Domain Adaptation for Retail Demand Prediction

Niloofar Tarighat, Department of Electrical and Computer Engineering

McGill University, Montreal

May, 2022

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of

Masters of Science in Electrical Engineering

Abstract

Demand Forecasting is an important tool in many industries including retail. Although many approaches have been developed to accurately predict the demand of products based on their historical sales data, demand prediction is still a complex issue especially when there is a domain shift between training and testing data.

In this work, we study three examples of domain shifts in the context of retail: outbreak of the COVID-19 pandemic, opening a new store, and introducing a new product. We first show that the accuracy of demand prediction models suffers after each sudden change. Then, we use domain adaptation methods, such as Frustratingly Easy (FE) and Kernel Mean Matching (KMM) to help improve the demand prediction accuracy by leveraging the available data from the period before the shift (source domain) and adapting it to the data after the shift (target domain). Additionally, we show that using a pairing technique further helps improve the prediction accuracy.

We use two methods as our base forecasting model: XGBoost and Transformers, and we show that in the context of our data, it is better to use XGBoost.

Our dataset comprises of point-of-sales data from 89 locations of Alimentation Couche-Tard convenient stores in the island of Montreal gathered between 2019-07 and 2021-02. We use product price information in addition to sales information to predict the demand of products in each store. In this study, we focus our attention on the two high-selling categories of coffee and energy drinks.

Abrégé

La prévision de la demande est un outil important dans de nombreux secteurs, y compris le commerce de détail. Bien que de nombreuses approches aient été développées pour prédire avec précision la demande de produits en fonction de leurs données de vente historiques, la prévision de la demande reste un problème complexe, en particulier lorsqu'il existe un changement de domaine entre les données de formation et de test.

Dans ce travail, nous étudions trois exemples de changement de domaine dans le contexte du commerce de détail : l'apparition de la pandémie de COVID-19, l'ouverture d'un nouveau magasin et l'introduction d'un nouveau produit. Nous montrons d'abord que la précision des modèles de prévision de la demande souffre après chaque changement soudain. Ensuite, nous utilisons des méthodes d'adaptation de domaine telles que Frustratingly Easy (FE) et Kernel Mean Matching (KMM) pour aider à améliorer la précision des prévisions de la demande en utilisant les données disponibles avant le changement (domaine source) et en les adaptant aux données après le changement (domaine cible). De plus, nous montrons que l'utilisation d'une technique d'appariement permet d'améliorer encore plus les précisions.

Nous utilisons deux méthodes comme modèle de prévision de base : XGBoost et Transformers et nous montrons que dans le contexte de nos données, il est préférable d'utiliser XGBoost.

Notre ensemble de données comprend les données des points de vente de 89 emplacements de dépanneurs Alimentation Couche-Tard sur l'île de Montréal. Nous utilisons

les informations sur les prix des produits en plus des informations sur les ventes pour prévoir la demande de produits dans chaque magasin. Dans cette étude, nous concentrons notre attention sur les deux catégories les plus vendues : le café et les boissons énergisantes.

Acknowledgements

I would like to express my sincere gratitude to my master's supervisors, Professor James Clark and Professor Maxime Cohen, for their insightful feedback, continuous guidance, and endless support throughout my research. Their motivation, enthusiasm, and immense knowledge have deeply inspired me. I truly appreciate the freedom they gave me in exploring my ideas.

I am also grateful to Alimentation Couche-Tard and the McGill Retail Innovation Lab for their efforts that made this project possible. I am grateful to Magnus Tagtstrom, Zahoor Saeed Chughtai, and Eve Fontaine for their support in providing me with the data and tools necessary to complete my research.

I would like to thank my family and friends for their support and kindness throughout my master's research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Demand Prediction or Demand Forecasting is an important tool in many industries, especially in retail. Retail is a very wasteful industry, whether it is caused by changing clothing trends or due to the nature of perishable items in grocery stores [FAO-Gustavsson et al., 2011]. Accurate demand prediction can help reduce this waste, and yield a positive environmental impact.

Although many approaches have been developed to accurately predict the demand of products based on their historical sales data [Winters, 1960a] [Gardner Jr, 1985], demand prediction is still a complex issue especially when there is a shift [Huyen,] between training and testing data. In such situations, the data from the period before the shift may be no longer relevant, and the amount of data in the new domain is often not sufficient to train a predictive model.

In the context of retail, this domain shift can be due to a sudden change such as the outbreak of the COVID-19 pandemic [Ciotti et al., 2020]. Two other common shifts in retail is the opening of a new store or the launch of new product. In such cases, we have to wait until sufficient data is gathered in the new domain before we can predict the demand with an accuracy that is comparable to before the domain shift.

This period of not having access to accurate demand prediction can be costly for retailers. Domain Adaptation methods can be used to help close this gap in prediction accuracy by leveraging available data from the previous domain (source domain) and adapting it to the new domain (target domain).

In this thesis, we consider three case studies of domain shift and confirm that a domain shift negatively affects the accuracy of demand prediction models. Then we use domain adaptation techniques [Daumé III, 2009] [Sun et al., 2016] [Sun and Saenko, 2016] to mitigate this negative effect.

Chapter 2 reviews the time series forecasting literature. We divide the existing methods into two categories: traditional [Winters, 1960a] [Gardner Jr, 1985] and modern [Chen et al., 2015] [Vaswani et al., 2017] . We next consider state of the art in demand forecasting as a special case of time series forecasting. In the rest of this chapter, we study domain adaptation techniques and consider some of the most common ones.

Chapter 3 starts by describing the characteristics of our dataset. Furthermore, we describe three case studies we consider in this work. We explain the goals of each case study and the architectures used.

Chapter 4 discusses our criteria for baseline selection and presents the results of each experiment. Finally, the managerial implications of the previous results are discussed.

We conclude our work by summarizing our findings and mentioning some feature steps.

I developed all of the techniques and ran all of the experiments described in the thesis. My co-supervisors Professor J. Clark and Professor M. Cohen provided advice during the research.

Chapter 2

Background and Related Work

2.1 Time Series Forecasting

Time series forecasting has always been an influential part of research because of its various applications in numerous significant fields.

In finance, it plays a critical role in determining which stocks or options should be bought or sold at what time frame. Complex algorithms based on time series prediction are embedded in appropriate hardware and make such decisions in a fraction of a second [Andersen et al., 2005].

Time series prediction is an integral part of the retail industry. It helps retailers determine how much of each product they should expect to sell at different times. This information is critical in organizing the supply chain and ensuring that customers are not facing empty shelves and at the same time that the stores do not carry excess inventory [Huber and Stuckenschmidt, 2020, Ma and Fildes, 2021, Mahmoudyan and Zeqiri, 2021].

Climate change is one of today's pressing issues. Time series prediction is also a key element in climate studies. The algorithms are used to predict future trends in climate change. For instance, the application of state-of-the-art statistical methods to the time

series of global surface temperatures conclude accelerated warming since the year 1974 [Mudelsee, 2019].

The time-series data comprise three components, which are discussed next:

1. **Trend**: A trend refers to a long-term increase or decrease in the data, which does not necessarily have to be linear. A trend can also be referred to as a "changing direction" when it might go from an increasing trend to a decreasing trend.

2. **Seasonality**: A seasonal pattern occurs when a time series is affected by seasonal factors such as the time of the year or the day of the week. Seasonality is always of a fixed and known frequency.

3. **Cyclic**: A cycle occurs when the data exhibit rises and falls that are not of a fixed frequency. These fluctuations are usually due to economic conditions and are often related to the "business cycle." The duration of these fluctuations is usually at least two years.

   Cyclic behaviour can easily be confused with seasonal behaviour, although they are different. If the fluctuations have a fixed frequency, they are cyclic; if the frequency is unchanging and associated with some aspect of the calendar, then the pattern is seasonal. The length of the cycle is usually larger than the length of the seasonality.

   Many time series include trends, cycles, and seasonality. When choosing a forecasting method, we first need to identify the time series patterns in the data and then select an approach that can properly capture the patterns [Hyndman, 2018].

   In **Figure 2.5**, we can infer that there exists an increasing trend in the sales of anti-diabetic drugs over the years. These sales also show seasonality that might be due to the change in the drug price at the end of the calendar year.

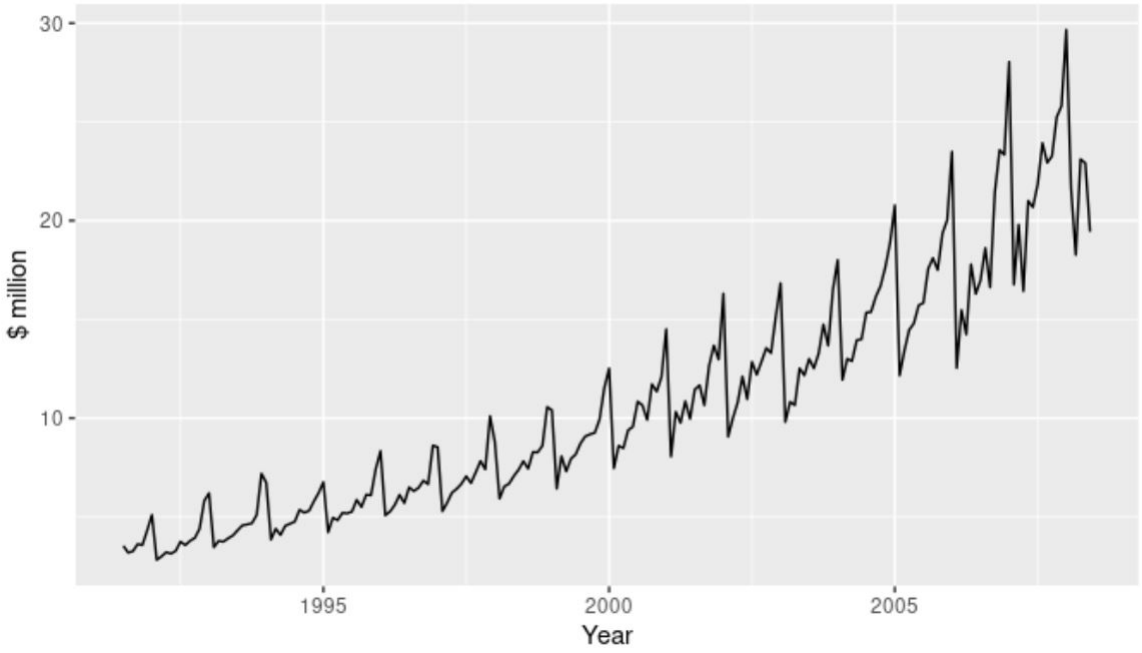


Figure 2.1: Monthly sales of anti-diabetic drugs in Australia. Image credit: [Hyndman, 2018]

2.1.1 Traditional Approaches

Traditional approaches in time series prediction are parametric models such as auto-regressive and exponential smoothing [Gardner Jr, 1985, Winters, 1960a].

1. Auto-Regressive models

Unlike linear regression [Weisberg, 2005], in auto-regressive models, the variable of interest is predicted using a linear combination of only the past values of the variable. In other words, lagged versions of the variable are used to predict its future value.

The general format of an auto-regressive model of order p can be seen below:

$$y_t = c + a_1 y_{t-1} + a_2 y_{t-2} + ... + a_p y_{t-p} + \epsilon_t,$$

where ϵ_t represents the white noise. Auto-regressive models perform well in predicting a wide range of different time series patterns [Hyndman, 2018].

2. Moving Average

The formulation of weighted average is as below:

$$y_t = (y_{t-1} + y_{t-2} + ... + y_{t-p})/p,$$

where p is the size of the sliding window.

3. Weighted Moving Average

We can extend the moving average equation to a weighted form. The formula of this method is given by:

$$y_t = (w_{t-1}y_{t-1} + w_{t-2}y_{t-2} + ... + w_{t-p}y_{t-p})/(w_{t-1} + w_{t-2}... + w_{t-p}),$$

where p is the size of the sliding window. We note that the sum of the weights is naturally designed to be equal to one.

An special case of weighted moving average would be exponential weighted moving average. [Winters, 1960b] has explored this method to forecast sales. [Holt, 2004] has done so to predict seasonality and trends.

4. Exponential Smoothing

Exponential smoothing is used for smoothing time series data using the exponential window that is used to assign exponentially decreasing weights over time. Whereas in the simple moving average the past observations are weighted equally.

A simple form of exponential smoothing is as follows:

$$s_0 = x_0,$$

$$s_t = ax_t + (1 - a)s_{t-1}.$$

The raw data is represented by x and the output of the exponential smoothing is represented by the variable s.

5. Autoregressive Integrated Moving Average (ARIMA)

ARIMA considers standard structures in time series data. However, it is still a powerful forecasting model for time series. ARIMA stands for Autoregressive Integrated Moving Average. It is a generalization of the simpler Autoregressive Moving Average and adds the notion of integration.

Here we explain each characteristic of the model:

AR: Autoregression. As described in Section 1, this feature considers a relationship between a variable and some of its lagged versions.

I: Integrated. The use of differencing of raw observations (e.g., subtracting an observation from the observation at the previous time step) to make the time series stationary, i.e., to remove the trend and the seasonality that negatively affect the regression model.

MA: Moving Average. A model that uses the dependency between an observation and a residual error from a moving average model is applied to lagged observations.

These components make up the parameters in the model in a standard notation of ARIMA (p, d, q).

The parameters of the ARIMA model are defined as follows:

p: referred to as the lag variable, is the number of lagged variables included in the model.

d: The number of times the raw observations are differenced is also called the degree of difference.

q: The order of moving average is the size of the moving average window.

SEATS models which stand for Seasonal Extraction in ARIMA Time Series were developed in Bank of Spain and now are widely used in order to decompose the time series.

This method only works for monthly and quarterly seasonality [Dagum and Bianconcini, 2016]. X1, X12 ARIMA, and X13 ARIMA are examples of other decomposition methods [Dagum and Bianconcini, 2016].

2.1.2 Modern Approaches

Great advancements in deep learning, especially natural language processing, image classification, and reinforcement learning, have increased the interest in such models for time series prediction.

Machine learning methods enable us to predict the time series without the extra hassle of feature engineering. These methods perform in a purely data-driven approach, in contrast to traditional approaches that require some level of expert knowledge [Lim and Zohren, 2021].

Here, we discuss some of the most prominent machine learning models used to predict time series.

2.1.3 XGBoost

XGBoost [Chen et al., 2015], stands for Extreme Gradient Boosting. It is a scalable, distributed gradient-boosted decision tree (GBDT).

XGBoost is similar to Random Forest [Breiman, 2001] since both algorithms use a collection of decision trees. Their difference is rooted in how they selected the collection. Random Forest uses the bagging technique while XGBoost uses gradient boosting.

In Gradient Boosting, an ensemble of decision trees are trained iteratively. At each iteration, the error residuals of the previous model are used to fit the next model. A weighted sum of the tree predictions at each iteration is considered as the final model. The bagging approach in Random forest minimizes the variance and overfitting, whereas the "boosting" approach in the gradient boosting minimizes the bias and underfitting.

XGBoost is a highly scalable end-to-end tree boosting system that uses parallel tree learning instead of sequential [Chen et al., 2015].

Convolutional Neural Networks (CNN)

Convolutional Neural Networks or CNNs are designed to extract spatially local features from the input data. CNNs were designed for images, so we cannot directly apply them to time series data because time series have a natural order in them [Krizhevsky et al., 2012a, Goodfellow et al., 2016]. To be able to use CNNs for time series data, causal convolutions have been designed. These convolutions make sure that the future information is not used [Borovykh et al., 2017]. A simple expression of such convolutions is given by:

$$h_t^{l+1} = A((W * h)(l, t)),$$

$$(W * h)(l, t) = \sum_{\tau=0}^{k} W(l, \tau) h_{t-\tau}^{l}.$$

The variable A is an activation function, h_t^l is an intermediate state layer at layer l and step t, and $W(l, \tau)$ is the filter at layer l. A more intuitive understanding of this structure is reported in **Figure 2.2**

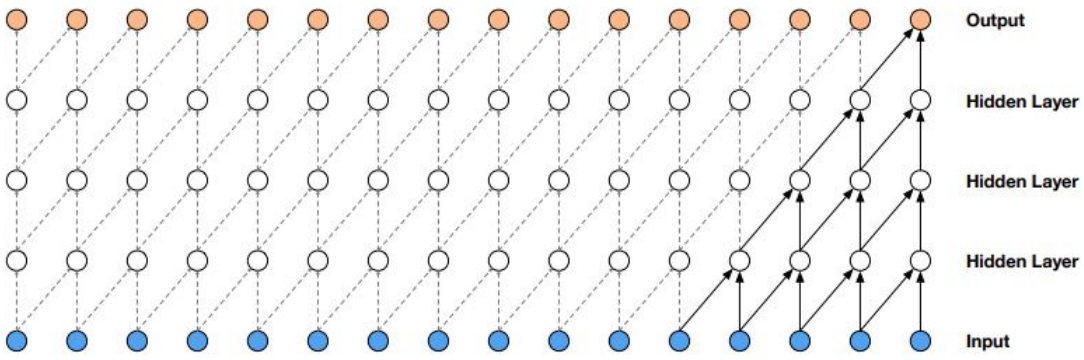


Figure 2.2: Intuitive graph showing the structure of causal convolution [Oord et al., 2016]

.

Dilated Convolutions are one type of convolutions that reduce the challenges of computational complexity [Oord et al., 2016]. A dilated causal convolution would follow the form below:

$$(W * h)(l, t, d_l) = \sum_{\tau=0}^{\lfloor k/d_l \rfloor} W(l, \tau) h^l_{t-d_l\tau},$$

where $\lfloor\rfloor$ is the floor operator and $d_{l\tau}$ is a layer-specific dilation rate. Dilated convolutions can hence be interpreted as convolutions of a down-sampled version of the lower layer features – reducing the resolution to incorporate information from the distant past. As such, by increasing the dilation rate with each layer, dilated convolutions can gradually aggregate information at different time blocks, allowing for more history to be used in an efficient manner [Lim and Zohren, 2021]. **Figure 2.3** shows the inner workings of such structure.

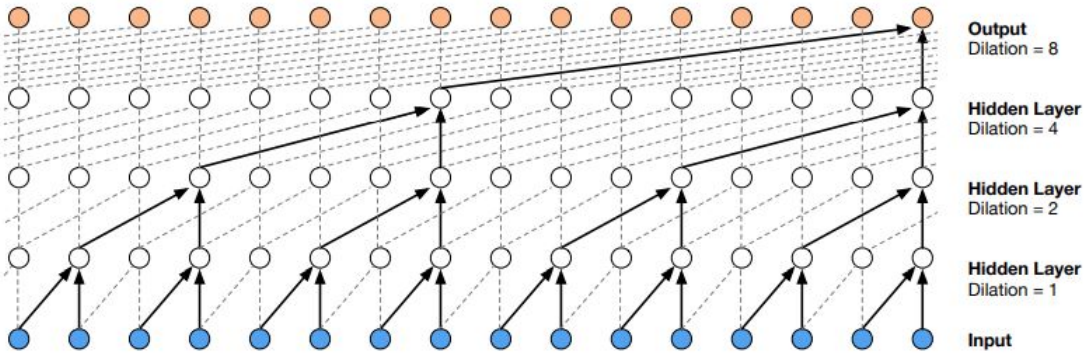


Figure 2.3: Intuitive graph showing the structure of dilated causal convolution [Oord et al., 2016].

Recurrent Neural Networks (RNN)

Recurrent Neural Networks (RNN) have been proven very powerful in natural language processing tasks [Young et al., 2018]. Since time series data is in nature also a sequence, these methods have been used to predict time series [Salinas et al., 2017, Rangapuram et al., 2018,Lim et al., 2019,Wang et al., 2019]. RNNs have a memory state, which compacts information from the past values of the time series. This memory is updated at each step

with regards to the current value of time series as shown in **Figure 2.4** [Lim and Zohren, 2021].

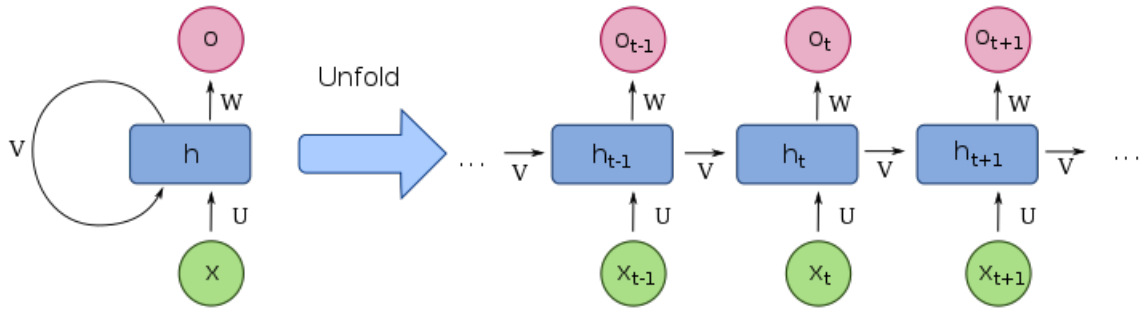


Figure 2.4: Graph showing the structure of a basic RNN, folded (left) and expanded (right) [Kim et al., 2021].

Due to problems with vanishing and exploding gradients [Goodfellow et al., 2016], RNNs might suffer when applying them to long time series data. To solve this problem, long short-term memory networks or LSTMs have been introduced [Hochreiter and Schmidhuber, 1997]. LSTMs use a cell state to improve the gradient flow through the network.

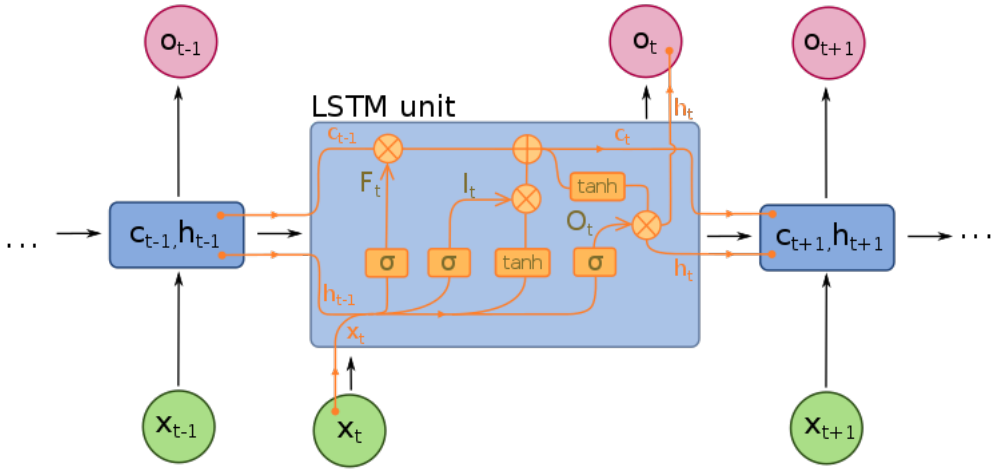


Figure 2.5: Graph showing the structure of LSTM unit [Hochreiter and Schmidhuber, 1997].

Transformers

[Vaswani et al., 2017] proposed the Transformer architecture, which has become the state-of-the-art model in natural language processing. This model is solely based on the attention mechanism and removes the need for convolution and recurrence. The Transformer allows parallelization, so it is more efficient when compared to competing models. Since attention mechanisms have played a significant role in their superior performance, we should first understand the concept of attention.

Attention Attention Mechanisms help the model to focus on parts of the sequence that are important, even if they are far past. Attention can be considered as a key-value lookup based on a query [Graves et al., 2014]. Attention mechanisms aggregate past information using dynamically generated weights [Lim and Zohren, 2021]. In more detail, given the key, value, and query, the output is computed as a weighted sum of the values, where the weight assigned to each value is computed based on the compatibility of the query and the given key [Vaswani et al., 2017]. The formulation of the output of attention mechanism can be written as below:

$$Attention(V, K, Q) = softmax(\frac{QK^T}{\sqrt{d_k}}).$$

An extension of the dot-product attention (formulated above) is the multi-head attention. It would be more efficient if, instead of doing a single attention, multiple attention mechanisms could be performed at the same time [Vaswani et al., 2017].

Multi-head attention allows the model to jointly attend to information from different representation sub-spaces at distinct positions by projecting the key, values, and query with different projection matrices, performing attention on each of these projected versions in parallel, and then concatenating the results. The structure of multi-head attention can be seen in **Figure 2.6**.

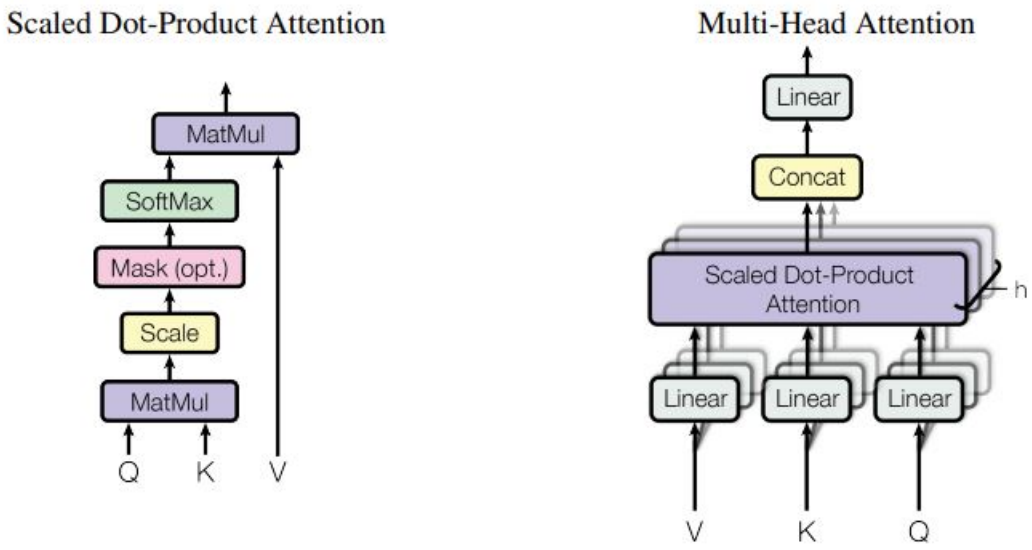


Figure 2.6: The structure of multi-head attention: (left) Scaled Dot-Product Attention. (right) Multi-Head attention [Vaswani et al., 2017].

Formulation of such attention would be as follows:

$$Multihead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V),$$

where W_i^Q, W_i^K, W_i^V, and W^O are the projections.

The Transformer architecture is comprised of an encoder and a decoder. The encoder maps a sequence of inputs to the state value z, and the decoder generates a sequence of elements given variable z, one value at a time. The generation process is auto-regressive, meaning that only the past values are used to generate the new element [Vaswani et al., 2017]. **Figure 2.7** shows the architecture of the Transformer model.

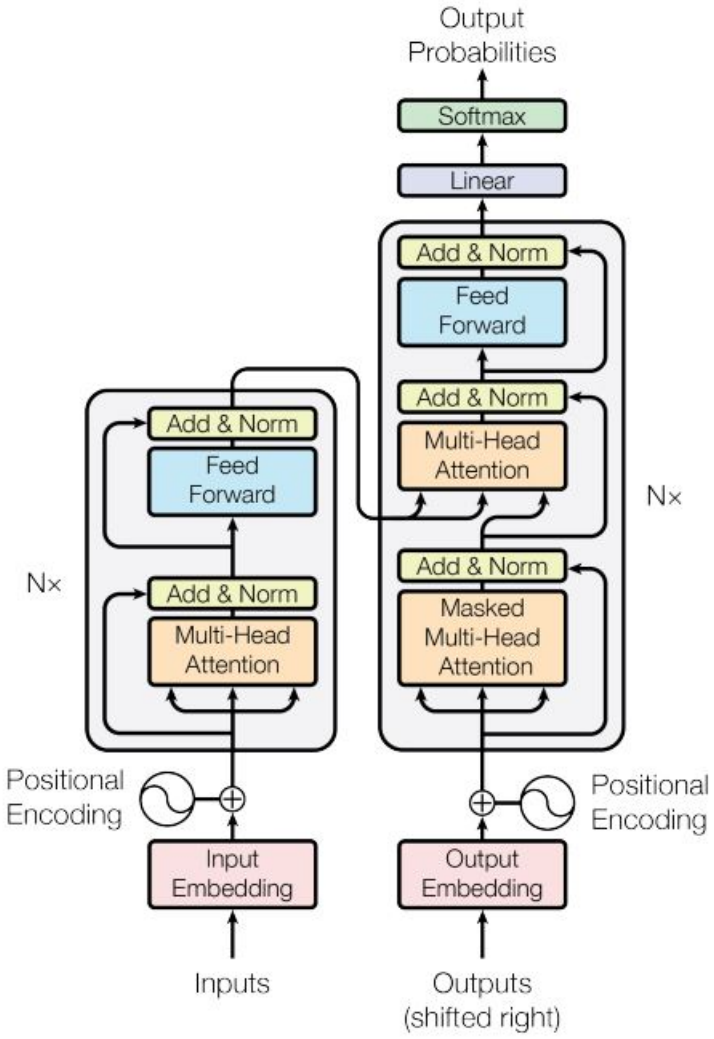


Figure 2.7: Architecture of Transformer model [Vaswani et al., 2017].

2.2 Demand Forecasting

2.2.1 Retail Demand Forecasting

Demand Forecasting is an integral part of the retail industry since many important organizational decisions, such as pricing, inventory management, distribution, and replenishment are done based on product demand [Ma and Fildes, 2021]. Retail is by nature a very wasteful industry, so that accurate demand forecasting can mitigate this issue, increase customer satisfaction, and ultimately increase revenue and profits [Ma and Fildes, 2021].

Retail demand forecasting mostly deals with predicting the demand of each product over a large number of stores for a short horizon [Ma and Fildes, 2021].

According to [Ma and Fildes, 2021], retail organizations need to predict the demand for three levels of decisions:

1. **Strategic level**: Retailers need long-term forecasts to set their organizational strategy for the years ahead. For example, a prediction of an increase in online sales should encourage the retailer to invest more into its online presence. This level deals with decisions in the span of years ahead.

2. **Tactical level**: Tactical decisions fit under the strategy of the organization. This level deals with more short-term decisions (e.g., a few months) and determines the communication and advertising plans of the retail organization [Ma and Fildes, 2021].

3. **Operational level**: A retail organization should make day-to-day operational decisions to satisfy its strategy and tactics. These decisions concern handling the supply planning process, staffing schedules, and avoiding customer service issues [Ma and Fildes, 2021].

Any demand prediction in retail is aggregated over a certain level, such as products, locations, and time brackets, based on the goal of the prediction. **Market-level aggregation** refers to sales being aggregated over a certain region or country [Cohen et al., 2019]. This type of forecast is substantial when defining the strategy of the organization. **Chain- and channel-level aggregation** are needed for the organization's financial management. **Store-level aggregation** deals with predicting the demand for products in a store. Sales in each store are significantly affected by the local demographics and economic factors. Store-level forecasting can be divided into two groups: (1) Sales forecasting for existing stores, and (2) New Store sale forecasting [Ma and Fildes, 2021]. Product-level forecasts concern predicting the demand for a large number of products over a short horizon. Predicting the demand for each product in each store is an integral part of making opera-

tional decisions in the organization, such as pricing [Cohen et al., 2020], promotions [Cohen et al., 2017,Cohen et al., 2021], space allocation, and inventory management [Ma and Fildes, 2021]. Due to the smaller amount of data, product-level forecasts are more challenging and most researchers have focused on presenting a single model that works best for all items in the store, which results in sub-optimal performance. For example, [Ma and Fildes, 2021] proposes a meta-learning framework that first learns from an ensemble of base-forecasting methods, and then uses this knowledge to generate an optimal ensemble model.

2.2.2 Techniques for Demand Prediction

Traditional techniques such as simple moving averages, exponential smoothing and its extensions, and ARIMA have been widely used to predict demand. Multiple linear regressions or more complex econometric models that have an explicit set of parameters have also been used. Studies have shown that using multivariate models often improves the prediction performance compared to univariate models [Ma and Fildes, 2021].

According to [Cohen et al., 2022], nonlinear models that can be useful for predicting demand include linear and nonlinear regressions, Regression Trees, Support Vector Machines [Ali et al., 2009], and Recurrent Neural Networks [Gasthaus et al., 2019]. Nonlinear models perform better in general.

It is not always possible to select traditional approaches or neural networks as the superior approach. The results of the most recent M4 competition suggest that (combinations of) statistical methods outperform pure machine learning methods, while a hybrid approach performed best at forecasting univariate time series. The M4 competition is an extension of the previous M competitions, which focus on time series forecasting [Makridakis et al., 2020].

[Huber and Stuckenschmidt, 2020] found that machine learning methods perform better when predicting the demand on special days and are also more suitable for the retail industry, which deals with large-scale demand forecasting scenarios.

[Alon et al., 2001] concluded that neural networks perform better than ARIMA models when forecasting sales with a strong trend and seasonality aggregated over a month. [Aburto and Weber, 2007] proposed a hybrid of ARIMA and neural networks in which the neural network is trained on the residuals of the ARIMA model.

2.3 Domain Adaptation

The presence of abundant data has led machine learning models to be very successful in many tasks, such as self-driving cars [Wang et al., 2012], machine translation [Young et al., 2018], automatic diagnosis [Kononenko, 2001], exo-planet discovery [Ball and Brunner, 2010], and online commerce [Lu et al., 2015]. These models have even surpassed human abilities in object detection tasks [He et al., 2015].

Despite this success, machine learning models will fail to generalize well whenever there is a significant difference between the distribution of the data they are trained on versus the data they are applied to later on. By the ability to generalize, we mean performing well on new distributions of data and the distribution that the model is trained on. In other words, if the source data (i.e., primary data that the model is trained on) is not a good enough reflection of the target data (i.e., the data that we are interested in for prediction), the system is not expected to perform well [Kouw and Loog, 2018].

Domain Adaptation techniques aim to mitigate this shift between source and target distributions and create a model that generalizes well to the target domain [Kouw and Loog, 2018]. We will discuss different approaches that domain adaptation techniques use in the following sections. The domain shift between source and target data can be caused by numerous factors. Below we provide some examples of possible cases.

