A Machine Learning Model Marketplace to Overcome Low Price Elasticity of Sales Forecasting

Marek Hütsch

University of Duisburg-Essen, Germany, marek.huetsch@icb.uni-due.de

A Machine Learning Model Marketplace to Overcome Low Price Elasticity of Sales Forecasting

Marek Hütsch

University of Duisburg-Essen,
Institute for Computer Science and Business Information Systems, Essen, Germany
marek.huetsch@icb.uni-due.de

Abstract. Machine learning (ML) sales forecasting applications occupy a special position in industry and retail, as used in numerous optimization activities, such as price optimization. While optimizing prices, sales forecasting tools require elasticity regarding the optimization objective. When ML methods are used for sales forecasting, there must be sufficient data to learn the proper elasticity requirements. To overcome the problem of low price elasticity of sales forecasts, when historical price-change data are scarce, we present a sales forecasting model marketplace in which trained ML models are shared across companies. Evaluations of a design science project with 43 retail stores show that a *model marketplace* with neural networks and dropout profiles achieves good forecasting accuracy and high price elasticity. With these results, companies can implement automated optimization algorithms while avoiding extant cumbersome stovepiped methods.

Keywords: Model Marketplace, Neural Networks, Price Elasticity, Wine Retailing

1 Introduction

Machine learning (ML) sales forecasting applications occupy a special position in industry and retail, as the forecast is used in numerous activities, such as marketing, sales, finance/accounting, production/purchasing, and logistics [1]. For example, a sales forecast can help reduce the bullwhip effect in the supply chain [2]. The COVID-19 pandemic is a great example of purchase behaviors changing in ways that contradict previous regularities. In this paper, wine retail companies are examined in view of the increased consumption of wine during the COVID-19 pandemic [3]. The increased demand can be interpreted as a strong trend that resulted in under-forecasting during the post-COVID-19 period. When commodity flows change and demand returns to normal, retailers face an excess of stock. Poor demand forecast then influences the subsequent processes. For example, promotional prices [4], normal prices [5], and assortments are optimized [6]. For red wine, prices have the highest influence on demand, followed by the wine's origin [7]. The price elasticity of products can vary. For example, higher-priced wines are more likely to have higher price elasticity [8]. Hence, it is important to create elasticity in the sales forecast so that optimization can occur afterward. Price optimization using a sales forecast that produces the same prediction for each price will interpret every price as optimal. A sales forecasting tool can make relatively accurate predictions in general, even when the price elasticity is weak. Therefore, elasticity

17th International Conference on Wirtschaftsinformatik
February 2022, Nürnberg, Germany

concerning optimization objectives must be studied separately. If ML forecasting models are used for optimization (e.g., price optimization), the following problem arises. To accurately train a price elasticity using ML algorithms, well-organized historical price-change data must be collected beforehand. For many retailers, including the 43 stores in this case study, this condition cannot be met. This problem becomes even worse when demand structures suddenly change, such as with COVID-19. In some cases, historical data can still be used, but it must first be tested. One solution to the problem of missing data is to generate or purchase data from vendors [9]. Alternatively, historic data can be shared among companies to improve the performance of ML models, which is called creating a *data marketplace*. Thus, we propose a specialized solution called the *model marketplace*, in which companies use extant models from the marketplace and then share the parameters of the trained models. Even if a company has few training data, the models can perform well, owing to federated learning, in which models are trained collaboratively and exchanged between marketplace participants [10, 11].

Therefore, the following research questions are studied using a design science research approach:

- RQ1: Can a price elasticity in wine retailing be improved by a model marketplace for cross-company training of ML forecasting models?
- RQ2: Can a model marketplace achieve results similar to a data marketplace?

The answers to these questions will be relevant to practitioners and researchers in the field of wine retailing, and, generally, the field of sales forecasting. In particular, this paper attempts to establish a link between the two research areas: ML-based sales forecasting and the optimization algorithms of operations research. Thus, this paper intends to help retailers gain technical access to operation research optimizations using ML-based forecasting models. Argumentatively, we derive the generalizability of the results for other product groups and for other industries. Furthermore, this paper contributes to the goal of sustainable transport which is one of the grand challenges of the information system (IS) discipline [12] by improving sales forecasting and price elasticity. This generally results in a more accurate forecast when the price is changed for real, which can reduce for example bullwhip effects triggered by inaccurate sales forecasts.

To set the foundation for answering the research questions, the technical background of the proposed implementation is presented in Section 2. Section 3 explains the research design of this paper. In Section 4, the case study is presented, and the solution approach is elaborated on. Section 5 evaluates the solution approach, and Section 6 concludes the paper with a discussion and suggestions for possible future work.

2 Background

The marketplace in this study assumes the economic terms of Evans and Schmalensee [13], in which two-sided marketplaces serve distinct but codependent groups of customers, and the core business of the marketplace provides a common (real or virtual) meeting place and facilitates interactions between members of the distinct groups. Here, the customers are companies that want to implement a price-elastic ML-based sales forecast but lack systematic price changes in their historical sales data. The marketplace customers must coexist in a way that the historic sales database of price changes can be extended to increase the price elasticity of the customer-unique ML-based sales forecasts.

