Forecasting Customer Behaviour in Constrained E-Commerce Platforms

Eric Michael Smith, Alfredo Nantes, Andrew Hogue, Ilia Papas

Blue Apron, Inc., New York, NY. {eric.smith, alfredo.nantes}@blueapron.com

Keywords: Sales forecasting, customer behaviour forecasting, meal-kit services, multinomial logistic regression, random forests.

Abstract

Business-to-customer platforms have become a popular option for digital and physical product shopping, accounting for high portions of total retail sales. Over the last decade, considerable effort has been directed towards capturing customer preferences and choices for recommendation and sales prediction purposes in large domains like music and shopping. However, modelling the dynamics of customer behaviour in the case of choice-constrained spaces has remained largely unexplored. Yet, accurate behaviour forecasting for meal-kit-delivery services, one such choice-constrained platform, is crucial for optimal inventory planning and control and for reducing food waste.

In this work, we formulate customer behaviour within choice-constrained spaces as a multivariate multinomial problem, in order to capture the influence that multiple items may have on the final choice of customers. We will show how training our models using an autoregressive scheme enables predicting future user choices, and how this prediction can be corrected over time, using the data available from service-customer interaction. We compare the performance of logistic regressions and random forests in predicting the weekly purchase behavior of individual users, and we use these models to demonstrate high accuracy in predictions of the aggregate number of orders placed by all users each week.

1 Introduction

E-commerce growth has continuously outpaced the overall retail market over the past six years. As a result, online sales account for an increasingly large portion of total retail sales. It was projected that, in 2016, e-commerce generated an estimated 8.7% of all retail spending worldwide, with sales of about $1.9 trillion [9].

E-commerce platforms distinguish themselves from one another based on several factors, including their catalog of products as well as how and where those products are sourced. Within the family of e-commerce platforms, business-to-customer (B2C) companies sell their online goods to consumers, who are the end users of the companies' products or services. Such platforms differentiate, primarily, by the type of product being offered. Today, the majority of B2C sale volume is for physical goods offered by online marketplaces [8]. These types of B2C companies must deal with finding optimal solutions for storing and insuring their inventory, shipping items, and preventing item breakage [13]. Other B2Cs are more focused on digital goods like music, movies, audiobooks, videos and images. Compared to their more popular counterparts offering physical goods, businesses offering digital products have the advantage of not needing inventory storage or physical delivery, although they may face other issues like piracy, consumer rights legislation that varies across countries, and increased competition due to the increasing popularity of selling digital goods [6]. The third type of B2C company belongs to the category of online services like consulting, web design and development, and content writing and editing.

In recent years, a new category of retailers offering physical goods has emerged within the first B2C category above, focused on delivering perishable goods directly to consumers. These companies include online grocery shopping and meal-kit delivery. Meal-kit delivery retailers, such as Blue Apron[1] and Hello Fresh[2], offer weekly recipes along with pre-portioned ingredients and step-by-step instructions that allow customers to easily cook the meals themselves, thus lightening the burden of figuring out what to eat and needing to shop for groceries. Since these retailers' ingredients are pre-portioned, there is less risk that the purchased food will spoil and thus go to waste [21]. What distinguishes these retailers' services from other physical-product-based businesses is the perishable nature of the product being offered: because the ingredients will spoil quickly, businesses must consider the seasonal availability of their products, and they must work to offer innovative packaging, quick transport, and stable delivery times, in order to maximize freshness and avoid product spoilage.

Given the significant growth that these e-commerce platforms are experiencing [9], demand, sales, and inventory forecasting have become crucial components of such businesses. Precise forecasting can enable accurate predictions of future revenue, less wasted inventory through a stricter inventory control system, optimized staffing costs through earlier awareness of needed personnel, and better customer satisfaction [12]. As a side effect of the ever-growing popularity of these forecasting initiatives, businesses now have richer purchasing and activity logs, which give them increasingly detailed information about

[1] https://www.blueapron.com/
[2] https://www.hellofresh.com/

their customer preferences. As has been shown, these data can be effectively leveraged to build highly accurate machine learning models of consumers' taste and ordering behaviour [1, 23].

