Food Delivery
Demand Forecasting

Source: https://unsplash.com/photos/phUtWl8RyFE

Written By: Sayesha Aravapalli, Prathik Ullur, and Thirumurugan Vinayagam

Abstract

Food delivery and restaurants benefit from forecasting food demand since it increases profits by reducing uncertainty in labor and food waste which are the two largest costs for restaurants. The data for this project was obtained from this competition. This project tries to forecast weekly food deliveries for one company using historical data. We found the best model to forecast demand is a dynamic regression model which beat the baseline average model by over 40%. If implemented this model would drastically reduce labor and food costs for our company by creating more certainly in demand meaning our food delivery client could hire fewer drivers and reduce food waste.

Business Understanding

Food delivery and restaurants benefit from forecasting food demand since it reduces uncertainty and waste increasing margins for the industry. Restaurants in particular need around eighty percent filled-capacity to be profitable and many have not started or partnered with delivery services. By helping restaurants forecast weekly demand we aim to increase the net profit for the industry.

The largest benefit of food demand forecasting is the reduction of inventory, or food waste in the restaurant industry. Food is the highest cost for a restaurant, especially perishable food with a low shelf life. Therefore, reducing food waste has a large environmental and monetary effect for a given restaurant. It also has a marketing benefit, depending on the city the company is located in, since it can be marketed as a green business. Forecasting also helps with understocking since either too much inventory or not enough inventory can lead to customers choosing another restaurant.

Forecasting sales in a given week can help with labor scheduling and cost. The restaurant industry employees around fifteen million people in the US. Since many workers in the industry are part time or depend on hours set around a week to a month ahead some labor cost can be reduced if demand is predicted to be low in a future week or increased if the demand is expected to spike over the average orders.

Reducing uncertainty is a benefit for forecasting in any industry. However, uncertainty in the restaurant and delivery industry has an effect on real lives, since many service jobs in the US are restaurant jobs. Excluding food and labor costs, which are the two largest costs in running a restaurant, reducing uncertainty can help restaurant owners arrange payroll, utilities, marketing and expansion plans. The savings from forecasting demand can be used for expansion or to add new menu items to draw more customers.

Forecasting food demand has a direct effect on restaurant profits by reducing food and labor costs and reducing uncertainty for other costs. Implementing the forecasting methods in this paper will help the restaurant and food delivery industries manage profits. This project focuses on one food delivery client, such as Favor, which delivers food in many different cities through distribution networks and fulfillment centers, i.e. local restaurants.

Data Understanding

The data consisted of 489121 rows and 88 columns. Each row represents a delivery with a unique order ID. The columns provide information regarding the region, city, food type, cuisine, price, promotional index and week number of the delivery. A detailed description of the data columns is provided in the appendix.

We first determined the most popular cuisine ordered. From fig(i), we see that the most popular cuisine ordered is Italian followed by Thai, Indian and Continental.

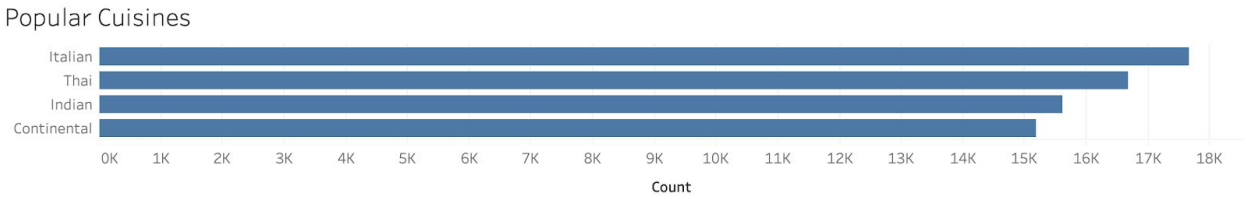

Fig (i)

We further looked at popular types of food to order and from fig(ii), we see that the most popular food item ordered is Beverages. The next popular food items are Rice Bowls, Salads, Sandwiches and Pizzas with their total counts being very close to each other. While starters, seafood, and biryanis are the most unpopular food items to order.