2.1 Sales Forecasting

In retail markets and supply chains, sales forecasting is critical because it is the starting point for many subsequent activities. Inventory planning is a topic that has received a lot of attention and on which Syntetos et al. [14] published a review. Inventory planning can be divided into three sub-problems: planning the safety stock [15], disposition/order planning [16, 17], and inventory optimization [18]. Price optimization [4], promotion optimization [19], placement optimization [20], and assortment optimization [6] all require a sales forecast. Without a demand forecast, Fildes and Hastings found that operations can only respond retroactively in the short term [21]. To optimize a business goal, the forecast must be responsive to changes in the input data. Therefore, we define the price elasticity of demand as the relative change in demand caused by a relative change in price [22, 23]:

$$PED = \left(\tfrac{D_2 - D_1}{D_1}\right) / \left(\tfrac{P_2 - P_1}{P_1}\right). \tag{1}$$

Here, D_2 is the demand caused by price P_2, and D_1 is the demand caused by price P_1. In this paper, we focus on the price elasticity of the demand forecast. This refers to how much the forecast value differs as prices change and how accurate it remains. A highly elastic forecast can produce false values if the forecasted product has little price elasticity in reality. Furthermore, a distinction can be made between short-term price elasticity and long-term price elasticity [24]. Short-term price elasticity is the relative change in demand resulting from a relative change in price shortly after the price change. Long-term price elasticity, on the other hand, includes the relative change in demand that arises from a relative change in price a long time after the price change. In this case study, price increases are not implemented systematically. Price increases can be imagined in situations where very high demand cannot be met owing to limited product inventories. By increasing prices, the demand can be adjusted to the available quantity. In wine retailing, this situation is very rare, and the availability of the wine is usually taken into account in the initial pricing. Therefore, this paper deals exclusively with price reductions, which are usually implemented as a result of promotions. We, therefore, focus on short-term price elasticity. A related work by Tondolo et al. [23] forecasted the demand for electricity using neural networks with an implicitly learned price elasticity. In particular, for beer, Ruhm et al. [25] found that learning the company-specific elasticity using user receipt data was much more accurate than using elasticity values from country-specific studies. This research continues the work on implicitly learning the price elasticity of wines using receipt data from retail stores. Throughout the paper, demand and sales forecasting are treated as interchangeable terms. For example, facing an out-of-stock wine, the customer may have a demand that does not result in a sale, so the demand forecast must be higher than the sales forecast. In this paper, we assume an idealized situation in which the demand forecast is equal to the sales forecast and thus the terms are interchangeable.

2.2 Neural Networks and Federated Learning

Neural networks are trained by adjusting the weights, W, and bias, b, of each neuron. Characterized by the fact that no cycles between the layers are allowed, feedforward neural networks are different from recurrent neural networks. Multi-layer perceptrons are feedforward neural networks that have at least one hidden layer [26–28]. In this work, we use a multi-layer neural network with dense layers. When a layer is fully connected to the previous and next layer, the layer is called a dense layer. This means that a neuron of a layer receives inputs from each neuron of the previous layer and gives its output to each neuron of the next layer. Neural networks with at least one hidden dense layer are commonly referred to as deep neural networks, feed-forward neural networks, or back-propagation neural networks, which are often used for sales forecasting in research [2,29–31]. Neurons of a feedforward neural networks [32–34] receive their input values from the neurons of the previous layer and put these, weighted and summed with a bias, into an activation function, h. Here, the form, $a^{(k)}\,(x) = b^{(k)} + W^{(k)}x$, is the pre-activation function, which is passed to the activation function, $h^{(k)}(x) = g^{(k)}(a^{(k)}(x))$. x is the input of layer k. The activation function, g, must be chosen individually as a hyperparameter. The activation function's outcome is then passed on to the neurons of the following layer until it reaches the output layer.

The sales forecasting marketplace is based on the ML concept of federated learning, which was introduced by Google [10, 11]. Google trained an algorithm across multiple decentralized edge devices or servers holding local data samples without exchanging them. This approach contrasts traditional centralized ML techniques, in which all the local datasets are uploaded to one server (i.e., the data marketplace). With federated learning, common and robust ML models are built across multiple actors without sharing data. This allows the actors to address critical issues such as data privacy, data security, data access rights, and access to heterogeneous data. In subsequent research, data privacy was statistically produced [35]. Therefore, this concept is suitable for sharing knowledge of trained ML solutions between companies that may be competitors. Federated learning makes use of the ML technique of transfer learning, which is based on the assumption that it is possible for neural networks to make valid inferences during operations, despite differences in the feature space and feature distribution between the training and validation dataset [36]. Transfer learning is used to learn a target task $\mathcal{T}_t$ based on a target dataset $\mathcal{D}_t$. To learn the target task $\mathcal{T}_t$ data $\mathcal{D}_s$ from source s can be helpful. A task can be represented by $\mathcal{T} = \{y, f(x)\}$, where y is the result space, and $f(x)$ is the target prediction function. The predictive function $f_{\mathcal{T}}(\cdot)$ for learning task $\mathcal{T}_t$ is improved by transferring knowledge from the predictive function $f_{\mathcal{S}}(\cdot)$. One area of demand forecasting where transfer learning is already being applied is in the forecast of new products. For new products without historical sales data, Karb et al. [37] found product similarities to build an accurate forecast for the new product. When unseen promotions occur for the first time, they were able to reduce prediction variance, demonstrating improved handling of biased training data.

Negative transfer is a major problem [37] in which the task problem space of neural networks can be leveraged to solve a task but the two problems are so different that the solutions are worse when applying transfer learning. Dropout profiles can be used to avoid negative transfer without manually deciding whether to apply transfer learning [38].