The importance of accurate sales forecasts on efficient inventory management has long been recognized. Early attempts to model time-series data on sales tried to capture the relationship between retail stocks and sales at the aggregate level [3] and study the variations in consumer demand [24]. Later approaches helped to overcome the difficulties intrinsic to linear models by investigating the use of neural networks [2, 5], regression splines [15], and the combination of neural networks with linear predictors [7, 14]. Predicting customer preferences on online e-commerce platforms in general has usually been addressed in isolation from the problem of predicting sales volume, perhaps due to the fact that unconstrained services offer a very large catalog of items to choose from. Modeling the dynamics of customer choices in such highly dimensional spaces is computationally hard.

On the other hand, meal-kit delivery services present customers with a much smaller set of items and, as a result, represent an excellent case study to try to identify both how preferences towards those items change over time and how customer choice is influenced by the items that are simultaneously offered at any given point in time. Using the example of Blue Apron as one of the leading businesses in this domain, we will introduce a formulation that is general enough to describe the prediction problem at hand. We will then show how to construct models based on our theoretical framework that can be used to effectively capture the behaviour of the customers down to the level of individual preferences over the items being offered. We will finally show that our models are accurate at predicting overall sales trends, as well as users' choices regarding individual items.

The rest of this paper is organized as follows: Section 2 introduces the specific prediction problem considered in this work. The theoretical framework is presented in Section 3, and its implementation is discussed in Section 4. In Section 5 we show how the predicted customer behaviour can be corrected with data available from service-customer interactions. Empirical results are presented in Section 6. Future research directions are discussed and proposed, and conclusions are presented, in Section 7.

2 Use Case: Prediction of Blue Apron Customers' Behaviour

Meal kit delivery services present their customers with a restricted menu of weekly recipes, each with pre-portioned ingredients. These recipes are decided in advance by the company, in order to ensure that most of the ingredients are in season, arrive fresh, and do not require considerable food preservation and refrigeration. This restriction of deciding recipes in advance constrains the type and amount of ingredients that are to be purchased, helps to reduce inventory volatility, and contributes to improved inventory planning and reduced food waste.

In the case of Blue Apron there is a minimum and maximum number of recipes that a user can choose from every week, and there is also a limit to the number of times that the user can choose the same recipe. (At the time of this paper's publication, Blue Apron customers cannot order more than one of the same recipe within a given week.) From the standpoint

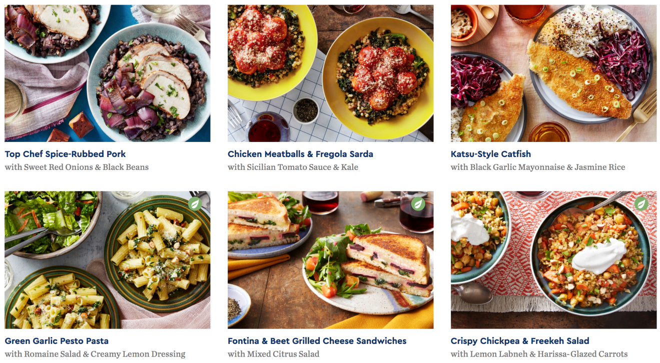

Figure 1. Example of Blue Apron weekly recipe selection.

of inventory and demand planning, it is crucial to accurately forecast both (1) the total volume of orders that will be placed a number of weeks ahead of delivery and (2) how many of each recipe will be ordered over each week. Additionally, it is useful from a culinary, purchasing and operations standpoint to gain insights about how changes in these forecasts result from changes in the purchasing behaviour of the customers. This helps to bundle recipes together in such a way as to maximize purchasing volume, and it allows for asserting whether behavioural changes are attributable purely to changes in customer preferences, business decisions or seasonal factors.

