Multi-Region/Multi-Product Demand Forecasting

Alex Fish

1 Overview

The primary goal of this project is to create a forecasting model able of predicting demand on variety of products across a range of store locations. The idea is that predicting trends in demand for a particular product at a certain location can improved by tracking trends in sales figures and other data from other products and other locations.

The data used in this project represents sales, product, and store location information for a set of 51 food products sold at 77 store locations which are spread across 51 cities for 145 weeks.

I created a dashboard for exploring the effect of advertising and pricing on predicted demand trends for a user-chosen locations and products.

In Section 2, I describe the data. In Section 3, I briefly analyze the data. In Section 4, I discuss various ways to models the data. In Section 5, I evaluate the performance of these models. In Section 6 I discuss the dashboard's capabilities. Finally, in Section 7, I wrap up and provide concluding remarks.

1.1 Keywords

forecasting, regression, time series, auto-regressive (AR) models, XGBoost, ensemble modeling, parameter tuning, feature selection, table joins

1.2 Libraries Used

pandas, statsmodels, scikit-learn, XGBoost, Matplotlib, NumPy

2 Data

The data comes from a Kaggle dataset. The dataset did not come with location or any company information and was deficient in details on the meaning behind some of the variables, especially on the store location data.

The dataset is comprised of three tables, a table describing each store location (referred to as "centers"), a table describing each food product (referred to as "meals"), and a table of the total orders per week of each product per location. Table 1 contains the features in the "Centers" table, Table 2 contains the

features in the "Meals" table, and Table 3 contains the features in the "Orders" table.

Column Name	Column Description	Column Value Type
Center ID	Unique identifier for each store location	Categorical integer code
City Code	Identifier for the city in which the center is located	Categorical integer code
Region Code	Identifier for the region in which the center is located	Categorical integer code
Center Type	Identifier differentiating types of centers	Categorical {TYPE_A, TYPE_B, TYPE_C}
Operational Area	Land area which the center has influence over (square mi. or square km)	Positive real number

Table 1: Centers

Column Name	Column Description	Column Value Type
Meal ID	Unique identifier for each product	Categorical integer code
Category	Type of food product	Categorical {Beverages, Biryani, Desert, Extras, Fish, Other Snacks, Pasta, Pizza, Rice Bowl, Salad, Sandwich, Seafood, Soup, Starters}
Cuisine	Ethnic origin of food product	Categorical {Continental, Indian, Italian, Thai}

Table 2: Meals

3 Analysis

There are 77 centers in the dataset, which reside in a total of 51 cities. One city contains nine centers, one contains nine, one contains 3, nine cities contain two centers each, and the rest of the cities contain one center each. One region contains 30 centers, one contains 21, one contains 17, one contains 5, and the

Column Name	Column Description	Column Value Type
ID	Unique identifier for the weekly order record	Categorical integer code
Week	Week in which the orders occurred	Integer in [1,145]
Center ID	Center in which the orders occurred (see Centers table)	Categorical integer code
Meal ID	Meal which was ordered (see Meals table)	Categorical integer code
Checkout price	Product price (unclear)	Positive real number
Base price	Product price (unclear)	Positive real number
Emailer for promotion	Indicates whether the center ran an email advertisement for the product	Boolean {0,1}
Homepage featured	Indicates whether the center features the product on their website	Boolean {0,1}
Total orders	Total orders of the product at that center throughout the week	Positive integer

Table 3: Orders

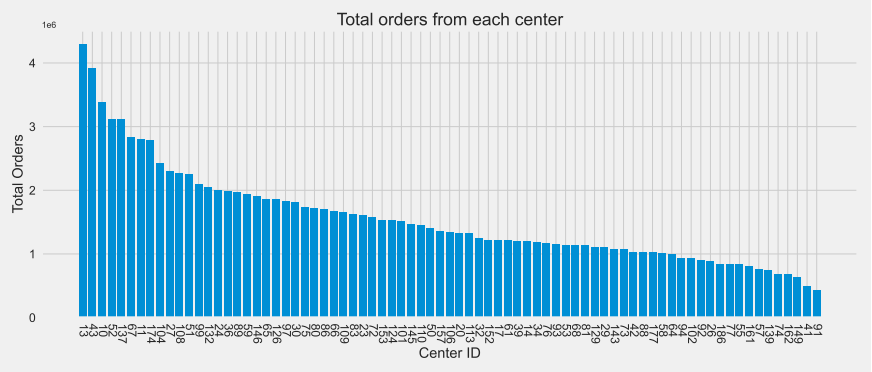
remaining four regions contain one center each. There are 43 type A centers, 15 type B centers, and 19 type C centers. Without location information or any indication on the meaning of type A, B, and C centers, I will refrain from diving deeper there.