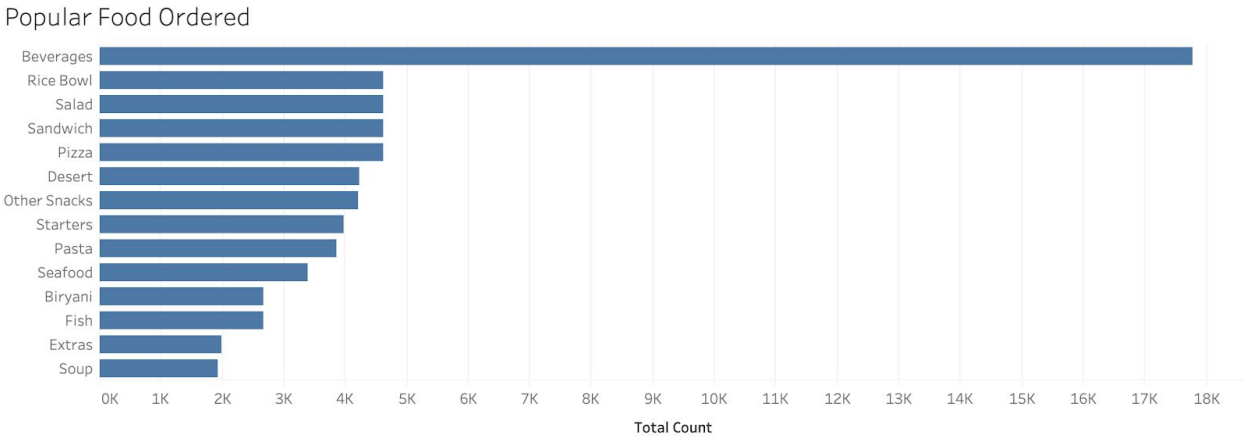

Fig (ii)

Next, we looked at popular cities with most orders and from fig(iii), we see that orders in city18 and city9 are overpowering all other cities while city30 follows closely. The next few cities have an average order count range between 1500-2000 while the remaining cities have a relative lower order count with a range of 600-1000.

