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.

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. Finally, in Section 6, I wrap up and provide concluding remarks.

1.1 Keywords

forecasting, regression, time series, auto-regressive (AR) models, XGBoost, ensemble modeling, 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 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

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

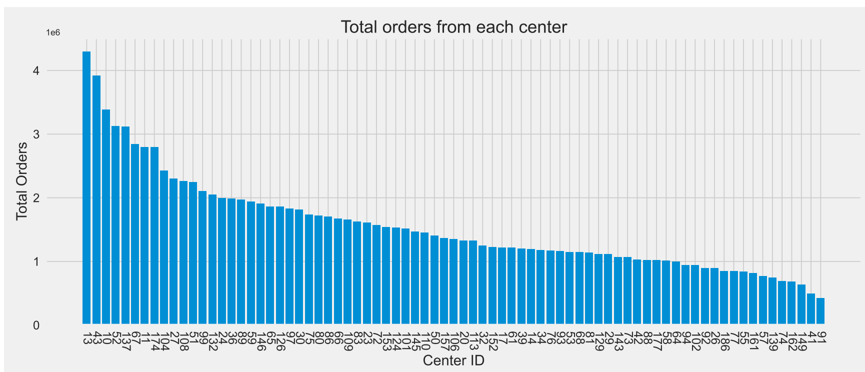
Table 3: Orders

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.

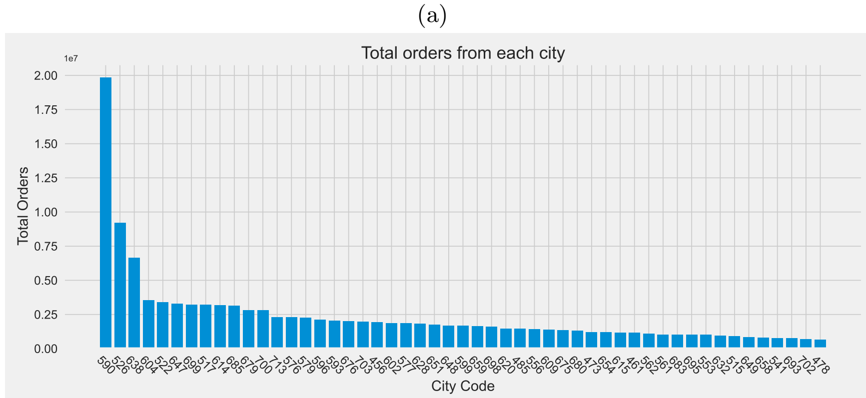
The orders data spans 145 weeks ($\sim$2.7 years). Not every center had orders for every meal in every week, so while there are no missing values in the dataset, there are entries which may be expected to exist that don't and special care must be taken when processing the data for modeling.

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 timespan. 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


Total orders from each center
1e6
Total Orders
0
1
2
3
4
Center ID

(a)


Total orders from each city
1e7
Total Orders
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
City Code

(b)

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.

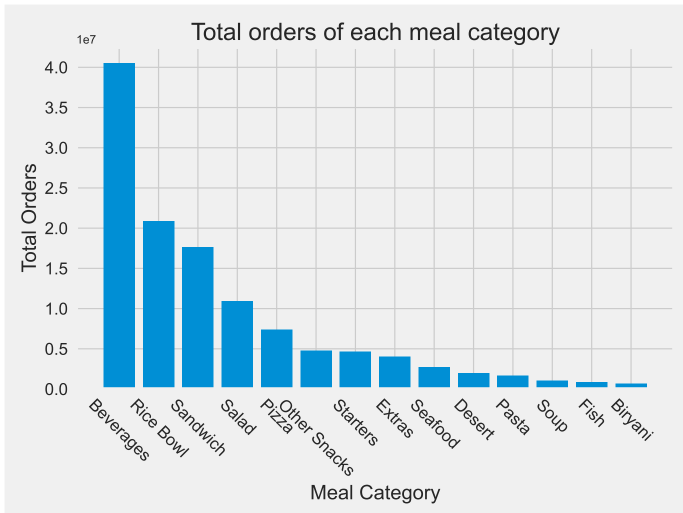
For forecasting purposes, let's consdier a few center's meal orders over time as a time series. Let's focus on center 13 as was done in Figures 3 and 4. Figure 5a shows the autocorrelation plot of center 13's orders of meal 1885 (the most popular meal at center 13) over time. This particular pair of 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.

