Machine Learning Under An Optimization Lens Final Project

Anders STEINESS
Michel LEROY de LARRE de la DORIE

4th of November 2024

Contents

Abstract

This report explores predictive and prescriptive approaches to optimize weekly rental pricing for Vibrent, a Norwegian sustainable clothing rental company. By leveraging multimodal data—including tabular, text, and image embeddings— and TabText techniques the study aims to prescribe revenue-maximizing rental prices for new products. Predictions results suggest model generalization varies a lot especially in the context of temporal data. XGBoost and K-nearest neighbor (KNN) had the best and most stable price prediction performance for different modality combinations. Linear Regression, LASSO, Ridge, and Elastic Net had relatively poor out of sample performance while CART's result were highly modality sensitive. Prescription results indicates that weighted prescriptive KNN demonstrated higher out of sample revenue than XGBoost and KNN point prediction prescription. A weighted KNN enhanced with optimal transport for domain adaptation prescriptive model, achieved the highest revenue. The results contribute to advancing prescriptive pricing in the fashion industry by underscoring and evaluating the interplay between feature weighting, multimodal integration, optimization techniques for temporal domain adaptation, and prescription methodologies in driving actionable pricing strategies

1 Introduction

Determining the optimal price for a product or service is one of the most critical decisions a company can make. Pricing not only impacts immediate revenue but also influences customer perception, demand, and long-term profitability. A price point that maximizes revenue ensures that a business can sustain operations, invest in growth, and deliver value to its stakeholders. For companies like Vibrent, which are at the forefront of sustainable fashion, prescribing the right price is particularly important—it allows them to balance economic success with their mission to promote environmental sustainability. This report focuses on developing a pricing strategy that maximizes revenue, leveraging data and analytics to support Vibrent's business goals.

Vibrent is a leading Norwegian clothing company that promotes sustainability in fashion through rental. They released their data this year, which dates back to 2016 and consists of four separate components: outfits, users, pictures, and transactions. Notably, this dataset originates from real-world business operations and was initially developed for use in recommender systems. Its imperfections presented an opportunity to simulate a realistic business scenario. This allowed us to explore its potential and derive actionable insights despite inherent limitations, reflecting the challenges and demands faced in real-world analytical setting.

1.1 Objective and Assumptions

The aim of this project is to help Vibrent maximize revenue on new product releases by prescribing an optimal price. An optimal price equates a revenue maximizing price. For us to conduct this project, we need to assume that the rental price in Vibrent's dataset is the true revenue maximizing price. In economic terms, this would mean that the rental price in Vibrent's dataset is associated with a price elasticity of 1 where price elasticity defines the proportional change in demand as price changes. This assumption will defining for the optimization model's objective and constraints later on.

1.2 Predictive and Prescriptive Models

To prescribe an optimal price, we will both work with point prediction prescription methods and weighted prescription method using KNN. Thus, this report can be viewed in two parts: First, obtain the best possible machine model to predict the true price. Second, prescribe the optimal revenue maximizing price using point prediction prescription and weighted prescription.
This project will also work on enhancing the weighted prescription using KNN *a priori* by utilizing optimal transport to improve scenario generation. While not explicitly covered in the course, optimal transport aligns with the mission of leveraging optimization principles to address distributional shifts between domains to improve generalization of machine learning model predictions under certain conditions.

2 Motivation and Background on Optimal Transport

This section will briefly cover the motivation and technical implementations of optimal transport since we did not cover it in class.

A method we aim to investigate for improving our weighted prescriptive model is discrete optimal transport. The motivation stems from changes in fashion trends over time. For example, if our model was trained in time period where "orange was out of style" and in a new time period "orange is the new black" (or for seasonal fashion adjustments) it might make sense to use the features of a "black sweater" to predict the price of an "orange sweater". The same principle applies to branding.

More formally, we expect that fashion has been subject to temporal covariate-transformation from our in-sampe distribution, Ω_s, to our out of sample distribution, Ω_t. However, the transformation function that describes the possible change in fashion is unknown to us. Discrete optimal transport is an optimization method that allow us find a transformation that minimizes the cost of mapping a discrete set of target points to a discrete set of source points (Courty et al. 2015).

The exact discrete optimal transport seeks to find the matrix γ that contains the relative contribution of elements i and j to the barycenter to align the distributions with each other (Courty et al. 2015). The exact discrete optimal transport mapping can be set up as an linear optimization problem (Courty et al. 2015) where we are trying to find $\hat{\gamma}$:

$$\hat{\gamma} = \arg\min_{\gamma} \sum_{i,j} \gamma(i,j) \|x_i^s - x_j^t\|^2 \quad \text{s.t.} \quad \sum_j \gamma(i,j) = \frac{1}{n_s}, \quad \sum_i \gamma(i,j) = \frac{1}{n_t}$$

