Dynamically pricing tickets for cultural venues

Dennis van Peer

Dynamically pricing tickets for cultural venues

by

Dennis van Peer

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday June 19, 2020.

Student number:	4321138	
Thesis committee:	Dr. Neil Yorke-Smith,	TU Delft, supervisor
	Tomas Pereira de Vasconcelos,	Tiqets, supervisor
	Dr. Pradeep Murukannaiah,	TU Delft

TU Delft

Abstract

Entrance tickets for cultural venues are often sold through online retailers. Internally, these retailers have access to demand insights of a large amount of globally distributed venues. Using Tiqets as our industry partner, we were able to access data from one of the largest retailers with global reach. This thesis explores the possibility of leveraging prices to adjust the demand for a venue.

In finding the relation between price and demand from historical sales records we encounter several obstacles. This thesis explores three different ways of minimizing the negative effects of using observational data.

By introducing a decomposable time-series regression model to predict demand we can isolate the effects of trend, seasonality and recurring events without having to introduce new data. Introducing price as a module enables us to control for the desired future demand.

Secondly, we explore exogeneity in the variation observed from historical price changes and introduce ways of ensuring exogeneity on future data.

Finally, we look for an automated way of finding similarity across venues using the associated historical sales data and use this to estimate the variation in error when grouping venues in a controlled experiment.

Acknowledgements

The following thesis has been the result of months of hard work. I was fortunate to have been given the freedom and opportunity to work on a project of my own choosing, allowing me to expand my knowledge on some fascinating topics within computer science, statistics and economics. This would never have been possible were it not for the support of some amazing people.

First and foremost I would like to thank my supervisor Neil Yorke-Smith, who has supported the idea behind this thesis from conception to completion, for providing me with guidance throughout the entire process and allowing me the freedom to approach this project on my own terms. Because of you, I was able to work on a project I am passionate about in an environment that allowed me to learn so much in such a short time.

I want to thank Uri Shimron for his deep expertise in statistics and data science, his ability to explain concepts visually and his patience for me understanding them. In fact, I would like to thank everyone at Tiqets for providing me not only with the data and expertise to make this project successful, but also one of the warmest and most welcoming environments I have been in so far. Within the data team, I felt included right from the start. You guys make it possible for me to look back at even the hardest parts of the project fondly.

Within Tiqets I would especially like to thank Tomas Pereira de Vasconcelos for taking on the role of supervisor on very short notice and for finding the time within an already busy schedule to provide me with feedback, help me with my implementation and lead me to some very interesting papers within my problem domain.

I also want to thank my friends for supporting me through this time, for helping me to take my mind of the thesis and for keeping me sane. I could not have completed this thesis without the many movie nights, gaming sessions and half-hearted attempts at exercise to contrast the work.

Finally, I would like to thank my parents and brothers for their unwavering support, always allowing me to vent and encouraging me to carry on. I can never express how grateful I am for all that you have done to support me. Not just for this thesis, but for everything in my life that has led me to where I am today!

Dennis van Peer
Delft, June 2020

Contents

Terminology

Product date:	In the context of tickets for cultural venues, product date refers to the date this ticket is valid.
Booking date/Order creation date:	In the context of tickets for cultural venues, product date refers to the date the ticket is bought.
BDIA (Booking Days In Advance):	Time delta between the product date and the booking date. i.e. BDIA is 1 when a ticket is bought the day before its product date. 0 When bought on the same day as it is used.
Channels:	Channels refer to the sales channels Tiqets uses to sale their tickets. Current channels include android, ios, web and google reserve.
Price posted time:	The total amount of time a product was sold for price X
Venue:	A venue resells products (tickets) through the Tiqets platform.
Product:	A venue can provide tickets for one or more attractions, these attractions are called products.
Product Variant:	A product variant is a version of the base product that differs in either price or included features (e.g. children/senior tickets, included audio-guide, etc.).
Base Product:	The main product variant for that product, often the most sold variant
Order:	An order is attached to one customer, one product, booking date and product date, but can consist of multiple tickets.
Posted price time:	The amount of time a certain price was active for a product
Price-elasticity (of demand):	The degree to which the demand for a product changes as the price changes. Usually expressed in $\mathbb{R}^1$.
Brick and mortar stores:	Physical stores selling physical products. As opposed to digital platforms selling either physical or digital products.
Observational data:	Refers to information gathered without the researcher having control over the independent variable. The observational data discussed in this research are the historical sales records of the platform.

1 Introduction

1.1. Company

Tiqets is a Dutch company founded in 2014. The company has set up an online platform reselling tickets for globally distributed cultural venues such as museums, shows, and attractions. At the point of writing the company sells tickets from over 3000 venues scattered over more than 600 cities around the world. Each venue comes with its own set of rules and restrictions regarding reselling tickets. The customer-base of Tiqets is as varied as its product catalogue, attracting customers worldwide. The platform is available in multiple languages. Tiqets aims to provide a clean and consistent view of nearby venues and their properties. This helps international tourists explore the cultural possibilities in the area and provides a unified way for the customer to see things venue properties, availability and purchase options.

Tiqets makes its money via either a commission or purchase margin on the sold tickets. At the moment the price of a venue changes at either the venue's request or at the discretion of an in-house expert. This happens rarely compared to the daily updates that a possible automatic solution can provide and requires more experts when the amount of partnered venues grows. It is therefore that the company has expressed interests in finding such an automatic solution capable of handling a large number of venues.

Partnering with Tiqets allows us to take advantage of the sales data of all venues contained in the platform. This thesis will focus on extracting useful pricing information from this available data. We will also recommend ways for the company to improve its data collection methods[1], allowing for more advanced analyses in the future.

1.2. Research questions

The goal of this research is to find the extent to which we can extract price information from observational data. For this we have outlined several research questions along with some subquestions:

1. To what extent are we able to extract the relation between price and demand on a per venue basis from observational data?
   (a) What methods are available and what are their advantages and drawbacks?
   (b) What kind of data is necessary to quantify the quality of the observational data?
2. What model most accurately predicts demand on a per venue basis?
   (a) What is the effect of including or omitting price as a regressor?
3. In what ways can we extract and compare similarities the sales data of venues?
   (a) Can a method of clustering on this similarity be used to the benefit of the company?

[1]Tiqets is already implementing some of these changes at the time of writing.

1.3. Outline

The thesis aims to show the dangers of extracting price information from observational data, as well as provide pointers on how to improve this particular dataset in such a way to mitigate these dangers. Each chapter starting with chapter 3 will provide a way of extracting the relation price and demand on a per venue basis in an automatic way. Since we are working with observational data these methods are flawed, their shortcomings and differences will be discussed at the end of each chapter as well as in the conclusion (chapter 7). A more detailed outline follows:
After discussing related works (chapter 2) and introducing some terminology, we will explore the state of the data in chapter 3. This chapter also includes the baseline analysis, calculating the price-demand relation in a naive way.

Chapter 4 explores a way of improving the baseline analysis by modelling price unrelated patterns extracted from the sales data and removing those from the analysis. This chapter also attempts to model price directly as a linear basis expansion of the price regressor.

Chapter 5 focusses on detecting causality per price change. The chapter proposes a randomized price experiment that would be free from the limitations intrinsic to observational data. The results from this experiment could be used to quantify the accuracy of each previous method[2].

Chapter 6 introduces and quantifies several methods of clustering venues. These clusters can be used to create treatment and control groups for A/B tests (pricing or otherwise). While chapter 7 summarizes the results and outlines use cases of the research for Tiqets.

[2]One such experiment was planned for this thesis. Unfortunately, this was cancelled during the global COVID-19 pandemic.

2 Related works

2.1. Dynamic pricing

The field of dynamic pricing and demand in academic research is an interesting combination of operations research, econometrics, marketing and recently computer science. This field has been steadily rising in popularity along with the growing efforts of businesses to improve their data infrastructure. Possible advantages are even higher for online retailers with little to no cost associated with price adjustments, although research in cases where the opposite is true does exist [25]. Our industry partner (Tiqets) is in the situation where price adjustments in response to updated information or changed circumstances for most products incur a practically zero cost penalty.

Current industries benefiting from dynamic pricing systems include airline companies, the hospitality sector, car rental, retail stores, internet advertisement and electricity industry. A great example to familiarize yourself with the topic is the electricity industry. This industry's approach to changing its infrastructure to what they call a smart grid significantly increases: the amount and quality of data collected, the awareness of pricing to the consumer and adds the ability for consumers to provide power to the grid instead of just taking power [11, 12, 26]. For the electricity industry, dynamic pricing is especially interesting since the cost of electricity generation and distribution increases significantly under higher loads and the cost of storage is also more expensive than that of most physical or digital products.

Just like our problem domain, the theoretical cost of price adjustments is next to zero. Gabr et al. [11] however found by comparing different pricing schemes that customers were most reactive to systems with fewer price changes and less variation in prices. The electricity industry also prioritises demand control over time to revenue optimisation. Our industry partner has expressed their preference for methods that allow them to switch between objectives such as revenue optimisation, increasing brand awareness and inventory control. We, therefore, focus in this thesis on modelling the demand and demand-price relation, while at some points providing concrete examples for certain objectives.

Fisher et al. [10] includes a state-space model to determine price elasticities across multiple products. Their model is tested in a field experiment partnered with an online retailer. They include an empirical comparison of the quality of their observational data. While their model captures less complex price-demand relations than we do in chapters 4,5 and 6, they do include competitor price and even stock-out information. Unfortunately, we do not have this data available for historical data, but based on their results it would greatly improve demand-price relation estimations. It is therefore highly advisable for Tiqets to start collecting similar information if possible.

Fisher et al. [10] paper was chosen as the guideline for the practical experiment of chapter 5 over several other papers outlining similar experiments including the ones from Ashraf et al. [2], Karlan and Zinman [21] and Gneezy and Rustichini [14]. The preference for Fisher et al. [10] is because of out of these papers it is the one with the most similar product and distribution method as the one used by Tiqets. That is consumable products sold in an online environment.

As opposed to the previous papers the papers Ferreira et al. [9] and Gaur and Fisher [13] do research consumable products and are therefore more relevant. The former even concerns an online retailer. Ferreira et al. [9] limit their scope to static prices for flash sales on new products and Gaur et al. run their experiment in brick-and-mortar stores. The paper by Fisher is also more recent than the previously mentioned papers.

2.2. Time series analysis

Since we are primarily dealing with time series, this section introduces some of the related concepts used throughout the thesis along with their usage in relevant research. For most machine learning concepts such as cross-validation, we use adjusted versions designed for time-series analysis. One exception to this being the methods in the clustering chapter applied to the features from section 6.1.1, where feature engineering is used to represent time series data without the time axis as proposed in Hyndman [19].

The Makridakis Competitions (or M competitions) are a set of time series forecasting competitions intended to evaluate and compare the accuracy of different forecasting methods. In 2000 the third M competition introduced a benchmark of 3003 time-series compared across forecasting models. Since then an informal requirement of papers applying to the International Journal of Forecasting has been to evaluate your proposed method using the M3 competition dataset [18]. The data was taken from several business, demography, finance and economics sources and contains data with non-seasonal, monthly or quarterly seasonality. This is representative of the sales data of our venues.

STL stands for Seasonal and Trend decomposition using Loess. This is a well-known method to remove the trend and/or seasonal component from a time series. These components can be used individually for further analysis, the remainder is also useful since it is often stationary [20]. In this thesis STL is used primarily in chapter 6, this is because analyses that rely on the demand model already include a form of trend and seasonality removal that is more lenient to a changing trend over time (see section 4.2.2).

Within the field of time series forecasting the Prophet model introduced in 2017. Taylor and Letham [31] implements a solution that is scalable on a large corpus of time series with similar characteristics. This model is what is known as a modular regression model, fitting interpretable parameters to (by default) 3 different modules that are separable and decompose the time series data in a manner similar to STL decomposition. Other popular models allowing time series forecasting include ARIMA and Exponential smoothing [20]. Chapter 4 discusses these in more detail.

3
Data and Initial experiments

This chapter outlines what data is available within Tiqets and how it is structured, where data is missing and how this was handled in the analyses from this section onward. Non-trivial decisions made regarding restructuring or excluding data can also be found in this section. Terms introduced in this chapter relating to the data will be used throughout the thesis report, but can also be found (briefly) in the terminology reference.

More concretely the objectives of this chapter are:

- To define the structure of the available data and the restrictions this brings.
- To define how to extract information form this data that will be used throughout the thesis.
- To set a baseline for the estimation of the relationship between price and demand and to discuss Results and limitations.

Appendix A shows auxiliary plots for this chapter.

3.1. Structure of available data

Historical data within Tiqets is stored within a relational database which contains data on (among others) all Products, Orders and Venues since the start of the current platform in 2014. The database has since then grown along with the needs of the company to accommodate multiple channels (see the terminology reference), A/B testing for user interface enhancements, a growing need for data analyses and the challenges of accommodating for differences and complications that come with accommodating venues on an international level.

Venues (e.g. The Sagrada Familia) can contain multiple products (e.g. Skip The Line & Guided Tour ticket), which in turn can contain multiple product variants (e.g. Adults or Seniors). An order in the database is linked to one (and only one) product but can consist of multiple product variants.

Tiqets sells its products across multiple channels with varying prices. Customers and their behaviour may differ per channel. As such, customers booking through the google-integrated google reserve API app may have a different reaction to a price change than customers booking through the ios app. For the initial analysis, only the biggest channel (tiqets.com) is considered.

These orders can be linked to financial information through their id's. Orders have an associated status (e.g. NEW, CANCELLED, EXPIRED, PAID, FULFILLED,...). Statuses corresponding to orders placed but never paid such as NEW or EXPIRED can give additional information on the demand of a product. The FAILED status can give additional information on restrictions set by the venue such a maximum ticket amount. For the initial pricing analysis the PAID, BOOKED and FULFILLED statuses are used to predict demand, while only the FULFILLED status is used to estimate the historical price (since information from the financial records is needed which is only available for fulfilled orders).

3.2. Restrictions on considered data

The initial pricing analysis in this chapter focuses on the base product (see the terminology reference) for selected products. These products are picked to both have enough data to accommodate a meaningful pricing analysis and to have a minimal amount of confounding factors impacting the pricing and demand data. One common confounding factor is the venue putting a maximum on the number of tickets that Tiqets is allowed to sell over certain periods. Effectively setting a cap on the demand data that is not stored in the database.

The subset of products that fit these criteria are chosen manually taking into account the popularity of the venue, its internal representation in the Tiqets platform[1] and specifics on the contract with the venue (such as inventory or seasonal constraints).

Within each product, several more restrictions apply. The first is a restriction on the channel on which the order is placed. Tiqets facilitates several channels including the web platform, mobile apps and an integrated google-reserve widget. Users can, but often do not switch from one channel to another. Tiqets is also known to vary prices based on the channel. For the initial analysis, each user-base corresponding to a channel is considered to have their own independent price-elasticity.

The second restriction is the restriction on product variants. Each product is comprised of several product variants. When buying a product, the user always selects a product variant. These usually differ in price. A product can be divided into variants on buyer characteristics as well as features of the product (e.g. child/senior or audio-book availability). The effect of price is derived on the product-variant level since product-variants are considered to have their own independent price-elasticity.

One can imagine that these independent price-elasticities are not completely independent. Therefore in future work information derived from orders on channel, product-variant or even product level can be combined.

3.3. Retrieving historical pricing

Product variants can change prices for several reasons. Venues can adjust their prices according to a change in demand or a change in the product itself (e.g. a museum adding an exhibit). For most product types Tiqets is able to change the final price for the tickets. The cases where this has been done solely for the reason of exploring price elasticity are the most interesting. In cases like this, the change in demand can be seen as a reaction to the change in price as opposed to a product-based or seasonal influence. Section 5.1 proposes an experiment based on this principle.

Unfortunately, Tiqets has not stored historical pricing data[2]. This means that while the database contains products and product-variants. The price of these products gets overwritten each time it is changed. The reason for these price changes is also not stored in a structured way, while they can usually be found in a log relating to the venue. Since payment information is stored in the database we can estimate pricing information based on the orders placed during that time. This section shows how this is done and discusses the limitations of this method.

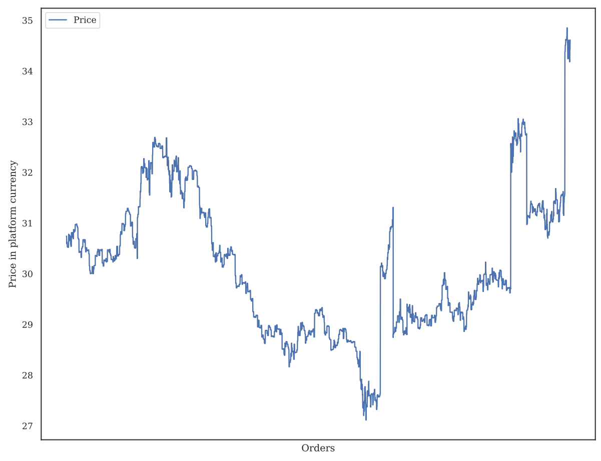
3.3.1. Currency

The amount of money paid for the tickets is stored in the currency used by the customer. Using information from the order, the ticket price and booking fee can be retrieved per ticket. A booking fee is a percentage of the ticket price as opposed to a constant added onto the order. This makes the price of a large order of the same product variant equivalent to the sum of several smaller orders adding to the same amount.

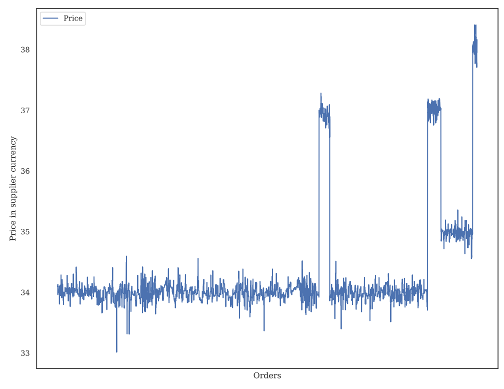
For analysis purposes, we look at the price in the currency of the supplier. We can do this using historical exchange rate data provided by the European Central Bank [3]. Conversion rates in the ECB dataset are stored per day, updated at around 16:00 CET on working days. The rate used in the supplier currency calculation is the interpolated value between the nearest data points. Since the ECB dataset does not contain rates on the weekend, the calculated supplier currency is made more stable thanks to the interpolation.

An improved method for supplier currency calculation relies on a conversion rate value stored within the Tiqets financial database. Conversion rates are stored once per day if an order is made in that currency for that day. Since this data includes conversion rates from the weekends it is more complete for the data that we need to consider. Some small variation on the calculated price still exists but becomes negligible when averaging over a 24 hour period. Figure 3.1 shows a visual comparison of the two methods.

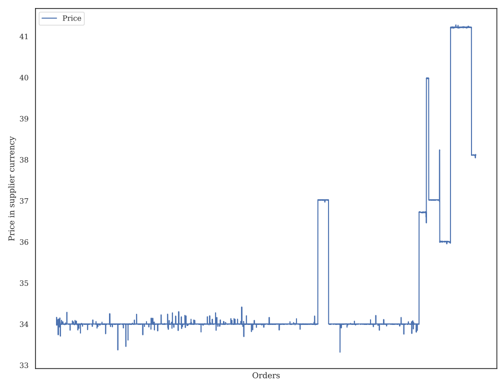
[1] One ticket for a venue may be spread into different products with specific differences complicating the analysis.
[2] Since the moment of writing this has changed, allowing for more accurate analyses on data from 2020 onwards.
[3] `https://www.ecb.europa.eu` ECB historical rates for 42 currencies against the Euro since 1999.



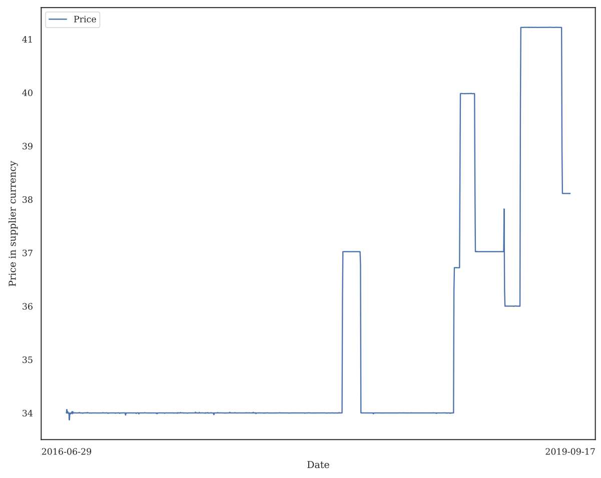
(a) Ticket price in platform currency per order



(b) Price converted into supplier currency using ECB data per order



(c) Price converted into supplier currency using Tiqets data per order



(d) Price converted into supplier currency using Tiqets data (averaged) per day

Figure 3.1: Price over time for a product with different currency conversion methods

3.3.2. Posted price time

We define the posted price time for a product as the amount of time a certain price was active for a product. Certain analyses such as the one from section 3.5.1 require the posted price time as a metric.

Since price changes are not stored historically this is not a known metric. However, the posted price time can be estimated rather reliably for most products because of the frequency of the orders.

For the set of date/time values when an order was made $t \in T$ and corresponding prices $p \in P$ where $|T| = |P|$ we chronologically go through all orders based on order creation date [4] and select values where $p_i \neq p_{i+1}$. A price change is then estimated to have happened at $\frac{t_{i+1}-t_i}{2}$. The posted price is calculated by summing the duration between price changes for each price point.

3.3.3. Limitations

The biggest limitation of retrieving historical prices from orders is that all price change moments are essentially estimations. Any metric based on the posted price time[5] will be off by a certain amount. Price changes can also be omitted completely from the history if no orders are made during the price window.

3.3.2 The posted price is calculated by summing the duration between price changes for each price point. As such prices that are posted over multiple disjoint periods will share a posted price time. This means that in a simple demand estimation based on posted price times the demand may be greatly influenced by factors such as the trend and seasonality of the product.

3.4. Retrieving historical demand

Since order information is readily available to us, successful orders can be used as a measure for demand. This information is essential in extracting the relation between price and demand.

Several product types exist that generalize over properties of the products sold by Tiqets. Two properties of these products are of particular interest when retrieving demand: pre-buy and intermediate products.

A pre-buy product has a limited inventory as opposed to an intermediate product where a ticket from the venue is acquired directly when a product is bought. When this inventory is sold out, Tiqets (but not necessarily the venue itself or different ticket retailers) are unable to keep selling the product for the relevant product data. While historical demand for an intermediate product can be retrieved by the number of orders, a pre-buy product where the inventory is sold out generally has more demand than the number of successful orders. In such cases, the actual demand can be estimated by looking at the rate of orders for a product per product date.

A product can have multiple online or offline retailers, including (optionally) the venue itself. The competitors for a product, as well as the ease of switching to a competitor, are relevant to the price elasticity of the demand for the product. When retrieving demand for a product with multiple competitors only the order data from Tiqets is used to determine demand since this is the only data available. This means that historical demand for products where Tiqets is not the only retailer is the demand of that product on the platform, which is subject to change according to actions from both the venue, Tiqets and that of competitors for that product.

With the retrieved historical demand, as well as the historical prices retrieved in section 3.3 we can construct a combined graph showing the relation between price and demand for a product over time. An example of a price/demand graph for a product can be seen in Figure 3.2. The Figure shows a graph containing the sales over time for a venue starting from the date it was included in the platform. The blue values indicate a sale at a certain price in the currency of the venue for that product. The venue name is obscured in this thesis.

