Forecasting own brand sales: Does incorporating competition help?

EI2019-35

10th October 2019

Wei Li, Dennis Fok, Philip Hans Franses

Abstract

This study aims to investigate how much value is added to traditional sales forecasting models in marketing by using modern techniques like factor models, Lasso, elastic net, random forest and boosting methods. A benchmark model uses only the focal brand's own information, while the other models include competitive sales and marketing activities in various ways. An Average Competitor Model (ACM) summarises all competitive information by averages. Factor-augmented models incorporate all or some competitive information by means of common factors. Lasso and elastic net models shrink the coefficient estimates of specific competing brands towards zero by adding a shrinkage penalty to the sum of squared residuals. Random forest averages many tree models obtained from bootstrapped samples. Boosting trees grow many small trees sequentially and then average over all the tree models to deliver forecasts. We use these methods to forecast sales of packaged goods one week ahead and compare their predictive performance. Our empirical results for 169 brands across 31 product categories show that the Lasso and elastic net are the safest methods to employ as they are better than the benchmark for most of the brands. The random forest method has better improvement for some of the brands.

Key words: Sales forecasting, high-dimensional data, principal components, factor model, Lasso, Elastic Net, random forest, boosting, data mining

1 Introduction

Forecasts of brand sales are relevant to both retailers and manufacturers. Forecasts give an impression of what future sales patterns can look like, and it helps to understand the

competition between brands. This can facilitate the brand level organization for the retailers and help the manufacturers to gear changes in the future marketing mix.

Brand sales forecasts are often generated from econometric time series models (Hanssens et al., 2003), where the well-known SCANPRO model (Wittink et al., 1988) is an illustrative example. Such models usually include past sales and own marketing activities (current and past), but frequently also variables concerning past competitor behaviour are included, at least if one knows this competition. Such variables can substantially improve the predictive performance. As retailers have the most complete information regarding to sales and promotions, in this paper we take a retailer's point of view and address various ways to include information on competitors for the prediction of within-store brand sales.

Our key conjecture is that in practice it is often not known which brands are effectively the main competitive brands. One may then resort to a couple of strategies. One option is to simply ignore competition. This makes the model simple to analyse, as there is no need for the sometimes cumbersome collection and preparation of data from competitors. A second strategy is to spend effort in studying which are the most relevant competitive brands. Data can be obtained, for example, by interviewing consumers or by analysing cross-promotion information. The latter approach can be rather successful, see Moon et al. (2007), Blattberg and Wisniewski (1989), Sethuraman et al. (1999), and Sethuraman and Srinivasan (2002), among others.

The third strategy, which we will address in the present paper, is to consider all other possible brands as potential competitors that might be relevant for the forecasts of the own brand. This approach is relevant if we do not know beforehand which brands have predictive content, and in this case we can let the data help to decide on this each time we make a forecast.

Naturally, this third strategy challenges the usual regression based forecast methodology. Common categories in FMCG markets can easily involve more than ten brands. When a typical SCANPRO-based regression model includes current and past sales as well as current and past marketing mix, then that amount of brands leads to the inclusion of more than one hundred variables. In this paper, we wish to address the question whether agnostically including all other brands and using modern data science technologies would lead to better forecasts of one's own brand sales.

A simple way to summarise competitor variables is to take weighted averages across all competitors, where the weights can be obtained from the brands' market shares. This method dramatically reduces the number of additional variables to be included in the forecast model. Other methods to exploit the rich information on competitors in a more refined way include

dimension reduction methods like Principal Components Analysis (PCA), shrinkage methods like the Least Absolute Shrinkage and Selection Operator (Lasso) and elastic-net, and tree-based methods like random forest and boosting. We will use a range of these methods and test them against a simple benchmark model that does not include any competitive information.

The dimension reduction idea to extract a small number of factors for use in prediction has been widely used in forecasting macro-economic time series like production and inflation, starting with the seminal work of Stock and Watson (1999; 2002). Our modeling strategy includes various specification options, including variable selection, variable grouping, the choice of the estimation window, the choice of the number of factors and of the lag structure in the sales model. The shrinkage methods shrink the estimated coefficients towards zero compared to the least squares estimates (James et al., 2013). Among different types of shrinkage methods the Lasso shrinks the coefficients of unimportant predictors exactly to zero and therefore performs variable selection. These shrinkage methods have recently gained popularity in forecasting sales (Ma et al., 2016; Sagaert et al., 2018) and macro-economic time series (Li and Chen, 2014; Medeiros and Vasconcelos, 2016; Smeekes and Wijler, 2018) due to their superior forecasting performances in a high-dimensional data environment.

Different from all linear (and log-linear) models, tree-based models segment data into groups using a decision tree format. Both random forest and the boosting tree method combine a large number of trees to generate usually more accurate forecasts than a single tree does. These tree-based methods are suitable to analysing complex non-linear relationships as they do not impose a particular structure on the data.

Our empirical test of the different methods concerns weekly data for brands in 31 categories, where we have a total of 169 brands. Our main conclusion, where we summarise across all cases and settings, is that although the own-brand-only benchmark model performs reasonably well, forecast accuracy can be improved for most brands using a certain way of including competitive information. Among all the methods we tested, the random forest method and the two shrinkage methods, that is, the Lasso and the elastic net, show the best forecast performance in terms of accuracy.

The outline of the paper is as follows. In Section 2, we describe methods to include large amounts of competitor variables, where we only focus on those methods that have shown to be most reliable in the available literature. In Section 3, we discuss the data that we use for our illustrations and we provide the details of our empirical methodology. Section 4 contains the forecast results comparison and section 5 draws the conclusion and discusses some potential future research areas.

2 Methods and models

In this section we discuss various ways of forecasting sales. We start with a straightforward linear time series model for log sales that will serve as the benchmark. Next we introduce models that account for competitive effects in different ways. Each of these models includes competition by summarizing or selecting from all the competing brands in some way.

To forecast the sales at time t one period ahead, that is, for time $t+1$, we use the available past information up to time t as well as the marketing efforts of all brands at time $t+1$. This forecasting situation is relevant for retailers, as in general they have full information on all (in-store) marketing efforts in the forecasted period.