A prominent example of the need for domain adaptation would be in bio-engineering. MRI images are used in hospitals to detect abnormalities. However, each of the scanning machines has specific calibration and configuration, and hence will produce different images [De Bruijne, 2016]. Another example is in the field of natural translation. The words used to describe a book are different from what is used to describe an electronic device, making sentiment analysis context reliant [Blitzer et al., 2007]. Similarly, self-driving cars have to adapt to different surroundings as well as to different people [Van Kasteren et al., 2010].

In summary, we can say that the goal of domain adaptation is to create a model that performs well on the target data by properly leveraging the information in the source data.

2.3.1 Domain Adaptation vs. Transfer Learning

In the context of transfer learning and domain adaptation, a domain consists of three parts: (i) an input space X (feature vectors), (ii) an output space Y (labels), and (iii) a joint probability distribution $P(X, Y)$. Machine learning models try to model $P(Y|X)$. Given two domains, if each of these three components differ, domains are different, so a domain shift exists between them. In transfer learning tasks, each of these three components is free to differ. However, domain adaptation is considered a special case of transfer learning in which only the probability distribution can change between the domains [Kouw and Loog, 2018]. The same concept can be seen in **Figure 2.8**.

Note that the joint probability distribution between the input and label spaces can be decomposed as follows: $P(X, Y) = P(Y|X)P(X)$, $P(X, Y) = P(X|Y)P(Y)$.

Given the above, we define the three following domain shifts:

1. **Covariate shift**: $P(X)$ changes, but $P(Y|X)$ remains constant (first decomposition above). In this context, a covariate is a variable that is not of direct interest, but its

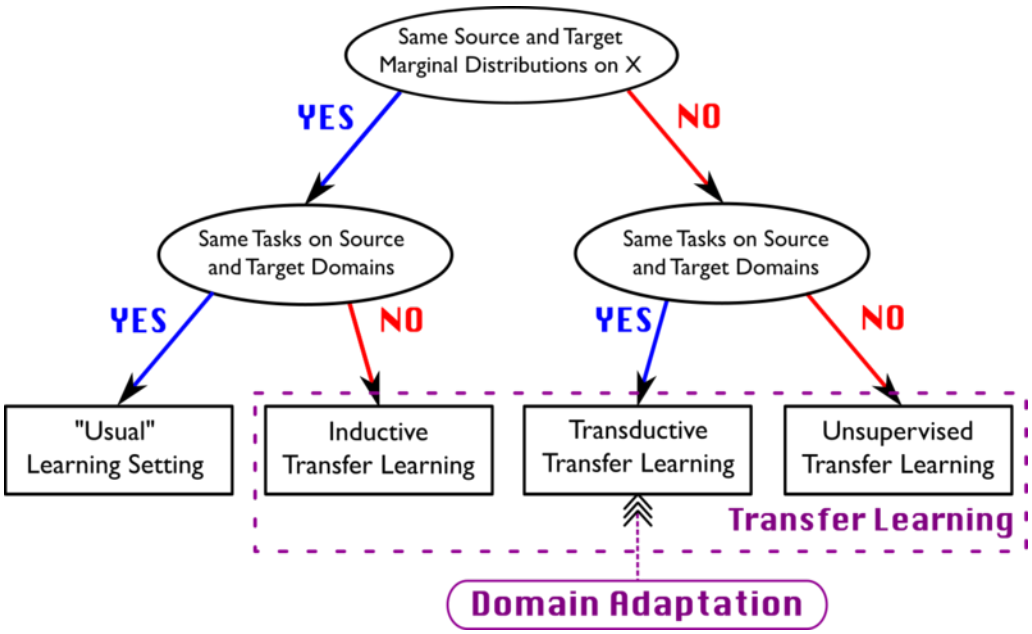


Figure 2.8: Graph showing the relationship between transfer learning and domain adaptation. Image credit: [Redko et al., 2020]

change affects the variable of interest. For example, in the case of detecting breast cancer, age is a covariate variable since it is not of direct interest but it affects the risk of the patient having breast cancer. Covariate shifts can happen because of biases during data collection. For example, in the case of breast cancer, one might collect data from clinics where women have come for checkups. Since women over 40 years old are encouraged by their doctors to do checkups, the frequency of women over 40 in the data may be higher than it should (i.e., be biased). Another common reason behind covariate shift is oversampling. Since it is challenging for machine learning models to learn from imbalanced data, we might over-sample rare test cases to help the model train. Another reason that covariate shifts can happen in real life stems from the difference between the environment in which the training data and test data are collected. For example, since training data are usually collected in laboratories, many environmental factors have been controlled in them. However, if the model is to be used by people in their homes, the test data may be noisy. An illustration of covariate shift can be seen in **Figure 2.9**.

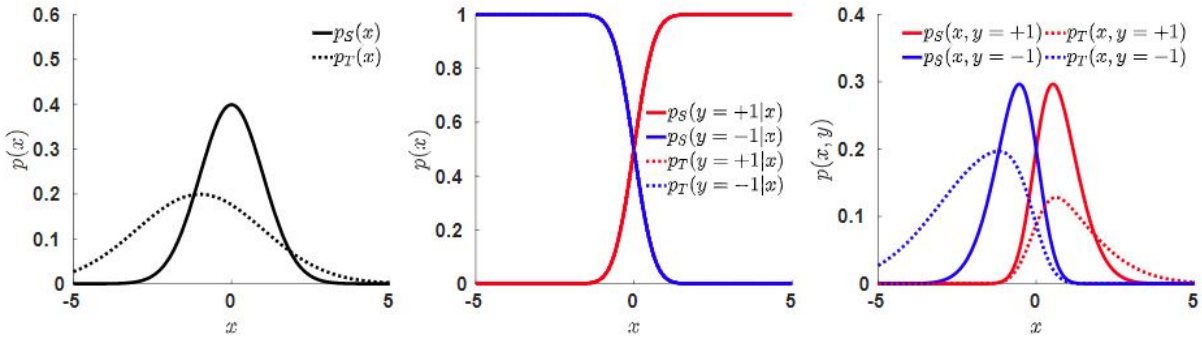


Figure 2.9: An example of covariate shift. (Left) The target data distribution has shifted with respect to the source data distribution. (Middle) The posterior distributions are not changed. (Right) The resulting joint distributions are also changed. Image credit: [Kouw and Loog, 2018]

2. **Label shift**: $P(Y)$ changes, but $P(X|Y)$ remains constant (second decomposition above). Note that when the input distribution changes, the output distribution also changes, so that both the covariate and label shifts happen simultaneously. Consider the breast cancer example from above. Since there are more people over the age of 40 in the clinic, the ratio of positive samples increases. However, given two patients with breast cancer, they have the same probability of being over 40. In other words, $P(X|Y)$ is the same, thus making this scenario also an example of label shift. This is not always the case, however. It is possible to have only label shift. For example, using a preventing drug will reduce the probability of having breast cancer regardless of age, so there is no covariate shift in this scenario. Since $P(X|Y)$ is still constant, we have a label shift. An illustration of label shift can be seen in **Figure 2.10**.

3. **Concept shift**: $P(Y|X)$ changes, but $P(X)$ remains constant (first decomposition above). An example of such a scenario would be an airplane ticket pricing system. Given the same flight (same route, airplane, seating), prices of tickets can change during different seasons or throughout the week based on demand. An illustration of concept shift can be seen in **Figure 2.11**.

In general, multiple types of domain shift can happen between two data distributions and this makes domain adaptation hard [Kouw and Loog, 2018].

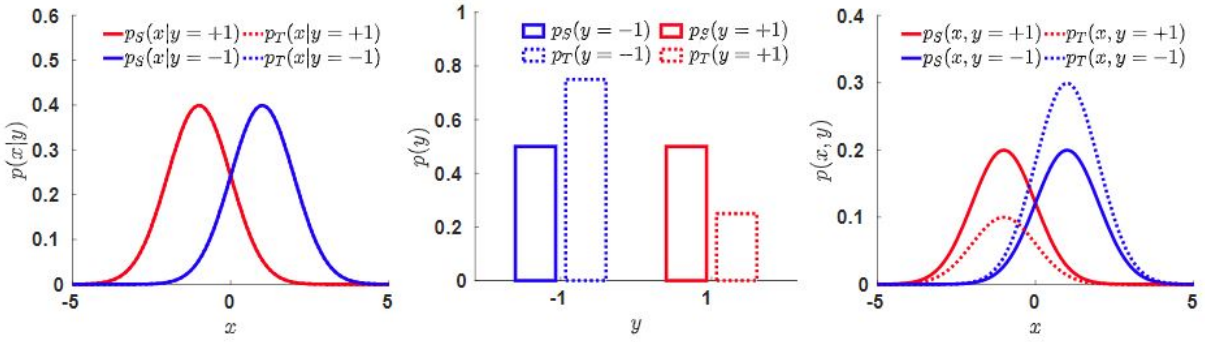


Figure 2.10: An example of prior shift. (Left) The conditional distributions in each domain are different. (Middle) The prior distribution changes from 1/2 for both classes in the source domain to 3/4 and 1/4 in the target domain. (Right) The joint distributions are also updated. Image credit: [Kouw and Loog, 2018]

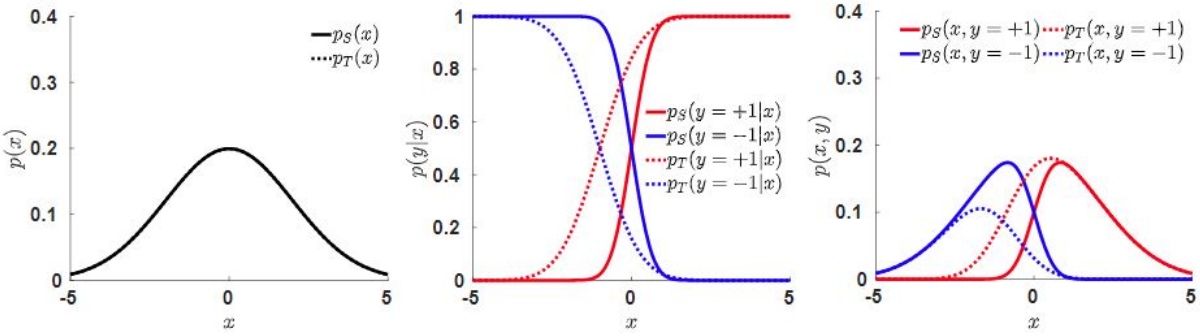


Figure 2.11: An example of concept shift. (Left) The data distributions remain constant. (Middle) The target posterior distributions are shifted to the left with regards to the source. (Right) The resulting joint distributions are changed. Image credit: [Kouw and Loog, 2018]

2.3.2 Methods

[O'donovan et al., 2015] categorized domain adaptation techniques into three main sections: Feature Based, Instance Based, and Parameter Based. Here we will elaborate on each category and discuss prominent methods for each.

Feature Based

This type of methods generally assume that the input space was shifted by uncertainties in the data collection process and try to counteract this shift by finding a feature space in which the source and target distributions match [de Mathelin et al., 2021]. Most of the

algorithms in this category are used for unsupervised domain adaptation. **Figure 2.12** shows an illustration of the inner workings of feature-based domain adaptation methods.

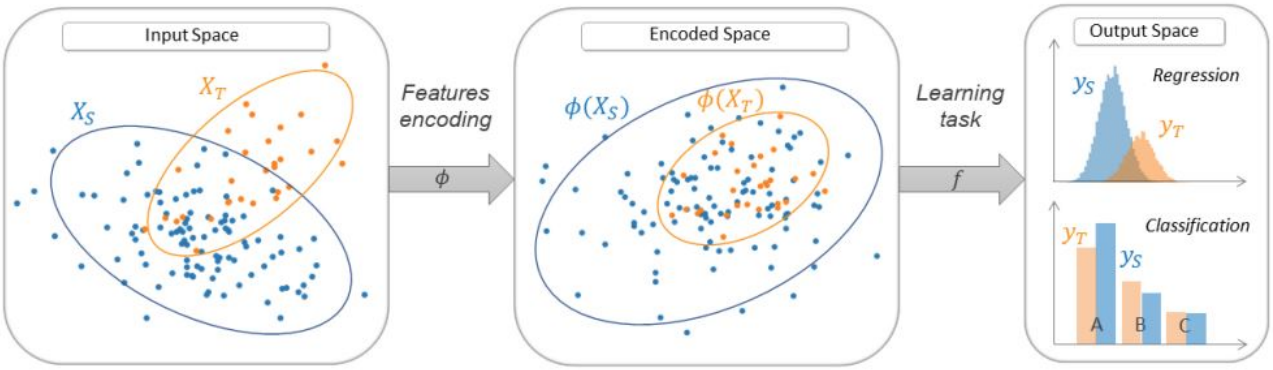


Figure 2.12: Illustration of how feature-based methods work. In these methods, the source and target features are mapped to a new space in which their distributions match. Image credit: [de Mathelin et al., 2021]

We next discuss different examples of feature-based domain adaptation methods:

1. **FE: Frustratingly Easy Domain Adaptation**

   [Daumé III, 2009] proposed FE as a supervised domain adaptation method, so it is useful when we have a few labelled target data to help us perform better than only using source data.

   As indicated by the name, although this method performs really well, it is surprisingly easy to implement. The authors have implemented FE in ten lines of Perl code [Daumé III, 2009]. It is also easy to implement in other programming languages.

   FE is considered as a preprocessing step since it is similar to data augmentation. The method tries to map the source and target samples to a new space in which it is clear to which domain features correspond.

   To give more details, each feature in the original data space is mapped to three versions of it: the general version, the source version, and the target version. The augmented source data will contain only the source and the general features. The augmented target data will contain only the general and target features. This augmentation process helps identify to which domains each feature belongs [Daumé III, 2009].

   Consider $X \in R^F$ as our input space and Y as our output space. $\phi^s(x)$ is defined as the transformation that maps the source data to the augmented space and $\phi^t(x)$ as the transformation that maps the target data to the augmented space. These transformations are as follows:

$$\phi^s(x) =< x, x, 0 >,$$

$$\phi^t(x) =< x, 0, x > .$$

In the above notation, 0 is $< 0, 0, ..., 0 > \in R^F$. The transformations augment the F dimensional input space to a new $3F$ dimensional space.

To explain how this transformation would help, we borrow an example from [Daumé III, 2009]. Suppose that the task of part of speech tagging (POS tagging) in the two different domains of wall street journals and electronics reviews. In the context of wall street journals, "monitor" would be considered as a verb, whereas in the context of electronics reviews, it would be considered as a noun. In both, "the" should be considered as a determiner. Now, we define $X \in R^2 < x_1.x_2 >$. $x1$ captures whether the word is "the" and x_2 captures whether it is "monitor." In the new space, $\hat{x_1}$ and $\hat{x_2}$ would be the general version of the features, $\hat{x_3}$ and $\hat{x_4}$ would be the source-specific features, and $\hat{x_5}$ and $\hat{x_6}$ would be the target specific features.

Given the above explanations, to show that a word is a determiner, the model would give it the weight of $< 1, 0, 0, 0, 0, 0 >$, and to show that a word is a noun, it would give it the weight $< 1, 0, 0, 0, 0, 1 >$. To assign a word to the verb category, the model would set the weight to $< 0, 0, 0, 1, 0, 0 >$.

It is now clear how by augmenting the F dimensional input to a new $3F$ space, the model can identify the space that each feature corresponds to.

[Daumé III, 2009] mentions that instead of augmenting from R^F to R^3F, the space can be mapped to R^2F. However, the above notation is more general and intuitive. Finally, the augmentation process above can potentially be extended to a kernelized version.

2. **CORAL: CORrelation ALignment**

   If the covariances of the source and target data are different (although having the same mean and variance), this presents a problem for transferring the model trained on the source to the target.

CORAL, proposed by [Sun et al., 2016], as an unsupervised domain adaptation method, tries to match the source and target distributions by aligning the second-order statistics (i.e., covariance) of the source with the target distribution. To do so, a transformation (denoted by A in the formulation below) is applied to the source features and tries to minimize the Frobenius norm of the distance between the source and target covariances.

Suppose that we denote the source features by S and the target features by T. Then, we can write the aforementioned steps as follows:

$$min_A||C_s - C_t||_F^2$$

$$= min_A||A^T C_s A - C_t||_F^2,$$

where C_t denotes the covariance of the target data and C_s denotes the covariance of the source data. The goal of the above optimization problem is to find a transformation matrix A such that the distance between the transformed source covariance matrix and the target covariance matrix is minimal. In other words, this will minimize the domain shift (i.e., the norm of the difference of the covariances) between the source and target distributions.

If Rank(C_s) is greater than Rank(C_t), then we can set A so that $C_t = C_t^*$. However, this is usually not the case. In the general format, it can be shown that we can align the source and target features by using the following algorithm:

$$I = eye(size(D_s, 2))$$

$$C_s = cov(D_s) + \lambda I$$

$$C_t = cov(D_t) + \lambda I$$

$$D_s = D_s * C_s^{\frac{-1}{2}}$$

$$D_s^* = D_s * C_t^{\frac{-1}{2}},$$

where D_s denotes the source data and D_t denotes the target data. The intuition behind this algorithm is to whiten the source data (i.e., removing the feature correlations of the source domain) and then recolour it by using the covariance of the target data. This process will make the covariance of the source and target data equal and the domain shift minimal. Note that whitening both the source and target distributions will not align them since the source and target data are likely to lie on different sub-spaces due to the domain shift. **Figure 2.13** explains this idea more clearly.

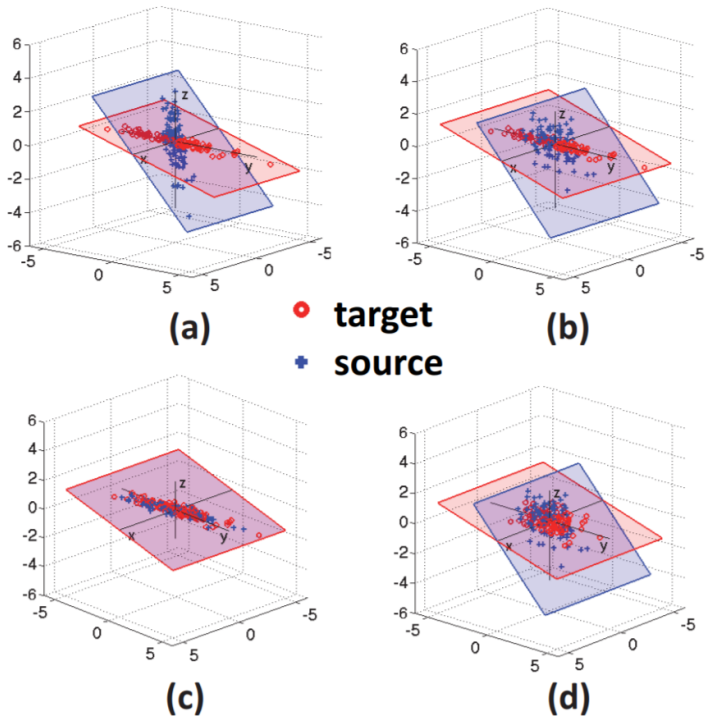


Figure 2.13: Illustration of the Correlation Alignment (CORAL) algorithm: (a) It is clear that the source and target data have different covariances although both being normalized (b) Source domain data is de-correlated (c) In the next step, the source data is recoloured using the covariance of the target data. This process will make the covariance of the source and target data equal and the domain shift minimal. The classifier trained on the adjusted source domain is expected to work well in the target domain. (d) Note that whitening both the source and target distributions won't align them since the source and target data are likely to lie on different sub-spaces due to the domain shift (best viewed in colour) [Sun and Saenko, 2016].

3. **DeepCORAL: Deep Correlation Alignment**

   DeepCORAL is an extension of the CORAL method. It learns a nonlinear transformation to align the correlations of activation layers in a deep neural network, whereas CORAL uses a linear transformation. So DeepCORAL is more generalizable.

   CORAL Loss

   Before discussing DeepCORAL, we first define the CORAL loss for a single feature layer of a deep network. Assume that S denotes the samples in the source data, T

denotes the samples in the target data, n_s refers to the number of data points in the source domain, and n_T refers to the number of target data points. We define the CORAL loss as the Frobenius distance between second order statistics of the source and target data:

$$L_{coral} = \frac{1}{4d^2}||C_s - C_T||_F^2,$$

where C_s and C_T are the covariances of the source and target features, respectively and can be calculated from the following expression:

$$C_s = \frac{1}{n_s - 1}(D_s^T D_s - \frac{1}{n_s}(1^T D_s)(1^T D_s)),$$

$$C_T = \frac{1}{n_s - 1}(D_T^T D_T - \frac{1}{n_T}(1^T D_T)(1^T D_T)).$$

The CORAL loss is differentiable and its gradients with respect to the input features are given by:

$$\frac{\partial L_{coral}}{\partial D_s^{ij}} = \frac{1}{d^2(n_s - 1)}((D_s^T - \frac{1}{n_s}(1^T D_s)^T 1^T)^T (C_s - C_T))^{ij},$$

$$\frac{\partial L_{coral}}{\partial D_T^{ij}} = \frac{1}{d^2(n_s - 1)}((D_T^T - \frac{1}{n_T}(1^T D_T)^T 1^T)^T (C_s - C_T))^{ij}.$$

Adding CORAL Loss to Deep Networks

We now consider a deep network used to perform a classification or a regression task. When there is a domain shift between the source and target distributions, we expect to see some over-fitting when only optimizing the loss function of our deep network. Consider a CORAL loss defined on the desired layers of the network. Note that if we only optimize for this loss, then the classification/regression will not be successful. So, intuitively, we should optimize for a weighted summation of the network's original loss and the CORAL loss (a summation over all the layers over which we defined CORAL loss). We have

$$L = L_{class} + \Sigma_{i=0}^{t} \lambda_i L_{coral},$$

where t denotes the number of layers that the CORAL loss was defined on. In other words, our goal is to optimize the trade-off between the network's original loss function and the newly defined CORAL loss, so we measure this trade-off with a regularization parameter λ. λ_0 corresponds to the case where we have no CORAL loss, and larger indices can lead to degenerated features. Our goal is to set λ so that at the end of the training, the original loss function of the deep network and the new CORAL loss reach an equilibrium (i.e., are equal).

It is worth mentioning that by DeepCORAL, we mean any deep network that contains a CORAL loss.

Implementation details:

In the original paper, a task of classification over images in the Office31 dataset was considered. For this purpose, the authors used the Alexnet [Krizhevsky et al., 2012b] architecture pre-trained on the ImageNet dataset [Deng et al., 2009] as their base classifier. They added the CORAL loss into the last fully connected layer of Alexnet (fc8).

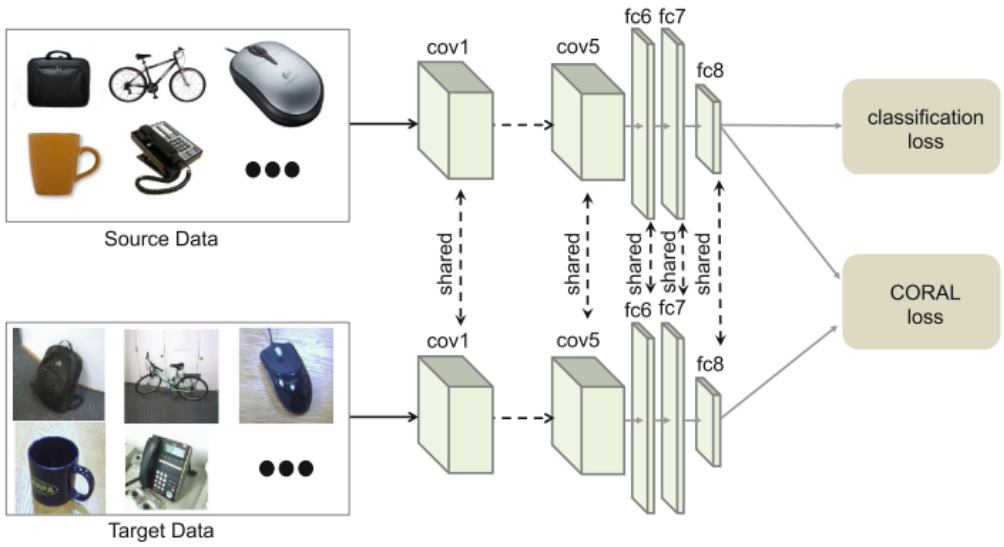


Figure 2.14: Deep CORAL architecture in the original paper based on Alexnet. For generalization and simplicity, here we apply the CORAL loss to the fc8. Integrating it into other layers or network architectures is also possible. Image credit: [Sun and Saenko, 2016].

Note that the CORAL loss can be added to any of the layers of Alexnet. However, the authors decided to add it only to the last fully connected layer to have a more general analysis, since most CNNs have a fully connected layer for classification at the end.

The architecture used in the original paper can be seen in **Figure 2.14**. As we can see, the Alexnet architecture has a classification loss on source samples originally, and now we add a new loss, CORAL loss, to the network.

As explained before, the CORAL loss is defined between the source and target samples and ensures that their second-order statistics align. We train the network by optimizing for a weighted summation of classification loss and CORAL loss (trade-off). Our goal is to train the network in a way that these two losses reach an equilibrium at the end of the training.

In the original paper, the dimension of the last fully connected layer was set to 31 (number of categories in the dataset). This layer was initialized by the weights

$N(0, 0.005)$, there was no initialization set for the layer bias in the original paper, so I set them to be constant at 0.001. The learning rate was set to 0.001 for all the layers in Alexnet except for the last fully connected layer. The learning rate for the last layer was set to 10 times 0.001 since it was training from scratch, whereas the rest of the network was using the pre-trained weights.

No specific value was set for λ, and it has to be set so that at the end of the training, the classification loss and the CORAL loss are the same, so the features are generative and discriminating enough.

In summary, DeepCORAL consists of both an encoder and a task network. The encoder network maps input features into a new space in which the task network is trained. The model parameters are optimized so that the summation of the original loss and the coral loss is minimized. A regularization parameter sets the trade-off between the two aforementioned losses. In other words, the encoder network learns a new feature representation in which the correlation matrices of the source and target data are aligned. Finally, the task network uses the source labelled data to learn the task [de Mathelin et al., 2021].

Instance Based

These methods aim to eliminate the difference between the source and target distributions by reweighing the source samples. These methods assume that the difference between the source and the target is due to a sample bias and $P(Y|X)$ remains constant (i.e., covariate shift) [de Mathelin et al., 2021].

Figure 2.15 shows an illustration of the inner workings of instance-based domain adaptation methods.

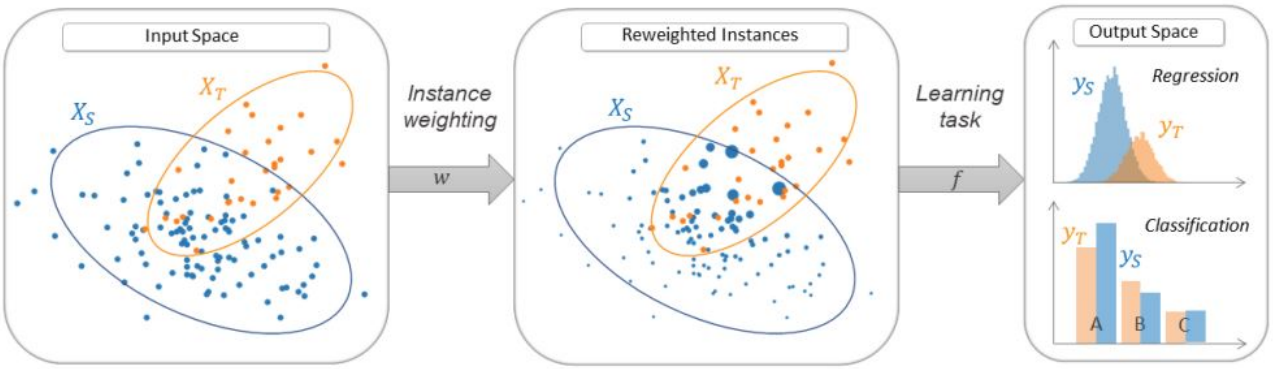


Figure 2.15: An illustration of how instance-based methods work. In these methods, the source samples are reweighted so that their distribution would match the distribution of the target samples better. Image credit: [de Mathelin et al., 2021]

We next discuss different examples of instance-based domain adaptation methods:

1. **KMM: Kernel Mean Matching**

   KMM minimizes the Maximum Mean Discrepancy (MMD) between the source and target domains to correct the sample bias.

   This algorithm aligns the source and target distributions by reweighing the source samples so that the difference between the means of the source and target data is minimized in a reproducing kernel Hilbert space.

This leads to solving the following quadratic optimization problem [Huang et al., 2006]:

$$min_w \frac{1}{2}(w^T K w - K^T W),$$

subject to:

$w_i \in [0, B]$ and $|\sum i = 1 n_s w_i - n_s| < m * \epsilon$,

Where:

$K_{ij} = k(x_i, x_j)$ with $x_i, x_j \in X_s$ and k a kernel

$k_i = \frac{n_s}{n_T} \sigma_{x_j \in K(x_i, x_j)}$ with $x_i \in X_s$

w_i are the weights assigned to source samples

X_s, X_T are source and target data respectively.

B, ϵ are hyper-parameters.

After finding the optimal value of w, the reweighed source samples are used to fit the model.

The KMM method was originally introduced for unsupervised domain adaptation but it could be extended to supervised domain adaptation by simply adding labelled target data to the training set [Huang et al., 2006].

2. **Transfer AdaBoost**

This supervised domain adaptation technique is based on a reverse boosting algorithm. At each iteration, the weights of the source samples that were predicted poorly decreases. The source and target data are denoted by $(X_s, Y_s), (X_T, Y_T)$ with weights of W_s and W_T respectively.

The algorithm is as follows [Dai et al., 2007]:

1. Normalize weights.

2. Fit an estimator f on the source and target data.

3. Calculate the following errors:

$$\epsilon_s = L_{01}(f(X_s), y_s),$$

$$\epsilon_s = L_{01}(f(X_T), y_T).$$

4. Calculate the total error:

$$E_{total} = \frac{1}{n_T} W_T^T \epsilon_T.$$

5. Update the weights:

$$w_s = w_s B_s^{\epsilon_s},$$

$$w_T = w_T B_T^{\epsilon_T},$$

where

$$B_s = \frac{1}{(1+\sqrt{2ln(n_s)/N})},$$

$$B_T = \frac{E_{total}}{1-E_{total}}.$$

6. Repeat the steps for N rounds.

Finally, the prediction is based on the weighted last $\frac{N}{2}$ estimators.

Parameter Based

Parameter-based models update the parameters of the model trained on the source data, so that it performs as well on the target data. In other words, the model is fine-tuned on the new data [de Mathelin et al., 2021]. These methods are mostly used in computer vision tasks. **Figure 2.12** shows an illustration of the inner workings of parameter-based domain adaptation methods.

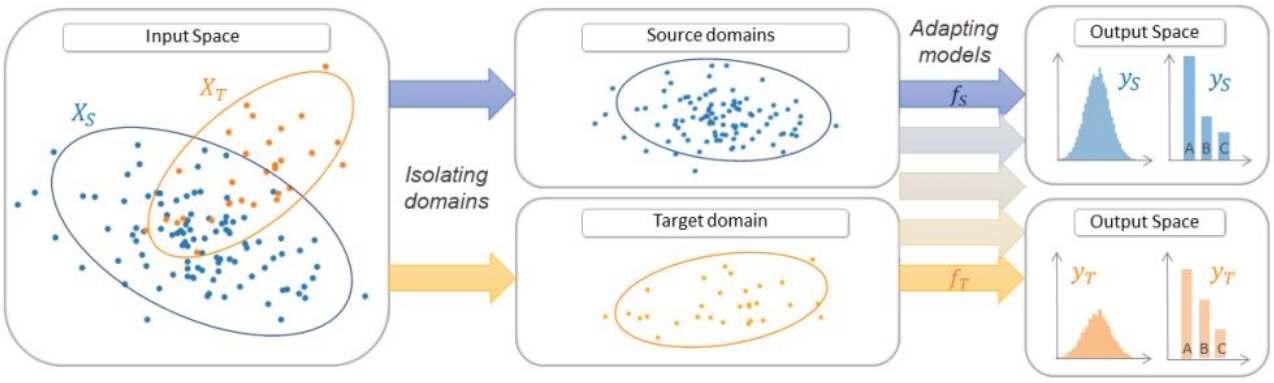


Figure 2.16: An illustration of how parameter-based methods work. In these methods, the model parameters are updated so that the model performs better on the target distribution. To do so, the model is fine-tuned using the target dataset. Image credit: [de Mathelin et al., 2021]

We next discuss different examples of parameter-based domain adaptation methods:

1. **Regular Transfer**

   In this method, the parameters of the pre-trained method are updated by using a few labelled target data.

   Our goal is to minimize both the previous loss function and the distance between the source and target parameters. This can be formulated as follows [Chelba et al., 2007]:

$$B_T = \underset{B_T \in R^p}{\arg\min}(Loss) + \lambda ||B_T - B_S||^2,$$

   where

   B_T are the target parameters.

   B_S are the source model parameters which are calculated as:

   $argmin_{B_T}||X_s B - y_s||^2$.

   (X_s, y_s) and (X_T, y_T) are the source and target labelled data, respectively.

   p is the number of target features.

λ is a trade-off parameter

2.3.3 Domain Adaptation in Time Series

As mentioned before, an important stream of domain adaptation techniques assumes an invariant feature space between the source and target domains [Daumé III, 2009,Sun et al., 2016,Sun and Saenko, 2016]. Since time series domain adaptation has gained significant interest in the past years, an easy extension of these methods to time series data would be to use RNN, or variational RNN-based feature extractor networks [da Costa et al., 2020,Che et al., 2018].

These methods assume that the conditional distribution of the output given the transformed features from previous time steps between the source and the target are equal. Unfortunately, time-series data may not always satisfy this assumption because the nature of the data is not static. However, we can assume that the causal structure of the source and target data is domain invariant [Cai et al., 2020]. According to [Cai et al., 2020], considering only the domain invariant associations and excluding domain-specific associations is key to avoid over-fitting.