Where n_s is the number of sample points in the source distribution (training set) and n_t is the number of sample points in the target distribution (test set). Note that in the above problem we use the euclidiean distance, but another distance measure could be used. This exact discrete optimal transport problem quickly becomes intractable as the number of data points increases (in our case, the problem became intractable for n $>$ 2000 datapoints and p $>$ 2000). As outlined by Peyré and Cuturi (2018), adding entropy regularization $-\lambda \sum_{i,j} \gamma(i,j) \log \gamma(i,j)$ to the objective function together with the sinkhorn algorithm, improves scalability and approximates the optimal solution to the exact discrete optimal transport problem. The entropy regularization term makes it a strictly convex optimization problem and smoothens the coupling matrix γ (moving the optimal solution from extreme point to interior) where the sinkhorn algorithm afterwards utilizes properties of the gradient of the Lagrangian. The smaller λ is, the closer we are to the optimal solution of the exact discrete optimal transport problem (Peyré & Cuturi, 2018). However, if λ is too small, convergence issues can occur. The objective in this paper is to approximate exact discrete optimal transport as good as possible. The smallest λ where we experienced convergence for our dataset was 5 (we also included 10 for comparison).

3 Method

In this section, we will review the problem setting more formally. We will then cover the preproccesing of our dataset with a focus on dealing with unstructured image, text data, and TabText. Finally, we describe the models investigated for predicting the optimal price.

3.1 Problem Setting

The objective is to maximize revenue by prescribing the optimal weekly rental price for new product obtained by Vibrent. We are assuming that the true revenue maximizing rental price is what is reported in the dataset. We are also assuming a linear demand curve. As a new product release is of one unit, quantity will range between zero and one. A fractional quantity can be interpreted as the percentage chance of selling the product. In our model, if we set the price to zero then the quantity is one. The revenue maximizing price (price elasticity of 1) is at a quantity of 0.5. Extrapolating the linear demand curve, we get a maximum price (when quantity hits 0) of two times the revenue maximizing price. Thus, the relationship between the price and quantity can be written as:

$$P = 2y(1 - Q), \quad st. \quad 0 \leq Q \leq 1$$

Where P denotes the price sat, Q determines the quantity, y is the revenue maximizing price. We assume that we (the resellers) know about the structure of the demand curve. Our goal is to predict the true maximizing price. If we let P be the decision variable for price then actual revenue is: $R = PQ = P(1 - \frac{P}{2y})$.
We are comparing the out of sample performance of different prescription models: Point prediction model using XGBoost, point prediction using KNN, weighted prescription model using KNN, and weighted prescription using KNN with optimal transport.
Since we are predicting the revenue maximizing price, our optimal price for point prediction models will be the predicted prices. Thus, the predicted revenue for point prediction models will be the predicted price times 0.5 quantity.
For weighted/scenario based prescription, one risk is that a scenario will have much lower predicted price (than the other scenarios) such that it's quantity constraint will become active and then subsequently constraining the optimal price (through the relationship between the quantity and the price). To mitigate this risk, we allow one scenario to have a non-active relationship between the quantity and the price with the condition the quantity is being sat to zero (the price can be set to greater than two times the predicted price). Appendix Y illustrates this.
The reason why we do not allow more than one scenario is that the scenarios will generally be close to each other so the need for this procedure is limited. Further, it will combat the possibility of one scenario with an abnormal high price that potentially would push the pricing decision to too high compared to the actual true revenue mazimizing price.

3.2 Data Processing

The objective of the data processing is to extract as much relevant information to predict the optimal weekly rental price of a new product from the dataset given. With inspiration from TabText and multi modality covered in class, we seek to utilize multi modal data including tabular data together with clothing images and text descriptions in the models through embeddings (Carballo et al., 2022). It will then be investigated what model, hyperparameters, and modality combinations, that provides the best predictive performance.

3.2.1 Dataset Summary and Initial Preprocessing

The entire data available is composed of individual datasets covering 15.6k unique outfits, 9.7k outfit types, 7.4k anonymized users with accompanying rental histories, 50.1k pictures, and a total of 77.1k transactions. We will focus on 15.6k outfits, each described by a name, a date sold, the retail price when the item came out, a text description, a picture, and a set of tags. Tags are categorical descriptors of an outfit's attributes, such as "Dress", "Green", or "Wool". These have been manually annotated to the outfits by Vibrent's employees during the onboarding process.

At this stage, although the dataset is extensive, it is characterized by only 2 cleaned tabular features (original retail price and sale time). Thus, further preprocessing is needed. First, the list of tags are one-hot encoded. Without the methodological framework provided by the course, the analysis might have concluded at this point, yielding a R^2 score of .938 using XGBoost . The remainder of the preprocessing demonstrates the additional value achieved through incorporating multi modality features using techniques from the class.

3.2.2 Tabular Data

For the baseline model, we only kept the tabular data. That is the retail price, time step feature that counts the number of days since the start of the dataset, and the list of tags. The tags were unordered and in list format, so one-hot-encoded was done. This feature set constituted the "No Embedding" features.
However, one disadvantage of this feature set is that it does not reflect the specificity of each unique items that would develop through tear and wear or sophisticated aesthetic differences. Hence, the need to enrich our data with the unstructured data.