[4]In this case order creation date is chosen over the product date since historically most prices were changed without regard to the product date. E.g. if the Sagrada Familia price was raised by €5 on Jan 1st. Customers that bought tickets for a visit on Jan 10th before Jan 1st will have paid a lower price than customers who bought that same ticket after the price change while visiting the same venue on the same date

[5]The amount of time a certain price was active for that product

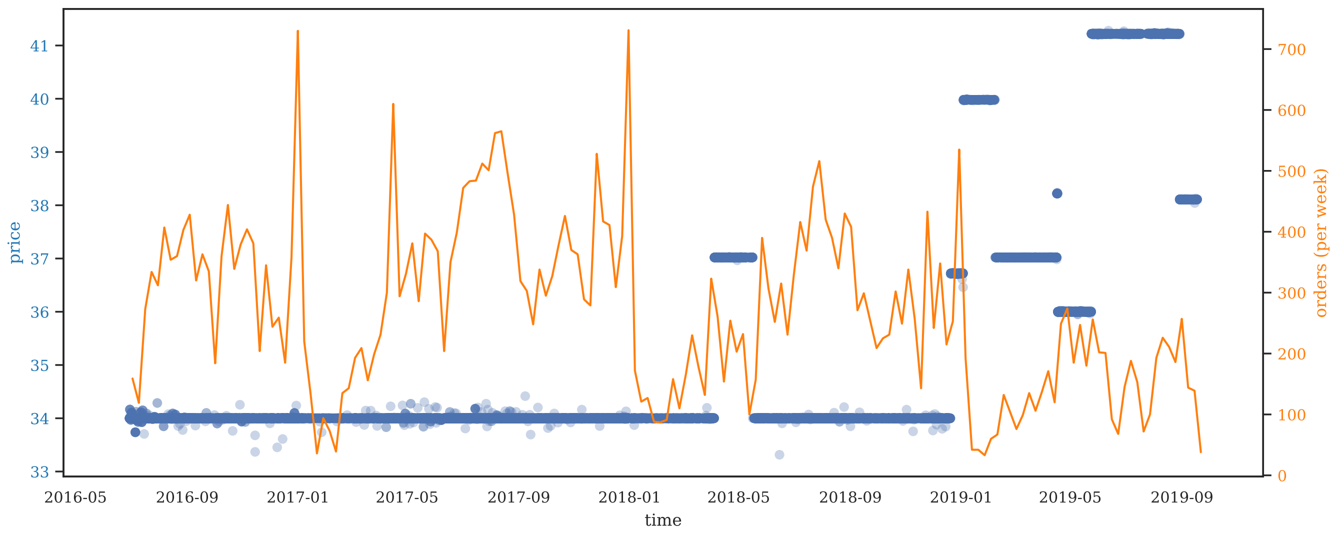


Figure 3.2: Price/Demand graph for a product over time

3.5. Baseline analysis

To determine the optimal price either manually or automatically, we need some way to describe the price demand relation. This chapter goes over a simple analysis that can be performed with the data retrieved in this chapter. Because of the nature of the historical data and simplicity of the analysis, some concerns naturally arise. These are addressed in section 3.5.2 and keep coming back throughout the thesis. While we may not be able to avoid all problems that arise when using historical/observational data, chapters 4, 5 and 6 each tackle a problem and introduce ways to compensate.

3.5.1. Price elasticity

The baseline analysis starts with plotting the historical prices derived in section 3.3 and the historical demand from section 3.4. Price posted time differs for each price point. As such prices with a longer posted time contain more information and are expected to give a more robust estimation of the demand. Therefore the posted price time is represented as the weight of the data-point in the price elasticity regression.

Figure 3.3 shows an example of such a price elasticity plot with the results of a weighted least-squares linear regression (plotted in red) on sales per hour using price as the independent variable. The X-axis shows the price while the Y-axis shows the sales per hour for the venue under that price. The size of the dots indicates the total posted price time for that price (see Terminology). This size is the weight value in the weighted least squared analysis. The F-statistic probability of this regression is 0.0141, indicating roughly a for this venue a 1 in 71 chance ($\frac{1}{\text{F-stat}}$) that the null hypothesis is true. The null hypothesis, in this case, says that there is no correlation between the independent variable (price) and the dependent variable (demand).

This analysis has been performed on 10 venues[6] passing the following criteria:

- The venue has at least 2 years of data
- The venue has at least 4 price changes with a posted price time of at least 2 months
- There are no obvious irregularities or gaps in the historical data for this venue.

The results of the baseline regression analysis of the 10 test venues can be seen in Table 3.1. The regression parameters are discussed in section 3.5.2.

The F probability for the venues shows that the chance of the price and demand being completely uncorrelated is relatively low (below 10%) for 4 venues, but above 10% for the other 6, with 3 venues having an above 80% chance of the price and demand being completely uncorrelated. The reason for this and how to mitigate this problem are discussed in section 3.5.2.

[6]the historical data corresponding to these venues has been manually extracted and cleaned

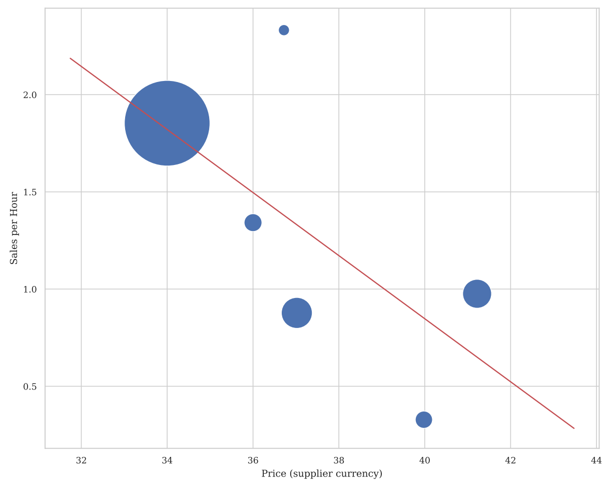


Figure 3.3: Weighted least squares regression on price elasticity graph for Venue 7

The R-Squared metric gives an indication of how well the variance of the independent variable can be explained by the regression. A value of 0.734 suggests that ≈ 73% of the variance seen in the price can be explained by the demand. High R-squared values for linear regression on only one dependent variable indicate that the regression parameters sufficiently indicate the price elasticity.

The Durbin-Watson value represents the likelihood of the residuals having a first-order autoregression component. The regression in this analysis is a simple linear regression on only one dependent variable. This regression was chosen to avoid over-fitting on a dataset with a small number of observations (see Table 3.1). Because of this, we can be happy with a high R-Squared value without worrying about the regression modelling artefacts in the data. Unfortunately, this also means that we might be completely missing some structure that is well represented in the data (a quadratic structure for example). The best way to detect this would be to look at the deviation of the points from the line. and see if there is a non-random pattern that can be explained by a more complex model. As a single value indicator, the Durbin-Watson value is able to highlight instances where the model could be improved by an autoregression component. Values under 0.8 generally indicate likely autocorrelation. None of the regressions has a Durbin-Watson value that warrants a more complex model. The residual plots also show no indication of a missed structure in the data.

In short: the F probability indicates whether there price and demand in this analysis are completely uncorrelated. the R-Squared value indicates explainability with the current model (linear regression). And the Durbin-Watson value indicates whether a more complex model (including an autoregressive component) on

	Regression	F probability	R-Squared	Durbin-Watson	No. Observations
Venue Name					
Venue 1	-0.02X + 2.23	0.9215	0.0021	2.7539	7
Venue 2	0.03X + 0.35	0.8288	0.0131	2.0748	6
Venue 3	0.07X - 1.16	0.0016	0.8845	2.2750	7
Venue 4	-0.59X + 18.45	0.1448	0.3735	2.2371	7
Venue 5	-0.12X + 3.31	0.0373	0.8102	3.2020	5
Venue 6	0.29X - 1.99	0.1891	0.2321	2.2399	9
Venue 7	-0.16X + 7.36	0.0291	0.7348	2.3819	6
Venue 8	-0.04X + 2.32	0.8192	0.0094	1.4134	8
Venue 9	0.24X - 2.42	0.3508	0.1457	2.9622	8
Venue 10	-0.00X + 0.09	0.0531	0.8966	2.7098	4

Table 3.1: Results of the baseline analysis

this representation would increase explainability.

3.5.2. Discussion

Section 3.5.1 has justified regression on the data. This section aims to explain the meaning of the fitted regression parameters and to what extent they can be used to determine the price for a venue.

As discussed in the previous section, we have 6 venues with a significant chance of having uncorrelated price and demand (more than 10%). Of those other 4, one indicates a constant demand, regardless of price, while another indicates an increase in demand with increased sales (the plots show this more clearly and can be found in appendix A). The plots and F probability scores show that replacing the linear regression model with a more complex one would not solve the issue. It is likely a problem with the way our data is represented. In some cases, it is possible that we simply do not have sufficient data. The price demand relation might very well be constant over the small variation in price in our observed data but might react completely different outside these bounds.

In our case, it is likely that a variable to which we do not have access is influencing both the dependent and independent variable for most price changes. An example of such a hidden variable factor might be a renovation or expansion of a venue. Since this historical data is not accessible to us, it is not feasible to factor out this influence. However due to the highly seasonal nature of tickets for cultural venues another important hidden variable can be derived directly from the data. Chapter 4 aims to explore how the introduction of seasonality (among others) as a sort of instrumental variable affects the price-elasticity estimation results.

4
Demand model

This section discusses the construction of a demand model on the historical ticket sales per venue. The purpose of the demand model is to alleviate the problems with the baseline analysis discussed in Section 3.5.2. To achieve this one could use a difference in differences approach to determine the effect of a price change using a time-series forecasting model to estimate the counterfactual [5] (more on this in Section 5.2).

Another solution is to use a regression model that is decomposable and incorporates price information into the model. The relation of demand to price can, in this case, be contained within the model.

More concretely the objectives of this chapter are:

- To define the best choice model for forecasting demand.
- To define how this model is trained in an automatic way.
- To discuss the benefits of a decomposable regression model
- To introduce a novel price module into the model
- To evaluate this module on observational data and discusses it's limitations.

Appendix B shows auxiliary plots for this chapter.

4.1. Model selection

The data in the problem domain concerns the sale data of over 3000 venues distributed globally. Ideally, price-elasticity information can be derived from venues without the need of a time-series specialist modelling each venue individually. Therefore 3 general purpose time series forecasting models are compared in this chapter.

To determine the best compatibility ARIMA, Exponential smoothing and Prophet models are trained on 6 selected venues with adequate sales data. Nested 2-fold time-series cross-validation (see below) is used to compare hyperparameters of the models. Hyperparameter optimization is done using a random walk on the hyperparameter grid which can be found in Appendix B.1.

4.1.1. Nested 2-fold time-series crossvalidation

Nested 2-fold time-series cross-validation is chosen to compare the performance of models and to avoid overfitting. By averaging over the results from multiple folds we get a better indication of the performance of the model on unseen data, alleviating concerns that the results on the chosen train-test split are an anomaly [30].

Time-series cross-validation in this context means that we use a blocked form of cross-validation. Since regular cross-validation requires data to be independent and identically distributed and in our case we expect values in the time series to correlate with previous and future values we have chosen to implement a blocked form as proposed in Snijders [28]. This method has been extensively tested by Bergmeir and Benítez [3] and is one of the recommended approaches for a robust time-series cross-validation. Our implementation 'grows' the train set with each fold where the test set is always the last set chronologically. This approach is illustrated in Figure 4.1

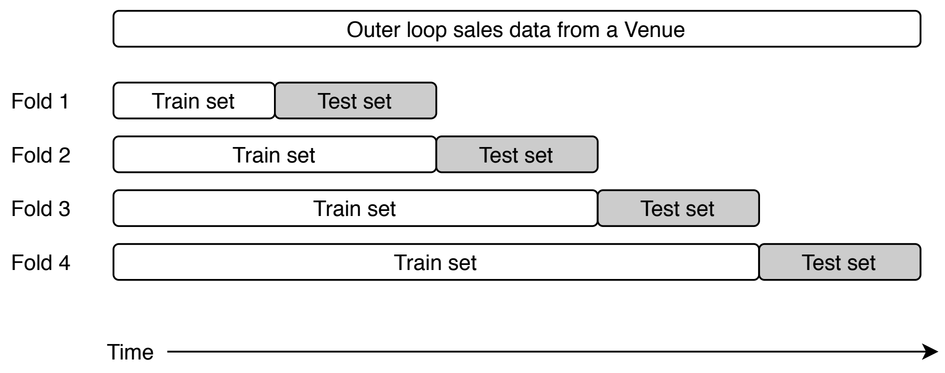


Figure 4.1: Illustration of the inner loop in our crossvalidation method

In single-level k-fold cross-validation, the available data would be divided into k folds. The performance of the model would be derived from the performance of the many models trained in each fold. The resulting performance metric could be used in hyperparameter optimization to compare against models with different hyperparameters. Nested cross-validation or double cross-validation takes this one step further by adding an additional loop [30]. Hyperparameter optimization is now done inside each fold, leaving the performance metric derived from the outer loop as a score indicating the quality of the pipeline. This is preferable in our case since we aim to compare the performance of model including hyperparameter optimization. Since the best hyperparameters for the model on all trained data might differ from that of one of the folds, we can not just do hyperparameter optimization once on the entire dataset.

4.1.2. Performance metrics

Both SMAPE and RMSE are used in the process of comparing the models. RMSE is defined as in equation 4.1 where n is the total number of samples. SMAPE is defined as in equation 4.2.

$$RMSE = \sqrt{\sum_{i=1}^{n} (\text{Predicted}_i - \text{Actual}_i)^2} \tag{4.1}$$

$$SMAPE = \frac{100\%}{n} * \sum_{i=1}^{n} \frac{|\text{Predicted}_i - \text{Actual}_i|}{(|\text{Predicted}_i| - |\text{Actual}_i|)/2} \tag{4.2}$$

RMSE is used to compare between model configurations on a dataset, but for comparing the goodness of fit of the model between datasets, SMAPE is used. Since RMSE is scale-dependent comparing the goodness of fit between models on different datasets would lead to misinterpretations. e.g., a more popular venue with higher ticket sales per day might produce a larger RMSE compared to a less popular venue, even when the percentual error it makes on the popular venue is smaller.

4.2. Prophet

Prophet is an implementation of a time series model first introduced in Taylor and Letham [31]. The authors claim their model has a high amount of flexibility, making it applicable to all venues in our data. The implementation of the model in the original paper has been continuously expanded since its introduction in 2017. This section details the model, as well as the chosen configuration of components and aims to explain why this generalizes over all venues in the data.

4.2.1. Overview

Prophet builts on the decomposable time series model from Harvey and Peters [16] with three main components: trend, seasonality and holiday effects. The original paper describes the relation of these components in an additive way:

$$y(t) = g(t) + s(t) + h(t) + \epsilon \tag{4.3}$$

With $g(t)$ representing the trend in the data, $s(t)$ the combination of seasonalities with different periods and $h(t)$ the effects of holidays. $y(t)$ represents the actual value and the error term ϵ the difference between the predicted and actual value. The model configuration implemented in this paper allows for multiplicative instead of additive effects between components as well. This is modelled as a hyperparameter for each component on a per venue basis.

The Prophet implementation relies on Stan, a probabilistic programming language in which the core statistical model is described [6]. This allows us to the MAP (Maximum a Posteriori) method during model fitting to estimate the model parameters that best explain the data. Stan also supports MCMC (Markov chain Monte Carlo) sampling to gain a greater understanding of the distribution of each parameter. This option significantly increases the training time of the model.

4.2.2. Trend

Taylor describes two possible ways of modelling the general trend of the data. Logistic and Linear growth, both incorporating a changing trend over time.

Logistic growth represents a non-linear saturating growth with a carrying capacity C. The function increases with a decreasing slope until it reaches C. An exact definition is:

$$g(t) = \frac{C}{1 + exp(-k(t-m))}$$

where k represents the growth rate and m the offset parameter. For this implementation of trend to work on our problem domain, we require an estimation for C. Taking a combination of the population around a venue along with the expected international tourism numbers would give such an estimation. The sales for any given venue are low in comparison, which mitigates the advantages compared to using a linear trend with changepoints. For this reason, we have chosen to model trend as linear growth.

For a visual representation of the implemented linear trend see Figure 4.2. Mathematically this trend can be represented as in equation 4.4.

$$g(t) = (k + a(t)^T\sigma)t + (m + a(t)^T\gamma) \tag{4.4}$$

where k represents the growth rate and m the offset parameter. Suppose there are S changepoints at times $j = 1,...,S$. $\sigma \in \mathbb{R}^S$ is then defined as a vector of rate adjustments at time s_j. The rate at time t is then the base rate k plus all of the adjustments up to point s_j. This is represented more cleanly in the vector $a(t) \in \{0,1\}^S$ such that

$$a_j(t) = \begin{cases} 1, & \text{if} t \geq s_j \\ 0, & \text{otherwise} \end{cases}$$

γ is then defined as $-s_j\sigma_j$ to make the function continuous.

This way of specifying the trend allows for the model to derive a generalized linear trend over the data that can change as a reaction to events that happened within the time covered by the data. It is possible to specify these changepoints manually, but as this is not information that is available to us, we specify a large number of changepoints and a sparse prior on σ facilitating automatic changepoints detection.

4.2.3. Seasonality

Time series data is often influenced by regular patterns repeated over time. Such patterns are generally referred to as seasonal variation or seasonality. Since our case covers daily sales data in the tourism industry spanning multiple years it is highly likely that these patterns are present in the data and can cause a significant influence.

Prophet allows for us to model this seasonality as periodic functions of the time using Fourier series. In our case, we define two patterns: one with a weekly and one with a yearly period. Equation 4.5 shows the formula which captures these patterns. Each pattern is represented as a function s which depicts seasonal variation over time represented as t.

$$s(t) = \sum_{n=1}^{N}\left(a_n\cos\left(\frac{2\pi nt}{P}\right) + b_n\sin\left(\frac{2\pi nt}{P}\right)\right) \tag{4.5}$$

P and N represent the period length and complexity respectively. Since we aim to capture yearly and weekly seasonality and one data-point represents a day in our data P is set to 365.25 in our yearly pattern and to 7 in our weekly pattern. N is set to 10 for our yearly and 3 for our weekly pattern as suggested by the original authors. The parameters $a_1, b_1, ..., a_N, b_N$ are fitted to the training data.

Because of the way this component is implemented it is possible for the model to explain effects caused by an unobserved variable as seasonality in problems with a training set size smaller than the period length for the largest pattern P_y. A training set longer than P_y allows the model to average the Fourier series over multiple periods. This is why we only consider data for which we can acquire a training set of at least $2P \approx 730$ days.

4.2.4. Holiday effects

To capture effects of holidays and events that are either not periodic (i.e. the at the point in time each seasonality period) or too short to be captured in the Fourier series (this is dependent on the regularization parameter N) prophet allows for a third component that assumes an independent effect for each date in a pre-determined list of dates. This component is commonly used to capture holidays and recurrent events. A forecast on data would require the dates of these holidays and events as input.

Even when events are non-recurring this component can increase the quality of the model by making sure that the effects of the one-off event are not skewing the seasonality or trend components of the model.

The implementation of this component is achieved by generating a matrix of regressors with binary values for the holiday. Each regressor is multiplied by a parameter corresponding to the change in the forecast, fitted to the training data. Holidays and events spanning multiple days are represented as a regressor with multiple active values.

4.2.5. Component decomposition

Figure 4.2(a) shows The different components outlined in section 4.2. Each component is represented as a graph showing its influence on the trend. The x-axis shows time over the period associated with its component. The y-axis can be either multiplicative or additive this is defined as a hyperparameter and is decided per venue (see 4.1). In this case, the components are combined in a multiplicative fashion.

4.3. Pricing using the demand model

With the current demand model, it should be possible to generate new price elasticity estimations that are not influenced by the general trend, seasonality or predicable outliers (such as holidays). Unfortunately, the model can be seen to incorrectly attribute the effect of price changes to the trend and sometimes even the seasonality. Since the model up until now does not have access to the historical price information it is not able to separate these effects.

From this section on we focus on how venue price over time can be incorporated into the demand model, allowing us to more accurately predict venue sales as well as determining the optimal price for desired demand. Several methods of including the price are discussed and compared.

4.3.1. Price regressor

The model as described in section 4.2.1 by equation 4.3 can be extended to form:

$$y(t) = g(t) + s(t) + h(t) + p(t) + \epsilon \tag{4.6}$$

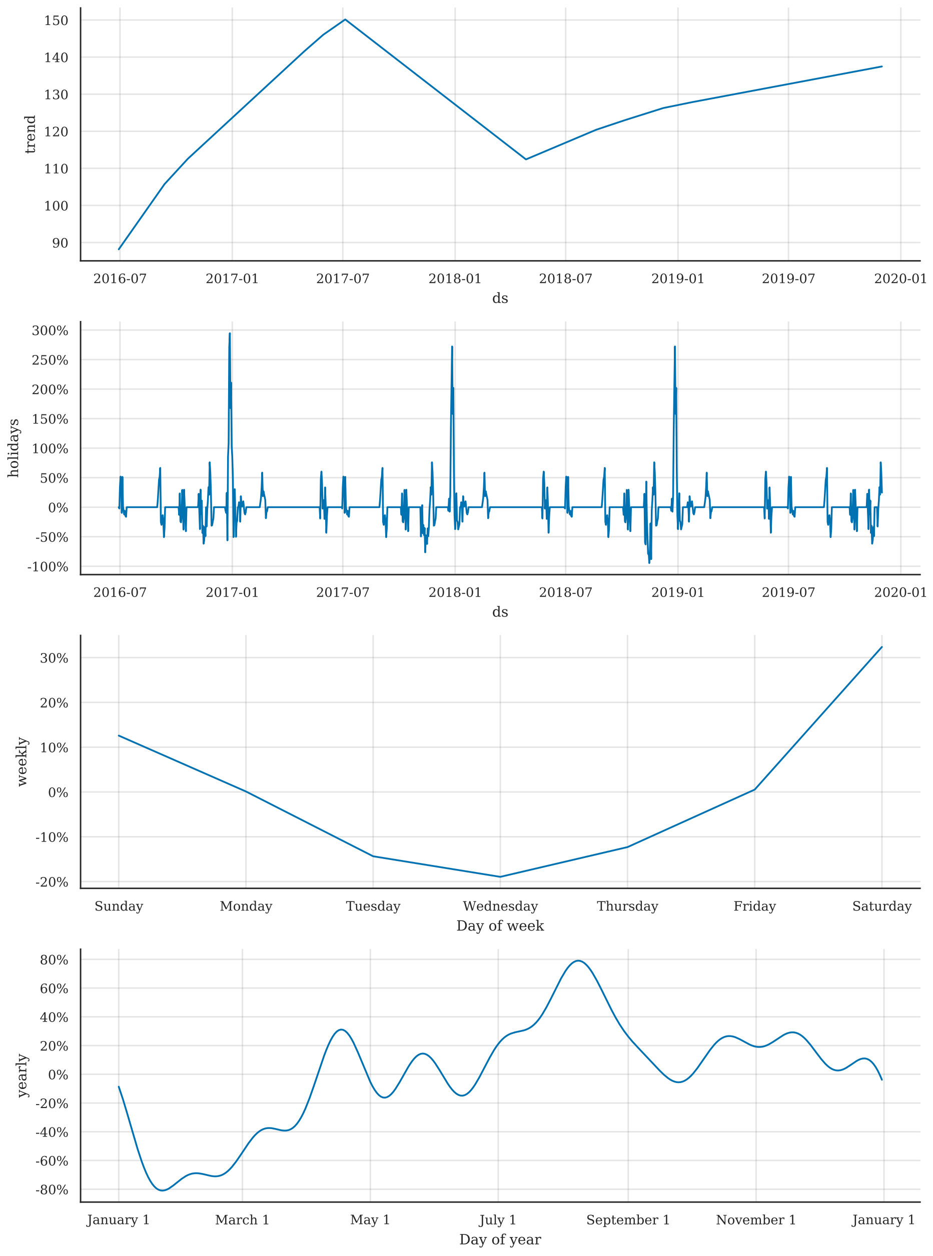
Where $p(t)$ represents a continuous variable describing the price for each day t. Since the actual historical price for t is not known the price per sale is retrieved as described in section 3.3 and averaged per day.

$$p(t) = \beta \cdot r_t \tag{4.7}$$

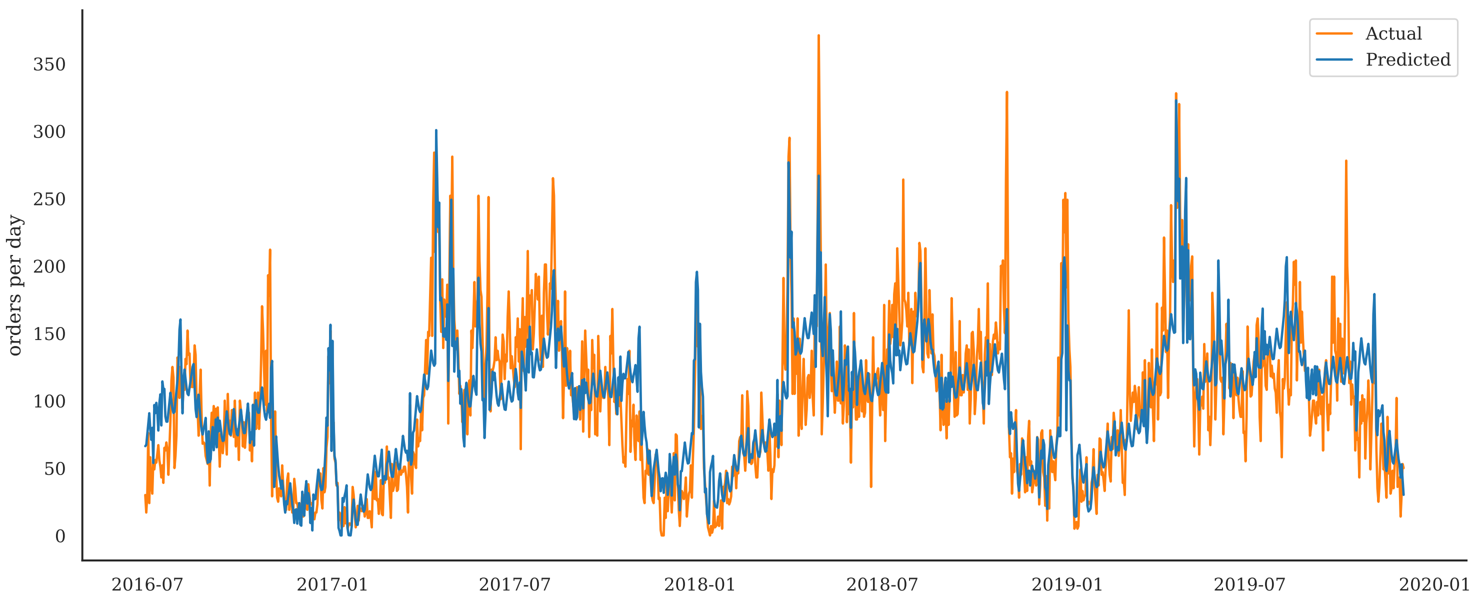
$p(t)$ can be defined as in equation 4.7 where $\mathbf{r}$ is a vector containing the estimated price per day and β a parameter to be fitted by the model. Such a definition is able to capture a linear relationship between price and demand. Section 4.3.3 shows whether this is sufficient and the remaining subsections (4.3.4, 4.3.5, 4.3.6) discuss variations of $p(t)$ that are able to capture more complex relationships.

