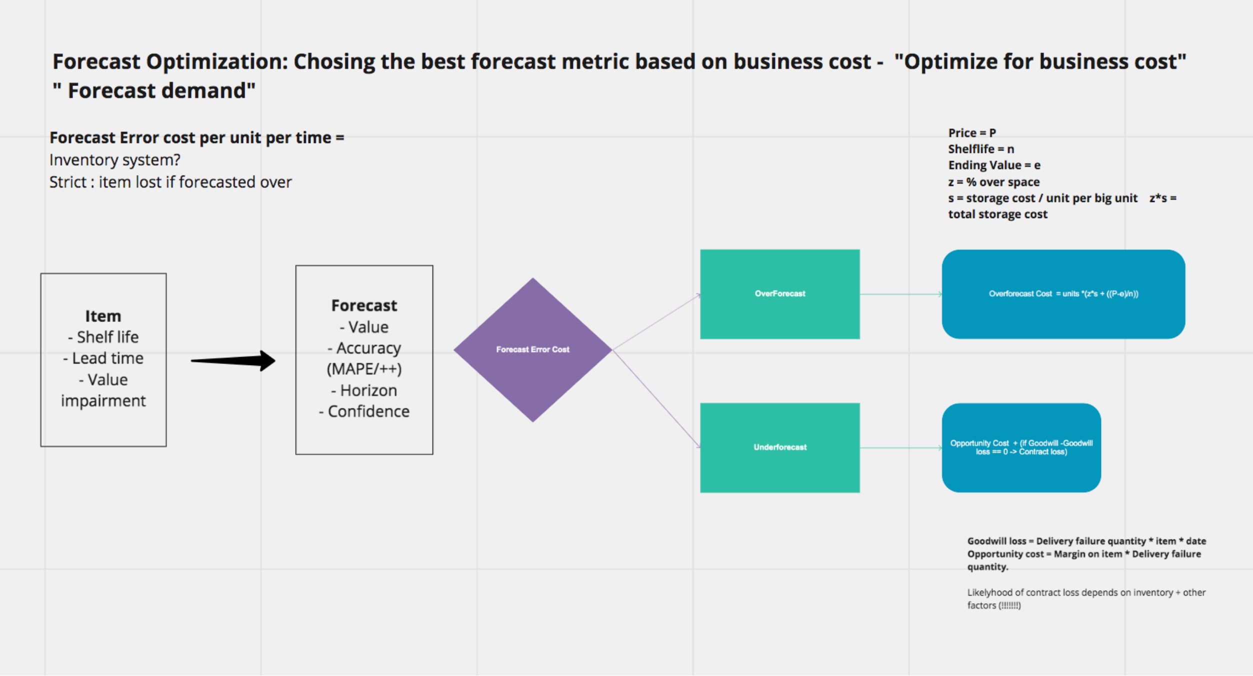
5.1.4 Scenario 2: Test

The test validated that order decisions were likely to depend on several factors, with forecast results being one of several considerations. Importantly, the risk of contract loss as captured in “thinks” and “feels” was also validated as an important consideration for order makers. The risk of being stuck with too much inventory was also validated, with both risk of contract loss and a cost model for excess inventory carried through to the artifact design phase.

5.1.5 Scenario 3: Forecast Impact on Business Value

Scenario 3 uses a conceptual model to test hypotheses about how the artefact can capture forecast evaluation criteria relevant to the clients and their operational processes. Using a process flow chart combined with text, the forecast business value evaluation captures specific hypotheses about the dynamics of different forecast outcomes, and how their losses might be evaluated by the order decision person.

Product specific characteristics such as shelf life, lead time and value impairment upon expiry feed into a forecast evaluation process. The forecast itself has attributes such as a forecast value, a statistical accuracy measure (Mean Average Percentage Error in this example), the forecast horizon and the forecast confidence. Forecast error Costs can be decomposed into two main categories, Costs due to Overforecast and costs due to Underforecast. Overforecasts lead to build up of inventory, incurring costs associated with inventory storage.



**Forecast Optimization: Chosing the best forecast metric based on business cost - "Optimize for business cost"
" Forecast demand"**

Forecast Error cost per unit per time =
Inventory system?
Strict : item lost if forecasted over

Price = P
Shelflife = n
Ending Value = e
z = % over space
**s = storage cost / unit per big unit z*s =
total storage cost**

Item
- Shelf life
- Lead time
- Value impairment

Forecast
- Value
- Accuracy (MAPE/++)
- Horizon
- Confidence

Forecast Error Cost

OverForecast

Underforecast

Overforecast Cost = units *(z*s + ((P-e)/n))

Opportunity Cost + (if Goodwill -Goodwill loss == 0 -> Contract loss)

**Goodwill loss = Delivery failure quantity * item * date
Opportunity cost = Margin on item * Delivery failure quantity.**

Likelyhood of contract loss depends on inventory + other factors (!!!!!!!)

Figure 15: Forecast Impact on Business Value.

5.1.6 Scenario 3: Test

The workshop discussion test of scenario 3 provided internal validation of the core hypotheses of the model by team members who had extensive client understanding through repeated, iterative in-depth customer problem research. The validated hypotheses included the inclusion of factors such as shelf life, lead time and value impairment, as well as the division of forecast losses into overforecast and underforecast losses. The factors included as part of the calculations were also validated internally.

The hypotheses that goodwill loss from client somehow depended on product amount, specific product and the date of the missed delivery were also confirmed. However, the test also revealed the complexity of modelling an actual goodwill loss function as it would entail an attempt at capturing psychological evaluations by downstream supplier as a function of missed order delivery characteristics. The frequency of missed orders was also proposed as a new factor in ensuing discussion, reflecting the element of patience loss from downstream organizations.

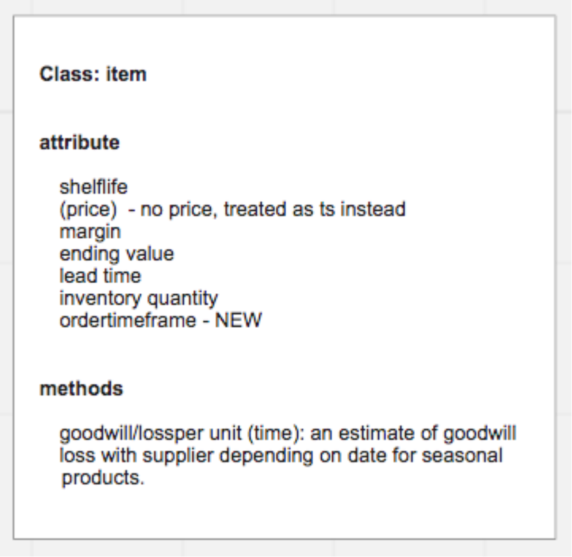
5.2 Artefact Design Phase

With the dual goal of the present design science study of both investigating how AI enabled IT artefacts add business value to food supply chain decision makers and developing an artefact ensuring that the AI enabled IT artefacts add such value, artefacts fitting this definition of a minimum viable product were designed then tested in order to meet the research objectives. The composite Minimum Viable Product artifact includes:

- ***Requirements documentation for the evaluation model consisting of:***
    - Requirements documentation for Product item Class (Figure 17)
    - Requirement documentation for the Evaluation Algorithm method taking instances of the Product item Class as input (Figure 18, Figure 21*)*
- ***Resource Artefacts for validating and iterating on the programme requirements including:***
    - Evaluatiom Model Python Implementation (Appendix 1)
    - Implementation notebooks (Figure 19, Appendix 2).

5.2.1 Evaluation model Iteration 1

The first evaluation model iteration was designed in three stages. Validated hypotheses from the problem design phase were first captured in a spreadsheet model during a joint workshop session with the Data Scientist and CTO (Appendix 4). This model was used for further iteration, while the concept was implemented in a requirements documentation using an UML class for relevant item attributes (Figure 17) and a general functionality list for the hypotheses (Figure 18). Initially, the class diagram did not include order timeframe, assuming daily forecasts would be relevant for all customers. The lowest abstraction implementation of the first iteration in a Jupyter notebook showcasing key hypotheses, including how inventory would build up with Overforecasts, and this would result in storage costs per time unit (Figure 19).



Class: item

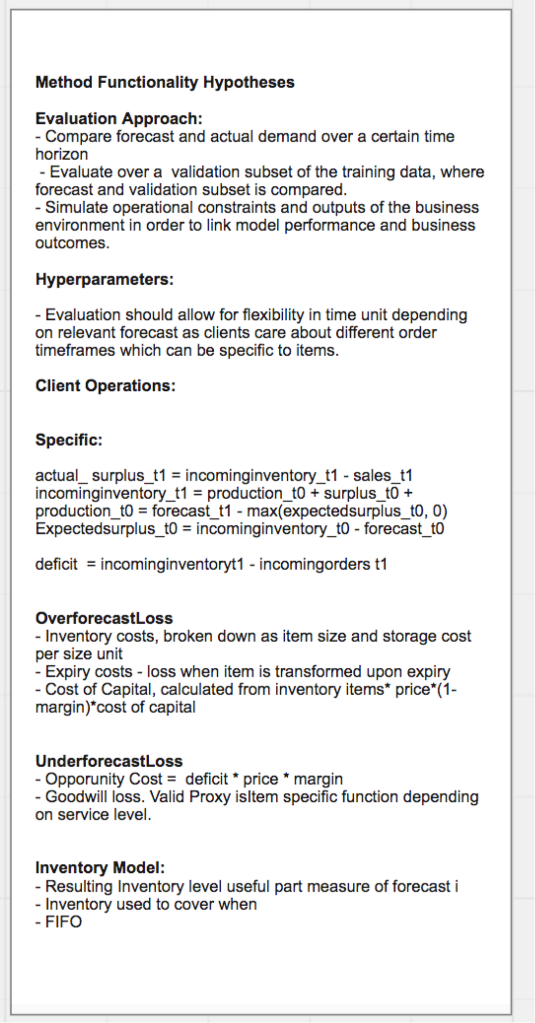
attribute

shelflife
(price) - no price, treated as ts instead
margin
ending value
lead time
inventory quantity
ordertimeframe - NEW

methods

goodwill/lossper unit (time): an estimate of goodwill loss with supplier depending on date for seasonal products.

Figure 17: Item Class Hypotheses it.1



Method Functionality Hypotheses

Evaluation Approach:
- Compare forecast and actual demand over a certain time horizon
- Evaluate over a validation subset of the training data, where forecast and validation subset is compared.
- Simulate operational constraints and outputs of the business environment in order to link model performance and business outcomes.

Hyperparameters:

- Evaluation should allow for flexibility in time unit depending on relevant forecast as clients care about different order timeframes which can be specific to items.

