Recruit Restaurant Visitor Forecasting

Final Report

6th March, 2018

Foodie Analytics

Shipra Sethi
Dong Bing
Daniel Colvin
Subba Muthurangan
Erik Platt

Foodie Analytics
401 N. Michigan Ave.
Chicago, IL 60611

March 6, 2018

Dr. Donald Wedding, CEO
Recruit Holdings Co., Ltd.
Chuo-ku, Tokyo 104-0061 Japan

Dear Dr. Wedding,

Foodie Analytics is pleased to inform you that our development team has successfully accomplished all the project goals laid out at the beginning of the project, and has finalized the concluding deliverable within the original timeline discussed. We completed an in-depth exploratory data analysis to ensure all the intricacies and nuances of the data have been captured and understood. After identifying 5 promising machine learning models/techniques, we chose to proceed with a K-Means with XGBoost model given its high accuracy. Foodie Analytics developed data visualizations and a dashboard to help Recruit Holdings derive immediate value from our efforts. Finally, we developed a mobile application which provides flexibility to utilize our product by those who are constantly on the move.

As laid out in the project goals, this deliverable includes the final recommended model to not only predict the future customer visits to a restaurant, but also define the competitive environment for an existing restaurant and the most optimal location for a new restaurant. In this report, we have provided a detailed section on exploratory data analysis (EDA), model development and validation, data visualizations and dashboard development, mobile and web application development, assumptions, and limitations related to this project, conclusions and our final recommendations. The appendix section provides details on the datasets which were used for this project, source code for model development and web/mobile application, and links on how to access the final product. With this deliverable, we have also provided an Executive Summary to share the findings from this project with the executive team. In addition, we will be providing a high-level summary of the project in the form of a supplemental slide deck by March 18, 2018.

We look forward to having an opportunity to present this final deliverable to you. In the meantime, should you need any additional material for your Board of Directors, please do not hesitate to contact any team member.

Sincerely,
Foodie Analytics

Table of Contents

1. Problem Statement

Restaurant start-ups face a notoriously high failure rate. While the often advertised metric stating 90% of restaurants fail within the first year is exaggerated, it is still far from an easy industry to find success. There are many coordinating factors to drive customers in; an appetizing menu simply isn't enough to guarantee traffic. However, increasing utilization of online reservation systems has offered an opportunity to gain a competitive advantage through customer data.

Recruit Holdings' partnership with Foodie Analytics will leverage this customer data. Using existing datasets within Hot Pepper Gourmet and AirREGI platforms, Foodie Analytics will build detailed analytics and predictive models to inform customer traffic patterns. Understanding when and where customers plan to dine allows the opportunity for efficient staffing, fresh ingredients, and an idealized experience.

2. Data Overview

2.1 Overview

The primary datasets used for data insight generation and modeling purposes come from 2 separate data sources:

- Hot Pepper Gourmet (HPG): Similar to Yelp, users can search restaurants and make a reservation online. This data source contains 2 datasets.
- AirREGI / Restaurant Board (Air): Similar to Square, a reservation control and cash register system. This data source contains 3 datasets.

In addition to above datasets, 2 more datasets are provided, one identifying common restaurants between HPG and Air data sources and another indicating when Japanese holidays occur. We also used historical weather data from the Japan Meteorological Agency (JMA). This dataset contains several weather factors including Precipitation, Average Temperature etc. A summary of the data provided can be seen in Table 1 and a description of each variable is summarized in Table 2:

Category	Description
Number of Sources	2
Total Number of Datasets	8
Number of Unique Records	963,705
Number of Fields	10
Number of Observations with Missing Values	57
Location of Restaurants	Japan
Time Frame of Observations	1/1/2016 - 4/23/2017

Table 1: Summary of Data

Variable	Description
air_store_id	Unique identifier for a restaurant from the “AIR” data source
hpg_store_id	Unique identifier for a restaurant from the "HPG" data source
visit_datetime	Date which the customer visited the restaurant
visitors	Number of visitors who visited the restaurant
reserve_datetime	Date which the customer made a reservation for the restaurant
reserve_visitors	Number of visitors for a reservation
genre_name	Restaurant food type
area_name	City where restaurant is located
latitude	Geographical latitude
longitude	Geographical longitude
holiday_flg	Indicator of days which were a local holiday

Table 2: Data Dictionary

Please refer to section 12.4 in the appendix for details on the data files used for this project.