4.3.2. Dynamic pricing

This means that to forecast we now need to extend the price vector $\mathbf{r}$ for future t's. It is possible to assume an unchanging price by extending $\mathbf{r}$ the last known historical value to forecast future sales $y(t)$, but we could also adjust the fitted model to allow for the desired demand as an input:



(a) Example of component decomposition for Venue 7



(b) Components combined to model per day ticket sales for Venue 7

Figure 4.2: Component decomposition for Venue 7

$$\begin{aligned} y(t) &= g(t) + s(t) + h(t) + p(t) \\ p(t) &= y(t) - (g(t) + s(t) + h(t)) \\ \beta \cdot \mathbf{r} &= y(t) - (g(t) + s(t) + h(t)) \\ \mathbf{r} &= \frac{y(t) - (g(t) + s(t) + h(t))}{m} \end{aligned} \tag{4.8}$$

In equation 4.8 ϵ is assumed to be 0 since our assumption is that in a well-fitted model, we can not further reduce this term. This leaves the right-hand side of the fitted model with only known parameters when the desired demand is supplied. This can either be a constant or a vector containing the desired amount of tickets sold per day. The model is then able to estimate what the corresponding price values per day should be, taking into account the learned trend, seasonality and holiday effects along with the price elasticity. This approach is only practical when the error term ϵ is negligible and $p(t)$ is accurately modelled, that is all the effects of price are modelled in $p(t)$ and only in $p(t)$.

4.3.3. Need for non-linearity

Equation 4.8 captures a linear relation of price to demand in the form a single value price-elasticity. It is possible to capture more complex effects of price by creating variations of the price module that include basis functions on the price.

By using basis function we can model $p(t)$ as linear combination of functions on the price regressor:

$$p(t) = \sum_{k=1}^{K} \beta_k \phi_k(\mathbf{r}) \tag{4.9}$$

With basis functions $\phi(\mathbf{r}), k = 1, .., K$ where $\boldsymbol{\beta}$ is a vector in $\mathbb{R}^k$ of fitted parameters. This method allows for non-parametric regression of the price that fits nicely within our existing model, allowing us to capture more complex relations by fitting a linear combination of functions [22]. One potential problem that this introduces is the ability for the model to over-fit on the training data, inferring false relations. Using cross-validation as described in section 4.1.1 we are able to quantify this effect.

4.3.4. Polynomial terms

One choice of basis is set $\phi(\mathbf{r}), k = 1, .., K$. This allows us to model polynomial relations between price and demand. Since we just have the one regressor this method is both straightforward to implement and gives us the freedom to estimate the price-demand relation in a non-parametric way.

The complexity of the data that this relation is able to fit can be adjusted with the parameter K. Setting K to $1, 2$ and 3 allows us to model linear, quadratic and cubic relations respectively. With higher K values comes more complexity and thus the ability to over-fit. To combat this we fit the model with increasing K values, starting from one. Each variation is evaluated using the nested cross-validation method as described in section 4.1.1.

It must also be taken into account that higher-order polynomials may return unstable results when extrapolating on data [33]. For us, this means that the error estimate of the price-elasticity grows quickly for prices further removed from prices in the training data, and this effect is more pronounced with higher values of K. In fact, the error is estimated to increase with the degree of the polynomial (I.e. K) [33]. There are several ways to mitigate this problem, including using piecewise or Bernstein polynomials [24]. This might be necessary for venues for which the price-elasticity is best defined with a higher-order polynomial and the aim is to predict the demand for an abnormal price. Piecewise or Bernstein polynomials are good additions for future work on this method of modelling price.

4.3.5. Lagged price

Consumers are expected not only to react to the current price of the venue but also to a change in price (e.g. discounts or special offers). For the tourism industry, this effect estimated to be less pronounced compared to retail. To describe this effect we allow the model access to a time-shifted version of $\boldsymbol{r}$. In this case, we introduce a lagged price in the model by shifting $\boldsymbol{r}$ such that $\hat{r}_t^l = r_{t-l}$ where l is the number of days in the past the model is able to account for. We include l versions of the lagged $\boldsymbol{r}$ in the model such that when $l = 3$ the lagged price is described by $\hat{\boldsymbol{r}}^1 + \hat{\boldsymbol{r}}^2 + \hat{\boldsymbol{r}}^3$.

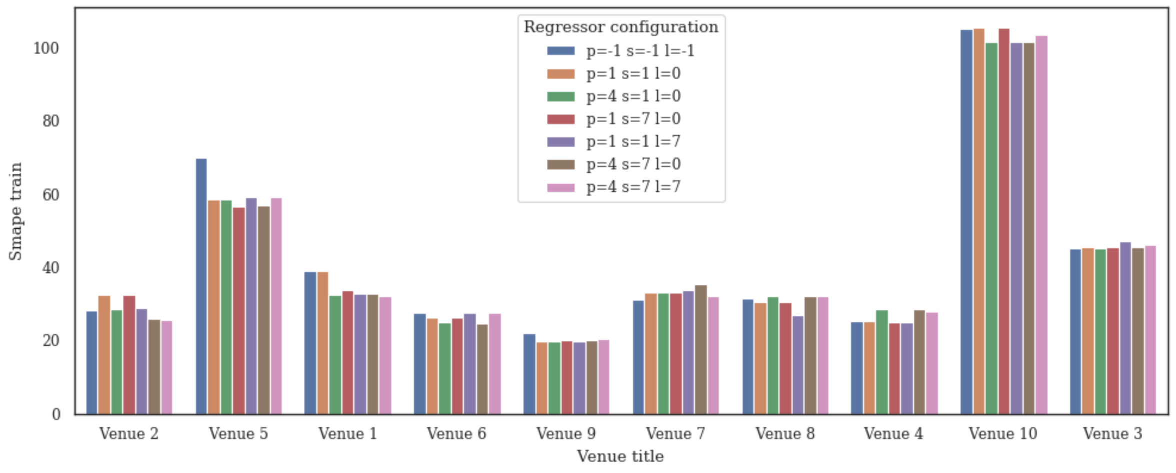


Figure 4.3: SMAPE score on the training data for each venue on a subset of configurations for the price regressor

4.3.6. Smoothing

When modelling venues in this way we often found increasing the number of polynomial terms and lagged price values resulted in a combined regressor that resembled a more smooth version of the original price. For this reason, we decided to include a variation of the model that used a smoothed version of the price regressor instead of the original. The smoothing was done by taking the moving average of the price with a window size of 7 days, this smooths outliers and effects of one-time events to some degree. Since 7 days was chosen as the window size weekly seasonality (recurring sales differences within a week, see 4.2.3) will also not be an issue.

4.4. Results

To see whether the price regressor for a venue is able to explain demand we can look at whether the extracted price as a time-series, applied with different variations of the modifications discussed above improves the performance of the model. For this, we look at the SMAPE scores of the model on the training data. For each venue, we show the model performance without any additional regressors (depicted as `p:-1 m:-1 l:-1`). We then try a combination of the following parameters:

- `p=4` indicating that we not only include the price regressor, but the polynomial basis functions to a degree of 4 as described in section 4.3.4 for or `p=1` indicating that for this result no polynomial basis functions have been added.

- `s=7` indicates smoothing (rolling window mean) as described by section 4.3.6 by 7 days or `s=1` indicating that we do not apply smoothing in this configuration

- `l=7` indicates that 7 lagged variables have been added as described in section 4.3.5. Each variable has a lag of k days where k=1,2,...,l. `l=0` means that no lagged variables have been added.

Figure 4.3 shows the SMAPE scores obtained on the training set for these configurations. Note that since SMAPE indicates relative error it is possible to compare scores between venues.

Each of the venues seems to show next to no difference in performance between configuration. This is expected as we are looking at the performance of the model on the training dataset. Effects price changes may be already (over)fitted in the model with the `p=-1 s=-1 l=-1` configuration. It is also possible that there is no consistent effect on the demand deducible from the observational data.

Venue 10 is an example of a venue meeting the price change criteria from section 3.5.1 but with significantly lower sales than other venues. This makes it harder for the model to extract useful patterns.

To combat overfitting we also include scores on the test sets from the nested cross-validation approach (section 4.1.1). These can be seen in Figure 4.4. As expected the error rate is generally higher for unseen data. For some venues, such as venue 2, 3 and 4, the presence of the pricing regressors does not affect the performance of the model much. This could be due to a lack of price changes in the test periods during cross-validation. It is also possible that the model recognizes that the price data is available, but is not able to fit

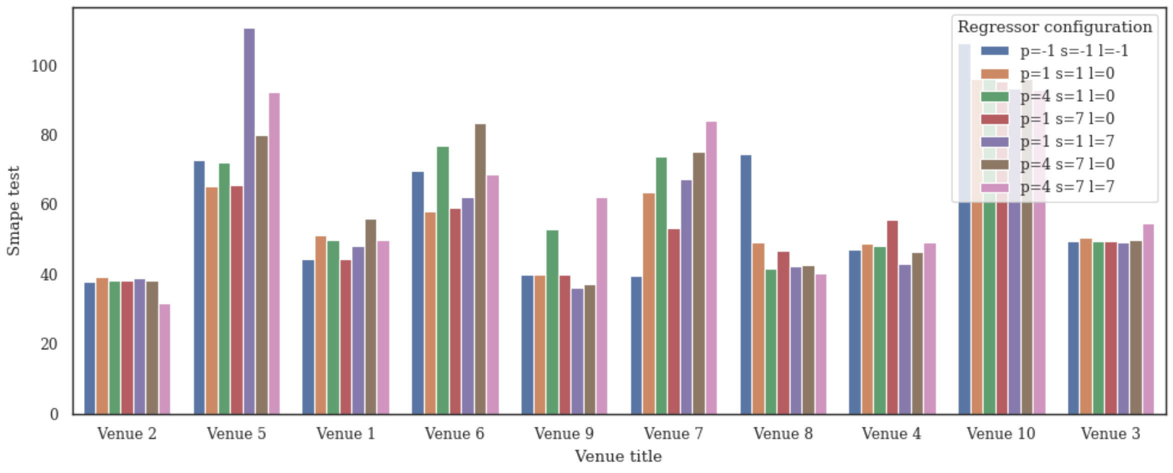


Figure 4.4: SMAPE score for each venue on a subset of configurations for the price regressor using the test sets from crossvalidation.

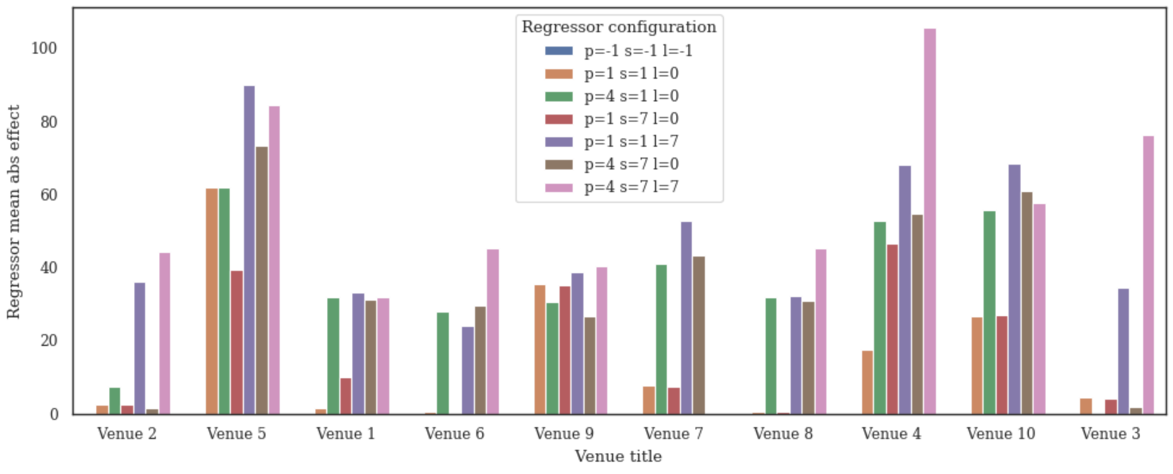


Figure 4.5: The mean of the absolute regressor influence.

the regressors to the data in a significant way. For this refer to figure 4.5. This would support the hypothesis above stating that no consistent price demand relation can be drawn from the available data.

Of course, it could also be possible that our model is not complex enough to extract the demand price relation on these venues. Intuitively we are hesitant to increase the model complexity even more due to the low amount of price changes available to us. This intuition is confirmed when looking at Figure 4.4. Configurations with increasing complexity generally return a higher error rate. Since this cannot be said for Figure 4.3 we can conclude that these more complex configurations are overfitting the model on the training data.

Figure 4.5 shows the mean of the absolute (since the effect can be negative) effect that the price regressors have on the model predictions per configuration. The effect is expressed as a percentage of the total ticket estimation. A high effect for configurations containing a price regressor in combination with a performance increase of the model indicates that: the price regressor can explain some of the variation in the observational data. A comparison with the cross-validation results ensures that the model does not use the extra complexity gained by adding regressors to (over)fit on the error term.

4.5. Conclusion

In this chapter, we have introduced a demand model to on one side model the demand by breaking it down into separable, logical modules and on the other hand forecast future demand for dynamic pricing purposes (see section 4.3.2. The model is autoregressive since we only have sales data available to us. A consequence of this is that it is hard to identify and separate individual influences to the demand. The focus of this chapter has been to separate the influence of price from all other factors influencing demand.

The results section (4.4 compared the performance the model in 7 different configurations over 10 different venues. Because of the complications with managing a platform with venues all over the world, finding venues with sufficient stable data and price changes is not an easy task. It is therefore in the interests of Tiqets to keep this data as clean as possible. More price changes in venue, even small, and especially for the sake of experimentation are highly encouraged. Luckily the company since the start of this thesis improved the process of changing venue prices for internal price experts. This will hopefully promote the practice of changing the prices for venues experimentally.

A possible improvement to the method from this chapter would be to introduce more complexity in the price module. This can (among other ways) be done by introducing new basis functions to the price regressor. Figure 4.4 indicates that with the amount of data currently available this would likely lead to overfitting.

Implementing an automatic dynamic pricing system based on these techniques will require both the accuracy of the model and the accuracy of the price module within the model to be as high as possible. While the former is easy to estimate and measure after the fact, the latter is more troublesome. Experiments such as the one from section 5.1 might give some indication. Plots such as the ones from section 4.4 allow us to quantify the accuracy of the model as well as the effect of the price module on modelled demand.

5 Causality

This chapter explores causality within our data. Section 5.1 explores a pricing experiment that could best determine the causal effect per price change, while section 5.2 looks at the causal effects of price changes in historical data.

More concretely the objectives of this chapter are:

- To define a randomized price experiment best fitting to our problem domain and discuss how this eliminates the largest problem with observational data.
- To adapt a method from Brodersen et al. [5] that uses a regression model to estimate the counterfactual in causal analysis.
- To use this counterfactual to determine both the effect size and shape of individual price changes.
- To evaluate this method, discussing both its strengths and limitations.

Appendix C shows auxiliary plots for this chapter.

5.1. Randomized price experiment

Section 3.5.2 briefly discusses the endogeneity that arises when using observational data such as the data explored up until now. Trying to extract the effects of a price or change in price purely from the sales data as done previously is prone to errors caused by hidden variables. We can see this in the demand model from section 4.3 failing to fit a price-demand relation that is predictive on unseen data, or even more simply in the baseline analysis giving a positive price-elasticity (increased demand with an increase in price) for some venue. The latter could, for example, be caused by the growing popularity of a venue, which in turn increases the price. A randomized price experiment can be performed with co-operating venues to obtain price-change data without a hidden cause for the change. Furthermore, since a randomized price experiment should not be bothered by these omitted variables the value of observational data in its current form can be quantified simply by comparing previous methods on the observational data to the same methods using the data gathered from the experiment as input.

It is worthwhile to note that this experiment was planned to be included in the scope of the thesis. Unfortunately due to the COVID-19 crisis participating venues were forced to shut down. Section 5.1.3 summarizes expected results

5.1.1. Related experiments

The details of the experiment were taken from. This paper was chosen as the guideline over several other papers outlining similar experiments including the ones from Ashraf et al. [2], Karlan and Zinman [21] and Gneezy and Rustichini [14]. The preference for the paper by Fisher et al. is because of out of these papers it is the one with the most similar product and distribution method as the one used by Tiqets. That is consumable products sold in an online environment.

Treatment Level	Change in Price
High	+10%
Medium high	+5%
Medium	0%
Medium low	-5%
Low	-10%

Table 5.1: Possible treatment levels

Ferreira et al. [9] and Gaur and Fisher [13] do experiment on consumable products and are therefore more relevant. The former even concerns an online retailer. Ferreira et al. [9] limit their scope to static prices for flash sales on new products and Gaur and Fisher [13] run their experiment in brick-and-mortar stores. The paper by Fisher is also more recent than the previously mentioned papers.

5.1.2. Setup

Participating venues are split into two groups: group A and group B. Both groups experience different treatment during the experiment. Each venue in the experiments gets a randomly assigned treatment level which changes every 3 days. There are five possible treatment levels each with a corresponding change in price. These can be found in Table 5.1.

Both Group A and B apply the price-change of their treatment group to a so-called base price. The difference in the groups lies in how venues in that group determine their base price. Group A changes its price according to the last price before the experiment started. This is defined as the mean of the prices estimated for the last 8 weeks before the experiment. Group B applies price-changes to the current average competitor price. This price is scraped daily from the website of each competitor.

After 3 days of treatment, each venue gets assigned a different randomized treatment group. The group to which a venue belongs (A or B) stays the same. For group A, previous treatments will not affect a venue's base price. The treatment is applied for 3 consecutive days instead of differing per day to avoid customers catching on to and possibly exploiting the price experiment. Note that while the price of the venues in group A stays constant during these 3 days, the price of the venue's in group B is allowed to change.

The duration of the experiment should be at least 30 days. This duration is derived from the results of Fisher et al. [10] Longer experiments will give us more samples and therefore more complete information. During the initial experiment, we can decide to expand or reduce the length based on power analysis to determine the trade-off between the significance of the results and the costs for the venue of extending the experiment.

5.1.3. Expectations

The results of the experiments can be used as input for the baseline analysis from section 3.5.1. The results from Fisher et al. [10] show a significant improvement in price elasticity estimate created from this data as compared to observational data.

Fisher et al. show even more improvement when including competitor information. Our models do not include competitor price information since up until the experiment this data was not available to us. The data collected from the experiment will allow us to directly implement the consumer choice model described in their paper. This would yield per venue price coefficient as well as a per competitor consumer bias. It is worthwhile to note that because of the different domains in which this experiment is executed, it would be wrong to assume a similar improvement before seeing the data.

Since competitor prices have been directly observed in a period where the prices of Tiqets venues rapidly change, we are also able to see to which extend and with what speed the competitors respond to a price change for a venue.

Lastly, the results of this analysis can be incorporated into the demand model from chapter 4. Competitor price can be added as a regressor in a linear fashion or can be used to learn more complex relationships as in section 4.3.1. In fact, it might be useful to scrape and incorporate competitor data into the model continuously. The analysis on this experiment should quantify not only how much the effect of hidden variables distorts historical data, but also how much value there is in storing competitor data[1]. Incorporating data from

[1]Given that there is no endogeneity caused by hidden variables

the demand model might also reduce the effects of holidays or high seasonal peaks on the data, which would be especially noticeable in experiments with a short duration.

5.2. Causal inference from observational data

As explained in section 5.1 a randomized price experiment of sufficient length (see section 5.1.1) would remove the problem of hidden variables, but there are several other ways to achieve this result. One way that is especially popular in interface design is called A/B testing. A/B testing randomly splits customers into a control and treatment group. To quantify the effects of a change (in our case a price change, in the case of interface design a new version of the interface or even a small modification to it) it is possible to compare the results directly. Since the only thing that changes is the one change that you are interested in, a causal effect is directly visible. Aside from the technical obstacle of showing different prices to different users on a platform that does not require user login before reaching a product page, A/B testing for pricing is often considered a bad idea when trying to build trust in a customer base and several companies have received a lot of consumer backlash for doing so [8]. The randomized price experiment circumvents the problem of showing multiple prices by varying the prices over time instead of users.

A common way to minimize the effects of hidden variables in observational data is called difference in differences. By comparing the treatment group with a control group that has a similar trend before the price change it is possible to remove any effects of hidden variables that are present in both groups from the analysis. This method requires a control group with a similar trend to the venue with the price change. Trend in this context means more than just the general trend as described in section 4.2.2 and includes seasonality and any other patterns the data may exhibit. The control group needs to be large enough that averaging would remove the effects of any hidden variables influencing the data after the price change date (since we can not check for that) and the treatment group has the same constraints. Since a specific price change generally only happens to one venue at a time the treatment group would be very vulnerable to hidden variable effects after the price change. Other problems with the difference in differences method stem from the fact that we're using time series data [5, 15].

5.2.1. Model based causal inference

Both A/B testing and difference in differences use a control group te estimate what would have happened to the treatment group if the treatment had never occurred (i.e. the counterfactual). Brodersen et al. [5] infer causal impact by predicting the counterfactual using bayesian structural time-series models. The authors train a model to predict the dependent variable (demand in our case) using a model based on a set of variables which does not include the independent variable (price). Any change in the independent variable should not be reflected in the model. This allows us to see the effect of the change in the residuals of the model as illustrated by figure 5.1. This method gives a lot of freedom in modelling the counterfactual and allows us to see the effect of a change in the independent variable evolve over time, as opposed to a more simple before/after analysis. This could be especially useful in our case since it would answer the question of whether a customer reacts to the event of a price change or simply to the value of the price.

In the case of figure 5.1 the intervention period lasts 6 weeks. In this figure Brodersen et al analyse the effect of an advertisement campaign on search-related website visits. 5.1(a) shows the model fit pre-intervention and its predictions afterwards. 5.1(b) shows the residuals by taking the actual website clicks over time and subtracting the model predictions. One thing to note about this analysis is that the mean of the residuals pre and post-intervention tend to 0, while the model predictions during the intervention are consistently lower than the actual amount of clicks.

Concerns

Brodersen uses a Bayesian state-space model for his causal analysis. In the paper, it is mentioned that this model could be substituted. In this chapter, we apply a modified version of this method using our autoregressive demand model from section 4.2. A few things to note when substituting for this model are:

- The model uses the dependent variable as training data, it is, therefore, necessary to exclude any dates after the price change in the training set, as those can contain effects of the independent variable which the model may then try to fit.
- Hidden variables that only affect the demand after the price change can still influence the results. The model can not account for these effects since they are outside of the model's training data.