Dropout is a method of reducing overfitting and underfitting in neural networks [39]. It has been shown that dropout profiles can be used to avoid transferring knowledge in situations where it would be better to avoid transfer knowledge, known as negative transfer [38]. Through dropout profiles, different parts of the network can be used, learned, and shared in multiple phases of changes. In this paper, dropout profiles are used to avoid negative transfer, and we compare results using dropout profiles against results using normal dropout [40].

3 Research Design

In this paper we used Hevner et al's design science research (DSR) methodology [41] as a research approach, focusing on the (technical) evaluation of the generated artifact. The artifact is an architecture to improve trained neural networks for sales forecasting regarding price elasticity using a DSR approach [42–44]. According to Hevner et al. [41], a project should undergo three kinds of cycles. The rigor cycle is discussed in the previous section [41]. In this paper, the relevance cycle is discussed in relation to two large German retailers with 43 retail stores, which is presented in the next section. During the design cycles, the artifacts are developed as described in this section and elaborated on in detail in the next section. The artifact developed here can be understood as a method according to Peffers et al. [45] since it provides actionable instructions that are conceptual. Section 4 provides insights on the architecture artifact implementation that can be understood as conceptual instructions in practice for the implementation of an ML model marketplace. This should help to ensure the scientific reproducibility of the evaluations.

Here, three artifacts are evaluated: *no marketplace*, *data marketplace*, and *model marketplace*. Various design cycles are utilized to arrive at these three architectures. We start with the artifact no marketplace to provide a data-centered data marketplace using sales data from multiple companies to add value. The no marketplace artifact is the naive approach and can be viewed as evidence for the existence of the formulated problem. Hence, if the no marketplace artifact generates a suitable price elasticity, the defined problem definition is falsifiable. Also, the data marketplace is a common approach used to address missing data by purchasing data from other companies and has a major disadvantage. As data is considered a digital raw material [46], business marketplace participants may have a high interest in not sharing it. In general, data is k-anonymized if the identifying information of each individual is indistinguishable from at least k-1 other individuals, making it difficult to correctly link them to their associated sensitive attributes [47]. Because it is unlikely that data are shared in a competitive market, it is reasonable to consider a third approach. In a model marketplace, data privacy can be enabled using differential privacy [35]. Wei et al. [35] proposed a solution that combined federated learning with differential privacy. In the next section, the three marketplace approaches are described in terms of the specific context of our case study.

4 Case Study Description

We chose two German retailers for our case study, where one has a turnover of over 100 million euros per year, and the other has a turnover of 55 billion euros per year. There is a clear difference in size between the two companies. These and other differences (e.g. customer base) are beneficial to a realistic evaluation of the model marketplace quality. If effective methods for improving the forecast models can be found, added value can be provided. It has already been shown that transfer learning can be successfully applied to scenarios in which individual products are very similar and are sold by the same company [37]. In this paper, a forecasting approach is developed for a more general purpose. In the case study, wine is sold in brick-and-mortar stores. The wine market is characterized by two opposing phenomena that arose years ago [48–50]. On the one hand, the market for mass-produced wine rose dramatically, while on the other hand, specialization toward niche markets with exceptional product quality in terms of terroir, appellation, and regional identity grew. The companies of this study are subject to both phenomena. The smaller company sells a wide range of high-quality wines from all over the world, and the larger company serves a mass market. This is a suitable setup for the evaluation of the model marketplace because Contini et al. [8] found that higher-priced wines are more likely to have higher price elasticity. Therefore, if models benefit from a model marketplace approach here, it is expected that other companies will fare similarly. Both companies face the problem that sales forecast models cannot be trained with good price elasticity. Both companies require price elasticity because automatic price optimizations are needed. However, a price optimization cannot be implemented with a sales forecast lacking elasticity (i.e., forecasts remain the same at different prices). In the case study, two retail companies participate in the marketplace, with the first company having 10 stores and the second having 33. In both cases, the stores are independent contractors that are treated as independent clients having their own price management per article. Hence, one store may have better price elasticity forecasting data than the other. Therefore, we treat the 43 stores as independent marketplace clients.

In the following section, we explain the implementation of the marketplace artifacts introduced as a method to increase sales forecast accuracy and price elasticity. To implement the three marketplace artifacts, we use Python 3.7 and TensorFlow 2.6. As indicated in the right margin of Figure 1, we develop a marketplace GUI where the training progress is displayed across companies and a forecast client graphical user interface (GUI) where each company can see their training progress. The GUIs are developed in SAPUI5 1.93 and are deployed to the SAP CAR 4.0 systems of both companies. The neural network models consist of five dense layers of 8, 64, 8, 4, and 1 neurons with linear activation functions. The mean squared error (MSE, later explained in more detail) is used as the error function. There are N (here N=43) data owners and the set of companies $C = \{1, \ldots, N\}$ with their respective data $\{D_1, \ldots, D_N\}$. We assume that the companies join in the order $1, \ldots, N$. In the following, the three architectures no marketplace, data marketplace, and model marketplace are explained and are visualized in Figure 1.

No marketplace: Step 1: A company, $n \in C$, trains the ML models using its own data. Hence, no mapping between commodity group hierarchies is made.