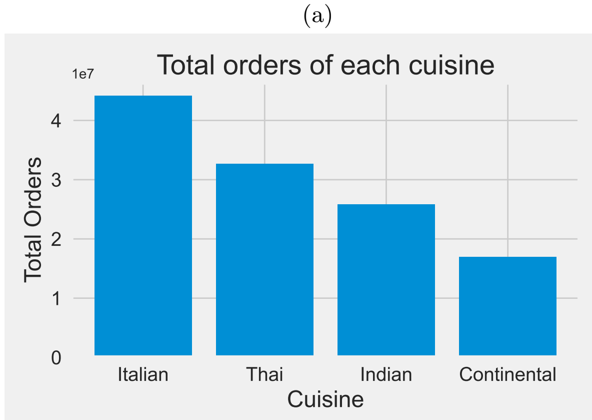
4 Models

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.

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.


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

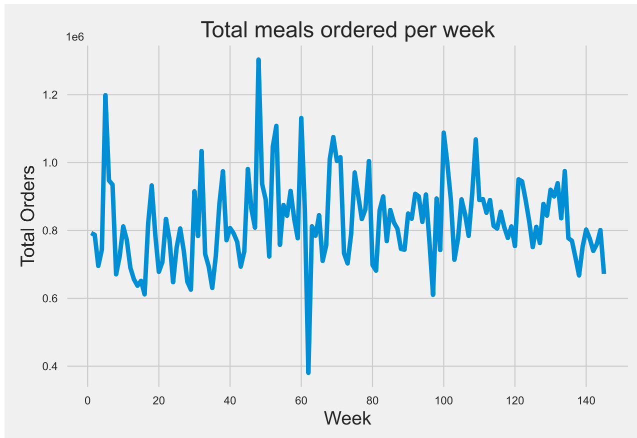
(a)


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

(b)

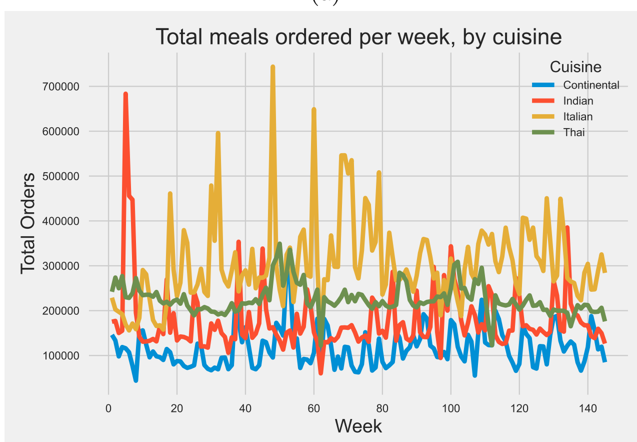

Total orders of each meal
1e7
1.0
0.8
0.6
0.4
0.2
0.0
Total Orders
Meal ID

(c)



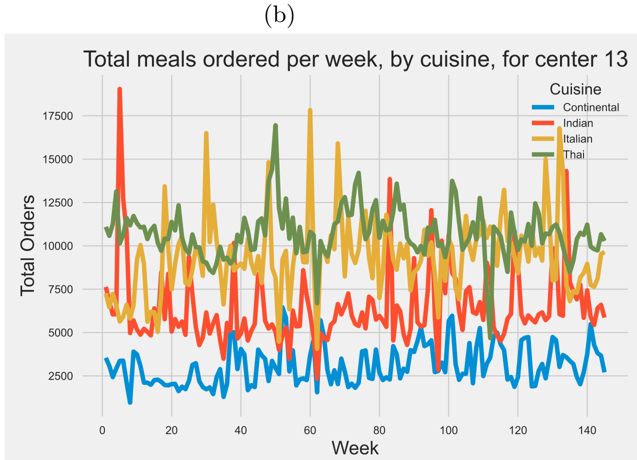
Total meals ordered per week
1e6
1.2
1.0
0.8
0.6
0.4
Total Orders
0
20
40
60
80
100
120
140
Week

(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

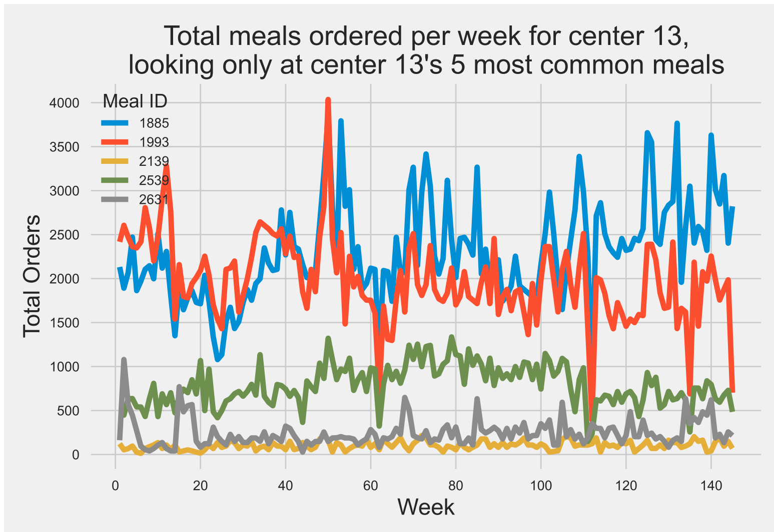


Figure 4

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.