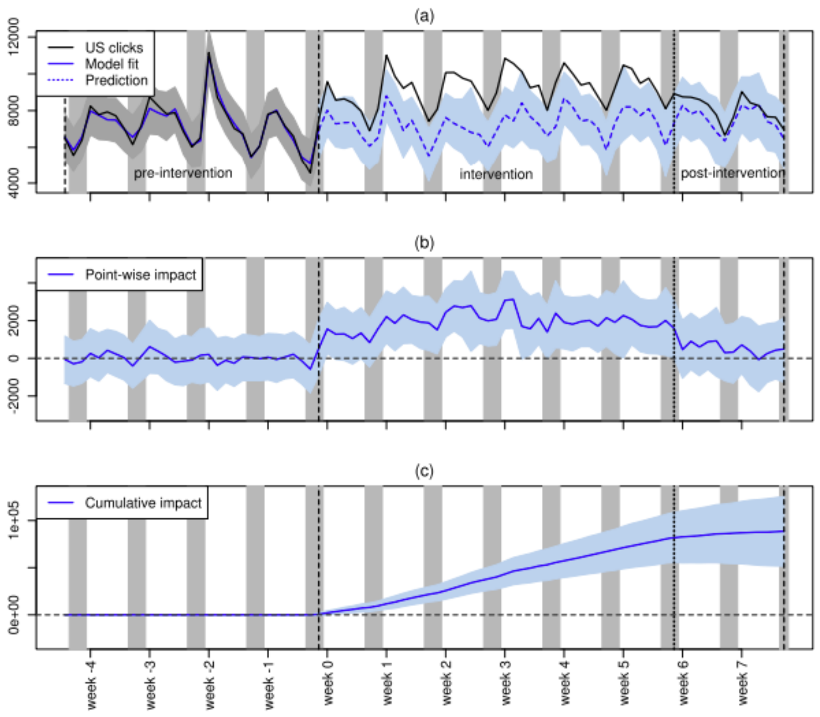


Figure 5.1: A visual explanation of causal inferrence using model residuals taken from Brodersen et al. [5]

- We expect the absolute residuals of the model to increase on data outside of the training set, it is, therefore, necessary to separate these effects from the effects of the independent variable.

Example of a price change indicating causality
Example of a price change indicating causality
Figure 5.2 shows 4 plots showing a causality analysis of a price change in Venue 1. The first plot shows an overview of all sales and pricing data for the venue. Plot sales are aggregated by week to make it easier to see patterns on a larger timescale. Sales are indicated by the orange line, while purchases at a price are shown using a blue scatterplot estimated price change dates are indicated by a horizontal line. These price changes are only recognized when the price is considered stable for at least 20 days. This both ensures that we have enough sales data and minimizes the chance of a one-off event not captured by the holidays component of the model (section 4.2.4) determining both the sales and price, as this would lead to skewed results.

The second plot (b) shows the model fit on the sales data. Trained until 10 days before a price change. The training set cut-off[2] is indicated by a blue horizontal line.

(c) shows the residuals. These are calculated as:

$$R_i = Y_i - \hat{Y}_i$$

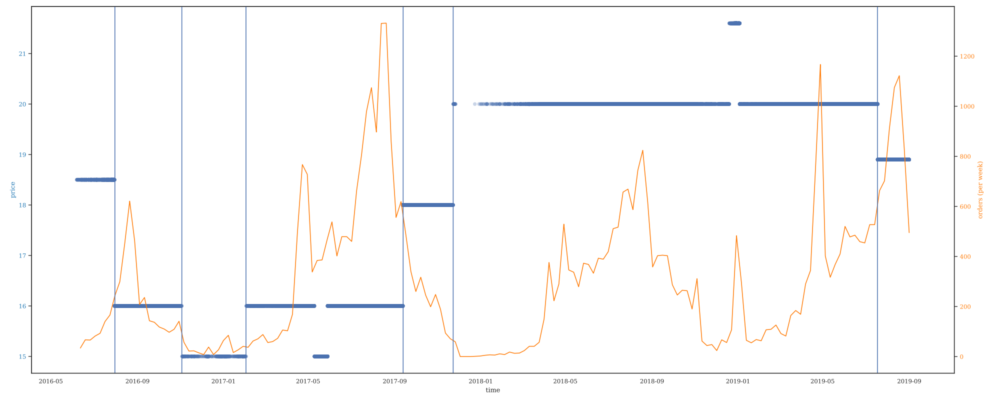
With $Y_1, Y_2, ..., Y_n$ indicating the daily sales for all n days in both the training and test data. $\hat{Y}$ is defined similarly for the model predictions. The two horizontal lines indicate the date training set cut-off date and date of the price change respectively.

A price change with a strong causal effect should show:

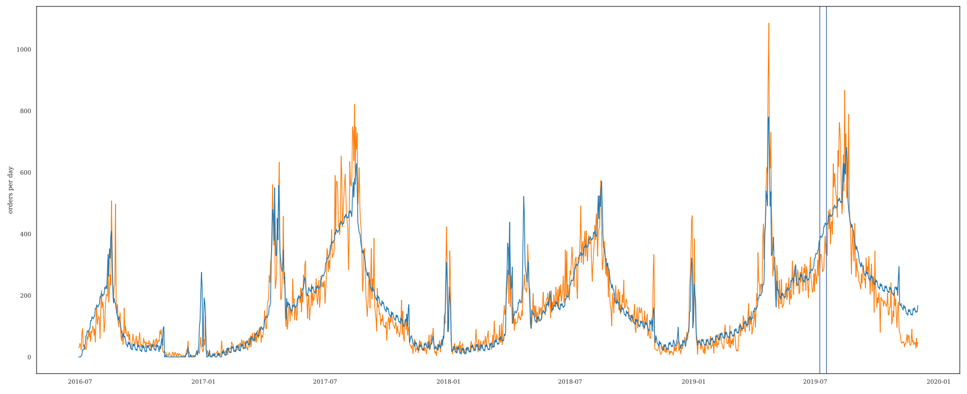
- A consistent deviation in the residuals after the price change. This indicates that without knowledge of the price change the model expects the sales to be either consistently lower or higher than the actual sales.

- No or almost no such deviation in the 10 days after the training set cut-off date and before the price change. This shows that the consistent deviation seen in the plot is not simply because of poor prediction accuracy. Since the model is completely oblivious to the price change date a deviation starting not

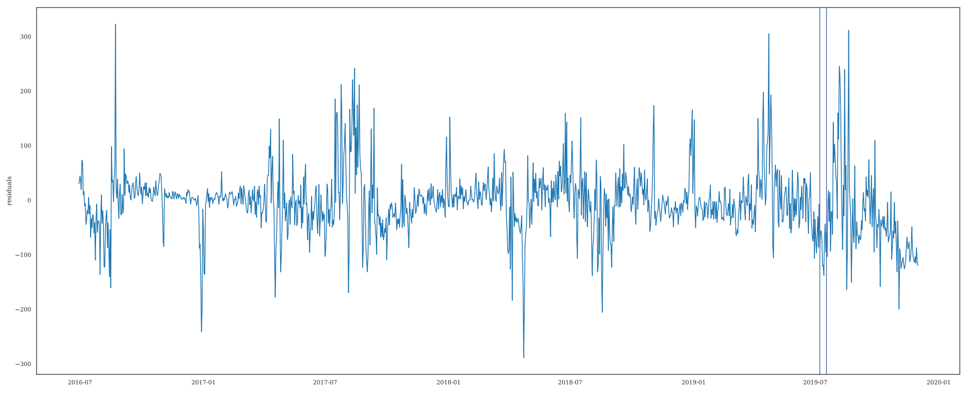
[2]All data and only data before the training set cut-off is used to fit the model parameters



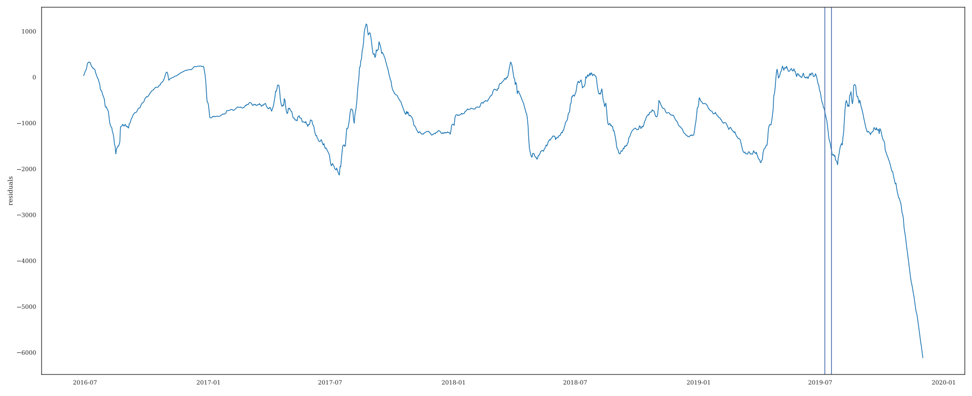
(a) Sales and pricing data for Venue 1. Vertical lines indicate estimated price changes.



(b) Sales data for Venue 1. Orange indicates historical sales while blue indicates the model output. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.



(c) Residuals for Venue 1. Residuals for a day are calculated by the actual sales for that day minus the model prediction. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.



(d) Cumulative residuals for Venue 1. The cumulative residuals for a day is defined as the sum of all previous residuals. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.

Figure 5.2: Causality plots for Venue 1 on its last price change.

after the training cut-off date but exactly on the price change date further supports the presence of a causality.

In this case, the residuals are consistently higher than 0 after the price change, while they were centred around 0 before (this is an indication of a good model fit). This indicates that $Y_i > \hat{Y}_i$ for values after the price change. That is to say that the actual sales are consistently higher than the predicted sales by the model. This is expected when lowering the price.

This effect can be seen more clearly in (d). Here the cumulative residuals are shown. These are defined as:

$$R_i = \sum_{j=1}^{i} Y_j - \hat{Y}_j$$

The same rules apply as for (c). The difference is that this plot is able to show consistent deviations more clearly. An ongoing consistent deviation shows a consistent slope, while the slope of a line corresponding to a temporary effect eventually trends back to 0.

Example of a price change not indicating causality
Figure 5.3 shows a causality analysis in a similar way as figure 5.2. Only this time a price change with a low causal effect is shown. Here we are looking at the second to last price change, which is a slight price increase. Here we expect the model to show a slight decrease in sales, however, the model predictions remain close to the actuals for both the 10 day period before the price change and roughly a 5 month period after the price change. After these 5 months, the residuals start to show a consistent deviation. This deviation can not be related to the price change as a 5 months delayed reaction is not realistic. This deviation does not correspond to another price change, its cause is unknown.

venue title	price change dates	MAE training	MAE after training 10 days	MAE after price change 10 days	MAE after price change 80 days
Venue 1	1	35.71	83.27	52.70	65.84
Venue 2	2	16.12	28.37	41.58	30.11
Venue 3	1	7.16	15.32	9.24	10.04
Venue 4	4	51.07	113.04	189.05	218.92
Venue 5	3	9.67	18.66	37.69	39.01
Venue 6	4	32.65	58.17	58.77	140.29
Venue 7	6	33.43	48.66	54.44	45.90
Venue 8	2	29.84	27.82	48.86	32.67
Venue 9	6	56.95	214.10	160.79	215.42
Totals	9	30.29	67.49	72.57	88.69

Table 5.2: Aggregated causal inference results per venue

Results
Table 5.2 shows a summary of the analysis in this chapter applied to 9 venues in the historical sales database. For a venue to qualify it needs to adhere to the rules from section 3.5.1. Along with the pre-requisite of table price discussed earlier, a price change is only included in this table if it occurs after at least 2 years of data, since the model has a higher chance of over-fitting on the training data with less data, influencing prediction quality. The table measures the Mean Absolute Error at certain points from the training cut-off date. Measuring the absolute error allows us to which quantify the extent to which a price change effect can be seen in the data, whether that's a positive or negative effect.

The 4 periods over which the MAE are measured are:

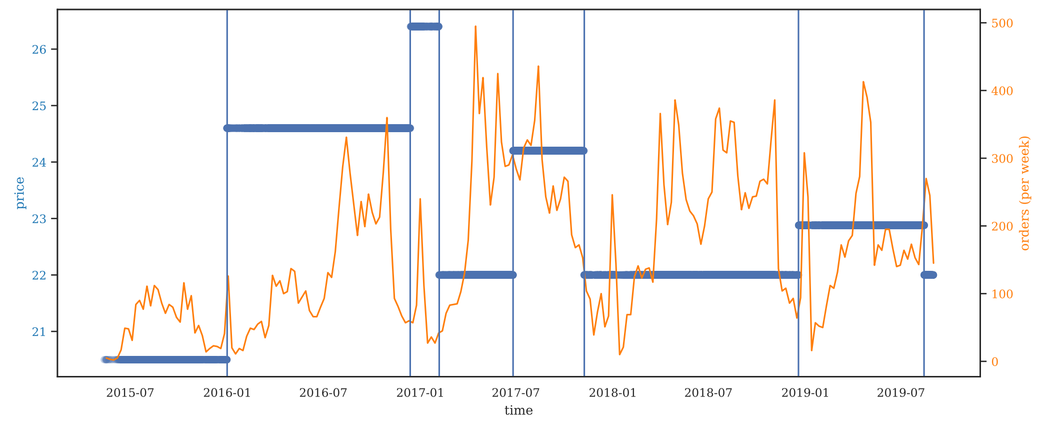
- The training period. Overall this period is expected to have the lowest MAE since the model uses this data to fit its parameters.
- 10 days after the training cut-off date. This period gives an indication of model fit since this period is before the price change the model should react as if a random date has been chosen. Two caveats are: 1) we can not guarantee that this period does not contain an unidentified event and 2) The period is only 10 days since we are taking the mean this period is more sensitive to outliers in the residuals than the training period (which contains at least 2 years of data).
- MAE 10 days after the price change. With the same period length as the previous metric, this period can be used to detect a causality by comparing it to the 10 days after training cut-off date MAE. A high relative value indicates a deviation of the model within 10 days of the price change.
- MAE 80 days after the price change. A long-lasting price change should show a similar MAE in 80 days. Keep in mind that the model is expected to perform better on earlier predictions and that the possibility of another event changing the actual sales over a long period of time is higher when looking at a larger period.

Since the number of eligible price changes is small, it is recommended to look at the plots (see figures 5.2 and 5.3) individually. These contain more information than this table, reducing them to averages. The appendix also contains a table showing the same information on a per price change date basis (see Table C.1 in the appendix).

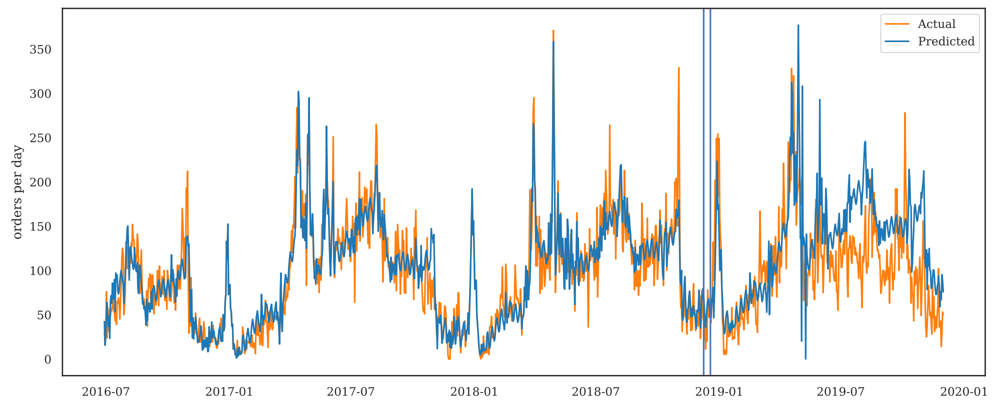
The totals[3] do show that the following holds true on average:

- The MAE for the training set is lower than that of the 10 days after training (and before price change). Indicating a higher performance on the training set as expected.
- The MAE 10 days after the price change is higher than the period before, albeit only slightly. This shows that some price changes cause a significant loss in accuracy for the model (see section 5.2.2), while some outliers even cause the MAE to be lower in this period. The same holds for the 80 days after price change period.

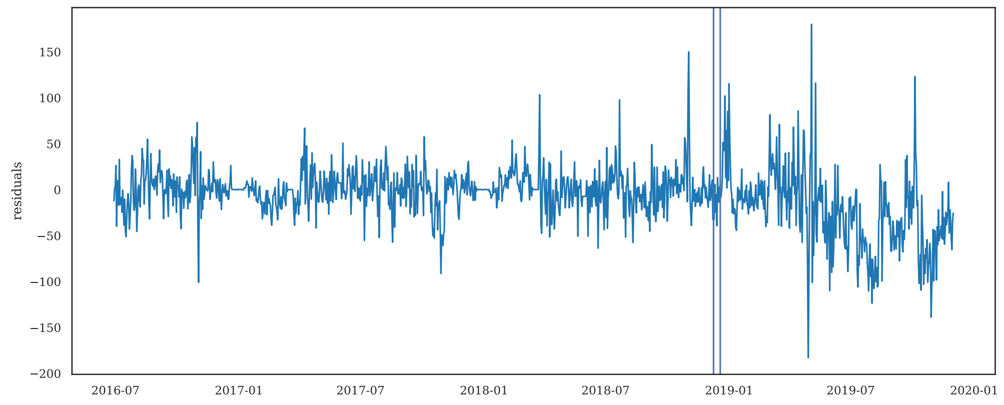
[3]The totals row contains values averaged per venue. Price change dates for this column, therefore, refers to the number of venues.



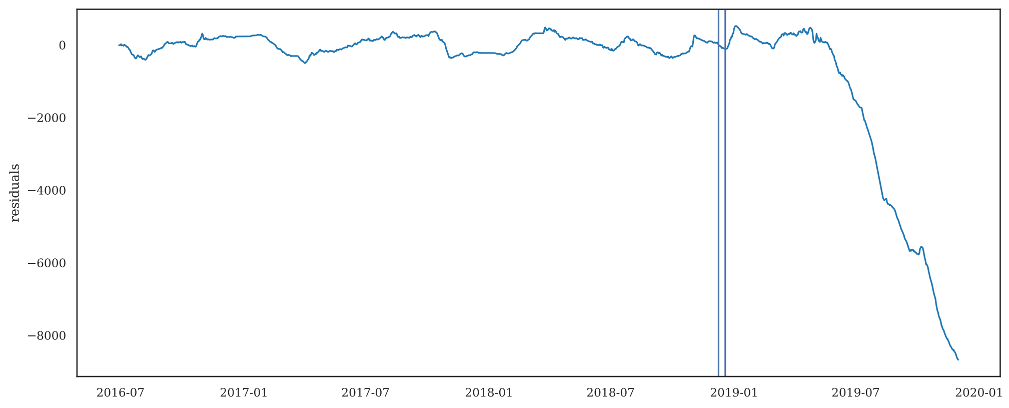
(a) Sales and pricing data for Venue 2. Vertical lines indicate estimated price changes.



(b) Sales data for Venue 2. Orange indicates historical sales while blue indicates the model output. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.



(c) Residuals for Venue 2. Residuals for a day are calculated by the actual sales for that day minus the model prediction. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.



(d) Cumulative residuals for Venue 2. The cumulative residuals for a day is defined as the sum of all previous residuals. The vertical lines indicate the training cutoff date and relevant estimated price change respectively.

Figure 5.3: Causality plots for Venue 2 on its second to last price change.

5.2.2. Conclusion

The main issue with extracting the causal effect of a price change from observational data is the presence of hidden variables. Variables that are not available to the researchers that influence the sales, price of both can skew any price analysis on the data. This chapter opens with a way to remove, if not minimize this effect by proposing a randomized price experiment in section 5.1. Randomly changing the price for certain venues over a pre-determined interval of time makes sure that the underlying reason for a price change is known (since it will be random). Doing this for a long enough period of time will make sure that the effect of any events unknown to us influencing the sales are averaged out as well.

One such experiment was set up with several cooperating venues for this thesis. This would have allowed us to quantify the error in not just the causality analysis from section 5.2 but also the baseline method (3.5.1), price module in the demand model (4.3) and even the results from the A/B testing part of chapter 6. Unfortunately due to the closing of all venues during the COVID-19 pandemic we are unable to get results from this experiment. It is still recommended for Tiqets to run this experiment as described in, or similar to section 5.1.

An attempt at finding the causal effect of price changes in historical data is described in 5.2.1. Using the model from chapter 4 without pricing information we can estimate the counterfactual and compare it to the actual historical sales. The counterfactual in this context the sales data that would have been logged if no price change had occurred. Section 5.2.1 discusses using an autoregressive model such as the one from chapter 4 instead of a state-space model as used in the paper by Brodersen et al. [5].

Results show a different effect from each eligible price change. Some indicating a clear causal effect while some showing none at all. This is to be expected since the reason for a price change might have been a reaction on an expected change in sales. In such a case the price change was a way for the venue to control the number of customers and the effect of the price change will be cancelled out in the sales data by the event that caused it.

It would be very interesting to see whether there is a high correlation between an indication of no causal effect (i.e. MAE after training but before price change similar to MAE after price change) and price changes as a reaction to a predicted event such as the opening or closing of a new attraction within the venue. Requiring a price change to be tagged from now on would yield significant advantages for future pricing analyses for Tiqets. This can be as simple as tagging a price change as either 1) reaction to a future event, 2) reaction to a perceived change in recent sales or 3) a pricing test with as goal determining the reaction of customers[4].

[4]Note that combinations are possible

6 Clustering

With over 3000 venues on the Tiqets platform, a subgroup of venues might exhibit similar characteristics. Identifying these groups allows for the creation of a control/test group split in experiments done within the company. Controlling the result for a similar venue removes the effects of any variable unrelated to the experiment, but present in both venues. Several departments within Tiqets such as finance, marketing and front-end use this method to quantify the effects of their decisions. Most of these groups are created manually when needed, often by domain experts. The creation of these groups is not necessarily data-driven, reducing reproducibility and increasing labour. This chapter explores the possibility of using historical sales data to group venues with similar characteristics together and the advantages this brings.

More concretely the objectives of this chapter are:

- To explore the possibilities of automatically clustering venues on their historical sales data.
- To define a clustering metric reflecting the usefulness of the clustering to Tiqets. This metric should be usable on existing (i.e. manually created) clusters as well.
- To compare different clustering methods and their parameters to each other and the base case (no clustering/random clustering), showing their effectiveness in A/B tests.

We do this by first defining our data, along with two ways of representing it. We then visualize the data in both feature spaces and determine different methods of clustering. We explore the optimal parameters for these methods and finally compare them in a way that makes sense to somebody that would use these results for their own analyses.

Appendix D shows auxiliary plots for this chapter.

6.1. Data and Features

The data that is available to us is the historical sales data of each venue. We will focus on clustering purely based on this data, as opposed to characteristics of the venue itself (country,target-audience) as there is not sufficient data of this type stored in a structured way for all venues. This section defines and explains the features that we can extract from the available data

We aggregate the number of tickets sold by the day. This removes the ability to draw conclusions on a smaller scale (hours or minutes) but will also allow us to avoid any complications that may arise from venues having different policies around reservation times (e.g. timeslot vs non-timeslot products, see the terminology reference) as well as reducing the run-time of algorithms used in this chapter and the number of seasonal periods to consider.

Even though the Tiqets platform contains more than 3000 venues, the data must be rich enough to infer patterns. Since we are only using historical orders we set a limit on this by selecting the top 1000 venues, sorted on the number of historical orders in the database. With this restriction in place, the venue with the lowest amount of historical orders has a total of 189 orders at the time of writing. Due to our method of extraction, some duplicates have been extracted as well, after removing these we retain 917 unique venues

6.1.1. Feature extraction

The features used in this chapter were taken from Hyndman [19]. In this lecture, the 3003 time-series used in the M3 forecasting competition are expressed in 6 features and used for visualization, anomaly detection and forecasting. Please see the related works section (chapter 2) for more context for this choice. In this chapter, these features will be used along with the features from section 6.1.2 to cluster the venues. The six features are described below.

Strength of trend
Three features are generated by first calculating the seasonal decomposition of the time-series data. Using the LOESS method [**?**]. The result is the time series represented as an additive model of three components as in 6.1.

$$Y_t = S_t + T_t + R_t \tag{6.1}$$

Where Y_t represents the original time series, S_t the seasonal component and R_t the remainder series. For strongly trended data, the seasonally adjusted data should have much more variation than the remainder component. Therefore $Var(R_t)/Var(T_t + R_t)$ should be relatively small. But for data with little or no trend, the two variances should be approximately the same. So we define the strength of trend as in 6.2[20].

$$max\left(0, 1 - \frac{Var(R_t)}{Var(T_t + R_t)}\right) \tag{6.2}$$

Because we clip the strength at 0, we get a measure of trend strength defined as a value between 0 and 1.