[Cai et al., 2020] proposed a novel approach, Sparse Associative Structure Alignment (SASA), that distills the sparse associative structure and filters the domain-specific information. First, they use adaptive segment summarization. Then, they extract the sparse associative structure via attention mechanisms. And finally, they transfer the sparse associative structure from the source domain to the target. A summary of this method can be seen in **Figure 2.17**.

[Ragab et al., 2022] proposed SeLf-supervised AutoRegressive Domain Adaptation (SLARDA) as a new approach for time series specific domain adaptation. Their experiments confirm that their approach significantly improves the state-of-the-art for time series domain adaptation. A self-supervised (SL) learning module is used to improve the transferability of source features. Temporal dependencies of both source and target features are incorpo-

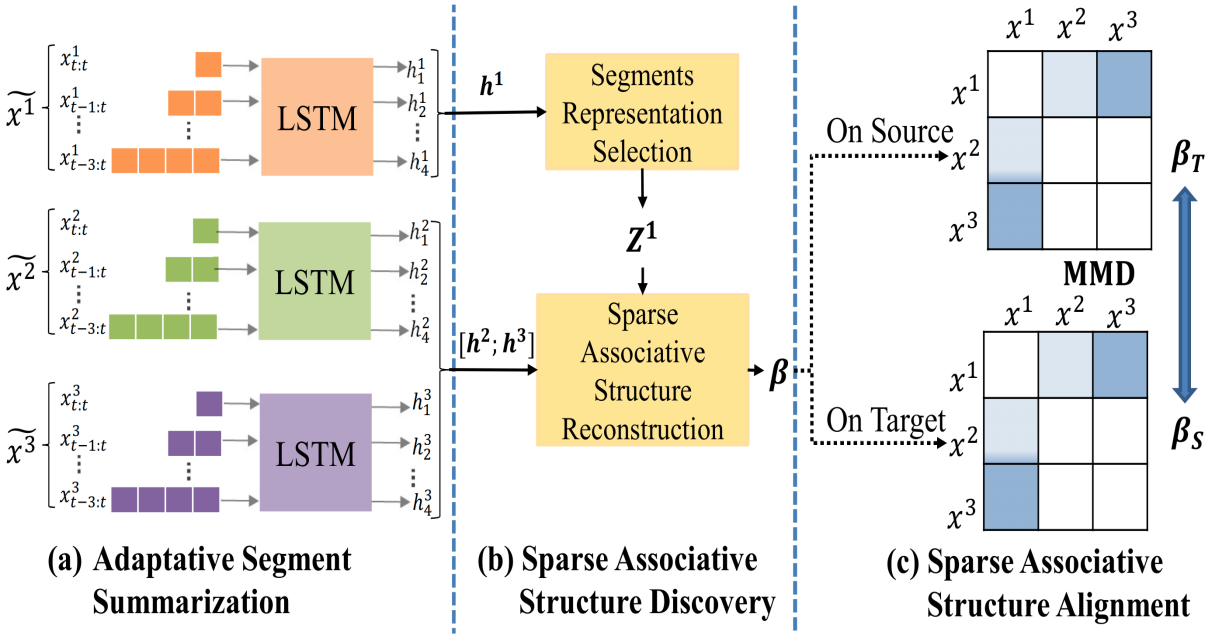


Figure 2.17: The sparse associative structure alignment model. (a) Adaptive segment summarization: Methods that use the entire time series as input may not capture when a variable affects others, this module allocates an independent LSTM to the different length segments constructed from each of the time series, the output of these LSTMs creates a new representation for the segments. (b) Sparse Associative structure structure discovery: Inter-variable and intra-variable attention mechanisms are used to extract the associative structure. (c) Sparse Associative Structure Alignment: In this step, the distance between the source and the target associative structure is minimized. [Cai et al., 2020] uses Maximum Mean Discrepancy metric (MMD) for this part. Image credit: [Cai et al., 2020]

rated in the model through a autoregressive domain adaptation technique. An ensemble teacher model aligns class-wise distribution in the target domain.

A novel Convolutional deep Domain Adaptation model for Time Series data (CoDATS) was proposed by [Wilson et al., 2020]. This approach, applied to real-world sensor data benchmarks, significantly improves accuracy and training time over state-of-the art DA strategies.

[Chang et al., 2020] presents a in-depth study of unsupervised domain adaptation (UDA) algorithms in the context of wearing diversity. They develop and evaluate three adaptation techniques on four HAR datasets. They perform an analysis to learn the downsides of each UDA algorithms.

As discussed in the previous section, some unsupervised domain adaptation methods try to minimize discrepancy distance. These methods, when applied to time series data are not robust since they contain only low-order and local statistics. [Liu and Xue, 2021] proposed an Adversarial Spectral Kernel Matching (AdvSKM) method. This method is able to precisely detect nonstationary and non-monotonic statistics in time series data resulting in precise discrepancy metric and better domain matching. Moreover, the adversarial kernel learning brings creates discriminatory expression for discrepancy matching.

2.3.4 Domain Adaptation on Transformer Architectures

Since the Transformers [Vaswani et al., 2017] have been deemed very successful in numerous tasks, such as natural processing and vision, they are expected to improve state-of-the-art methods in domain adaptation. Transformers are good at aligning distributions even from different tasks such as vision to vision, vision to text, and text to speech.

[Xu et al., 2021] proposed cross-domain transformers (CDtrans), comprising three weight sharing transformers, to solve the challenge of noisy pseudo-labels in the context of unsupervised domain adaptation.

CDtrans first uses a two-way centre aware labelling approach to create pseudo-labels for the unlabelled target samples.

To elaborate, the most similar target sample is paired with each source sample to create data pairs P_S to train the cross attention module. To eliminate the bias created by only using sections of the target data in creating the aforementioned training pair, the same process is repeated for the target samples. In other words, the closest sample of source data is matched with each of the target samples to create a new pairing set referred to as P_T. The union of P_S and P_T is used as the final training pairs to train the cross attention module [Xu et al., 2021].

The target samples are fed into a model pre-trained on source samples to generate the probability that the target sample belongs to each of the categories present in the source data. These probabilities are then used to create pseudo-labels for target samples by applying K-means clustering on the target features and their respective labels [Xu et al., 2021].

Additionally, for every pair, if the pseudo-label of the target sample is the same as the source sample label, this pair would be kept otherwise, it is discarded. **Figure 2.18** elaborates on the architecture of the CDtrans model.

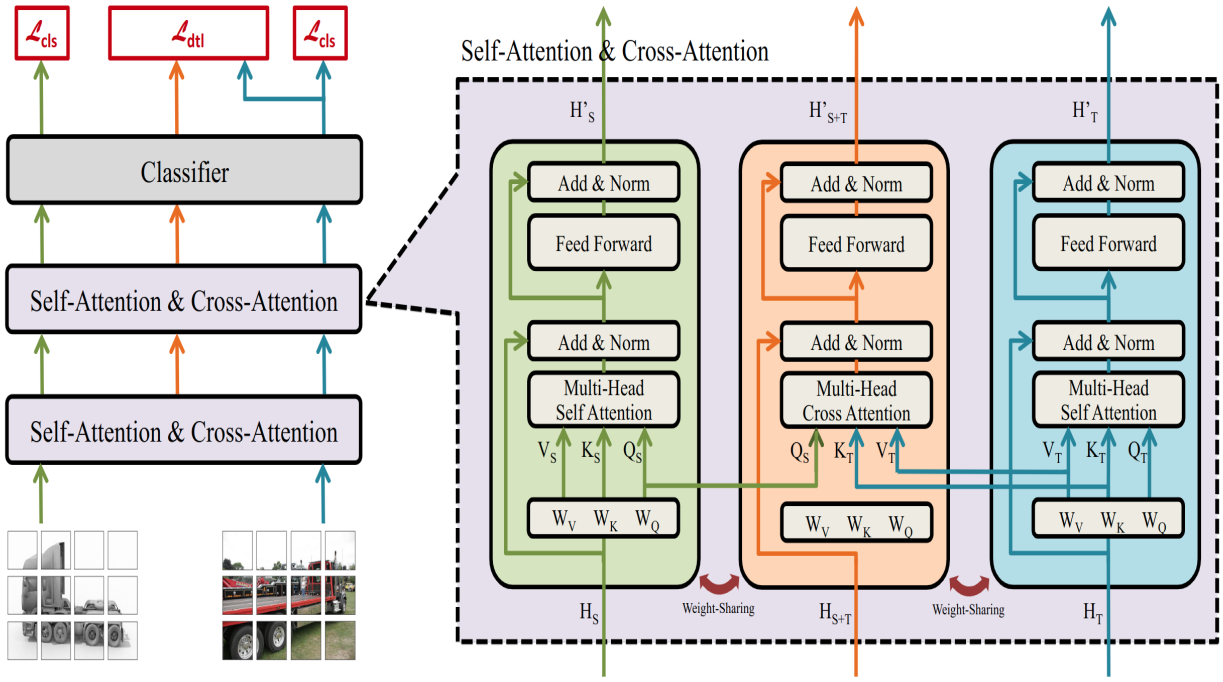


Figure 2.18: CDtrans framework. This architecture consists of two attention layers followed by a classifier layer. Each of the attention layers consists of three weight-sharing transformers: source (green), source-target (orange) and target (blue). After the set of training pairs are created (as explained above), each sample of the paired (source data, target data) is fed to the first attention layer. At this layer, the source sample is fed into the source transformer (green), the target sample is fed into the target transformer (blue), and the source-target transformer (orange), which is responsible to align the features of the source and target transformers does not have any external input. However, it gets its query from the source transformer and its key and value from the target transformer. The output of this layer is fed into the second attention layer. At the classifier layer, a classification loss is defined on the output of the source and target streams and a distillation loss is applied between the output of source-target and target stream. Image credit: [Xu et al., 2021]

The pseudo-label generation part of this architecture can be replaced by the actual labels in the case of supervised domain adaptation.

Chapter 3

Methodology

In this chapter we will first study the characteristics of our Dataset. Then we will describe each of our three case studies: outbreak of COVID-19 pandemic, opening a new store and introduction of a new product. In each case study, we apply Machine Learning techniques to predict product demand across domain shifts and try to mitigate the negative effect of such shifts using domain adaptation techniques.

3.1 Dataset

We use point of sale data from Alimentation Couche Tard convenient stores gathered between 2019-07 and 2021-02. This dataset contains data on 8,869 products sold in 89 locations in Montreal. We highlight that our data is gathered before, during, and after the outbreak of the COVID-19 pandemic.

In addition to quantity sold, we also have access to the promotion and price information. Although the direct information on product pricing was not directly accessible, we were able to calculate the price based on the dollar amount and quantity of each product sold. On days that a product had zero sales, we used the average price of that product over our dataset. These sales values can be seen for two specific products in Figure 3.1.

In this study, we focus our analysis on the **two categories of coffee and energy drinks**. These categories are consistently high selling products because they have a high average and a low variance in their sales values during the period of our study, so predicting their demand is of much more importance.

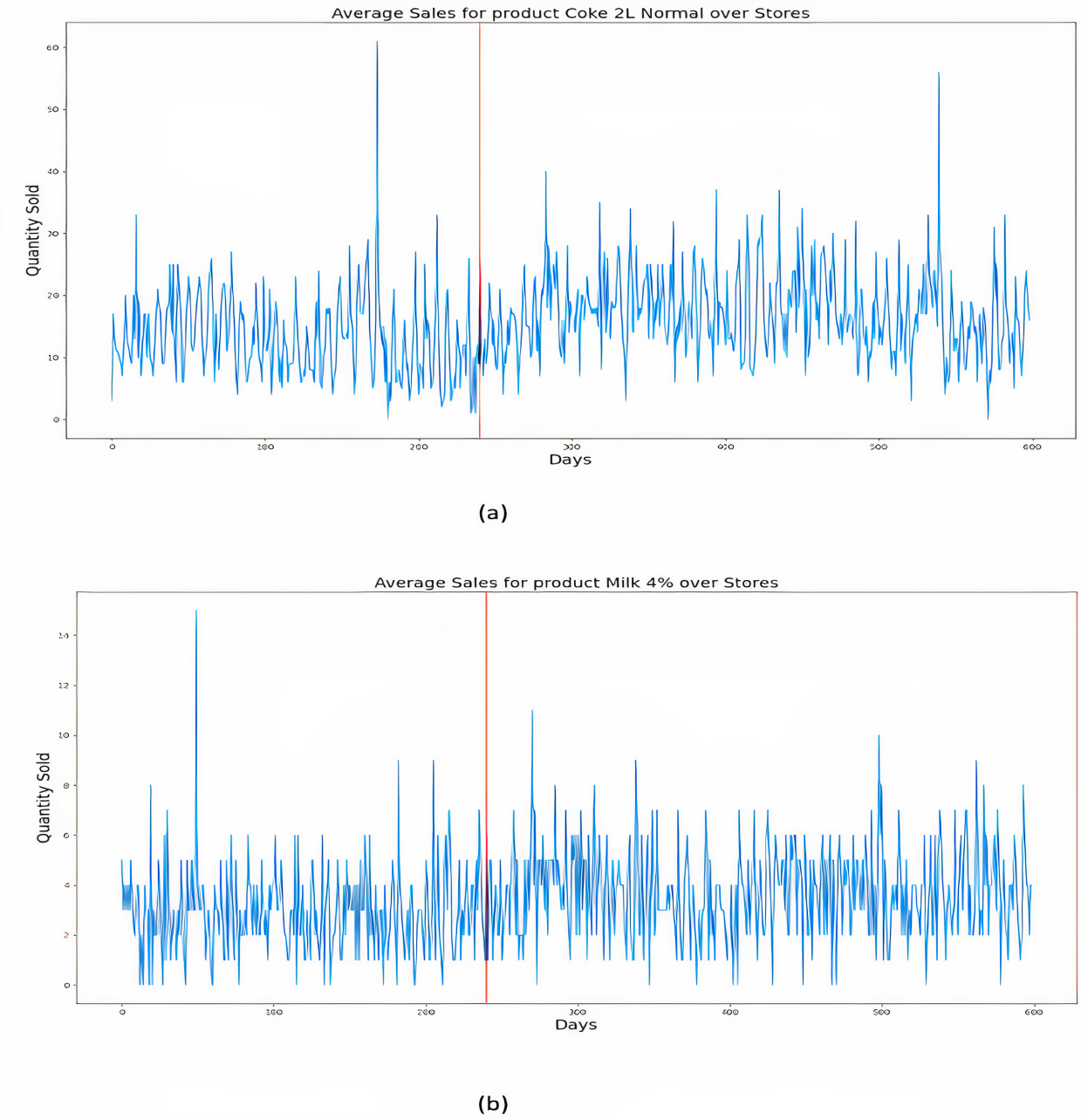


Figure 3.1: Sales quantities for two products: (a) Sales quantities for one type of Coke from 2019-07 to 2021-02. (b) Sales quantities for one product in the category of Milk from 2019-07 to 2021-02, the red vertical line indicates the start of the COVID-19 pandemic

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Coffee	238.59	189.945	120.0	208.0	320.0
Energy drinks	604.29	403.619979	372	532	764

Table 3.1: Comparison of the different characteristics of the sales data for product category coffee and energy drinks before the COVID-19 pandemic.

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Coffee	168.46	113.53	88.0	156.00	232.0
Energy drinks	618.136378	344.142	388	564	796

Table 3.2: Comparison of the different characteristics of the sales data for product category coffee and energy drinks after the COVID-19 pandemic.

There are 249 and 307 different products in product categories coffee and energy drinks, respectively. There are 3136 and 4654 different products in the parent categories of hot beverages and other beverages, respectively.

Tables below elaborate statistical characteristics of our data for categories coffee and energy drinks:

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Coffee	386	223.14	193	386	579
Energy drinks	81	21.9	33	51	73

Table 3.3: Comparison of the different characteristics of the sales data for product category coffee and energy drinks in the old stores averaged over all the products in the category.

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Coffee	135	78	67.5	135	202
Energy drinks	65.4	36.7	37	56	87

Table 3.4: Comparison of the different characteristics of the sales data for product category coffee and energy drinks in the new store (store 1208) averaged over all the products in the category.

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Energy drinks	355	310	142	200	306

Table 3.5: Comparison of the different characteristics of the sales data for product category coffee and energy drinks in the old products averaged over all stores.

Data characteristics					
Category	Sales-Mean	Sales - Std	Sales - Q1	Sales-Median	Sales - Q3
Energy drinks	245	158	50	272	358

Table 3.6: Comparison of the different characteristics of the sales data for product category coffee and energy drinks in the new product: GURU GUAYUSA 355ML or 81843 averaged over all stores.

3.2 Defining the Analysis

As mentioned before, the presence of abundant data has led machine learning models to be very successful in numerous tasks such as self-driving cars [Wang et al., 2012], machine-translation [Young et al., 2018], automatic diagnosis [Kononenko, 2001], exo-planet discovery [Ball and Brunner, 2010], and online commerce [Lu et al., 2015]. These models have even surpassed human abilities in object detection tasks [He et al., 2015].

Despite this success, machine learning models might fail to generalize well whenever there is a significant difference between the distribution of the data they are trained on versus the distribution of the data they are applied to. By the ability to generalize, we mean performing well on the new distributions of data. In other words, if the source data (i.e., primary data that the model is trained on) is not a well enough reflection of the target data (i.e., the data that we are interested in for prediction), then the system is not expected to perform well [Kouw and Loog, 2018].

This change in distribution from the source to the target can be due to a sudden change or a shift in the data domain. In the context of a convenient store, this shift can be due to a sudden economic change, health crises or introducing a new product or opening a new store.

In this study, we consider three examples of domain shifts in the context of convenient stores. These examples are either very common in retail or they are expected to have a significant impact on demand prediction. First, we define the analyses and their challenges. In the next chapter, we strive to use domain adaptation techniques to solve the challenges presented for each analysis.

3.2.1 Analysis-1: Outbreak of the COVID-19 Pandemic

The COVID-19 pandemic is still an ongoing (as of March 2022) global pandemic caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). This disease was first detected in Wuhan, China in December 2019, then speared to the rest of the word since it was not contained successfully [Ciotti et al., 2020].

As of March 10, 2022, the pandemic had caused more than 451 million cases and 6.02 million deaths, making it the fifth deadliest pandemic in history. The World health organization (WHO) has declared an emergency situation regarding the COVID-19 pandemic [Ciotti et al., 2020].

The outbreak of the COVID-19 pandemic had severe effects on the retail industry, causing stock-outs and shortages globally. Supply chain issues worsened by the cancellation of flights and new policies regarding freight transition.

Due to the uncertain situation worldwide, people started panic buying as a copying mechanism against the fear of uncertainty. This behaviour resulted in essentials such as food, toilet paper, and bottled water being stocked out at grocery stores [Ciotti et al., 2020, Adulyasak et al., 2020].

Supply shortages were initially due to disruptions in factory and logistic operations. They continued as managers underestimated the speed of economic recovery after the initial crash. The technology industry, in particular, has suffered from underestimating the semiconductor demand for vehicles and other products [Ciotti et al., 2020]. According to the

WHO, demand for personal protective equipment (PPE), such as masks, rose one hundredfold, pushing prices up twenty-fold [Ciotti et al., 2020].

In September 2021, the World Bank estimated that food prices would remain stable, but we witnessed a sharp increase in food prices especially in poorer countries, reaching the highest level since the pandemic began. The Agricultural Commodity Price Index stabilized in the third quarter but remained 17% higher relative to January 2021 [Ciotti et al., 2020]. At the beginning of the pandemic Petroleum was in surplus, since the demand for gasoline and other products collapsed due to reduced commuting and other trips [Ciotti et al., 2020].

In this study, our data spans from 2019-07 to 2021-02, so it covers the start of COVID-19 pandemic. Since the COVID-19 pandemic was a sudden change that modified many purchasing behaviours and supply chain logistics, we expect to see a change in the demand of products after the start of the pandemic. As a result, the data before the pandemic cannot be used to predict the demand of products after the start of the pandemic. In other words, the distribution of the sales data pre-pandemic and post-pandemic is different. We define pre-pandemic as the period before 2019-07 and post-pandemic as the period after 2021-02. Immediately after the pandemic, since we do not have a lot of data in the new domain of post-pandemic to train the demand prediction model, we expect the prediction accuracy of the model to suffer in comparison to pre-pandemic.

In this task, our goal is to predict the demand of each product for two categories (coffee and energy drinks) in each of the 89 stores in Montreal. We first calculate the accuracy of this demand prediction task before and after the COVID-19 pandemic. We next confirm that although the accuracy of pre-pandemic demand prediction is high, this accuracy is not satisfactory post-pandemic. In the next chapter, we use domain adaptation techniques to mitigate the negative effect of the COVID-19 pandemic on demand prediction.

In this analysis, we use the price and quantities sales information of each product in each store over a period of 30 days to predict its demand on the next day. We are using the

data from 2019-07 to 2020-03 as the pre-pandemic data. As discussed, we consider the data gathered after 2020-03 as post-pandemic data. Different end dates are considered for the post-pandemic period to examine the effect of increasing the amount of data in the new domain on the prediction accuracy. These different end dates result in using a range of 10 to 120 days of post-pandemic test data.

We define the pre-pandemic accuracy as the accuracy generated by the model trained and tested on pre-pandemic data. By post-pandemic accuracy, we refer to both the accuracy of the model that was trained on pre-pandemic data and tested on post-pandemic data and the accuracy from the model that was trained and tested on post-pandemic data.

We test two methods as the base demand prediction model in this analysis: XGBoost [Chen et al., 2015] and Transformers [Vaswani et al., 2017]. XGBoost, as described in the previous chapter, is a gradient boosted random forest, which has been proven superior in many forecasting tasks such as the M4 forecasting competition. Transformers, as described in the previous chapter, have been proven powerful in many tasks such as computer vision and natural language processing.

For the Transformer architecture, we used one sixteen dimensional attention head. We have set the embedding dimension to 59 or the length of time-series sequence set in this task (e.g., here the number of features is two referring to sales quantity and price information).

By conducting a manual grid search, the optimal maximum tree depth and the number of trees in XGBoost were found to be three and 100, respectively.

In this analysis, the R^2 metric, calculated between the ground truth and the predicted demand, is used to measure how accurate the models can predict the future demand. In statistics, the coefficient of determination, denoted R^2 or r^2 refers to the proportion of the variation in the dependent variable that is predictable from the independent variable(s).

Given $y_1, y_2, .., y_n$ as the real values and $f_1, f_2, ..., f_n$ as the predicted values, R^2 can be defined as below:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}},$$

Where:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i,$$

$$SS_{res} = \sum_i (y_i - f_i)^2,$$

$$SS_{tot} = \sum_i (y_i - \bar{y})^2.$$

R^2 can be negative (when evaluating on the test set) and is not a symmetric function. The best (highest) possible score is 1.0. A constant model that always predicts the expected value of y, regardless of the input features, would get a score of 0.0 [Nagelkerke et al., 1991].

In the next chapter, we show that as the amount of data post-pandemic grows, the accuracy of post-pandemic models also grows, thus decreasing the need for domain adaptation techniques. However, it might take months until the post-pandemic accuracy reaches the same level of performance as pre-pandemic models. We confirm that by using domain adaptation, the accuracy of post-pandemic models are substantially improved.

We test three different domain adaptation methods: Frustratingly Easy Domain adaptation (FE), Kernel Mean Matching (KMM), and an ensemble of the previous two methods. Furthermore, we test the pairing method, as explained before, in addition to each of the domain adaptation techniques so as to improve the post-pandemic accuracy.

3.2.2 Analysis-2: Opening of a New Store

Opening new (physical) stores is an essential part of expanding retail businesses. Fashion retailers frequently open new stores in different cities. Similarly, grocery stores are constantly trying to expand their reach in new cities and new neighbourhoods.

Predicting the demand for products in a new store is a critical task because it can help ensure the successful expansion of the business. Unfortunately, the data distribution in a new store is likely to be different from the rest of the stores since each store has unique features, such as its location, square footage, and average neighbourhood demographics. So, when attempting to predict the demand for products in a new store, the data from the other stores are not necessarily relevant. At the same time, there is not enough sales data available in the new store. Consequently, the prediction accuracy of traditional demand prediction models can often be very low.

In this task, we first calculate the demand prediction accuracy in the new store as well as in the rest of the stores (in each step, one of the "old" stores is held out, and the data from the rest of the old stores is used to predict the demand of the held-out store). Finally, these accuracy values are averaged out and ultimately represent the old-store model accuracy. We show that even though the accuracy of demand prediction in the old stores is high, this accuracy is not satisfactory for a newly-opened store without sufficient data. In the next chapter, we will use domain adaptation techniques to improve the prediction accuracy of new-store models.

As before, we use XGBoost and Transformer methods as the base demand prediction model. The parameters of the Transformer and XGBoost models are set in the same way as in the previous case study.

In this analysis, we use the price and quantities sales information in a each store over a 30-day period to predict the next-day demand aggregated at the product-category level. As

before, we perform this analysis for the two high-selling categories of coffee and energy drinks.

We first consider a simulated new store introduction. After we confirm the performance of our approach, we regenerate the results for a store that has been recently opened by our retail partner in Montreal. For the case of the simulated new store, we consider each of the stores on the island of Montreal as a newly-opened store with a hypothetical opening date (we substitute zeros for the sales before the hypothetical start date). The rest of the stores in Montreal would be considered as the old stores. We then define the old-store prediction accuracy as the accuracy generated by the model trained and tested on data from all previously-opened stores (historical data). By new-store accuracy, we refer to the accuracy of the following different models:

- A model trained using the data from previously-opened stores.
- A model trained using the data from the 50 most similar old stores (based on the Euclidean distance).
- A model trained using the data from the 25 most similar old stores (based on the Euclidean distance) and tested on the data from the new store.
- A model trained using the data from the 50 closest old stores (based on the geographical distance) and tested on the data from the new store.
- A model trained using the data from the 25 most similar old stores (based on the geographical distance) and tested on the data from the new store.
- A model that was both trained and tested using the data from the new store.

We repeat the above calculations for each of the 89 stores being considered as the new store. Finally, we report the average and confidence interval of these values.

For the case of an actual new store, the rest of the stores in Montreal would be naturally considered as the old stores. We define the old-store accuracy as the accuracy generated

by the model trained and tested using the data from all previously-opened stores (i.e., historical data). By new-store accuracy, we refer to both the accuracy of the model trained using the data from previously-opened stores and tested on the data from the new store, and the accuracy from the model trained and tested using the data from the new store. In this case, since we are only considering a single new store, we cannot report confidence intervals.

As before, we use the R^2 metric to measure how accurate the different models predict the future demand. We highlight that we are only using the data from 2019-07 to 2020-03 (i.e., before the COVID-19 pandemic) to isolate the effect of one domain shift. In other words, if we were to use the data from both before and after the pandemic, then two domain shifts would co-exist between the training and testing data: one shift due to the opening of the new store and the other due to the outbreak of the pandemic.

We show that as the amount of data available for the new store grows, the accuracy of new-store models also grows, thus reducing the need for domain adaptation techniques. However, it may take several months until the point where the new-store accuracy reaches an acceptable performance level. In the next chapter, we show how we can use domain adaptation techniques to enhance the accuracy of new-store models. Specifically, we consider FE, KMM, and an ensemble of these two methods. We also test the effect of using a pairing technique on top of these three domain adaptation approaches. The amount of data available for the new store ranges from 10 to 120 days in this analysis.

3.2.3 Analysis-3: Introducing a New Product

In the retail industry, new products are routinely introduced. For example, in the clothing industry, new designs are released every season. In fast-fashion, new items are introduced much more often (e.g., every couple of weeks). In this context, to design the supply chain planning, assortment planning, and stock allocation efficiently, an accurate demand forecast is necessary. While time-series forecasting algorithms discussed in previous chapters

can be used for existing products to forecast the sales, the same cannot be done for new products because they do not have any historical time-series data [Ekambaram et al., 2020].

Demand prediction for a new product at early stages is crucial in deciding whether or not to continue selling the product. However, the lack of market and consumer data during the early stages makes demand prediction incredibly difficult and unreliable, often underestimating or overestimating the product's demand [Afrin et al., 2018].

Due to the specific characteristics of each product, its sales distribution can often be unique. The same is true for new products. As a result, a domain shift exists between the data distribution of old products and the data distribution of the new product. When a new product is introduced, the data from previous products may not be useful to predict its demand. At the same, there is not enough data available for the new product, so it is not clear how we should predict the demand of the new product.

We calculate the demand prediction accuracy both before and after introducing a new product and we confirm that although the demand prediction method performs well for the old products, it would not be as well performing for the new product. In the next chapter, we use domain adaptation techniques to mitigate the negative effect of the domain shift caused by introducing a new product.

As in the previous two case studies, XGBoost and Transformer methods are considered as the base forecasting methods. In this analysis, we use the price and quantities sales information of products aggregated at the store level for a certain product over a period of 30 days to predict its demand for the next day. Once again, we perform the analysis for the two high-selling categories of coffee and energy drinks. We iteratively consider each of the products in these categories as a newly-introduced product. A hypothetical introduction date is set for it (we substitute zeros for the sales of the product before the hypothetical start date). The rest of the products in the same category is considered as the old products.

We define the old-product accuracy as the demand prediction accuracy generated by the model trained and tested using the data from all the previously-introduced products (historical data). By new-product accuracy, we refer to two scenarios. First, we consider the accuracy of the model trained using the data from previously-introduced products or from a subcategory of previously-introduced products (e.g., for the coffee category, we select "Black coffee" as a sub-category and for energy drinks we select "enriched water" as a sub-category) and tested on the data from the new product. Second, we consider the accuracy of the model that was both trained and tested using the data from the new product. We repeat these calculations for each of the products in the categories of coffee and energy drinks and report the average value.

We test the same three domain adaptation techniques in addition to the pairing technique to leverage the information from the old products. As before, we use the R^2 metric to measure the model prediction accuracy. Once again, we are only using the data gathered from 2019-07 to 2020-03 (i.e., before the COVID-19 pandemic) to isolate the effect of a single domain shift. In other words, if we were to use the data from before and after the pandemic, two domain shifts would co-exist between the training and testing data: one shift due to the change of product and the other due to the outbreak of the pandemic.

We show that as the amount of data available for the new product grows, the accuracy of new-product models also grows, thus reducing the need for domain adaptation techniques. However, it may take several months to reach the point where the new product accuracy reaches the same performance level as the predictive models for old products. In the next chapter, we show how we can use domain adaptation techniques to enhance the accuracy of new-product models. The amount of data available for the new product ranges from 10 to 120 days in this analysis.

Chapter 4

Experiments

4.1 Baseline Methods

We test different methods as the base forecasting model to select the best performing method based on the prediction accuracy on the test set. **Table 4.1** shows the accuracy comparison of the different baseline methods using distinct subsets of features from the dataset. The Sales, Price, Promotion, ID, Day, Month, and All columns refer to using only the sales quantity data, the sales quantity data plus the price information, the sales quantity data plus the promotion information, the sales quantity data plus the ID of the data collection day (from 1 to the number of data collection days), the sales quantity data plus the day of the month, the sales quantity data plus the month of the year, and all the previous features respectively.

As elaborated in **Table 4.1**, using both the quantity sales and the price information results in the best prediction accuracy, so we use sales data and price features in all the subsequent analyses. In addition, it is showed that the Transformers and XGBoost [Chen et al., 2015] outperform several commonly-used models including Linear Regression [Tibshirani, 1996], DecisionTreeRegressor [Steinberg and Colla, 2009], RandomForestRegres-

Baseline Accuracy - R^2 metric							
Baseline method	Sales	+Price	+Promotion	+ID	+Day	+Month	All
LinearRegression	0.87	**0.87**	0.87	0.87	0.87	0.87	0.72
DecisionTree	0.84	**0.85**	0.85	0.83	0.84	0.83	0.84
RandomForest	0.85	**0.86**	0.87	0.84	0.87	0.85	0.85
XGBRegressor	0.84	**0.89**	0.89	0.86	0.88	0.86	0.84
s lstm	0.7	**0.83**	0.67	0.86	0.15	0.5	0
Transformer	0.85	**0.874**	0.87	0.83	0.75	0.85	0.82

Table 4.1: A comparison of the different baseline methods with different subsets of features. Sales, Price, Promotion, ID, Day, and Month columns refer to using sales data solely, sales data plus price information, sales data plus promotion information, sales data plus ID of the day the data was collected on (from 1 to number of data collection days), and sales data plus day of the month and sales data plus month of the year, respectively.

sor [Breiman, 2001], and lstm [Hochreiter and Schmidhuber, 1997]. Hence, the Transformers [Vaswani et al., 2017] and XGBoost are considered as the baseline forecasting methods.

The parameters of each method were tuned to ensure the best possible performance.

It is shown in the above table that the LSTM method performs very poorly when the date and month information are added. This may be because LSTMs do not use attention mechanisms to focus on the more important parts of data so when adding the new information they become misguided.

4.2 Analysis-1: Outbreak of the COVID-19 Pandemic

In this analysis, we first present the results of using XGBoost as the base forecasting method. We confirm that using domain adaptation and the pairing technique discussed in Section 2.3.4 improves the accuracy of the post-pandemic demand prediction model. We then repeat the same analysis with Transformers as the base forecasting method and draw the same conclusion. These analyses were performed for both the coffee and energy drinks categories.

4.2.1 XGBoost

In this section, we compare the accuracy of different demand prediction scenarios with XGBoost as the base forecasting method. The pre-COVID-19 model (blue line in the graphs below) represents the accuracy of a XGBoost model trained and tested using the data prior to the pandemic. In other words, this line corresponds to the baseline accuracy before the pandemic. As shown in the graphs below, it is expected that the accuracy of the post-pandemic models (orange and green lines in graphs below) are lower than this baseline. Post-pandemic models refer to both the model trained and tested using the data post pandemic (green line in graphs below) and the model trained using the data prior to the pandemic and tested on the data post pandemic (orange line in graphs below).

Right after the start of the pandemic, the accuracy of the post-pandemic model (green line) is much lower than the pre-pandemic baseline (blue line). However, this trend increases over time as more data become available in the new domain. Our goal is to decrease the time needed to wait before the accuracy of the post-pandemic model (green line) reaches an acceptable performance level. In other words, we want to improve upon the post-pandemic models (green and orange lines) and increase their prediction accuracy to be as close as possible to the pre-pandemic accuracy (blue line). To achieve this, we test three domain adaptation approaches approaches: FE, KMM, and an ensemble of the previous two. Furthermore, we test the pairing method, as explained before, in addition to each of these approaches to improve the post-pandemic accuracy.