Client Operations:

Specific:

actual_ surplus_t1 = incominginventory_t1 - sales_t1
incominginventory_t1 = production_t0 + surplus_t0 +
production_t0 = forecast_t1 - max(expectedsurplus_t0, 0)
Expectedsurplus_t0 = incominginventory_t0 - forecast_t0

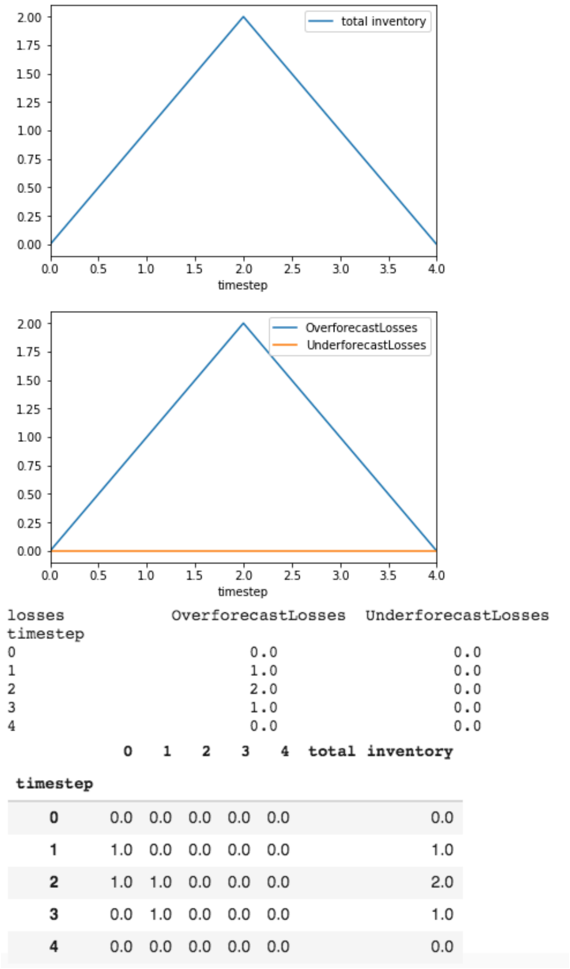
deficit = incominginventoryt1 - incomingorders t1

OverforecastLoss
- Inventory costs, broken down as item size and storage cost per size unit
- Expiry costs - loss when item is transformed upon expiry
- Cost of Capital, calculated from inventory items* price*(1-margin)*cost of capital

UnderforecastLoss
- Opporunity Cost = deficit * price * margin
- Goodwill loss. Valid Proxy isItem specific function depending on service level.

Inventory Model:
- Resulting Inventory level useful part measure of forecast i
- Inventory used to cover when
- FIFO

Figure 18: Evaluation model It. 1 Functionality Hypotheses



```
losses             OverforecastLosses  UnderforecastLosses
timestep
0                                 0.0                  0.0
1                                 1.0                  0.0
2                                 2.0                  0.0
3                                 1.0                  0.0
4                                 0.0                  0.0
```

```
             0    1    2    3    4   total inventory
timestep
    0      0.0  0.0  0.0  0.0  0.0               0.0
    1      1.0  0.0  0.0  0.0  0.0               1.0
    2      1.0  1.0  0.0  0.0  0.0               2.0
    3      0.0  1.0  0.0  0.0  0.0               1.0
    4      0.0  0.0  0.0  0.0  0.0               0.0
```

Figure 19: Evaluation Model IT.1 Simulation Notebook

5.2.2 Iteration 1 Test:

The initial model was tested in dialogue with a and revealed that the timeframe of which clients care about forecast evaluations differ, which the client in question preferring weekly forecasts to daily because of their operations processes. Thus, the first model iteration was successfully able to capture client process and forecast value hypotheses, and facilitate iteration around these. An early client test also indicated forecasts on a weekly rather than daily basis were valuable. This led to the addition of the order timeframe product item attribute, which lets the evaluation model evaluate on a period unit relevant to each product. During internal testing in data science operations feedback was also given that the characteristic of product price could be variable and need to be input data rather than product item attribute in the model (Figure 17).

5.2.3 Artefact Design Phase - Evaluation model Iteration 2

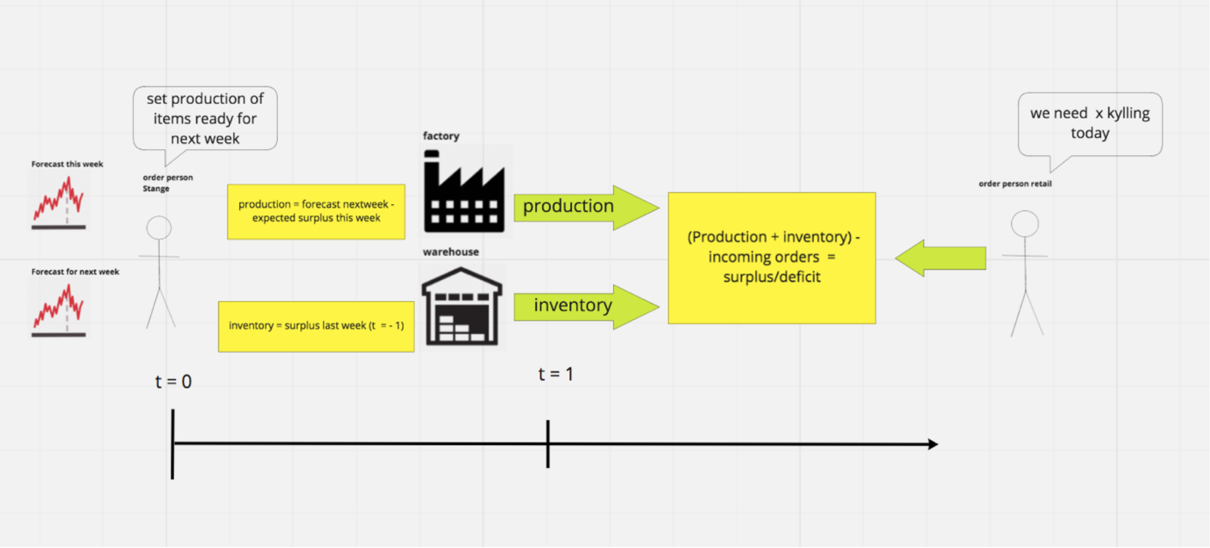

set production of items ready for next week
Forecast this week
order person Stange
Forecast for next week
production = forecast nextweek - expected surplus this week
inventory = surplus last week (t = - 1)
factory
warehouse
production
inventory
(Production + inventory) - incoming orders = surplus/deficit
we need x kylling today
order person retail
t = 0
t = 1

Figure 21: Key hypotheses about client´s operations

The second iteration of the artefact design included the complete implementation of the evaluation model, summarized in figure 21 with implementation detailed in Appendix 1. The evaluation was conducted on client data, and illustrative example on the authors own simulated data is included in appendix 2.

Design Artefact Iteration 2 - Validated Requirements for the forecast evaluation model artefact

This table covers hypotheses about what the forecast evaluation model should capture to reflect link forecast model performance and outcomes in the business context.

Element	Source	Validated	Next Iteration
Evaluation can be done by cross validation, using a subset set of time series training data and comparing forecast to validation dataset.	Rigor: Hyndman & Athanasopoulos, (2019; Géron, (2017);	Internal Client 1 Client 2 Client 3	Validate further
Simulation of operational environment can capture relevant business outcomes of the forecast accuracy, providing a business performance metric	Rigor: Sanders & Graman, (2009));	Internal Client 1 Client 2 Client 3	Validate further
The desired absolute performance metric is a monetary unit value as this is the premier metric for business performance	Rigor: Mentzer & Moon, (2005); Brealy, Myers, Allen, (2011); Relevance Cycle	Internal Client 1 Client 2	Service level more important for Client 3,
Inventory Levels are used as an indicator of forecast performance. Simulated levels allow clients to better understand the business impact of forecast	Rigor: Taha, (2017); Kerkkänen et. al., 2009); Design Cycle	Internal Client 1 Client 2 Client 3	Validate further
Expired Item quantities relevant indicator metric of forecast performance	Design Cycle	Client 3	Implement & Validate
Surplus is the leftover stock after this period. It is incoming inventory minus sales	Taha, (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Validate further

Incoming inventory is all inventory available for use this period. It is the surplus and production from last period	Taha, (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Validate further
The production last period is available to sell this period. It was set to cover the difference between the forecast for this period and the expected surplus from last period.	Taha, (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Validate further
The deficit reflects the amount which could not be delivered. It is the difference between incoming inventory and incoming orders from downstream	Taha, (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Validate further
Stock can be used to cover sales	Taha, (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Validate further
Inventory used on a First-in-First-Out basis	e.g. Weetman, (2010)	Internal	Validate further
All costs associated with individual time steps accounted for each time step	e.g. Weetman, (2010)	Internal	Validate further
Forecast Error cost decomposed into **Overforecast loss** and **Underforecast Loss**	Sanders & Graham, (2009);	Internal Client 1 Client 3	Validate further
Inventory storage cost • Inventory storage cost = item size fraction * storage cost per space unit	Kerkkänen et. al, (2009);	Internal Client 3, for frozen items	Validate further

Inventory /stock investment cost of working capital • Cost of capital = Total inventory units * production cost of inventory item *cost of capital per period	Pennings, van Dalen, and van der Laan (2017); Brealy, Myers, Allen, (2011); Weetman, (2010)	Internal Client 3, for frozen items	Validate further
Cost opportunity cost is the lost revenue due to lost sales: • Opportunity cost = deficit * product price * net profit margin.	Pennings, van Dalen, and van der Laan (2017); Frank & Cartwright, (2013).	Internal Client 1 Client 2 Client 3	Validate further
Goodwill loss can be used as a measure of the risk of contract loss. This is hard to quantify.		Internal	Validate further
Falling below a Minimum accepted Service Level can be used as a criteria for loosing contracts. This can be a proxy for goodwill loss.	Pennings, van Dalen, and van der Laan (2017); Ivanov et. al., (2017)	Internal Client 2 Client 3	Implement "Cost of required safety stock by service level"

Figure 21: Design Artefact Iteration 2 - Validated Requirements for the forecast evaluation model artefact

5.2.4 Design iteration 2: Test Feedback

5.2.5 Internal Test: Data Science Model Selection

The second iteration of the evaluation model artefact successfully met the evaluation criteria of enabling AI forecasting model comparison and selection in Data Science operations practice. As illustrated by figure 22, it is ambiguous from visual analysis which of the forecast models more accurately match the demand. Figure 23, however, shows how the evaluation model produces a clear total loss estimate based on a specific product, resulting in an absolute business value comparison between the two models. This enabled Data Science operations to make an informed model selection based on a metric that reflected the business performance impacts of each forecast.