Strength of seasonality
The strength of seasonality is defined in a similar way in 6.3. Again as a value between 0 and 1. This time using the de-trended data as opposed to using the seasonality adjusted data.

$$max\left(0, 1 - \frac{Var(R_t)}{Var(T_t + R_t)}\right) \tag{6.3}$$

First autocorrelation
First autocorrelation in this context means the autocorrelation of the time-series with a lag of 1. In our case, this means the correlation of a day in the venue's sales with the previous day.

First autocorrelation of STL remainder series
Similar to the first autocorrelation this feature represents autocorrelation with a lag of 1, but instead of the venue's sales, we look at the remainder component of the seasonal decomposition on the sales R_t (as in equation 6.1). This means that we are looking at de-trended, seasonality adjusted data.

Spectral entropy
Spectral entropy is a measure of the forecastability of a time-series [23]. Where low values indicate a higher signal to noise ratio. This value is obtained by fitting a Fourier series on the data.

Optimal MLE Box-Cox transformation of data
The Box-Cox transformation is part of a series of power transformations that can be applied to data to reduce the effect of variance in the data changing over time [4]. The exact transformation required to achieve this changes per time series and therefore a different power transformation might be preferable for each time series. The Box-Cox transformation incorporates a λ parameter that changes the actual transformation applied to the data, allowing more flexibility. It is defined as:

$$w_t = \begin{cases} log(Y_t) & \text{if } \lambda = 0 \\ (Y_t^{\lambda} - 1)/\lambda & \text{otherwise.} \end{cases} \tag{6.4}$$

The λ parameter, therefore, defines the transformation required to normalize the variance of the data. The optimal value of this parameter can be found by maximizing the log-likelihood. This parameter can be used as a feature describing the time-series.

6.1.2. Daily sales as independent features

Instead of extracting features from the daily sales data, it is also possible to define each day in the dataset as a separate feature. This has several consequences.

Advantages

- This preserves more data as more data gets passed to the clustering algorithm.
- This directly corresponds to what we care about. Since we measure residuals in one way or another in section 6.6 this method of representing the time series is beneficial for methods that use Euclidean distance as their main way of clustering (such as our implementation of KMeans).

Disadvantages

- Removes the relation between days. The features from section 6.1.1 capture the relation between sales on different days (autocorrelation with lag, trend, seasonality, etc...). This information is not present in this representation.
- Massively increases the number of dimensions (in our case from the 6 defined in section 6.1.1 to 917). This might be a problem for some clustering algorithms that are heavily affected by the curse of dimensionality [29].
- Increases the input size of the data, slowing down clustering efforts.

Implementation

Taking the minimum and maximum date of all combined venues we transform the data into a $d \times n$ matrix. Where d is the total amount of days between the first date encountered in the data and the last, and n is the number of venues. When a venue has no sales data available for one or multiple days, the sales for those days are considered 0. We then apply normalization to the data before feeding it into the clustering algorithm.

6.2. Clustering

6.2.1. Clustering tendency

Since we now have the data represented in its features we can start to make sense of its structure. A good place to start is to measure the clustering tendency of the data. We can do this by quite literally comparing it to a random, uniform sample of points in the feature space. This measure is called the Hopkins statistic [17]. This measure will give us an indication of whether the data as represented by our features is able to be clustered.

Let us define our data as represented in the features from section 6.1.1 as $\mathbf{X}$ where $\mathbf{X} \in \mathbb{R}^{m \times n}$. In our case $\mathbf{X} \in \mathbb{R}^{6 \times 917}$. Consider a random sample of l points in $\mathbf{X}$ where $l \leq n$, $l \leq 917$, we call the members x_i. We then generate a set $\mathbf{Y}$ of uniformly distributed random samples where $\mathbf{Y} \in \mathbb{R}^{m \times l}$ with members y_i. Finally we define two measures u_i and w_i, where u_i is defined as defined as the distance between y_i and its nearest neighbour, and w_i the distance of x_i and its nearest neighbour. The Hopkins statistic H is then defined as in 6.5

$$H = \frac{\sum_{i=1}^{m} u_i}{\sum_{i=1}^{m} u_i + \sum_{i=1}^{m} w_i} \tag{6.5}$$

This should give us a measure of the clustering tendency in the data, expressed as a number ranging from 0.5 to 1. Where 0.5 shows no clustering tendency and a number tending to 1 a high amount of clustering tendency. Calculating the Hopkins statistic on our data the maximum amount of samples $l = 917$ we get $H = 0.895$ indicating that our data contains considerably more structure than a randomly generated, uniform sample in our feature space.

6.2.2. Visualisation

Another way to get a feeling of the structure of the data is through data visualisation. There are several different ways of visualizing the data, 3 of which are discussed in this section.

Figure D.1 (in the appendix) shows a scatter-plot matrix of the data as defined in the features from section 6.1.1. Each column and row represent a feature, an item in column 1 and row 2 shows a scatter-plot from which we can see the direct relation between feature 1 and 2. The diagonal shows histograms of the associated

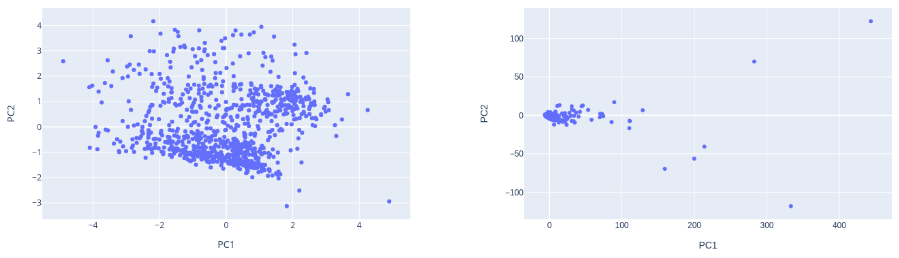


Figure 6.1: Dimensionality reduction using PCA from features described in section 6.1.1 (left) and the features described in section 6.1.2 (right)

features. These plots allow us to see correlations between 2 individual features. In our case, this is hard to see, given the complexity of the data this plot does not show clearly separable venues among two features.

PCA and T-SNE are dimensionality reduction techniques that allow us to visualise the entire dataset across all features in a 2-dimensional plot. This makes it easier to see the relation between venues and helps with determining the right parameters for the clustering methods. Dimensionality reduction methods are sometimes also suggested as a preprocessing step for clustering methods that do not handle higher-dimensional data well. Since in our case the models perform better in the higher dimensional variant this is not implemented, but it could be a topic for future work.

PCA

A more compact way of visualizing the data is through dimensionality reduction techniques. One of the more common ways of doing this is by using Principal Component Analysis to redefine the dimensions of the feature space in such a way that the individual dimensions of the result, called the principal components are uncorrelated. By sorting the principal components according to the amount of variance they explain in the original dataset, we can remove the 'least important' components, making the feature space more manageable.

Doing so to reduce the amount of components to 2 allows us to visualize the data in the 2-dimensional plot shown in 6.1. The right plot (using the features from section 6.1.2) shows an example of what happens in pca if high dimensional, correlated data is reduced to 2 dimensions.

T-SNE

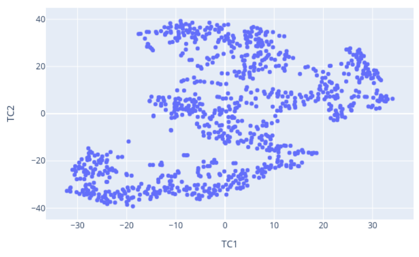
T-SNE is a non-linear dimensionality reduction technique that tries to preserve clustering in the lower dimensions. Using a similarity measure based on its closest neighbour, the similarity between each point and every other point in the data is calculated. Averaging these similarities across both directions (since the parameters measure is are different for each point) gives us a similarity matrix $S \in \mathbb{R}^{n \times n}$ with $n = 917$ for our data. In a lower-dimensional space T-SNE then calculates a similarity matrix $\hat{S} \in \mathbb{R}^{n \times n}$. Iteratively the algorithm moves points in the lower dimensional space around in such a way that the difference between S and $\hat{S}$ is as low as possible.

This method is able to reduce the dimensionality of the data in a non-linear way, and as such, it is usually able to preserve more of the clustering present in the original data compared to PCA. This comes at the cost of reduced explain-ability since the Principal Components resulting from PCA can be expressed as a combination of the original features.

6.3. Clustering methods

6.3.1. KMeans

KMeans is a clustering algorithm with a n parameter, indicating the number of clusters. As a distance metric, we use Euclidian distance. The algorithm works by assigning points in the input space as cluster means. In our case, this happens randomly (with a seed for reproducibility). Each venue is then assigned to the closest cluster mean and the means are recalculated. The algorithm finishes when the cluster means do not change anymore after recalculation.



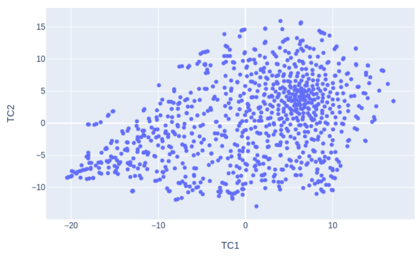


Figure 6.2: Dimensionality reduction using T-SNE from features described in section 6.1.1 (left) and the features described in section 6.1.2 (right).

6.3.2. DBSCAN

DBSCAN is a clustering algorithm with a eps parameter, indicating the number of clusters indirectly. The abbreviation stands for Density-Based Spatial Clustering of Applications with Noise [7, 27]. The technique works by distinguishing high-density areas from lower-density areas using eps as a sensitivity parameter. An interesting property of this method is that it labels samples found in low-density areas as -1. In our analysis we consider these samples belonging to no cluster. This must be taken into account when comparing DBSCAN with different methods that do cluster all venues such as KMeans.

A lower eps parameter increases the number of clusters found by the method. It also increases the number of venues belonging to no cluster. An example of this can be found seen in section 6.6.2.

6.3.3. OPTICS

OPTICS is a clustering algorithm that requires no parameters. It is similar in its workings to DBSCAN but loses the eps parameter. Its abbreviation stands "Ordering Points to Identify Cluster Structure". Internally the algorithm orders points based on the distance in the input space. Much like DBSCAN it then defines clusters based on high and low-density areas [1].

OPTICS also defines venues belonging to no cluster. Keep this in mind when interpreting the results from section 6.6.

6.4. Evaluation

Evaluating the quality of clustering on the data is in inherently a hard problem since it depends heavily on the reason for clustering, as well as the specifics of the data. Since the purpose of these clusters is to provide the analyst with control/treatment groups, we define a measure of quality reflecting this use case in this section.

6.4.1. Silhouette coefficient

The Silhouette coefficient of a sample in a clustered dataset is a measure that measures both the mean intra-cluster distance as well as the mean nearest cluster distance of that sample. The silhouette coefficient is defined as in 6.6

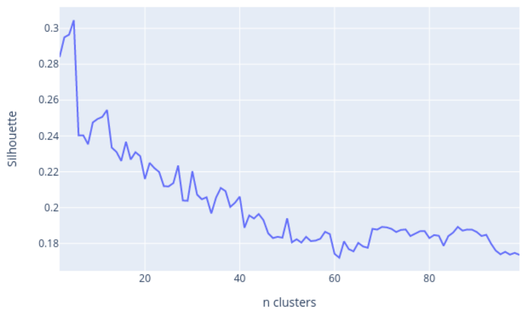
$$s(i) = \begin{cases} \frac{b(i)-a(i)}{max(a(i),b(i))}, & \text{if } |C_i| > 1 \\ 0, & \text{if } |C_i| = 1 \end{cases} \tag{6.6}$$

For data points i in a cluster C_i with arbitrary distance function $d(i,j)$ Euclidean distance in our case. $a(i)$ is the mean distance between the point in question and all other points of the same cluster. It can be interpreted as a measure of how well point i is assigned to its cluster, with smaller values indicating a better match. The definition for $a(i)$ is given in equation 6.7.

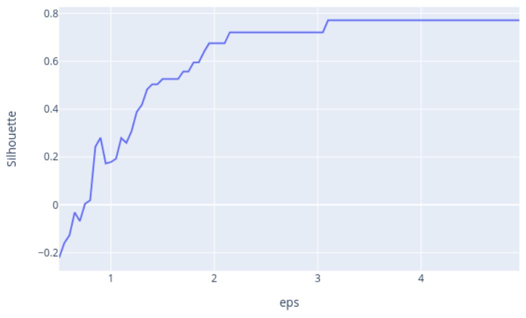
$$a(i) = \frac{1}{|C_i|-1} \sum_{j\in C_i, i\neq j} d(i,j) \tag{6.7}$$

$b(i)$ can be interpreted as the mean of distances from point i to the points in the neighbouring cluster. The neighbouring cluster is found by minimizing the terms after the min function in equation 6.8.

$$b(i) = \min_{i\neq k} \frac{1}{|C_k|} \sum_{j\in C_k} d(i,j) \tag{6.8}$$



(a) Average silhouette score of the data clustered by KMeans per n parameter. Interesting values are n=5 (the first peak) and n=61 (the lowest value)



(b) Average silhouette score of the data clustered by DBSCAN per eps parameter. Interesting values are the first two peaks: eps=0.65 and eps=0.9

Figure 6.3: Average silhouette score KMeans and DBSCAN

We can calculate the average silhouette coefficient of the clustered dataset for different clustering methods and parameters to find the best clusters for our data. Doing so allows us to find the most appropriate parameters for our clustering methods. Plotting the average silhouette score for KMeans on its number of clusters parameter and DBSCAN on its eps parameter gives us the plots in figure 6.3.

One thing to note is that while the silhouette score of DBSCAN keeps increasing with an increasing eps parameter, DBSCAN actually clusters fewer venues when eps increases. It is, therefore, preferable to have a lower eps value (more of this in 6.3.2). Another thing to note is that while the silhouette score is a common metric when clustering, it does not necessarily reflect the quality of the clusters for our use case.

6.5. Simulating A/B tests

Since one of the goals of the chapter was finding an automated way of getting control and treatment groups for an A/B test, our metric for cluster quality should reflect the expected error on the outcome of these tests. This section helps to illustrate how we defined those metrics by simulating the effect of an A/B test with a control and treatment group and attempting to extract the effect without using the counterfactuals.

We defined a test period of 3 months from `2019-09-01` to `2019-12-01`. `2019-12-01` was chosen as a safe date before the influence of Covid-19 could be seen. Choosing the start of December also allows us to not worry about the effect of the holidays on sales. 3 months are chosen because this corresponds to the test period used for hyper-parameter optimization from the demand model (Chapter 4). This might allow us to compare results. All data after this point is discarded. We then perform our clustering: `KMeans` with `n=5` on the features from section 6.1.1 on only the data up to `2019-09-01`, as data from the test set would include counterfactuals not available to us in a real-life scenario.

We then define 4 different effects, each 90 days[1] in length as in equation 6.9 for $\{i \in \mathbb{R} : 1 \leq i \leq 90\}$. With f_1 and f_2 representing linearly increasing and decreasing effects, while f_3 and f_4 represent temporary effects. The effects can also be seen visually in figure 6.4. Since the effect functions have a range of $[0, 1]$ a scaling parameter m is used to control the effect size.

$$
\begin{aligned}
f_1(i) &= m \cdot \frac{i}{90} \\
f_2(i) &= m \cdot 1 - \frac{i}{90} \\
f_3(i) &= m \cdot 1, \text{ if } 15 < i < 75, \text{ else } f_3(i) = 0 \\
f_4(i) &= m \cdot e^{\frac{i-0.5}{2\cdot(0.15)^2}} = m \cdot e^{\frac{i-0.5}{0.045}}
\end{aligned}
\tag{6.9}
$$

With the clusters from our `KMeans` clustering, we define a treatment group by randomly sampling 3

[1]The third effect contains 90 days. Parts of this effect are defined as having no effect on the sales. It would technically more accurate to say that this effect spans 60 days, the days with zero effect are left in to see the contrast in the effect estimation.

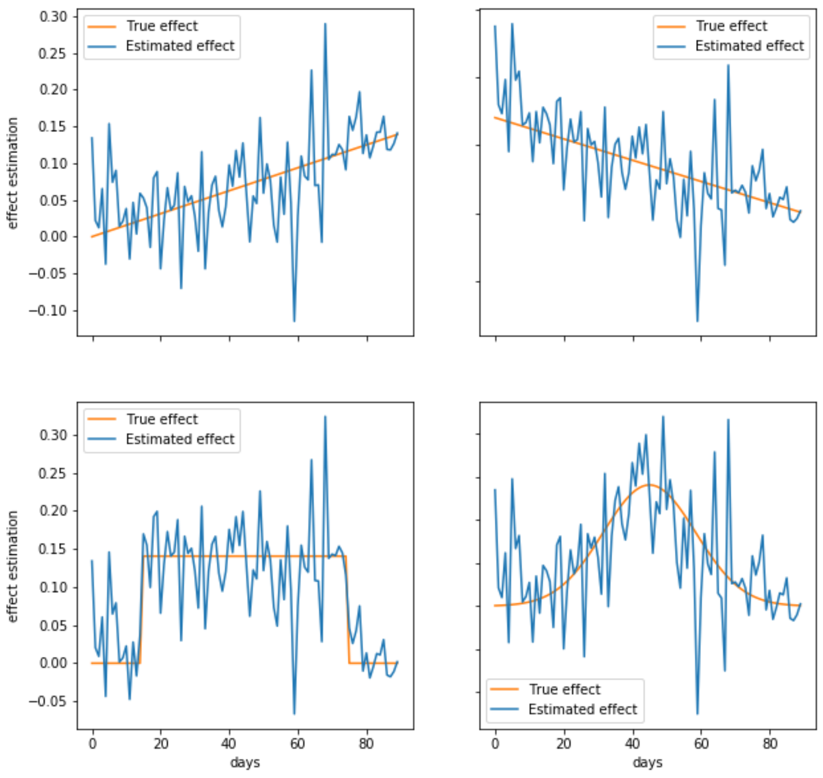


Figure 6.4: Effect estimation for all effects additively applied to the data.

venues from the cluster. We sample 1 venue if the cluster size $s \leq 3$, 2 if $4 \leq s \leq 5$ and discard the cluster when $s = 1$. In the data from this visualisation $s > 5$ for all clusters, so the treatment group size is 3. The control group consists of the rest of the venues. Note that this approach is also used in the scores from section 6.6.

We apply min-max scaling on the data with the min and max values from the training period. We now define the difference between an additive effect and a multiplicative one. Figure 6.4 shows our attempt to extract from an additive effect without using the counterfactual (in this case the test data before adding the effect), while figure D.3 in the appendix shows extraction from a multiplicative effect. For the sales data $Y_1, Y_2, ..., Y_n$ in the test period (with $n = 90$ in our case), the adjusted vector $\hat{Y}$ is defined as in equation 6.10 for the additive effect and as in 6.11 for the multiplicative effect.

$$\hat{Y}_i = Y_i + \frac{f(i)}{n} \sum_{j=1}^{n} Y_j \tag{6.10}$$

$$\hat{Y}_i = Y_i + Y_i \cdot f(i) \tag{6.11}$$

Figure 6.4 shows the effect estimation on one randomly chosen cluster with a treatment group size of 3 and a control group size of 105. Table 6.1 shows the actual and estimated total effect values over the test period. The actual effect $f(i)$ can be written as:

$$f(i) = \hat{Y}_i - Y_i$$

With $\hat{Y}$ containing the mean sales of venues in the treatment group and Y the mean of the original values. The estimation of the actual effect can be written as:

$$\hat{f}(i) = \hat{Y}_i - C_i$$

With the mean of the control group $C_1, C_2, ..., C_n$ replacing the counterfactuals $Y_1, Y_2, ..., Y_n$.

Common A/B testing practice is to compare effect over the entire period. This results in:

$$\hat{Y}_{total} = \sum_{i=1}^{n} \hat{Y}_i \quad Y_{total} = \sum_{i=1}^{n} Y_i \quad C_{total} = \sum_{i=1}^{n} C_i$$

The estimated and actual total effect over the test period can be represented as:

$$\text{Estimated total effect: } \hat{Y}_{total} - C_{total}$$

$$\text{Actual total effect: } Y_{total} - \hat{Y}_{total}$$

Table 6.1 shows these as a percentage of Y_{total}. The difference the estimation and actual effect can be written as:

$$(\hat{Y}_{total} - C_{total}) - (Y_{total} - \hat{Y}_{total}) = \hat{Y}_{total} - C_{total}$$

This value does not contain the treatment data ($\hat{Y}$). Looking at table 6.1 we see that the difference between the actual and estimation for each effect is exactly 2.13%. In fact, this value is the same no matter the **shape** and **size** of the effect, and since we are looking at totals this is also the case for the multiplicative scenario. We can, therefore, use this value as a metric defining the cluster quality in a non-parametric way. We call this value the residual percentage and define it as in equation 6.12.

$$\text{Residual percentage} = \left| \frac{\hat{Y}_{total} - C_{total}}{\hat{Y}_{total}} \right| \tag{6.12}$$

Another use case might be defining the shape of the effect. One might want to estimate the effect over time of a marketing campaign on a venue. The success of this estimation depends heavily on the prior knowledge of the distribution belonging to the effect. That is to say that the estimation shown in Figure 6.4 can be greatly improved with prior knowledge of the shape of the true effect. Without assuming that the analyst has such knowledge, a metric that indicates the quality of such an effect estimation can be any non-parametric accuracy measure. Because of familiarity and compatibility with earlier analyses we choose SMAPE (as defined in section 4.1.2).[2]

The metrics for this cluster are a residual percentage of 0.0213 and a SMAPE score of 29.16. This makes it a fairly high scoring cluster (see section 6.6). The appendix contains an average performing cluster as well as the results of the multiplicative scenario for both clusters.

6.6. Scoring

We now have two relevant ways of measuring the quality a cluster: residual percentage for a totals comparison over a period of time, and SMAPE for shape estimation. Using a combination of methods from section 6.3 we can explore the ideal clustering method for each use case.

[2]This means that both this effect estimation and forecast quality of the demand model are defined using SMAPE. Keep in mind that forecasts with the demand model are possible for future data, while this estimation requires the sales data from venues in the control group for the estimation period.

Effect	Total mean sales treatment (Y_{total})	Total mean sales control (C_{total})	Total mean sales treatment w/o effect ($\hat{Y}_{total}$)	Estimated total effect ($\frac{\hat{Y}_{total}-C_{total}}{Y_{total}}$)	Actual total effect ($\frac{\hat{Y}_{total}-Y_{total}}{Y_{total}}$)
f_1	18.31	11.80	12.06	+53.98%	+51.85%
f_2	18.45	11.80	12.06	+55.15%	+53.02%
f_3	20.48	11.80	12.06	+72.04%	+69.91%
f_4	16.81	11.80	12.06	+41.53%	+39.39%

Table 6.1: Total effect estimation values corresponding to figure 6.4

6.6.1. Methodology

The evaluation method from section 6.5 is repeated for each cluster for several clustering method and parameters. In short, for each clustering method/parameter combination we cluster all venues using the clustering method/parameter combination on the training period only. Then for each cluster we:

- Retrieve the test period data for all venues in the cluster.
- Repeat the following r times to get a spread of metrics for larger clusters, who would otherwise return just one (possibly unrepresentative) value:
- Pick a treatment and control group by randomly picking n venues for the treatment group, all other venues belong to the control group. n is defined as in equation 6.13 for a cluster size of s with $s > 1$.

$$\begin{aligned} n &= 1, \text{if } 2 \leq s \leq 3 \\ n &= 2, \text{if } 4 \leq s \leq 5 \\ n &= 3, \text{ otherwise} \end{aligned} \tag{6.13}$$

- Scale the sales data of each venue individually using minmax scaling.
- Compare the mean of the sales data for the treatment and control groups to retrieve the residual percentage and SMAPE scores for this particular treatment/control split.