3 Theoretical Framework

Our problem can be formalized as follows: given the history X_i of purchasing behaviour and recipes offered for customer i, we wish to predict the menu configuration that customer i will choose on a given future week. Assuming there were N recipes available on any given week and that every recipe could be chosen at most M times, the combined order outcome of customer i can expressed as a set of M-dimensional random variables:

$$C_i = \{Y_1, Y_2, \ldots, Y_N\}_i \tag{1}$$

where $Y_{i,k} \in \{0, \ldots, M\}$ is a multinomial choice about recipe k with $M+1$ potential outcomes, indicating whether recipe k will be chosen $0, 1, \ldots, M$ times. Under this formulation, each customer can choose from $(M+1)^N$ possible alternatives, or configurations. Mapping from X_i to C_i involves finding the parameters of a multivariate multinomial distribution. When the customer can choose at most $M = 1$ recipes of each type, the outcome distribution reduces to a multivariate Bernoulli. In general, inferring whether customer i will choose recipe k a number of times j involves computing the conditional probability

$$\Pr(Y_{i,k} = j \mid Y_{i,l}\, \forall l \neq k, X_i) = \frac{\Omega_{i,k,j}}{\sum_{l=0}^{M} \Omega_{i,k,l}} \tag{2}$$

where $Y_{i,l}$ are choices over recipes other than $Y_{i,k}$. When using linear classifiers, the functions $\Omega_{i,k,j}$ are typically written as [4]:

$$\log(\Omega_{i,k,j}) = \alpha_{k,j} + \phi(X_i)\,\beta_{k,j} + \sum_{l \neq k} \varphi_{k,l,j,Y_{i,l}} \quad (3)$$

where $\alpha_{k,j}$, $\beta_{k,j}$ and φ are the parameters to be learned, and which can be used to interpret the prediction carried out by the algorithm. The φ are called association parameters and quantify the relationship between choices, e.g. how the choice of one recipe influences all other choices. These parameters are typically learned from the data. Finally, $\phi(\cdot)$ denotes a set of transformations applied to the input data in order to extract useful features. When non-linear classifiers such as artificial neural networks are used, the functions $\Omega_{i,k,l}$ become harder to interpret, because their behaviour is defined through a typically large number of parameters (i.e. weights) and interconnected computational units (i.e. neurons).

In order to minimize model complexity and maximize interpretability, we investigate the use of linear classifiers and ensemble-based classifiers. We will ignore all association parameters and assume that the outcomes $Y_{i,k}$ are all independent of one another, given the input features. This assumption effectively turns the $Y_{i,k}$ into standard multinomial random variables whose posterior distribution is easier to interpret. Henceforth, we shall use the term "choice model" to refer to these classifiers.

As discussed earlier, in order to ease inventory and demand planning tasks, it is important to be able to predict the total volume of orders over a number of future weeks. From our formulation, it follows that the expected number of recipes N_i that a customer i will order on a given week is given by

$$\mathrm{E}[N_i] = \sum_{k=1}^{N} \sum_{j=1}^{M} j \; \Pr(Y_{i,k} = j) \quad (4)$$

with $\Pr(Y_{i,k} = j)$ as defined as in Eq. (2). In this work, we determine the distribution of N_i directly from the data, using another classifier, which we shall call the volume model.

The rationale for using both volume and choice models is to easily control the granularity at which ordering behaviour is predicted and explained, namely, at the volume vs. recipe level. The volume model can also be used to capture restrictions that the business may impose on the total number of recipes that users can purchase in a given week. While these constraints could be learned by a multivariate multinomial model, they may not be captured sufficiently well by independent multinomial models that do not learn the joint distribution of customer choices.

From these considerations, it follows that the outcome predicted by the volume model may not coincide with the outcome inferred from the choice model. In order to ensure agreement between the two models, we scale the predicted distribution over recipe choices predicted by the choice model so that the sum of the probabilities over all recipes equals the volume-model prediction.

4 Predicting Order Dynamics through Choice and Volume Models

In the previous section, we explained how to learn a map from user and recipe data to the volume of orders and recipes. In this section we will show how the choices of customers at any future time can be predicted using the customer's history of past choices to date, together with contextual information available to the customers.