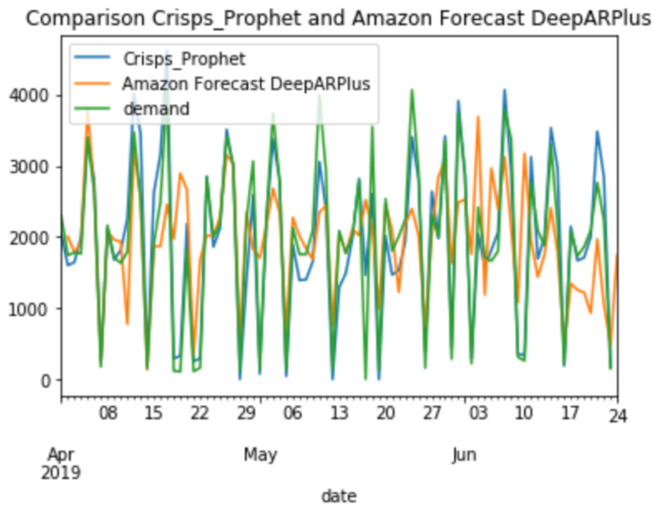
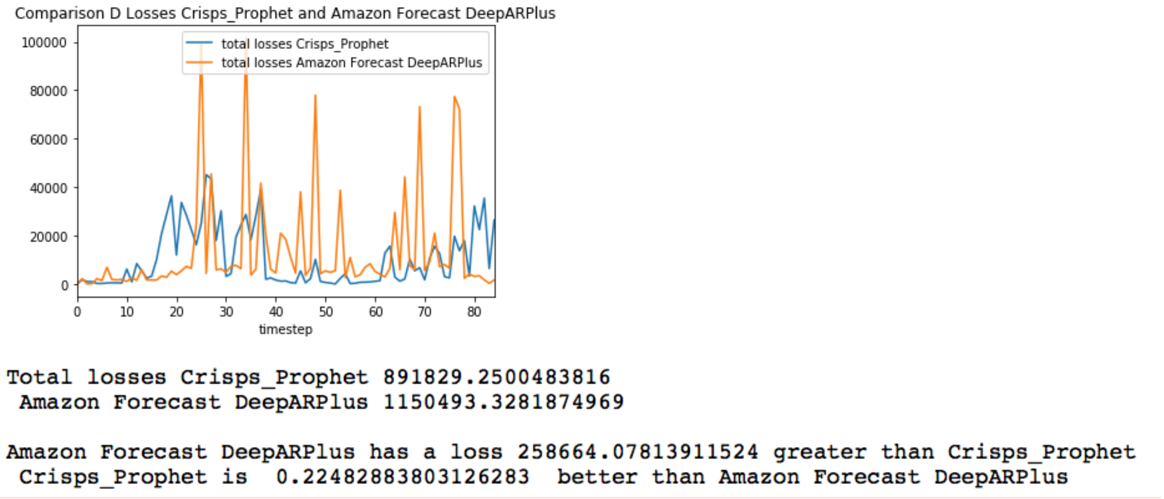


Figure 22: Model Forecast and demand comparison


Comparison D Losses Crisps_Prophet and Amazon Forecast DeepARPlus
total losses Crisps_Prophet
total losses Amazon Forecast DeepARPlus
100000
80000
60000
40000
20000
0
0
10
20
30
40
50
60
70
80
timestep

```
Total losses Crisps_Prophet 891829.2500483816
 Amazon Forecast DeepARPlus 1150493.3281874969

Amazon Forecast DeepARPlus has a loss 258664.07813911524 greater than Crisps_Prophet
 Crisps_Prophet is  0.22482883803126283  better than Amazon Forecast DeepARPlus
```

Figure 23: Model Loss Comparison

5.2.6 Internal Test: Feedback from internal stakeholders

The evaluation model was tested internally with feedback from internal stakeholders. The Internal stakeholders involved in the product development process found the model to be very valuable as a measure of model goodness, explaining that it would allow the team to iterate "much quicker and more productively" on models. The model was referred to as a "game changer", as it provided a previously non-existent link between forecast performance and potential impact on client business operations. Team members involved in the sales process also expressed that the evaluation model would be useful in proving the value of forecasting models to potential clients, providing a quantitative measure of the system value, which is unusual for SaaS products.

5.2.7 External test: Client feedback

The model was also tested in dialogue with representatives from three client organizations. All three organizations were suppliers to food retail stores delivering products in the food group meat, poultry and fish. During the client tests, the model artefact successfully enabled validation of several key hypotheses about client operations and the value of forecasts. The model also enabled a conversation in which new insights about these themes emerged, enabling further model specification for future iterations. As such, the model met the acceptance criteria specified in the relevance cycle.

5.2.8 Operations Simulation as evaluation

All three clients confirmed inventory modelling (figure 24, top graphs) as a relevant approach to evaluate how forecasts impact operations, with the representative from client 1 responding "I wish we were able to do this" upon seeing the model. Inventory modelling was as indicator of forecast quality was also validated by Client 1. Client 1 and 2 already had some evaluation practice in place, with Client 1 using instance based/ad hoc inventory monitoring. Client 2 had a spreadsheet solution in place, meaning the client had both recognized the problem linking forecast performance to business impact, and invested some resources towards developing internal solutions. This was strong validation of the evaluation model (Blank, 2013).

5.2.9 Error cost decomposition

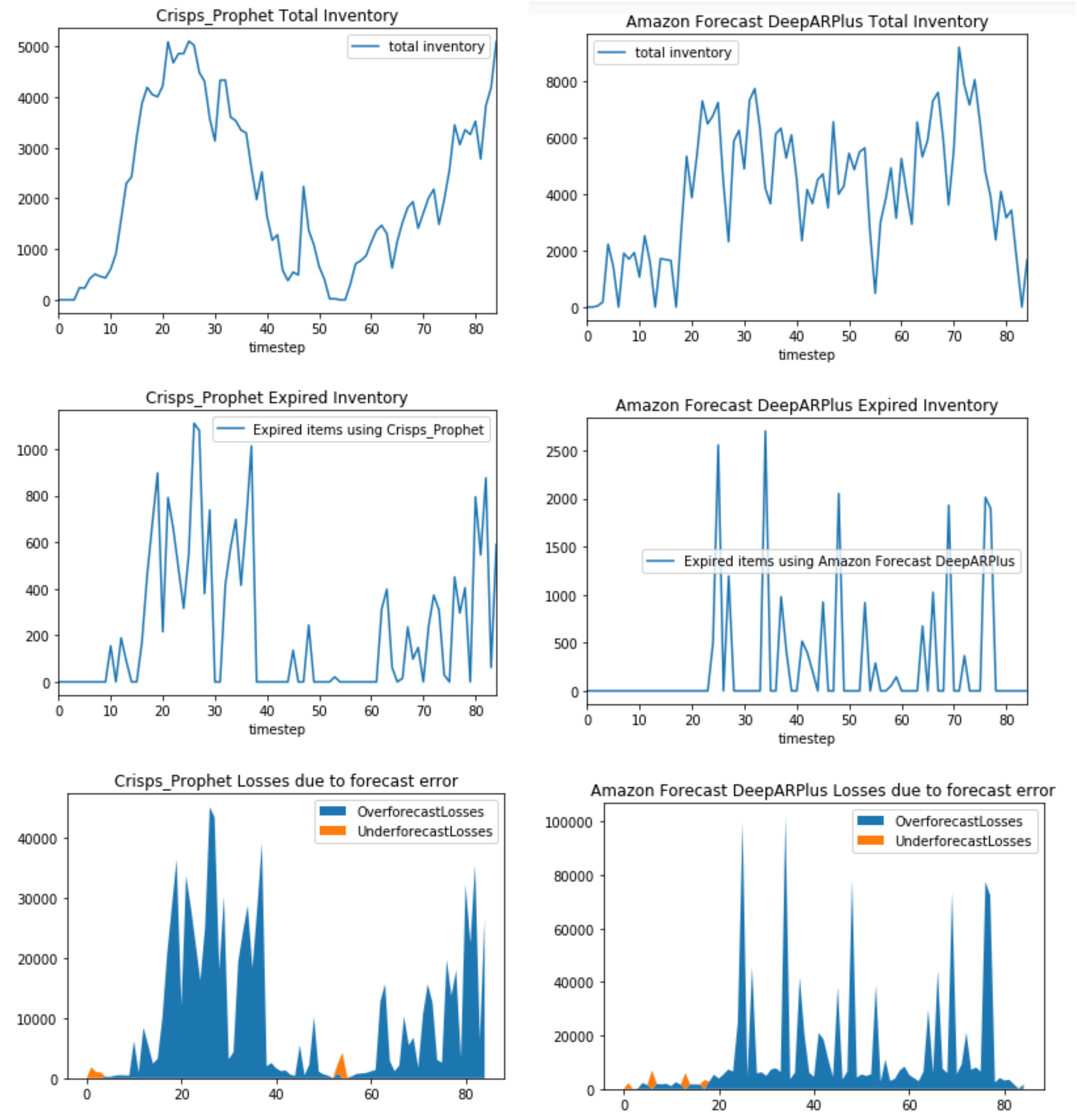
The decomposition of forecast losses into validated by all three client organisations (figure 24, bottom graphs), with storage costs, bound up capital, expiry and lost potential sales confirmed as relevant cost measures. Client 1 the client found service level a valid proxy for goodwill loss and risk of contract loss. However, client Client 2 and 3 deficits for some or all products were unacceptable, as inability to fill orders meant running the risk of losing future incoming contracts. Modelling the cost of safety stock needed to meet incoming orders under a given forecast was therefore a more realistic evaluation for these two organisations. Value impairment at the end of shelf life was confirmed as an important consideration by the clients. As these insights emerged as responses to the model, further validating the model to as a Minimum Viable Product.

5.2.10 Metric Valuation

Although a monetary comparison as evaluation metric was confirmed by all three clients, the discussions also confirmed the usefulness of inventory simulation for both and minimum safety stock. The latter two might therefore be useful inputs for the operations managers to use in their judgment based day to day operations. For Client 3, the most valuable metric was achieved service level, as they needed to comply with delivery criteria from their customer. Along with several client´s emphasis on safety stock, this suggests future iterations of the evaluation artefact should incorporate these additional metrics as outputs of the model.

5.2.11 New insights for future iterations

New insights also emerged from the client test of the evaluation model. It was suggested by Client 1 that production constraints at the time of the order decision impact the usefulness of forecasts, as the production adjustments which follow from the forecasted amount must be realistic within these constraints for the forecast to be useful. For Client 3, the raw materials for several products were related, which means a realistic forecast evaluation would need to include spillover between different product items. While the design cycle was mainly focused on capturing ways in which the business value of the forecast could be measured in a monetary amount, the client tests also revealed the emphasis given to inventory level information as a way of evaluating forecasting accuracy.



Crisps_Prophet Total Inventory

Amazon Forecast DeepARPlus Total Inventory

Crisps_Prophet Expired Inventory

Amazon Forecast DeepARPlus Expired Inventory

Crisps_Prophet Losses due to forecast error

Amazon Forecast DeepARPlus Losses due to forecast error

Figure 24: Evaluation Model Inventory modelling, Expired Item accounting and Losses breakdown

6.0 Discussion

6.1 Artifact Domain Hypotheses

The findings show that context specific machine learning model evaluations can be developed by running simulations based on the standard validation structure using context specific modelling criteria (Pennings, et. al., 2017), modelled "according to what one actually wants in the environment" (Russell & Norvig, 2016). Such Context specific AI performance evaluation model can be developed by identifying the context specific value of the AI performance and building a simulation model using standard validation techniques in machine learning where the simulation operates according the constraints specified by a certain business environment and is evaluated against the performance value identified in the earlier stage. This supports earlier findings on customized forecast evaluation models (Dellino et. al., 2018; Kerkänen et. al., 2009; Pennings, et. al., 2017; Sanders & Graman, 2009).

The classic "no free lunch" theorem within statistical machine learning states that it is impossible to know a priori which learning algorithm will perform best on any dataset, and to know which will perform best all will have to be evaluated (Wolpert,1996). Similarly, the observations in this paper can be synthesized to a "no free lunch" theorem for forecasting; That one cannot know a priori which model will have the best business valued performance on specific products, with context specific evaluations necessary to determine this. The value of multiple of evaluation outputs affirm that although the model implementation artifact acts upon data with techno rational logic captured in code, its value is highly dependent on the social use of its output metrics and its ability to communicate several aspects of potential impacts on the business environment to human decision makers. This both confirms the socio-technical nature of the evaluation model as an IT artefact (Mathiassen & Sørensen, 2008), and supports Agrawal et. al., (2018a,b)´s argument on AI prediction models as inputs to human judgement processes.