In forecasting sales, both recursive expanding window and rolling window approach are popular. The former approach expands the estimation window period by period, when one more period of data is included, we re-run all the necessary procedures that result in the final out-of-sample forecast. The procedures vary across methods, may include variable selection, decomposition, determining the number of factors and the number of lags, choosing the tuning parameters, re-estimating the coefficients, and finally predicting with the updated coefficients. The rolling window approach, on the other hand, fixes the length of the estimation window, shifts the window forward by period, and re-run the forecasting steps each shift. The rolling window approach is more suitable for varying coefficient situations, while the expanding window is better for constant coefficient situations. We will implement both in our forecasting procedure and see which one is better for our extensive dataset.

2.1 Benchmark model

In the benchmark model we only use information of the focal brand.This model is an autoregressive model of order L with explanatory variables written as ARX(L) for the sales of the brand (in logarithms). As explanatory variables we take seasonal dummies and the own marketing efforts. There are 13 seasonal dummies, each covers four consecutive weeks. Denoting the sales of the focal brand at time t by s_t, the one-period-ahead forecasting model for sales after the natural log transformation is

$$\ln(s_{t+1}) = \alpha + M'_{t+1}\beta + D'_{t+1}\mu + \sum_{l=1}^{L} \gamma_l \ln s_{t+1-l} + \epsilon_{t+1}, \tag{1}$$

where M_{t+1} is a vector of marketing instruments and D_{t+1} is a vector of seasonal dummies, both for time $t+1$. In general M_{t+1} will contain the brand's price (in natural logarithms) and display and feature variables. By including lagged sales, the model captures dynamic

effects like stockpiling and purchase inertia. The number of lags L is chosen by minimising the Bayesian information criterion (BIC).

2.2 Average competitor model (ACM)

One very parsimonious way to include information of all competing brands is to summarise these competing brands in a few variables. In the average competitor model, we summarise this information by taking a weighted average of all competitive marketing variables (denoted by $\bar{M}_t^c$). The weights are given by the current market shares in week t, which vary per week. The competitive sales are also summarised by taking their weighted average (denoted by $\bar{s}_t^c$). These average competitive variables capture possible cross effects on sales. The average competitor model with L lags for own sales and Q lags for averaged competitor sales reads as

$$\ln(s_{t+1}) = \alpha + M'_{t+1}\beta + D'_{t+1}\mu + \sum_{l=1}^{L} \gamma_l \ln(s_{t+1-l}) + \bar{M}^{c\prime}_{t+1}\beta^c + \sum_{q=1}^{Q} \gamma_q^c \ln(\bar{s}^c_{t+1-q}) + \epsilon_{t+1}. \quad (2)$$

2.3 Principal component regression

The average competitor model summarises the competition by taking a market-share weighted average. Another weighting scheme is obtained by principle component analysis (PCA), where the weights are chosen to maximise the retained variance and can be used to find the "optimal" linear combination(s) of competitive variables. The "optimal" here means that the constructed linear combinations explain as much variance as possible of the competitive variables using a less number of components - usually much less than the number of original predictors. The obtained principal components can be added to the benchmark model to obtain a model that accounts for competition. We label this model as the Principal Components Regression (PCR) model. The number of components can be set to a fixed number or chosen data-drivenly.

Forecasting by means of principal components has proven to be very effective in macroeconomics, see for example the review chapters in Stock and Watson (2006) and Stock and Watson (2012). In our setting, the competitor information is first summarised by a number of principal components, and then these components and their lags together with the variables in the benchmark model are used to forecast sales of the focal brand.

Let N_t be a K-dimensional vector of competitive variables consisting of the information that is available at time t on all competitors for the variables price (in natural logarithms),

display, feature, and sales (in natural logarithms). In our retailing setting, when at time t we forecast $\ln(s_{t+1})$, the prices and promotion variables at $t+1$ are already known, whereas sales are of course not yet known for time $t+1$. Therefore, N_t consists of mixed information for time t (sales) and $t+1$ (price and promotion). If the start of the observation period is denoted by $t=1$, the competitor information that is available at time t is collected in the $t \times K$ matrix $N = (\ N_1, \ \ldots, \ N_t)'$, where each column (variable) of N is standardized to have mean zero and variance one. The leading r principal components (with $r < k$) of this matrix are collected in the $t \times r$ matrix $F = (F_1, \ldots, F_t)'$, where F_t is the vector of r principal components at time t. The principle component regression (PCR) is now given by

$$\ln(s_{t+1}) = \alpha + M'_{t+1}\beta + D'_{t+1}\mu + \sum_{l=1}^{L} \gamma_l \ln(s_{t+1-l}) + \sum_{g=0}^{G} F'_{t-g}\lambda_g + \epsilon_{t+1}. \tag{3}$$

In our retailing application, the competitive variables can be classified into four groups: prices, feature variables, display variables, and sales. To exploit this grouped variable structure, we can also perform PCA separately per group. Grouped PCA has the advantage of yielding meaningful factors like a competitive price factor and a competitive sales factor. These factors and their lags can then be used in the forecast equation (3), and we label this method as Grouped PCR.

2.4 Forecast-oriented factor construction

The PCA factors capture maximal variance of the predictor variables irrespective of the target variable that is to be forecasted. It may help to construct the factors in a way that reflects the final forecasting objective. Possible options are to select predictors according to their relation with the target variable or to derive factor weights from the correlation of the predictors with the target variable.

We consider two variable selection methods, known as "hard" and "soft" thresholding as introduced by Bai and Ng (2008), to which we refer for technical details. These selection methods provide a ranking of the predictor variables according to their importance in predicting the target variable. The principal components are then constructed from the subset of variables that are found to be most important in this sense. The resulting (subset) principal components are then used in the forecast equation (3).

2.4.1 Hard thresholding

In hard thresholding, the importance of the k-th predictor variable $N_{k,t}$ is assessed by means of its t-value in the following regression equation, which controls for the predictors of the benchmark model

$$\ln(s_{t+1}) = \alpha + M'_{t+1}\beta + D'_{t+1}\mu + \sum_{l=1}^{L} \gamma_l \ln(s_{t+1-l}) + N_{k,t}\beta_k^c + \epsilon_{t+1}, \tag{4}$$

where $N_{k,t}$ denotes the information at time t of the k-th predictor variable, that is, the sales in period t or the price or promotion activity in period $t+1$ of a specific competitor. This regression is applied for each of the variables $k = 1, \ldots, K$ separately, and the variables are ranked according to their absolute t-value. A potential drawback is that this method may select similar variables, as each predictor is evaluated separately without considering the other predictors. While this disadvantage does not exist in the soft thresholding method, which we introduce next.