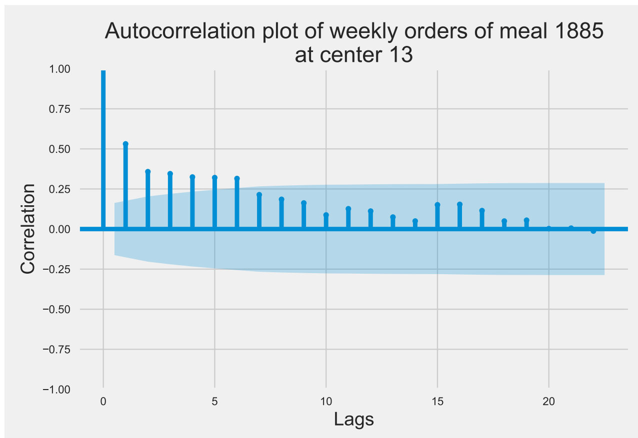
Next, I added lag data (up to six lags) such that each entry additionally includes the number of orders of the entry's meal from the the entry's center's from the previous six weeks. While not every center's meal orders had six significant lags, I decided to include the maximum relevant information. 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.).

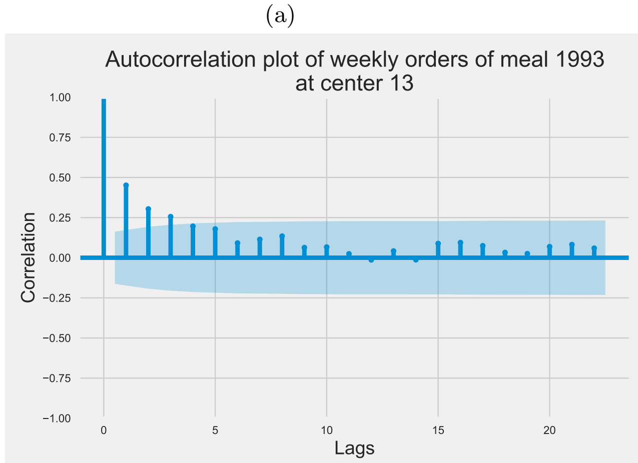
Because the exploration process did not lead me to believe the order history was seasonal, I did not include seasonal lag terms nor encode the week to be seasonal with any modulo, sin-cos, or other encoding.

Once the data was processed, I used both a linear regression and an XGBoost model for two 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 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.


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

(a)


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

(b)

Figure 5

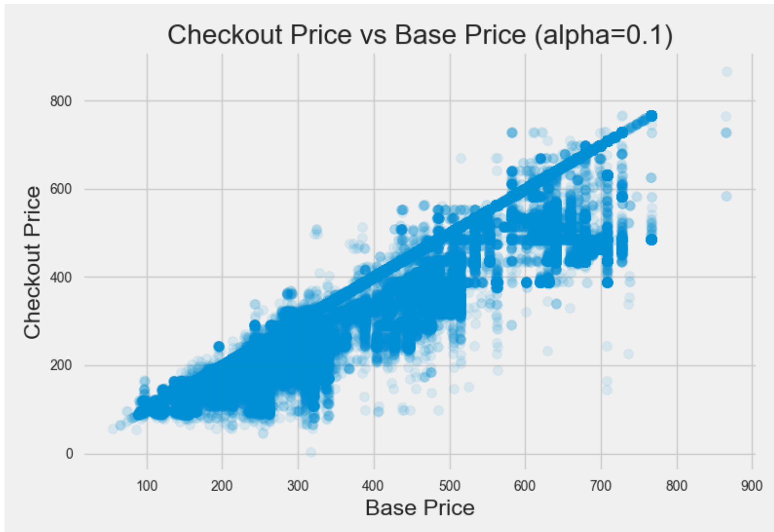


Figure 6

The XGBoost model using the full set of data used 350 trees, which provided a good balance between model simplicity and performance, and the ensemble XGBoost models used 50 trees per individual model.