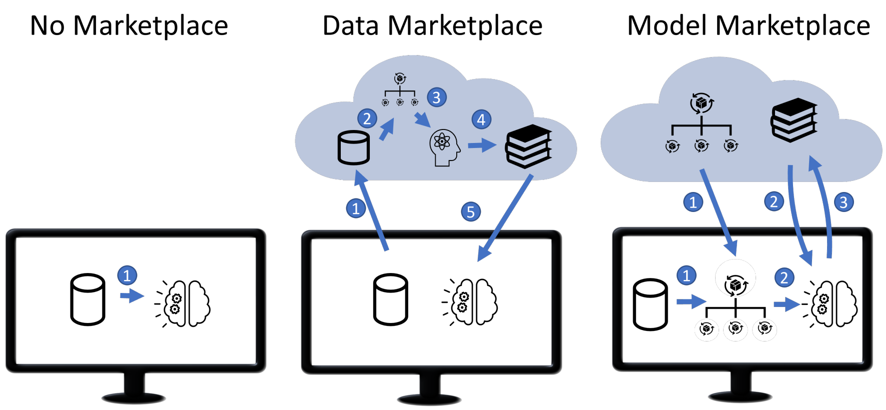

Figure 1. DSR Artifacts: ML Architectures

Data marketplace: Step 1: A new company, $n \in C$, joins the data marketplace and sends its data; the new sales data are stored in the marketplace database. Step 2: All data, including the sales data of the new company, are mapped with a given commodity group hierarchy. Step 3: The data marketplace trains the ML model, meaning that the predictive model, $f_{T_{N\prime}}(\bullet)$, is learned for all companies, $N\prime \subseteq N$, that are part of the data marketplace with the learning task, $T_{N\prime}$, and data source, $D = D_1 \cup \ldots \cup D_{N\prime}$. Step 4: The trained model is stored in a model library at the data marketplace. Step 5: The trained model is transferred to the new company. (Optional) Step 6: A company can receive updates to the trained models when enough new companies join the data marketplace that the model is significantly improved.

Model marketplace: Step 1: A new company, $n \in N$, joins the model marketplace and inherits the commodity group hierarchy from the model marketplace. Then, company n maps its sales data to the commodity groups. Step 2: The model marketplace sends its trained models to the new company as received from company $n-1$. The new company then uses the transferred models and trains them further with the sales data, D_n, of the new company according to the mapped commodity groups. Hence, the predictive model $f_{T_n}(\bullet)$ for learning task T_n of company n is improved by transferring knowledge from $f_{T_{n-1}}(\bullet)$, which is improved by transferring knowledge from $f_{T_{n-2}}(\bullet)$ and so on. Step 3: The additionally trained predictive model $f_{T_n}(\bullet)$ is transferred to the model marketplace. There, the models are saved in the model library. (Optional) Step 4: A company can receive updates to the trained model when enough new companies join the model marketplace that the model is significantly improved.

As step 6 of data marketplace and step 4 of model marketplace are optional, they are not included in Figure 1. Notably, a commodity group hierarchy is defined by the marketplace. Each participant must then map their data to the commodity group hierarchy. When data contradict, the marketplace owner must extend its commodity group hierarchy so that articles of the new participant fit and the map between similar articles is the largest possible. The marketplace has little influence on the sales data used in either the data marketplace or the model marketplace. In the data marketplace, sales data can be provided at any quality and in any quantity (e.g., with or without weather data). Participants can then expand or filter the data as needed. Similarly, in the model marketplace, models can be trained and provided with many features, and a downloading company must find a model that fits their use case. There are mechanisms in the literature describing how a marketplace operator can improve the quality of the provided items (e.g., marketplace governance [51, 52]). However, this is outside the scope of this work.

5 Evaluation

Next, we evaluate the three architecture artifacts to identify which is most suitable for calculating sales forecasts with accurate price elasticity. For the purpose of evaluation, we create a fully known scenario. As commodity group level 1 is forecasted, all historic sales data are aggregated on a weekly, store, price, and commodity group basis. Next, we filter the sales data on commodity group and store combinations having sales for at least 40 weeks a year with at least three prices and at least two sales per price and store. As both retailers have a large assortment of wines with many variants per product, 14 stores and 16 commodity groups fulfill these conditions. For these commodity group and store combinations, we create a complete dataset for the purpose of evaluation. Hence, we use a generative adversarial network (GAN) wherein a generator is trained to create a sales number for a given store, commodity group, calendar week, and price. It has been shown that GANs are suitable for multivariate time-series generation of arbitrary sets of statistically similar data sets based on small subsets of real data [53]. A second discriminator neural network tries to detect which sales are generated by the generator network and which ones are real. After training both neural networks for 5,000 epochs, the generator network is used to create sales data for each store, commodity group, calendar week, and price combinations. This results in 26,884 sales data, which are used to train and validate the neural networks using the different architectural artifacts. Figure 2 displays the aggregated sales per week and price changes for the 16 commodity groups. The first 3 and the last 3 weeks are displayed. Notably, the commodity groups are very stable in sales, and the lowest price (price change = 1) mostly has more sales than the normal price (price change = 0). For commodity groups 17 and 4, there is a strong effect, and for commodity group 6, the different prices have mostly no effect on sales quantities. Furthermore, a general sales increase is more notable in the later weeks than in the first few weeks. This is also true for commodity groups 19 and 17 and hardly true for commodity group 24.

We use a 1/3 validation split of the data. Therefore, 2/3 of the complete scenario data is used for training, and 1/3 is used for validation. In Figure 3, the training process over the 2,000 epochs is visualized. Only the data from two stores exist in both the training and validation data; the training process for these stores is visualized in Figure 3. For the model marketplace, the model is pre-trained by each store and given to each store that provided validation data. When the store also has training data, the model is fine-tuned. The training using no marketplace (i.e., each model trains its own forecast models) has the highest error for both stores that provided training and validation data.