3.2.3 Unstructured Data

Access to the images of each garment was provided in the dataset. We used embeddings generated by a pre-trained EfficientNet V2 L model from its second to last layer. Empirically the model has shown to be high performing in regards to zero shot images (Tang & Shustin, 2023).
To make use of the written description of the garment, which might reflect the quality and condition of a piece, we computed the embeddings of this text by the LLM model BERT.

Finally, had we stopped there, we would potentially have lost a lot of valuable information about the relationships between the textual features and the tag description. First, one-hot encoding losses the semantic similarity between categories by making each vector orthogonal. Second, the tags might shine a new light on the meaning of the description, resulting in different embeddings if they were integrated in the attention layer. For these reasons, a new version of the dataset was created with the tags features of the tabular data being converted into strings and embedded through a LLM , in the spirit of the TabText approach (Carballo et al., 2022). To that end, we also created versions where we concatenated the tags with the textual description of each garment before feeding them to the LLM. This resulted in two additional embedding features: Name + Description and Name + Description + Tags.

At the end of this heavy data processing set, we had 6 datasets to evaluate, each temporally decomposed into a training, validation and test set:

Table 1: Feature composition of each dataset tested

Feature Type	Retail Price	Time	Tags One Hot Encoding	Concatenated Name and Description Embeddings	Concatenated Name, Description and Tags Embeddings	Image Embeddings
No Embedding	✓	✓	✓			
Name + Description	✓	✓	✓	✓		
Name + Description + Tags	✓	✓			✓	
Image	✓	✓	✓			✓
Name + Description + Image	✓	✓		✓		✓
Name + Description + Tags + Image	✓	✓			✓	✓

The features of these dataset are not heavily correlated with one another or weekly rental price. To see this we condensed for clarity the embeddings in 10 features using PCA, see Appendix 3.

3.2.4 Training, Validation, and Test set

The data will be split into a train, validation, and test set based upon temporality. This better reflects our business challenge as trends and brands change with time and respect that in practice we only have access to past data. The test set will be from 2023 and onward, which includes beginning months of 2024. This constitutes approximately 20% of the data. The first 80% of the data from 2016 to 2022 will be used as training data, and the last 20% will be used as validation data.

3.3 Models Used

We settled on this list of model for the prediction of the weekly rental price for each of our datasets. The hyperparameters used can be found in Appendix 1.

Table 2:

Model Used	**Reason**
Linear Regression	Classic model that would be used as a benchmark and correct potential mistakes.
Lasso	More robust version of linear regression; the induced sparsity would help interpret the great number of features in our dataset.
Ridge	More robust version of linear regression and an alternative to Lasso that helps with multicollinearity.
KNN	A non-parametric method that uses feature proximity particularly well-adapted to the dataset according to the original paper (Borgersen et al., 2024).
Elastic Net	Combination of Ridge and Lasso, widely used for linear regression.
CART	An interpretable decision tree-based approach that captures non-linear relationships.
XGBoost	Highly efficient gradient boosted tree method, widely used for regression.

For every data set, we will tune each model over the set of hyperparameters and report the R^2 on the validation set. The baseline model will be the sample mean. The model with the highest R^2 on the validation set will be the model used for point prediction prescription.

4 Prediction Results

In this section, the validation performance for all models, hyperparameters, and datasets will be assessed. This will lead to the choice of the models representing the point prediction prescription models. Finally, takeaways from the prediction part will summarized.

4.1 Validation Results

See Appendix 2 for all the optimal hyperparameters for every model for every feature set. Note that only computational constraints limited the hyperparameter space and the number of models we were investigating.

Table 3: Best Validation R2 for Models with Optimal Hyperparameters

Model	No Embedding	Name + Description	Name + Description + Tags	Name + Image	Name + Description + Image	Name + Description+ Tags + Image
Linear Regression	0.494	0.306	0.181	0.230	-0.135	-0.334
Lasso	0.483	0.480	0.461	0.378	0.341	0.174
Ridge	0.470	0.352	0.221	0.141	0.055	0.038
KNN	0.909	0.910	0.909	0.908	0.908	0.908
Elastic Net	0.463	0.435	0.465	0.272	0.266	0.228
CART	0.942	0.817	0.587	0.966	0.941	0.421
XGBoost	0.938	0.975	0.916	0.984	0.980	0.914

We obtained the best result with XGBoost using the dataset consisting of one-hot encoded tags and image embeddings. The hyper-parameters chosen were learning rate: 0.1, max depth: 6. The final R2 on the validation set was 0.984. As the weighted prescription model will be based upon KNN, we will include KNN as a point prediction model as well. Note that the R^2 for KNN was consistently around 0.908 - 0.91 for all dataset. Further, as 7 is the optimal choice for the number of nearest neighbors, it will also be the number used for the weighted prescription model.

Since the one-hot encoded tags and image embeddings were the optimal feature set for the best-performing prediction model, XGBoost, we will use the same feature set for the prescription phase of the project to maintain consistency. This will also apply to weighted prescription.

4.2 Test Set Results

To get an unbiased estimate of the out of sample performance of the best performing prediction model, XGBoost, we refit the model on entire training and validation set using the one-hot encoded tags and image embeddings feature set. As we are also conducting a point prediction method for KNN, we will investigate its performance on the test set using same steps as with XGBoost. The test results were: XGBoost with an $R^2 = 0.862$, and KNN with an $R^2 = 0.873$. Surprisingly, KNN had a better performance on the test set.