In each graph, we display three levels of granularity as sub-graphs to make the observations clearer. The first granularity level, shown in subplots (a), corresponds to the original scale. The second, shown in subplots (b), is the original graph while imposing limits on the axes. The third granularity level, shown in subplots (c), corresponds to the best performing domain adaptation technique. The graph's axes are also limited in these subplots.

Figure 6.1 and **Figure 6.2** exhibit a comparison of the different scenarios for the coffee category. **Figure 6.2** corresponds to the case when the pairing technique is used in addition to the domain adaptation approaches. As elaborated in **Figure 6.1b** and **Figure 6.2b**, the accuracy of the post pandemic model improves as more data become available in the new domain (i.e., the post-pandemic period). As shown in **Figure 6.2b**, the ensemble method that uses the pairing technique allows us to improve the accuracy of the post-pandemic models most effectively. In **Figure 6.1b**, the ensemble method slightly supersedes the accuracy of the post-pandemic models. However, in **Figure 6.2b**, all three domain adaptation methods perform better than the post-pandemic models (and by a larger margin). Specifically, we observe a 6% improvement for the 180-day mark.

We next repeat the same experiments for the energy drinks category.

Figure 6.3 and **Figure 6.4** show the comparison of the different scenarios for the energy drinks category. As elaborated in **Figure 6.3b** and **Figure 6.4b**, the accuracy of the post-pandemic model, improves as more data become available in the new domain (i.e., post-pandemic).

Figure 6.3 and **Figure 6.4** confirm that the accuracy of the post-pandemic models do not reach the pre-pandemic baseline. Fortunately, by using the ensemble method, we can improve the accuracy of the post-pandemic models, hence reducing the gap between this accuracy and the accuracy of the pre-pandemic baseline, especially after 90 days. We speculate that the drop in accuracy, seen in **Figure 6.3** and **Figure 6.4**, can be due to continuing domain shifts as the days go on after the outbreak of COVID-19 pandemic. In the months following the outbreak, different and new domain shifts occurred.

In **Figure 6.4b**, the ensemble method reaches the same level of pre-pandemic accuracy. As elaborated in **Figure 6.4b**, the ensemble method that uses the pairing technique improves the performance of the post-pandemic models most effectively. In **Figure 6.3b**, the ensemble method supersedes the accuracy of the post-pandemic models. However, in

		10	20	30	60	90	120	150	180
Coffee	pre-covid / post-covid	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
	pre-covid / post-covid	0.84	0.83	0.85	0.8	0.85	0.855	0.795	0.8
	post-covid / post-covid	0.425	0.5	0.755	0.76	0.81	0.83	0.74	0.8
	FE	0.65	0.73	0.82	0.8	0.85	0.85	0.85	0.83
	KMM	0.87	0.85	0.84	0.84	0.84	0.89	0.85	0.86
	Ensemble	0.42	0.5	0.75	0.75	0.81	0.84	0.74	0.8
	FE w Pairing	0.65	0.72	0.82	0.8	0.84	0.87	0.85	0.83
	KMM w Pairing	0.87	0.85	0.845	0.85	0.845	0.865	0.85	0.85
	Ensemble w Pairing	0.85	0.81	0.845	0.845	0.85	0.89	0.85	0.86
Energy Drinks	pre-covid / post-covid	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
	pre-covid / post-covid	0.765	0.77	0.76	0.755	0.77	0.73	0.71	0.7
	post-covid / post-covid	0.35	0.68	0.735	0.725	0.74	0.725	0.71	0.71
	FE	0.47	0.73	0.74	0.76	0.75	0.74	0.735	0.73
	KMM	0.74	0.75	0.74	0.755	0.75	0.74	0.735	0.73
	Ensemble	0.68	0.765	0.77	0.74	0.73	0.76	0.75	0.73
	FE w Pairing	0.7	0.76	0.77	0.75	0.8	0.76	0.65	0.755
	KMM w Pairing	0.79	0.775	0.745	0.755	0.8	0.76	0.7	0.75
	Ensemble w Pairing	0.79	0.78	0.79	0.75	0.8	0.77	0.75	0.77

Table 4.2: Comparison of results for product Coffee Energy Drinks over period of time when using XGboost as the forecasting method, e.g. pre-covid / post-covid refers to train on pre-covid and test on post-covid. Method in the bold represents the best performing domain adaptation technique.

Figure 6.4b, all three domain adaptation methods perform better than the post-pandemic models and by a larger margin (7% when considering 180 days).

These results are summarized in **Table 4.2**.

4.2.2 Transformers

In this section, we compare the accuracy of different scenarios when using Transformers as the base forecasting method.

Figure 6.5 and **Figure 6.6** show the same trends for the pre-pandemic and post-pandemic prediction accuracy as the results from the previous section. In each graph, we display two levels of granularity as sub-graphs to make the observations clearer. The first granularity level, shown in subplots (a), corresponds to the original scale. The second, shown in subplots (b), is the original graph while imposing a limit on the axes.

Figure 6.5 and **Figure 6.6** elaborate on the comparison of the different scenarios for the coffee category. As shown in **Figure 6.6b**, the ensemble method that uses the pairing technique improves the results of the post-pandemic models most effectively. In **Figure 6.5b**, the ensemble and the FE methods supersede the post-pandemic models. However, in **Figure 6.6b**, all three domain adaptation techniques perform better than the post-pandemic models (and by a larger margin).

Although domain adaptation still helps improve the accuracy of post-pandemic models, the absolute value of demand prediction is lower than when we use XGBoost as the baseline forecasting method. We speculate that this decrease in performance is due to the nature of our dataset. Transformers are very complicated models, and they need abundant data to learn from. However, the amount and complexity of our data are small in comparison.

Figure 6.5 and **Figure 6.6** show that as time elapses after the start of the pandemic and more data become available post pandemic, the accuracy of the (green) post-pandemic model improves; and around 120 days, it reaches the pre-pandemic accuracy. As elaborated in **Figure 6.5b** and **Figure 6.6b**, by using domain adaptation, we can reach the pre-pandemic accuracy after 30 days. This improvement is significant in reducing the time needed to wait before the point where the post-pandemic models reach an accept-

able accuracy. This reduction can help decrease the cost incurred to the retailer by not having an accurate demand prediction model after the domain shift.

We next repeat the same experiments for the energy drinks category.

Figure 6.7 and **Figure 6.8** show the comparison of the different scenarios for the energy drinks category. As shown in **Figure 6.8b**, the ensemble method that uses the pairing technique yields the most effective improvement. As shown in the graphs, the prediction accuracy of the post-pandemic models does not reach the pre-pandemic baseline. However, by using domain adaptation techniques, the prediction accuracy improves, especially as time elapses after the outbreak of the COVID-19 pandemic.

Using Transformers as the base forecasting method yields a lower prediction accuracy relative to using XGBoost as the base forecasting method (as in the previous section). Although the ensemble method combined with the pairing technique allows us to improve the accuracy of the post-pandemic models, the absolute value of the accuracy remains lower, compared to using XGBoost (compare **Figure 6.2b** and **Figure 6.6b**). Thus, in this case study based on our data, we conclude that it is better to use XGBoost as the base forecasting method.

These results are summarized in **Table 4.3**.

4.3 Analysis-2: Opening a New Store

In this subsection, our goal is to predict the demand of products in a newly opened store. We first calculate the demand prediction accuracy in the new store and in the rest of the stores (referred to as "old" stores). Specifically, we iterate over all stores, so that in each iteration, one of the old stores is held out, and the data from the rest of the old stores are used to predict the demand of the held-out store. The resulting accuracy values are averaged out and represent the old-store accuracy. We confirm that although the accuracy of the demand prediction in the old stores is high, the accuracy for a newly opened store

		10	20	30	60	90	120	150
Coffee	pre-covid / post-covid	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	pre-covid / post-covid	0.22	0.52	0.45	0.54	0.55	0.63	0.5
	post-covid / post-covid	0.425	0.5	0.755	0.76	0.81	0.83	0.74
	FE	0.71	0.6	0.76	0.74	0.8	0.69	0.73
	KMM	0.66	0.7	0.64	0.65	0.64	0.67	0.66
	Ensemble	0.74	0.7	0.75	0.74	0.75	0.7	0.73
	FE w Pairing	0.76	0.7	0.53	0.805	0.81	0.84	0.75
	KMM w Pairing	0.65	0.7	0.6	0.81	0.82	0.84	0.75
	Ensemble w Pairing	0.45	0.65	0.65	0.81	0.82	0.84	0.75
Energy Drinks	pre-covid / post-covid	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	pre-covid / post-covid	0.72	0.61	0.6	0.64	0.52	0.47	0.46
	post-covid / post-covid	0.18	0.25	0.3	0.22	0.05	0.35	0.44
	FE	0.63	0.42	0.41	0.5	0.54	0.33	0.39
	KMM	−0.2	0.33	0.47	0	0.07	0.55	0.45
	Ensemble	0.45	0.47	0.52	0.36	0.48	0.55	0.56
	FE w Pairing	0.65	−1	0.45	0.6	0.65	0.65	0.6
	KMM w Pairing	0.75	0	0.6	0.6	0.65	0.75	0.6
	Ensemble w Pairing	0.7	0.75	0.65	0.54	0.52	0.5	0.6

Table 4.3: Comparison of results for product Coffee Energy Drinks over period of time, when Transformer as the forecasting method, e.g. pre-covid / post-covid refers to train on pre-covid and test on post-covid. Method in the bold represents the best performing domain adaptation technique.

without sufficient data is not satisfactory. Domain adaptation techniques can thus be used to improve the demand prediction accuracy in the new store.

In this analysis, we use the price and sale quantity of the products aggregated at the category level in a store over 30 days to predict its demand on the next day. Again, we test both XGBoost [Chen et al., 2015] and Transformers [Vaswani et al., 2017] as our base forecasting methods.

First, we consider simulated new store introduction. After our heuristics are validated, we regenerate the results for an actual new store that was recently opened in Montreal.

4.3.1 Domain Adaptation for Simulated New Stores

For simulated new stores, we consider each of the stores in our dataset as a potential newly opened store with a hypothetical introduction date (we substitute zeros for the sales before the hypothetical start date in that store). The rest of the stores in our dataset would be considered the old stores.

XGBoost

We expect that as we accumulate more days pass from the new store opening (i.e., the data in the new domain becomes more prevalent), the prediction accuracy of the new-store model improves. For the coffee category, this intuition is readily confirmed in **Figure 4.1**. As shows in **Figure 4.1**, the prediction accuracy of the new-store model (brown line) increases as a function of the number of days pass after the new store opening.

The blue, orange, green, and red lines correspond to the models that have been trained using the data from all the old stores, the 50 most similar old stores (based on the Euclidean distance between retail time series data), the 25 most similar old stores (based on the Euclidean distance), the 50 closest old stores (based on the geographical distance), the 25 closest old stores (based on the geographical distance), and tested using the data from the new store, respectively. The pink line shows the model accuracy using domain adaptation (out of the six domain adaptation methods, only the one with the highest accuracy is reported, which in this case is the ensemble method).

As shows in **Figure 4.1**, domain adaptation improves the prediction accuracy of the new-store models (the blue, orange, green, red, purple, and brown lines in the graph). We note that the accuracy of the best performing domain adaptation method even supersedes the old-store baseline.

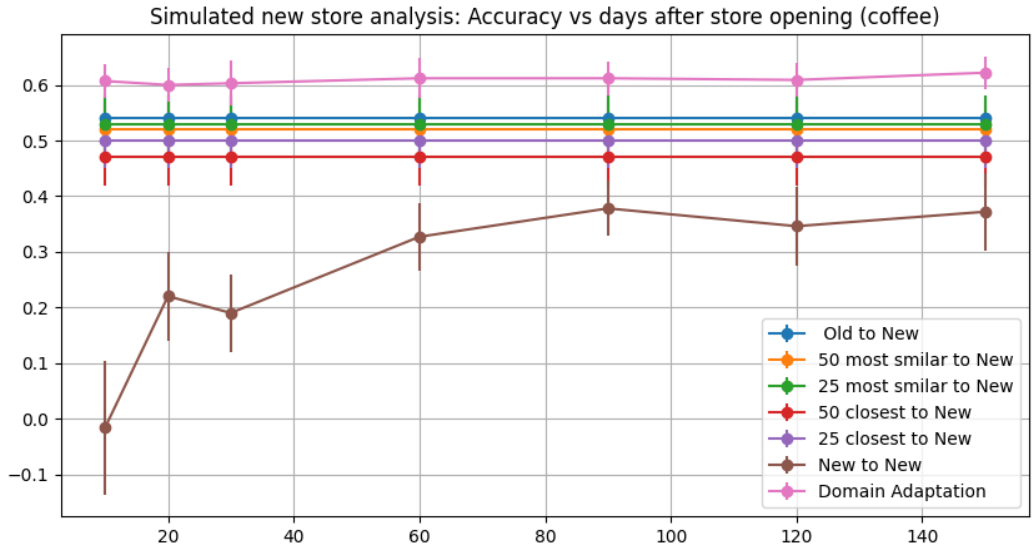
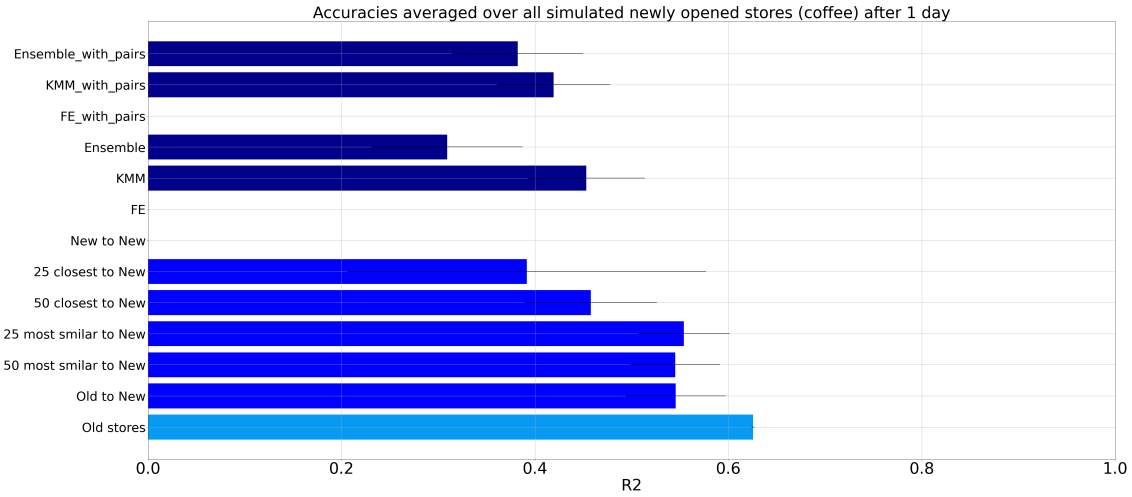


Figure 4.1: Comparison between the accuracy of different models for the coffee category as a function of the number of days from the opening of a new store (when the base forecasting method is XGBoost).

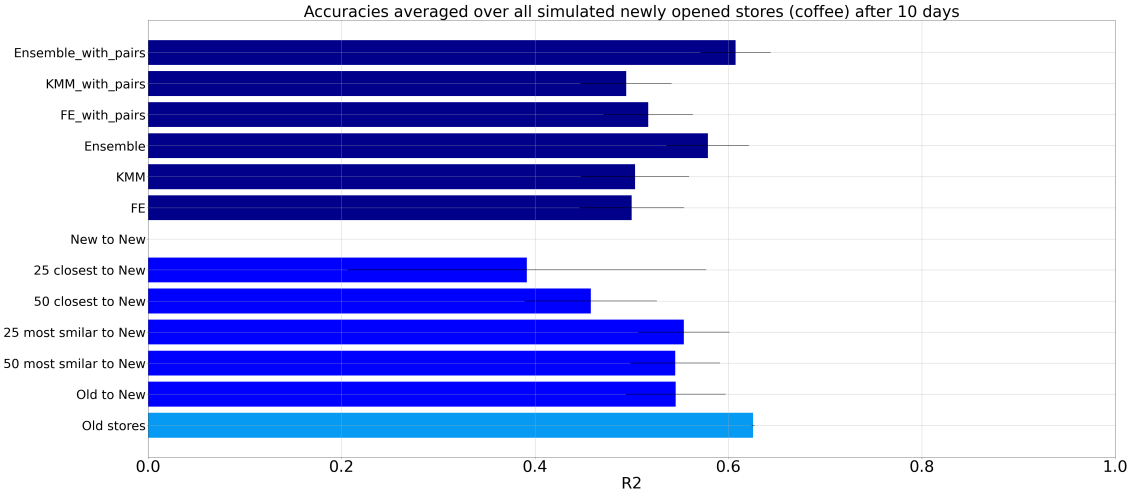
Figure 4.2 shows the accuracy levels of the different models for the coffee category. The bottom-most bar (shown in the lightest shade of blue) represents the accuracy of the model trained and tested using the data from the old stores. The middle bars show the accuracy of the models trained using the data from all the old stores, the 50 most similar old stores, the 25 most similar old stores, the 50 closest old stores, the 25 closest old stores, the data from the new store and tested using the new store from bottom to top.

The bars in the darkest shade of blue represent the domain adaptation methods. They correspond to FE, KMM, the ensemble of FE and KMM, the FE with the pairing method, the KMM with the pairing method, and the ensemble method with the pairing method from bottom to top. The bars having negative or a zero R^2, are not reported in the figures.

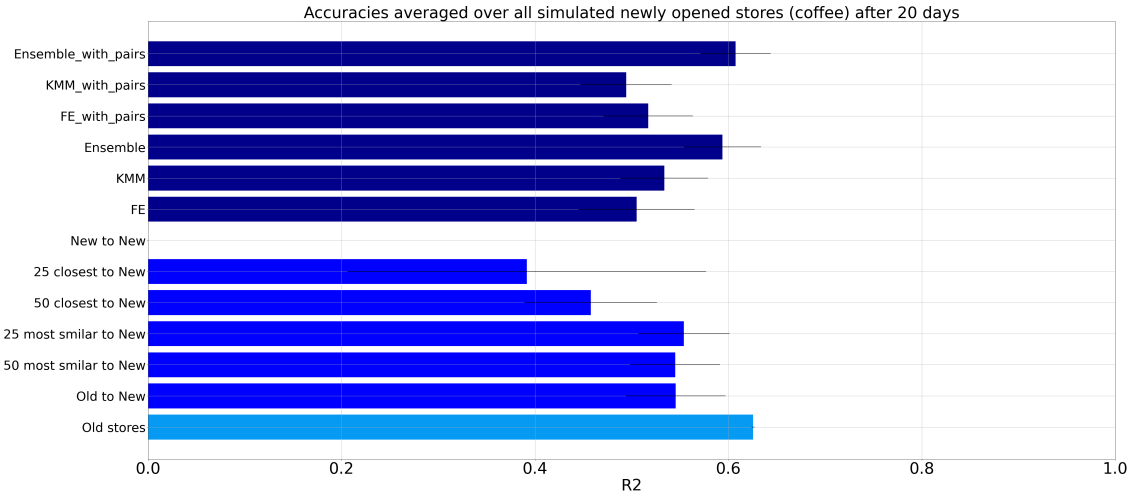
As more time elapses after the opening of the (simulated) new store, the p-value of the increase in prediction accuracy between the domain adaptation and the new-store model decreases. In other words, as more days pass from the new-store introduction, the improvement in prediction accuracy becomes more statistically significant.


Accuracies averaged over all simulated newly opened stores (coffee) after 1 day
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

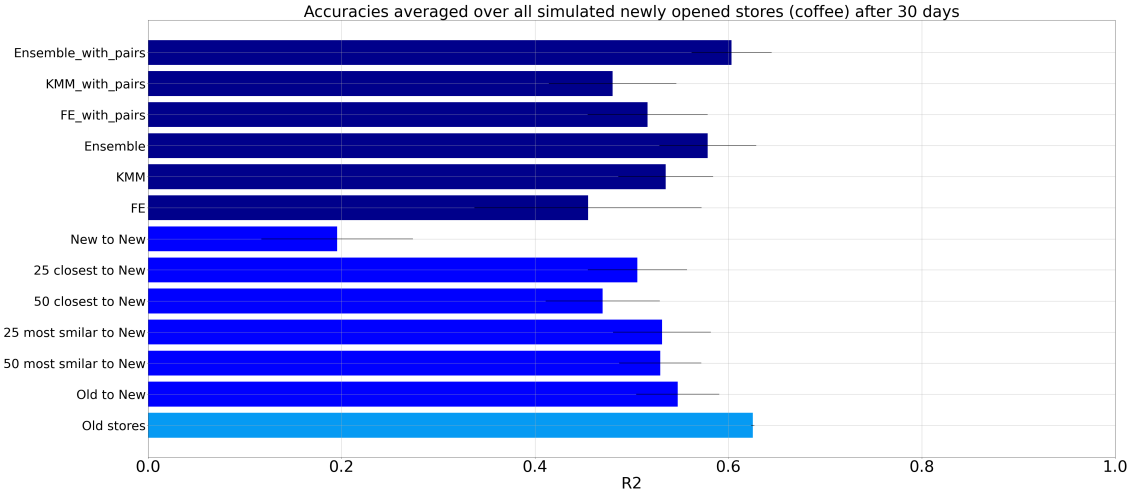
(a)


Accuracies averaged over all simulated newly opened stores (coffee) after 10 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

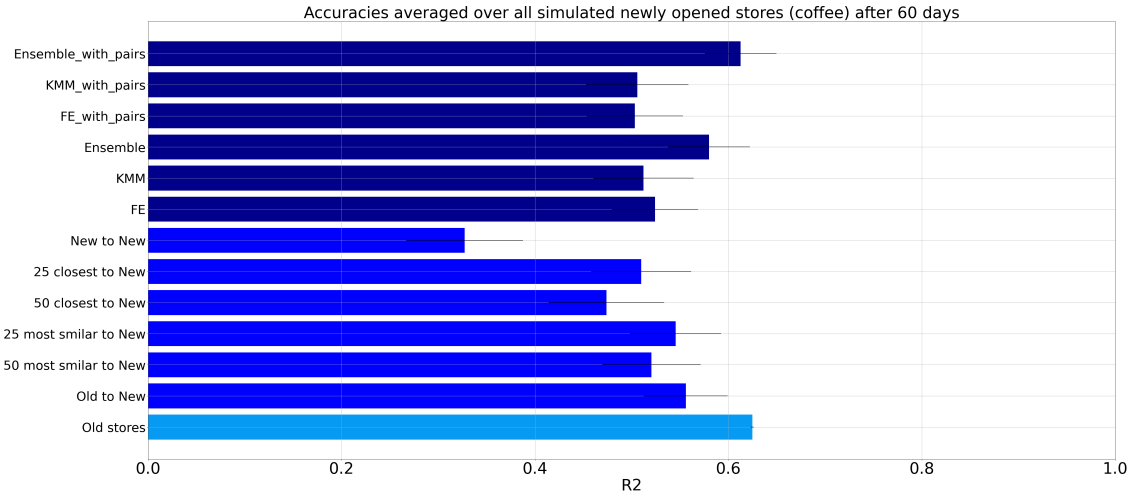
(b)


Accuracies averaged over all simulated newly opened stores (coffee) after 20 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

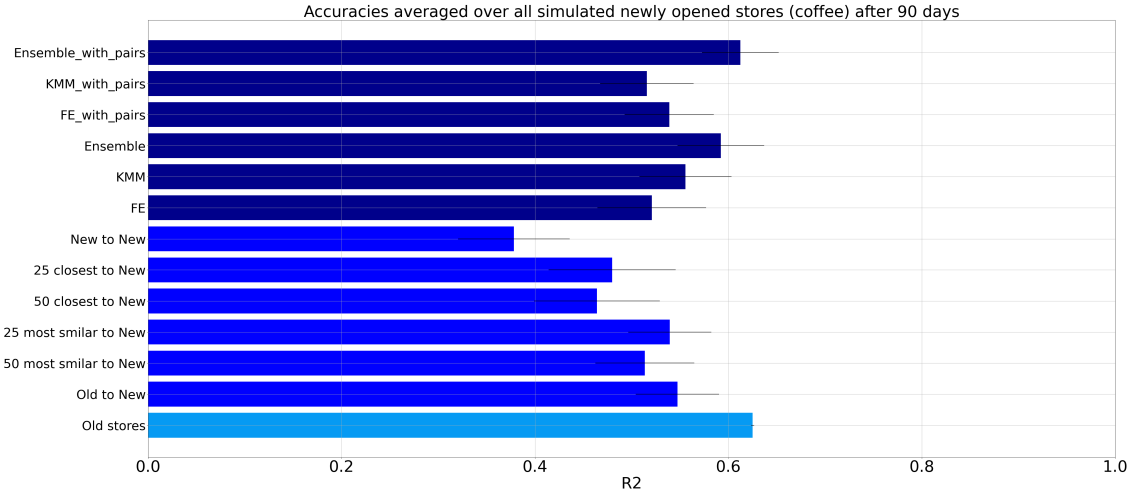
(c)


Accuracies averaged over all simulated newly opened stores (coffee) after 30 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

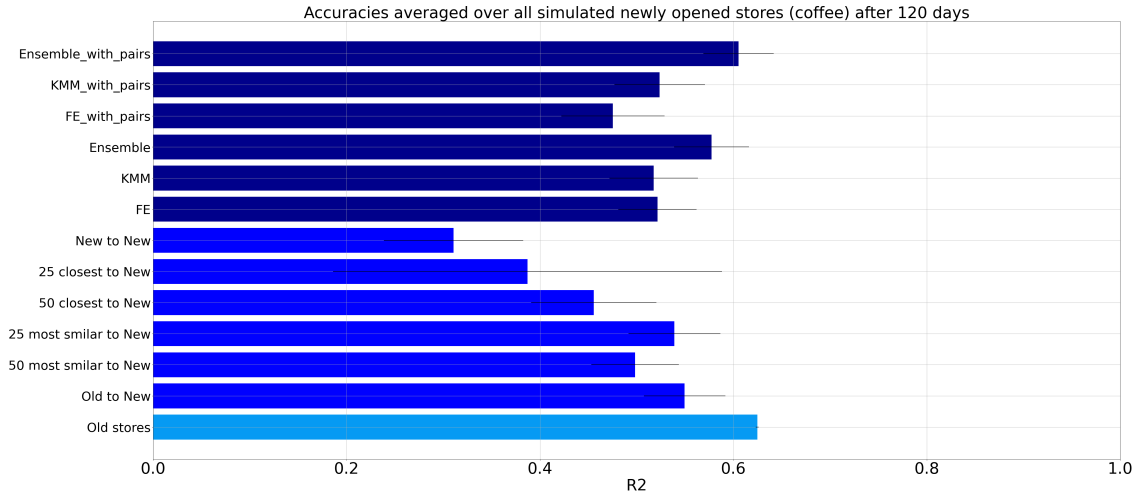
(d)


Accuracies averaged over all simulated newly opened stores (coffee) after 60 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

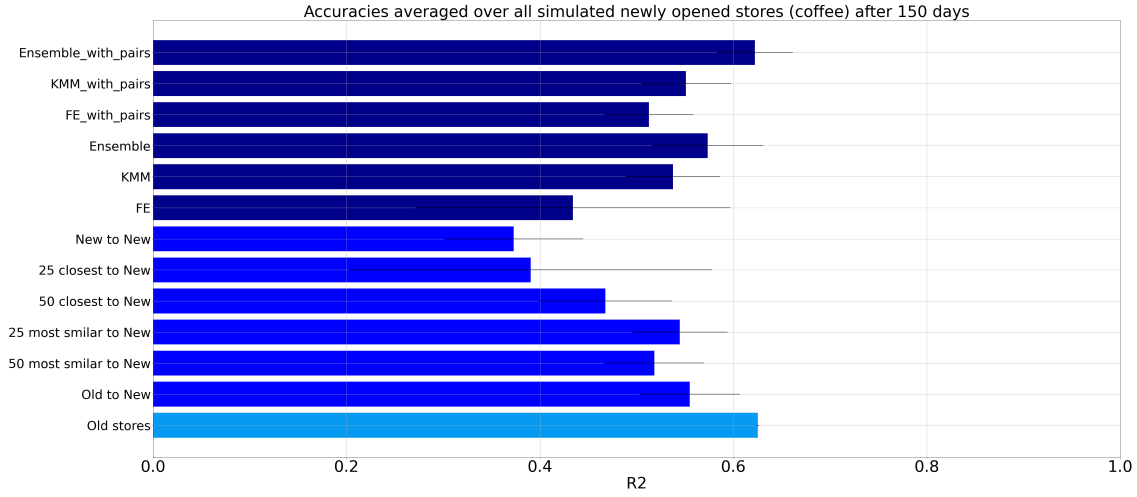
(e)


Accuracies averaged over all simulated newly opened stores (coffee) after 90 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

(f)



(g)



(h)

Figure 4.2: The accuracy levels of different models for the coffee category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

Table 4.4 shows the p-value of the difference between the best performing domain adaptation accuracies and the new-store accuracies, as the days go by after opening the new store. These p-values are calculated using paired t-test.

p-value							
Category	10	20	30	60	90	120	150
Coffee	0.004995	0.004995	0.01298	0.07462	0.0924	0.0005494	0.00025

Table 4.4: Comparison of p-values between the best performing domain adaptation technique and the new-store model.

The same analysis is repeated for the energy drinks category and the results show the same patterns (refer to **Figure 4.3**). As before, the prediction accuracy of the best performing domain adaptation method supersedes the old-store baseline.

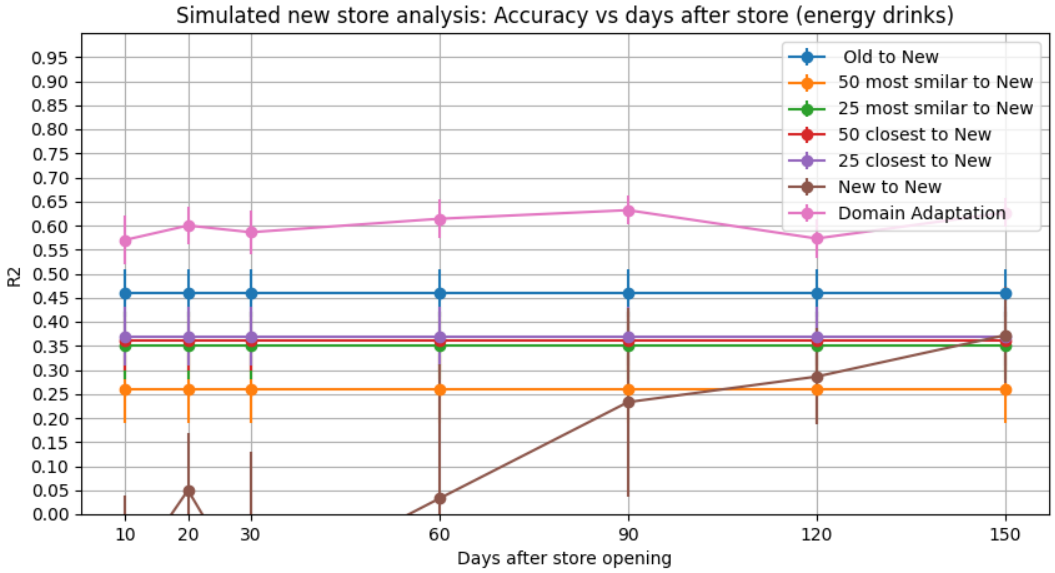
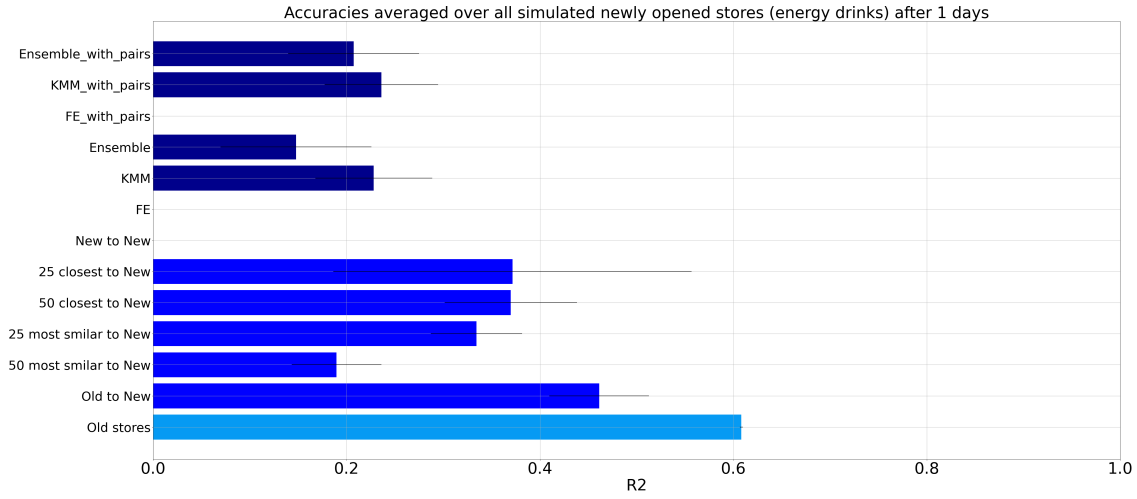
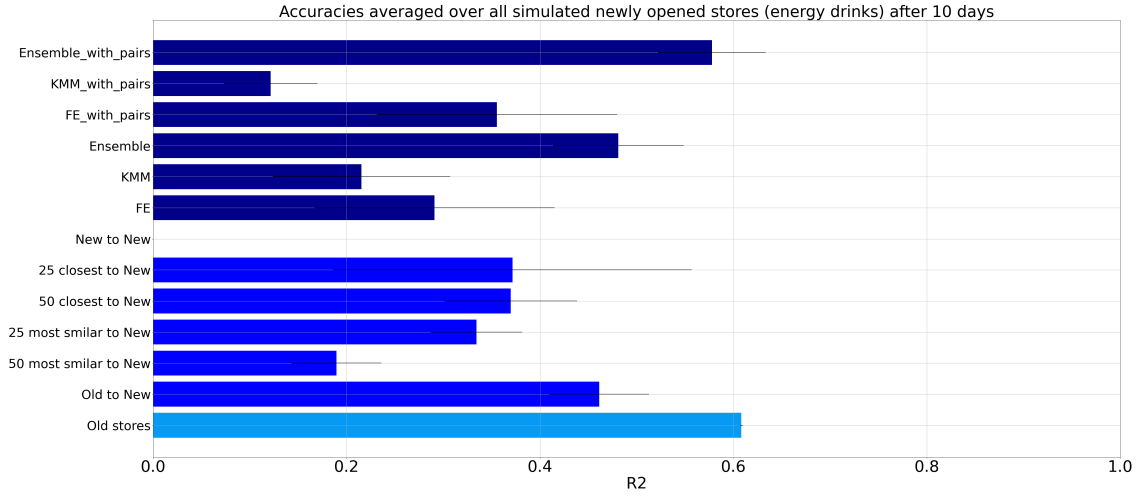


Figure 4.3: Comparison between the accuracy of different models for the energy drinks category as a function of the number of days from the opening of the new (simulated) store (XGBoost is considered as the base forecasting method).