To that end, we use a simple autoregressive mechanism which utilizes the volume and choice models presented earlier. Our goal is to have the mechanism learn a map from the historical data, such as user choices and recipes offered to date, to future customer choice. Formally, such a mechanism involves finding a function f such that

$$C^{(t+1)} = f(\phi_h(C^{(0:t)}), \phi_c(e^{(t+1)})) \quad (5)$$

where $C^{(t+1)}$ indicates the choice of a customer at the next time step, i.e., next week, and $C^{(0:t)}$ denotes the history of choices up to time t. The functions $\phi_h(\cdot)$ and $\phi_c(\cdot)$ are sets of transformations that convert historical and contextual data, respectively, into feature vectors of fixed dimensionality; and $f(\cdot)$ denotes our volume or choice model presented earlier. Finally, $e^{(t+1)}$ are control signals available at time $t+1$ that are independent from past user choices, such as holidays and recipe attributes. Once the parameters of f are found, Eq. (5) can be used to predict user behaviour over an arbitrary number of future weeks, using the mechanism presented in Algorithm 1. After initializing the set $\widehat{\mathbf{C}}$ with the initial choice of the customers, e.g. up until the first week of prediction, the algorithm predicts P future weeks using an auto-regressive strategy, by iteratively calling the function f defined earlier. Accordingly, the predicted outcome is concatenated to the previous history of user choices and utilized at the next iteration to predict subsequent choices. As can be noted, this mechanism will produce

Algorithm 1 Monte Carlo simulation of choice dynamics

$\widehat{\mathbf{C}} \leftarrow$ user choice at time 0
for t in 1 through P **do**
    $e \leftarrow$ contextual information at time t
    $C \leftarrow f(\phi_h(\widehat{\mathbf{C}}), \phi_c(e))$
    $\widehat{\mathbf{C}} \leftarrow \text{concatenate}(\widehat{\mathbf{C}}, C)$

a sequence of predictions maintained by the set $\widehat{\mathbf{C}}$ that simulate the future choices of the customers, up to P time steps (i.e. weeks) in the future. In practice, given the probabilistic nature of the volume and choice models, each customer is simulated multiple times in order to collect enough data and approximate sufficiently well the statistics of the target outcome. Note the algorithm presented here can also be used to predict the future dynamics of newly acquired subscribers. This is easily achieved by extending the history of choices $\widehat{\mathbf{C}}$ with the choices of these new customers.

4.1 Choice and Volume Model Features

The features chosen for our volume and choice models fall into two main categories: user-related features and recipe-related features. User-related features capture whether the user is likely to order in a given week, regardless of what is on the menu on that week: these features are particularly useful because they learn whether a given user is one who orders most weeks, one who orders very infrequently, or one who has likely stopped ordering entirely. Our recipe-related features encode (1) whether a specific recipe has certain attributes (for instance, whether it has an ingredient such as salmon or asparagus, or a cuisine like Japanese), (2) how many other recipes on offer that week contain the same attributes, and (3) how likely users are to have ordered recipes with those attributes before. Together, these three categories of recipe-related features teach the model which users have had affinities for which types of recipes in the past, and they allow the model to learn which upcoming recipes users are likely to order in the future.

5 Prediction Correction

At Blue Apron, each customer has a set of orders associated with them. Each order contains information about the recipes that are scheduled to be shipped, if any, to the customers on a specific week, as well as other information such as delivery day of the week. We define the state of an order as the specific configuration $C^{(t)}$ that an order takes on a given week t, as defined as in Eq. (1). This state can be changed by the customer an unlimited number of times before cutoff time, which is the time at which no more changes are allowed and the order starts being prepared for shipment. The order state information can be used to correct the prediction made by the auto-regressive models presented earlier.

Recall that the objective of the volume and choice models is to estimate the final state of the orders for some arbitrary week t in the future. Let $C^{(t)}$ be the final state of an order, $C^{(0:t-1)}$ the past customer choices, $Z^{(k)}$ the state of the same order observed at time $k \leq t$, and $e^{(t)}$ the contextual information at time t. Using Bayes' rule, we have that

$$\begin{aligned} P(C^{(t)} \mid Z^{(k)}, C^{(0:t-1)}, e^{(t)}) = \\ \eta\, P(Z^{(k)} \mid C^{(t)})\, P(C^{(t)} \mid C^{(0:t-1)}, e^{(t)}) \end{aligned} \tag{6}$$