4.3 Takeaways

The challenges we imposed on ourselves led to some significant result.
Due to the temporal data split and the evolving trends, the test set data distribution was not equivalent to the training data distribution. This led to some negative R^2 for validation set (Appendix 2) for linear regression. This justified the investigation of other models that could potentially generalize better to a change in the underlying distribution.
It was observed that a naive implementation of TabText still rivaled the image embeddings. The implementation TabText could have been improved by testing more ways to semantically describe the tags and concatenation with name and description. This shows the flexibility and potential additional predicting power from a TabText implementation.
We observed that it can be detrimental to standardize the data before feeding them to the KNN. The reasoning is that standardization assumes uniform weight to the features, which might not always be appropriate especially with high-dimensional embeddings. Scaling / not standardizing the data is appropriate when you have prior knowledge on a system. In our case, retail price has great importance and not scaling it it down adds act as a weight (Appendix 4). In the future, one might consider the scaling as a hyper parameter after having standardized the features,

5 Prescription

5.1 Model Formulation

The problem setting give rise to following model formulation.

We obtain predicted revenue maximizing prices (scenarios) for a new product release: $\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_K\}$ (e.g., from K nearest neighbors). Then we find the optimal price z_i to assign the new product by solving the following Mixed-Integer Quadratic Programming problem for product i:

$$\max_{z_i} \sum_{k=1}^{K} w_k c(z_i, \hat{y}_{i,k})$$

Where w_k is the weight put on scenario k, z_i is the price assigned, and $c(z_i, \hat{y}_{i,k})$ is:

$$c(z_i, \hat{y}_{i,k}) = R_{i,k}$$

$$R_{i,k} = z_i Q_{i,k} = z_i \left(1 - \frac{z_i}{2\hat{y}_{i,k}}\right)$$

$$z_i \leq 2\hat{y}_{i,k} + M\delta_{i,k}$$

$$Q_{i,k} \leq \left(1 - \frac{z_i}{2\hat{y}_{i,k}}\right) + M\delta_{i,k}$$

$$Q_{i,k} \geq \left(1 - \frac{z_i}{2\hat{y}_{i,k}}\right) - M\delta_{i,k}$$

$$Q_{i,k} \leq M(1 - \delta_{i,k})$$

$$\sum_{k=1}^{K} \delta_{i,k} \leq 1$$

$$z_i \geq 0, \quad R_{i,k} \geq 0, \quad Q_{i,k} \geq 0$$

$$\delta_{i,k} \in \{0, 1\}$$

Objective

The objective isto maximizes the weighted average revenue for product i across scenarios $\sum_{k=1}^{K} w_k c(z_i, \hat{y}_{i,k})$.

Decision variables

$R_{i,k}$: Revenue for product i in k'th scenario.

z_i: The price assigned to product i. The price assigned is non-negative.

$\delta_{i,k}$: A binary variable. $\delta_{i,k}$ will be used to enable one scenario to have a non-active quantity constraint that else could constrain the optimal pricing.

$Q_{i,k}$: Quantity for product i in k'th scenario. Quantity is non-negative.

Constraints

First constraint: States $R_{i,k}$ is equal to the price times quantity.

Second constraint when $\delta_{i,k} = 0$: This constraint ensures that the price z_i assigned has to be less or equal to two times the predicted revenue maximizing price as a result of the demand curve assumptions.

Second constraint when $\delta_{i,k} = 1$: This enables one scenario to have this constraint inactive so the optimal price is not constrained by this scenario. In the subsequent constraints, the quantity in this scenario will be mapped to zero.

Third, fourth, and fifth constraint when $\delta_{i,k} = 0$: The quantity of product i in the k'th scenario is equal to $\left(1 - \frac{z_i}{2\hat{y}_{i,k}}\right)$, derived from the demand curve assumption.

Third, fourth, and fifth constraint when $\delta_{i,k} = 1$*:* The quantity of product i in the k'th scenario will be mapped to zero if it is optimal to set a price z_i above $2\times$ the scenario's predicted revenue maximizing price prices.

5.2 Prescriptive models

The prescription models employed will include XGBoost and KNN Point Prediction where both models are fitted on the entire traininig and validation dataset with one-hot encoded tags and image embeddings, using optimal hyperparameters. For the weighted KNN prescription model, the 7 nearest neighbors from the training and validation set will be used as predicted prices for every test data point. Weighted KNN prescription model with optimal transport will use same methodology as standard weighted KNN prescription model but the optimal transport transformation will be applied on the image embeddings beforehand. Appendix 7 shows a 2D (first two principal components) visualization of the image embeddings before and after the optimal transport transformation where it can be observed that the embedding distributions aligns to much higher degree after optimal transport.

5.3 Results and Takeaways

As shown in the prior sections, the dataset contains a lot of signal regarding the weekly rental price, which the non-linear machine models are able to capture. If the true revenue maximizing prices were known and subsequently assigned to the products, the total revenue would be 1,100,720. The table below shows the actual total revenue obtained for every prescription model out of sample.