2.4.2 Soft thresholding

The soft thresholding method selects variables sequentially such that each next variable adds most information for the target variable after controlling for the previously selected variables. The soft thresholding method (Bai and Ng, 2008) selects variables using least-angle regression (LARS) (Efron et al., 2004) and determines the optimal number of variables using an information criterion, such as the BIC. After the subset $\tilde{M}^c$ is selected, the forecasting procedure is exactly the same as above, that is r factors are extracted using PCA and these factors are used in (3).

LARS is a forward stepwise regression proposed by Efron et al. (2004). At the first step it selects the variable that correlates most with the target variable. Next the coefficient of the first selected variable is set to zero. Starting from zero, the coefficient of the variable is moved towards its least squares value. This way the correlation between the variable and the residual moves towards zero. When the correlation between the variable and the residual equals the correlation between the residual and a second variable, this second variable is added to the "active set". In this second step, the coefficients of the two variables are moved together in a way that their correlations with the evolving residual are tied and moved towards zero. As soon as a third variable "catches up" in terms of the correlation with the residual, the third variable enters the set. The whole process stops when all the variables are included in the model and at this point we get the common least squares fit. We use LARS

on the residuals of the benchmark equation (1) of regressing the focal brand sales on its own marketing instruments and own lagged sales. This leads to a ranking of all the competitor variables and then the actual number of selected variables is determined by the BIC.

2.5 Shrinkage methods

Different from information summarising methods, shrinkage methods fit a model that contains all the J predictor variables but impose a penalty term onto the least squares estimation to constrain the size of the coefficient estimates, therefore shrink the estimated values towards zero. The shrinkage penalty term can take different forms, the most commonly used are L1 norm (the Lasso), L2 norm (ridge regression), and a linear combination of both L1 and L2 norm (the elastic net). As the ridge estimation shrinks all the coefficients towards zero proportionally, it will not exclude any predictor. We would like to select important predicting variables especially from all the competitors and therefore we will use the Lasso and the elastic net methods in the paper.

The Lasso regression proposed by Tibshirani (1996) has become very popular over the last decades for the purpose of dealing with high dimensional data, that is the number of variables is relatively large to the number of observations. By adding L1 norm penalty to the sum of squared errors, the Lasso shrinks the coefficients of unimportant variables exactly to zero, and therefore performs variable selection. Our regression model now can be written concisely as

$$\ln\left(s_{t+1}\right) = \alpha + \sum_{j=1}^{J} \beta_j x_{jt} + \epsilon_{t+1}, \tag{5}$$

where x_{jt}, $j = 1, ..., J$ represents all the J candidate predictors from the forecasted brand and all the competing brands, together with the seasonal dummies. The information available at time t includes own and competitors' sales at t and their lags, own and competitors' price and promotion variables at $t+1$, and seasonal dummies at $t+1$.

The penalised least squares estimate is obtained by minimising

$$\sum_{t=1}^{T} \left(\ln s_{t+1} - \alpha - \sum_{j=1}^{J} \beta_j x_{jt} \right)^2 + \lambda \sum_{j=1}^{J} \mid \beta_j \mid . \tag{6}$$

Here the tuning parameter λ is to adjust how strong the penalty is. A very large λ will generate all zero estimates. The parameter λ is usually chosen via cross-validation, which is a data driven method that tries to minimise out-of-sample squared prediction errors.

The Lasso is useful to identify which competitor(s) and what type of promotion(s) are

important in forecasting the focal brand sales. However, when there are highly correlated predictors, the Lasso will randomly choose one from the correlated group. The elastic net solves this problem by adding L2 penalty to (6).

Zou and Hastie (2005) proposed the elastic net regression that minimises

$$\sum_{t=1}^{T}\left(\ln s_{t+1}-\alpha-\sum_{j=1}^{J}\beta_j x_{jt}\right)^2+\lambda\sum_{j=1}^{J}\left(\gamma\mid\beta_j\mid+(1-\gamma)\beta_j^2\right). \tag{7}$$

This is a compromise of the Lasso and the ridge regression and can encourage grouping of highly correlated predictors. Both tuning parameters λ and γ here can be done via cross-validation. The details of how to do cross-validation can be found in James et al. (2013).

2.6 Tree-based models

Unlike traditional regression analysis, tree-based methods impose no specific structure on the data. Instead, they use a series of splitting rules to form decision trees. In the end the data will be partitioned into different groups and for every observation that falls in the same group, the group mean or mode of the dependent variable would be its predicted value. Regression tree analysis uses a procedure called recursive binary splitting to construct the rules. Starting from the top of the tree, that is, all the observations are in one group, the method considers all the predictors and all the possible values of each predictor as the cutpoint to split the data into two groups. The selected split is the one that minimises the residual sum of squares (RSS). The same process repeats for each of the two partitioned groups to split the data further. The binary splitting continues until a pre-specified stopping criterion is satisfied. The final groups produced are called terminal nodes. We refer to Breiman et al. (1984) for more details.

Regression trees are easy to interpret as it mimics human decision-making process. In addition, it is good at picking up the nonlinearity in the data and potential complex interactions between independent variables. For instance, the effect of own price on sales may be not (log) linear but there is a cut point in price, below which it will lead to a boom in sales. According to James et al. (2013), one major problem with trees is their instability due to the high variance, that is, small changes in the data can produce very different splits, therefore the output tree can be highly divergent or unstable. So their prediction accuracy is typically not as good as traditional regression methods. However methods like bagging, random forest and boosting generate multiple trees and aggregate the outcomes to make predictions. These methods reduce the variance significantly and hence improve accuracy.

2.6.1 Random forests

The random forest method is proposed by Breiman (2001) on the basis of bagging (Breiman, 1996). Bagging first bootstraps (sample with replacement) from data and generates multiple bootstrapped samples, then fits a tree to each of the samples, and finally averages the predictions of all the decision trees. Random forest is different from bagging on the part of building trees. Instead of including all the predictors for growing each tree, the random forest method randomly chooses a subset from all the predictors for consideration at each split in a tree. This way it can avoid generating very similar trees as does the bagging method and consequently reduces the correlation between them. Averaging many uncorrelated trees reduces the variance further compared to averaging many correlated ones (Hastie et al., 2001). For the detailed procedure of random forest see Appendix A.1.