6.6.2. Results

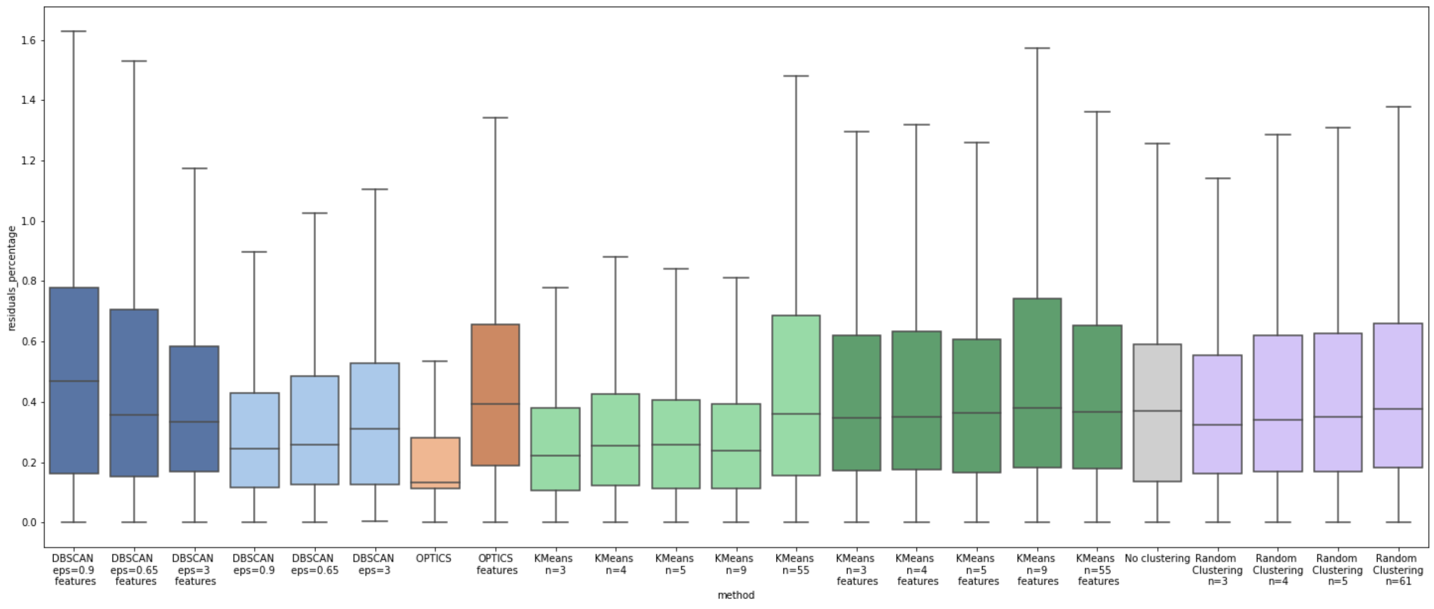


Figure 6.5: Boxplot of different clustering methods/parameters on the residuals percentage metric

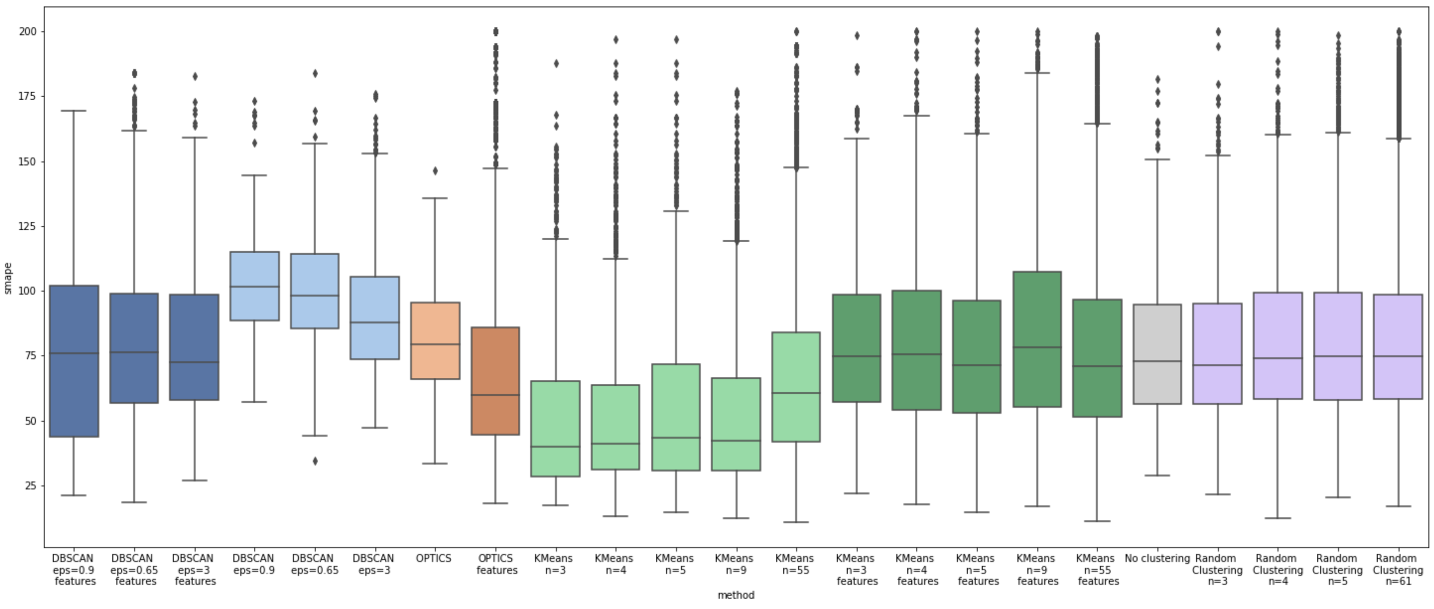


Figure 6.6: Boxplot of different clustering methods/parameters on the residuals percentage metric

Residual percentage
Figure 6.5 shows a boxplot of different clustering methods/parameters on the residuals percentage metric. Note that in this plot the outliers are hidden. This is because there a several large outliers present that would

make the plot unreadable. For transparency, a result with outliers is included in the appendix (Figure D.6). Results are discussed below:

Features
The features subscript below some of the methods indicates that the features from 6.1.1 are used as input. All other methods use the feature per day approach as described in 6.1.2. This is with the exception of the 'No Clustering' and 'Random Clustering' entries, which do not require input.

DBSCAN
DBSCAN on the extracted features perform worse than KMeans on the extracted features, and in some cases even worse than no clustering/random clustering. DBSCAN also clusters some venues in class "-1", representing its inability to cluster those venues (see section 6.3.2). Of the 891 venues the following amounts are labelled "-1":

```
DBSCAN eps=0.9  features  : 133
DBSCAN eps=0.65 features  : 359
DBSCAN eps=3    features  : 3
DBSCAN eps=0.9            : 648
DBSCAN eps=0.65           : 774
DBSCAN eps=3              : 620
```

Since these venues do not belong to any cluster, any analysis that requires these venues to be included can not benefit from the DBSCAN clustering method. Ignoring this filtering (considering all "-1" as one cluster) significantly increases the expected errors in Figures 6.5 and 6.6.

OPTICS
OPTICS with daily data as the features is actually the best performing method for the residuals percentage metric. Unfortunately, it achieves this with the following counts for each label: `'0': 620, '-1': 271`. This means that it just discards 271 venues from the analysis. This is still a useful use case however since the 620 perform better than any other cluster. If you want smaller clusters the KMeans methods on daily data are the next best-performing methods.

Kmeans
KMeans with daily data as the features performs best for both the residuals percentage metric and the SMAPE metric given that all venues need to be included. As indicated by the Silhouette score[3] n=3 gives the best fit, followed closely by the others, this gives the user some freedom in choosing the number of clusters without sacrificing much quality.

On the features from section 6.1.1 KMeans performs only slightly better than the no clustering, random clustering methods. In some cases, it even performs worse.

No clustering & Random clustering
No clustering is achieved by assigning all venues the label "0". Random clustering assigns each venue a label (integer) randomly sampled from a uniform distribution with range $[0, n)$.

Summary
We have scored several configurations of different clustering method and ways of representing the data against two main metrics. Figure 6.5 shows the residuals percentage metric, which indicates the expected error margin of a regular A/B test using the clusters in the way defined in methodology (i.e. 3 venues treatment, the rest control). Figure 6.6 gives the SMAPE score, which indicates the expected error on a more complex effect analysis.

Optics seems to be a clear winner for the residuals percentage in Figure 6.5, but it is important to remember that OPTICS excludes a lot of venues. In the absence of a venue preference when executing an A/B test, or in the case that the preferences of the analyst happen to align with the venues included in the OPTICS clusters, OPTICS would give tighter margins on the expected error.

A runner up for the residuals percentage and the clear winner for the SMAPE score in Figure 6.6 is KMeans $n = 3$. This method does not exclude venues and is, therefore, more usable in general. We found the n

[3]Figure 6.3 shows the same process but on the data of features from section 6.1.1 instead.

parameter by looking at the optimal silhouette score in a similar way to Figure 6.3 (these plots use the features from section 6.1.1 as opposed to the ones from section 6.1.2).

The methods are also compared against not clustering the data and clustering by assigning random numbers as cluster identifiers to the venues. While playing with the parameters we find that KMeans will approach these results when moving away from the optimal choice of parameters.

6.7. Conclusion

In this chapter, we have explored clustering using purely sales data. We have evaluated the clustering methods according to its utility in determining control/treatment groups for A/B testing, where a treatment applied to the venue would show direct results in the sales of that venue.

Since we are using purely sales data it is interesting to see if any semantic features (country, venue-type, age average demographic, etc..) is reflected in the clustering. However, we have not been able to find any. The appendix contains a plot (figure D.7) colouring venues by country in T-SNE reduced data.

There might also be different use cases for clustering within the company. One of these is the use of clusters to reduce the hyper-parameter grids searched when training the demand model. A metric can be defined which corresponds to the sizes of these grids used in the model without losing predictive power (e.g. max grid size). This is an interesting topic to explore in future analyses.

The goal of this chapter; automatically creating clusters for use in A/B testing has been achieved. We have explored and compared several clustering techniques (KMeans, DBSCAN, OPTICS, No Clustering and Random clustering) across several parameters and along with 2 ways of representing the data (Extracted vs daily features).

We have used several methods of exploring the data to get a better sense of both its structure and the best parameters for our clustering methods.

Finally, we have found a way of scoring the clusters by simulating A/B tests on the data, which is shown visually in section 6.5. It is important to note that we have defined the maximum size of a treatment group to be 3. If in practice it is possible to apply the treatment to more venues this usually greatly reduces the error rates. It also lowers the importance of clustering as the variance between the mean scores of each algorithm decreases when increasing this limit.

6.7.1. Future work

This chapter raises several interesting possibilities for future work:

- sections 6.1.1 and 6.1.2 discuss the advantages and disadvantages of each feature set. An interesting thing to compare would be a feature set that manages to include both the relation between sales in the time direction in some way (e.g. trend, seasonality, autocorrelation) and the specificity of the daily features. It might be useful to note that even with the daily features the current clustering methods are able to run in a negligible amount of time.

- Minmax scaling was used when extracting metrics from normalized venue sales data. It makes sense to apply some normalization when comparing venues since the absolute difference in sales often matters less than the relative difference over time. But the current normalization is very sensitive to outliers in the data. Perhaps by scaling to the upper (.75) quantile of the data instead of the maximum we could get better results.

- OPTICS and DBSCAN sometimes exclude a lot of venues. As already discussed this can be beneficial if you as an analyst do not happen to care about the specific venues included in the A/B test. It might be possible to predict the residual percentage metric on the test period by looking at that metric on the training period. This could allow you to achieve a similar venue excluding effect with KMeans, increasing the accuracy on the venues that are left.

7 Conclusion

Our industry partner is Tiqets, a company managing a ticketing platform that handles the sales of tickets for over cultural venues 3000 worldwide. The company has expressed interest in gaining more insight into their customer's reaction to prices of their venues. Specifically, they are looking for ways to automatically infer price elasticity from available data on a per venue basis, allowing the company to make pricing decisions automatically. Due to Tiqets's large database of heterogeneous products, most of which are inflexible to experiments we are interested in extracting price information in the least intrusive way possible (i.e. using observational data).

The company is somewhat limited by the fact that it is not allowed to differ the price on a per user basis. Randomly varying the price in a structured way would allow the company to perform an unbiased price test in a shorter time than the price test discussed in section 5.1. It would even enable the use of multi-armed bandit models to update the price and learn from customer reactions while optimizing the price [32]. While this should be technically possible with the current infrastructure of the platform, it was decided to be out of scope for legal and public-relations reasons.

The studies undertaken in chapters 3 through 6 allow us to now answer our research questions:

1. To what extent are we able to extract the relation between price and demand on a per venue basis from observational data?
   (a) What methods are available and what are their advantages and drawbacks?
   (b) What kind of data is necessary to quantify the quality of the observational data?
2. What model most accurately predicts demand on a per venue basis?
   (a) What is the effect of including or omitting price as a regressor?
3. In what ways can we extract and compare similarities the sales data of venues?
   (a) Can a method of clustering on this similarity be used to the benefit of the company?

7.1. Summary

Chapter 3 discusses the state of the historical data collected over all venues. The historical price for a venue is not stored, but can be estimated using the payment data from historical orders. One straightforward way of improving price analyses on future data would be for Tiqets to store price changes in their database. This would improve the accuracy of all analyses by a factor inversely proportional to the popularity of the venue. It would be even more beneficial to require a price change to be tagged with the reason for that price change, section 5.2.2 proposes a method to do this.

Section 3.5 introduces the naive way of determining price elasticity. This method is insufficient as the result can be (and often is) influenced by many un-modelled factors. Chapter 4 focusses on removing some of these factors by modelling using a modular regression model. The model is trained on the historical sales data of a venue and should not require human intervention. By removing the trend, seasonality and predictable outliers (such as holidays) we are able to get an improved estimate over the one obtained in section 3.5.

Unfortunately given the autoregressive nature of the model, it tends to fit the effects of a price change in one of its modules. This is why section 4.3 focusses introducing the historical price of the product as an input of the model. This minimizes the incorrectly modelled effect of the price change by explicitly modelling the price as a regressor.

The effectiveness of such a method relies on the assumption of a causal relationship between the price change and the sales after the price change date. Chapter 5 explores ways of estimating these causal relations. The gold standard given the legal restrictions applied to the platform[1] is a randomized price experiment as in section 5.1. Such an experiment would have been able to accurately estimate the causal effect of each price change. Given this information, all other methods for inferring the price elasticity can be quantified. This has value for Tiqets since it would, with a quantifiable error margin allow, them to substitute a price experiment with an analysis on historical data for venues that are not able to comply with the changing prices required for a price experiment.

Chapter 5 also introduces a way of estimating the causal effect on a per price change level, using the demand model from chapter 4. Section 5.2.1 shows the causal effect of price changes not only as a value per price change but also in plots that can illustrate the changes in customers reactions over time. Note that these causality-estimates do not take competitor information into account, while the randomized price experiment from section 5.1 does. This is because the competitor prices are also not stored in the historical database. Storing these would greatly improve future analyses relating to price.

Chapter 6 explores automated ways of clustering venues according to historical data. These clusters can be used by Tiqets when creating control/treatment groups for A/B tests (pricing or otherwise) that rely on sales as a measure of quality. Several methods are explored and compared against results obtained by either not clustering or assigning venues a random cluster.

7.1.1. Extracting the relation between price and demand from observational data

Extracting the relation between price and demand from observational data has been the primary focus of this thesis. Chapter 4 discusses an automated way of modelling this relation to allow for demand manipulation using price. Because of the nature of observational data and the absence of competitor information results have been limited. The price module was unable to improve the model for most venues and the ones that are improved have such varied results between configurations that it leads us to believe that the true relation between price and demand is not sufficiently captured.

The pricing experiment from 5 would give us the information needed to quantify exactly the extent to which the demand model is able to model price. This experiment included competitor information, but more importantly, consists only of price changes with exogenous variation. Combining this with the analysis from section 5.2 would show us which price changes in the observational data contain useful information. Such insights may allow us to develop better ways of retrieving the relation between price and demand from observational data for venues that for any reason can not initiate a pricing experiment.

Chapter 6 attempts to sidestep this issue by using the data from other venues. This removes the influence of any hidden variables that are present in both the venue of interest and similar venues in the database. It does so with quantifiable error. Once the venues under consideration are known a stable error estimate can be given. Section 6.6 gives an idea of the distribution of these estimates over all venues.

7.1.2. Demand model

Chapter 4 discusses the demand model in detail. This model was chosen not just because it is the best performing model, but also because it allows us to decompose a single time-series into multiple variables usable for analysis.

The effectiveness of adding the price regressor is discussed in section 4.4 and summarized in the first paragraph of the previous subsection (section 7.1.1).

7.1.3. Similarities

Chapter 6 discusses similarities between venues in detail. We define two different methods of representing venues by their sales data and compare clustering methods on their ability to create treatment and control groups for controlled experiment analysis (e.g. price change analysis).

[1]Randomizing the price on a per-user level in a similar way would remove effects caused by hidden variables changing during the test period. These effects are currently minimized by averaging over a large enough experiment period.

We found an improvement over randomly clustering venues and outlined a method to retrieve an error estimate for the experiment once the involved venues are known.

7.2. Limitations and Future work

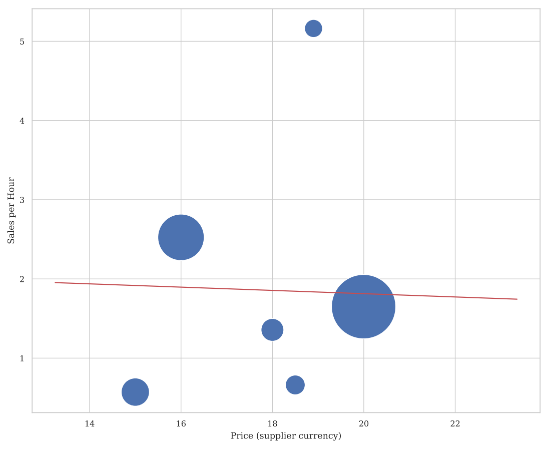
Results from this thesis have been limited mostly by a lack of data. Primarily by not having the full information around a price change. Requiring a price change to be tagged from now on would yield significant advantages for future pricing analyses for Tiqets. This can be as simple as tagging a price change as either 1) reaction to a future event, 2) reaction to a perceived change in recent sales or 3) a pricing test with as goal determining the reaction of customers. Note that combinations are possible.

Another highly useful dataset is competitor information. All of the analyses in this thesis were done without competitor price and stock information which could have had a huge influence on historical sales. Section 5.1 describes an experiment that would include competitor information, along with ways to interpret the data both with and without this information. This would allow us to quantify the value of adding competitor price and stock[2] information to the price demand relation analysis. Unfortunately, the plan to execute this experiment was cut short due to the global COVID-19 pandemic halting tourism around the world. It is our recommendation for Tiqets to restart this experiment when sales are back to normal.

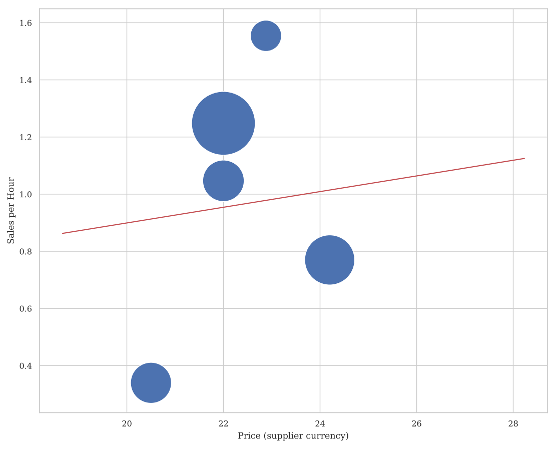
Another avenue for future work would be replacing the model from chapter 4 with a Bayesian state-space model as in Brodersen et al. [5]. Such a model retains the decomposability of the current model as it requires several independent streams of information over time that combined can predict demand. Finding these time-series would be the main challenge. It would be interesting to see how this affects the price module (section 4.3) and causality (section 5.2) analyses that depend on the demand model.

[2]Competitor stock information for inventory based products would likely never be made available to us, but the possibility for estimating this information using stock-out signals does exist [10].

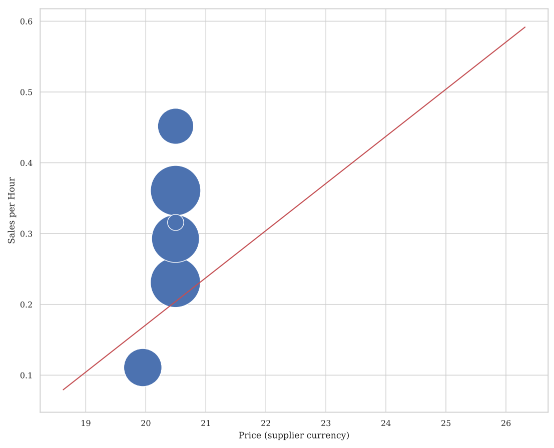
A
All baseline plots



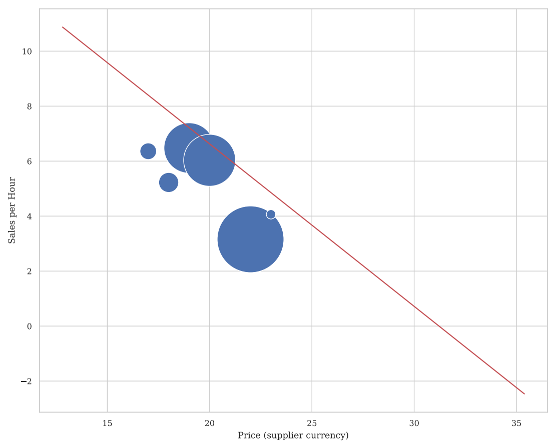
(a) Baseline plot for Venue 1



(b) Baseline plot for Venue 2

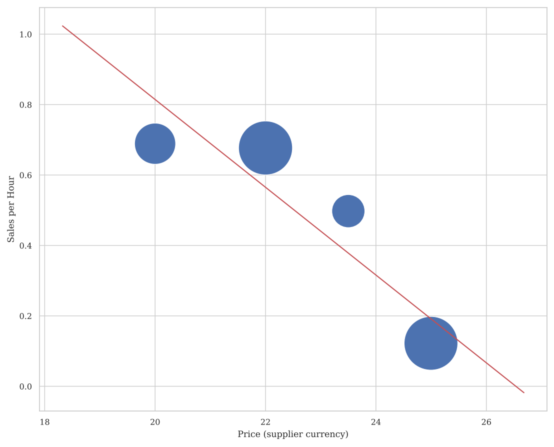


(c) Baseline plot for Venue 3

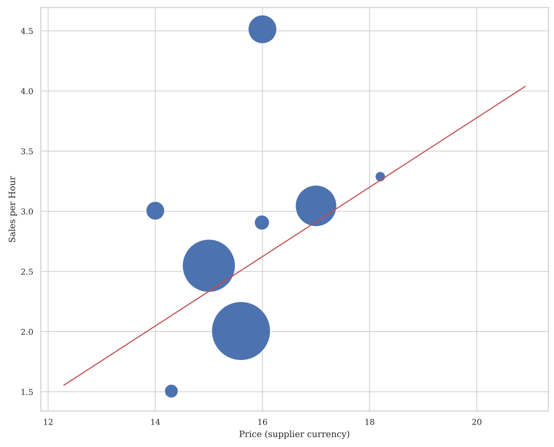


(d) Baseline plot for Venue 4

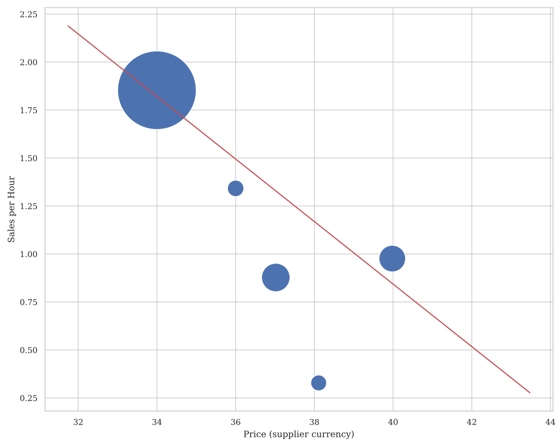
Figure A.1: Baseline plots for Venues 1-4