The above equation holds true if we assume that the observed order state is independent from past customer choices and from the context, given the choice $C^{(t)}$ that the customer will eventually make. Note that $P(C^{(t)} \mid Z^{(k)}, C^{(0:t-1)}, e^{(t)})$ is the model described in Eq. (5) and that η is a normalization constant. The above equation says that the corrected distribution is obtained by weighting the predicted choices $C^{(t)}$, from the volume and choice models, according to the probability of the observation model $P(Z^{(k)} \mid C^{(t)})$. For instance, assuming that the order state is predicted to be $C^{(t)} = c$ for a future week t and that the current state at week $k < t$ is $Z^{(k)} = z$, then $P(Z^{(k)} = z \mid C^{(t)} = c)$ represents the weight assigned to the prediction $P(C^{(t)} = c \mid C^{(0:t-1)}, e^{(t)})$. Intuitively, if the combination $(Z^{(k)} = z \mid C^{(t)} = c)$ has been observed frequently (infrequently) in the past, the related prediction will be given high (low) weight. Therefore, we can regard the prediction from the volume and choice models as priors and the corrected choices upon observed states as the posterior distribution over the expected outcome from the customers.

6 Results

6.1 Comparison of User-Level Accuracy of Models

The autoregressive model of Eq. (5) was implemented using both a multinomial logit classifier and a random forest classifier with scikit-learn[3] [20]. We found that multinomial logistic regression generally outperforms random forests in terms of predicting whether a user will select a given recipe in a given week (Table 1). The training set was taken from the customer

	Korean Pork Tacos	Blackened Chicken	Roasted Eggplant Pitas
Logistic regression	90.15%	96.47%	63.86%
Random forest	86.86%	94.45%	54.84%

Table 1. Comparison of F_1 scores between a logistic regression model and a random forest model, calculated on our prediction of which users (out of a sample of roughly 7,000 users total) would receive each of the three listed recipes during the week of September 19, 2016.

history of the same set of users that were predicted on, during the period between January 5, 2015 and September 12, 2016. Grid searches were performed on both models to find the optimal F_1 scores on individual-level predictions over 24 Blue Apron recipes offered in September/October 2016. Based on the results of those grid searches, an L2 penalty of 1.5×10^5 was applied for the logistic regression, and for the 100-tree random forest, 5 features were considered at each split, and at least 30 samples were required to be at each leaf node. (L1 and elastic-net penalties failed to improve upon the performance of the L2 penalty.) For both models, roughly 150 features (varying slightly on the type of recipe) were used in total: these features were extracted from various attributes of the recipes offered (ingredients, cuisine type, etc.) as well as users' past ordering behavior regarding recipes with those specific attributes. We found that features encoding how likely users are to order recipes with various attributes (i.e. users' past order rates for beef recipes, for Italian recipes, etc.) are generally more predictive than features encoding whether recipes in the week that we're predicting contain various attributes (i.e. whether they contain beef, whether they are Italian dishes, etc.). Despite the fact that we find random forests to have generally worse performance than multiclass logistic regressions at predicting the likelihood of a user selecting a certain recipe, random forests have several properties that make them generally more useful for our specific application than logistic regressions, especially