2.6.2 Boosting

Boosting is a method that can be applied to many statistical methods, here we use it on decision trees. Instead of fitting one large tree to the data set, boosting grows many small trees sequentially in a way that each tree is grown using the residuals from a previous tree. Then the new decision tree is multiplied with a shrinkage parameter and then added into the fitted function to update the residuals. This approach allows us to gradually fit the data and slowly improve the fitted function in areas where it does not fit well. The procedure is described more specifically in Appendix A.2.

Different from bagging and random forest, which construct relatively big trees for each bootstrap sample, boosting fits a small tree each time. This means that the number of splits of each tree takes a small value. According to James et al. (2013) one to four splits often work well. There are three meta-parameters to choose here, and these are the shrinkage parameter, the number of splits, and the total number of trees. We will run trial regressions to choose among the different combinations.

A summary of all the models is presented in Table 1. Since PCR, Hard thresholding, and Soft thresholding use two ways to determine the number of factors included, there are actually 13 methods in total that we consider for our data.

3 Data

We apply the forecasting methods in Section 2 on data of one store gathered from the Information Resources Inc. (IRI) data set (Bronnenberg et al., 2008), which contains information

Table 1: Methods of incorporating competitor information

Name	Description	Equation	Meta-parameters
Benchmark model	An autoregressive model with own marketing mix variables	(1)	Number of own lagged sales L
Average competitor model (ACM)	Adding competitor variables to the benchmark model by taking a market-share weighted average of all competitors' sales and marketing variables	(2)	Number of own and competitors lagged sals L, Q
Principal component regression (PCR)	Summarizing all competitive variables by a number of factors and adding the factors and their lags to the benchmark model	(3)	Number of own lagged sales L, number of factors, number of lagged factors G
Grouped PCR	Summarizing the four competitive variable groups (price, feature, display, and sales) with one factor for each group and adding the factors and their lags to the benchmark model	(3)	Number of own lagged sales L, number of lagged factors G
Hard thresholding	Constructing factors on a subset of variables that are selected by the hard thresholding rule	(4) & (3)	Number of own lagged sales L, number of factors, number of lagged factors G
Soft thresholding	Constructing factors on a subset of variables that are selected by the soft thresholding rule	(4) & (3)	Number of own lagged sales L, number of factors, number of lagged factors G
Lasso	Least squares with a penalty of L1 norm	(5) & (6)	Tuning parameter λ
Elastic net	Least squares with a penalty of a linear combination of L1 and L2 norm	(5) & (7)	Tuning parameters λ and α
Random forest	Averaging tree models grown on multiple bootstrap samples, at each split of a tree, considering a random subset of all the predictors	(8)	Number of trees
Boosting tree	Growing trees sequentially in a way that each tree is grown using the residuals from a previous tree, then averaging over all the trees	(9)	Shrinkage parameter λ, number of splits in each tree, number of trees

of 31 consumer product categories[1] spanning over five years. The store we use is randomly chosen from medium-sized ones, having five years data, having at least two brands in each category, and having sufficient variation in price. Some of the IRI categories are defined in a broad way and thus contain different types of products. We split these broad categories into narrower ones and for each category we aggregate stock keeping units (SKUs) data to the brand level by weights. For a more detailed explanation of how the data is compiled we refer to Horváth and Fok (2013). Finally, we have 31 product categories and 169 brands and the number of brands in each category varies from 2 to 10.

To choose between the recursive expanding window and the rolling window approach in forecasting, we estimate the benchmark model, the ACM and all the factor models using both methods and compare the forecasting accuracies. For the recursive window estimation we started from using data over the first 108 weeks, among which the first 4 weeks are used to obtain the lags of sales up to 4 periods thus not used in the parameter estimation. The parameters are obtained from data over 104 weeks[2] and are then used to forecast the sales of one week ahead, that is week 109. Then the real data of week 109 are included in the sample and we re-estimate the model. The new parameters are in turn used to forecast the sales of week 110. This window expanding re-estimation process continues till the sales of the last week of the five year period is forecasted. For the rolling window forecast we estimate the models each time over a window of 104 weeks and forecast one week ahead. All the forecasted natural logarithms of sales are added by one-half of the mean squared prediction error and then exponentiated back into sales units (Ma et al., 2016; Cooper et al., 1999).

Due to the fact that there are some brands whose display and feature variables may not have enough variation in an estimation window, this may cause a problem for least squares based method including the shrinkage methods. We set a threshold of at least five distinct values for a display/feature variable to be included in an estimation. But this is not necessary for tree-based methods. So for the random forest and boosting methods, we include all the display/feature variables for all the estimation windows. Having few or even no variation will

[1]The categories are: beer, blades, carbonated beverages, cigarettes, coffee, cold cereal, deodorant, diapers, facial tissue, frozen dinners/entrees, frozen pizza, household cleaners, hot dog, laundry detergent, margarine/butter, mayonnaise, milk, mustard & ketchup, paper towels, peanut butter, razors, salty snacks, shampoo, soup, spaghetti/Italian sauce, sugar substitutes, toilet tissue, toothbrushes, toothpaste, and yogurt. We exclude category photo from the original data set.

[2]To choose the window width we tried the width of 1.5 years, 2 years, and 2.5 years in rolling window forecasts for all the 169 bands across 31 categories. The results show that there are much more brands performing better with 2 year window width than with 1.5 year window width (103 brands out of 169). While the number of brands performing better with 2.5 year window width compared with 2 year window width is not significant (87 out of 169). Therefore we choose 2 years as our rolling window width and as our starting window width of the recursive forecasting.

not influence the estimation process.

4 Forecasting procedure and accuracy evaluation

4.1 Forecasting accuracy evaluation

Forecasting accuracy is measured by Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE), defined as

$$RMSE = \sqrt{\frac{1}{T}\sum_{t}\left(y_t - \hat{y_t}\right)^2}, \tag{8}$$

$$MAE = \frac{1}{T}\sum_{t}|y_t - \hat{y_t}|, \tag{9}$$

and

$$MAPE = \frac{1}{T}\sum_{t}\frac{|y_t - \hat{y_t}|}{y_t}, \tag{10}$$

where T is the number of the out-of-sample forecasts. A smaller RMSE, MAE, or MAPE means a more accurate forecast. Here RMSE and MAE are used to select models. MAPE is useful to compare cross categories and understand how much the forecasts relatively deviate from the real values as it is not scale dependent.

To compare how much better or worse a competing model is relative to the benchmark , we use relative MAE (RelMAE)

$$RelMAE = \frac{MAE^{comp}}{MAE^{bench}}, \tag{11}$$

which measures the MAE of the competing model relative to the benchmark model, a value smaller than 1 means an improvement of the forecast accuracy.