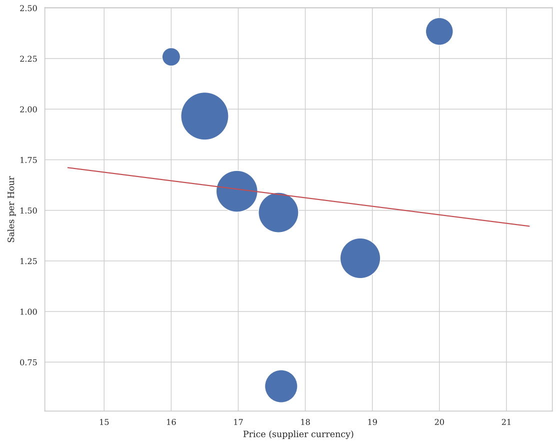
(a) Baseline plot for Venue 5



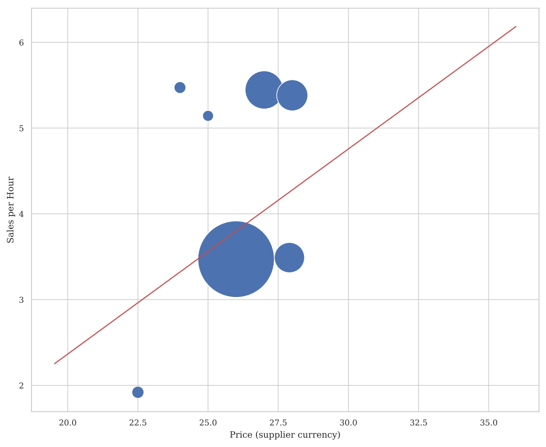
(b) Baseline plot for Venue 6



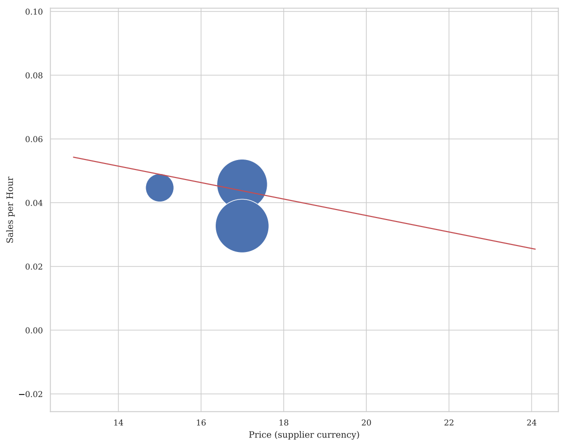
(c) Baseline plot for Venue 7



(d) Baseline plot for Venue 8



(e) Baseline plot for Venue 9



(f) Baseline plot for Venue 10

Figure A.2: Baseline plots for Venues 5-10

B

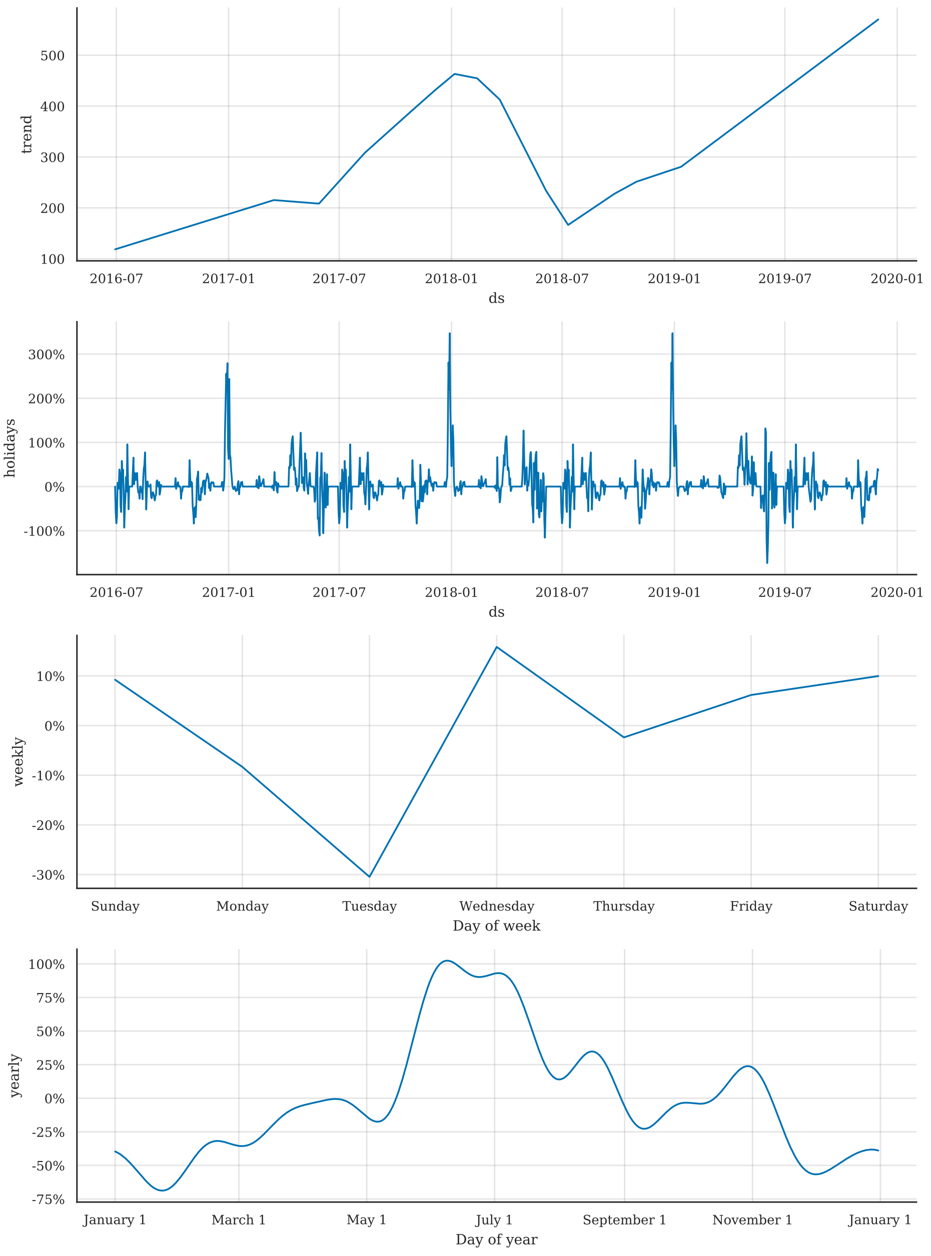
Demand model

B.1. hyperparameter grid

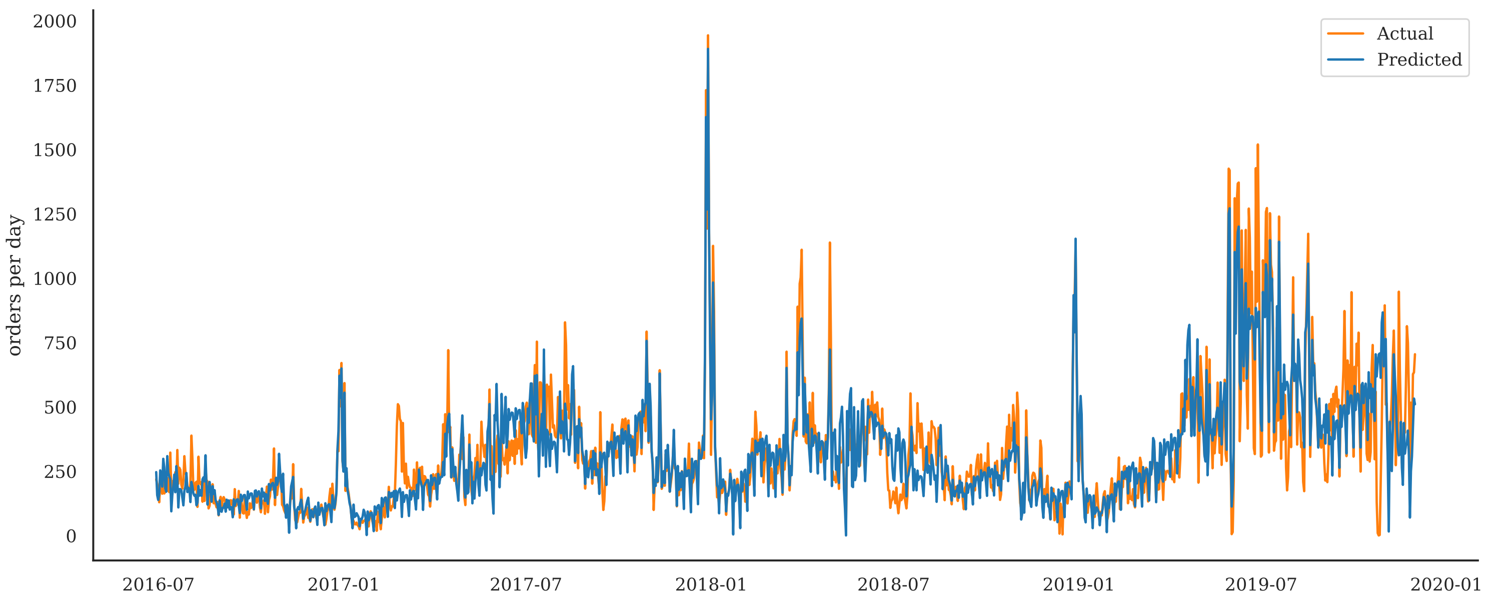
```
# trend
growth=["linear"],
# seasonality
seasonality_mode=["additive", "multiplicative"],
# holidays
holidays=["country", "weighted_country"],
holiday_lower_window=[0, -4],
holiday_upper_window=[0, 4, 7],
# priors
seasonality_prior_scale=[0.01, 1, 10],
holidays_prior_scale=[1, 10, 100],
changepoint_prior_scale=[0.01, 0.1, 1],
# extra regressors (if applicable)
<regressor>_prior_scale"= [0.01,0.1,1,10,100] (for each regressor)
```

B.2. examples of model fit and component decomposition

These figures can be interpreted in the same way as Figure 4.2

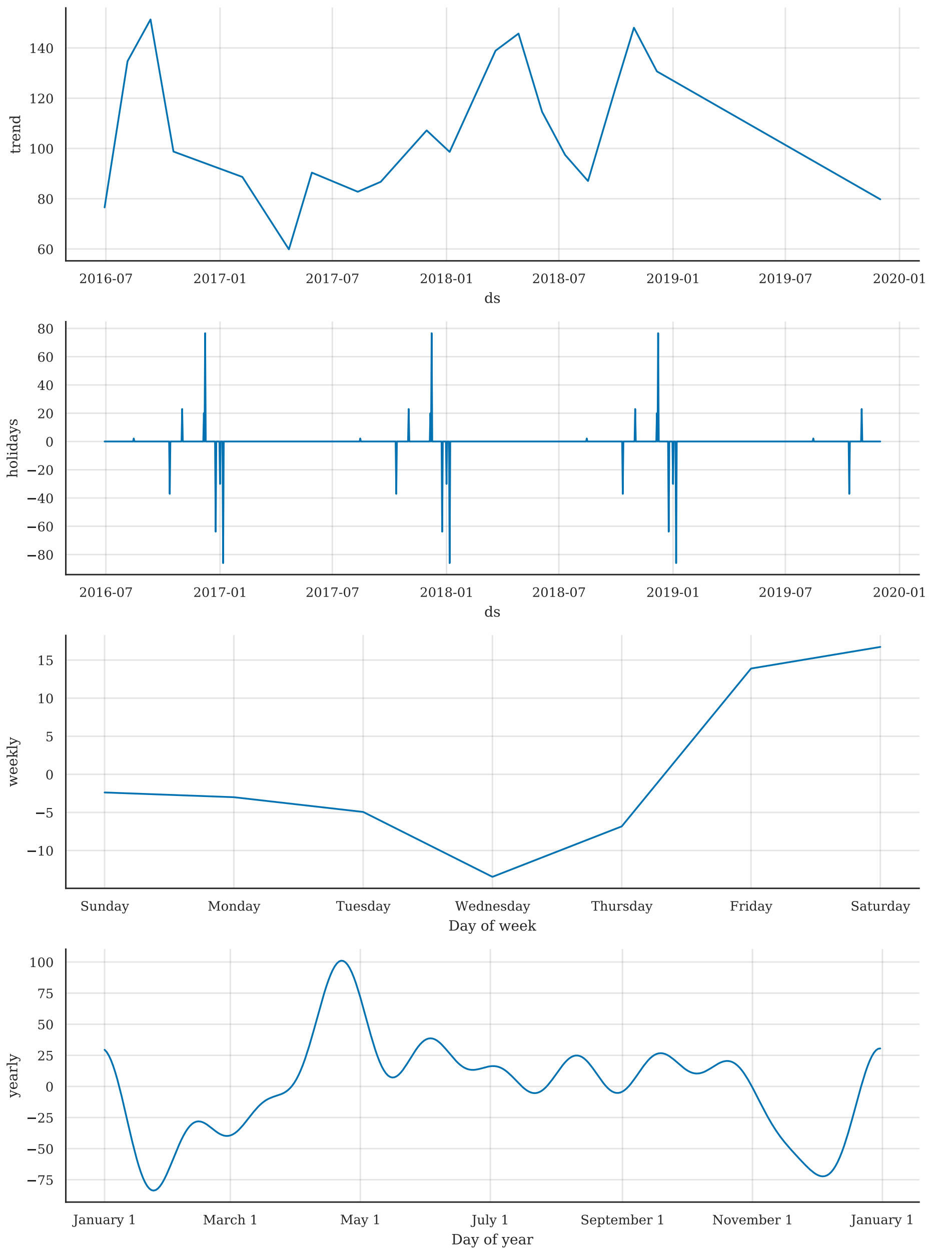


(a)

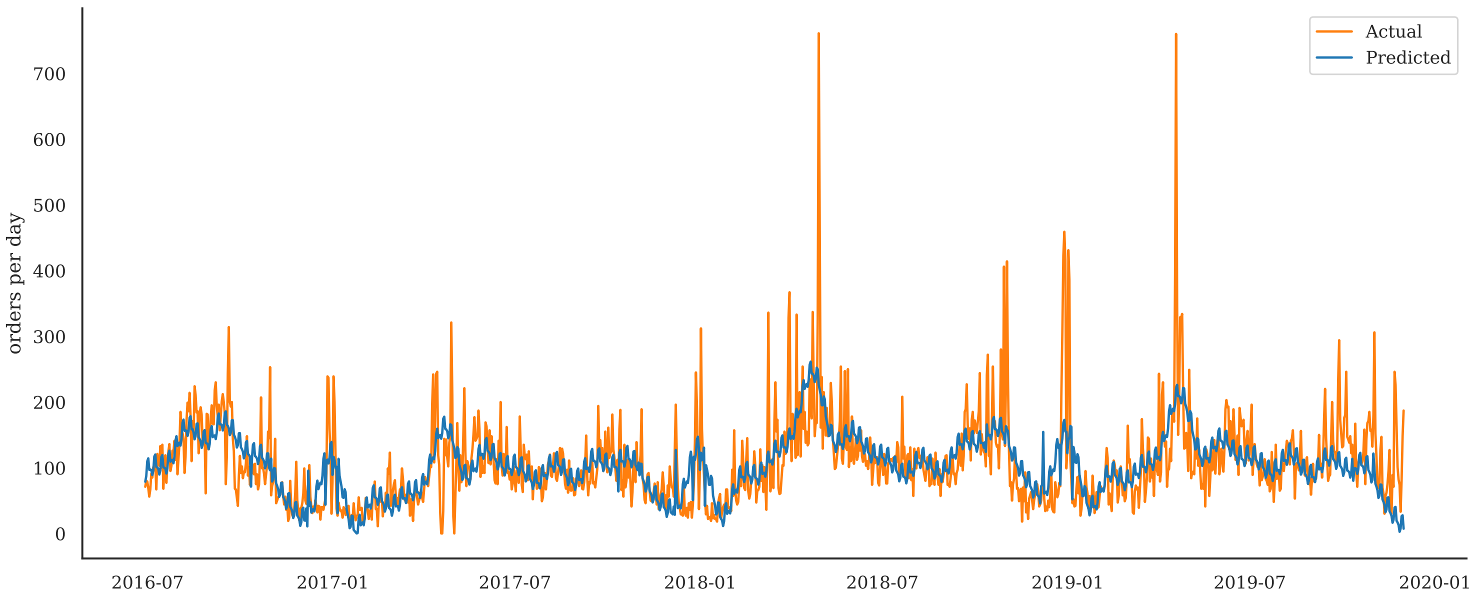


(b)

Figure B.1: Component decomposition for Venue 4

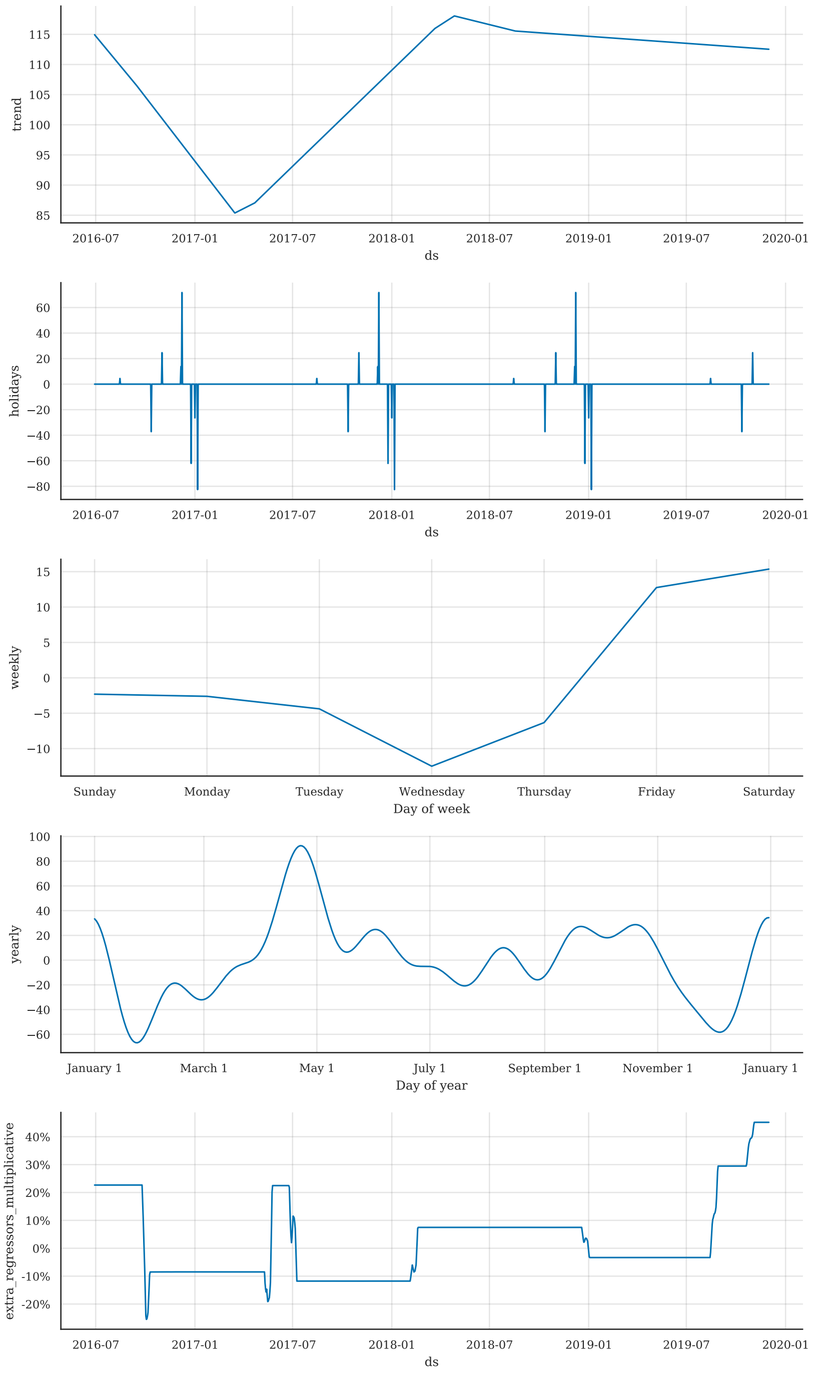


(a)

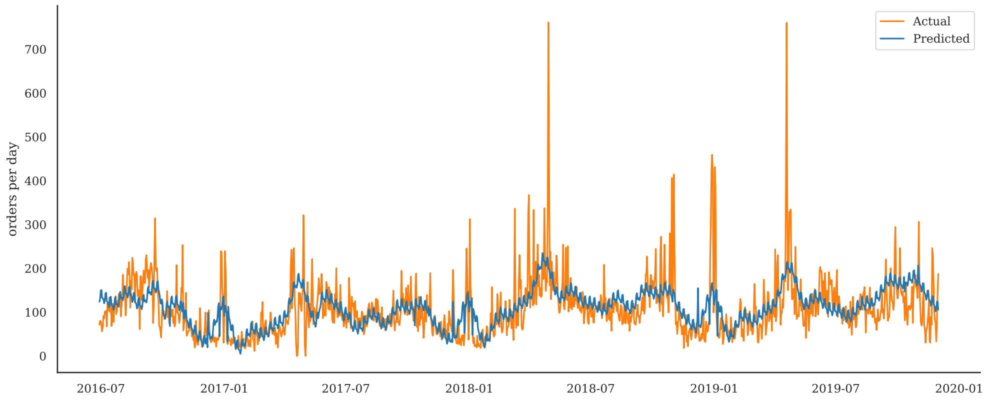


(b)

Figure B.2: Component decomposition for Venue 8



(a)



(b)

Figure B.3: Component decomposition for Venue 8 with price regressor (p=4 s=7 l=7)

C
Causality per price change

	venue title	price change date	MAE training	MAE 10 days	MAE 10 days after price change	MAE 80 days after price change
1	Venue 7	2018-05-17	29.60	66.85	49.88	50.52
2	Venue 7	2019-01-03	33.84	118.30	35.22	23.28
3	Venue 7	2019-02-06	34.38	13.94	26.96	38.36
4	Venue 7	2019-04-16	33.45	33.06	106.55	55.47
5	Venue 7	2019-05-23	33.89	26.72	63.92	66.02
6	Venue 7	2019-08-29	35.41	33.07	44.12	41.73
10	Venue 5	2018-04-23	7.47	13.36	25.53	32.51
11	Venue 5	2018-07-25	8.09	25.67	73.60	66.56
12	Venue 5	2019-04-02	13.44	16.95	13.96	17.96
16	Venue 3	2019-01-01	7.16	15.32	9.24	10.04
17	Venue 9	2018-04-30	39.85	215.89	161.24	248.53
18	Venue 9	2018-09-18	51.61	66.63	140.20	152.32
19	Venue 9	2018-10-09	53.17	144.01	131.51	116.98
20	Venue 9	2019-02-22	59.42	61.72	70.74	367.69
21	Venue 9	2019-04-15	63.37	580.94	268.53	236.32
22	Venue 9	2019-08-28	74.26	215.38	192.53	170.69
28	Venue 8	2018-12-19	28.91	20.25	67.13	34.75
29	Venue 8	2019-08-15	30.76	35.38	30.59	30.59
34	Venue 4	2018-05-15	37.31	235.52	208.35	300.15
35	Venue 4	2018-06-21	39.79	40.66	248.22	276.54
36	Venue 4	2018-08-16	45.73	64.56	30.17	71.60
37	Venue 4	2019-08-29	81.45	111.43	269.44	227.39
41	Venue 6	2018-07-22	25.13	43.67	39.82	99.16
42	Venue 6	2018-08-27	26.02	33.53	30.44	146.54
43	Venue 6	2018-12-19	36.06	114.99	123.76	226.66
44	Venue 6	2019-08-29	43.41	40.48	41.07	88.82
50	Venue 1	2019-07-18	35.71	83.27	52.70	65.84
56	Venue 2	2018-12-19	15.14	15.97	30.68	21.49
57	Venue 2	2019-08-14	17.11	40.77	52.48	38.72

Table C.1: Every price change date with associated effect.

D
Clustering

This chapter shows the detailed results from the both the high performing cluster discussed in section 6.5 and an average performing cluster. It contains a quick overview of the important metrics as well as the visualisations for the multiplicative scenario.