There are 51 types of meals in the dataset. There are 12 types of beverages and 3 of each type of food product. Each category of food product is associated with a single cuisine, for example all rice bowl meals are Indian, all sandwich meals are Italian, etc. The beverages are evenly distributed across the cuisines.

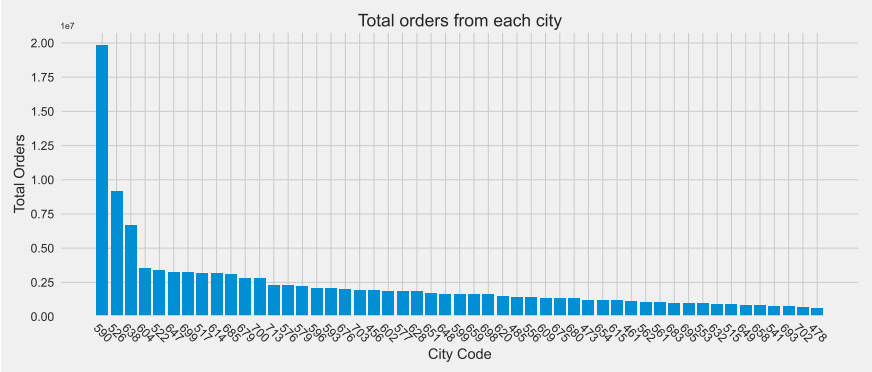
The orders data spans 145 weeks ($\sim$2.7 years). Not every center had orders for every meal in every week, so while there are no missing values in the dataset, there are entries which may be expected to exist that don't and special care must be taken when processing the data for modeling.

Figure 1a shows the total orders per center throughout the entire timespan. There is a little variation in the popularity of the centers (the center with the most orders has 15 times the total orders of the smallest) but the variation is rather tame–there are no centers with a tiny amount of orders. Figure 1b shows the total orders per city throughout the entire time span. These values are approximately proportional to the number of centers per city.

Figure 2a shows the total orders by category throughout the entire times-


Total orders from each center
1e6
Total Orders
Center ID

(a)


Total orders from each city
1e7
Total Orders
City Code

(b)

pan. Beverages are by far the most popular item, which makes sense as they are likely purchased frequently with another food item. However, beverages are also the item with the most entries on the menu, so they are likely to be disproportionately represented in the total orders. The most common meal category is rice bowl, followed by the three Italian food products, and then the rest. The grouping of popular Italian items shows in Figure 2b, where Itlian is by far the most commonly ordered cuisine. Figure 2c shows the disparity between popular and unpopular items. The most popular items, with IDs 2290, 1885, and 1754 are an Indian rice bowl, Thai beveraage, and Italian sandwich, respectively. The least popular items with IDs 1770 and 2104 are Indian biryani and Continental fish, respectively.

Figure 3a shows the total meals ordered per week. We can see that, overall, the series is stationary around a mean of 800,000 with occasional sharp spikes up or down. Diving a little deeper, Figure 3b splits this plot by cuisine. We can see the high variability in the total Italian orders, although they maintain the highest orders. Thai is consistently second place with little variability. For a brief window of time early in the dataset, Indian became the dominant cuisine. Continental is consistently the worst performing cuisine. Diving further still, eliminating any summing across centers, Figure 3c shows the orders per week by cuisine for just the center with the most orders across the dataset, center 13. Interestingly, Thai is center 13's most common cuisine sold rather than Italian. We also see a greater disparity between the cuisines here.