To compare all the models on their overall performance, Davydenko and Fildes (2013) proposed a measure called Average Relative MAE (ARMAE), which can be obtained from

$$ARMAE^{comp} = \left(\prod_{b=1}^{N}\frac{MAE_b^{comp}}{MAE_b^{bench}}\right)^{\frac{1}{N}}. \tag{12}$$

This measures the MAE of the competing model relative to the benchmark model for each brand b and then takes a geometric mean over all the N brands. Similar to RelMAE,

an ARMAE value smaller than 1 means an improvement over the benchmark model.

4.2 Forecasting procedure and selection of meta-parameters

All our models involve lagged predictors, for model (1) it is lagged own sales, for model (2) it is lagged own sales and lagged average competitor sales, and for model (3) it is lagged own sales and lagged factors. We include up to four lags for all the lagged variables and the actual number of lags is determined by the BIC for each model. Note that model (2) and (3) both have more than one lagged variable, and in this case the optimal number of lags of different variables within the same model can be different. Shrinkage methods and tree-based models will select useful predictors from all the candidates no matter whether they are lagged variables or not. So there is no need to use a information criterion for them.

For all the models that involve factors, except for Grouped PCR, we tried two ways of determining the number of factors, one is fixing the number at two and another is including up to four factors, and then choosing the number with best predictive performance. In the latter data-driven method, we forecast for models with either 1, 2, 3, or 4 factors respectively and then choose the number of factors resulting in the smallest RMSE over the most recent 26 weeks (half year) within each estimation window. Next, the model with the optimal number of factor lags is used for one-week-ahead out-of-sample prediction.

The implementation of the shrinkage methods involves choosing optimal tuning parameters. For both Lasso regression and elastic net, the tuning parameter λ is determined by five-fold cross-validation. Firstly, the data in the estimation window is randomly divided into five parts. Then we leave one part out and obtain results for a grid of 100 λ-values using the remaining four parts. Next we forecast over the left out part. This process continues until we obtain the "out-of-sample" forecasts for all the five parts. Then the RMSE over the whole estimation window is calculated and the λ that results in the smallest RMSE is chosen. Finally the chosen λ is used in fitting on the whole estimation window and forecasting one week ahead out of the estimation window. Strictly speaking, cross-validation on time series data is not correct because the temporal dependencies are interrupted when the data is randomly divided into parts. But it is still widely used as a heuristic, for example see Li and Chen (2014) and Ma et al. (2016). The tuning parameters γ in elastic net regression can be chosen data-drivenly as well, but to limit the number of models needed to be considered, we set $\gamma = 0.5$.

We implemented random forest using the randomForests package in R. As the number of bootstrapped samples B will not cause overfitting, we choose a value of B that is large enough for the prediction error to settle down. Our trial estimation results from 31 brands,

one from each category, show that the errors have settled down when B is larger than 100. In our final estimation we set B equal to 500.

We applied boosting tree model using the gbm package in R. As the number of trees Bneeds to be large enough for the errors to converge but not too large to overfit, we run a trial regression on one beer brand. We try four different numbers of split, and for each number of split, two different shrinkage parameters, so in total eight different combinations. An optimal combination in terms of convergence speed and out-of-sample RMSE is then chosen to for forecasting sales of all the brands. The details of this process can be found in Appendix A.3.

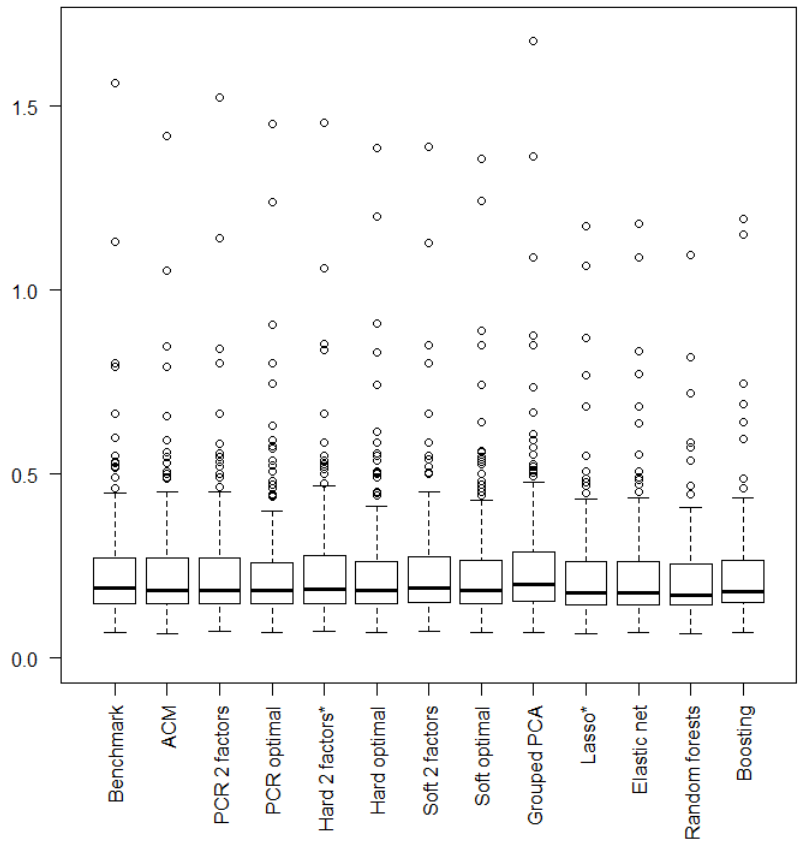
5 Results

For the benchmark model, the ACM, and all the factor models, we perform both expanding window and rolling window estimation and compare the forecast accuracies (measured with RMSE). The results from 169 brands show that the expanding window procedure is better for the majority of the brands. The number of brands for which expanding window estimation performs better varies from 107 (the benchmark model) to 132 (the ACM). This may due to the fact that in general there is no structural break in our data span. Hence increasing the number of observations improves the accuracy. All the results presented hereafter are therefore from expanding window estimations.