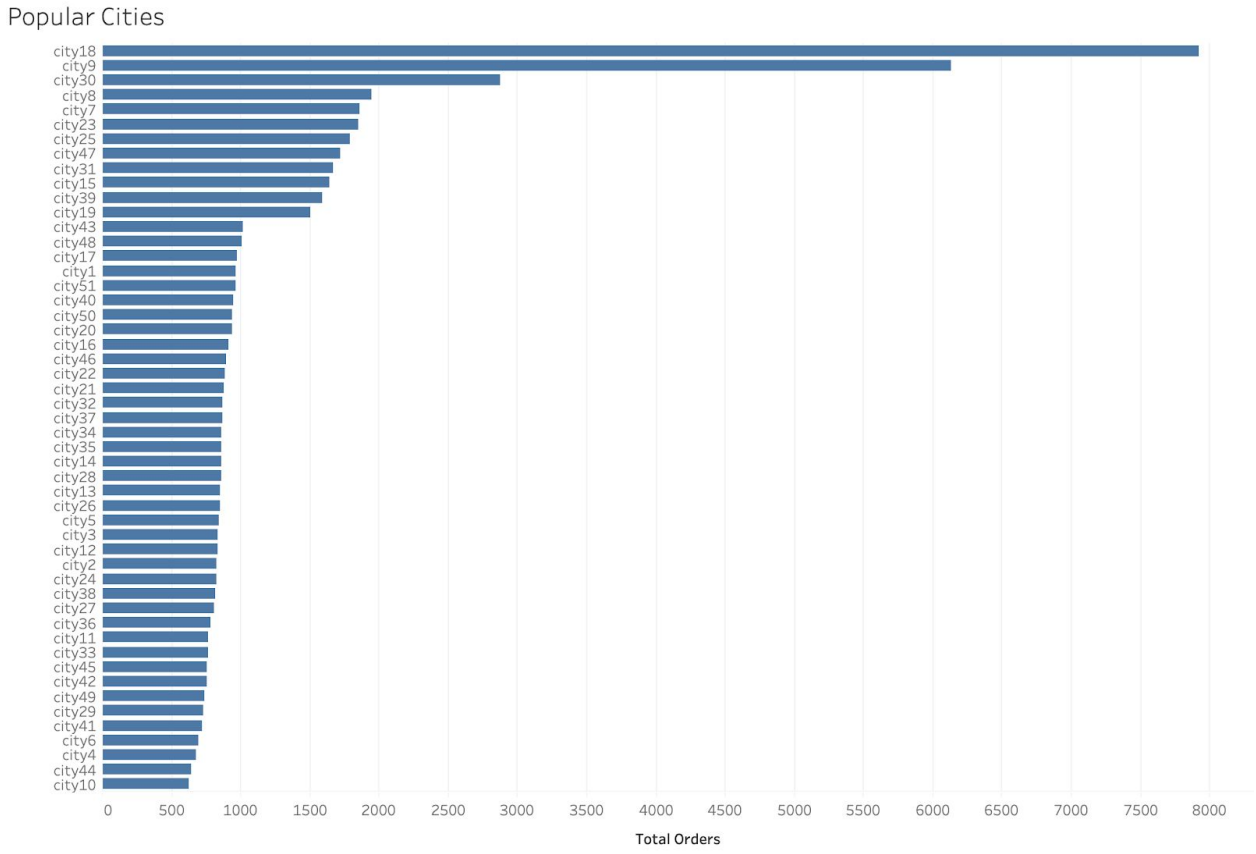

Fig (iii)

We further looked at popular regions to order and from fig(iv), we see that the most popular region is region4 followed by region2, region6 and region7. The most unpopular regions with the least order counts are region1, region8 and region4.

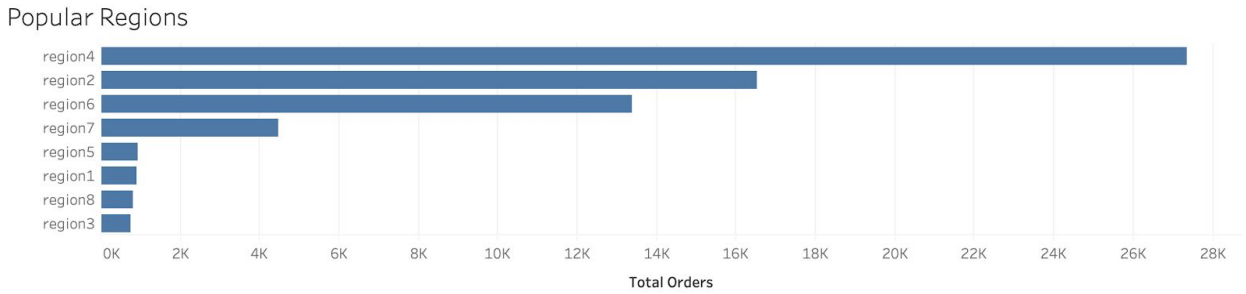

Fig (iv)

Let us take a look at the trend of orders per week. From the plot, we see that the graph spikes in week 50 and dips unusually low in week 65. From the graph, the time-series looks stationary with no trend seen as such.