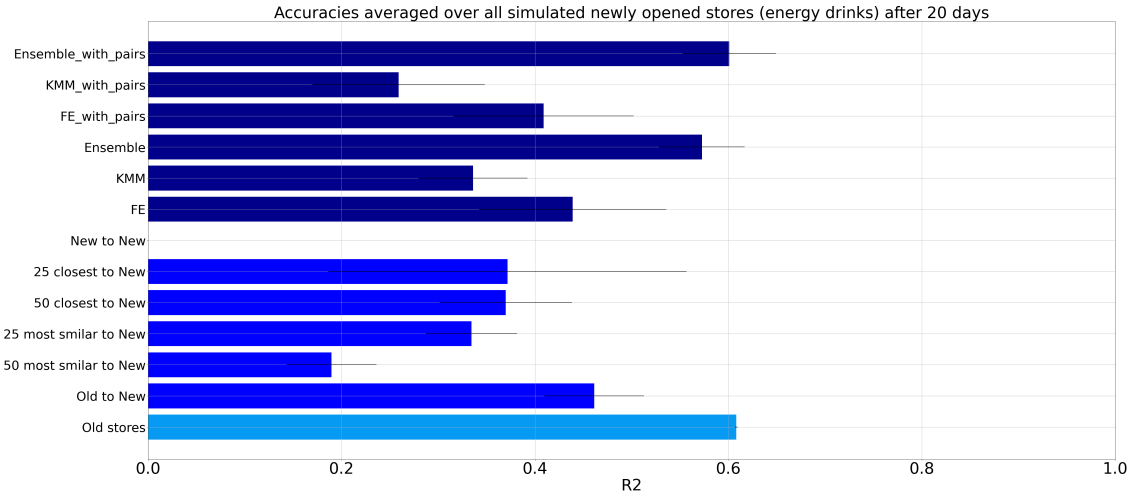
Figure 4.4 shows the accuracy levels of the different models for the energy drinks category. As in the coffee category, when the number of days from the opening of the new store increases, the demand prediction accuracy's improvement due to domain adaptation becomes more statistically significant.


Accuracies averaged over all simulated newly opened stores (energy drinks) after 1 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

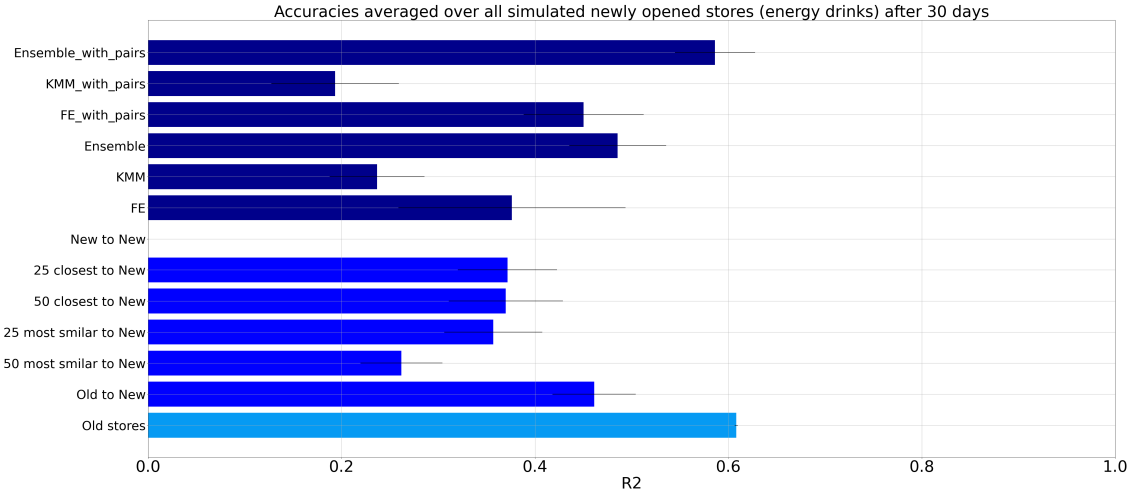
(a)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 10 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

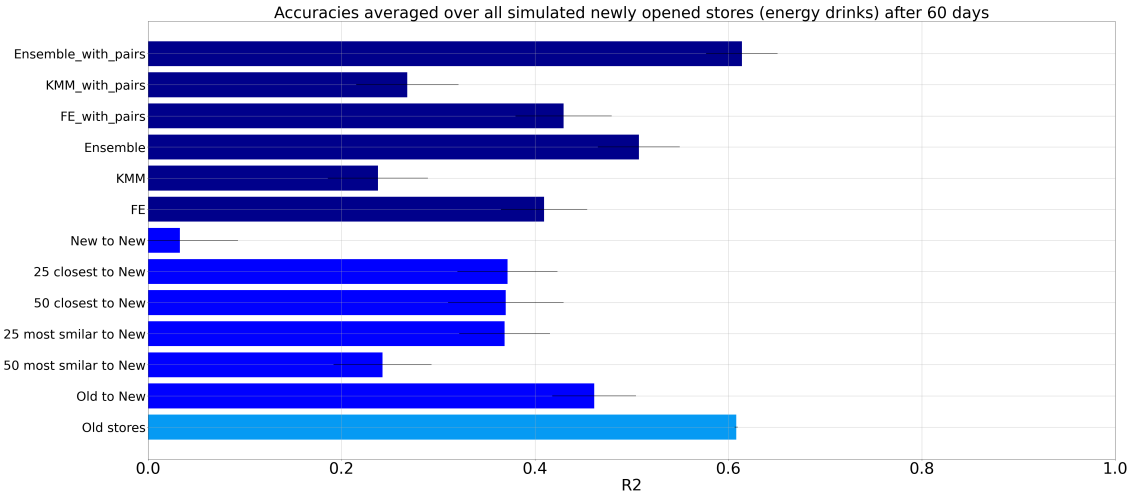
(b)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 20 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

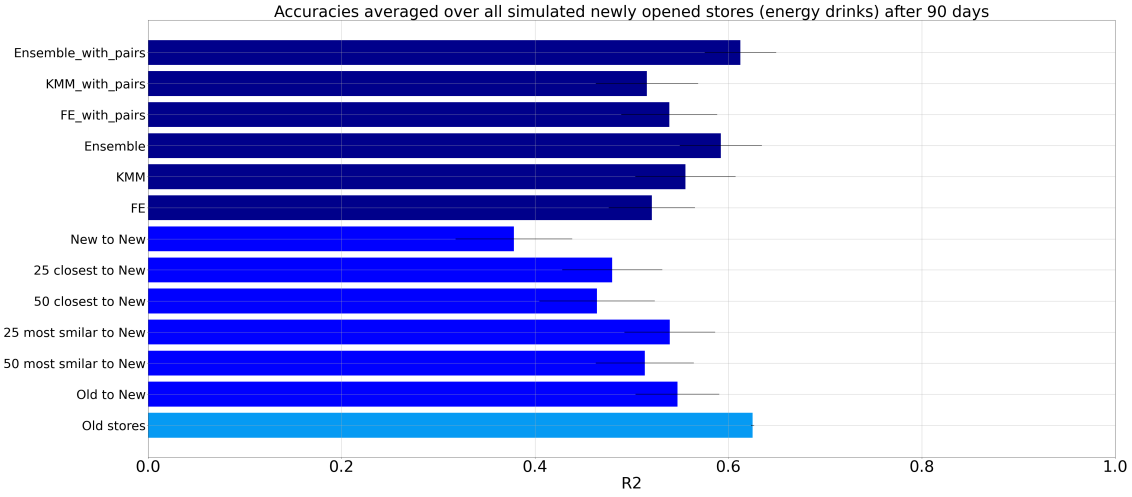
(c)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 30 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

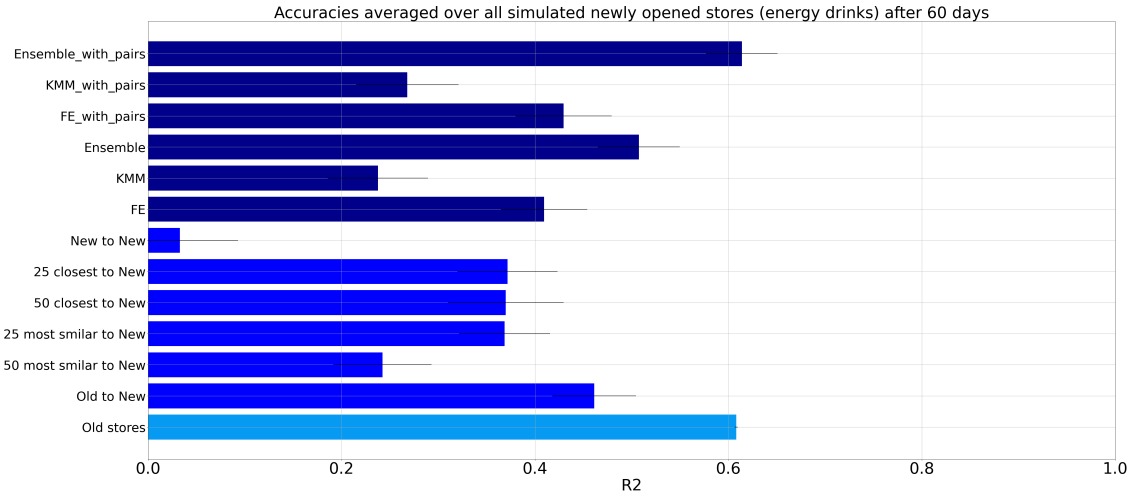
(d)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 60 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

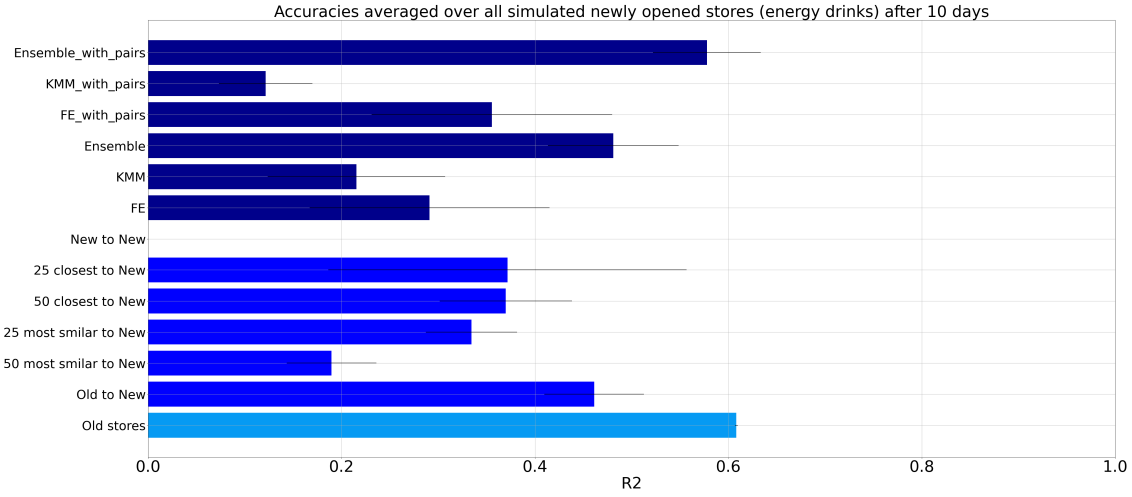
(e)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 90 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

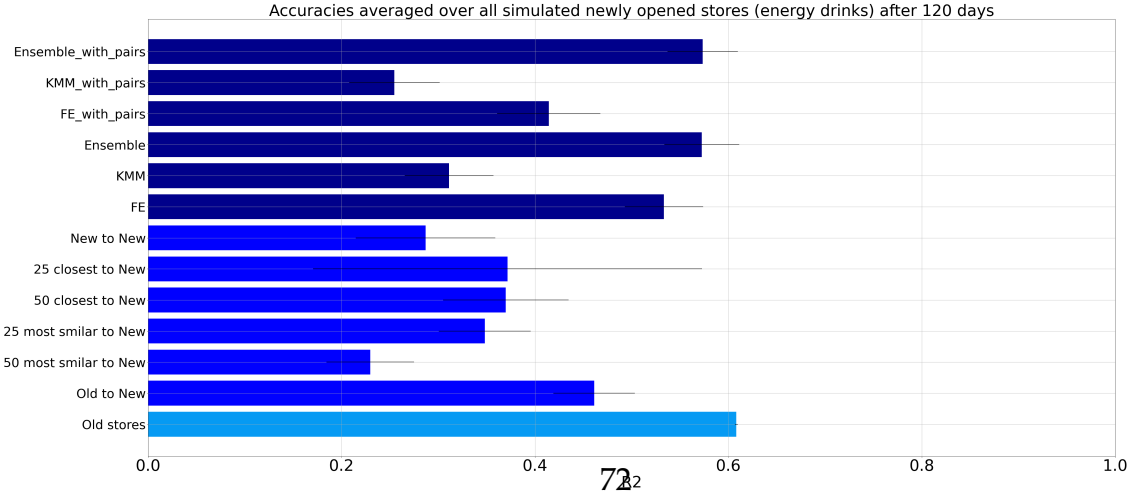
(f)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 60 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

(g)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 10 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

(h)


Accuracies averaged over all simulated newly opened stores (energy drinks) after 120 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
25 closest to New
50 closest to New
25 most smilar to New
50 most smilar to New
Old to New
Old stores
0.0
0.2
0.4
0.6
0.8
1.0
R2

(i)

Table 4.5 shows the p-value of the difference between the best performing domain adaptation accuracies and the new-store accuracies, as the days go by after opening the new store. These p-values are calculated using paired t-test.

p-value							
Category	10	20	30	60	90	120	150
Energy drinks	0.001897	0.00104	0.03435	0.01764	0.02058	0.09312	0.005347

Table 4.5: Comparison of p-values between the best performing domain adaptation technique and the new-store model.

Transformers

We next extend the analysis to the case where the base forecasting method is Transformers. **Figure 4.5** shows the comparison between the accuracy of different models for the coffee category when using Transformers as the base forecasting method versus when using XGBoost.

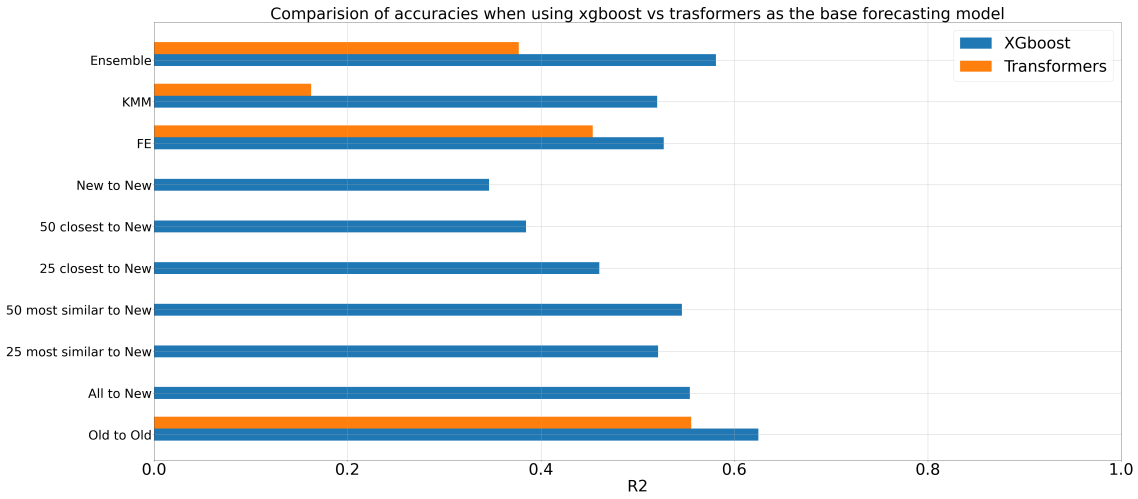


Figure 4.5: Comparison between the accuracy levels when using XGBoost and Transformers as the base forecasting method

As shown in **Figure 4.5**, using domain adaptation allows us to improve the prediction accuracy of the new-store models. However, the absolute value of the prediction accuracy when using Transformers is lower relative to XGBoost. Thus, we conclude that our study based on our dataset, it is better to use XGBoost as the base forecasting method. The same analysis was repeated for the energy drinks category and led to the same conclusion. The rest of the graphs are omitted for conciseness.

4.3.2 Domain Adaptation on an Actual New Store

In this section, we consider an actual store that was recently opened to confirm that the results from the previous section can be applied to a real-world setting. To identify a new store in our dataset, we asked for managerial information from our industry partner regarding a store that was recently opened. We asked for a store that was opened after pandemic- March 2020. The selected new store(store 1208) was opened June 22 2021 in Montreal. We repeat the same analysis as the previous section for both the categories of energy drinks and coffee and reproduce the results displayed in in **Figure 4.6**.

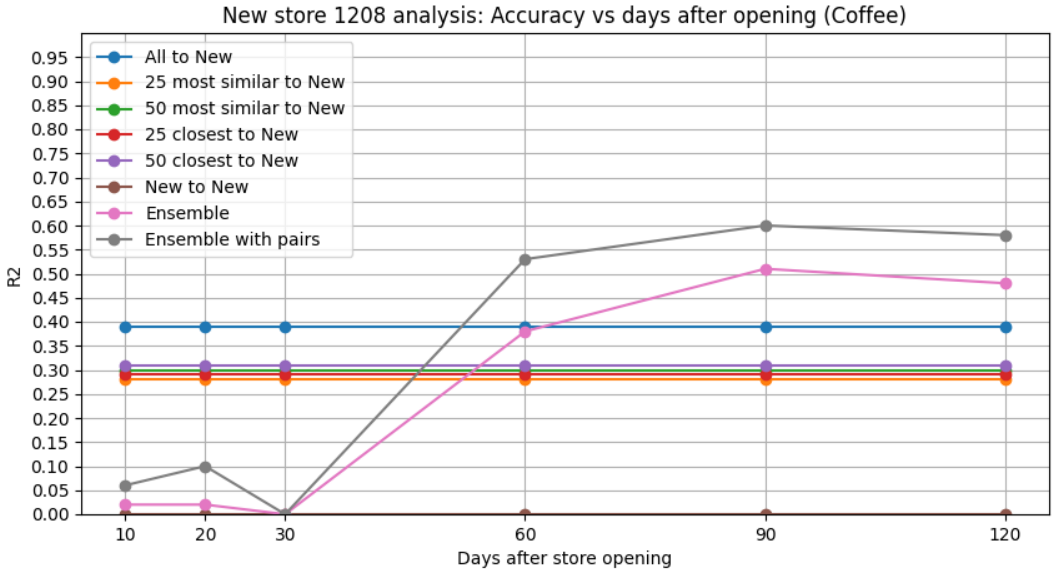
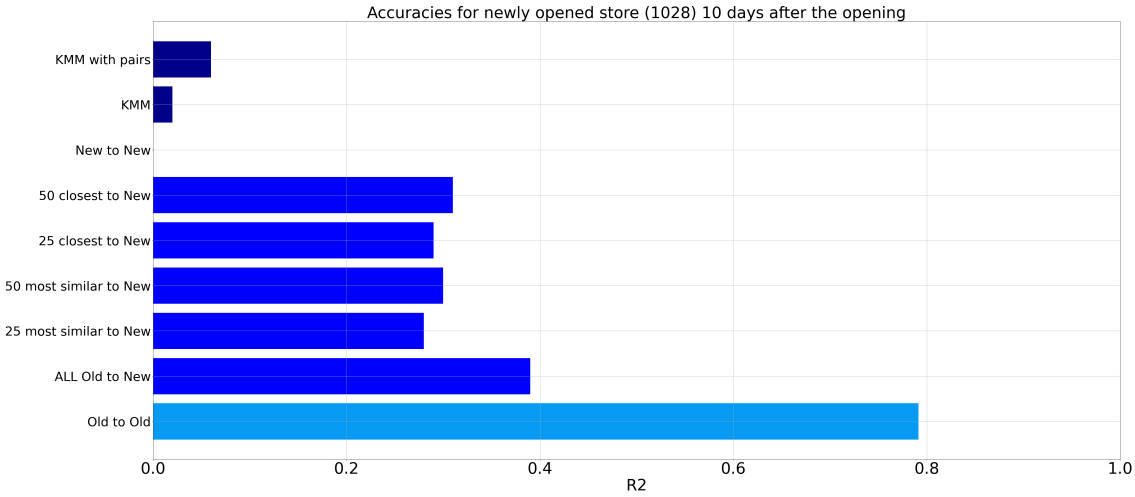


Figure 4.6: Comparison of the prediction accuracy of different models as a function of the number of days after the new store opening (XGBoost is considered as the base forecasting method).

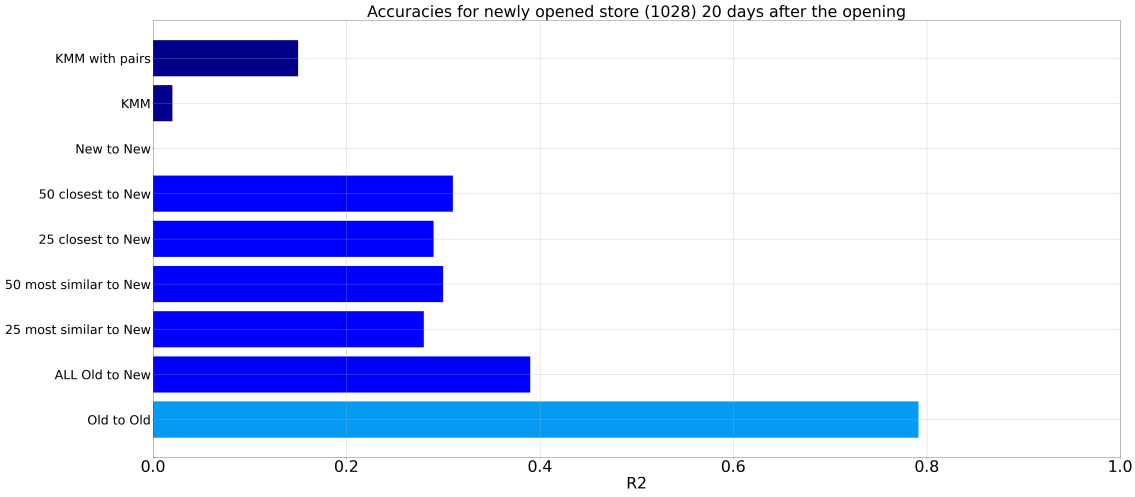
We expect that, as the number of days from the store opening increases and the data in the new store (new domain) becomes available, the accuracy of new-store models increases. For the coffee category, this intuition is readily confirmed in **Figure 4.6**. As shown in the figure, the prediction accuracy of the new-store model (brown line) increases with the number of days after the new store opening.

As shown in **Figure 4.6**, domain adaptation improves the prediction accuracy of the new-store models (the blue, orange, green, red, purple, and brown lines in the graph). Specifically, the prediction accuracy of the domain adaptation methods (grey and pink lines) outperform the blue line (i.e., the best performing new-store model) after 60 days.

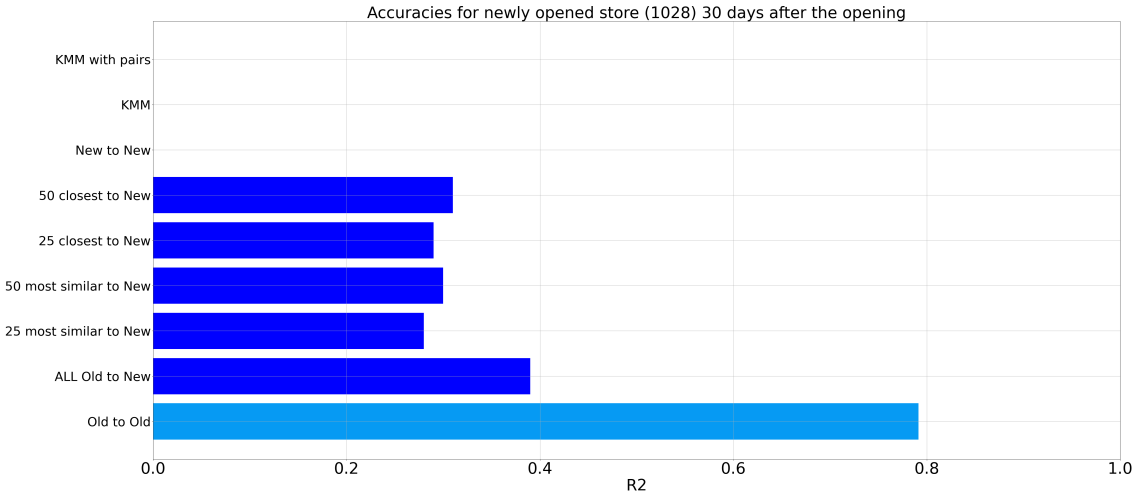
Figure 4.7 reports the accuracy levels of different models for the coffee category. As shown in the graphs, after 60 days, the domain adaptation methods supersede the accuracy of new store models. We note that neither the new store models nor the domain adaptation methods can attain the same level as the old-store baseline.


Accuracies for newly opened store (1028) 10 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

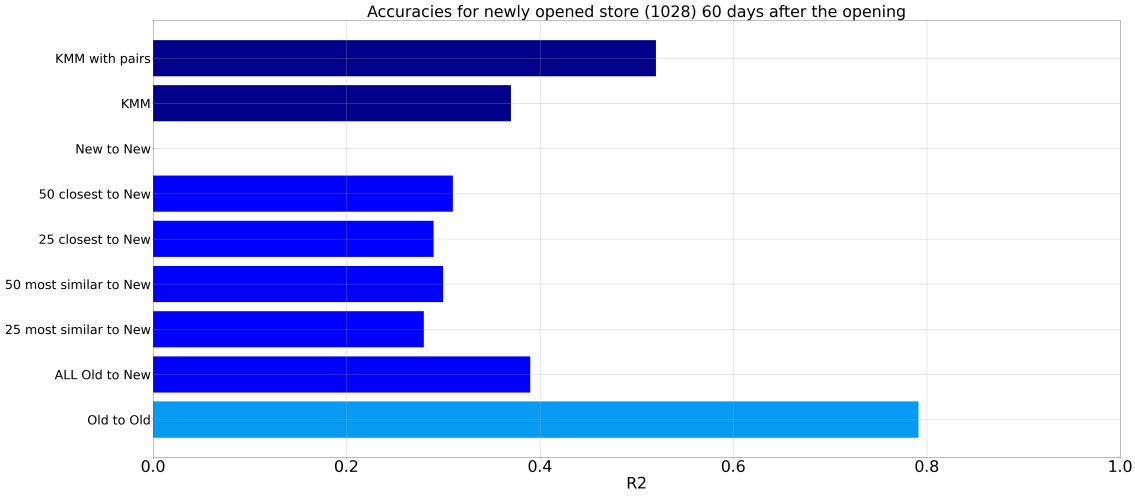
(a)


Accuracies for newly opened store (1028) 20 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

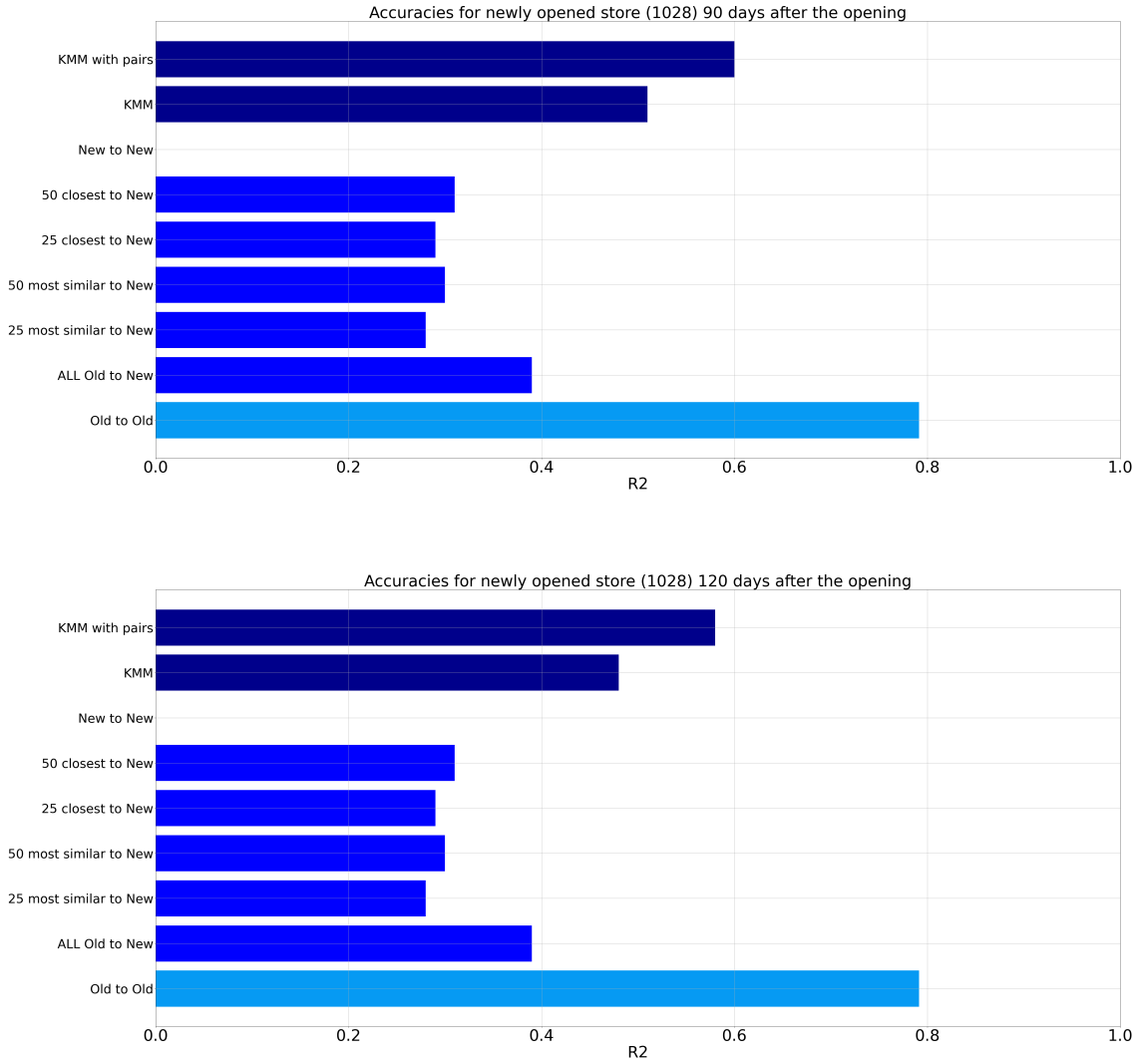
(b)


Accuracies for newly opened store (1028) 30 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

(c)


Accuracies for newly opened store (1028) 60 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

(d)



(e)

Figure 4.7: Comparison of the accuracy of different models for the coffee category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

The same analysis is repeated for the energy drinks category and the results confirm the same trends (refer to **Figure 4.8**).

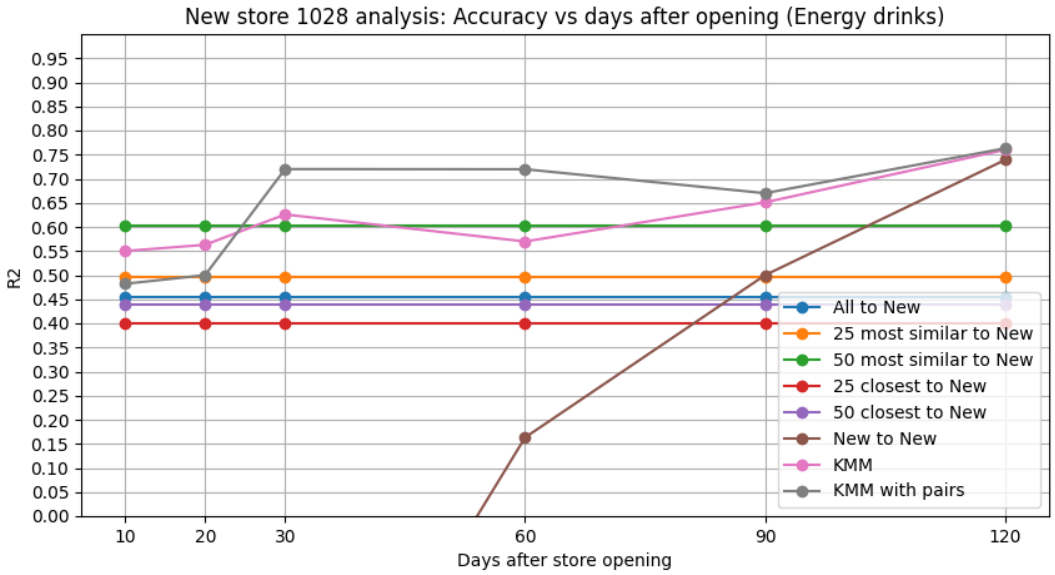


Figure 4.8: Comparison of the accuracy of different models for the energy drinks category as a function of the number of days from the opening of the new store (XGBoost is considered as the base forecasting method).