Since MAPE is not scale dependent, the MAPE values can be compared across categories and brands, which will give us an idea about how accurate the forecasts from different models are. Table 2 presents the comparison results with the left panel showing the mean, median, and the standard deviation of the MAPE values, the right panel showing the number of brands whose MAPEs are smaller than 0.1, 0.2, and 0.3 respectively. It can be seen that the random forest achieves the best result in terms of lowest mean, median, and standard deviation and the highest number of brands whose MAPE is smaller than 20% and 30%. Our benchmark forecasts made without competitor information have MAPE smaller than 30% for 135 (that is about 80%) of the brands. The median and the mean of the MAPEs are 18.9% and 24.1% respectively. The boxplot of the MAPEs from all the models are shown in Figure 1. It can been seen that the errors of random forest are the least dispersed. We also find that the outliers of all the models highly (but not completely) overlap, which means that some brand sales are just more difficult to forecast than others.

Next we compare the RMSEs and MAEs of all methods for each brand and present the results in Table 3. The first panel of the table shows the number of brands each method

Figure 1: Boxplot of MAPEs



* The Hard 2 factors model and the Lasso have one outlier each that is not shown in the plot, both are from brand "mayo3". The MAPE values are 73.17 and 5.67 respectively.

Table 2: Mean Absolute Prediction Error comparison of all the models

	MAPE			Number of brands whose MAPE		
Models	Mean	Median	SD	$< 10\%$	$< 20\%$	$< 30\%$
Benchmark	0.241	0.189	0.177	5	90	135
ACM	0.239	0.183	0.171	**7**	95	134
PCR 2 factors	0.241	0.185	0.176	6	94	134
PCR optimal	0.242	0.184	0.183	6	98	137
Hard 2 factors	0.674*	0.187	5.612*	5	94	132
Hard optimal	0.241	0.184	0.180	6	98	137
Soft 2 factors	0.253	0.189	0.226	4	90	132
Soft optimal	0.243	0.183	0.180	3	92	138
Grouped PCR	0.260	0.200	0.207	4	86	130
Lasso	0.259	0.178	0.447	5	101	140
Elastic net	0.229	0.176	0.160	5	100	140
Random Forest	**0.215**	**0.170**	**0.131**	4	**105**	**147**
Boosting	0.228	0.181	0.152	4	102	141

Note: The figures printed in bold show the best results in the column.

***** An outlier, brand "mayo3", has a MAPE value of 73.17, which leads to the high mean and standard deviation of the method.

performs the best in terms of achieving the lowest RMSE or MAE among all the models. The second panel presents the number of brands for which each method performs better than the benchmark. Finally the third panel shows the time cost in terms of minutes used to estimate the model and to forecast all the 169 brands. As can be seen in the table, the benchmark model only excels in 7 or 6 cases, which means that for all the other 162/163 brands, forecast accuracy can be improved by incorporating competition in some way. From the first panel we can see that the random forest model takes the lead with the most number of brands showing the best performance, 33 measured with RMSE and 45 with MAE. The elastic net comes the second with 23 (RMSE) and 27 (MAE) best performances. The Hard thresholding with optimal number of factors follows behind with 20 (RMSE) and 17 (MAE) best performances. However if we compare all the models with the benchmark, then the elastic net takes the lead with improved forecast accuracy for 117 (RMSE) and 130 (MAE) brands. The Lasso is nearly as good as the elastic net. This implies that in general, the Lasso and elastic net can improve forecasts over the benchmark for majority of the brands. The random forest method does not show such general good performance, however for some of the brands where that the other models do not perform well, random forest can improve forecast accuracy substantially. It can also be seen in the table that for models involving factors, the method using an optimal number of factors in general performs better than the one using fixed number of factors. It is also worth mentioning that the ACM delivers quite

good forecasts given how straightforward the idea (just averaging the competitors variables) behind it is.

From the results we conclude that the Lasso, elastic net and the random forest are all good options to incorporate competition information to improve forecast accuracy. The former two shrinkage methods perform better than the benchmark for most of the brands. The computation burden, measured in minutes used, of the shrinkage methods is comparable to that of the factor models, however their performances are better than the latter. As to the random forest , when it captures the underlying nonlinearity and interactions right, it can provide the most accurate forecasts. But the computation time is much longer than the shrinkage models. Nevertheless, nowadays parallel computing is easily available so the computation burden is not necessarily a problem.

Table 3: Number of brands each method achieving lowest or lower than the benchmark RMSE/MAE

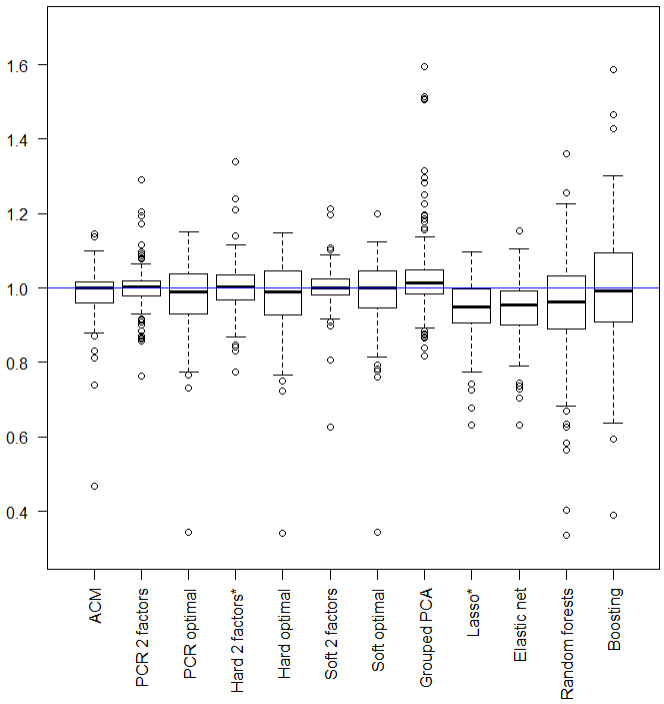
Model	Number of brands of each model performs BEST among all the models		Number of brands of each model performs BETTER than benchmark		Total time cost in minutes
	RMSE	MAE	RMSE	MAE	
Benchmark	7	6	-	-	**5**
ACM	12	12	97	88	16
PCR 2 factors	8	4	79	80	20
PCR optimal	18	17	94	92	61
Hard 2 factors	7	4	82	81	28
Hard optimal	20	17	93	91	67
Soft 2 factors	6	3	88	83	52
Soft optimal	9	9	82	84	92
Grouped PCR	1	0	69	60	18
Lasso	17	13	115	**130**	30
Elastic net	23	27	**117**	**130**	30
Random Forest	**33**	**45**	89	108	276
Boosting	8	12	79	91	2651
Total	169	169	169	169	-

Note: The figures printed in bold show the best results in the column.