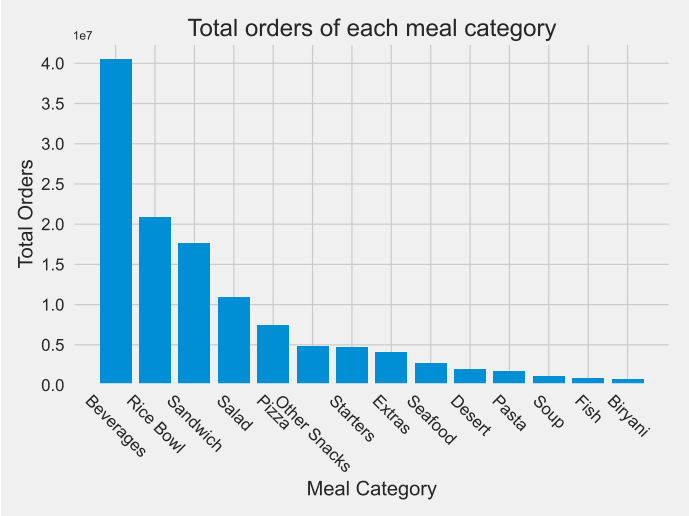
Figure 4 shows the 5 most-ordered products at center 13 over time. These 5 are a Thai beverage (1885), another Thai beverage (1993), an Indian beverage (2139), the third Thai beverage (2539), and another Indian beverage. The Thai beverage (1885) has started gaining popularity over the Thai beverage (1993) in the past year or so, while the Indian beverage is on a notable downward trend.

For forecasting purposes, let's consdier a few center's meal orders over time as a time series. Let's focus on center 13 as was done in Figures 3 and 4. Figure 5a shows the autocorrelation plot of center 13's orders of meal 1885 (the most popular meal at center 13) over time. This particular pair of center and meal has six significant lag values. This of course implies that the previous six weeks' order totals are generally informative of the next week's order totals. Not every pair has so many significant lags in the autocorrelation plot, Figure 5b shows the autocorrelation plot of center 13's orders of meal 1993 over time, in which only 3 significant lags are present.

The features Base Price and Checkout Price have a correlation of 0.9533 and are strongly collinear, as seen in Figure 6.

4 Models

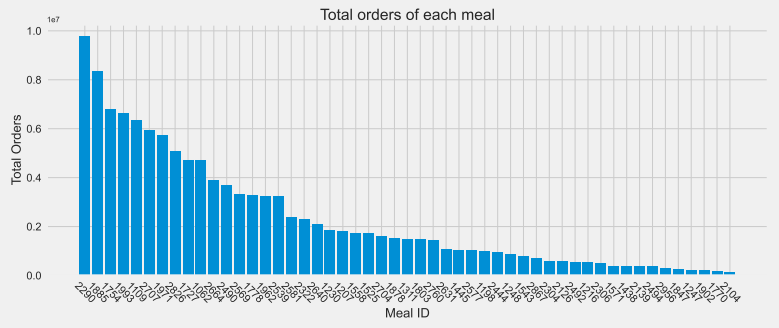
I will use a variety of model structures to get the best forecasting capabilities that we can. I will split the order data into training and testing sets by taking the first 116 weeks (80% of the total 145 weeks in the data set) for training data, and using the last 29 weeks (the remaining 20% of the data) as testing data.


Total orders of each meal category
1e7
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Total Orders
Beverages
Rice Bowl
Sandwich
Salad
Pizza
Other Snacks
Starters
Extras
Seafood
Desert
Pasta
Soup
Fish
Biryani
Meal Category

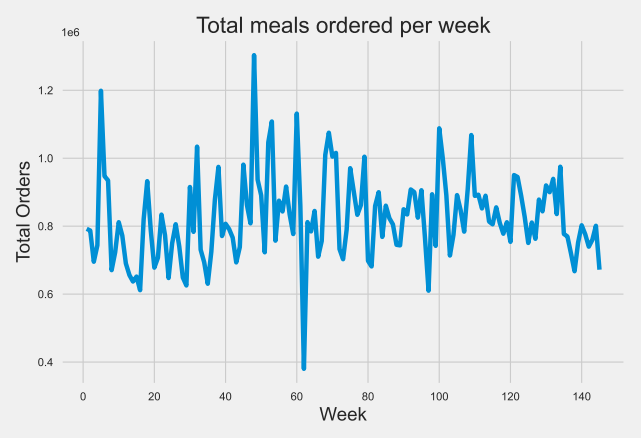
(a)


Total orders of each cuisine
1e7
4
3
2
1
0
Total Orders
Italian
Thai
Indian
Continental
Cuisine