Model	Actual Revenue ($)	Absolute Increase ($)	Improvement Over Baseline (%)
KNN PP (Baseline)	1,092,016	0	0.0000
XGBoost PP	1,094,104	2,088	0.1913
KNN WP	1,096,017	4,001	0.3663
KNN WPO $\lambda = 10$	1,096,025	4,010	0.3670
KNN WPO $\lambda = 5$	1,096,025	4,010	0.3670

Table 4: Comparison of Actual Revenues, Absolute Increases, and Improvements Over Baseline (KNN PP).

Notes: PP = Point Prediction, WP = Weighted Prescription, WPO = Weighted Prescription with Optimal Transport.

From the table above we see that KNN point prediction performs the worst followed by XGBoost point prediction. The weighted KNN prescription model has the best performance with the optimal transport transformation only increasing the performance slightly. Generally, all the models are close to the revenue upper bound (based on the demand assumptions) so there is a limitation to how large the increase in performance can be.
The results highlight that a higher R^2 value on a sample does not necessarily guarantee better performance in point prediction prescriptions as KNN actually had better performance than XGBoost on the test set; the size and context of the prediction error play a critical role in the effectiveness of the prescription model.
Moreover, prescribing over a distribution of prices, rather than a single point estimate, leads to more informative prescription decisions. Additionally, leveraging optimization for domain adaptation, particularly in covariate transformations, can improve the generalization of machine learning model predictions and is therefore a valuable tool when facing problems related to covariate transformations.

6 Conclusion

In this work, we explored predictive and prescriptive modeling approaches to optimize pricing strategies for Vibrent's sustainable clothing rental business. By integrating tabular, text, and image data, our models aimed to predict rental prices that maximize revenue. Despite challenges like temporal distribution shifts and imperfect datasets, methods such as KNN and XGBoost demonstrated robust predictive power, particularly when leveraging multimodal embeddings. We also examined the impact of advanced techniques, like optimal transport, on domain adaptation and pricing prescriptions, achieving marginal but meaningful revenue improvements. Key insights include the importance of contextual feature weighting, relevance of data sources in multimodal models, and the need to balance predictive accuracy with prescriptive efficacy for business applications. Future work could enhance this framework by refining embeddings, exploring dynamic pricing under real-time trends, and expanding optimization strategies for greater generalizability and operational scalability. These findings underscore the interplay between data quality, model sophistication, and actionable business insights in AI-driven pricing strategies. Finally, this project is going towards more sustainable consumption, emphasizing second-hand and sharing alternatives. Potential user won't have to guess themselves the quality online as are assured to pay what their product is truly worth.

7 Contributions

With regards to the code, the report and presentation, equal work was contributed by Anders and Michel. Both team members worked on the data preprocessing and feature engineering. The project can essentially be seen as two parts: The prediction and prescription part. Michel focused mostly on implementing the prediction part whereas Anders focused mostly on implementing the prescription part. However, both Michel and Anders brainstormed together on each part in regards to for example the methodology related to machine learning models used and prescription model formulation. Both contributed to writing the introduction and methodology. Michel focused more on writing the prediction part. Anders focused more on writing the prescription part. Anders wrote the abstract and Michel wrote the conclusion. Both proofread the reports separately and worked on presentation together.

8 References

Courty, N., Flamary, R., Tuia, D., & Rakotomamonjy, A. (2015). Optimal Transport for Domain Adaptation. ArXiv (Cornell University). https://doi.org/10.48550/arxiv.1507.00504

Peyré, G., & Cuturi, M. (2018). Computational Optimal Transport.
arXiv. https://doi.org/10.48550/ARXIV.1803.00567

Borgersen, K. A. K., Goodwin, M., Grundetjern, M., & Sharma, J. (2024). A Dataset for Adapting Recommender Systems to the Fashion Rental Economy. https://doi.org/10.1145/3640457.3688174

Tang, M., & Shustin, D. (2023). Renderers are Good Zero-Shot Representation Learners: Exploring Diffusion Latents for Metric Learning (Version 1). arXiv. https://doi.org/10.48550/ARXIV.2306.10721

Carballo, K. V., Na, L., Ma, Y., Boussioux, L., Zeng, C., Soenksen, L. R., & Bertsimas, D. (2022). TabText: A Flexible and Contextual Approach to Tabular Data Representation (Version 4). arXiv.
https://doi.org/10.48550/ARXIV.2206.10381

9 Appendix

9.1 Appendix 1

Table 5: Hyperparameter Search Space for Models

Model	Hyperparameters
Linear Regression	No hyperparameters
Lasso	α: [0.1, 1, 10]
Ridge	α: [0.1, 1, 10]
KNN	$n\_neighbors$: [3, 5, 7, 10]
Elastic Net	α: [0.01, 0.1, 1, 10]; $l1\_ratio$: [0.1, 0.5, 0.9]
CART	max_depth: [5, 10, 20]
XGBoost	$learning\_rate$: [0.01, 0.1, 0.3]; max_depth: [3, 6, 10]