To compare the three best performing models even further, that is, the Lasso, elastic net, and the random forest, we present the boxplot of their RelMAEs in Figure 2. Despite that the Lasso has one outlier "mayo3"[3], together with elastic net, it shows improvement (values

[3]The outlier value comes from an extremely high sales forecast at the early stage of the forecasting process.

Figure 2: Boxplot of Relative MAEs



* The Hard 2 factors and the Lasso both have one outlier "mayo3" with value of 316.45 and 3.20 respectively, which are not shown in the plot.

smaller than 1) for the most brands, although the magnitude of the improvements are less compare to the random forest model, the deterioration is less as well. It means that the two shrinkage methods are the safest method to incorporate all the competition variables for forecasting sales here, although they are not always the best.

We have 31 product categories and each category has different characteristics and competition environments. To look into the model performances within each category, we calculate the ARMAE (average relative MAE, see (12)) of the best three competing models relative to the benchmark and present the results in Table 4. The lowest ARMAE of each row is printed in bold and a value smaller than 1 means that the method is better than the benchmark on average in the category. It can be seen that the random forest performs the best in most of the categories and particularly well in *cigarettes, mayonaise*, and *razors,* where the improve-

Even when we exclude the variables with less than five distinct values, there are still cases when a predictor's value is within a certain range in the estimation window, but the value of the predictor at the prediction week is far out of the this range. This situation can be ruled out by increasing the length of the estimation window.

ments over the benchmark are much larger than those of the Lasso and elastic net. While in categories like *beer, cold cereal, hotdog, paper towel, soup*, and *toilet tissue*, the random forest performs no better than the benchmark model.

From all the results shown in Tables 2, 3 and 4 we can say that the Lasso and elastic net outperform all the competing models for most brands. If one wants to choose a method that can utilise the competitors information to improve sales forecasting performance, then the two shrinkage methods are the best choice. These methods are particularly good at picking up useful predictors from a large amount of variables. For this reason, they can identify what types of promotion from which competing brand affect the focal brand's sales the most. However, for products that do not promote much, for example razors, as we only include variables that have at least 5 distinct values[4], the valuable promotion information is excluded from the model at the early stage of forecasting. This information enters the model only when it has accumulated enough variation. However tree models do not require variables to have enough variation so they can include this type of information from the very beginning to contribute to forecasting. This is why the random forest model shows the least dispersed errors. On top of that, the random forest model mimics the process of aggregating much of people's decision making processes. For example, if a brand is on price promotion, some people would buy it straight away, some would like to see whether another brand they like is on promotion and some may also consider whether and how much they have already bought in the previous weeks. So the random forest can capture some underlying nonlinearities that linear models can hardly detect.

6 Conclusion and discussion

This study investigated whether incorporating competing brands information helps to improve sales forecasts. If yes, then how much value is added to the benchmark forecast model that excludes competitor variables? Furthermore, we compared different techniques of including competitors' sales and marketing activities.

The Average Competitor Model summarises all competitive information by weighted averages; Principal Component Regression (PCR) model summarises all competitive information by a number of factors and uses these factors and their lagged terms to forecast sales. The methods involve Hard and Soft thresholding constructing factors on a subset of variables that are selected by the hard and soft thresholding rule respectively. The method called Grouped

[4]As we aggregate the data to brand level, so the feature and display variables are measured in percentage of volume sold that is on promotion instead of dummies.

PCR summarises the four variable groups, namely price, feature, display, and sales, with one factor for each group and uses these factors and their lags for forecasting. Among these methods, PCR model, Hard thresholding and Soft thresholding require choosing the number of factors. For these three models we experiment with a fixed number of factors (2 factors) and with choosing an optimal number based on past forecasting performance. We perform one week out-of-sample forecasts on 169 brands from 31 product categories. The results show that the data-driven method outperformed the fixed number of factors for all three models.

The Lasso and elastic net put a constraint to the coefficients such that the most influential predictors will be selected. Random forest and the boosting method are based on tree models, which are suitable to fit complex nonlinearity in the data and are like human decision making processes.

Our forecasting results show that the benchmark model can be improved in its forecast accuracy, measured with RMSE, MAE, MAPE, and RelMAE, by incorporating competitive information for 162 brands out of the total 169. Among different alternative models the Lasso, elastic net and random forest show the best forecasting performance. For most brands, the Lasso and the elastic net are better than the benchmark and they are a safe choice in terms of forecast accuracy. Random forest on the other hand can improve the forecast accuracy substantially for some of the brands, especially when the other methods do not perform well. One possible explanation is that for those brands which are not often promoted, the two shrinkage methods and all the other linear models do not use the very infrequent promotion information well. On the other hand, the random forest method performs really well for these type of brands.

One could imagine if the shrinkage methods can incorporate infrequent promotion information wisely, then their forecast accuracy will improve considerately. A possible way to solve this is to include these variables regardless of their limited variation, but set an upper bound for insanely high forecasted sales, for example ten times previous mean sales. However the choosing of the bound can be tricky. Another way is to form a committee of the three best performing models: the Lasso, elastic net, and random forest, and find a way to take use of the advantages of each of them. We will leave these issues for future research.

A Appendix

A.1 Random forest

The random forest procedure is described as following:

1. Bootstrap number B of subsamples from the data.

2. For each bootstrapped sample, construct a big tree using recursive binary split, at each split consider only a random sample of p predictors from the total J predictors, where $p \approx \sqrt{J}$.

3. Average over all B trees to obtain the out-of-sample prediction

$$\ln \hat{s}_{t+1} = \hat{f}(x_t) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}^b(x_t), \tag{13}$$

where f stands for the tree function, f^b is the b-th tree generated from the b-th bootstrapped sample.

A.2 Boosting tree

The procedure of boosting tree is described as following, where the fitted function is denoted by $\hat{f}$, and B is the total number of trees generated sequentially:

1. Set starting fitted values to be zero, $\hat{f}(x) = 0$, so the starting residuals are the target variable $r_t = \ln s_t$ for all t in the estimation set.

2. For $b = 1, 2, \ldots, B$, fit a tree $\hat{f}^b$ with d splits to the estimation set that uses residuals r as target variable.

3. Update $\hat{f}$ with $\hat{f} + \lambda \hat{f}^b$, where λ, the shrinkage parameter, is a very small number and so $\lambda \hat{f}^b$ is a shrunken version of $\hat{f}^b$ obtained in step 2.