2.2 EDA Summary

The steps involved in exploratory data analysis (EDA) are summarized below:

- Analyze the components of each dataset to better understand the business goals
- Merge the datasets into one, coherent object
- Identify missing values, potential outliers, and/or missing information
- Aggregate the data from hourly to daily observations
- Explore ways to splice the data to extract the most valuable information

2.3 Tools Used for EDA

R and **Tableau** were used to meet our EDA goals. R is a valuable tool for statistical analysis, model creation, and basic graphics capabilities. Tableau provides more advanced graphics for further insight into the data and a more aesthetical display for the end user.

2.4 Response Variable

The response variable is “visitors”. It is an indicator of how many people visited a restaurant on any given day. This information will be used to predict how many people will visit at a future date.

2.5 Predictor Variables

There are 18 predictor variables used for initial data modeling, as described in Table 3 below.

Predictor Variable	Description
air_res_visitors	Date which the customer made a reservation for the restaurant
air_mean_time_ahead	Number of visitors for a reservation
air_genre_name	Restaurant food type
air_area_name	City where restaurant is located
latitude	Geographical latitude
longitude	Geographical longitude
holiday_flg	Indicator of days which were a local holiday
day_of_week	Day of the week
rank	Store rank based on number of visitors
min_visitors	Min visitors from visitors variable
mean_visitors	Mean visitors from visitors variable
median_visitors	Median visitors from visitors variable
max_visitors	Max visitors from visitors variable
count_visitors	Total count of visitors from visitors variable
month	Month of year
day	Day of month
precipitation	Rainfall on a given day
avg_temperature	Avg temperature on given day

Table 3: Predictor variables with descriptions

3. Exploratory Data Analysis

3.1 Overview

Foodie Analytics has conducted extensive data review to ensure the quality and integrity of the data. Our process looks to ensure the data used for model development is complete, accurate, and reliable. This assessment includes determining the number of missing values, detection of any outliers and their potential impacts, as well as any anomalies within the historical data. The goal is to address and resolve any data quality issues leading to significant adverse impacts on model feasibility.

3.2 Merging Datasets

Even though the given data came from two different sources, HPG and Air, there were commonalities in the fields which made it possible to merge them together. Three of the datasets contained the “store_id”, “reserve_datetime”, “visit_datetime”, and “reserve_visitors” fields. Two more datasets contained the “store_id”, “genre_name”, “area_name”, “latitude” and “longitude” of many restaurants. One dataset contained information on when holidays occurred, and another dataset indicated which restaurants were in both data sources (but had different IDs). Finally, we used weather historical data that contained “precipitation” and “avg_temperature”.

Using “store_id”, we were able to combine these 7 datasets into one coherent object. For the weather data, we used the nearest location to merge with all observations. In the training dataset, we only kept the records which contained a value for the “visitors” variable (i.e. 213,510 records). The final object contained the following columns: “air_store_id”, “hpg_store_id”, “visit_datetime”, “visitors”, “reserve_datetime”, “reserve_visitors”, “genre_name”, “area_name”, “latitude”, “longitude”, “holiday_flg”, “precipitation” and “avg_temperature”.

3.3 Missing Values

Once we merge the HPG and Air data sources together, we perform missing value analysis on this single data source. The goal of this exercise is to understand the nature of the missing values and assess their impacts on model performance. We want to make sure missing values do not compromise the integrity of the data or introduce significant bias in prediction results. Table 3 below shows the data fields with the number of missing values in the combined dataset.