Both the monetary measure and the inventory modelling component of the artefact received feedback during testing that suggested these would support user system acceptance. This lends support to Gregor & Benbasat´s (1999) thesis that context specific justifications in system knowledge outputs can lead to greater acceptance.

6.2 Artificial Intelligence Models and Business Performance

This has implications for the training and optimization of machine learning models in specific business environments. For the optimal business outcome of AI model implementation, the present findings suggest it might be possible to create and evaluation capturing the specific domain context of the model, then directly optimize the model towards this evaluation criteria instead of relying on purely statistical loss functions or performance measures. As such, it was the first stage of an implementation of the feedback mechanism between AI predictions and decision outcomes as suggested by Agrawal et. al., (2018).

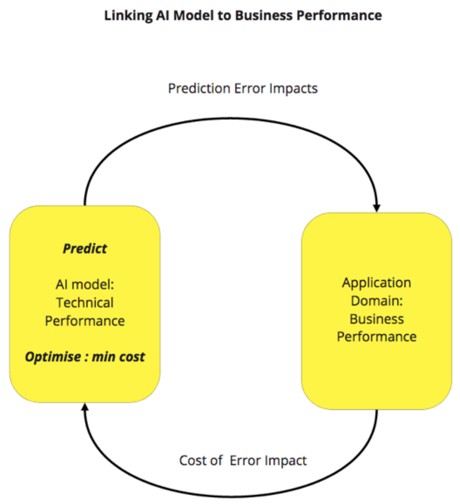

Linking AI Model to Business Performance
Prediction Error Impacts
Predict
AI model: Technical Performance
Optimise : min cost
Application Domain: Business Performance
Cost of Error Impact

Figure 25: *Successful link between AI model and business performance: Optimising AI models to fit a business context based on quantitative error impact measures.*

Conclusion: Results of Overall Acceptance Criteria			
Number	**Criteria**	**Stakeholder group**	**Result**
1	Enable the relative scale comparison of forecast models in order to select best performer for client use.	Data Science Operations	Successfully comparison between two models based on DeepARPlus and Facebook Prophet.
2	Providing an absolute scale business value measure of forecast performance in a specific product and organizational context thus linking technical forecast performance to business outcomes	Data Science Operations, Sales, Product Development	Criteria met
3	Capture hypotheses about what product factors affect the value of a forecast	Data Science Operations, Product Development, Sales	Criteria met
4	Capture hypotheses about client how clients value of forecasts	Product Development, Sales	Criteria met
5	Enable communication of forecast value to potential clients	Sales	Criteria met. Tested in final design iteration.
6	Capture hypotheses about the clients operational processes to enable validation of the captured hypotheses against internal research, internal stakeholders and direct client feedback	Product Development	Criteria met. Tested and validated in final design iteration.

7	Enable clients to understand how the performance of the host company´s forecast would impact their organization	Client	Criteria met. Validated with Client 1 and 2
8	Support system acceptance in use by enabling users to understand forecasts	Client, Product Development	Criteria met. Validated with Client 1 and 2

Figure 25: Model Acceptance Results

6.3 Artifact Acceptance Evaluation

In context evaluations through data science operations tests, internal stakeholder tests and tests with potential clients, the forecast evaluation model was able to meet the specific acceptance criteria for these groups as illustrated Figure 25. Thus, the design science study successfully attained specific goals through artifact design as prescribed by March & Smith, (1995), addressing a ‘wicked problem’ (Pries-Heje & Baskerville, 2008). By capturing and allowing iterative testing of business and solution hypotheses, the evaluation model artifact met the criteria of a Minimum Viable Product, asserting the usefulness of the concept for business solution design (Ries, 2011).

7.0 Conclusion

The present study successfully applied Hevner´s (2009) design science cycles to build an IT artefact addressing a relevant complex problem in a sociotechnical setting. Through the relevance cycle, the study identified a problem theme, research question and artifact acceptance criteria which concerned how to design a AI performance measure that links model performance to business outcomes. Potential evaluation model elements were identified in the Rigor cycle, along with information systems and entrepreneurship design frameworks.

These frameworks were applied in the Design cycle to first define the IS problem, then iteratively designing and testing an evaluation model artifact which links forecasts accuracy to business performance impacts in the food supply chain. The study successfully demonstrated how apply a design science approach to develop an AI evaluation model in a specific business context. This further validates the usefulness of the methodology in ability to jointly provide Information systems artefacts fit for a specific problem and produce academic insights through this process.

7.1 Contributions to the Literature

The present study provides an example of how the 'wicked' problem of designing an IS artifact capturing socio-technical business context variables can be addressed using design science, strengthening previous arguments about the usefulness of the methodology (Pries-Heje & Baskerville, 2008; Hevner, 2009; Hevner et. al., 2004; March & Smith, 1995). It makes contributions to the literature on how to develop domain specific AI and forecast model evaluation metrics models (Dellino et. al., 2018; Kerkänen et. al., 2009; Pennings et. al, 2017; Sanders & Graman, 2009).

7.2 Limitations and Further research

The present study has addressed the question of how to link artificial intelligence model accuracy to business performance in the context of the task demand forecasting for the application domain for food supply chain decision processing. The above findings can generalize analytically to theories in accordance with qualitative research methods (e.g. Yin, 2009). However, more research on evaluation models in the context of other prediction tasks and application domains are needed to further develop insights into the general question of AI model evaluation by business performance outcomes.

References

Agrawal, A., Gans, J. & Goldfarb, A. (2018a) Prediction Machines: The Simple Economics of Artificial Intelligence. Boston: Harvard Business Review Press

Agrawal, A., Gans, J.S & Goldfarb, A. (2018b) Exploring the Impact of Artificial Intelligence: Prediction Versus Judgement. *National Bureau of Economic Research.* Working Paper no. 24626.

Baecke, P., De Baets, S., & Vanderheyden, K.,(2017). Investigating the added value of integrating human judgement into statistical demand forecasting systems. *International Journal of Production Economic*s, 191, pp.85–96.

Biggs JR, Campion WM. (1982) The effect and cost of forecast error bias for multiple-stage production-inventory systems. *Decision Sciences*;13(4):570–84.

Blank, S. (2013). *Four Steps to the Epiphany: Successful strategies for Products that win.* 2nd edition. K & S Ranch.

Bjerknes, G., P. Ehn, & M. Kyng, ed. (1987): Computers and Democracy. *A Scandinavian Challenge*. Aldershot etc.: Avebudy.

Brynjolfsson, E. & McElheran, Kristina. (2016). The Rapid Adoption of Data-Driven Decision-Making*. American Economic Review: Papers & Proceedings*, 106(5): pp. 133–139

Brealy, A.B , Myers, S. C. & Allen, F. (2011). *Principles of Corporate Finance.* New York: McGraw-Hill

Brynjolfsson, E. & McAfee (2016) *The Second Machine Age: Work, Progress and Prosperity in a time of Brilliant Technologies.* W. W. Norton & Company

Buzby, J.C, Wells, H. F., & Hyman, J. (2014). The Estimaed Amount, Value and Calories of Postharvest Food Losses at the Retail and Consumer Levels in the Unuted States. *United States Department of Agriculture Economic Research Service. Economic Information Bulletin Number 121, February 2014.*

Changchit, C., Holsapple, C.W & Madden, D.L (2001). *Supporting managers' internal control evaluations: an expert system and experimental results.* Decision Support Systems 30 pp. 437–449

Checkland, P., and Poulter, J. 2006*. Learning for Action: A short definititive account of Soft Systems Methodology and its use for Practitioners, Teachers and Students*. Chichester, UK: John Wiley and Sons.

Crisp (2019). *Crisp Online.* https://www.gocrisp.com [Accessed 19.08.2019].

Csikszentmihalyi, M., (1996). *Creativity: Flow and Psychology of Discovery and Invention,* New York: HarperCollins,.

Demartini, M., Pinna, C., Tonelli, F., Terzi, S., Sansone, C. , Testa, C. (2018). Food industry digitalization: from challenges and trends to opportunities and solutions. *IFAC PapersOnLine,* 51(11), pp.1371–1378.

Eisenhardt, K. M., (1989). Building theories from case study research. (Special Forum on Theory Building). *Academy of Management Review*, 14(4), pp.532–550.

De Gooijer, J. G. & Hyndman, R. J. (2006), '25 years of time series forecasting', International Journal of Forecasting 22(3), 443–473

Ehn, P. (1988): Work-Oriented Design of Computer Artifacts. Stockholm: Arbetslivscentrum. Kyng, M. (1988): Designing for a dollar a day. *Office Technology and People, vol*. 4, no. 2, pp. 157-170.

Gustavsson, J., Cederberg, C., Sonesson, U., van Otterdijk, R., Meybeck, A., 2011. Global Food Losses and Food Waste. *The Food and Agriculture Organization of the United Nations*, Rome, Italy.

Géron, A. (2017). *Hands-On Machine Learning with Sci-kit Learn and Tensorflow.* Sebastopol: O´Reilly
Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning.* Cambridge: MIT Press

Greenbaum, J. & M. Kyng, ed. (1991): Design at Work: Cooperative Design of Computer Systems. Hillsdale, New Jersey: Lawrence Erlbaum.

Greenbaum, J. (1995): Windows on the Workplace: Computers, Jobs, and the Organization of Office Work in the Late Twentieth Century. New York: Cornerstone Books.

Kahneman, D, Slovic,P, Tversky, A. (Eds.) (1982) , *Judgement under uncertainty: heuristics and biases*, 3–20, Cambridge University Press, Cambridge.

Ha, Seok & Ok, 2018. Evaluation of forecasting methods in aggregate production planning: A Cumulative Absolute Forecast Error (CAFE). *Computers & Industrial Engineering,* 118, pp.329–339.

Hall, K.D., J. Guo, M. Dore, and C.C. Chow. 2009.. The Progressive Increase of Food Waste in America and its Environmental Impact. PLoS ONE, 4, 6.

Hevner, A. R. 2007. A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2), 87-92.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly,* 28(1), 75-105

Hyndman, R,J. & Athanasopoulos, G. (2019). *Forecasting: Principles & Practice.* https://otexts.com/fpp3/ [Accessed 08.08.2019].

Hyndman, R, J & Koehler, A. (2006) Another Look at measures of forecast accuracy. *International Journal of Forecasting no.* 22 pp. 679–688.

Editorial (2014). *Sustainable food supply chain management.* International Journal of Production Economics.(152) p.1-8.

IBM, (2019). Enterprise Design Thinking – Empathy Map. *IBM Online*. https://www.ibm.com/design/thinking/page/toolkit/activity/empathy-map [Accessed 19.08.2019]

Ivanov, D., Tsipoulanidis, A., Schönberger, J., (2017). *Global Supply Chain and Operations Management – A Decision Oriented Introduction to the Creation of Value.*

Jagtap,S & Rahimifard, S (2019). The digitisation of food manufacturing to reduce waste – Case study of a ready meal factory. *Waste Management*, 87, pp.387–397.

Kautz, K., Madsen, S. & Nørbjerg, J. (2007). Persistent Problems and PRactices in Information Systems Development*. Info Systems* J (2007) 17 , p. 217–239

Kerkkänen, Korpela, and Huiskonen (2009). Demand Forecasting Errors in Industrial Context: Measurement and Impacts*. International Journal of Production Economics* 118.1: 43-48. Web.