The training using model marketplace with normal dropout fluctuates for store 2, which is not the case when using dropout profiles. Furthermore, the lowest errors are

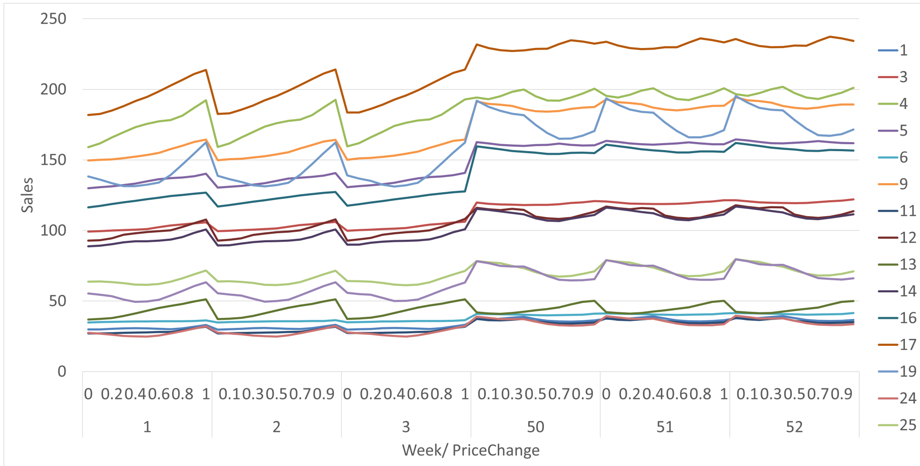

Figure 2. Synthetic Sales Data

achieved using model marketplace, with dropout profiles comparing the error of the model marketplace with dropout profiles of store 1 (2.16E-03) to model marketplace with normal dropout of store 1 (3.16E-03), data marketplace with dropout profiles of store 1 (2.19E-03), and data marketplaces with normal dropout of store 1 (3.18E-03). The same is true for store 2, comparing the error of model marketplace with dropout profiles of store 2 (1.26E-03) to model marketplace with normal dropout of store 2 (2.03E-03), data marketplace with dropout profiles of store 2 (2.19E-03), and data marketplace with normal dropout of store 2 (3.18E-03). Furthermore, comparing the values given previously, we notice that training using dropout profiles results in lower errors.

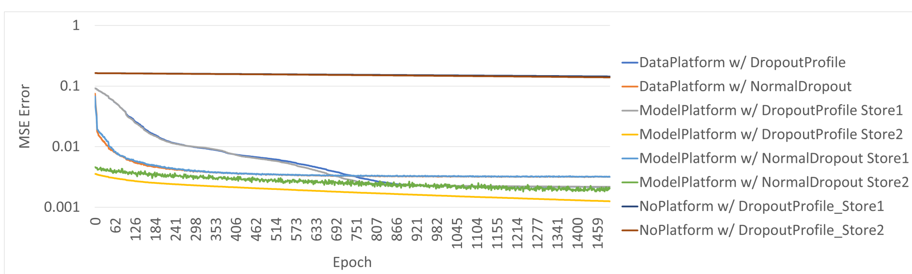

Figure 3. Neural Network Training MSE Errors

Next, we discuss the evaluation of the validation data. Unlike the evaluation of the training phase, we include all validation data. To make the different prices comparable between commodity groups, we use price changes instead of real prices. Therefore *price 0* indicates the normal price, and *price 1* indicates the lowest promotion price detected in the real sales data used for this article. Sales data for all prices between zero and one at a step of 0.1 are generated. Price 0 (normal price) is used as the starting point, so that we can use the following formula to calculate the price elasticity, PED, for a given store, S, article, A, and calendar week, CW, of a new price, p_{new}, with new sales forecast, s_{new}, compared to the the normal price, p_{old}, and with the base sales forecast, s_{old}: $PED = \left(\frac{s_{new} - s_{old}}{s_{old}}\right)/p_{new}$. The aggregated price elasticities per price change are visualized in Figure 4. The model marketplace with dropout profiles is closest to the real price elasticity. The data marketplace with dropout profiles and the model marketplace with normal dropouts are second and third closest, respectively, to real price elasticity, and all forecast means are lower than the real price elasticity. The price elasticity of the forecast using data marketplace with normal dropouts is higher than the real one. No marketplace generated very little price elasticity (e.g., price elasticity for price change 1 is -1.57659E-05), and the green bar is barely visible in Figure 4.

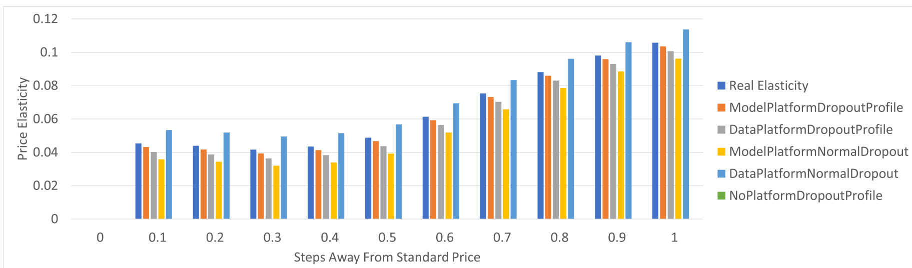

Figure 4. Forecast Elasticity

To compare the elasticity errors of the artifacts, we use the MSE metric [54]. MSE is a measure of the quality of an estimator, and it is always non-negative; values closer to zero are better. MSE is the second moment (the origin) of the error, and it thus incorporates both the variance of the estimator (how widely spread the estimates are from one data sample to another) and its bias (how far off the average estimated value is from the true value). Thus, $MSE = \frac{1}{n}\sum_{t=1}^{n}\left(D_t - \widehat{F}_t\right)^2$. Table 1 shows the MSEs of the forecast price elasticity. The artifact model marketplace artifact with dropout profiles generates the lowest MSE of the forecasted price elasticities.