We expect that, as the number of days after the store opening increases and the data in the new store (new domain) becomes more prevalent, the accuracy of the new-store models improves. For the coffee category, this intuition is readily confirmed in **Figure 4.8**. As shown in the figure, the prediction accuracy of the new-store model (brown line) increases with the number days after the new store opening. The blue, orange, green, and red lines correspond to the models that were trained using the data from the old stores, the 50 most similar old stores, the 25 most similar old stores, the 50 closest old stores, the 25 closest old stores, and tested using the data form the new store respectively.

The grey line shows the prediction accuracy of the model using domain adaptation (out of the six domain adaptation models, only the two best performing ones are reported: KMM and KMM with the pairing technique).

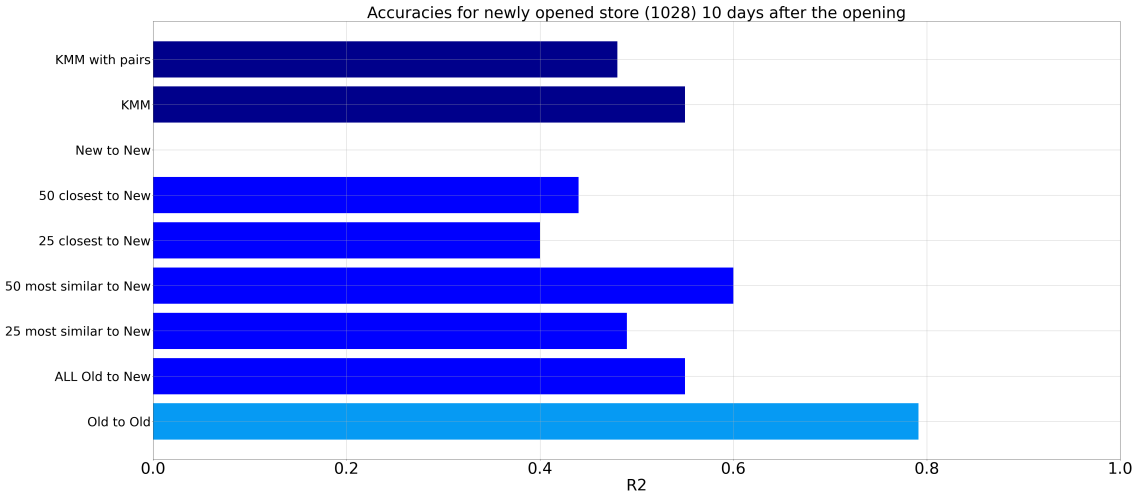
As shown in the figure, domain adaptation allows us to improve the prediction accuracy of the new-store models (the blue, orange, green, red, purple, and brown lines in the graph). After 30 days, the accuracy of domain adaptation supersedes the green line (i.e.,

the best performing new store model). This decrease in the time window needed before the accuracy of the new store demand prediction is reliable is substantial in reducing the unnecessary costs incurred by the retailer.

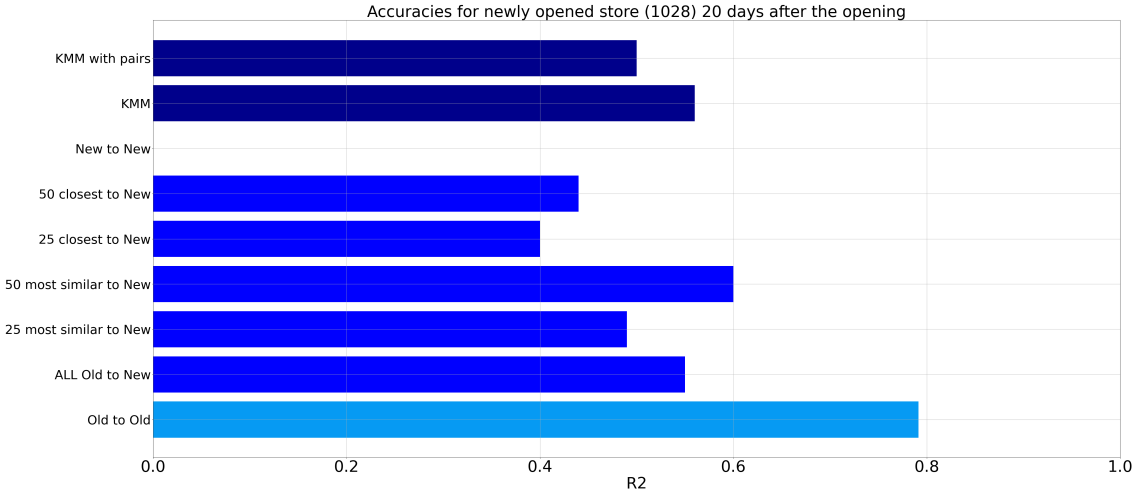
Figure 4.9 shows the accuracy levels of different models for the energy drinks category. We find that the accuracy of domain adaptation (KMM with pairs) is lower than the new-store accuracy levels, albeit, they become more prominent after 30 days.

Another point that is worth mentioning is the fact that the new store model never attains the old-store baseline (the lightest shade of blue). However, the result based on using KMM with pairs is at the same level as the old-store baseline after 120 days.

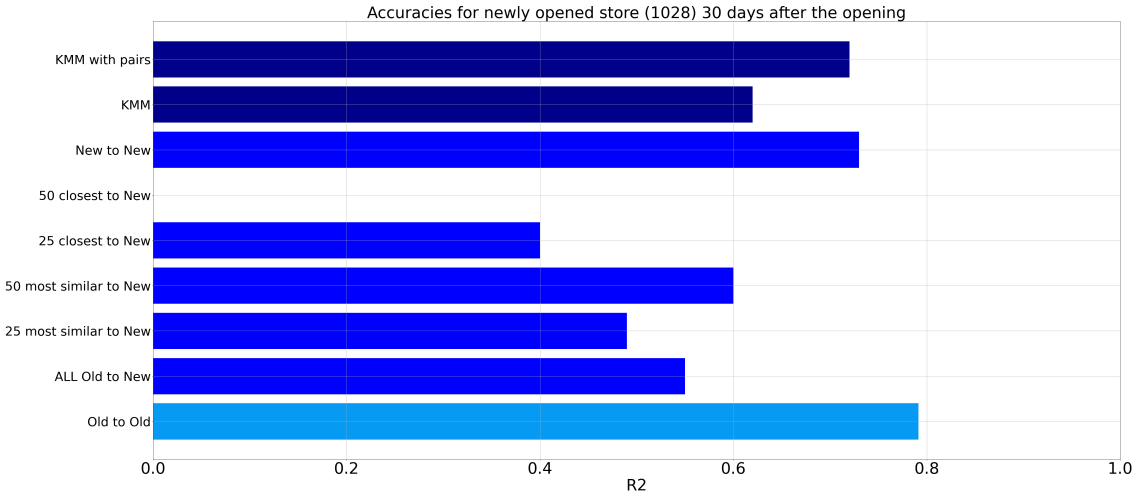
Comparing **Figure 4.1**, **Figure 4.3**,**Figure 4.8**, and **Figure 4.6**, we conclude that in the actual new store the accuracy of domain adaptation ramps up as the days pass after the new store opening. However, for the case of simulated new store, the accuracy of domain adaptation is more steady. This different behaviour can be due to the fact that in the simulated store case study, the domain shift between the data in the simulated new store and old stores is smaller. The simulated new store is in fact an old store with a hypothetical start date. So, the shifts that can be expected when we actually open a new store are not present in this case study.


Accuracies for newly opened store (1028) 10 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

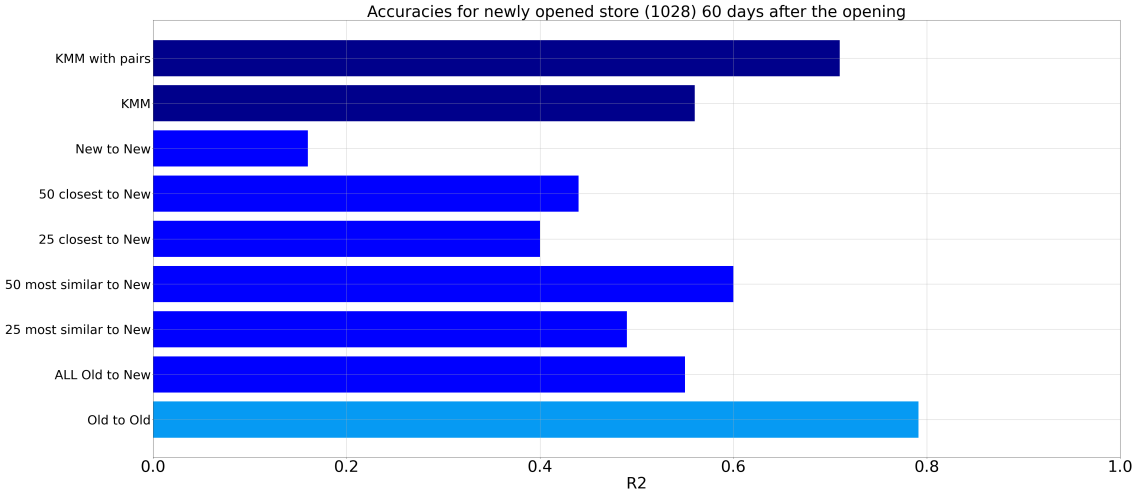
(a)


Accuracies for newly opened store (1028) 20 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

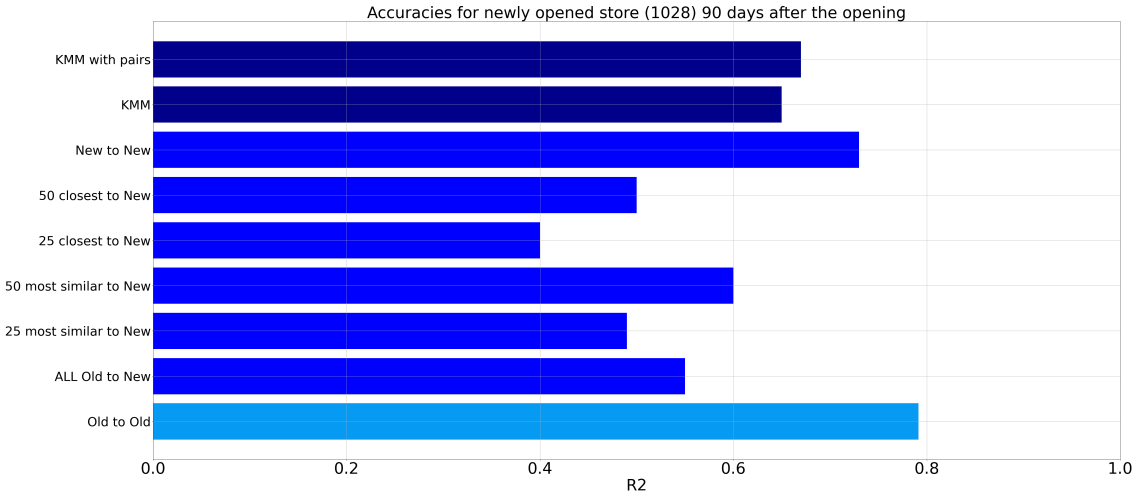
(b)


Accuracies for newly opened store (1028) 30 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

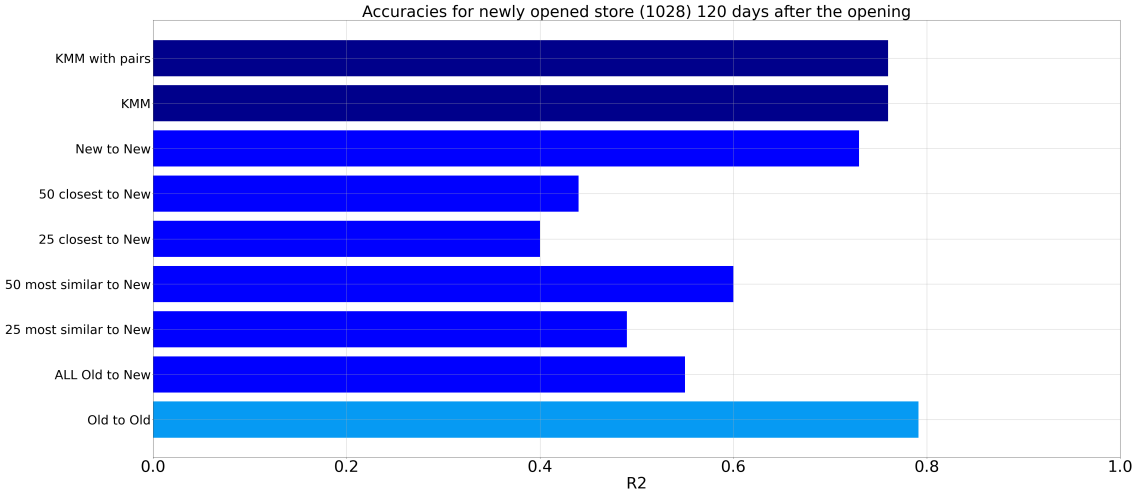
(c)


Accuracies for newly opened store (1028) 60 days after the opening
KMM with pairs
KMM
New to New
50 closest to New
25 closest to New
50 most similar to New
25 most similar to New
ALL Old to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

(d)



(e)



(f)

Figure 4.9: Accuracy levels of different models for the energy drinks category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

4.4 Analysis-3: Introduction of a New Product

In this analysis, our goal is to predict the demand for a new product in each of the 89 stores in our dataset. We first calculate the accuracy of this demand prediction task before and after the new product introduction and show that although the accuracy of demand prediction for old products is high, this accuracy is not satisfactory for the new products. We use domain adaptation techniques to mitigate the negative effect of the domain shift caused by product introduction.

Like in the previous two sections, we test two methods as the base demand prediction models in this analysis: XGBoost and Transformers.

In this analysis, we use the price and quantities sales information, and we conduct our analysis for the same two high-selling categories as before (coffee and energy drinks). We use simulated new products to validate our intuitions.

4.4.1 Domain Adaptation on Simulated New Products

XGboost

We consider each product in the category of coffee or energy drinks as a newly introduced product with a hypothetical introduction date. We substitute zeros for its sales before the hypothetical start date. The rest of the products in the same category are considered the "old" products. We repeat the calculations for each product in the category of coffee or energy drink, and report the average value.

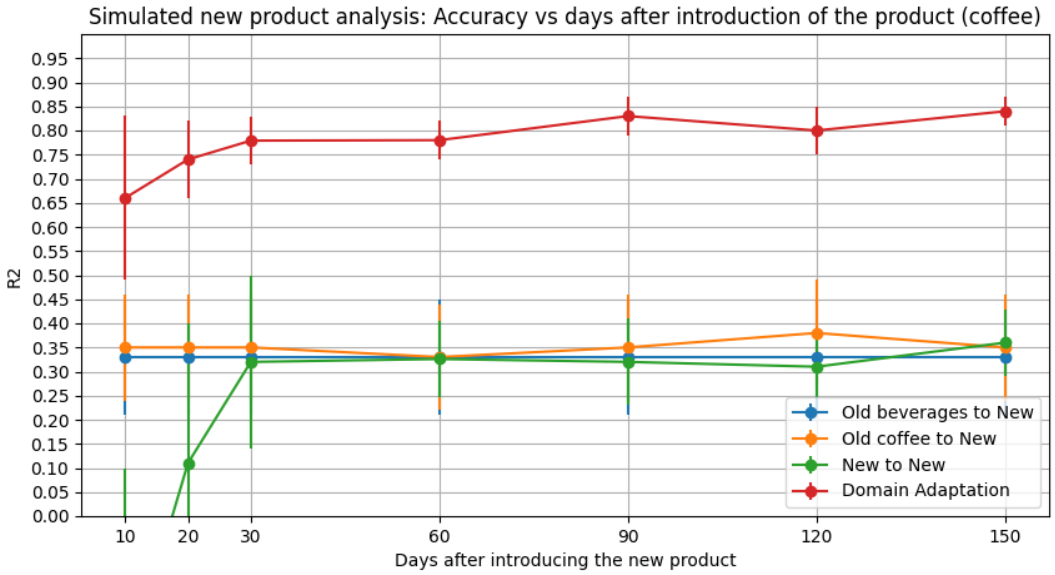


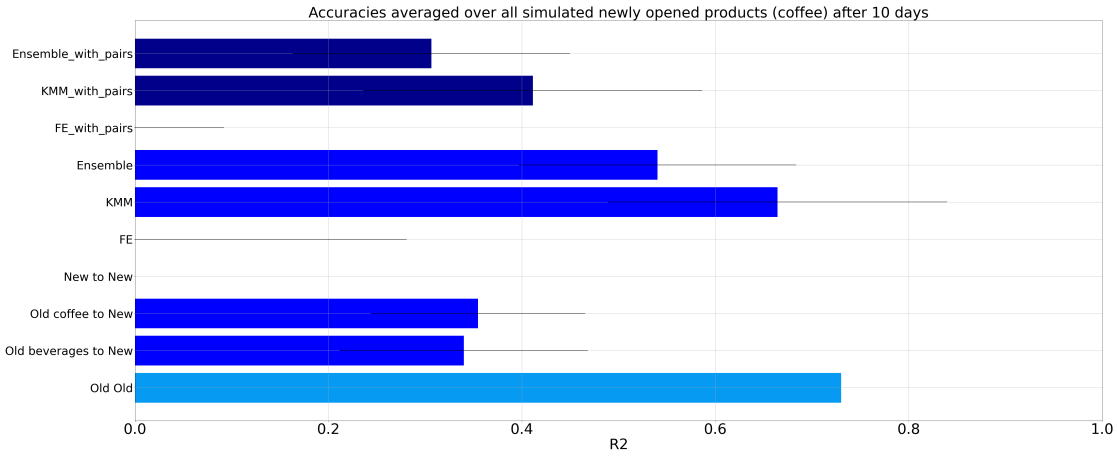
Figure 4.10: Comparison of the accuracy of different models when introducing a simulated new product in the coffee category (XGBoost is considered as the base forecasting method).

We expect that, as the number of days from the introduction of the new product increases and the data corresponding to the new product (new domain) becomes more prevalent, the accuracy of new-store models improves. For the coffee category, this intuition is readily confirmed in **Figure 4.10**. As shown in the figure, the prediction accuracy of the new-product model (brown line) increases with the number of days after the introduction of the new product. Similar to the new store analysis, we consider using the data from a subset of old products (i.e., the same way we previously used the data from the 25 closest stores, 50 closest stores, 25 most similar stores, and 50 most similar stores to predict the demand of the new store). In this analysis, we use the data from products in the beverage category and products in the coffee category to predict the demand for the new product. The blue, orange, green, and red lines correspond to the models trained using the data from beverages (the parent category) and coffee (a sub-category of beverages) and tested using the data from the new store, respectively.

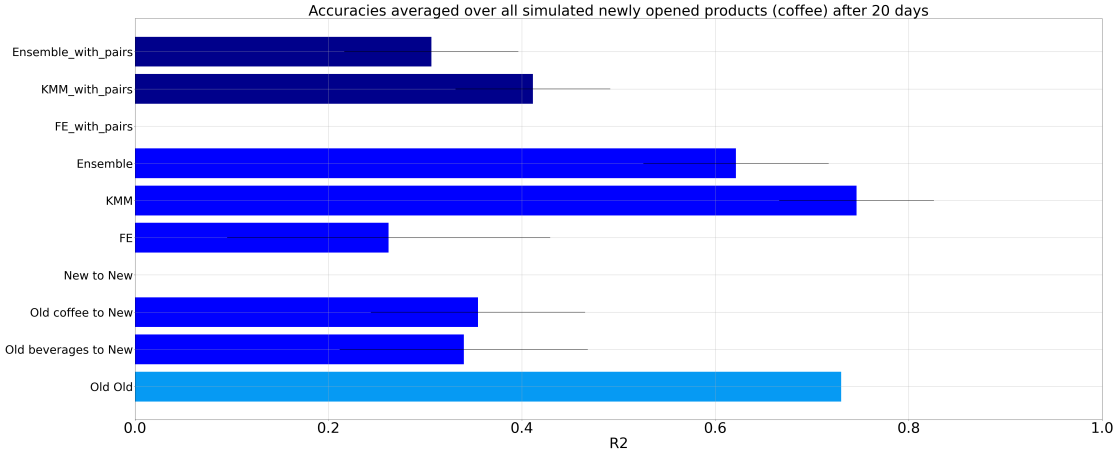
The red line shows the accuracy of the model using domain adaptation (out of the three domain adaptation models, only the model with the highest accuracy is reported, which

in this case is the ensemble method). As shown in the figure, domain adaptation allows us to improve the prediction accuracy of the new-product models (the blue, orange, and green lines in the graph).

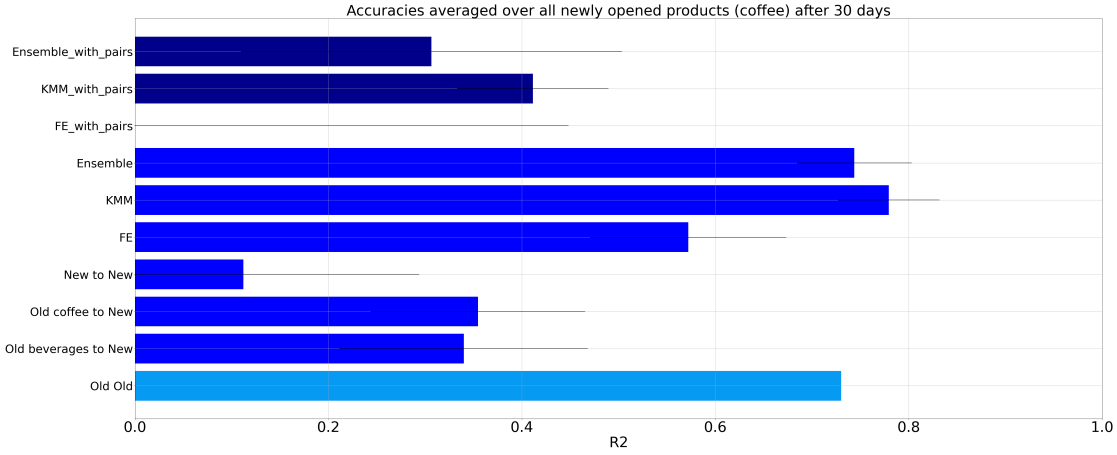
Figure 4.16 reports the accuracy levels of different models for the coffee category. The bottom-most bar (shown in the lightest shade of blue) represents the accuracy of the model trained and tested on data from the old products. The middle bars show the accuracy of the models trained on data from all the old products, the old products in the hot beverages category, the old products in the coffee category, the new products, and tested on the new product from bottom to top. The bars in the darkest shade of blue represent the domain adaptation methods. They correspond to FE, KMM, the ensemble of FE and KMM, the FE with the pairing method, KMM with the pairing method, and the ensemble method with the pairing method from bottom to top. We note that the prediction accuracy of domain adaptation is higher than the accuracy of the old-product model after 30 days.


Accuracies averaged over all simulated newly opened products (coffee) after 10 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old coffee to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

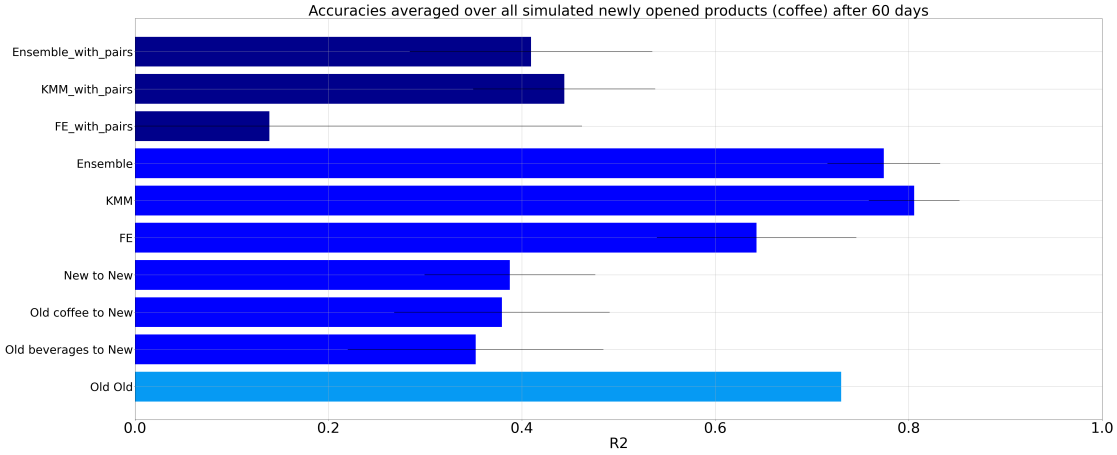
(a)


Accuracies averaged over all simulated newly opened products (coffee) after 20 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old coffee to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

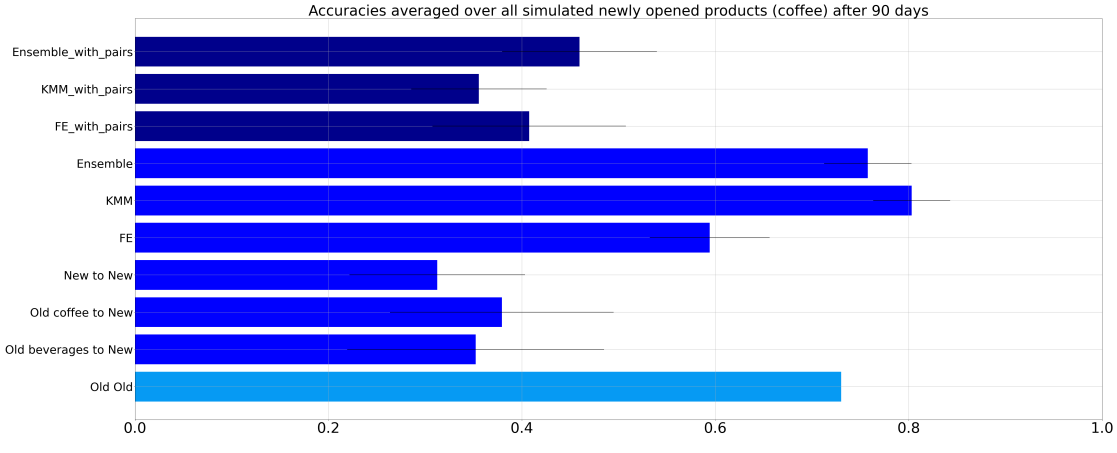
(b)


Accuracies averaged over all newly opened products (coffee) after 30 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old coffee to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

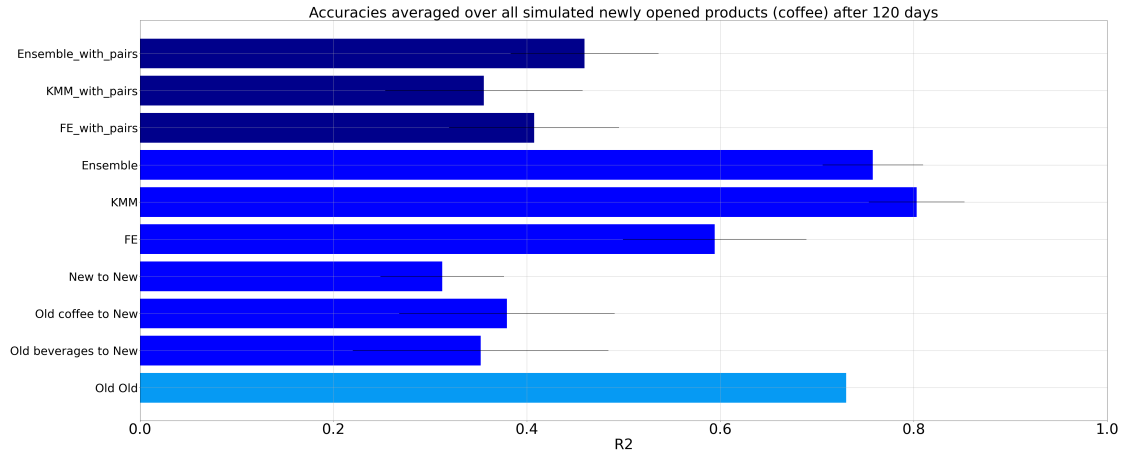
(c)


Accuracies averaged over all simulated newly opened products (coffee) after 60 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old coffee to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

(d)


Accuracies averaged over all simulated newly opened products (coffee) after 90 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old coffee to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0

(e)



(f)

Figure 4.11: Accuraciy levels of different models when introducing a simulated new product in the coffee category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

Table 4.6 shows the p-value of the difference between the best performing domain adaptation accuracies and the new-product accuracies, as the days go by after introducing the new product. These p-values are calculated using paired t-test.

p-value							
Category	10	20	30	60	90	120	150
Coffee	0.01942	0.0247	0.001725	0.003908	0.002745	0.002163	0.01472

Table 4.6: Comparison of p-values between the best performing domain adaptation technique and the new-product model.

The same analysis is repeated for the energy drinks category and the results show the same patterns (refer to **Figure 4.12**).

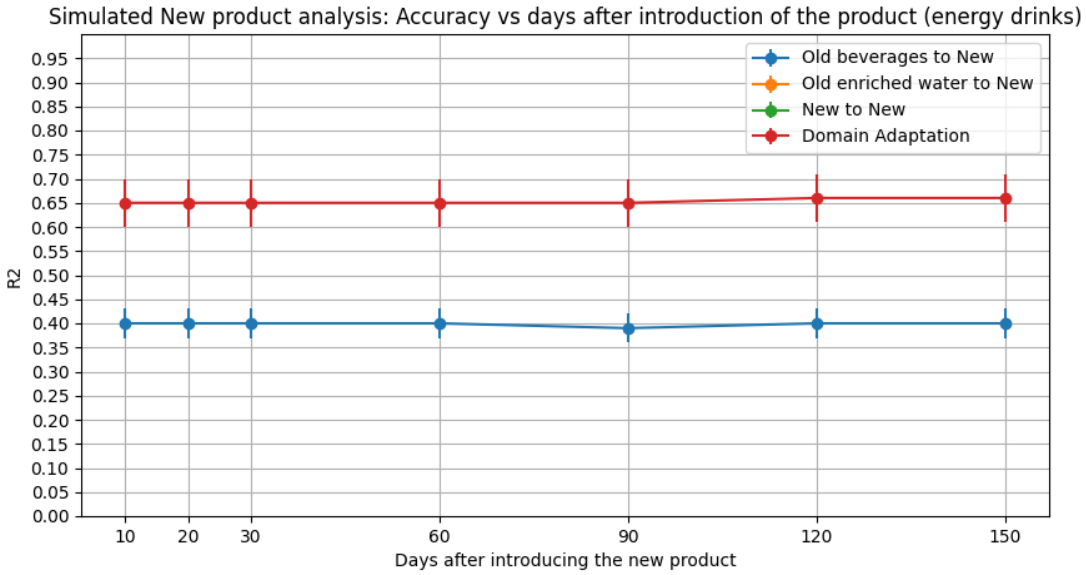
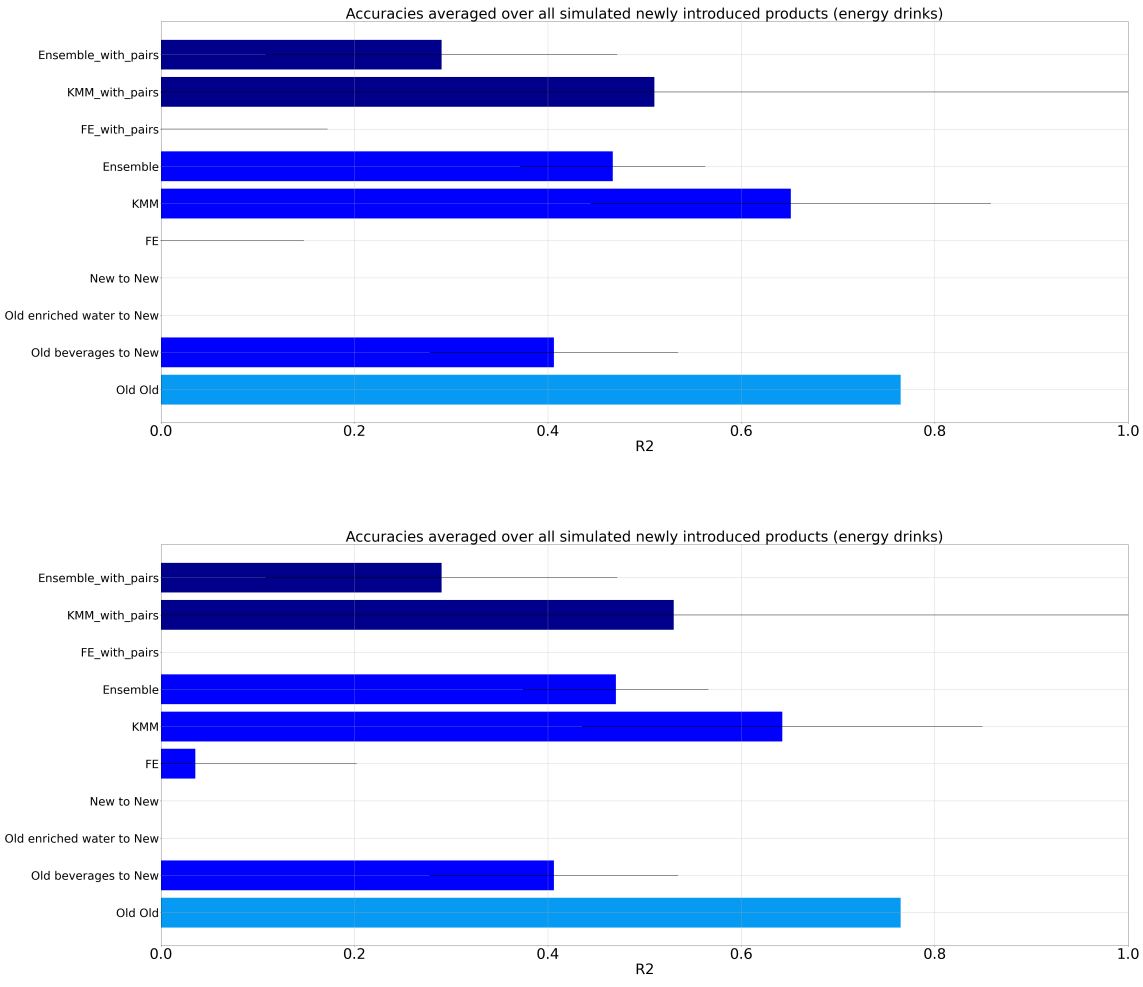


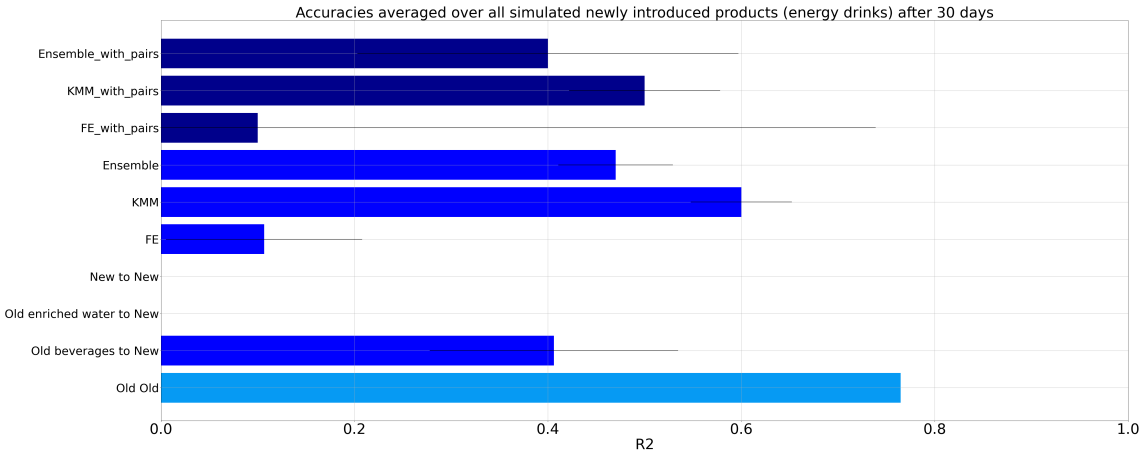
Figure 4.12:

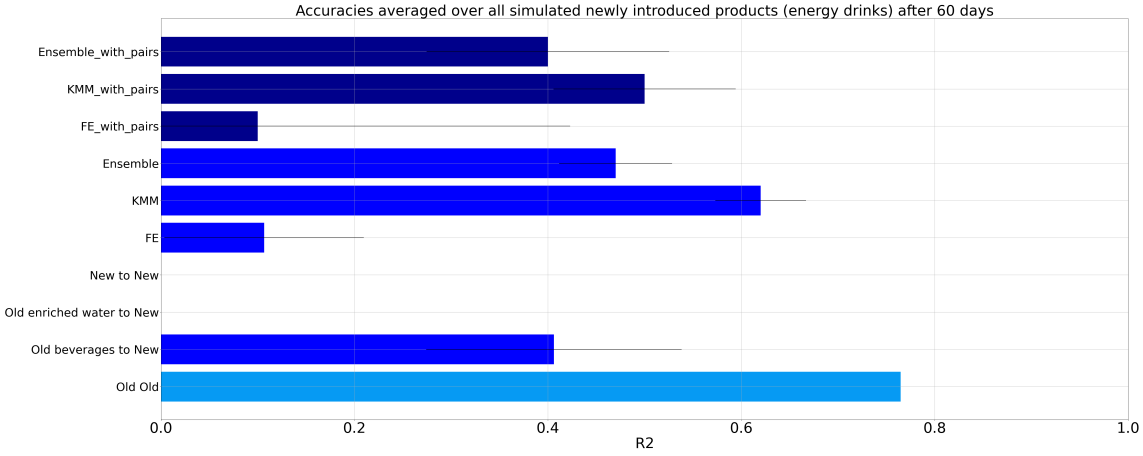
Comparison of accuracy of different models as a function of the number of days after the simulated new product introduction in the energy drinks category (XGBoost is considered as the base forecasting method).