9.2 Appendix 2

Table 6: Validation R^2 and Optimal Hyperparameters for Models

Model	Embedding Type	Validation R^2 & Optimal Hyperparameters
Linear Regression	No Embedding	R^2: 0.494, –
	Name + Description	R^2: 0.306, –
	Name + Description + Tags	R^2: 0.181, –
	Image	R^2: 0.230, –
	Name + Description + Image	R^2: -0.135, –
	Name + Description + Tags + Image	R^2: -0.334, –
Lasso	No Embedding	R^2: 0.483, α: 0.1
	Name + Description	R^2: 0.480, α: 0.1
	Name + Description + Tags	R^2: 0.461, α: 0.1
	Image	R^2: 0.378, α: 0.1
	Name + Description + Image	R^2: 0.341, α: 0.1
	Name + Description + Tags + Image	R^2: 0.174, α: 0.1
Ridge	No Embedding	R^2: 0.470, α: 1
	Name + Description	R^2: 0.352, α: 1
	Name + Description + Tags	R^2: 0.221, α: 0.1
	Image	R^2: 0.141, α: 10
	Name + Description + Image	R^2: 0.055, α: 10
	Name + Description + Tags + Image	R^2: 0.038, α: 10
KNN	No Embedding	R^2: 0.909, $n\_neighbors$: 7
	Name + Description	R^2: 0.910, $n\_neighbors$: 7
	Name + Description + Tags	R^2: 0.909, $n\_neighbors$: 7
	Image	R^2: 0.908, $n\_neighbors$: 7
	Name + Description + Image	R^2: 0.908, $n\_neighbors$: 7
	Name + Description + Tags + Image	R^2: 0.908, $n\_neighbors$: 7
Elastic Net	No Embedding	R^2: 0.463, α: 0.01, $l1\_ratio$: 0.9
	Name + Description	R^2: 0.435, α: 0.01, $l1\_ratio$: 0.9
	Name + Description + Tags	R^2: 0.465, α: 0.01, $l1\_ratio$: 0.9
	Image	R^2: 0.272, α: 0.01, $l1\_ratio$: 0.1
	Name + Description + Image	R^2: 0.266, α: 0.01, $l1\_ratio$: 0.1
	Name + Description + Tags + Image	R^2: 0.228, α: 0.01, $l1\_ratio$: 0.1
CART	No Embedding	R^2: 0.942, max_depth: 5
	Name + Description	R^2: 0.817, max_depth: 5
	Name + Description + Tags	R^2: 0.587, max_depth: 10
	Image	R^2: 0.966, max_depth: 10
	Name + Description + Image	R^2: 0.941, max_depth: 5
	Name + Description + Tags + Image	R^2: 0.421, max_depth: 20
XGBoost	No Embedding	R^2: 0.938, $learning\_rate$: 0.1, max_depth: 6
	Name + Description	R^2: 0.975, $learning\_rate$: 0.1, max_depth: 6
	Name + Description + Tags	R^2: 0.916, $learning\_rate$: 0.1, max_depth: 3
	Image	R^2: 0.984, $learning\_rate$: 0.1, max_depth: 6
	Name + Description + Image	R^2: 0.980, $learning\_rate$: 0.1, max_depth: 6
	Name + Description + Tags + Image	R^2: 0.914, $learning\_rate$: 0.1, max_depth: 3

To calculate R^2 we used the naive estimator consisting of the mean over the validation set. However one could argue that the company would not have directly access to the mean as each sales comes one by one. In that case the company would have used the average on the training set as a baseline estimator. As a result it is likely we underestimated our R^2 results.