Table 1. Sales Forecast Price Elasticity MSE

	Dropout Profile	*Normal Dropout*
Data Marketplace	2.61573E-05	6.36936E-05
Model Marketplace	4.66494E-06	9.08114E-05
No Marketplace	0.004798873	

6 Discussion and Conclusion

The model marketplace artifact achieved good results, and its ML model successfully learned price elasticity. The model marketplace artifact using dropout profiles generated the best price elasticity. We attribute this to two effects. First, for the stores whose data were present in training and validation, which is equivalent to the real-world scenario of a store already having some historical data, the pre-trained models are subsequently fine-tuned to the data of the specific store. It has been demonstrated in other scenarios that pre-trained models with a fine-tuning step perform better than others [55–57]. Second, the dropout profiles protect against negative transfer, which is a known issue when creating forecast models using data from different sources [38,58]. From the information gained, the research questions can be answered.

RQ1: Can price elasticity in wine retailing be improved by a model marketplace for cross-company training of ML forecasting models? The data marketplace and model marketplace artifact achieved significantly higher price elasticity, and we conclude that a marketplace for cross-company training of ML forecasting models greatly improves the price elasticity of a forecast model, at least for the case study used in this work.

RQ2: Can a model marketplace achieve results similar to a data marketplace? The model marketplace artifact using dropout profiles achieved results at least as good as those of the data marketplace and is therefore preferable when data privacy is a concern.

We discuss the generalizability of the wine-retailing specific case study according to three dimensions: other commodity groups, other industries, and other optimization areas. Thus, there are two situations to consider:

- There are enough data: Both no marketplace and model marketplace artifacts can learn price elasticity. Even if the model marketplace scenario model does not fit well, there are enough data to fine-tune it to the given commodity group of the given retailer. Dropout profiles should protect marketplace models from negative transfer.
- There are too few data: The performance of no marketplace highly depends on the random initialization of the neural network. This generally should perform badly, as it is highly unlikely that a random initialization fits well with a given scenario. A given company profits from the model marketplace when another retailer has a commodity group that performed similarly. Therefore, model marketplace should always perform at least as good as no marketplace. When there is no good fit with existing models from the marketplace, the artifact model marketplace should perform like a randomly initialized neural network (no marketplace).

Forecasting sales should be similar in other industries, such as logistics and production, as there are always receipts that could be used as training data. Owing to legal requirements, sales data are particularly suitable, owing to their quantity and quality [59], but the standardization of data between companies for other industries may be much more difficult. Furthermore, the described problem of price elasticity can be abstracted to all optimization models that require a sales forecast, such as placement or assortment optimization. This should be explored in further research, including determining how well sales data are suitable to train dedicated elasticity. We expect that other features will be needed.

Next, we discuss the contributions of this paper. In various respects, this work contributes to the literature on IS and has implications for practice and research. First, this work contributes to the existing research on ML marketplaces and provides a current economic discussion based on technical artifacts [58]. In particular, this paper attempted to establish a link between the two research areas of ML-based sales forecasting from

IS research and the optimization algorithms of operations research. The optimization algorithm often makes use of a sales forecast; hence, this paper can help practitioners gain technical access to operation research optimizations with state-of-the-art ML-based sales forecasts. We found that price elasticity is improved by model marketplace artifact forecasts. Hence, price optimization can be improved. Related works have shown that optimization algorithms (e.g., price optimization) are highly dependent on the elasticity of the underlying forecast models [60]. Improved price optimization leads directly to added economic value through profit improvements. For example, a 1% increase in pricing resulted in an 11.1 percent increase in average operating profit [61]. Using the data and model marketplace artifacts, companies can create a price elastic forecast without having their own historical sales data. Therefore, this work supports practitioners by reducing the barriers to using ML. In addition to the specific contributions of this paper for practitioners, this paper also contributes to the 17 Sustainable Development Goals of the United Nations 2030 Agenda for Sustainable Development [12]. As discussed in the introduction, the bullwhip effect triggered by inaccurate sales forecasts can lead to inventory shortages. Predictions, including those of correct price elasticity, result in more accurate forecasts in general whenever the price is changed for real. With a more accurate forecast regarding price changes, appropriate quantities can be delivered to avoid long storage times, waste, or panic buying caused by shortages. An accurate forecast enables the transportation of the right amount of goods and therefore supports the goal of sustainable transport.

There are some limitations to this paper that could potentially be addressed in future work. The results of this work are limited to wine retailing and the limitations in generalizability discussed above. The argumentatively reasoned generalizability of this work could be validated by future work that evaluates the model marketplace for other commodity groups, such as fashion, other industries, such as logistics, and other optimization goals, such as assortment optimization. Although we noted that general data privacy is possible with model marketplace and is therefore preferable over data marketplace, a privacy mechanism was not implemented in this work. Future works should evaluate the performance of anonymization in model marketplace as it is to be expected that the performance of forecasting will decrease. As privacy mechanisms aim to minimize performance decrease while maximizing privacy, we expect the influence on the evaluation results to be insignificant. We make this assumption because a model marketplace should, on average, be at least as good as a randomly initialized neural network supporting no marketplace. For example, differential privacy can be used for anonymization [35]. Furthermore, as indicated in section 4, there are mechanisms in the literature describing how a marketplace operator can improve the quality of the provided items (e.g., marketplace governance [51, 52]). However, this is outside the scope of this work and should be addressed in future work.