4. Update residuals r_t with $r_t - \lambda \hat{f}^b(x_t)$.

5. After number B of trees are generated, average over all the models

$$\hat{\ln s} = \hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x). \tag{14}$$

For the tree based methods, it is not necessary to use the natural logarithm of sales. But to be consistent with all the other models, we still use the natural logarithm form.

A.3 Choosing meta-parameters of boosting tree

We try shrinkage parameter $\lambda = 0.01$ and 0.001, the number of splits of each tree or interaction depth $d = 1$ to 4. So there eight different models. The trial results from the eight models show that for $\lambda = 0.01$, the out-of-sample RMSE settled after 1000 to 2000 iterations depending on the d value, while for $\lambda = 0.001$, the error has not settled yet after 5000 trees. So we choose $\lambda = 0.01$ to save time of computation. The trial results also show that when interaction depth $d = 3$, which means the model allows up to 3-way interactions, the out-of-sample RMSE is the lowest compared to $d = 1$, 2, and 4. Therefore we use $d = 3$ and $\lambda = 0.01$ for all the brands sales forecasts. To avoid overfit due to too large number of iterations B, we set $B = 2500$ such that the errors have converged and then use five-fold cross-validation in each estimation window to choose optimal B.

References

Bai, J. and Ng, S. (2008). Forecasting economic time series using targeted predictors. *Journal of Econometrics*, 146:304–317.

Blattberg, R. C. and Wisniewski, K. J. (1989). Price-induced patterns of competition. *Marketing science*, 8(4):291–309.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

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

Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). Classification and regression trees. wadsworth int. *Group*, 37(15):237–251.

Bronnenberg, B. J., Kruger, M. W., and Mela, C. F. (2008). Database paper: The IRI marketing set. *Marketing Science*, 27(4):745–748.

Cooper, L. G., Baron, P., Levy, W., Swisher, M., and Gogos, P. (1999). "promocast" : A new forecasting method for promotion planning. *Marketing Science*, 18(3):301–316.

Davydenko, A. and Fildes, R. (2013). Measuring forecasting accuracy: The case of judgmental adjustments to sku-level demand forecasts. *International Journal of Forecasting*, 29(3):510–522.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004). Least angle regression. *Annals of Statistics*, 32(2):407–499.

Hanssens, D. M., Parsons, L. J., and Schultz, R. L. (2003). *Market response models: Econometric and time series analysis*, volume 12. Springer Science & Business Media.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

Horváth, C. and Fok, D. (2013). Moderating factors of immediate, gross, and net cross-brand effects of price promotions. *Marketing Science*, 32(1):127–152.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An introduction to statistical learning*, volume 112. Springer.

Li, J. and Chen, W. (2014). Forecasting macroeconomic time series: Lasso-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, 30(4):996–1015.

Ma, S., Fildes, R., and Huang, T. (2016). Demand forecasting with high dimensional data: The case of sku retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research*, 249(1):245–257.

Medeiros, M. C. and Vasconcelos, G. F. (2016). Forecasting macroeconomic variables in data-rich environments. *Economics Letters*, 138:50–52.

Moon, S., Kamakura, W. A., and Ledolter, J. (2007). Estimating promotion response when competitive promotions are unobservable. *Journal of Marketing Research*, 44(3):503–515.

Sagaert, Y. R., Aghezzaf, E.-H., Kourentzes, N., and Desmet, B. (2018). Tactical sales forecasting using a very large set of macroeconomic indicators. *European Journal of Operational Research*, 264(2):558–569.

Sethuraman, R. and Srinivasan, V. (2002). The asymmetric share effect: An empirical generalization on cross-price effects. *Journal of Marketing Research*, 39(August):379–386.

Sethuraman, R., Srinivasan, V., and Kim, D. (1999). Asymmetric and neighborhood cross-price effects: Some empirical generalizations. *Marketing Science*, 18(1):23–41.

Smeekes, S. and Wijler, E. (2018). Macroeconomic forecasting using penalized regression methods. *International journal of forecasting*, 34(3):408–430.

Stock, J. H. and Watson, M. W. (1999). Forecasting inflation. *Journal of Monetary Economics*, 44:293–335.

Stock, J. H. and Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business and Economic Statistics*, 20(2):147–162.

Stock, J. H. and Watson, M. W. (2006). Forecasting with many predictors. In *Handbook of Economic Forecasting*, pages 515–554. Elsevier.

Stock, J. H. and Watson, M. W. (2012). Dynamic factor models. In *The Oxford Handbook of Economic Forecasting*. Oxford University Press.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

Wittink, D. R., Addona, M. J., Hawkes, W. J., and Porter, J. C. (1988). Scan* pro: The estimation, validation and use of promotional effects based on scanner data. *Internal Paper, Cornell University*.

Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (tatistical methodology)*, 67(2):301–320.

Table 4: The models forecasting accuracy ARMAE in categories

	Lasso	Elastic net	Random forest
beer	**0.994**	0.989	1.117
blades	1.025	1.021	**0.973**
carbbev	0.945	0.938	**0.932**
carbbevSelect2	1.008	0.977	**0.968**
cigets	0.914	0.921	**0.795**
coffee	**0.875**	0.880	0.947
coldcer	0.976	**0.974**	1.004
deod	**0.938**	0.942	0.973
diapers	0.911	0.916	**0.849**
factiss	0.924	0.931	**0.885**
fzdinent	0.842	0.837	**0.797**
fzdinentSelect2	0.918	0.908	**0.885**
fzpizza	0.945	0.947	**0.917**
hhclean	0.958	0.961	**0.899**
hotdog	1.006	**1.002**	1.111
laundet	0.915	**0.906**	0.949
margbutrSelect2	**0.931**	0.935	0.952
mayo	2.959	1.504	**0.674**
mustketc	**0.888**	0.895	0.926
mustketcSelect2	0.942	0.936	**0.879**
paptowl	0.994	**0.993**	1.076
peanbutr	0.957	0.954	**0.895**
razors	0.777	0.772	**0.618**
saltsnck	0.979	**0.970**	0.994
shamp	0.929	0.929	**0.899**
soup	0.957	**0.955**	1.035
spagsauc	**0.955**	0.965	0.971
sugarsub	**0.935**	0.937	0.968
toitisu	0.964	**0.961**	1.019
toothbr	0.938	0.942	**0.920**
toothpa	**0.929**	0.932	**0.929**
Overall	0.962	0.949	**0.941**
Number of categories a model has best performance on average	8	7	17