Variable	**Missing value count**
hpg_store_id	51,232
reserve_datetime	19
reserve_visitors	19

Table 4: Missing Values

The field with the most missing values is “hpg_store_id”. Unlike the Air data source, the HPG data source only has reservation information and not the actual visitor information. Since our response variable is the number of visitors and not reserve visitors, most of the records in HPG data source cannot be tied back to the Air data source by store ID when we combined the data sources. In other words, records which have a missing “hpg_store_id” will not be considered during the model training phase due to its lack of actual visitor information.

We also see a small number of missing values in “reserve_datetime” and “reserve_vistors”. Since the count is less than 0.005% of the total sample size, removing these values is the best course of action.

3.4 Outliers

In this phase, we examine extreme values which deviate from other observations and could have undue influence on model performance. We took the univariate approach by looking at the distribution of the target variable.

The histogram in Figure 1 below shows there are a number of potential outliers in our response variable, as extreme values peak at a visitor count of 239. The top five highest values are 189, 199, 205, 216, 239. While these values are questionable and worth further investigation, it is plausible to expect an extreme number of visitors in certain instances such as company outings in holiday seasons. For this reason, we will not remove these outliers from the dataset for the initial model. The density distribution shows 90 percent of the samples consist of party sizes of 50 or less, indicating there are low occurrences of extreme values.

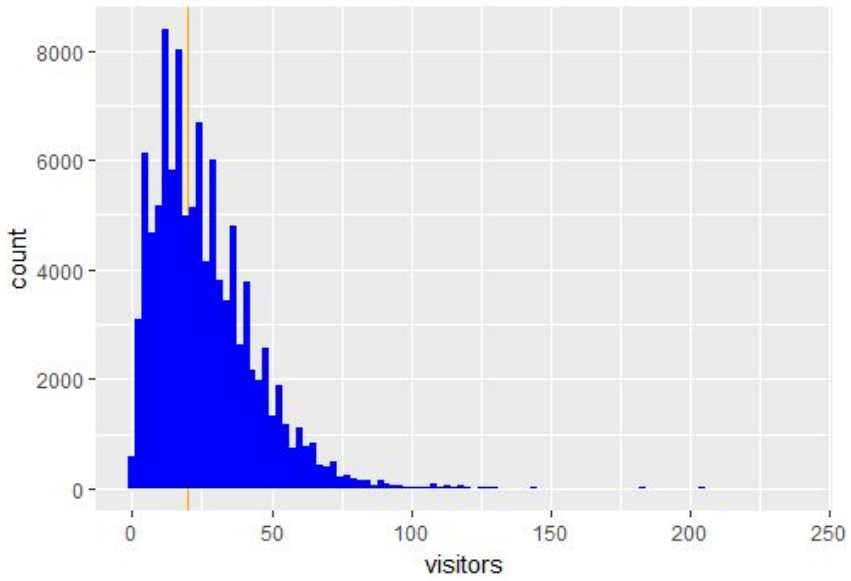


Figure 1: Histogram of the response variable

3.5 Data Trends and Business Insights

Foodie Analytics looks for important indicators in historical trends that can potentially improve model performance. Figure 2 below shows the historical trend of the number of reservations in the data source. As the trend line indicates, the number of reserved visitors is clearly influenced by seasonality. The fluctuations are in an orderly fashion, based on day of week. We also see a large increase in reservations leading up to Jan 2017, potentially related to the seasonal factor of New Year's custom in Japan. There also seem to be a higher level of activities in 2017 compared to 2016, which can be attributed to the increase in the usage of the reservation system. While many of the restaurants were present throughout the entire dataset, new restaurants entered the dataset at a steady pace. Starting October 2016, the rate at which new restaurants entered began increasing, helping to explain the artificial growth we are seeing in "all_visitors" counts.