References

1. Mentzer, J.T., Bienstock, C.C.: Sales forecasting management. Language 10(274p), 24cm (1998)
2. Carbonneau, R., Laframboise, K., Vahidov, R.: Application of machine learning techniques for supply chain demand forecasting. European Journal of Operational Research 184(3), 1140–1154 (2008)
3. Wolf, M.M., Wolf, M.J., Lecat, B.: Generation z and workers from home drove the growth of california wine consumption during the COVID-19 crisis lockdown. Wine Business.com (2020)
4. Cohen, M.C., Leung, N.H.Z., Panchamgam, K., Perakis, G., Smith, A.: The impact of linear optimization on promotion planning. Operations Research 65(2), 446–468 (2017)
5. Ferreira, K.J., Lee, B.H.A., Simchi-Levi, D.: Analytics for an online retailer: Demand forecasting and price optimization. Manufacturing & Service Operations Management 18(1), 69–88 (2016)
6. Kök, A.G., Fisher, M.L., Vaidyanathan, R.: Assortment planning: Review of literature and industry practice. Retail supply chain management pp. 175–236 (2015)
7. Jarvis, W., Rungie, C., Lockshin, L.: Revealed preference analysis of red wine attributes using polarisation. International Journal of Wine Business Research 19(2), 127–138 (2007)
8. Contini, C., Romano, C., Scozzafava, G., Boncinelli, F., Casini, L.: Wine consumption and sales strategies: The evolution of mass retail trading in italy. Wine Economics and Policy 4(2), 116–127 (2015)
9. Hütsch, M., Wulfert, T.: Implementation of Placement Optimization in Electronic Commerce: Towards a Data Generation Method to Increase Forecast Elasticity. In: Proceedings of the 25th Pacific Asia Conference on Information Systems (PACIS2021). pp. 1–12. Dubai (2021)
10. Konečnỳ, J., McMahan, H.B., Yu, F.X., Richtárik, P., Suresh, A.T., Bacon, D.: Federated learning: Strategies for improving communication efficiency. arXiv preprint arXiv:1610.05492 (2016)
11. McMahan, H.B., Moore, E., Ramage, D., Arcas, B.A.: Federated learning of deep networks using model averaging. arXiv preprint arXiv:1602.05629 (2016)
12. Assembly, U.G.: Transforming our world: The 2030 agenda for sustainable development. Tech. rep., A/RES/70/1 (2015)
13. Evans, D.S., Schmalensee, R.: Catalyst code: The strategies behind the world's most dynamic companies. Harvard Business School Press (2007)
14. Syntetos, A.A., Boylan, J.E., Disney, S.M.: Forecasting for inventory planning: A 50-year review. Journal of the Operational Research Society 60(sup1), S149–S160 (2009)
15. Beutel, A.L., Minner, S.: Safety stock planning under causal demand forecasting. International Journal of Production Economics 140(2), 637–645 (2012)
16. Kuo, R.J., Wang, Y.C., Tien, F.C.: Integration of artificial neural network and mada methods for green supplier selection. Journal of Cleaner Production 18(12), 1161–1170 (2010)
17. Lee, A.H., Kang, H.Y., Lai, C.M., Hong, W.Y.: An integrated model for lot sizing with supplier selection and quantity discounts. Applied Mathematical Modelling 37(7), 4733–4746 (2013)
18. Rezaei, J., Davoodi, M.: A deterministic, multi-item inventory model with supplier selection and imperfect quality. Applied Mathematical Modelling 32(10), 2106–2116 (2008)
19. Yang, Y., Hao, C.: Product selection for promotion planning. Knowledge and Information Systems 29(1), 223–236 (2011)
20. Corstjens, M., Doyle, P.: A model for optimizing retail space allocations. Management Science 27(7), 822–833 (1981)
21. Fildes, R., Hastings, R.: The organization and improvement of market forecasting. Journal of the Operational Research Society 45(1), 1–16 (1994)