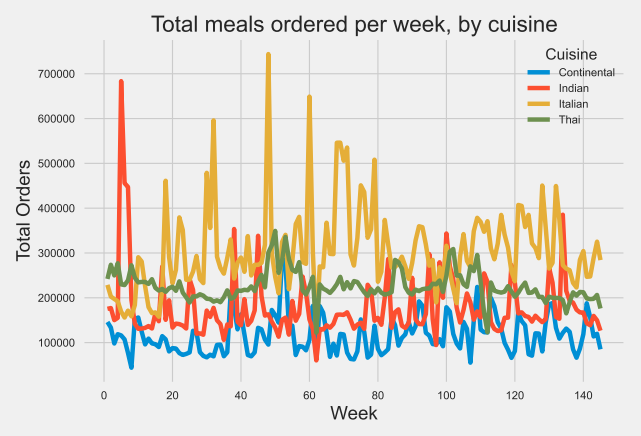
(b)


Total orders of each meal
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Total Orders
Meal ID

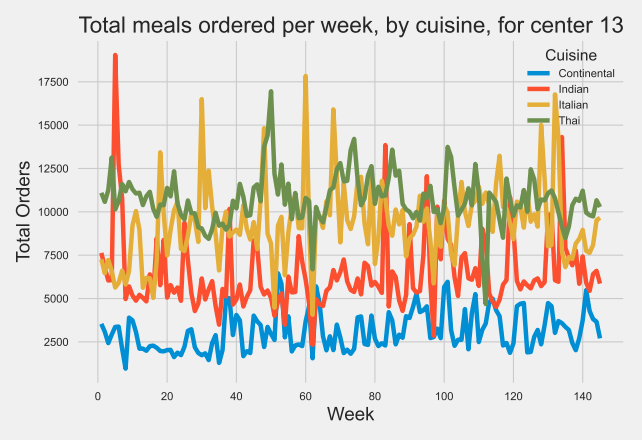
(c)


Total meals ordered per week
1e6
1.2
1.0
0.8
0.6
0.4
Total Orders
0
20
40
60
80
100
120
140
Week

(a)


Total meals ordered per week, by cuisine
Cuisine
Continental
Indian
Italian
Thai
700000
600000
500000
400000
300000
200000
100000
Total Orders
0
20
40
60
80
100
120
140
Week

(b)


Total meals ordered per week, by cuisine, for center 13
Cuisine
Continental
Indian
Italian
Thai
17500
15000
12500
10000
7500
5000
2500
Total Orders
0
20
40
60
80
100
120
140
Week

(c)

Figure 3

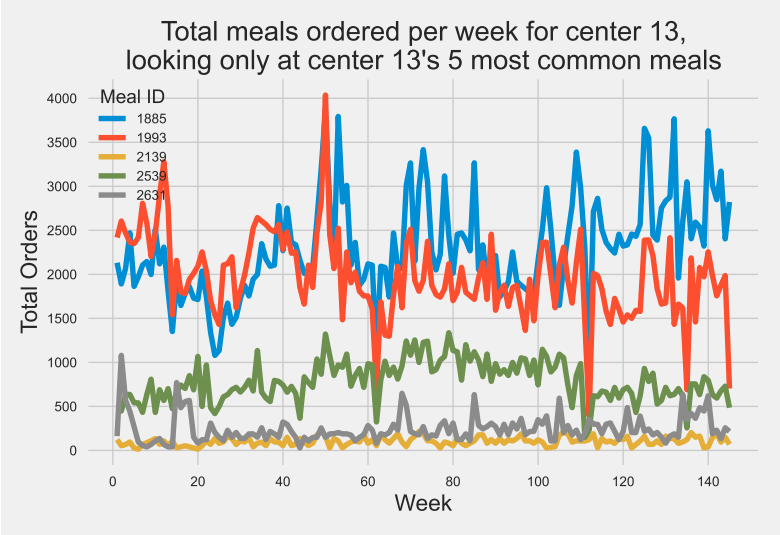


Figure 4

First, as a baseline model, we will simply predict the average of a center's meal demand as the mean of that center's demand for that meal throughout the training data. This is simply set up by the use of a `data.groupby(by=['Center ID','Meal ID']).mean()` function evaluation, and a custom `predict` function to use this reference data for this simplistic model.

The non-trivial require some data processing. Because the difference between Checkout Price and Base Price is unclear and they are so strongly correlated, I drop Checkout Price from the training and testing data.