Kilimci, Zeynep Hilal et al., (2019). An Improved Demand Forecasting Model Using Deep Learning Approach and Proposed Decision Integration Strategy for Supply Chain. *Complexity*, 2019, pp.1–15.

Knapp, J. (2016). *Sprint: How to Solve Big Problems and Test New Ideas in just Five days.* London: Penguin Bantam Press

Lai, G., Yang, Yiming., Chang,Wei-Cheng & Liu, Hanxiao (2018) Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. *Arxiv Cornell Univeristy.* https://arxiv.org/pdf/1703.07015 [Accessed 17.08.2019].

La Scalia, G., Settanni, L., Micale, R. and Enea, M. (2016), “Predictive shelf life model based on RF technology for improving the management of food supply chain: a case study”, International Journal ofRF Technologies, Vol. 7 No. 1, pp. 31-42.

Lee, T.S., Cooper, F.W & Adam, EE Jr. (1993) The Effect of Forecasting Errors on the Total Cost of Operations. *Omega Volume 21, Issue 5, September 1993, Pages 541-550*

March & Smith, 1995. Design and natural science research on information technology. Decision Support Systems, 15(4), pp.251–266.

Marshall, C. & Rossman, G.B., (1995) Designing qualitative research Second., Thousand Oaks, Calif. ; London: Sage.

Mathiassen, L. & Sørensen, C. (2008). Towards a theory of organizational information services. *Journal of Information Technology.* 23 pp. 313–329

Mentzer, J.T., Moon, M.A. (Eds.), 2005. Sales Forecasting Management: A Demand Management Approach, Second ed. Sage Publications, Inc., Thousand Oaks (CA).

Makridakis, S Spiliotis, E & Assimakopoulos, Vassilios. (2018) The M4 competition: Results, findings, conclusion and way forward. *International Journal of Forecasting,* 34(4): 802 – 808, ISSN 0169-2070. doi: https://doi.org/10.1016/j.ijforecast.2018.06.001. [Accessed 12.08.2019]

McKinsey, 2017. *Artificial Intelligence. The next digital Frontier?.* McKinsey Global Institute

M4 Competition. (2019) Competitor’s Guide: Prizes and Rules. *M4 Competition Online.* https://www.m4.unic.ac.cy/wp-content/uploads/2018/03/M4-Competitors-Guide.pdf [Accessed 12.08.2019]

Mitchell, T.M. (1997) *Machine Learning.* MacGraw-Hill, New York.

Nandhakumar, Panourgias, & Scarbrough (2013) Envisioning in Computer Games Development Information Systems Research 24(4), pp. 933–955

Pennings, C.L, van Dalen, J, van der Laan, E. A.(2017). Exploiting elapsed time for managing intermittent demand for spare parts. *European Journal of Operational Research 258 (2017) 958–969 Contents.*

Petropoulosa, F., Kourentzesb, N,. Nikolopoulosc, K., & Siemsen, E. (2018). Judgemental Selection of Forecasting Models. F. *Journal of Operations Management* 60 (2018) 34–46

Pries-Heje, J., & Baskerville, R. (2008). The design theory nexus. *MIS Quarterly*, 32(4), 731-755.

Russell, S & Norvig, P. (2016). *Artificial Intelligence; A modern Approach.* Essex: Pearson.

Salinas, D., Flunkert, D & Gausthaus, J. (2019). DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *Amazon Research.Cornell University arXiv Online.* https://arxiv.org/abs/1704.04110 [Accessed 12.08.2019]

Sanders, N.R and Graman , G.A (2009) "Quantifying Costs of Forecast Errors: A Case Study of the Warehouse Environment." *Omega* 37.1 116-25. Web

Shiller, R. J. (2003) From Efficient Markets Theory to Behavioral Finance*. The Journal of Economic Perspectives*, Vol. 17, No. 1. (Winter, 2003), pp. 83-104.

Simon, H., 1996. The Sciences of Artificial, 3rd Edition, MIT Press, Cambridge, MA.

Sirikhorn, K., Neech, P., Wu, Y., Ojiako, U.,Chipulu, M., & Marshall, A. (2018) "Evaluation of Forecasting Models for Air Cargo*." The International Journal of Logistics Management* 25.3 (2014): 635-55. Web

Stevens, P. (2006). *Using UML. Software Engineering with Objects and Components.* Pearson Education Limited: Essex.

Stensgård, A. E. & Hanssen, O. J. (2016). *Food Waste In Norway: Report on Key Figures 2016. Matvett*

Stensgård, A. E., Prestrud, K., Hanssen, O .J and Callewaert, P. (2018). *Food Waste In Norway: Report on Key Figures 2015-2017. Matvett AS.* https://www.matvett.no/uploads/documents/OR.28.18-Edible-food-waste-in-Norway-Report-on-key-figures-2015-2017.pdf [Accessed 13.08.2019].

Sungil, K & Heeyoung, K (2016) K "A New Metric of Absolute Percentage Error for Intermittent Demand Forecasts." *International Journal of Forecasting* 32.3 (2016): 669-79. Web

Sun, Y. S., Guo, Y.,Mao, R & Malik. (2014) A Wind Forecast Error Cost Included OPF Model and Its Fast Algorithm*. IEEE PES General Meeting | Conference & Exposition* 2014.October (2014): 1-5. Web.

Sutton, R. S & Barto, A, G. (2018) *Reinforcement Learning. An introduction.* London: MIT Press.

Zhang, N. & Chandrasekar, P., (2017). Sparse learning of maximum likelihood model for optimization of complex loss function. Neural Computing and Applications, 28(5), pp.1057–1067.

Zhong, R., Xu, X. & Wang, L., (2017). Food supply chain management: systems, implementations, and future research. Industrial Management & Data Systems, 117(9), pp.2085–2114.

Taha, H.A (2017). *Operations Research.* Edinburgh: Pearson Education Limited

Taylor, S.J. & Letham, B.J. (2017) *Facebook. PeerJ Preprints.* https://doi.org/10.7287/peerj.preprints.3190v2 [Accessed 12.08.2019].

The Norwegian Government (2019) Format- Forebygging av Matavfall. *Regjeringen.no* https://www.regjeringen.no/globalassets/upload/lmd/vedlegg/div/faktaark_format.pdf?id=2255600 [Accessed 13.08.2019]

Tschang,T. (2007) Balancing the Tensions between Rationalization and Creativity in the Computer Games Industry. Organisation Science, Vol. 18, No 6, Innovation at and across Multiple Levels of Analysis (Nov. –Dec., 2007), pp. 989-1005.

UN Environment. (2019). Minimizing Food Waste. *UN Environment Online.* https://www.unenvironment.org/regions/north-america/regional-initiatives/minimizing-food-waste [Accessed 11.08.2019] .

U.S Department of Agriculture. (2019). Food Waste FaQs. *U.S Department of Agriculture Online.* https://www.usda.gov/foodwaste/faqs [Accessed 13.08.2019].

Yin, R.K. (2009). *Case Study Research: Design and methods. 4th Ed. Sage Publications: Los Angeles*

Watson, H. J. 2017. Preparing for the Cognitive Generation of Decision Support, *MIS Quarterly Executive,* pp. 153-169

Webber, M.E. 2012. “More Food, Less Energy,” *Scientific American*, pp. 74-79.

Weetman, P. (2010) *Financial and Management Accounting: An Introduction.* Essex: Pearson Education

Wieringa, R., (2009). Design science as nested problem solving. Proceedings of the 4th International Conference on design science research in information systems and technology, pp.1–12.

Whitley, 1991. Two Approaches to Developing Expert Systems: A Consideration of Formal and Semi-Formal Domains. *AI & Soc* (1991: 5: pp.110-127.

Wolpert, David (1996), "The Lack of A Priori Distinctions between Learning Algorithms", *Neural Computation,* pp. 1341-1390