I'll note that I also tried a model which included no lag information, and an ensemble model split by meals and an ensemble model split by center-meal pairs. None of these structures proved effective enough to further consider them, widening the already wide range of models to consider.

5 Performance

The first thing to note is that the linear regression ensemble model essentially failed, providing a test-set RMSE (root mean square error) over 25 trillion, while the other models had RMSE values of around 200. It behaved fine for some centers, but its predictions exploded for others. I discarded this model as it seemed no amount of playing with the variables fed into it would bring it back to reality. This leaves us with the baseline model, the Linear Regression model, the XGBoost model, and the XGBoost ensemble model.

For the Linear Regression model, almost all variables had significant p-values. The only variables that did not were a few elements of the one-hot encodings of categorical variables. Because it makes the model easier to understand, I decided to keep all of the categorical entries in spite of their high p-values.

I'll report the performance of these models with two key metrics: RMSE and R2 (r-squared, the coefficient of determination). Table 4 shows how the models performed on the test set in these metrics. As expected, the baseline

Model	RMSE	R2
Baseline	248.96	0.54
Linear Regression	213.45	0.66
XGBoost	172.54	0.78
XGBoost Ensemble	180.11	0.76

Table 4: Model performance

performed the worst of the bunch and the XGBoost models out-performed the Linear Regression and performed similarly to each other.

Let's inspect the performance visually on the same center we explored in Section 3. Figure 7a shows all of the model predictions of center 13's meal 1885 demand. The XGBoost models both indeed behave very similarly, while the linear regression does about as well as the baseline model. Figure 7b shows all of the model predictions of center 13's meal 1993 demand. Again, the XGBoost models behave similarly with the linear regression's model a little closer to reality, while the baseline prediction is barely in the ballpark. Lastly, Figure 7c shows the model predictions of the 1855 meal at center 55, which has much lower sales for this meal. Its a toss up between which XGBoost model is better (both track the actual solution quite well), but the baseline and linear regression models do quite bad, only barely remaining in the same ballpark.

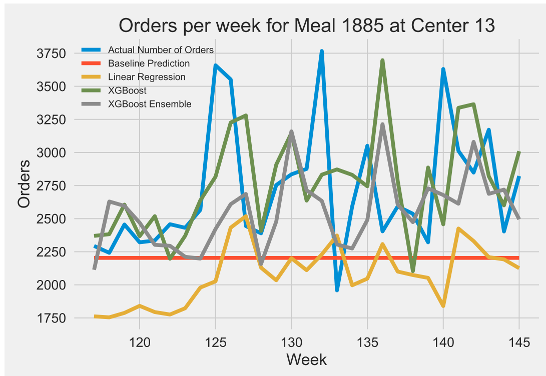
Generally the XGBoost models got pretty close to the true values across many centers and meals, but could not accurately capture the randomness of the large spikes (up or down) in the weekly orders. This is rather expected as unseen data is bound to have influences that do not appear in the training data.

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

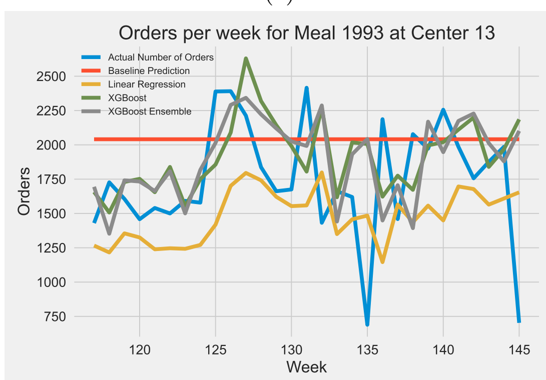
Figure 8a shows the XGBoost (non-ensemble) model's importance of the top 20 features. The non-categorical variables are all more important than the categorical variables. Beceause there are so many categorical variables, this is not too surprising. Similarly, Figure 8b shows the XGBoost ensemble model for center 13's top 20 feature importances, which tells the same story as before. The importance values are lower due to the smaller amount of trees used in the model, but the structure is the same. Both of these models heavily relied upon the lag features, with even the sixth lag being more important than both variables indicating whether the meal was advertised.