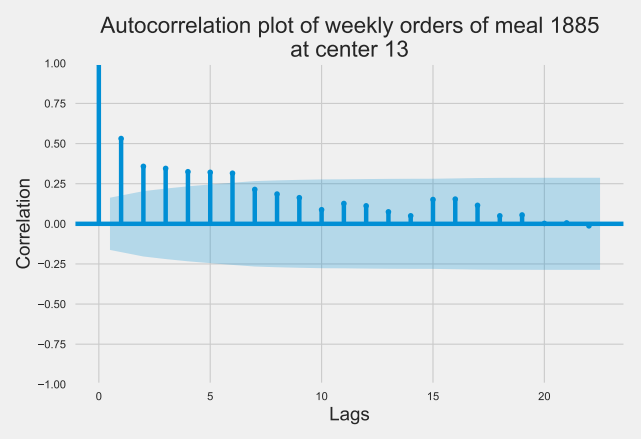
Next, I added lag data (up to six lags) such that each entry additionally includes the number of orders of the entry's meal from the the entry's center's from the previous six weeks. While not every center's meal orders had six significant lags, I decided to include the maximum relevant information. I limited it to six lags to follow the general rule of thumb of limiting the total number of ARIMA terms for interpretability and avoiding over-fitting.

I then normalized the numerical input training data: the lagged weekly total order values and the corresponding Operational Area of the center. I divided the week column by 52 to get the values within a reasonable range. Finally, I used a one-hot encoding for the categorical variables (IDs, area codes, etc.).

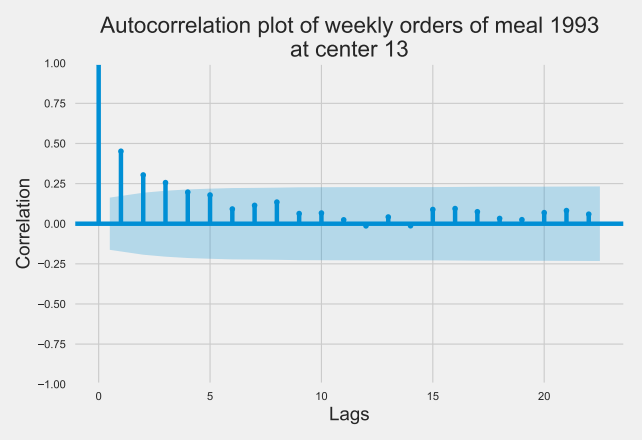
Because the exploration process did not lead me to believe the order history was seasonal, I did not include seasonal lag terms nor encode the week to be seasonal with any modulo, sin-cos, or other encoding.

Once the data was processed, I used both a linear regression with Lasso regularization and an XGBoost model for three different model structures.

The first model structure is the simplest, it takes in all of the training data to predict the demand for any meal, for any center. The Lasso regularization


Autocorrelation plot of weekly orders of meal 1885
at center 13
Correlation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
Lags

(a)


Autocorrelation plot of weekly orders of meal 1993
at center 13
Correlation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
5
10
15
20
Lags

(b)

Figure 5

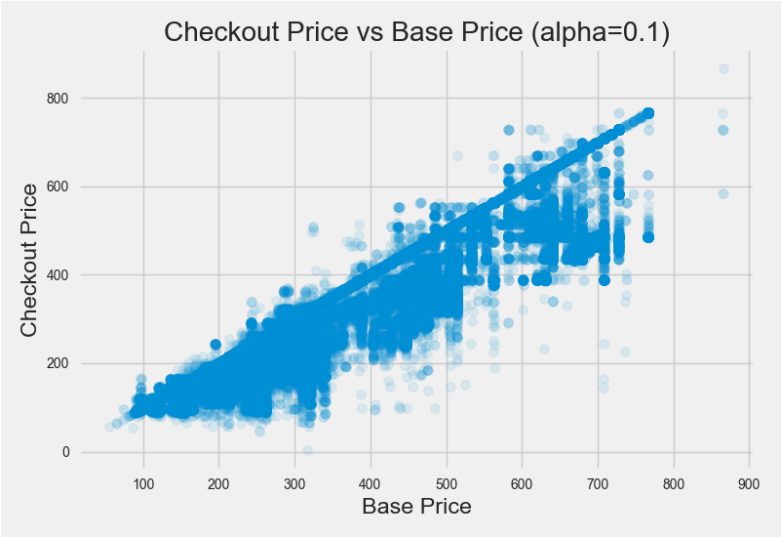


Figure 6

parameter was chosen through cross-validation on the test set using 3-fold time-series validation sets with weeks (1-71,72-86), (1-86,87-101), (1-101,102-116) for the (training,validation) sets, respectively (using 15 weeks for validation in each fold, approximately 10% of the dataset). The same Lasso regularization parameter was applied to the later model structures to reduce training time. The XGBoost model parameters were chosen by evaluating different parameters' training performance on the test set and was performed separately for the three XGBoost models due to their fundamentally different structure.