Appendix 1: Evaluation Algorithmic Implementation

The Evaluation model was implemented as a software package in the Python programming language for use by Data Science practitioners. It was designed to integrate with the host company´s existing software tools.

```
"""

FORECAST EVALUATION PACKAGE

Classes and functions used in the monetary forecast evaluation model.

"""
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt


# Product Item class
class Item:
    """A class constructing individial product items used in the forecast evaluation algorithm.
      Stores attributes and functions asssociated with individual products.  """

    # attributes
    def __init__(self, gtin, name, shelflife, margin, storagecost, endingvalue=0.0, leadtime=0, size=1.0,
            order_time_frame='D', batch_size = 1, service_level = 0.98, cost_of_capital = 0.0, price = 0.0):

        '''
        Attributes:
        --------------------------------------
        Each item has the following attributes

        name : string, name of the item
        gting: str,
        shelflife: int, nr of days before product expires or gets transformed (eg. fresh gets frozen)
        Price: float, standard price of the product. Should only be used if no price is available as time series c
        Margin: float, margin in absolute terms ( eg. 20% = 0.2)
        Endingvalue: float, the value of the
        Size: float, fraction of storage space unit the item takes up
        StorageCost: float, the cost of storage for one storage space unit
        OrderTimeframe: str, the timeframe for orders. Complies to values in pd.resample
        cost_of_capital: float, yearly cost of capital for production. will be resampled.
        '''
        try:
            self.gtin = gtin
            self.name = name
            self.price  = price
            self.Shelflife = shelflife   #Assumption: shelflife = time before transformation
            self.Margin = margin    # Profit margin, net
            self.EndingValue = endingvalue  # if expires, value of new item atfter transform  e.g. freeze, percent.
            self.Leadtime = leadtime
            self.orderTimeframe = order_time_frame
            self.Size = size
            self.StorageCost = storagecost
            self.batch_size = batch_size  #NEW -
            self.cost_of_capital = cost_of_capital  # NEW, implement(!!)  , resampling, add to overforecastloss
            self.service_level = service_level


            print('Item ' + name + ' constructed')

        except:
            print('Item' + name + 'construction failed')

    # GoodwillLoss function : This might need to be defined per product
    def goodwill_loss(self, deficit, demand):   #missed_deliveries, timeframe,
        '''
        A function approximating the goodwill loss of a
        supplier based on number of missedDeliveries.

        Args:
        -------------
        missed_deliveries:    int, number of missed deliveries
        timeframe:            int, timeframe the goodwill loss will be evaluated over.

                              This assumes goodwill can be restored over a certain timeframe.

        '''
        # we are currently assuming this can be broken down to individualorders.
        # A longer timeframe might be more realistic.

        # Add notes, formula. Pain threshold

        performed_level = (demand - deficit)/demand   # NEW .

        if performed_level < self.service_level:
            loss = 1000000   #  Replace with contract value for customer
        else:
            loss = 0


        #lossfactor = np.zeros(1)
        #Loss = np.multiply(np.multiply(nrmissedDeliveries, volume), lossfactor)

        return loss


def construct_item(df):
    '''
    """Constructs a dictionary item class instances from pandas dataframe with item attributes. """
      This can be unpacked for eval of all products, single keys used to retrieve indvidual product items.
      Values are initialised based on the assumptions detailed below.

    Args:
    -------------
    df:  A pandas dataframe with item attributes in each row. Columns:
        -'gtin':       int, item identifier
        -'name':       string, item name
        -'shelflife':  int, number of days before expiry. AssignmentDefaults to 1 shortest possible if value is None
        -'margin':     float, absolute marging between 0.0 - 1.0. Gross margin assumed.
        -'endingvalue': float, ending value of item at expiry as % of price
        -'leadtime':   int, leadtime from order of item to delivery.
        -'storagecost': float, the storage cost at item storage facility.

    Returns:
    --------------
    product_dict:      a dictionary containing pairings of product gtin as keys and constructed items and values.

    '''

    product_dict = {}
    for column in range(len(df)):
        item_instance = Item(name='unknown' if df['names'][column] == None else df['names'][column]
                             , gtin='unknown' if df['gtin'][column] == None else df['gtin'][column]
                             , shelflife=1 if df['shelflife'][column] == None else df['shelflife'][column]
                             , margin=0.1 if df['margin'][column] == None else df['margin'][column]
                             , endingvalue=0.0 if df['endingvalue'][column] == None else df['endingvalue'][column]
                             , leadtime=1 if df['leadtime'][column] == None else df['leadtime'][column]
                             , storagecost=1.0 if df['storagecost'][column] == None else df['storagecost'][column]
                             , size=1.0 if df['size'][column] == None else df['size'][column]
                             , order_time_frame='D' if df['ordertimeframe'][column] is None else df['ordertimeframe'][
            column]
                             )

        key_string = str(df['gtin'][column])
        product_dict.update({key_string: item_instance})

    return product_dict
```

```
def evaluate_forecast(item, df,
                      timeframe=None,  #Horizon
                      data_dir=None,
                      model_name=None
                      ):
    '''
    """ Evaluates a forecast in monetary terms """

    Args:
    --------------

    df: a pandas dataframe containing the following columns
        'yhat' - forecasted timeseries values, date actual demand
        'inco'- the incoming orders from downstream clients, aka. the actual demand.

    timeframe: int, number of time units to evaluate over
    period: int, number of days in each period. Aka period = 7 would allow for weekly modelling.


    Returns:
    -----------
      loss_sum: float, the total loss from the forecast
      losses = pandas dataframe with columns
        'timestep' - int, the timestep for the row
        'OverforecastLosses' - float, the loss incurred because of overforecasting
        'UnderforecastLoss' - float, the loss incurred because of underforecasting
      inventory_record = pandas dataframe, each row contains the inventory level that timestep
        'timestep' - int, the timestep for the row
        "

    '''
    import numpy as np
    import matplotlib.pyplot as plt

    period = item.orderTimeframe

    df.to_csv('./'+data_dir + model_name + '_forecast.csv')


    # INITIALISE ALL
    print(period)


    try:
        forecast1 = forecast1.set_index("ds")
        forecast1['ds']=forecast1.index
    except:
        None

    try:
        forecast2 =forecast2.set_index("ds")
        forecast1['ds'] = forecast1.index
    except:
        None

    #CHANGE TS FOR EVALUATION TIMEFRAME AND SET THE DIVISOR FOR CORRECTLY CALCULATING COST OF CAPITAL
    if period == 'D':

        df_rs = df
        capital_time_divisor = 365

    elif period == 'W':
        df_rs = df.resample('W-MON', label='left', closed='left').sum() #set_index('ds').
        capital_time_divisor = 52

    elif period == 'M':
        df_rs = df.resample('M', label='left', closed='left').sum()   #.set_index('ds')
        capital_time_divisor= 12

    forecast = df_rs['yhat']
```

```
incomingorders = df_rs['inco']


# Set timeframe to forecast length by default
if timeframe == None:
    timeframe = len(forecast)
print('timeframe',timeframe)

# Set forecast inventory to zero at start of evaluation timeframe    # incominginventory = np.array([0]) - this is pipeline

nrmissedDeliveries = 0

# Initialise loss variables
surplus = 0
over_forecast_losses = np.array([0])
under_forecast_losses = np.array([0])
loss_sum = 0

pipeline = np.array(np.zeros(item.Shelflife))
pipeline_matrix = np.array([pipeline])

incominginventory_t0 = np.sum(pipeline)
surplus = 0
total_deficit = 0
total_demand = 0
expiration_list = [0.0]
deficit_list = []

# incomingProduction = forecast[0]

# Calculate variables for each timestep
for i in range(timeframe):
    #print('i',i)

    '''
    How is timeline specified? default: nr days from last prediction.

    initialindex = -n
    timeindex = -timeframe + i

    '''
    # print(i)

    # Run Surplus pipeline calculation

    # Forecasted for this timestep, and next

    try:
        forecast_t0 = forecast[i - 1]

    except:
        forecast_t0 = 0
        # print('No forecast for t0')

    try:
        forecast_t1 = forecast[i]

    except:

        forecast_t1 = np.zeros(1)
        print('End of forecast at step' + str(i) + ' forecast_tplus1 set to zero')

    incomingorders_t1 = incomingorders[i]


    # Remember leadtime,

    surplus_t0 = surplus

    expectedsurplus_t0 = incominginventory_t0 - forecast_t0
    # print('expected surplus t0',expectedsurplus_t0, 'forecast t1', forecast_t1)
```