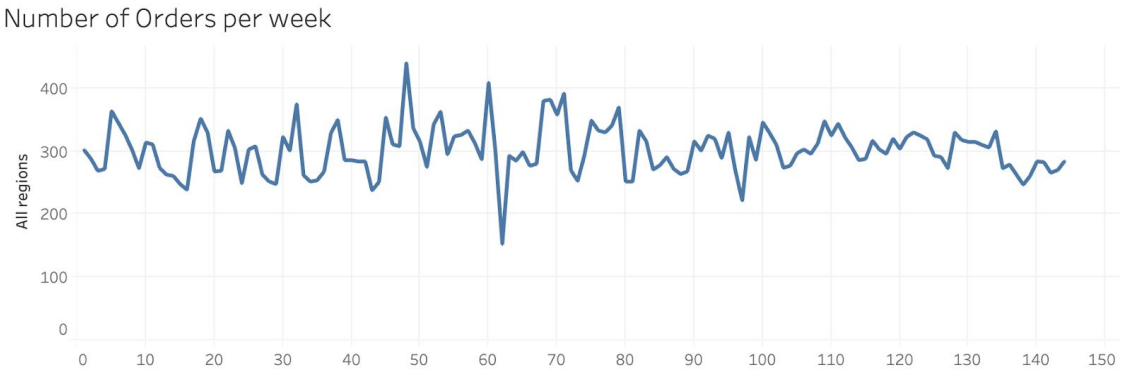

Fig (v)

We then decided to look at the trend of orders in each region to determine if they follow the same trend as seen in the above plot. From Fig (vi), we see that the popular regions somewhat follow the series especially the dip in week 65.

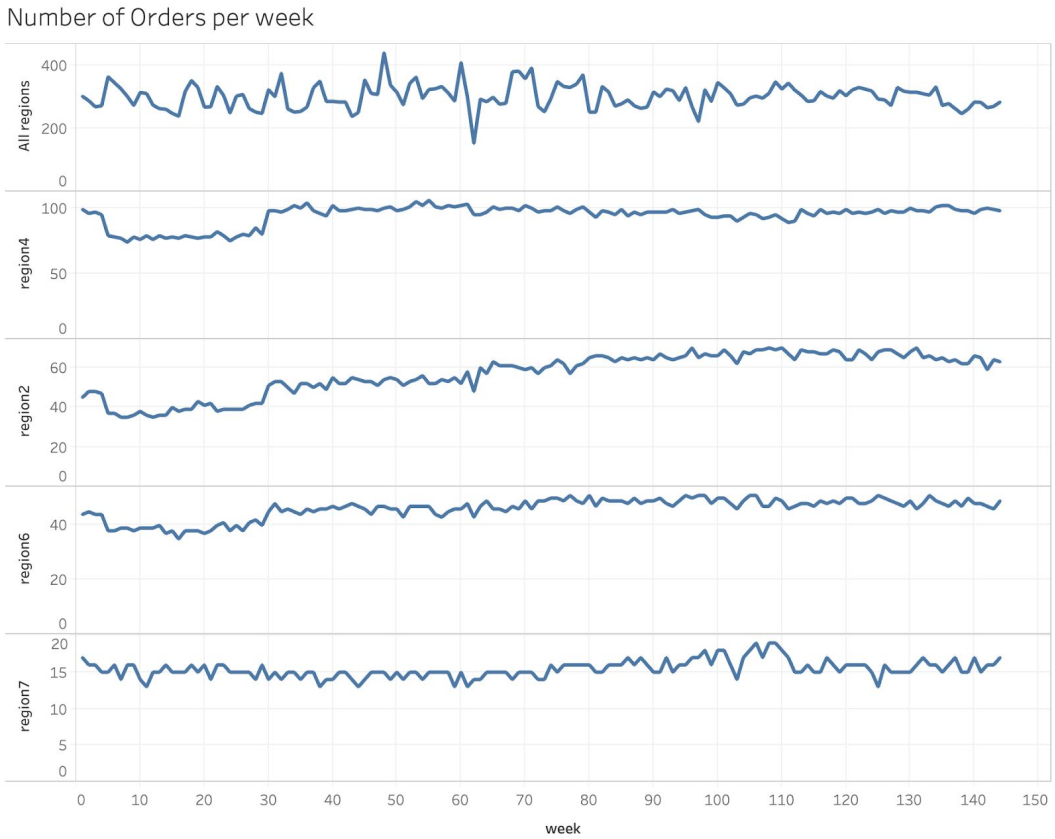

Fig (vi)