[3] http://scikit-learn.org/stable/

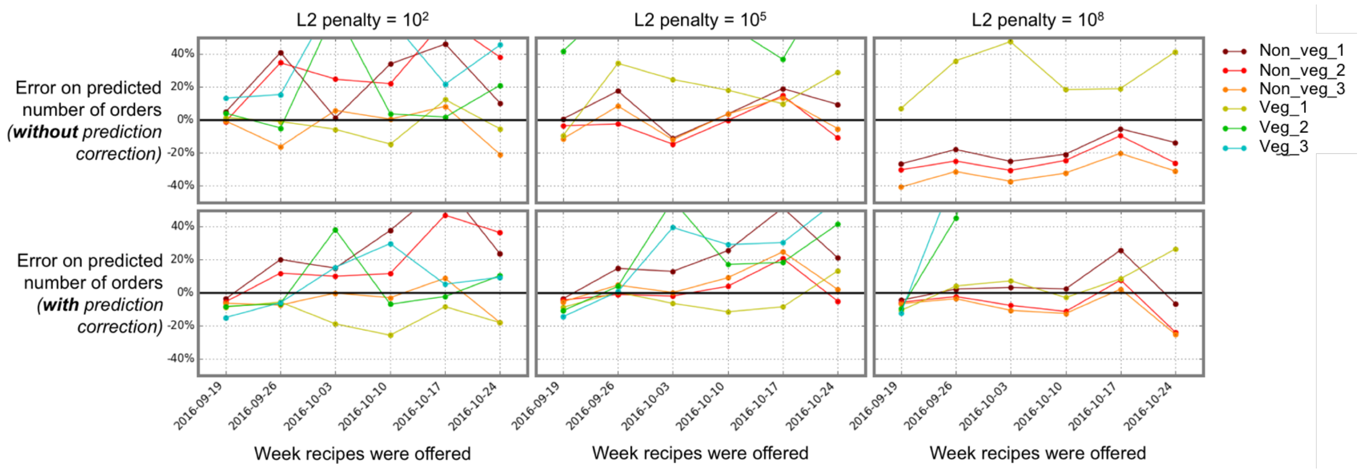

Figure 2. When using logistic regression to predict the number of orders of a specific recipe in a specific week, accuracy depends strongly on regularization strength. Figures show the percentage error in the predicted number of meals as a function of the type of recipe (indicated by the legend on the right), as a function of which week the recipes were offered (x-axis) and L2 penalty value (left, middle, right columns). Top row shows percentage errors without prediction correction, and bottom row shows percentage errors with prediction correction, as described in Prediction Correction, above.

for our volume model. First, random forests can implicitly learn the interactions between features, for instance, between the presence of a certain ingredient in a recipe and a user's past high rate of ordering recipes containing that ingredient, without having to manually specify the possibility of such interactions in the model [16, 17]. This eliminates the work otherwise needed to find the most suitable interaction terms among a very large set of features. In addition, with linear models such as logistic regression, it is often necessary to either apply mathematical transformations to continuous-valued features or split the range of feature values into several one-hot categorical values in order to adequately model a feature whose relationship with the response variable is nonlinear [18]. By comparison, nonparametric models such as random forests are naturally flexible enough to learn any mathematical relationship between features and output variables [10]. Not needing to search for the right mathematical relationship between features and the response variable saves a great deal of model development time, and avoiding one-hot encoding of different ranges of a continuous-valued feature allows us to drastically reduce the memory footprint of our design matrix.

6.2 Comparison of Aggregate-Level Accuracy of Models

Note that, whereas the above discussion judges logistic regressions and random forests on the basis of individual-level accuracy (namely, the fraction of cases in which the outcome of a user skipping/ordering a recipe is correctly predicted), the main use case for our models is to predict the total number of each type of recipe that will be ordered in a given week. Thus, aggregate-level accuracy is the metric on which these models are ultimately judged. We find that optimizing individual-level accuracy generally has the effect of increasing aggregate-level accuracy as well. Specifically, we have observed that roughly the same L2 penalty strength (1×10^5) is ideal in both maximizing the F_1 score over individual customer predictions and minimizing the absolute error on the total number of orders, in our sample of roughly 7,000 users. This result holds for four different validation sets (representing recipes in four different weeks in September/October 2016) and for several different categories of recipes (vegetarian and non vegetarian). When we use logistic regression to predict the total number of meals of a specific recipe that will be ordered in a given week, we find that our choice of L2 parameter strongly affects the accuracy of our predictions (Figure 2). We find here that aggregate-level accuracy is usually significantly higher for our non-vegetarian recipes than for our vegetarian recipes: we hypothesize that this is because our non-vegetarian recipes generally have higher order volume than our vegetarian recipes, leading to prediction fluctuations for vegetarian recipes due to small sample size. In contrast to the sensitivity of the logistic regression model to changes in L2 penalty strength, the accuracy of our random forest model in predicting the number of orders per recipe is relatively insensitive to changes in the number of features considered at each split in a single tree. Our predictions, on a sample of roughly 6,000 users, were modulated with a logistic-regression volume model (see Theoretical Framework in Section 3). When scaling up our volume model prediction to roughly 37,000 users and applying prediction correction, we see that the RMS error in our prediction of the total number of orders that users will place is consistently less than 6%, even when predicting as far as 8 weeks into the future (Table 2). We calculate RMS error by averaging over the error of our 100-tree random-forest-based volume model in predicting sales data for 8 different weeks. Features used related to users' age and purchase history, as well as their stated willingness to order recipes containing specific proteins. As discussed earlier, we perform Monte Carlo simulations so that

	Mean error	RMS error
2-week-out prediction	0.06%	3.54%
4-week-out prediction	2.19%	5.36%
6-week-out prediction	2.82%	3.90%
8-week-out prediction	1.32%	3.97%

Table 2. Mean and RMS error on the volume model's prediction of the total number of orders placed in a given week, as a function of the number of weeks in advance that the prediction was made.