6 Conclusions

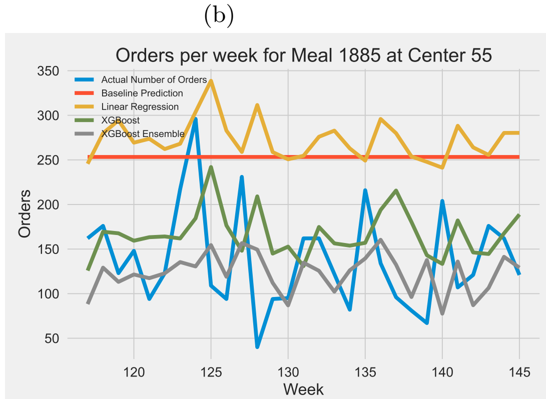
I analyzed a multi-region, multi-product sales history dataset with the aim of forecasting future sales. I constructed a small suite of models informed by the analysis and evaluated them on unseen data. The baseline model, formed from the assumption that the center's average meal sales over a given period,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


Orders per week for Meal 1885 at Center 13
Actual Number of Orders
Baseline Prediction
Linear Regression
XGBoost
XGBoost Ensemble
Orders
Week

(a)

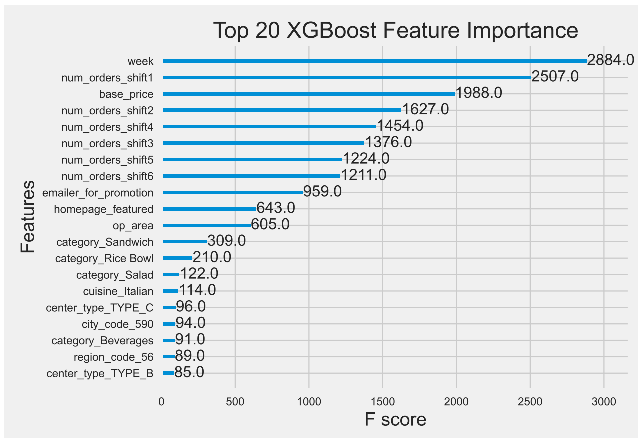

Orders per week for Meal 1993 at Center 13
Actual Number of Orders
Baseline Prediction
Linear Regression
XGBoost
XGBoost Ensemble
Orders
Week

(b)

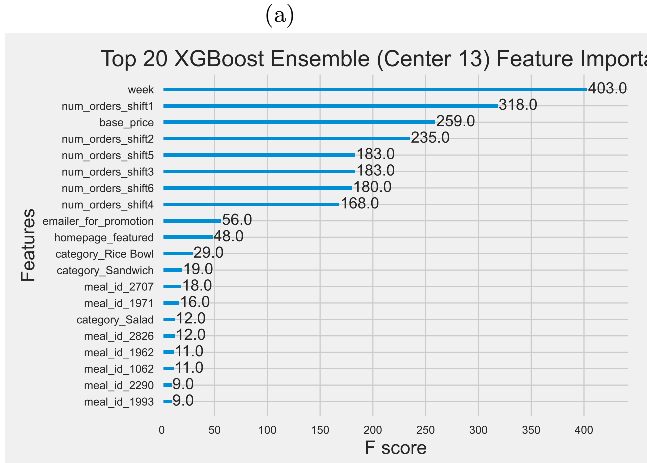

Orders per week for Meal 1885 at Center 55
Actual Number of Orders
Baseline Prediction
Linear Regression
XGBoost
XGBoost Ensemble
Orders
Week

(c)

Figure 7


Top 20 XGBoost Feature Importance
week
num_orders_shift1
base_price
num_orders_shift2
num_orders_shift4
num_orders_shift3
num_orders_shift5
num_orders_shift6
emailer_for_promotion
homepage_featured
op_area
category_Sandwich
category_Rice Bowl
category_Salad
cuisine_Italian
center_type_TYPE_C
city_code_590
category_Beverages
region_code_56
center_type_TYPE_B
2884.0
2507.0
1988.0
1627.0
1454.0
1376.0
1224.0
1211.0
959.0
643.0
605.0
309.0
210.0
122.0
114.0
96.0
94.0
91.0
89.0
85.0
Features
0
500
1000
1500
2000
2500
3000
F score

(a)


Top 20 XGBoost Ensemble (Center 13) Feature Importa
week
num_orders_shift1
base_price
num_orders_shift2
num_orders_shift5
num_orders_shift3
num_orders_shift6
num_orders_shift4
emailer_for_promotion
homepage_featured
category_Rice Bowl
category_Sandwich
meal_id_2707
meal_id_1971
category_Salad
meal_id_2826
meal_id_1962
meal_id_1062
meal_id_2290
meal_id_1993
403.0
318.0
259.0
235.0
183.0
183.0
180.0
168.0
56.0
48.0
29.0
19.0
18.0
16.0
12.0
12.0
11.0
11.0
9.0
9.0
Features
0
50
100
150
200
250
300
350
400
F score

(b)

Figure 8

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

I evaluated both a linear regression and XGBoost model both using the entire dataset and in an ensemble form with one 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.