The second model structure is an ensemble model which breaks apart the forecasting into an individual, smaller model for each center. The idea behind the ensemble model is that it might be able to more accurately forecast demand when it isn't distracted by other centers' data, while still retaining information on the full set of meal order data. I'll refer to these as the "Ensemble-Center" models.

The third model structure is similar to the second with similar rationale. It is an ensemble model which uses individual models for each *meal*. I'll refer to these as the "Ensemble-Meal" models.

I did not create an ensemble model which uses individual models for each center's demand for each meal because there is not enough information to get a good enough picture for each model. Some centers order certain meals very infrequently, or never sent out email advertisements for some meals, or never featured the meals on the home page. A wider view was necessary for robustly capturing behavior of the independent variables.

5 Performance

The single linear regression model (which used the whole dataset) ended up "dropping" a significant amount of features through the regularized optimization process. These "dropped features" are associated with coefficient values of 0. All of the dropped features were varying values from the one-hot encoding of the categorical variables, likely because they were all associated with numbers of meals ordered close enough to the of the dataset that it was not important to assign their own coefficient. For example, around half of the centers and cities had a coefficient of 0. The base price of the meal had a coefficient of -34.7, implying more expensive items are ordered less frequently. Advertising showed significant benefits for sales, with an email promotion providing an extra 316.9 weekly orders on average and a feature on the website homepage providing an extra 159.6 weekly orders on average.

Figure 7 shows the single XGBoost model's importance of the top 20 features. The non-categorical variables are all more important than the categorical variables. Because there are so many categorical variables, this is not too surprising. All of the lag variables, including the sixth lag, have high importance, showing the benefit of including them in the modeling process.

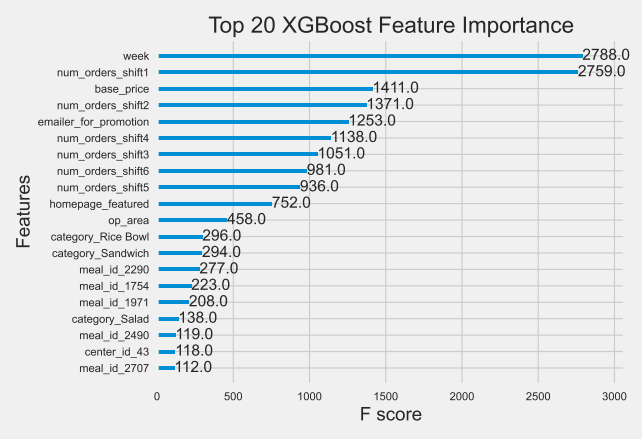


Figure 7

I'll report the performance of these models with three key metrics: RMSE and R2 (r-squared, the coefficient of determination), and MAPE (mean-absolute-percentage-error, where a value of 1.0 indicates 100% error from the actual value). Table 4 shows how the models performed on the test set in these metrics.

The baseline model only out-performed the single linear regression model. The ensemble linear regression models greatly out-performed the single linear regression model, showing that removing the noise from less-relevant data points improved performance. The Ensemble-Meal model outperformed the Ensemble-

Model	RMSE	R2	MAPE
Baseline	248.96	0.54	0.81
Linear Regression	211.37	0.67	0.93
XGBoost	161.98	0.80	0.51
Linear Regression Ensemble-Center	193.38	0.71	0.77
XGBoost Ensemble-Center	182.26	0.75	0.57
Linear Regression Ensemble-Meal	182.15	0.75	0.63
XGBoost Ensemble-Meal	174.99	0.77	0.55

Table 4: Model performance

Center model by a slight margin.

The XGBoost models, in contrast, saw a degradation in performance from the single model to the ensemble models. This is likely due to the XGBoost's nonlinearity allowing it to capture information available in the extra data points, improving its predictive capabilities.