9.3 Appendix 3

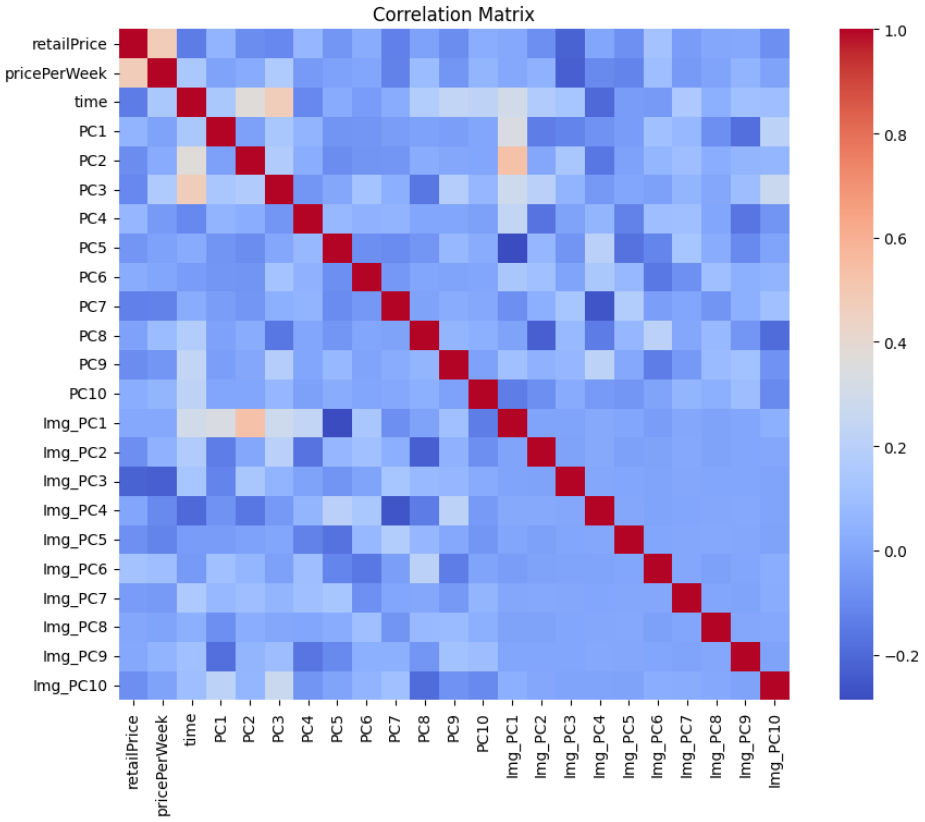

Figure 1: Correlation between retail price, tabular features and simplified image and textual embeddings