D.1. Scatterplot matrix

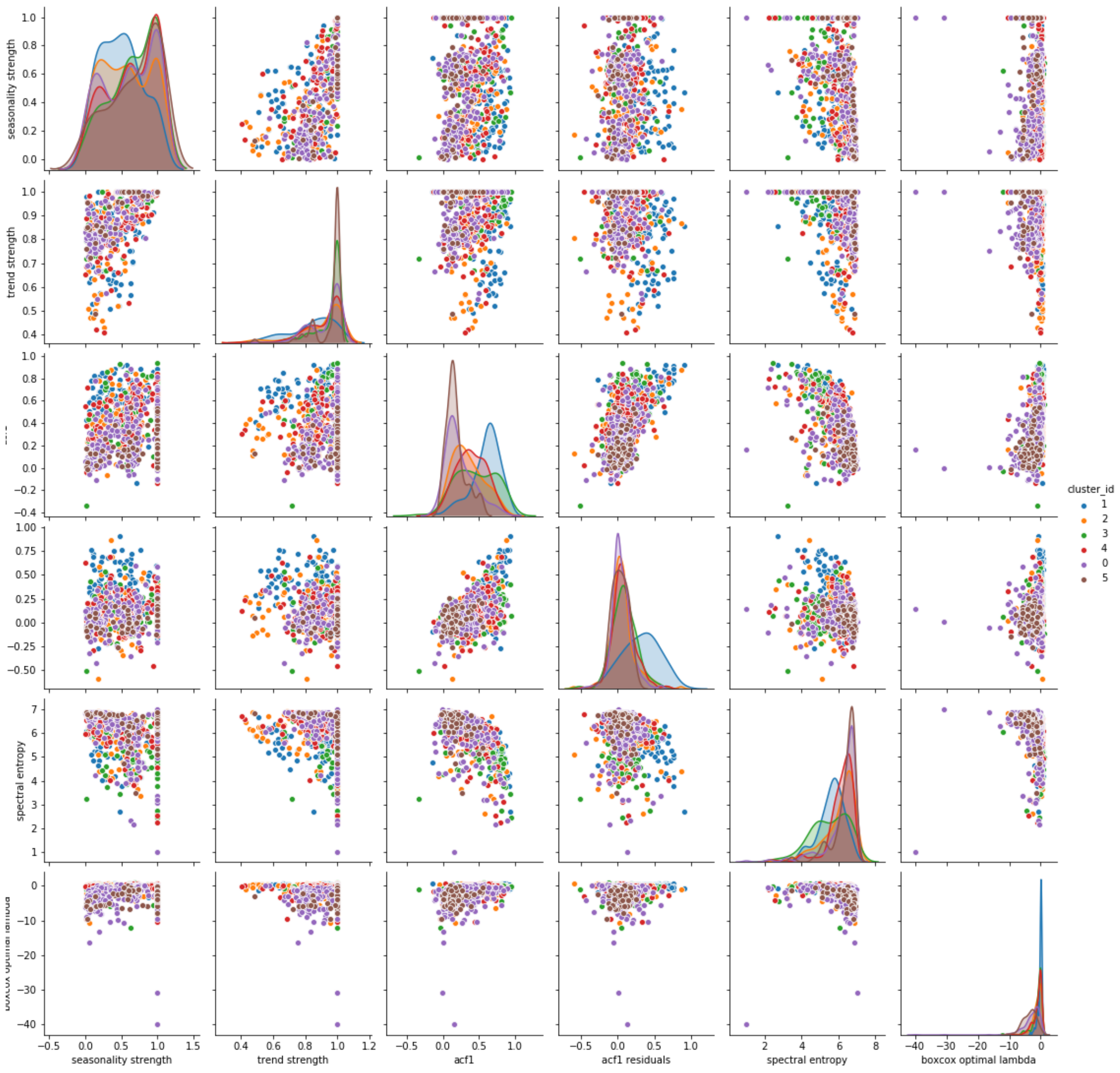


Figure D.1: Scatterplot matrix from features described in section 6.1.1. Clustering via KMeans with n=6

D.2. High performing cluster

D.2.1. cluster information

initial
cluster size = 108
treatment group size = 3
control group size = 105

metrics
Residual percentage = 0.0213
SMAPE = 29.16

D.2.2. effect simulation

additive

Figure D.2 and table D.1

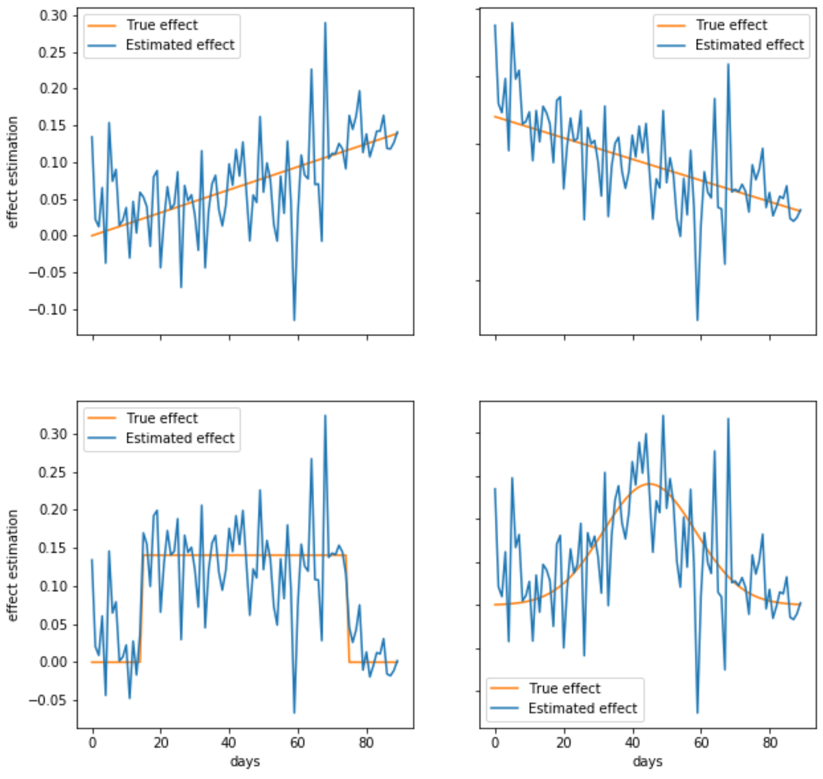


Figure D.2: Additive effect estimation for the high performing cluster.

Effect	Total mean sales treatment (Y_{total})	Total mean sales control (C_{total})	Total mean sales treatment w/o effect ($\hat{Y}_{total}$)	Estimated total effect ($\frac{\hat{Y}_{total}-C_{total}}{Y_{total}}$)	Actual total effect ($\frac{\hat{Y}_{total}-Y_{total}}{Y_{total}}$)
f_1	18.31	11.80	12.06	+53.98%	+51.85%
f_2	18.45	11.80	12.06	+55.15%	+53.02%
f_3	20.48	11.80	12.06	+72.04%	+69.91%
f_4	16.81	11.80	12.06	+41.53%	+39.39%

Table D.1: Total effect estimation values corresponding to figure D.2 for the high performing cluster.

multiplicative

Figure D.3 and table D.2

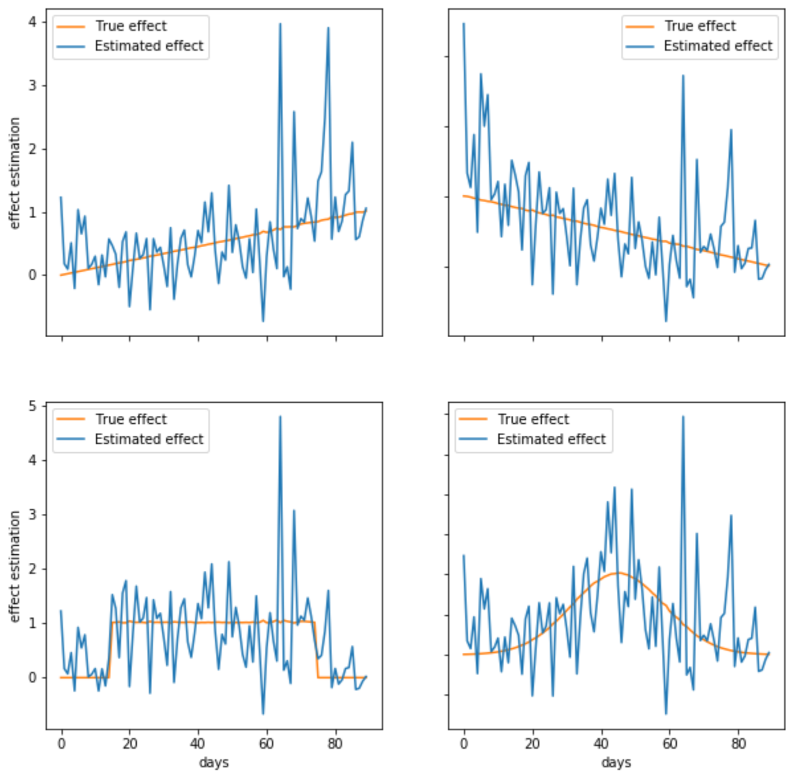


Figure D.3: Multiplicative effect estimation for the high performing cluster.

Effect	Total mean sales treatment (Y_{total})	Total mean sales control (C_{total})	Total mean sales treatment w/o effect ($\hat{Y}_{total}$)	Estimated total effect ($\frac{\hat{Y}_{total}-C_{total}}{Y_{total}}$)	Actual total effect ($\frac{\hat{Y}_{total}-Y_{total}}{Y_{total}}$)
f_1	17.58	11.80	12.06	+47.94%	+45.81%
f_2	18.78	11.80	12.06	+57.89%	+55.76%
f_3	20.25	11.80	12.06	+70.07%	+67.94%
f_4	16.86	11.80	12.06	+42.01%	+39.87%

Table D.2: Total effect estimation values corresponding to figure D.3 for the high performing cluster.

D.3. Average performing cluster

D.3.1. cluster information

initial

cluster size = 171
treatment group size = 3
control group size = 168

metrics

Residual percentage = 0.2550
SMAPE = 80.78

D.3.2. effect simulation

additive

Figure D.4 and table D.3

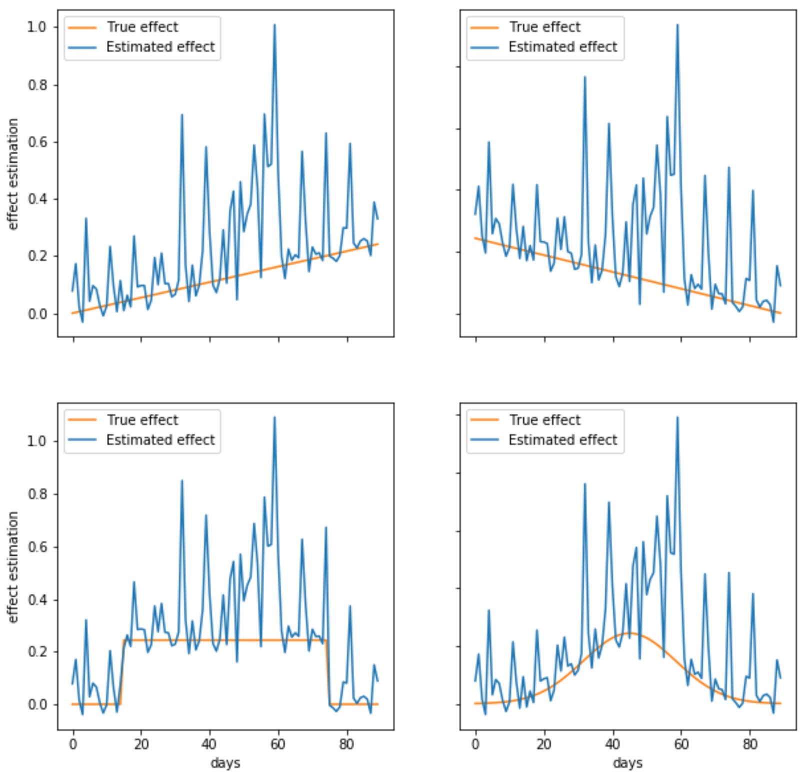


Figure D.4: Additive effect estimation for the average performing cluster.

Effect	Total mean sales treatment (Y_{total})	Total mean sales control (C_{total})	Total mean sales treatment w/o effect ($\hat{Y}_{total}$)	Estimated total effect ($\frac{\hat{Y}_{total}-C_{total}}{Y_{total}}$)	Actual total effect ($\frac{\hat{Y}_{total}-Y_{total}}{Y_{total}}$)
f_1	26.69	11.80	15.83	+94.04%	+68.53%
f_2	26.94	11.80	15.83	+95.58%	+70.07%
f_3	30.47	11.80	15.83	+117.91%	+92.40%
f_4	24.09	11.80	15.83	+77.58%	+52.07%

Table D.3: Total effect estimation values corresponding to figure D.4 for the average performing cluster.

multiplicative

Figure D.5 and table D.4

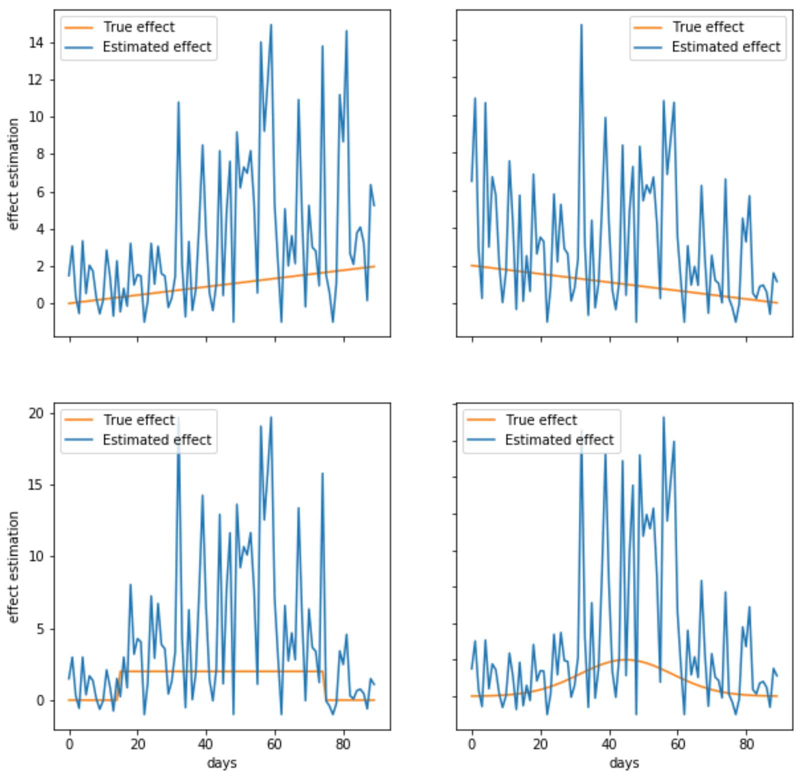


Figure D.5: Multiplicative effect estimation for the average performing cluster.

Effect	Total mean sales treatment (Y_{total})	Total mean sales control (C_{total})	Total mean sales treatment w/o effect ($\hat{Y}_{total}$)	Estimated total effect ($\frac{\hat{Y}_{total}-C_{total}}{Y_{total}}$)	Actual total effect ($\frac{\hat{Y}_{total}-Y_{total}}{Y_{total}}$)
f_1	32.29	11.80	15.83	+129.40%	+103.89%
f_2	31.06	11.80	15.83	+121.62%	+96.11%
f_3	40.50	11.80	15.83	+181.20%	+155.69%
f_4	31.40	11.80	15.83	+123.75%	+98.25%

Table D.4: Total effect estimation values corresponding to figure D.5 for the average performing cluster.

D.4. scores

Section 6.6 has the outliers for their scoring or the residuals percentage metric hidden, this is because it makes the plot unreadable. For transparency the results with outliers are included here.

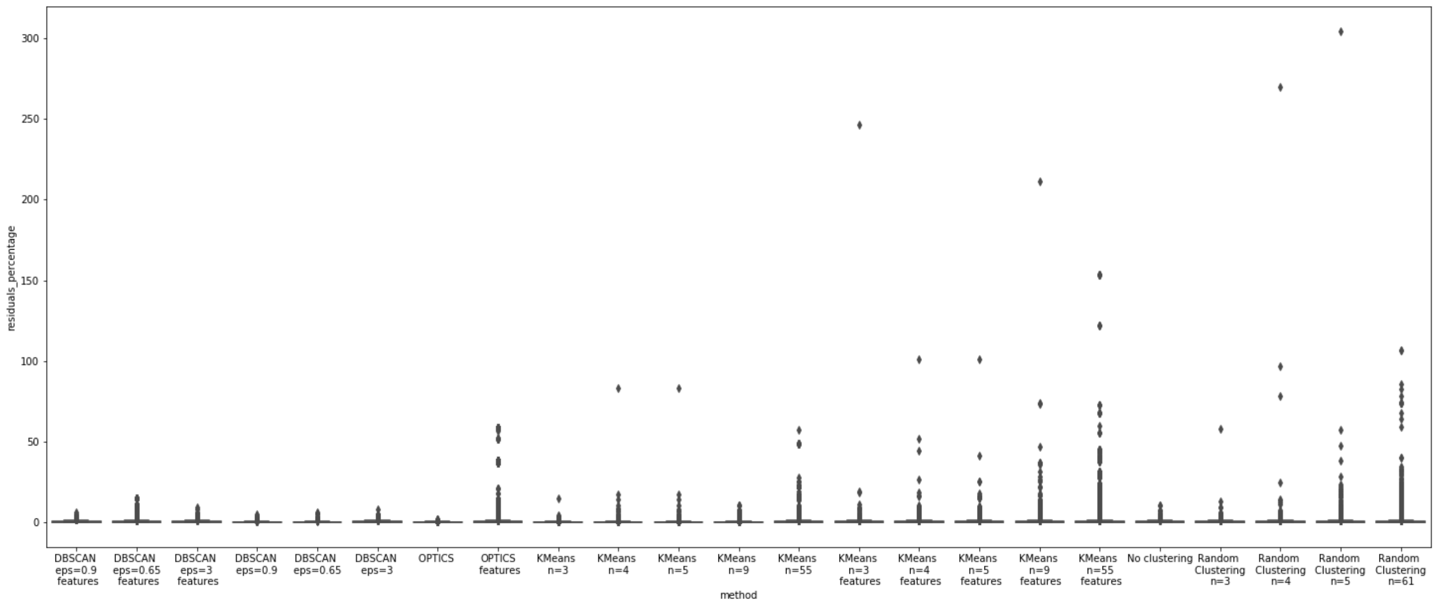


Figure D.6: Boxplot of different clustering methods/parameters on the residuals percentage metric with outliers

D.5. Countries

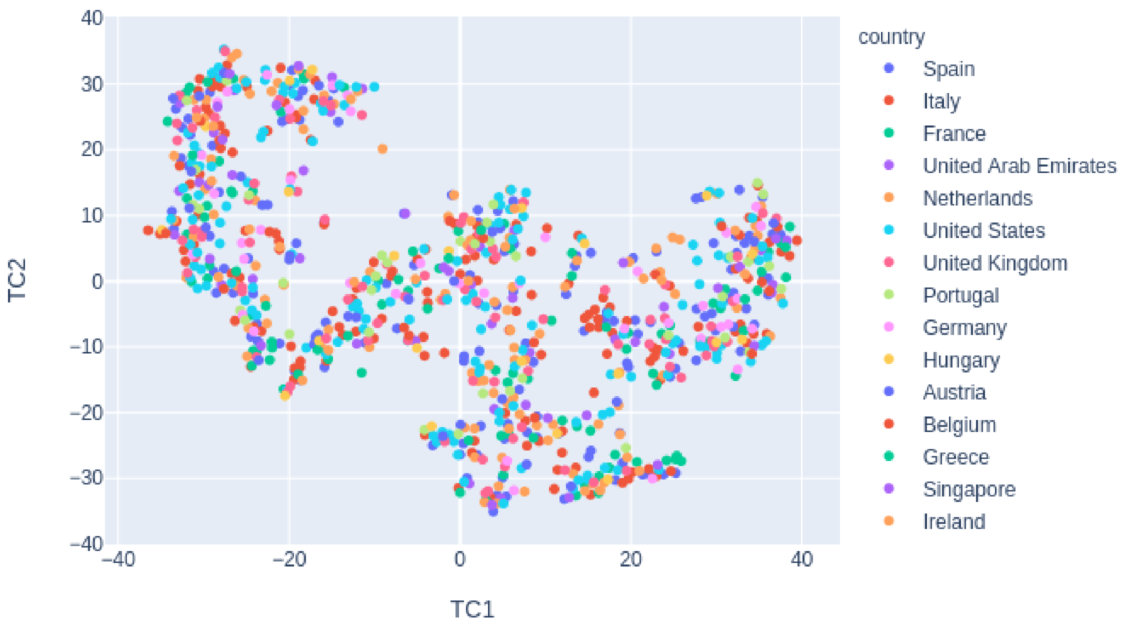


Figure D.7: TSNE on the data represented with the features from 6.1.1, colored by country

Bibliography

[1] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. Optics: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, SIGMOD '99, page 49–60, New York, NY, USA, 1999. Association for Computing Machinery. ISBN 1581130848. doi: 10.1145/304182.304187. URL `https://doi.org/10.1145/304182.304187`.

[2] Nava Ashraf, James Berry, and Jesse M Shapiro. Can higher prices stimulate product use? evidence from a field experiment in zambia. *American Economic Review*, 100(5):2383–2413, 2010.

[3] Christoph Bergmeir and José M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192 – 213, 2012. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2011.12.028. URL `http://www.sciencedirect.com/science/article/pii/S0020025511006773`. Data Mining for Software Trustworthiness.

[4] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society: Series B (Methodological)*, 26(2):211–243, 1964.

[5] Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. Inferring causal impact using bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1):247–274, Mar 2015. ISSN 1932-6157. doi: 10.1214/14-aoas788. URL `http://dx.doi.org/10.1214/14-AOAS788`.

[6] Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[8] Ahmad Faruqui. The ethics of dynamic pricing. In *Smart Grid*, pages 61–83. Elsevier, 2012.

[9] Kris Johnson Ferreira, Bin Hong Alex Lee, and David Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2016.

[10] Marshall Fisher, Santiago Gallino, and Jun Li. Competition-based dynamic pricing in online retailing: A methodology validated with field experiments. *Manage. Sci.*, 64(6):2496–2514, June 2018. ISSN 0025-1909. doi: 10.1287/mnsc.2017.2753. URL `https://doi.org/10.1287/mnsc.2017.2753`.

[11] Ahmed Z Gabr, Ahmed A Helal, and Nabil H Abbasy. Dynamic pricing; different schemes, related research survey and evaluation. In *2018 9th International Renewable Energy Congress (IREC)*, pages 1–7. IEEE, 2018.

[12] Alfredo Garcia, Enrique Campos-Nañez, and James Reitzes. Dynamic pricing and learning in electricity markets. *Operations Research*, 53(2):231–241, 2005.

[13] Vishal Gaur and Marshall L Fisher. In-store experiments to determine the impact of price on sales. *Production and Operations Management*, 14(4):377–387, 2005.

[14] Uri Gneezy and Aldo Rustichini. Pay enough or don't pay at all. *The Quarterly journal of economics*, 115(3):791–810, 2000.

[15] Christian B Hansen. Asymptotic properties of a robust variance matrix estimator for panel data when t is large. *Journal of Econometrics*, 141(2):597–620, 2007.

[16] A. C. Harvey and S. Peters. Estimation procedures for structural time series models. *Journal of Forecasting*, 9(2):89–108, 1990. doi: 10.1002/for.3980090203. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/for.3980090203`.

[17] Brian Hopkins and John Gordon Skellam. A new method for determining the type of distribution of plant individuals. *Annals of Botany*, 18(2):213–227, 1954.

[18] Rob J Hyndman. robjhyndman.com, Nov 2017. URL `https://robjhyndman.com/hyndsight/forecasting-competitions/`.

[19] Rob J Hyndman. Feature-based time series analysis, 2019.

[20] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[21] Dean Karlan and Jonathan Zinman. Observing unobservables: Identifying information asymmetries with a consumer credit field experiment. *Econometrica*, 77(6):1993–2008, 2009.

[22] Robert Kohn, Michael Smith, and David Chan. Nonparametric regression using linear combinations of basis functions. *Statistics and Computing*, 11(4):313–322, 2001.

[23] Pierre Legendre and Loic FJ Legendre. *Numerical ecology*. Elsevier, 2012.

[24] George G Lorentz. *Bernstein polynomials*. American Mathematical Soc., 2013.

[25] Serguei Netessine. Dynamic pricing of inventory/capacity with infrequent price changes. *European Journal of Operational Research*, 174(1):553–580, 2006.

[26] Mardavij Roozbehani, Munther Dahleh, and Sanjoy Mitter. Dynamic pricing and stabilization of supply and demand in modern electric power grids. In *2010 First IEEE International Conference on Smart Grid Communications*, pages 543–548. IEEE, 2010.

[27] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3): 1–21, 2017.

[28] Tom A. B. Snijders. On cross-validation for predictor evaluation in time series. In Theo K. Dijkstra, editor, *On Model Uncertainty and its Statistical Implications*, pages 56–69, Berlin, Heidelberg, 1988. Springer Berlin Heidelberg. ISBN 978-3-642-61564-1.

[29] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In *New directions in statistical physics*, pages 273–309. Springer, 2004.

[30] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974. doi: 10.1111/j.2517-6161.1974.tb00994.x. URL `https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1974.tb00994.x`.

[31] Sean J. Taylor and Benjamin Letham. Forecasting at scale. *PeerJ PrePrints*, 5:e3190, 2017.

[32] Francesco Trovò, Stefano Paladino, Marcello Restelli, and Nicola Gatti. Improving multi-armed bandit algorithms in online pricing settings. *International Journal of Approximate Reasoning*, 98:196–235, 2018.

[33] Shuangren Zhao, Kang Yang, and Xintie Yang. Reconstruction from truncated projections using mixed extrapolations of exponential and quadratic functions. *Journal of X-ray Science and Technology*, 19(2): 155–172, 2011.