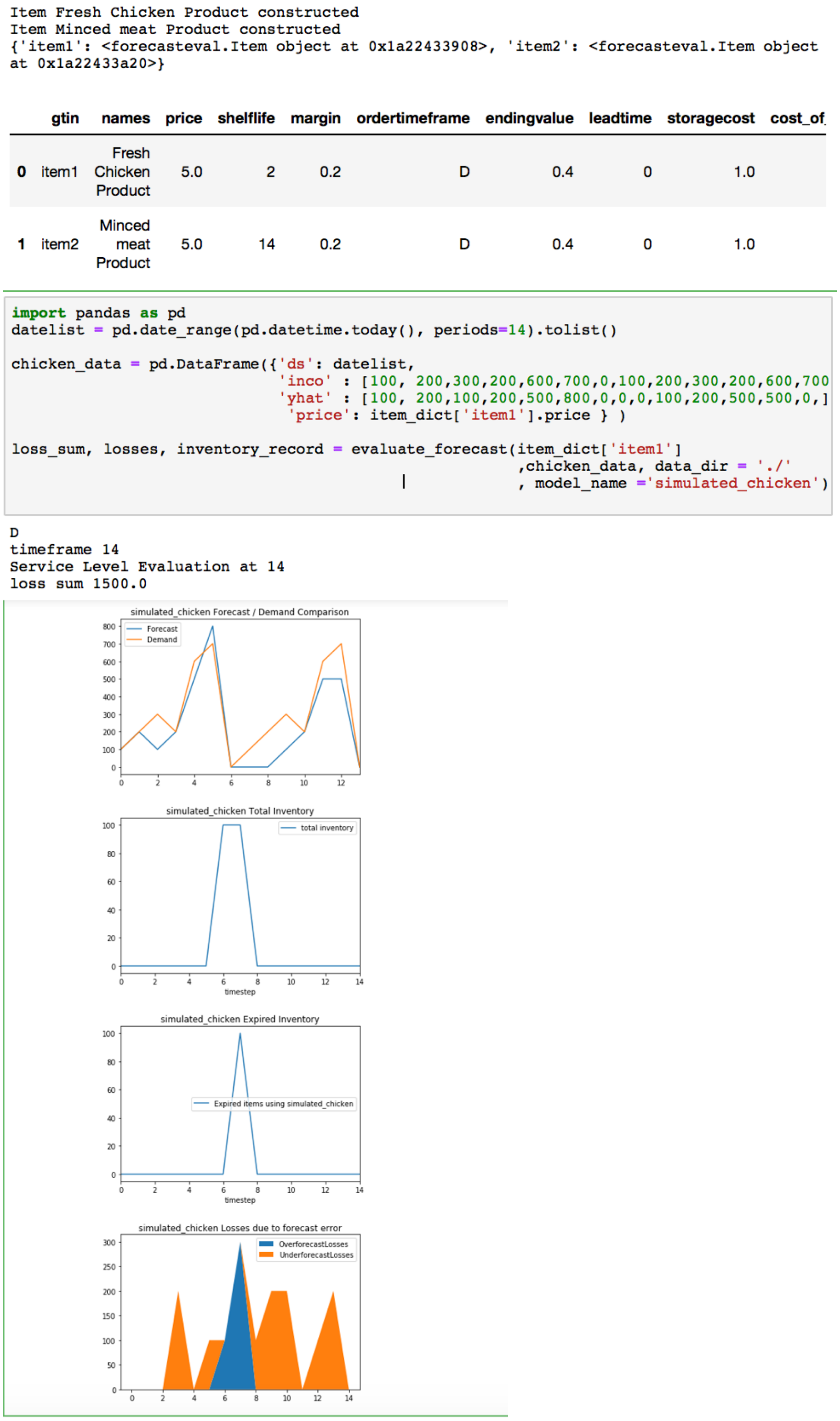
```
production_t0 = forecast_t1 - max(expectedsurplus_t0, 0)  # np.zeros(1)))
incominginventory_t1 = production_t0 + surplus_t0
# print('prod t0', production_t0)
# print('incoming inventory t1',incominginventory_t1,'incoming orders', incomingorders_t1)
actualsales_t1 = min(incominginventory_t1, incomingorders_t1)

# This is different from surplus.
# Surplus is takes production and inventory into account. deltaforecast is purely the absolute forecast error.
absoluteforecasterror_t = forecast_t1 - incomingorders_t1

# Calculate surplus and deficits
# print('inv_t1', incominginventory_t1, 'actualsales t1',actualsales_t1 )
surplus = incominginventory_t1 - actualsales_t1  # t1
# print('surplus', surplus)
deficit = incomingorders_t1 - actualsales_t1  # np.max(np.subtract(incomingorders_t1, actualsales_t1),0)
# print('deficit', deficit)

# Incoming production: Needed production delayed by the leadtime
'''
neededProduction = max(forecast_t1-incominginventory+forecast_t, 0)
incomingProduction = neededProduction
'''

# Move pipeline

# print('Old pipeline', pipeline)

surplus_from_t1 = max(surplus - np.sum(pipeline), 0)

pipeline = np.append(surplus_from_t1, pipeline[:-1])

# Adjust for used inv.
# Calculate inv usage array:

inv_needed = incomingorders_t1 - production_t0  # np.sum(pipeline)  #incominginventory_t1
# print('inv_needed',inv_needed)

# FIFO Accounting
# while inv_needed != 0:

# If there is a deficit, cover the deficit with the inventory from overforecasts using the FIFO inventory principle

if inv_needed > 0:
    # print('needed',inv_needed, 'len pipeline', len(pipeline))
    for k in reversed(range(len(pipeline) - 1)):

        # print(i)
        #print('needed',inv_needed)

        # Empty inventory spot
        # print('pipeline i', pipeline[i])
        used = min(inv_needed, pipeline[
            k])  # min(max(0,inv_needed-pipeline[-i]), pipeline[-i])   #   min(pipeline[-i],inv_needed )  #   max(min(inv_needed,pipeline[-i]),0)
        # print('used', used)

        pipeline[k] = pipeline[k] - used
        # print('pipeline',pipeline[i])

        inv_needed = inv_needed - used
        # print('inv_needed', inv_needed)
        # exit if zero
        if inv_needed == 0: break

# print('New pipeline', pipeline)
uncovered_deficit = max(inv_needed, 0)



# print(uncovered_deficit)
```

```
        # Calculate underforecast loss
        #print(df['price'][i])

        opportunity_cost_peritem = df['price'][i]* item.Margin


        under_forecast_loss = uncovered_deficit * opportunity_cost_peritem  #

        # Needed Safety stock add to deficit list
        '''
        deficit_list.append(uncovered_deficit)
        '''



        # Implement goodwilloss per service level period

        if (i+1) % timeframe == 0:    #timeframe:
            print('Service Level Evaluation at', timeframe)

            total_demand += incomingorders_t1
            total_deficit += uncovered_deficit
            good_will_loss = item.goodwill_loss(deficit=total_deficit, demand=total_demand)







        #print('ufl', under_forecast_loss, under_forecast_losses)
        under_forecast_losses = np.append(under_forecast_losses, under_forecast_loss)
        # Calculate overforecast loss

        # Calculate loss on storage of pipeline, incl current surplus
        item_storage_cost = item.Size * item.StorageCost  # currently storage cost
        inventory_loss = np.sum(np.multiply(item_storage_cost, pipeline[:-1]))

        period_cost_of_capital = item.cost_of_capital / capital_time_divisor      #   Add datetime implementation (period_factor)
        capital_cost =  np.sum(pipeline[:-1])*df['price'][i]*(1-item.Margin)*(1+period_cost_of_capital)    #Bound up

        # Calculate loss from item expiry write off
        expired = pipeline[-1]
        valueLossitem =  df['price'][i] - ( df['price'][i] * item.EndingValue)
        expiration_loss = np.multiply(expired, valueLossitem)
        expiration_list.append(expired)

        over_forecast_loss = np.add(expiration_loss.astype(np.float64), inventory_loss.astype(np.float64))

        # Append to overforecast array
        over_forecast_losses = np.append(over_forecast_losses, over_forecast_loss)

        # print('matrix', pipeline_matrix,'pipeline' ,pipeline)
        pipeline_matrix = np.append(pipeline_matrix, [pipeline], axis=0)
        # print('matrix app', pipeline_matrix)

        # Update incoming inventory
        incominginventory = np.add(np.sum(pipeline), production_t0)


    # Append to underforecast loss array
    #print('ufl', under_forecast_losses)
    # Sum up total loss
    loss_sum = np.sum(over_forecast_losses) + np.sum(under_forecast_losses) #+good_will_loss

    # Create Losses Dataframe
    losses = pd.DataFrame(
        {'timestep': list(range(0, len(over_forecast_losses))), 'OverforecastLosses': over_forecast_losses,
         'UnderforecastLosses': under_forecast_losses})
    losses = losses.set_index('timestep')
    losses['total'] = losses.sum(axis=1)

    # Creat Inventory Dataframe
    #DEBUG COMMENT:  print(expiration_list, len(expiration_list), len(pipeline_matrix))
    inventory_record = pd.DataFrame(data=pipeline_matrix)
    inventory_record['timestep'] = list(range(0, len(pipeline_matrix)))
    inventory_record['total inventory'] = inventory_record.iloc[:, :-1].sum(axis=1)
    inventory_record['expired'] = expiration_list
    inventory_record = inventory_record.set_index('timestep')

    #Calculate cost of safety stock
    '''
    min_safety_stock = max(abs(deficit_list))
    safety_stock_cost = min_safety_stock* #cost of stock
    '''


    # Print all variables
    # print("Timestep" +str(i)+"Success", "Surplus "+ str(surplus) , "deficit "+str(deficit), "pipeline" +str(pipeline))
    # Print pipeline matrix to csv/excel.

    # print("LossSum", loss_sum,"Losses ",losses,
    #   'Inventory Record', inventory_record)

    print('loss sum', loss_sum)

    # EVALUATION PLOTS
    import matplotlib.pyplot as plt

    #Plot absolute forecast comparison

    forecast.plot(label = 'Forecast')
    incomingorders.plot(label = 'Demand')
    plt.title(model_name+' Forecast / Demand Comparison')
    plt.legend()
    plt.show()



    # Plot inventory level
    inventory_record['total inventory'].plot()  #
    plt.title(model_name + ' Total Inventory')
    plt.legend()
    plt.show()


    #Plot Expiration list
    inventory_record['expired'].plot(label = 'Expired items using '+ model_name)
    plt.title(model_name+ ' Expired Inventory')
    plt.legend()
    plt.show()

    # Plot average inventory age

    # Plot total loss, with breakdown of over and underforecastloss

    plt.stackplot(losses.index, [losses['OverforecastLosses'], losses['UnderforecastLosses']],
                  labels=['OverforecastLosses', 'UnderforecastLosses'])
    plt.title(model_name + ' Losses due to forecast error')
    plt.legend()
    plt.show()

    # print('losses', losses, 'inventory_record', inventory_record)
    datadir = '/' if data_dir == None  else data_dir
    losses.to_csv(datadir + model_name + '_losses.csv')
    inventory_record.to_csv(datadir + model_name + '_inventory_record.csv')

    # Return total loss, under / overforecast losses and inventory record
    return loss_sum, losses, inventory_record


def compare_forecast_eval(item, forecast1, forecast2, demand, names, output_dir=''):
    """
```

```
Wrapper around evaluate_forecast that plots the absolute forecast difference, runs the evaluation
for two forecasts.

Args:
------------
item: Item, class instance for the product to be evaluated
forecast1:    pd.DataFrame with ds, y for forecaster 1
forecast2:    pd.DataFrame with ds, y for forecaster 2
demand:       pd.DataFrame with ds, y,price for demand ts.
names:        list of names for the two forecasts


returns:
----------

"""
import matplotlib.pyplot as plt

# Plot forecast and demand comparison
fc1_name = names[0]
fc2_name = names[1]


try:
    forecast1 = forecast1.set_index("ds")
    forecast1['ds']=forecast1.index
except:
    None

try:
    forecast2 =forecast2.set_index("ds")
    forecast2['ds'] = forecast2.index
except:
    None



if item.orderTimeframe == 'W':
    sampleperiod = 'W-MON'
else:
    sampleperiod = item.orderTimeframe

#Resample to show forecast comparison on right ordertimeframe
forecast1['yhat'].resample(sampleperiod, how = 'sum', label='left', closed='left').plot(label=fc1_name)
forecast2['yhat'].resample(sampleperiod, how = 'sum',label='left', closed='left').plot(label=fc2_name)  #
    # forecast1['yhat'].plot(label = fc1_name)
    # forecast2['yhat'].plot(label = fc2_name)

demand['y'].resample(sampleperiod, how = 'sum', label='left', closed='left').plot(label='demand')
plt.title('Comparison ' + fc1_name + ' and ' + fc2_name)
plt.legend()
plt.show()

product_item = item

#Run Eval algoritm for both forecasts
forecast1['inco'] = demand['y'].values
forecast1['price'] = demand['price'].values
forecast2['inco'] = demand['y'].values
forecast2['price'] = demand['price'].values

total_loss_fc1, losses_fc1, inventory_fc1 = evaluate_forecast(item=product_item,
                                                               df=forecast1,
                                                               timeframe=None,
                                                               data_dir=output_dir,
                                                               model_name=fc1_name)

total_loss_fc2, losses_fc2, inventory_fc2 = evaluate_forecast(item=product_item,
                                                               df=forecast2,
                                                               timeframe=None,
                                                               data_dir=output_dir,
                              model_name=fc2_name)

losses_fc1['total'].plot(label='total losses ' + fc1_name)
losses_fc2['total'].plot(label='total losses ' + fc2_name)
plt.title('Comparison '+str(item.orderTimeframe)+' Losses ' + fc1_name + ' and ' + fc2_name)
plt.legend()
plt.show()

#Calculate loss difference
loss_dif = total_loss_fc1 - total_loss_fc2


print('Total losses',fc1_name, total_loss_fc1,"\n", fc2_name, total_loss_fc2)

if loss_dif == 0:
  print('Total loss '+fc1_name+' and '+fc2_name+'are equal')

elif loss_dif > 0:
  fc2_better =  abs(total_loss_fc1 - total_loss_fc2)/total_loss_fc1

  print("\n",fc1_name+' has a loss',abs(loss_dif), 'greater than '+ fc2_name,"\n",
      fc2_name+' is ',fc2_better,'better than'+fc1_name)


else:
  fc1_better = abs(total_loss_fc1 - total_loss_fc2)/total_loss_fc2
  print("\n"+fc2_name+' has a loss',abs(loss_dif), 'greater than '+ fc1_name,"\n",
      fc1_name+' is ',fc1_better,' better than',fc2_name)


return total_loss_fc1, total_loss_fc2
```

Appendix 2: Example Client Simulation of Evaluation Algorithm

As the client specific data is anonymized, the following appendix illustrates an equivalent use case demonstration as presented during the final design cycle iteration, using data created to illustrate typical issues encountered in forecasting for the food industry.



```
Item Fresh Chicken Product constructed
Item Minced meat Product constructed
{'item1': <forecasteval.Item object at 0x1a22433908>, 'item2': <forecasteval.Item object
at 0x1a22433a20>}
```

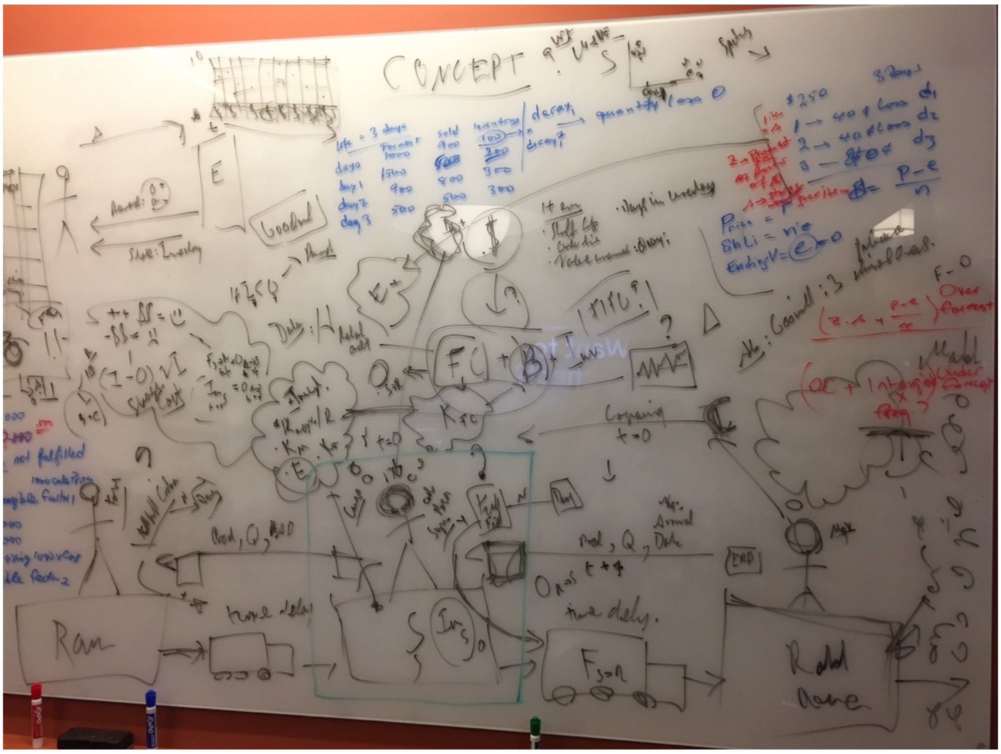
	gtin	names	price	shelflife	margin	ordertimeframe	endingvalue	leadtime	storagecost	cost_of
0	item1	Fresh Chicken Product	5.0	2	0.2	D	0.4	0	1.0	
1	item2	Minced meat Product	5.0	14	0.2	D	0.4	0	1.0	

```
import pandas as pd
datelist = pd.date_range(pd.datetime.today(), periods=14).tolist()

chicken_data = pd.DataFrame({'ds': datelist,
                             'inco' : [100, 200,300,200,600,700,0,100,200,300,200,600,700
                             'yhat' : [100, 200,100,200,500,800,0,0,0,100,200,500,500,0,]
                              'price': item_dict['item1'].price } )

loss_sum, losses, inventory_record = evaluate_forecast(item_dict['item1']
                                                       ,chicken_data, data_dir = './'
                                          |            , model_name ='simulated_chicken')
```