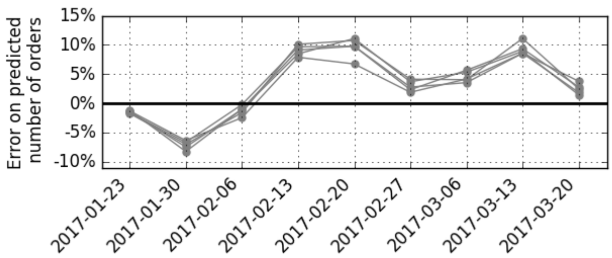

Figure 3. Variation resulting from multiple Monte Carlo simulations (5 shown) over several weeks using Algorithm 1. Values represent the percentage error in the total number of orders placed by customers, as predicted by the volume model.

predictions made at each timestep of our autoregressive model can be transformed into realistic feature values for the following timestep. The error that is introduced by this sampling is shown in Figure 3. Notably, the variability among Monte Carlo simulations is much lower than the overall error of our model, which implies that a low number of Monte Carlo simulations is sufficient to achieve reasonable coverage of the distribution of possible simulation results.

6.3 Contribution of Individual Features to Model Accuracy

One well-known drawback of random forests is their relative lack of interpretability. In the case of multiclass logistic regression, the values of the fitted coefficients can be directly used to calculate the impact that features have on the probability of a positive outcome. On the other hand, assessing the impact of any one feature in a random forest involves summing the contribution that that feature has on the predicted outcome across hundreds of locations within the random forest [11]. We've found the TreeInterpreter package for Python [22] to be useful in calculating the contribution of each feature to the prediction by compiling each feature's contribution on each tree, and then averaging over all trees in the random forest (see [19] for a theoretical justification). We use this package to track how much each feature's contribution to the probability of a user to order changes from one week to the next, which in turn helps us understand how each feature influences the change in our prediction of the aggregate number of orders over time.

7 Final Remarks and Future Work

Future versions of our prediction engine will advance both its interpretability and its predictive power. To increase our prediction engine's interpretability, we will extend on the work of the TreeInterpreter package to calculate the total additive contribution that each feature makes to the predicted number of orders. Such a calculation is possible because each split of each decision tree in a random forest creates an additive change in the prediction for each user, and these additive changes can be aggregated for each feature over users. By contrast, a generalized linear model would not allow for such a clean breakdown of feature contributions on the aggregate level to be calculated. This potential for enhanced interpretability is one reason that random forests are more suitable for our use case than logistic regression, despite the latter's higher user-level accuracy (Table 1).

To increase our engine's predictive power, we will explore the use of additional features that will allow us to identify other important behavioral signals, such as service cancellations, seasonal and holiday trends, and ordering patterns of infrequent orderers. We also plan to build a model to recommend specific recipes to users directly through our web/mobile interfaces: this model will allow us to work towards a conversational system of iterative recommendation and prediction that will enable refining our predictions over time and using those predictions to recommend better recipes.

Acknowledgements

Special thanks to the Demand Planning and Data Analytics teams for the constructive feedback on model features and data insights; and to the Data Engineering team for the continuous support on data persistency and model scalability.

References

[1] Lada Adamic, Moira Burke, Ama Herdadelen, and Dirk Neumann. Cat People, Dog People. `https://research.fb.com/cat-people-dog-people`, August 2016.

[2] D. Agrawal and C. Schorling. Market share forecasting: An empirical comparison of artificial neural networks and multinomial logit model. *Journal of Retailing*, 72(4):383 – 407, 1996.

[3] H. C. Barksdale and J. E. Hilliard. A cross-spectral analysis of retail inventories and sales. *The Journal of Business*, 48(3):365–382, Jul 1975.