Data Preparation

The data was obtained from a competition on AnalyticsVidhya for Food Demand Forecasting. After completing the exploratory data analysis, the dataset was split into a training excel file, with week one through one hundred forty five, and validation file, with week one hundred forty six to one hundred fifty-five. The data also included separate files with meal information and fulfillment center information.

The training, validation, meal information, and fulfillment center files were loaded into Python to merge all the data into a training and validation set. The training data was merged with the meal information file, using meal_id as the merge key, and the fulfillment center file, using center_id as the merge key. The meal_id and center_id columns were dropped from the training dataset to output the final large and wide training dataset. The same steps, with the same merge keys, were taken for the validation data to output the final large and wide validation dataset.

In both the validation and training sets, the regions and cities were coded as numeric values which we switched to string values to be able to call each city or region as a dummy variable. To change the numeric value to a string we used a dictionary of numeric keys with string values then renamed each region or city sequentially as they appeared in the data. The complete dataset, validation and training, had eight regions and fifty one unique regions and cities respectively. The numeric values were changed to sequential strings by using the replace function with the new value put in place of the numeric value.

The third transformation was coding the categorical variables as categorical in Python using the astype function. There were five categorical variables in the dataset: the food category, cuisine, city, region, and the fulfillment center type.

Using the get_dummies function from the Pandas library all of the categorical variables were changed into dummy coded variables then concatenated with the validation and training sets respectively. The final training set, after dropping the original five categorical variables, had eighty-eight variables and over four hundred and fifty thousand observations while the validation set had eighty-seven variables, since the number of orders is unknown, and over thirty-two thousand observations.

The training and validation datasets were then grouped by week since our project was to predict overall weekly demand. Now, with the grouped training and validation set we created models to predict the number of orders, demand, for a given week.

Modeling

The naive model predicts the mean number of orders over all the weeks as a baseline. The overall mean is used as the prediction for the next week since this is a likely way that restaurants currently schedule labor and order materials. The model was made as a comparison point to the other models we made for the project.

As another preliminary model we created an AR 1 model, which is a model where the order size of the next week is the number of sales of the previous week. This is also a possible way that restaurants and food delivery orders now. Although it was not a complex model, the lag model was another baseline for us to compare against and was a model that we can calculate the monetary value of our project if we had the financial impact of inventory and labor costs.

For a regression model we ran forward stepwise regression in R since we have many variables and regression is the simplest model to create. The null model is that the number of orders is predicted by only an intercept while the full model has thirty-five predictors. R automatically iterates by adding a single predictor to find the best model using the AIC score. The best regression model output has nine predictors: promotional emails sent, operational area, region five, featured homepage, continental cuisine, checkout price, region one, beverages, and extras. All of the predictors have a p-value of less than 0.5 meaning they are all significant and the model passes all validity tests.

After creating the simple regression we decided to run an ARIMA model, since the data is a time series. We looked at the ACF and PACF plot to determine the order of the ARIMA model. Since the time series of the number of orders was already stationary, there is no need for differencing the series. Since the d term for the order of the series is zero, then p and q can be determined from a loop by iterating through every value between zero and the highest significant value given by ACF and PACF plots. The highest p and q values were eight meaning we looped through eighty-one different models. The final order of the model was taken as the model with the lowest RMSE value. The best ARIMA model from the loop is an ARIMA model with an order of eight, zero, seven.