22. Esteves, R.B., Reggiani, C.: Elasticity of demand and behaviour-based price discrimination. International Journal of Industrial Organization 32, 46–56 (2014)
23. Tondolo de Miranda, S., Abaide, A., Sperandio, M., Santos, M.M., Zanghi, E.: Application of artificial neural networks and fuzzy logic to long-term load forecast considering the price elasticity of electricity demand. International Transactions on Electrical Energy Systems 28(10), e2606 (2018)
24. Norström, T.: The price elasticity for alcohol in Sweden 1984–2003. Nordic Studies on Alcohol and Drugs 22(1_suppl), 87–101 (2005)
25. Ruhm, C.J., Jones, A.S., Kerr, W.C., Greenfield, T.K., Terza, J.V., Pandian, R.S., McGeary, K.A.: What US data should be used to measure the price elasticity of demand for alcohol? Tech. rep., National Bureau of Economic Research (2011)
26. Kim, K., Lee, J.: Sequential manifold learning for efficient churn prediction. Expert Systems with Applications 39(18), 13328–13337 (2012)
27. Zhang, D., Zhou, L., Kehoe, J.L., Kilic, I.Y.: What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews. Journal of Management Information Systems 33(2), 456–481 (2016)
28. Gordini, N., Veglio, V.: Customers churn prediction and marketing retention strategies. an application of support vector machines based on the auc parameter-selection technique in b2b e-commerce industry. Industrial Marketing Management 62, 100–107 (2017)
29. Alon, I., Qi, M., Sadowski, R.J.: Forecasting aggregate retail sales: A comparison of artificial neural networks and traditional methods. Journal of Retailing and Consumer Services 8(3), 147–156 (2001)
30. Lee, W.I., Chen, C.W., Chen, K.H., Chen, T.H., Liu, C.C.: Comparative study on the forecast of fresh food sales using logistic regression, moving average and BPNN methods. Journal of Marine Science and Technology 20(2), 142–152 (2012)
31. Loureiro, A.L., Miguéis, V.L., da Silva, L.F.: Exploring the use of deep neural networks for sales forecasting in fashion retail. Decision Support Systems 114, 81–93 (2018)
32. Moraes, R., Valiati, J.F., Neto, W.P.G.: Document-level sentiment classification: An empirical comparison between SVM and ANN. Expert Systems with Applications 40(2), 621–633 (2013)
33. Zhao, X.: Research on e-commerce customer churning modeling and prediction. The Open Cybernetics & Systemics Journal 8(1), 800–804 (2014)
34. Kulkarni, S.V., Dhage, S.N.: Advanced credit score calculation using social media and machine learning. Journal of Intelligent & Fuzzy Systems 36(3), 2373–2380 (2019)
35. Wei, K., Li, J., Ding, M., Ma, C., Yang, H.H., Farokhi, F., Jin, S., Quek, T.Q., Poor, H.V.: Federated learning with differential privacy: Algorithms and performance analysis. IEEE Transactions on Information Forensics and Security 15, 3454–3469 (2020)
36. Dai Wenyuan, Y.Q., Guirong, X., Yong, Y.: Boosting for transfer learning. In: Proceedings of the 24th International Conference on Machine Learning, Corvallis, USA. pp. 193–200 (2007)
37. Karb, T., Kühl, N., Hirt, R., Glivici-Cotruta, V.: A network-based transfer learning approach to improve sales forecasting of new products. arXiv preprint arXiv:2005.06978 (2020)
38. Hütsch, M.: Transfer learning to enable flexibility of demand forecasting: Dropout profiles to protect from negative transfer. In: ECIS 2021 Proceedings. p. 1531 (2021)
39. Phaisangittisagul, E.: An analysis of the regularization between L2 and dropout in single hidden layer neural network. In: Proceedings - International Conference on Intelligent Systems, Modelling and Simulation, ISMS. vol. 0, pp. 174–179. IEEE Computer Society (2016)
40. Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R.: Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580 (2012)
41. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly pp. 75–105 (2004)

42. Ereth, J., Baars, H.: A capability approach for designing business intelligence and analytics architectures. In: HICSS. pp. 1–10 (2020)
43. Passlick, J., Lebek, B., Breitner, M.H.: A self-service supporting business intelligence and big data analytics architecture. In: Proceedings der 13. Internationalen Tagung Wirtschaftsinformatik (WI 2017). pp. 1126–1140 (2017)
44. Mettler, T., Raber, D.: Developing a collaborative business intelligence system for improving delivery reliability in business networks. In: 2011 17th International Conference on Concurrent Enterprising. pp. 1–7. IEEE (2011)
45. Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
46. Economist, T.: The world's most valuable resource is no longer oil, but data. The Economist: New York, NY, USA (2017)
47. Sweeney, L.: k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(05), 557–570 (2002)
48. Charters, S.: Wine and society: The social and cultural context of a drink. Routledge (2006)
49. Nelgen, S., Anderson, K.: Global wine markets, 1961 to 2009: A statistical compendium. University of Adelaide Press (2011)
50. Hammervoll, T., Mora, P., Toften, K.: The financial crisis and the wine industry: The performance of niche firms versus mass-market firms (2014)
51. Tiwana, A.: Platform Ecosystems. Morgan Kaufman, Waltham (2014)
52. Alves, C., Oliveira, J., Jansen, S.: Software Ecosystems Governance-A Systematic Literature Review and Research Agenda. In: ICEIS2017. pp. 26–29 (2017)
53. Leznik, M., Michalsky, P., Willis, P., Schanzel, B., Östberg, P.O., Domaschka, J.: Multivariate time series synthesis using generative adversarial networks. In: Proceedings of the ACM/SPEC International Conference on Performance Engineering. pp. 43–50 (2021)
54. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
55. Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., Smith, N.: Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv preprint arXiv:2002.06305 (2020)
56. Lee, J.S., Hsiang, J.: Patentbert: Patent classification with fine-tuning a pre-trained BERT model. arXiv preprint arXiv:1906.02124 (2019)
57. Jiang, H., He, P., Chen, W., Liu, X., Gao, J., Zhao, T.: Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. arXiv preprint arXiv:1911.03437 (2019)
58. Karb, T., Kühl, N., Hirt, R., Glivici-Cotruță, V.: A network-based transfer learning approach to improve sales forecasting of new products. In: European Conference on Information Systems (ECIS) - Marrakech, Marocco, June 15 - 17, 2020 (2020)
59. German VAT invoice requirements, howpublished = `https://www.german-tax-consultants.com/vat-services/german-vat-invoice-requirements.html`, note = Accessed: 2021-11-15
60. Borin, N., Farris, P.: A sensitivity analysis of retailer shelf management models. Journal of Retailing 71(2), 153–171 (1995)
61. Marn, M.V., Rosiello, R.L.: Managing price, gaining profit. McKinsey Quarterly pp. 18–18 (1992)