9.4 Appendix 4

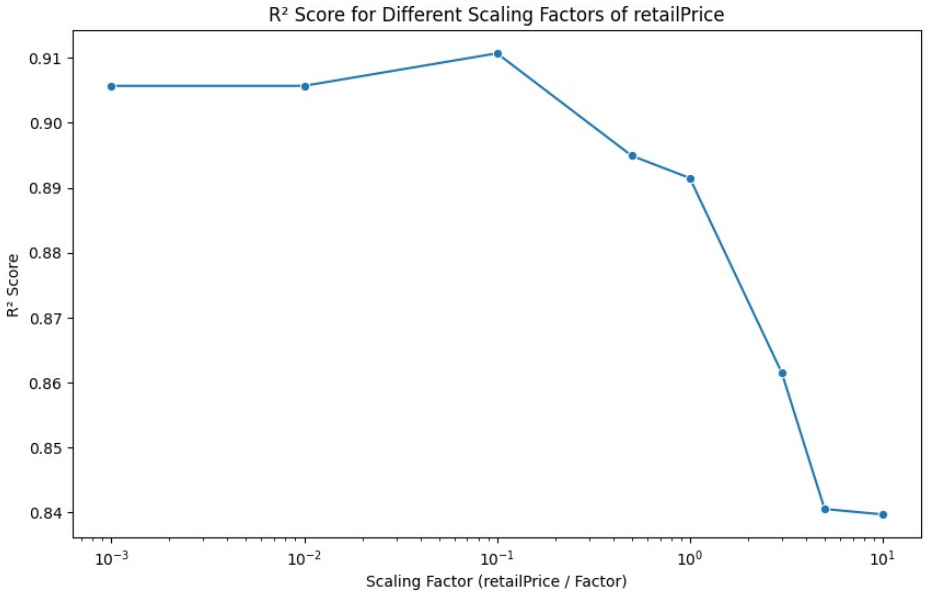

Figure 2: Affect of retail price scaling on KNN's performance

9.5 Appendix 5

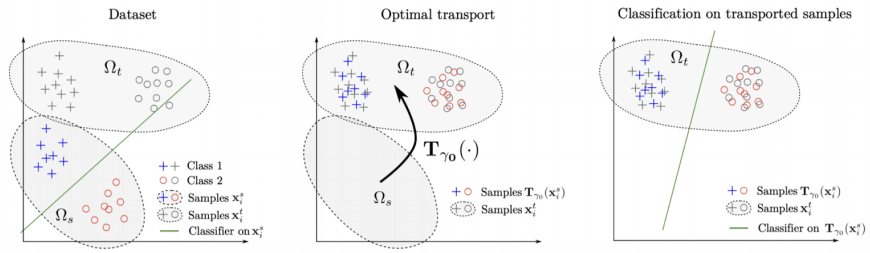

Figure 3: Optimal Transport Illustration. *Source: Courty et. al [2015]*

9.6 Appendix 6

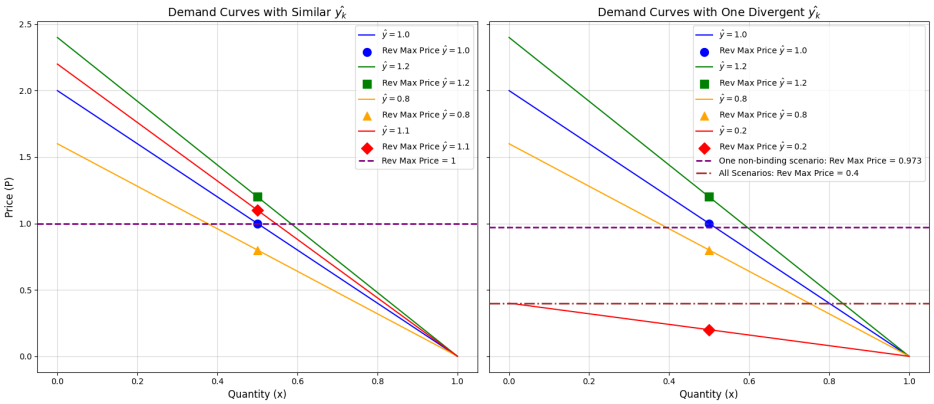

Figure 4: Demand Scenarios

When all the scenarios are close to each other as on the left figure, no scenario will have a quantity active constraint that would subsequently constrain the price. Here the optimal revenue will be 0.5.
On the figure to the right, we have one scenario (the predicted revenue maximizing price of 0.2), which has an active quantity constraint that subsequently constrain the optimal price. If this is allowed, the optimal price is 0.4, which yields a revenue of 0.238. If we allow one scenario to have a quantity of 0 but without it affecting the optimal price (through the Big-M constraints) then our optimal price will around 0.973 (and our quantity in $\hat{y} = 0.2$ scenario will be 0), which would yield a revenue of 0.365 the $\hat{y} = 0.2$ scenario will have a revenue of 0.

9.7 Appendix 7

..

The plots below are based upon image embeddings. We keep the training (this include both the training and validation set) constant and apply optimal transport transformation to the test set.

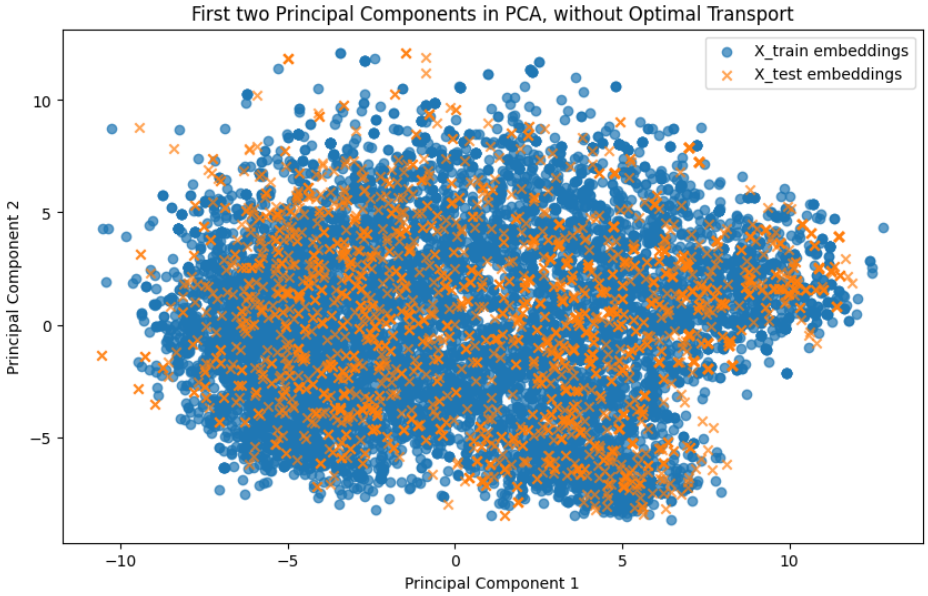

Figure 5: 2D PCA of Image Embeddings Before Optimal Transport

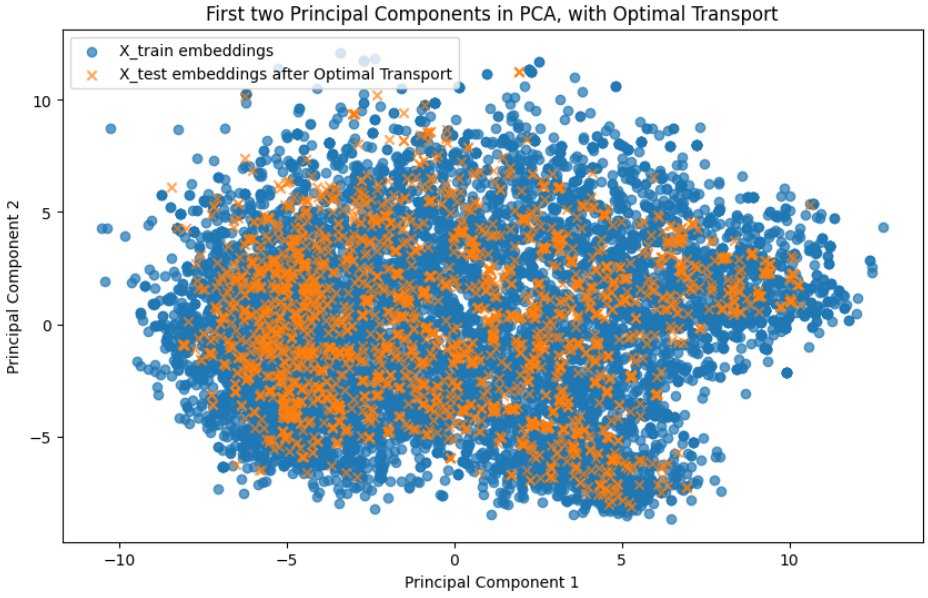

Figure 6: 2D PCA of Image Embeddings After Optimal Transport