Seeing that the ARIMA model also worked well on the sales data, we decided to use the automatic dynamic regression, which is a regression model with ARIMA errors, to find important variables and a starting point for a full dynamic regression. We added the nine variables from the forward stepwise regression to the automatic dynamic regression, after changing all of the variables to time series objects. From the output we decided to use only the

five most important variables. The importance of a variable was calculated as the absolute value of the t-statistic, which is the coefficient estimate divided by the standard error estimate given by the summary of the model. The five most important variables were promotional emails sent, featured homepage, continental cuisine, checkout price, and beverages. These variables were put into the dynamic regression where we manually tuned the ARIMA parameters.

The five variables from the automatic dynamic regression, promotional emails sent, featured homepage, continental cuisine, checkout price, and beverages, was the starting point of the dynamic regression. To tune the ARIMA model we checked for stationarity and we used the ggtsdisplay function to inspect the ACF and PACF plot. Although the sales time series was stationary we decided to difference the time series to create a stronger delineated ACF and PACF cutoff. The ARIMA model has an order of seventeen, one, eighteen since the ACF cutoff was eighteen, there was one difference, and the PACF cutoff was seventeen. The dynamic regression model had an extremely low RMSE and was the final model we created.

Evaluation

The RMSE of the naive model is 125236.2. All of the other models will be compared using this baseline error. The lag model has a RMSE of 120976.18 which is a 3.4% decrease in error than the naive model. Although the model is slightly better, the reasoning behind this model is as another baseline depending on the policy of the food delivery company.

The ARIMA model has an RMSE of 105998.30 which is a 15.4% decrease from the error of the baseline naive model and a 12.4% decrease from the error of the lag model. The ARIMA model only uses the order data with lags and moving averages to predict the sales for the next week. The model can be used when the only data collected is the sales data meaning the cost of collecting data and running this model is low compared to other models. However, because of its simplicity the model has the worst error with respect to the models which are not the baseline.

The automatic dynamic regression has an RMSE of 92068.35 which is a 26.5% decrease from the error of the baseline naive model. The automatic dynamic regression takes most of its improvement of the forward stepwise regression however the automatic ARIMA has an order of 0, 0, 0 which essentially finds no model for the residuals of the forward regression.

The forward stepwise regression model has an R squared of 51.92% and an RMSE of 86833.95. This is a 30.7% decrease from the error of the baseline naive model. This model is

one of the best models since it is an extremely simple model compared to the other ARIMA models.

The best model was the dynamic regression model with the five most important variables from the automatic dynamic regression. The RMSE of this model is 72362.58 which is 42.2% better than the error of the baseline naive model and is 40.2% better than the error of the baseline lag model. This model has the lowest RMSE and was the best model we created for this data. The dynamic regression takes the RMSE improvement from the forward stepwise regression and adds a manually tuned ARIMA model for the residuals which further improves the model error.

Conclusion

Our project delivered a calculated savings of around 40% given either the naive or lag ordering policy. The savings, as a result of our model, would translate into a realized increase in net profit for our client, a food delivery company.

After looking through our data to see that the trend overall and the trend per region were stationary, through the exploratory data analysis, and that the type of food ordered and the cuisine seemed to be important in sales we ran four different models to help our client reduce their costs in its competitive landscape. The result was that an dynamic regression model, with ARIMA errors, was the best model for predicting the number of orders of the following week. With this discovery we have provided our client value through our project.

Appendix

Data Dictionary

Variable	Definition
id	Unique ID
week	Week No
center_id	Unique ID for fulfillment center
meal_id	Unique ID for Meal
checkout_price	Final price including discount, taxes & delivery charges
base_price	Base price of the meal
emailer_for_promotion	Email sent for promotion of meal
homepage_featured	Meal featured at homepage
num_orders	(Target) Orders Count
city_code	Unique code for city
region_code	Unique code for region
center_type	Anonymized center type
op_area	Area of operation (in km^2)
category	Type of meal (beverages/snacks/soups....)
cuisine	Meal cuisine (Indian/Italian/...)