[4] K. Bel and R. Paap. A multivariate model for multinomial choices. Econometric Institute Research Papers EI 2014-26, Erasmus University Rotterdam, Erasmus School of Economics (ESE), Econometric Institute, 2014.

[5] Ching-Wu Chu and Guoqiang Peter Zhang. A comparative study of linear and nonlinear models for aggregate retail sales forecasting. *International Journal of Production Economics*, 86(3):217–231, December 2003.

[6] Darren DeMatas. What To Sell Online: Digital Or Physical Products? https://selfstartr.com/what-to-sell-online/, January 2017.

[7] Philip Doganis, Alex Alexandridis, Panagiotis Patrinos, and Haralambos Sarimveis. Time series sales forecasting for short shelf-life food products based on artificial neural networks and evolutionary computing. *Journal of Food Engineering*, 75(2):196–204, July 2006.

[8] Ecommerce Foundation. Global B2C E-commerce Report 2016.

[9] eMarketer. Worldwide retail ecommerce sales will reach $1.915 trillion this year. https://www.emarketer.com/Article/Worldwide-Retail-Ecommerce-Sales-Will-Reach-1915-Trillion-This-Year/1014369, August 2016.

[10] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer, New York, 1st ed. 2013, corr. 6th printing 2016 edition edition, August 2013.

[11] Victor E. Kuz'min, Pavel G. Polishchuk, Anatoly G. Artemenko, and Sergey A. Andronati. Interpretation of QSAR Models Based on Random Forest Methods. *Molecular Informatics*, 30(6-7):593–603, June 2011.

[12] Sarah Lambert. Top 10 Advantages of Forecasting in Business. http://www.brighthub.com/office/entrepreneurs/articles/109618.aspx, May 2011.

[13] Richard Lazazzera. How to Choose an Ecommerce Business Model. https://www.webelite.co.za/blog/50-how-to-choose-an-ecommerce-business-model.

[14] Wan-I. Lee, Cheng-Wu Chen, Kung-Hsing Chen, Tsung-Hao Chen, and Chia-Chi Liu. Comparative study on the forecast of fresh food sales using logistic regression, moving average and BPNN methods. *Journal of Marine Science and Technology*, Vol. 20(No. 2):142–152, April 2012.

[15] Chi-Jie Lu, Tian-Shyug Lee, and Chia-Mei Lian. Sales forecasting for computer wholesalers: A comparison of multivariate adaptive regression splines and artificial neural networks. *Decision Support Systems*, 54(1):584–596, December 2012.

[16] Kathryn L Lunetta, L Brooke Hayward, Jonathan Segal, and Paul Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics*, 5:32, December 2004.

[17] Brett A. McKinney, David M. Reif, Marylyn D. Ritchie, and Jason H. Moore. Machine Learning for Detecting Gene-Gene Interactions. *Applied bioinformatics*, 5(2):77–88, 2006.

[18] Andreas C. Müller and Sarah Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 1 edition edition, October 2016.

[19] A. Palczewska, J. Palczewski, R. M. Robinson, and D. Neagu. Interpreting random forest models using a feature contribution method. In *2013 IEEE 14th International Conference on Information Reuse Integration (IRI)*, pages 112–119, August 2013.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Adele Peters. Cooking With A Meal Kit May Waste 62% Less Food Than Grocery Store Ingredients. https://www.fastcompany.com/3063772/cooking-with-a-meal-kit-may-waste-62-less-food-than-grocery-store-ingredients, September 2016.

[22] Ando Saabas. Interpreting random forests. http://blog.datadive.net/interpreting-random-forests/, October 2014.

[23] Tom Talpir. Decreasing Game Churn: How Upopa used ironSource Atom and Amazon ML to Engage Users. https://aws.amazon.com/blogs/big-data/decreasing-game-churn-how-upopa-used-ironsource-atom-and-amazon-ml-to-engage-users/, January 2017.

[24] J.G.A.J. van der Vorst, A.J.M. Beulens, W. de Wit, and P. van Beek. Supply chain management in food chains: improving performance by reducing uncertainty. *International Transactions in Operational Research*, 5(6):487–499, 1998.