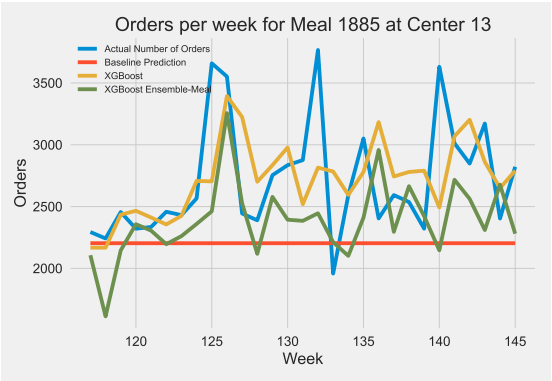
Let's inspect the performance visually using the baseline and the two best models, the XGBoost and XGBoost Ensemble-Meal models. Figure 8a shows all of the model predictions of center 13's meal 1885 demand. The single XGBoost model's predictions seem to more accurately track the actual values on average than the Ensemble-Meal model, while the baseline was consistently too low. In this case, the single XGBoost model had a MAPE value of 0.12, the XGBoost Ensemble-Meal model a value of 0.15, and the baseline a value of 0.18.

Figure 8b shows all of the model predictions of center 13's meal 1993 demand. Here, orders are down compared to the historical mean so the baseline is consistently too high. However, the XGBoost Ensemble-Meal model beats out the single XGBoost model fairly handily. In this case, the single XGBoost model had a MAPE value of 0.32, the XGBoost Ensemble-Meal model a value of 0.27, and the baseline a value of 0.32.

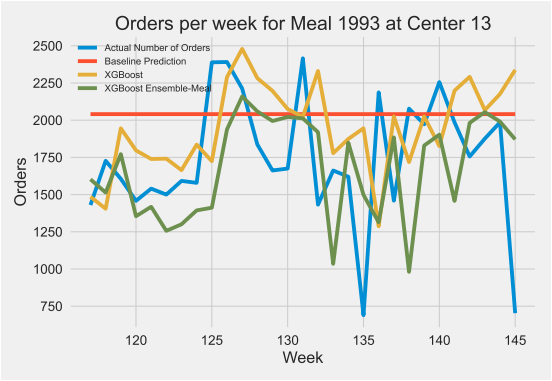
Figure 8c shows all of the model predictions of center 55's meal 1885 demand. Here, the actual orders are fairly noisy, having far sharper oscillations than the previous plots. The models tend to fail to capture this noise, which is expected as models cannot perfectly capture the behavior of the real world. The XGBoost Ensemble-Meal model again out-performs the single XGBoost model, as it sticks closer to the mean of the actual values. In this case, the single XGBoost model had a MAPE value of 0.58, the XGBoost Ensemble-Meal model a value of 0.42, and the baseline a value of 1.18.

Generally the XGBoost models got pretty close to the true values across many centers and meals, but could not accurately capture the randomness of the large spikes (up or down) in the weekly orders.

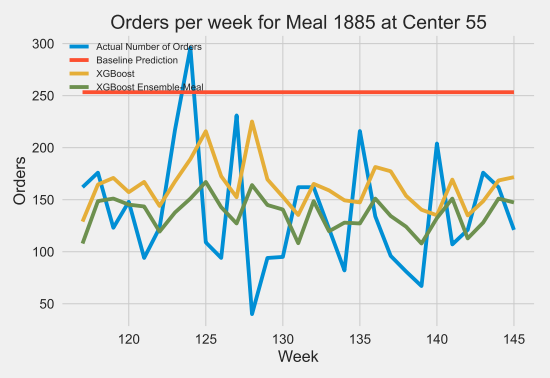
Since it is impossible to meaningfully analyze performance across all 77 centers and 51 products, I will stop here on that front.


Orders per week for Meal 1885 at Center 13
Actual Number of Orders
Baseline Prediction
XGBoost
XGBoost Ensemble-Meal
Orders
Week
3500
3000
2500
2000
120
125
130
135
140
145

(a)


Orders per week for Meal 1993 at Center 13
Actual Number of Orders
Baseline Prediction
XGBoost
XGBoost Ensemble-Meal
Orders
Week
2500
2250
2000
1750
1500
1250
1000
750
120
125
130
135
140
145

(b)


Orders per week for Meal 1885 at Center 55
Actual Number of Orders
Baseline Prediction
XGBoost
XGBoost Ensemble-Meal
Orders
Week
300
250
200
150
100
50
120
125
130
135
140
145

(c)

Figure 8

6 Dashboard

The dashboard is a small tool for exploring the effect of advertising and pricing on the demand of a meal at a given center. Figure 9 shows the user interface. The user chooses a given center and meal, and is shown the historical data from that pair, the price, advertising status, and number of orders from each week. Next, the user is given some plotting and forecasting options. They can choose the number of historical values to plot and the number of weeks ahead to forecast. They can also choose the price to set the item and how to advertise it, if at all. The pricing window allows values from the previous minimum minus 10% to the previous maximum plus 10% with a default value of the most recent price.