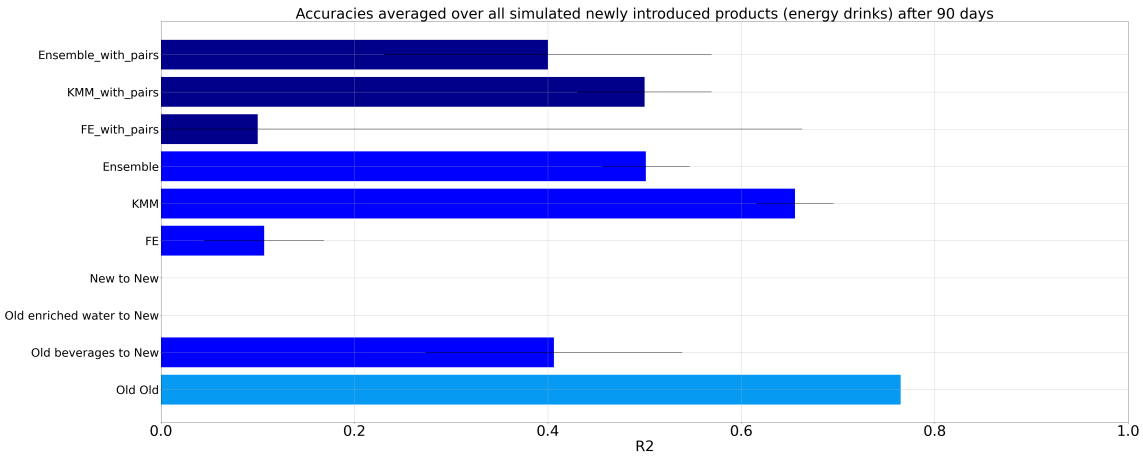
We expect that, as the number of days from the new product introduction increases and the data corresponding to the new product (new domain) becomes more prevalent, the accuracy of new-store models improves. For the energy drinks category, this intuition is readily confirmed in **Figure 4.10**. As shown in the figure, the accuracy of the new-product model (brown line) increases with the number of days after the new product introduction. The blue, orange, green, and red lines correspond to the models trained on data from old energy drinks (the parent category), old enriched water (a sub-category of beverages), and tested on the data from the new store, respectively. The red line shows the accuracy of the model using domain adaptation (out of the three domain adaptation models, only the model with the highest accuracy is reported, which in this case is the ensemble method). As shown in the figure, domain adaptation allows us to improve the accuracy of new-product models (the blue, orange, and green lines in the graph).

Figure 4.16 reports the accuracy levels of different models for the energy drinks category. The bottom-most bar (shown in the lightest shade of blue) represents the accuracy of the model trained and tested using the data from old products. The middle bars show the accuracy of the models trained on data from all the old products, the old products in the hot beverages category, the old products in the energy drinks category, and the new products and tested on the new product from bottom to top. The bars in the darkest shade of blue represent the domain adaptation methods. They correspond to FE, KMM, the ensemble of FE and KMM, FE with the pairing method, KMM with the pairing method, and the ensemble method with the pairing method from bottom to top.




Accuracies averaged over all simulated newly introduced products (energy drinks) after 30 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old enriched water to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2


Accuracies averaged over all simulated newly introduced products (energy drinks) after 60 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old enriched water to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2


Accuracies averaged over all simulated newly introduced products (energy drinks) after 90 days
Ensemble_with_pairs
KMM_with_pairs
FE_with_pairs
Ensemble
KMM
FE
New to New
Old enriched water to New
Old beverages to New
Old Old
0.0
0.2
0.4
0.6
0.8
1.0
R2

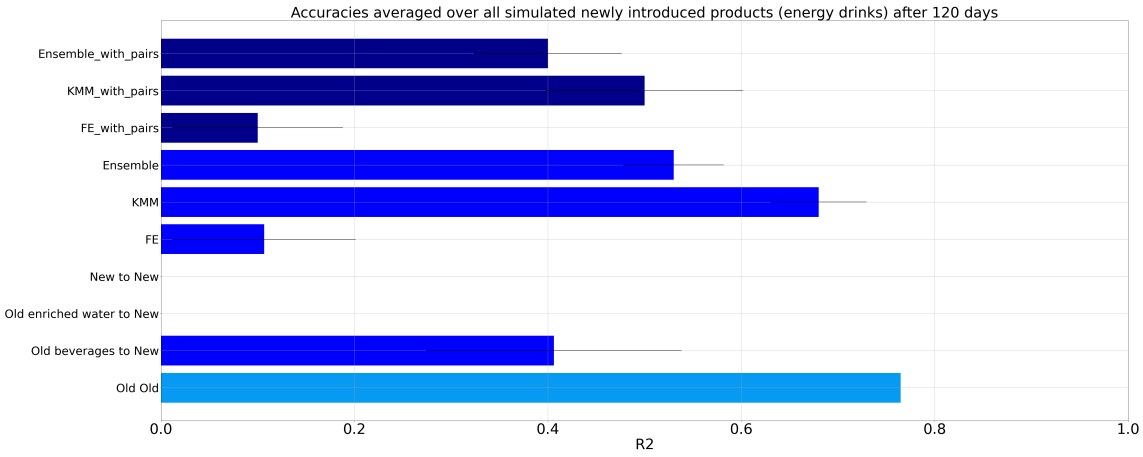


Figure 4.13: Accuracy levels of different models when introducing a new product for the energy drinks category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

Table 4.7 shows the p-value of the difference between the best performing domain adaptation accuracies and the new-product accuracies, as the days go by after introducing the new product. These p-values are calculated using paired t-test.

p-value							
Category	10	20	30	60	90	120	150
Energy drinks	0.02703	0.007601	0.005	0	0	0	0

Table 4.7: Comparison of p-values between the best performing domain adaptation technique and the new-product model.

Transformers

We next extend the analysis to the case where the base forecasting method is Transformers for the coffee category. **Figure 4.14** summarizes the comparison between the accuracy of different models when using Transformers as the base forecasting method versus when using XGBoost.

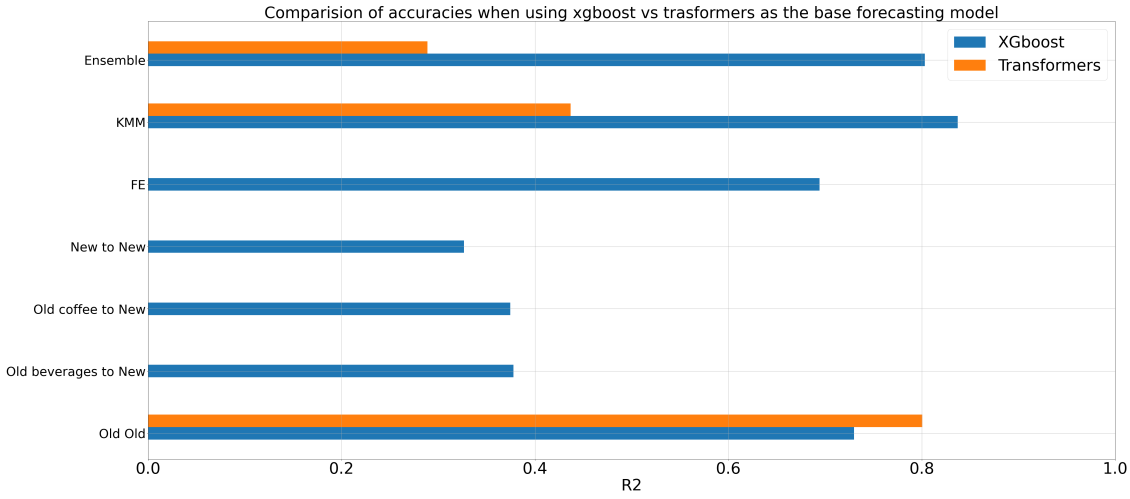


Figure 4.14: Comparison between the accuracy of models when using XGBoost vs. Transformers as the base demand forecasting method

As shown in **Figure 4.14**, using domain adaptation allows us to improve the prediction accuracy of new-product models. However, the absolute value of the accuracy obtained with Transformers is lower relative to XGBoost. Thus, we conclude that for our case study based on our dataset, it is better, once again, to use XGBoost as the base forecasting method. The same analysis was repeated for the energy drinks category and led to the same conclusion. The rest of the graphs are omitted for conciseness.

4.4.2 Domain Adaptation on an Actual New Product

In this subsection, we study a product that was recently introduced to confirm that the results from the previous section can be applied to a real-world setting. To identify a new product, we asked for managerial information from our industry partner regarding a product that was introduced recently in the energy drinks category. We asked for a product that was introduced after the COVID-19 pandemic. The selected new product, GURU GUAYUSA 355ML, 81843, is introduced on 2021-07-04 in all the 89 stores in Montreal. We repeat the same analysis as in the previous section and reproduce the results in **Figure 4.15**.

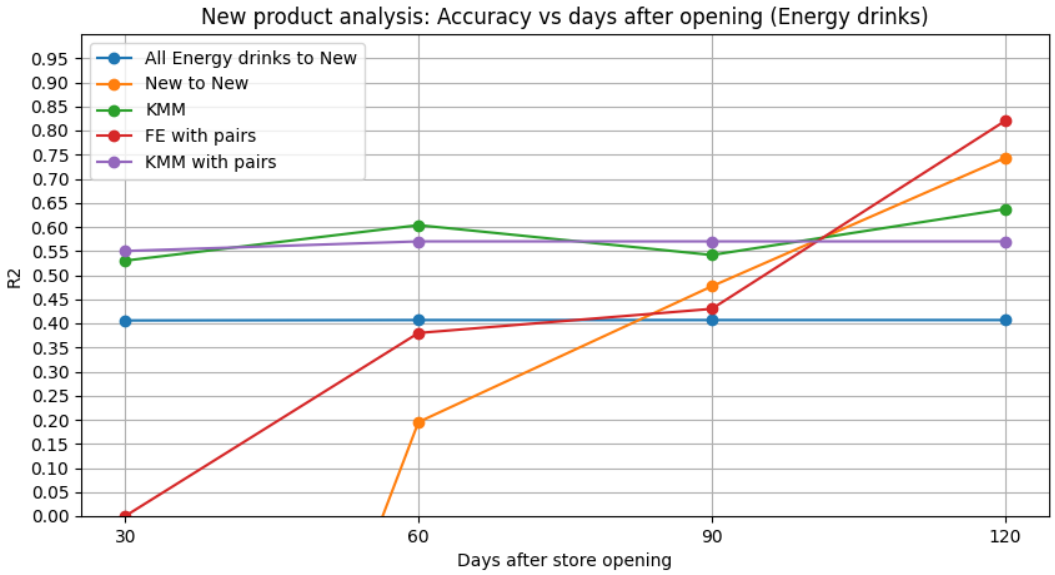
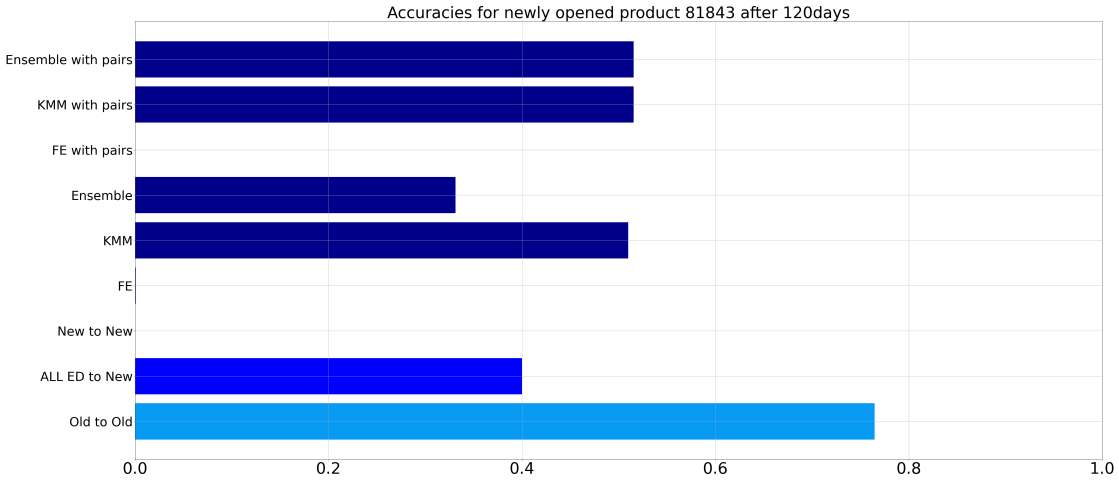
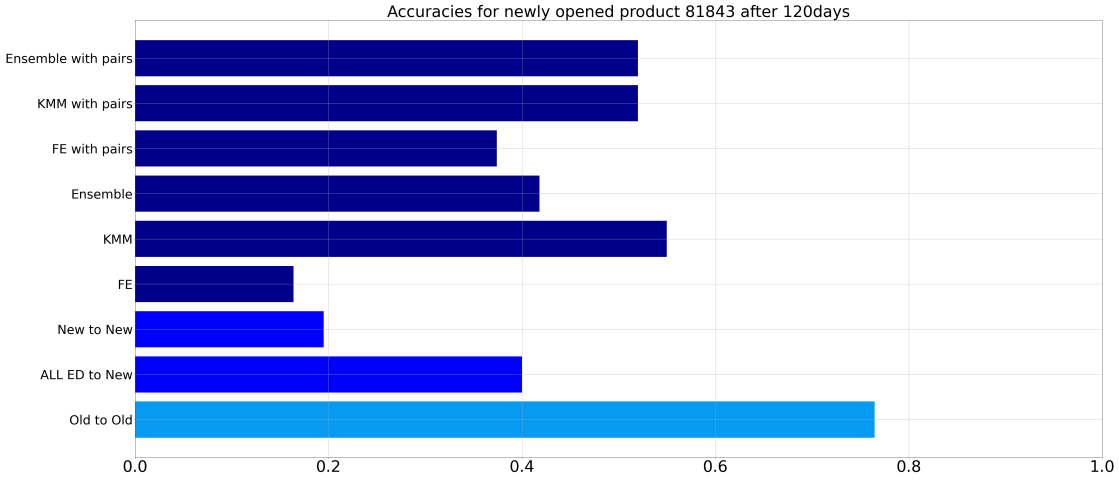


Figure 4.15: Comparison of the accuracy of different models as a function of the number of days after the new product introduction

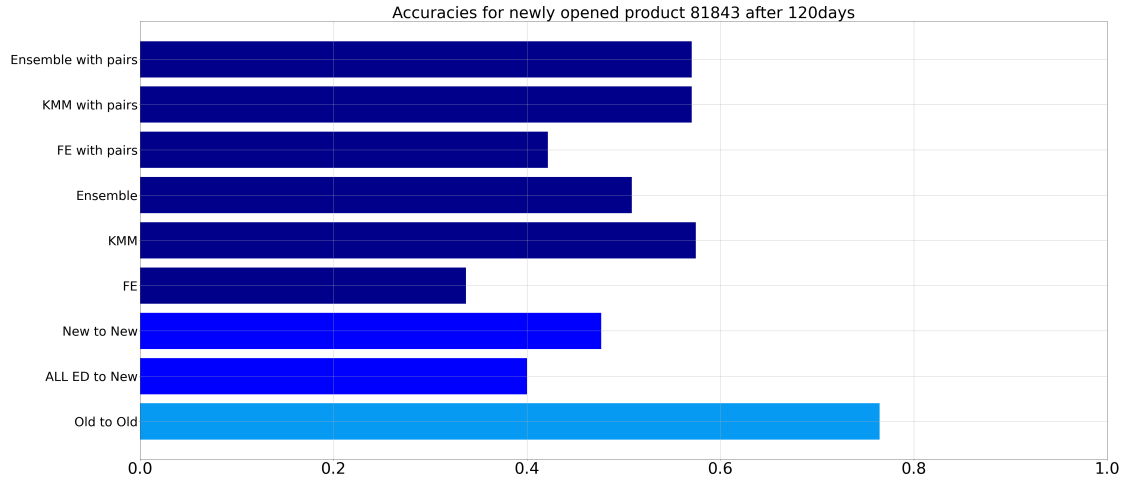
We expect that, as the number of days from the introduction of the new product increases and the data for the new product (new domain) becomes abundant, the accuracy of new-product models increases. **Figure 4.15** confirms this intuition. As shown in the figure, the prediction accuracy of the new-product model (brown line) increases with the number of days after the new product is introduced. As shown in **Figure 4.15**, domain adaptation allows us to improve the prediction accuracy of new-product models (the blue and orange lines in the graph). The accuracy levels of KMM and KMM with pairs (green and purple lines) are higher than the new product models from the beginning to 120 days. **Figure 4.16** reports the accuracy levels of the different models in greater detail.


Accuracies for newly opened product 81843 after 120days
Ensemble with pairs
KMM with pairs
FE with pairs
Ensemble
KMM
FE
New to New
ALL ED to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0

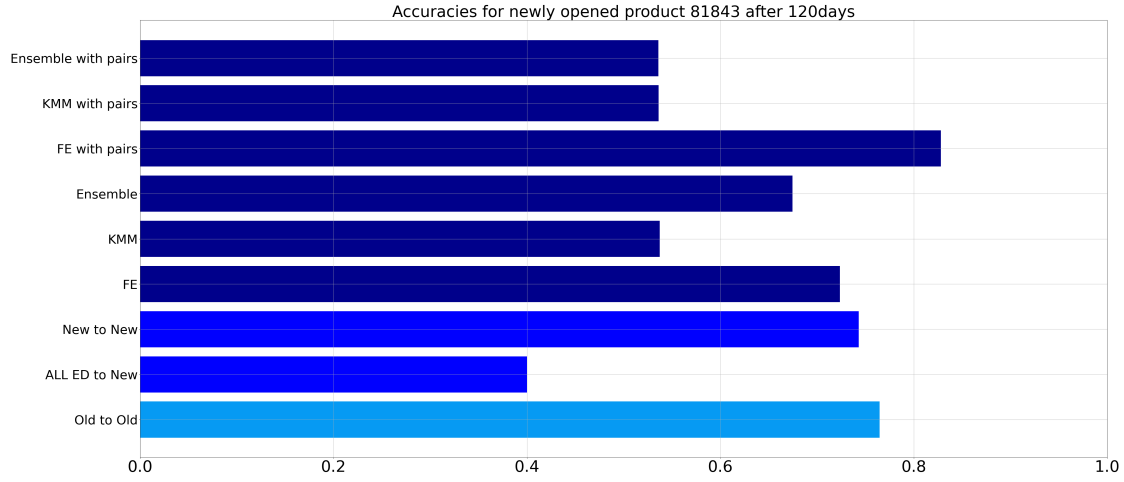
(a)


Accuracies for newly opened product 81843 after 120days
Ensemble with pairs
KMM with pairs
FE with pairs
Ensemble
KMM
FE
New to New
ALL ED to New
Old to Old
0.0
0.2
0.4
0.6
0.8
1.0

(b)



(c)



(d)

Figure 4.16: Comparison of the accuracy of different models when introducing a new product in the energy drinks category (XGBoost is considered as the base forecasting method), a) 10 days after opening, b) 20 days after opening, c) 30 days after opening, d) 60 days after opening, e) 90 days after opening, f) 120 days after opening, g) 150 days after opening, h) 180 days after opening.

Comparing **Figure 4.10**, **Figure 4.12**, and **Figure 4.15**, we conclude that in the actual new store the accuracy of domain adaptation ramps up as the days pass after the new store opening. However, for the case of simulated new store, the accuracy of domain adapta-

tion is more steady. This different behaviour can be due to the fact that in the simulated store case study, the domain shift between the data in the simulated new store and old stores is smaller. The simulated new store is in fact an old store with a hypothetical start date. So, the shifts that can be expected when we actually open a new store are not present in this case study.

4.5 Managerial Implications and Discussion

In this thesis, we analyzed three examples of potential domain shifts in retail: the outbreak of the COVID-19 pandemic, opening a new store, and the introduction of a new product. We conducted each analysis for both the coffee category and the energy drinks category, while considering XGBoost and Transformers as the base forecasting method (the results based on Transformers as the base forecasting method were omitted in the last two case studies for conciseness). For the case studies of opening a new store and the introduction of a new product, we first performed the analysis using a simulated new store and a simulated new product. After confirming our intuition, we repeated these two analyses with an actual new store and an actual new product in our dataset. Our industry partner has helped us identify a recent new store opening and a new product introduction.

In all three case studies, domain adaptation methods along with the pairing technique have allowed us to significantly improve the prediction accuracy of the models in the new domain. The pairing technique helps improve the prediction accuracy by up to 20% for the COVID-19 case study using Transformers.

As expected, as the time window in the new domain expands, the accuracy of new-domain models increases and can ultimately reach a satisfactory value. However, we showed that this value of the time window may often be equal to a couple of months. In fact, in some cases, even after 120 days, the new domain accuracy still remains lower than the previous-domain accuracy.

We have shown that since domain adaptation techniques can improve new-domain accuracy levels, they eliminate the need to wait for a long time before reaching a satisfactory demand prediction accuracy. In other words, by using domain-adaptation techniques, the prediction accuracy in the new domain improves. This accuracy can sometimes be even higher than the old-domain accuracy, albeit this is not guaranteed.

Chapter 5

Conclusion

In this thesis, we showed that when a domain shift occurs in our data domain, the accuracy of the demand prediction model in the new domain can suffer. In this context, we studied three examples of domain shifts in the field of retail: the outbreak of the COVID-19 pandemic, opening a new store, and introducing a new product. We tested three domain adaptation techniques to help mitigate the adverse effects of such domain shifts: Frustratingly Easy domain adaptation (FE), Kernel Mean Matching (KMM), and an ensemble of the previous two methods.

In all three case studies, domain adaptation methods allowed us to improve the prediction accuracy of models in the new domain. As expected, as the time window in the new domain expands, the accuracy of the new-domain models increases and can ultimately reach an acceptable accuracy level (often close to the model accuracy in the old domain). However, we found that this process can sometimes take several months. We have shown that since domain adaptation techniques improve the new-domain prediction accuracy, they also eliminate the need to wait for a long time before reaching a satisfactory prediction accuracy. In addition, using the pairing technique resulted in further improvements in the prediction accuracy of models in the new domain.

One limitation of this work is that although coffee is a very high-selling and stable product in our dataset, a new coffee product was not introduced. So, we did not have the chance to extend our analysis to a new product introduction in the coffee category, and we kept our analyses and experiments at the simulation stage. One possible use-case would be to extend this approach to data from a coffee shop that frequently introduces new coffee products. Another limitation of our approach is that although we expected Transformers to outperform XGBoost as the base forecasting method due to their success in the field of computer vision and natural language processing, it was not the case. Our data may not be a good representation of all possible sales data in the field of retail, and Transformers might perform better than XGBoost on a different demand prediction task. So, one possible direction for future improvement could be to test our approach on alternative datasets. An additional direction for future research could be to predict the demand of the source domain and use this additional data to enhance the results.

We should mention that in this study we focus on products individually due to simplicity. However, extending our study to multivariate time-series prediction would be a next step.

Overall, further research is needed to improve upon our results by testing alternative domain adaptation techniques and validating our results in the context of different retail business settings.

Bibliography

[Aburto and Weber, 2007] Aburto, L. and Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing*, 7(1):136–144.

[Adulyasak et al., 2020] Adulyasak, Y., Benomar, O., Chaouachi, A., Cohen, M. C., and Khern-am nuai, W. (2020). Data analytics to detect panic buying and improve products distribution amid pandemic. *Available at SSRN 4035703*.

[Afrin et al., 2018] Afrin, K., Nepal, B., and Monplaisir, L. (2018). A data-driven framework to new product demand prediction: Integrating product differentiation and transfer learning approach. *Expert Systems with Applications*, 108:246–257.

[Ali et al., 2009] Ali, Ö. G., Sayın, S., Van Woensel, T., and Fransoo, J. (2009). Sku demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348.

[Alon et al., 2001] Alon, I., Qi, M., and Sadowski, R. J. (2001). Forecasting aggregate retail sales: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services*, 8(3):147–156.

[Andersen et al., 2005] Andersen, T. G., Bollerslev, T., Christoffersen, P., and Diebold, F. X. (2005). *Volatility forecasting*. National Bureau of Economic Research Cambridge, Mass., USA.

[Ball and Brunner, 2010] Ball, N. M. and Brunner, R. J. (2010). Data mining and machine learning in astronomy. *International Journal of Modern Physics D*, 19(07):1049–1106.

[Blitzer et al., 2007] Blitzer, J., Dredze, M., and Pereira, F. (2007). Domain adaptation for sentiment classification. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 440–447.

[Borovykh et al., 2017] Borovykh, A., Bohte, S., and Oosterlee, C. W. (2017). Conditional time series forecasting with convolutional neural networks. *ArXiv Preprint ArXiv:1703.04691*.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

[Cai et al., 2020] Cai, R., Chen, J., Li, Z., Chen, W., Zhang, K., Ye, J., Li, Z., Yang, X., and Zhang, Z. (2020). Time series domain adaptation via sparse associative structure alignment. *ArXiv Preprint ArXiv:2012.11797*.

[Chang et al., 2020] Chang, Y., Mathur, A., Isopoussu, A., Song, J., and Kawsar, F. (2020). A systematic study of unsupervised domain adaptation for robust human-activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–30.

[Che et al., 2018] Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):1–12.

[Chelba et al., 2007] Chelba, C., Silva, J., and Acero, A. (2007). Soft indexing of speech content for search in spoken documents. *Computer Speech & Language*, 21(3):458–478.

[Chen et al., 2015] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., et al. (2015). Xgboost: extreme gradient boosting. *R Package Version 0.4-2*, 1(4):1–4.

[Ciotti et al., 2020] Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W.-C., Wang, C.-B., and Bernardini, S. (2020). The covid-19 pandemic. *Critical Reviews in Clinical Laboratory Sciences*, 57(6):365–388.

[Cohen et al., 2022] Cohen, M. C., Gras, P.-E., Pentecoste, A., and Zhang, R. (2022). *Demand Prediction in Retail: A Practical Guide to Leverage Data and Predictive Analytics*. Springer.

[Cohen et al., 2020] Cohen, M. C., Gupta, S., Kalas, J. J., and Perakis, G. (2020). An efficient algorithm for dynamic pricing using a graphical representation. *Production and Operations Management*, 29(10):2326–2349.

[Cohen et al., 2021] Cohen, M. C., Kalas, J. J., and Perakis, G. (2021). Promotion optimization for multiple items in supermarkets. *Management Science*, 67(4):2340–2364.

[Cohen et al., 2017] Cohen, M. C., Leung, N.-H. Z., Panchamgam, K., Perakis, G., and Smith, A. (2017). The impact of linear optimization on promotion planning. *Operations Research*, 65(2):446–468.

[Cohen et al., 2019] Cohen, M. C., Zhang, R. P., and Jiao, K. (2019). Data aggregation and demand prediction. *SSRN 3411653*.

[da Costa et al., 2020] da Costa, P. R. d. O., Akçay, A., Zhang, Y., and Kaymak, U. (2020). Remaining useful lifetime prediction via deep domain adaptation. *Reliability Engineering & System Safety*, 195:106682.

[Dagum and Bianconcini, 2016] Dagum, E. B. and Bianconcini, S. (2016). *Seasonal adjustment methods and real time trend-cycle estimation*. Springer.

[Dai et al., 2007] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning*, pages 193–200.

[Daumé III, 2009] Daumé III, H. (2009). Frustratingly easy domain adaptation. *ArXiv Preprint ArXiv:0907.1815*.

[De Bruijne, 2016] De Bruijne, M. (2016). Machine learning approaches in medical image analysis: From detection to diagnosis. *Medical Image Analysis*, 33:94–97.

[de Mathelin et al., 2021] de Mathelin, A., Deheeger, F., Richard, G., Mougeot, M., and Vayatis, N. (2021). Adapt: Awesome domain adaptation python toolbox. *ArXiv Preprint ArXiv:2107.03049*.

[Deng et al., 2009] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[Ekambaram et al., 2020] Ekambaram, V., Manglik, K., Mukherjee, S., Sajja, S. S. K., Dwivedi, S., and Raykar, V. (2020). Attention based multi-modal new product sales time-series forecasting. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3110–3118.

[FAO-Gustavsson et al., 2011] FAO-Gustavsson, J., Cederberg, C., and Sonesson, U. (2011). Fao-global food losses and food waste.

[Gardner Jr, 1985] Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28.

[Gasthaus et al., 2019] Gasthaus, J., Benidis, K., Wang, Y., Rangapuram, S. S., Salinas, D., Flunkert, V., and Januschowski, T. (2019). Probabilistic forecasting with spline quantile function rnns. In *The 22nd international conference on artificial intelligence and statistics*, pages 1901–1910.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[Graves et al., 2014] Graves, A., Wayne, G., and Danihelka, I. (2014). Neural turing machines. *ArXiv Preprint ArXiv:1410.5401*.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

[Holt, 2004] Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10.

[Huang et al., 2006] Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., and Smola, A. (2006). Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems*, 19.

[Huber and Stuckenschmidt, 2020] Huber, J. and Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, 36(4):1420–1438.

[Huyen,] Huyen, C. Data Distribution Shifts and Monitoring, year = 2022, url = https://huyenchip.com/2022/02/07/data-distribution-shifts-and-monitoring.html, urldate = 2022-02-08.

[Hyndman, 2018] Hyndman, R.J., . A. G. (2018). *Forecasting: principles and practice*. OTexts.

[Kim et al., 2021] Kim, J., Kim, J., and Choi, J. (2021). Sequential movie genre prediction using average transition probability with clustering. *Applied Sciences*, 11(24):11841.

[Kononenko, 2001] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109.

[Kouw and Loog, 2018] Kouw, W. M. and Loog, M. (2018). An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*.

[Krizhevsky et al., 2012a] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012a). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105.

[Krizhevsky et al., 2012b] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012b). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25.

[Lim and Zohren, 2021] Lim, B. and Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209.

[Lim et al., 2019] Lim, B., Zohren, S., and Roberts, S. (2019). Recurrent neural filters: Learning independent bayesian filtering steps for time series prediction. *ArXiv Preprint ArXiv:1901.08096*.

[Liu and Xue, 2021] Liu, Q. and Xue, H. (2021). Adversarial spectral kernel matching for unsupervised time series domain adaptation. In *IJCAI*, pages 2744–2750.

[Lu et al., 2015] Lu, J., Wu, D., Mao, M., Wang, W., and Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems*, 74:12–32.

[Ma and Fildes, 2021] Ma, S. and Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288(1):111–128.