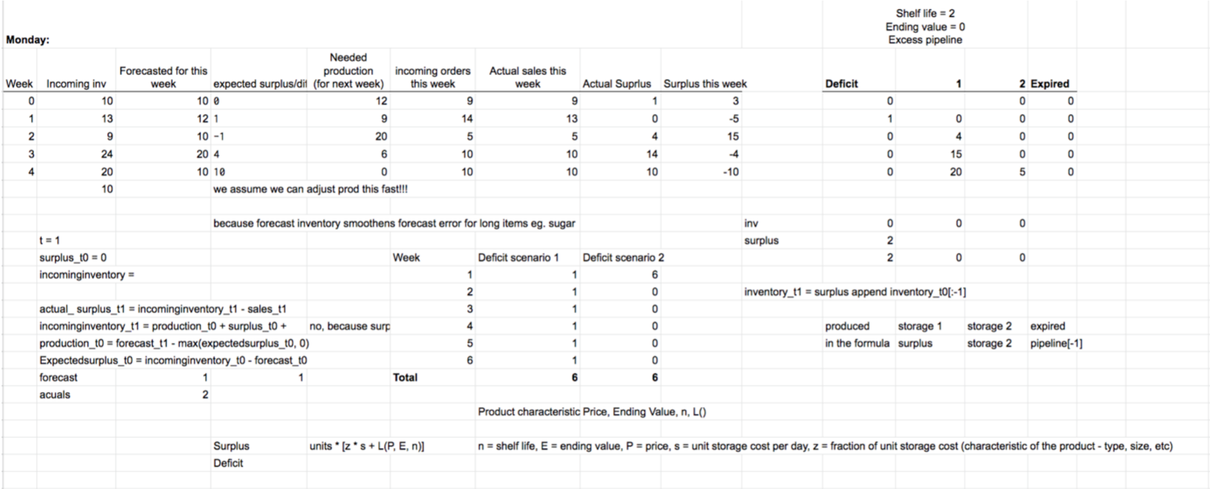
```
D
timeframe 14
Service Level Evaluation at 14
loss sum 1500.0
```

simulated_chicken Forecast / Demand Comparison
Forecast
Demand
800
700
600
500
400
300
200
100
0
0
2
4
6
8
10
12
simulated_chicken Total Inventory
total inventory
100
80
60
40
20
0
0
2
4
6
8
10
12
14
timestep
simulated_chicken Expired Inventory
Expired items using simulated_chicken
100
80
60
40
20
0
0
2
4
6
8
10
12
14
timestep
simulated_chicken Losses due to forecast error
OverforecastLosses
UnderforecastLosses
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14

Appendix 3: Models from the Creative Design Workshop



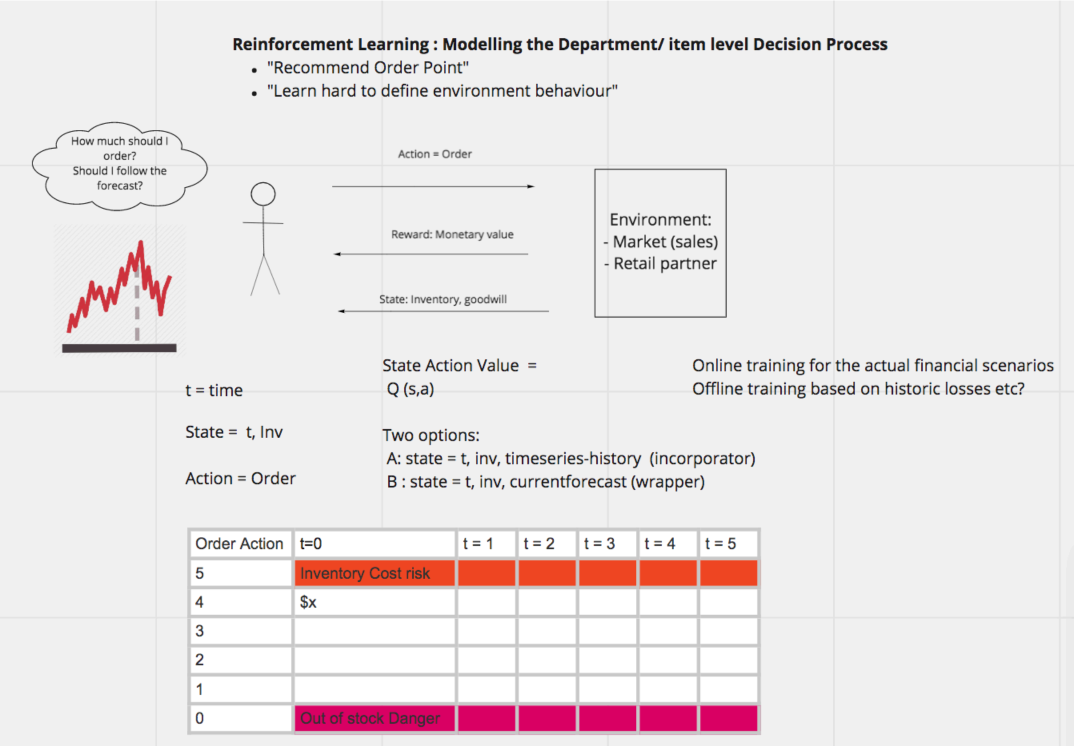
Appendix 4: Spreadsheet Concept Design



										Shelf life = 2 Ending value = 0 Excess pipeline			
Monday:													
Week	Incoming inv	Forecasted for this week	expected surplus/dił	Needed production (for next week)	incoming orders this week	Actual sales this week	Actual Suprlus	Surplus this week		**Deficit**	**1**	**2**	**Expired**
0	10	10	0	12	9	9	1	3		0		0	0
1	13	12	1	9	14	13	0	-5		1	0	0	0
2	9	10	-1	20	5	5	4	15		0	4	0	0
3	24	20	4	6	10	10	14	-4		0	15	0	0
4	20	10	10	0	10	10	10	-10		0	20	5	0
	10		we assume we can adjust prod this fast!!!										
			because forecast inventory smoothens forecast error for long items eg. sugar						inv	0	0	0	
	t = 1								surplus	2			
	surplus_t0 = 0				Week	Deficit scenario 1	Deficit scenario 2			2	0	0	
	incominginventory =				1	1	6						
					2	1	0		inventory_t1 = surplus append inventory_t0[:-1]				
	actual_ surplus_t1 = incominginventory_t1 - sales_t1				3	1	0						
	incominginventory_t1 = production_t0 + surplus_t0 +			no, because surp	4	1	0			produced	storage 1	storage 2	expired
	production_t0 = forecast_t1 - max(expectedsurplus_t0, 0)				5	1	0			in the formula	surplus	storage 2	pipeline[-1]
	Expectedsurplus_t0 = incominginventory_t0 - forecast_t0				6	1	0						
	forecast	1	1		**Total**	**6**	**6**						
	acuals	2											
						Product characteristic	Price, Ending Value, n, L()						
			Surplus	units * [z * s + L(P, E, n)]		n = shelf life, E = ending value, P = price, s = unit storage cost per day, z = fraction of unit storage cost (characteristic of the product - type, size, etc)							
			Deficit										

Appendix 5: Scenario 4: Order Decision as a Markov Decision / Reinforcement Learning Process

Conceptualising the food supply order decision process as a Reinforcement Learning Problem to capture hypotheses about factors decision makers consider when making an order decision based on a forecast.



Reinforcement Learning : Modelling the Department/ item level Decision Process

- "Recommend Order Point"
- "Learn hard to define environment behaviour"

t = time

State = t, Inv

Action = Order

State Action Value = Q (s,a)

Online training for the actual financial scenarios
Offline training based on historic losses etc?

Two options:
A: state = t, inv, timeseries-history (incorporator)
B : state = t, inv, currentforecast (wrapper)

Order Action	t=0	t = 1	t = 2	t = 3	t = 4	t = 5
5	Inventory Cost risk					
4	$x					
3						
2						
1						
0	Out of stock Danger					

Figure 16: Order Decision as a Markov Decision / Reinforcement Learning Process.

Scenario 4 Test

An important finding of the testing of the MDP/RL concept was the insight that optimal forecast performance and optimal order decision where two separate but interlinked problems. The forecast artefact is optimized to best reflect incoming product demand, ideally taking product and inventory management characteristics into account. The order decision process, however, include other practical factors the estimated risk of stocking out. As long as there are non-deterministic environment variables in the order decision process, the optimal decision of what to order given a forecast might not correspond to the value of the optimized forecast.

Appendix 6: Additional findings from client tests

Additional findings

For Client 3, the overforecastlosses were most relevant once items where passed shelflife and had to be frozen. For client 3, inventory modelling was most relevant for already frozen goods. Production constraints discussed by client 1 suggests item.batchsize should also be incorporated as a running time series of production capacity at each stage if the forecast is to be evaluated realistically. It also confirms forecast is one of several factors considered in the order/production decision process meaning optimal order/production point might not equate production needed to meet forecasted demand. It also suggests a decision support system/feature for optimal order/production point given forecast and other constraints could be useful. The timeframe for Client 3 was also more complex, as items were ordered on a weekly basis, but daily demand was relevant for production adjustments. Weekly average demand was however most important, as this allowed the raw materials to be turned into product before expiry.

Additional comments on future iterations

While the design cycle was mainly focused on capturing ways in which the business value of the forecast could be measured in a monetary amount, the client tests also revealed the emphasis given to inventory level information as a way of evaluating forecasting accuracy. Along with several client´s emphasis on safety stock, this suggests future iterations of the evaluation artefact should incorporate these metrics as outputs of the model. This would allow greater choice in evaluation emphasis for the users. It also reflects the complex nature of the impact forecasting through the supply chain. Inventory levels affect storage space capacities which is yet another constraint that goes into the overall decision processes. The client test findings suggest the valuation Goodwill loss will need to be further developed in future iterations Safety stock binds up working capital. A future iteration of the evaluation model might therefore include the option of setting minimun safety stock to the maxmimum amount of deficit to reflect the practice of safety stock to ensure delivery capabilities. This would effectively eliminate the underforecastloss in favor of while adding to the overforecastloss. This function might be more linear as it eliminates the step of "sudden death" by contract loss, and thus potentially better to optimize over. Findings from clients suggest this might also be a more useful measure of underforecastloss when required service levels are close to or at 100%.

For quantitative forecasting models the simulations of the evaluation model can potentially be scaled up on data series using standard statistical modelling techniques (such as Monte Carlo simulations). Such simulations also provide more in-depth understanding of the AI systems functionalities for AI system users.