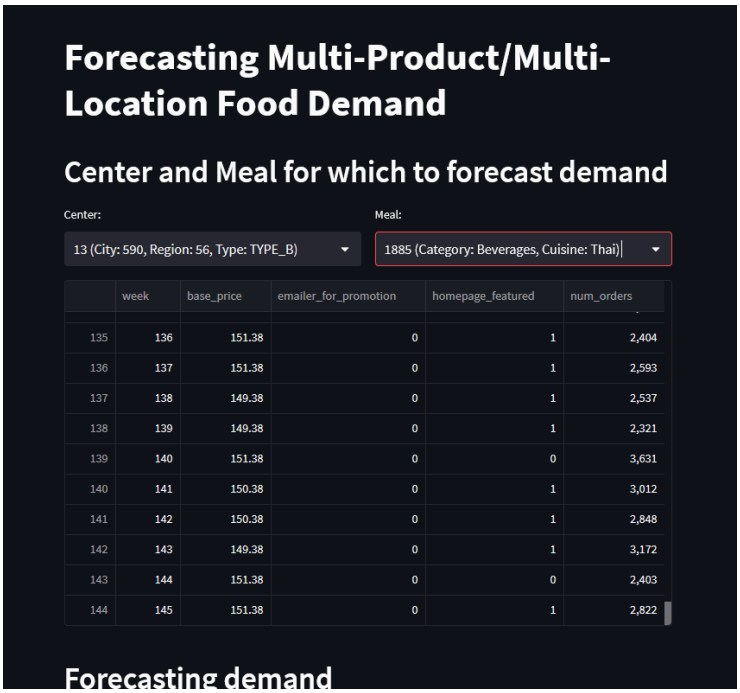
7 Conclusions

I analyzed a multi-region, multi-product sales history dataset with the aim of forecasting future sales. I constructed a small suite of models informed by analysis of the data and evaluated them on unseen data. The baseline model, formed from the assumption that the center's average meal sales over a given period, would be an accurate estimation of sales in the future, performed the worst.

I processed the data into a more suitable form for modeling. I added six lag features (informed by plotting the autocorrelation function) of each center's meal orders for the prvious weeks, removed a strongly collinear term, and transformed the variable values to remove undue favor towards larger values.

I evaluated a linear regression and XGBoost model using both the entire dataset and in ensemble form with a smaller, individual model per center. The linear regression in the ensemble model provided quite bad results but the other models were effective. The XGBoost models performed quite well, while the linear regression performed around as well as the baseline model, sometimes worse, but usually a little better.

I would recommend using the non-ensemble whole-dataset XGBoost model for forecasting because it behaves equal to or better than the ensemble XGBoost model and the simpler definition, usage, and training are desirable.


Forecasting Multi-Product/Multi-Location Food Demand
Center and Meal for which to forecast demand
Center:
13 (City: 590, Region: 56, Type: TYPE_B)
Meal:
1885 (Category: Beverages, Cuisine: Thai)
week base_price emailer_for_promotion homepage_featured num_orders
135 136 151.38 0 1 2,404
136 137 151.38 0 1 2,593
137 138 149.38 0 1 2,537
138 139 149.38 0 1 2,321
139 140 151.38 0 0 3,631
140 141 150.38 0 1 3,012
141 142 150.38 0 1 2,848
142 143 149.38 0 1 3,172
143 144 151.38 0 0 2,403
144 145 151.38 0 1 2,822
Forecasting demand

(a)


143 144 151.38 0 0 2,403
144 145 151.38 0 1 2,822
Forecasting demand
(Last known week's demand: 145)
Total previous weeks to plot:
50
0
145
Total weeks ahead to forecast:
10
1
25
Base price:
151.38
108.00
181.00
Homepage feature
Emailer for promotion
4,000
3,500
3,000
2,500
2,000
1,500
1,000
500
0
95 100 105 110 115 120 125 130 135 140 145 150 155
Week
Forecasted Orders
Previous Orders

(b)

Figure 9