[Mahmoudyan and Zeqiri, 2021] Mahmoudyan, M. and Zeqiri, A. (2021). Time series forecasting using neural networks minimizing food waste by forecasting demand in retail sales. Master's thesis.

[Makridakis et al., 2020] Makridakis, S., Spiliotis, E., and Assimakopoulos, V. (2020). The m4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74.

[Mudelsee, 2019] Mudelsee, M. (2019). Trend analysis of climate time series: A review of methods. *Earth-Science Reviews*, 190:310–322.

[Nagelkerke et al., 1991] Nagelkerke, N. J. et al. (1991). A note on a general definition of the coefficient of determination. *Biometrika*, 78(3):691–692.

[Oord et al., 2016] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *ArXiv Preprint ArXiv:1609.03499*.

[O'donovan et al., 2015] O'donovan, P., Leahy, K., Bruton, K., and O'Sullivan, D. T. (2015). Big data in manufacturing: a systematic mapping study. *Journal of Big Data*, 2(1):1–22.

[Ragab et al., 2022] Ragab, M., Eldele, E., Chen, Z., Wu, M., Kwoh, C.-K., and Li, X. (2022). Self-supervised autoregressive domain adaptation for time series data. *IEEE Transactions on Neural Networks and Learning Systems*.

[Rangapuram et al., 2018] Rangapuram, S. S., Seeger, M. W., Gasthaus, J., Stella, L., Wang, Y., and Januschowski, T. (2018). Deep state space models for time series forecasting. *Advances in Neural Information Processing Systems*, 31:7785–7794.

[Redko et al., 2020] Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. (2020). A survey on domain adaptation theory. *ArXiv Preprint ArXiv:2004.11829*.

[Salinas et al., 2017] Salinas, D., Flunkert, V., and Gasthaus, J. (2017). Deepar: Probabilistic forecasting with autoregressive recurrent networks. arxiv e-prints, art. *ArXiv Preprint ArXiv:1704.04110*.

[Steinberg and Colla, 2009] Steinberg, D. and Colla, P. (2009). Cart: classification and regression trees. *The Top Ten Algorithms in Data Mining*, 9:179.

[Sun et al., 2016] Sun, B., Feng, J., and Saenko, K. (2016). Return of frustratingly easy domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

[Sun and Saenko, 2016] Sun, B. and Saenko, K. (2016). Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Societys*, 58(1):267–288.

[Van Kasteren et al., 2010] Van Kasteren, T., Englebienne, G., and Kröse, B. J. (2010). Transferring knowledge of activity recognition across sensor networks. In *International Conference on Pervasive Computing*, pages 283–300. Springer.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

[Wang et al., 2012] Wang, J., Zhang, L., Zhang, D., and Li, K. (2012). An adaptive longitudinal driving assistance system based on driver characteristics. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):1–12.

[Wang et al., 2019] Wang, Y., Smola, A., Maddix, D., Gasthaus, J., Foster, D., and Januschowski, T. (2019). Deep factors for forecasting. In *International Conference on Machine Learning*, pages 6607–6617.

[Weisberg, 2005] Weisberg, S. (2005). *Applied linear regression*, volume 528. John Wiley & Sons.

[Wilson et al., 2020] Wilson, G., Doppa, J. R., and Cook, D. J. (2020). Multi-source deep domain adaptation with weak supervision for time-series sensor data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1768–1778.

[Winters, 1960a] Winters, P. R. (1960a). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342.

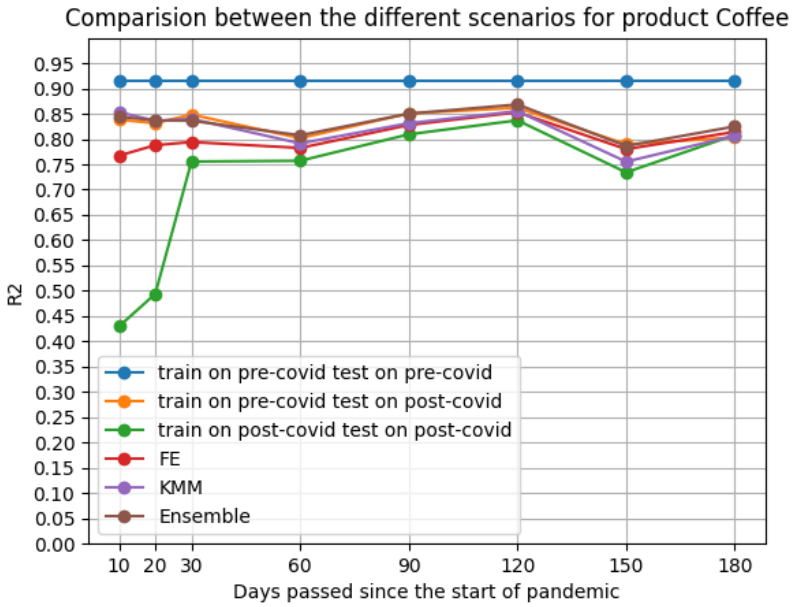
[Winters, 1960b] Winters, P. R. (1960b). Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342.

[Xu et al., 2021] Xu, T., Chen, W., Wang, P., Wang, F., Li, H., and Jin, R. (2021). Cdtrans: Cross-domain transformer for unsupervised domain adaptation. *ArXiv Preprint ArXiv:2109.06165*.

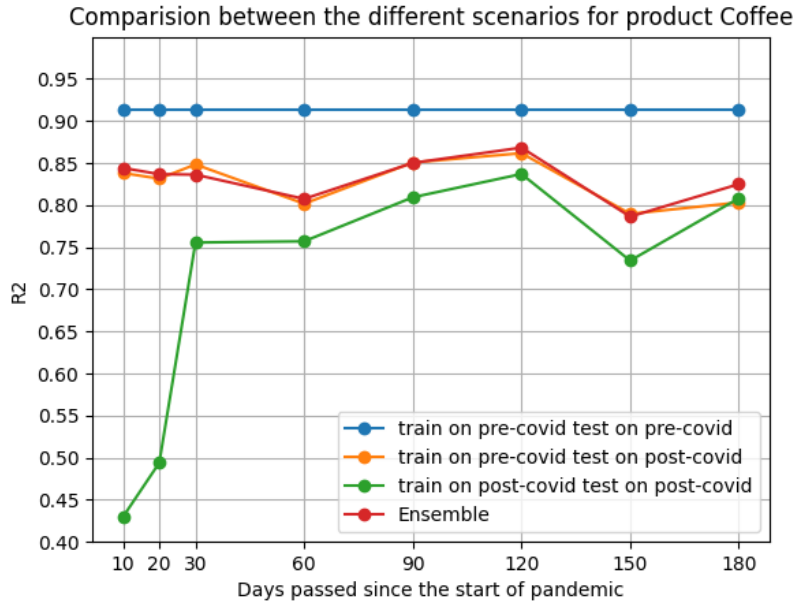
[Young et al., 2018] Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75.

Chapter 6

Appendix

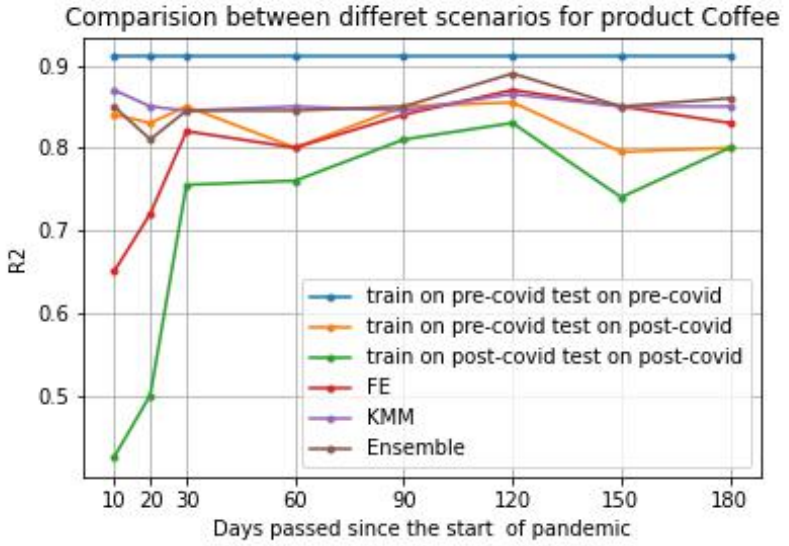


(a)



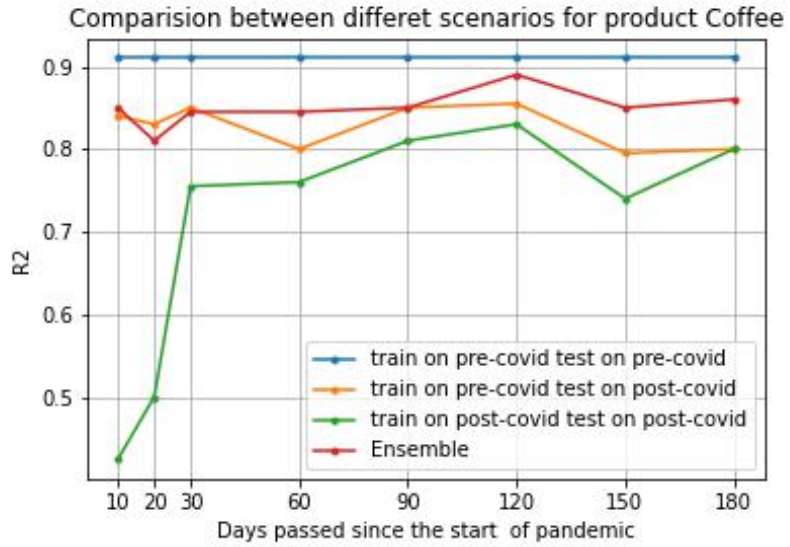
(b)

Figure 6.1: Comparison between different scenarios for the coffee category when using XGBoost as the base forecasting method. (a) original scale, (b) showing only the best performing domain adaptation model with axes being limited. As shown in part (b), the accuracy of the (green) post-pandemic model improves as more data become available in the new post-pandemic domain.



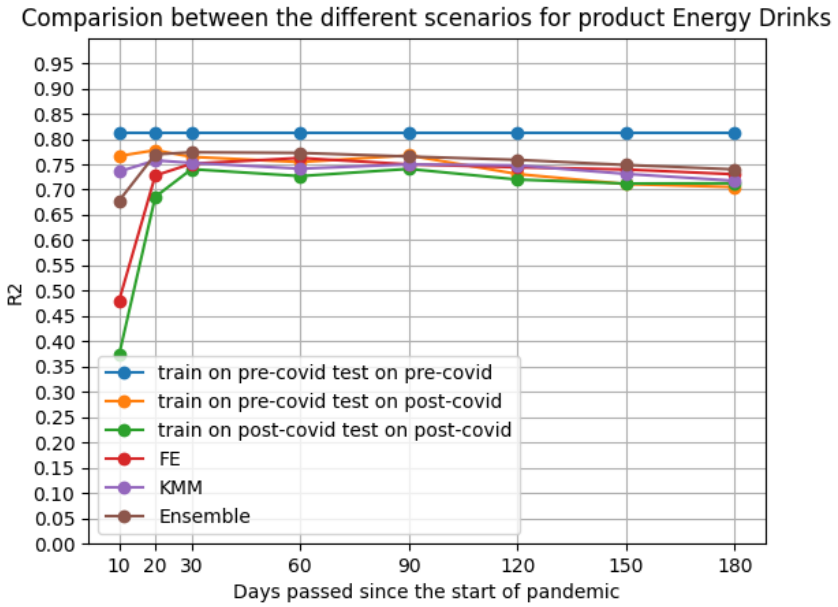
(a)

s

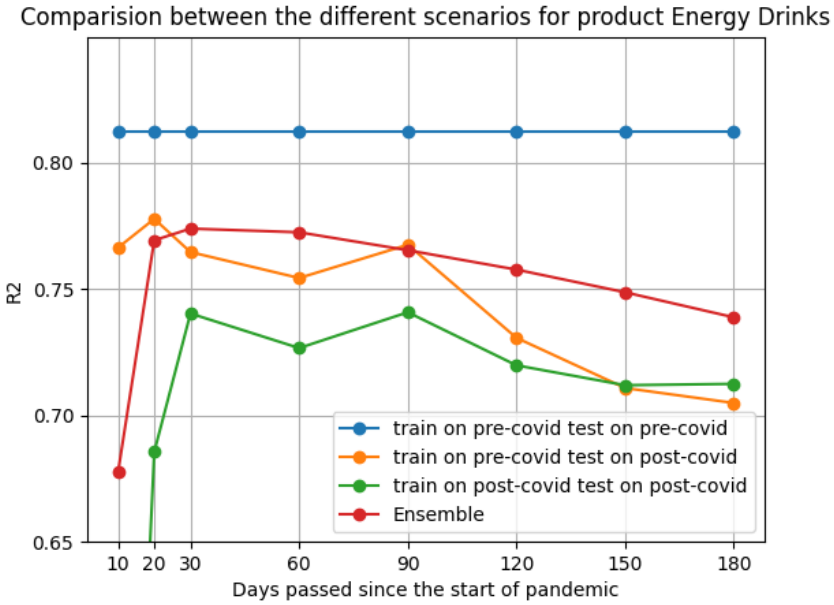


(b)

Figure 6.2: Comparison between different scenarios for the coffee category when using XGBoost as the base forecasting method and adding the pairing technique to the domain adaptation approaches. (a) original scale, (b) showing only the best performing domain adaptation model with axes being limited. As shown in part (b), the accuracy of the (green) post-pandemic model improves as more data become available in the new post-pandemic domain. When compared to Figure 6.1b, is is clear that using the pairing method allows to obtain a higher accuracy.

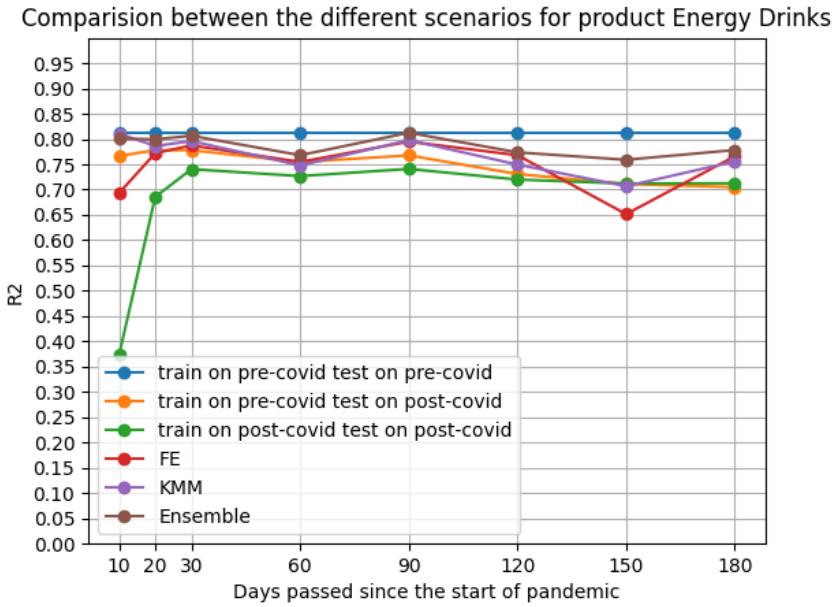


(a)

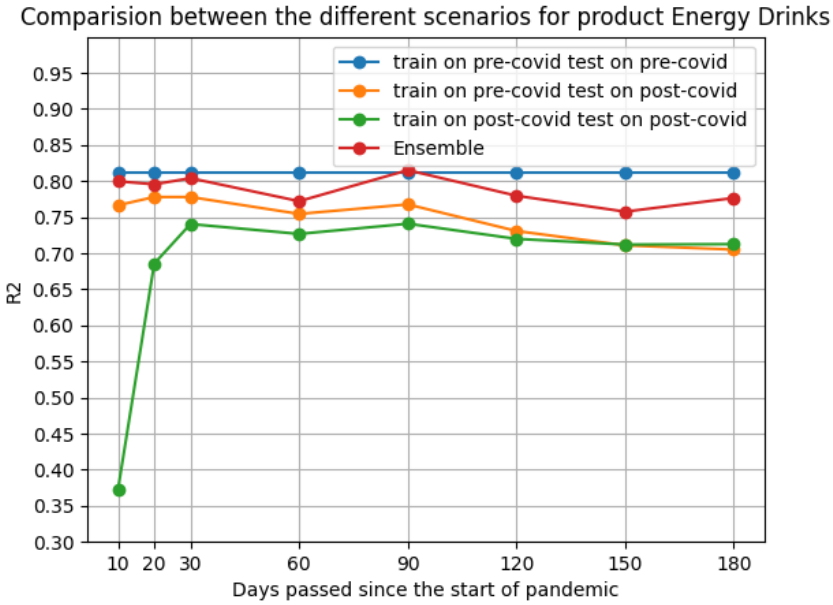


(b)

Figure 6.3: Comparison between different scenarios for the energy drinks category when using XGBoost as the base forecasting model. (a) original scale, (b) showing only the best performing domain adaptation model with axes being limited. As shown in part (b), the accuracy of the (green) post-pandemic model improves as more data become available in the new post-pandemic domain (the accuracy increases from 10 days to 30 days). By using domain adaptation (i.e., the red line), the accuracy of post-pandemic models improve mostly before 30 days and after 90 days.

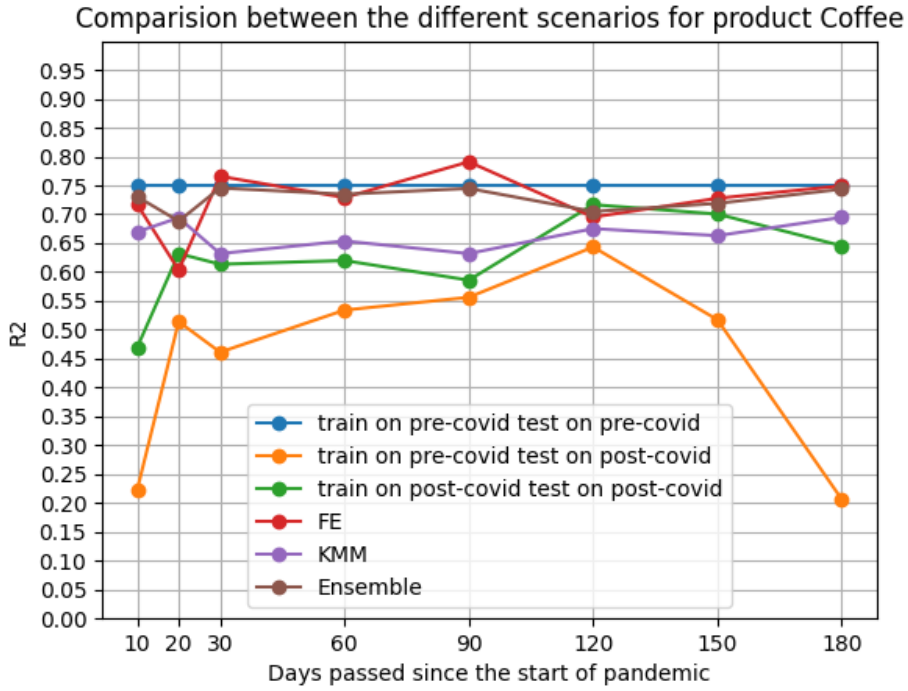


(a)

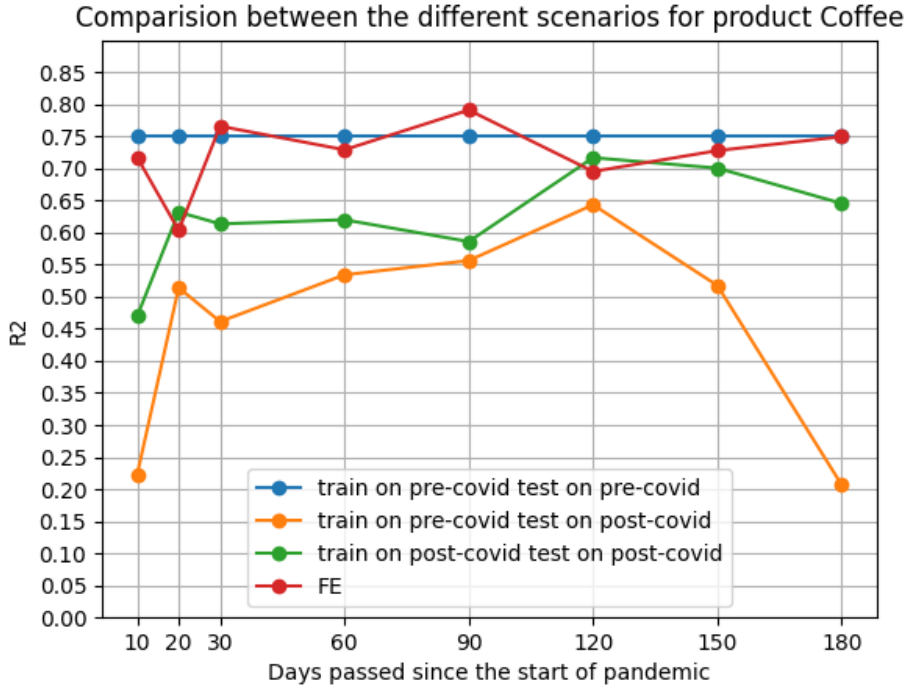


(b)

Figure 6.4: Comparison between different scenarios for the energy drinks category when using XGBoost as the base forecasting method and adding the pairing technique to the domain adaptation approaches. (a) original scale, (b) showing only the best performing domain adaptation model with axes being limited. As shown in part (b), the accuracy of the (green) post-pandemic model improves as more data become available in the post-pandemic domain (the accuracy increases from 10 days to 30 days). By using domain adaptation with the pairing technique (i.e., the red line), the accuracy of post-pandemic models improves more (especially before 30 days). When compared to Figure 6.4b, it is clear that using the pairing method allows us to obtain a higher accuracy.

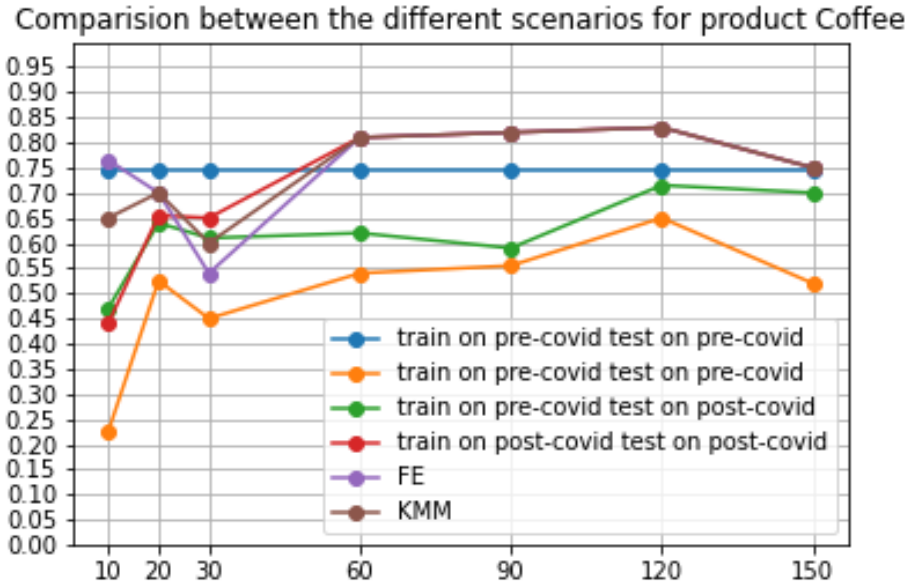


(a)

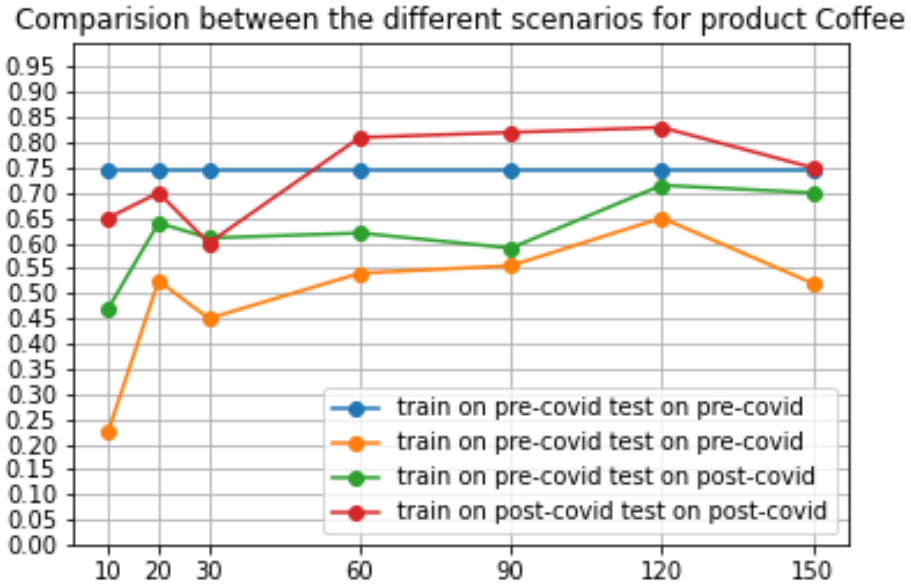


(b)

Figure 6.5: Comparison between different scenarios for the coffee category when using Transformers as the base forecasting method. (a) original scale, (b) showing only the best performing domain adaptation model with limited axes.

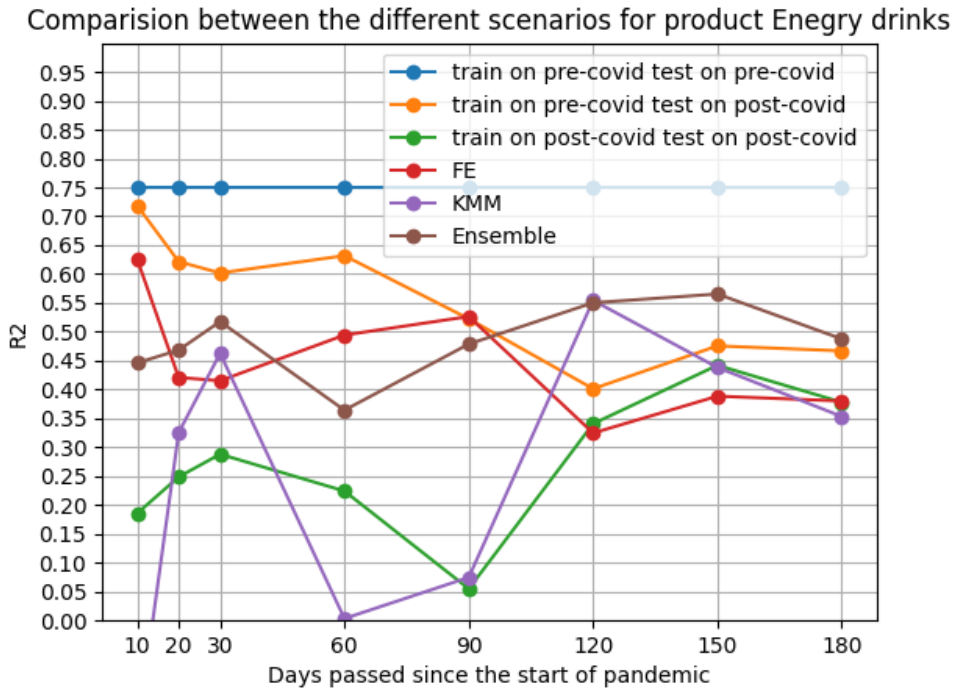


(a)

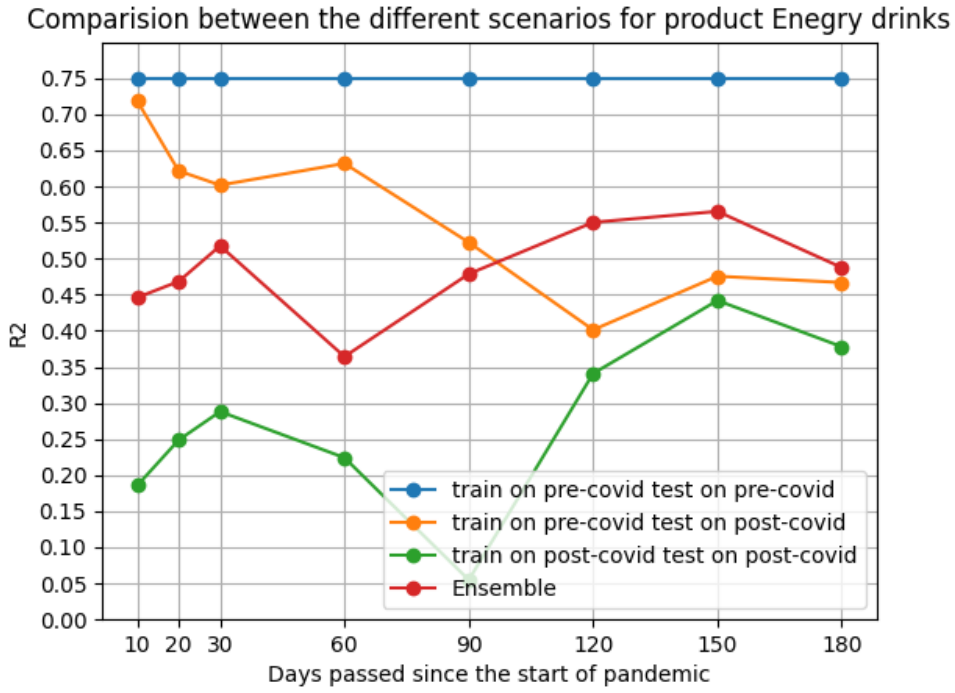


(b)

Figure 6.6: Comparison between different scenarios for the coffee category when using Transformers as the base forecasting method and adding the pairing technique to the domain adaptation approaches. (a) original scale, (b) showing only the best performing domain adaptation model with limited axes.

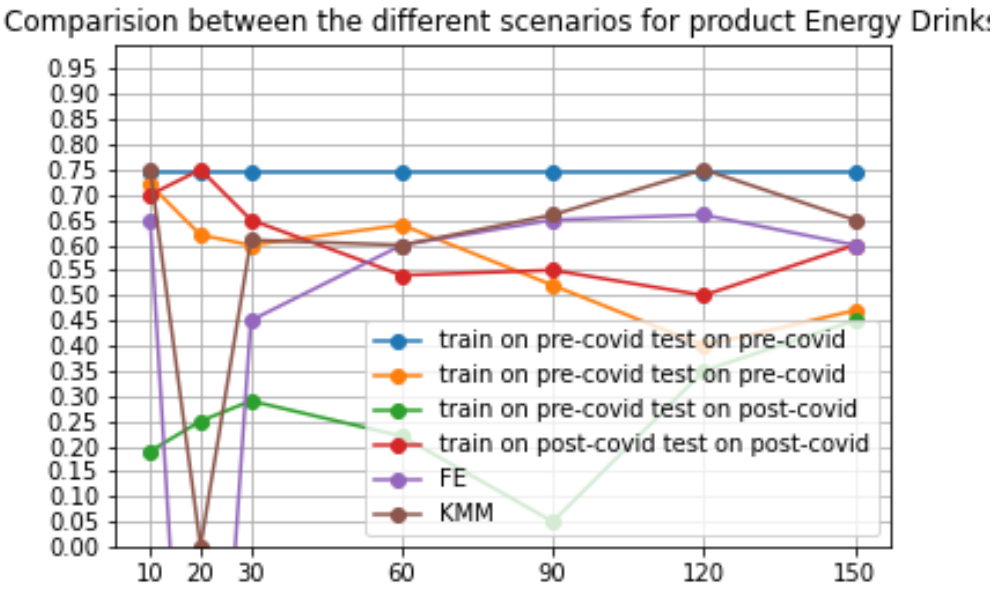


(a)

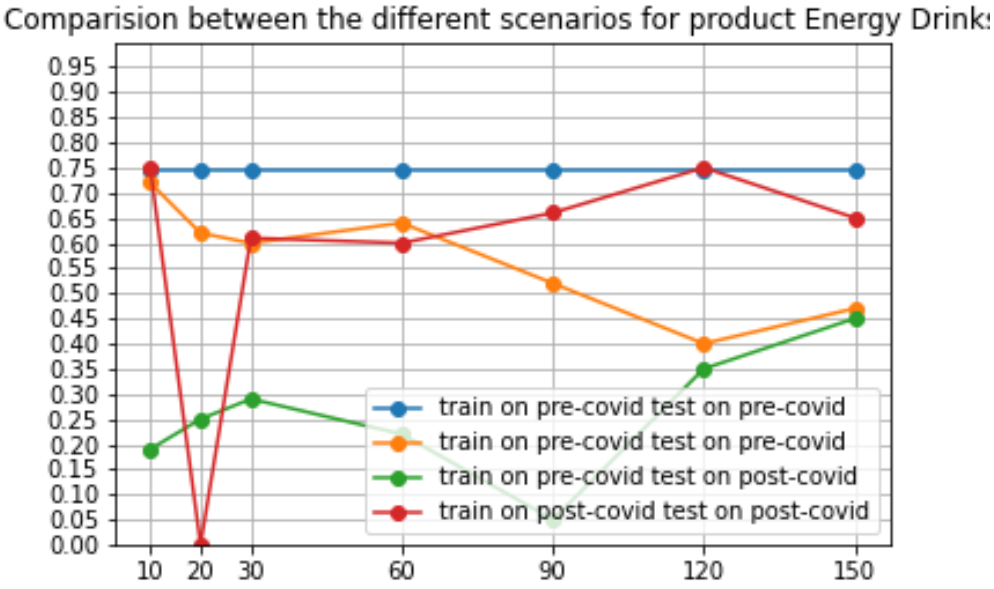


(b)

Figure 6.7: Comparison between different scenarios for the energy drinks category when using Transformers as the base forecasting method. (a) original scale, (b) showing only the best performing domain adaptation model with limited axes.



(a)



(b)

Figure 6.8: Comparison between different scenarios for the energy drinks category when using Transformers as the base forecasting method and adding the pairing technique to the domain adaptation approaches. (a) original scale, (b) showing only the best performing domain adaptation model with limited axes.