In the upper plot of figure 2 (above), the large spikes towards the end of 2016 correspond to celebration over three main holidays: the Emperor's Birthday (December 23rd), Christmas (December 24th and 25th), and New Year's Eve (December 31st). In addition, there are four national holidays in April-May deemed the "Golden Week": Showa Day (April 29th), Constitution Day (May 3rd), Greenery Day (May 4th), and Children's Day (May 5th). This period, displayed in

figure 2 as the large decrease in reserved visitors at the end of the upper plot, shows the opposite effect of previous holidays in restaurant reservations; fewer people make restaurant reservations during the Golden Week. These findings will help us improve our final models using the “holiday_flg” variable.

The bottom left chart shows the overall distribution for time of the reservation; it is no surprise most reservations are made for dinner in the evening hours. There is also a curious relationship between reservation time and visit time, as shown in the bottom right chart. There is a rough 24-hour pattern to be identified between the reservation and the visit time, while it is still most common to book few hours right before the dinner.

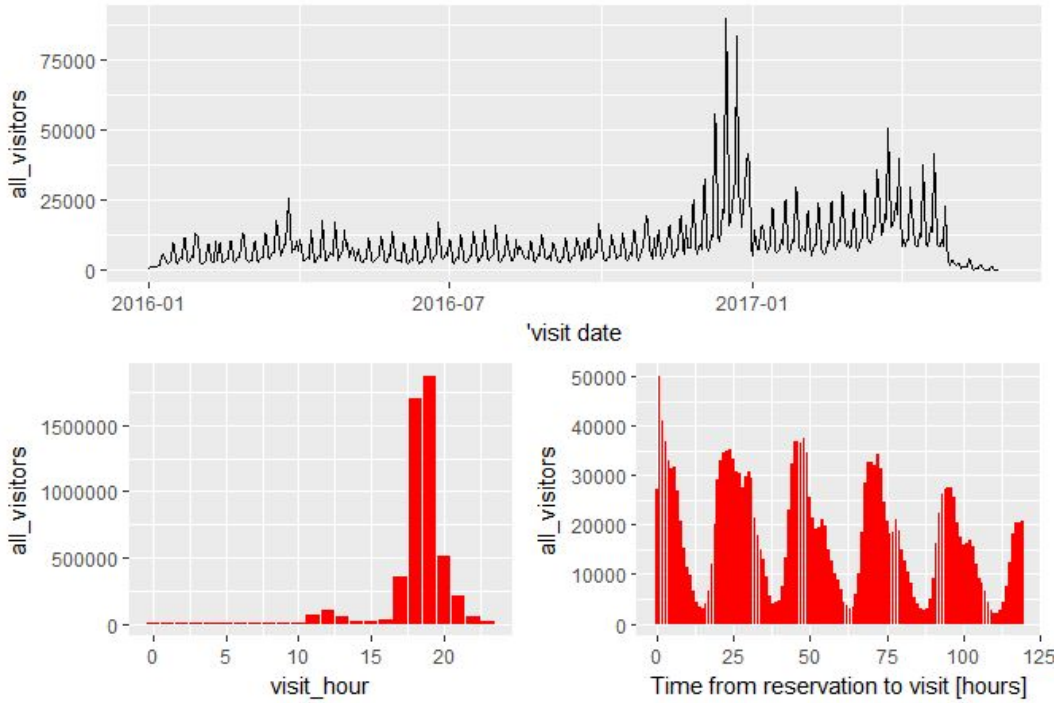


Figure 2: Trends of reserved visitors

Next, we examined the relationship between reserved visitors and actual visitors through the lens of a scatter plot, given these are the only two numeric variables in our dataset. As one can see from the Figure 3 below, most of the points fall above the line, indicating there are more actual visitors than reserved visitors on a given day. This observation makes sense because walk-in visitors who did not make a prior reservation are normally accepted in restaurants. The data points which fall below the line indicate people made reservations in advance but did not end up visiting. Again, this is within our expectations. There seems to be a low level of correlation between reserve and actual visitors, as a result, we will use reserved visitors as one of the predictor variables in our model.

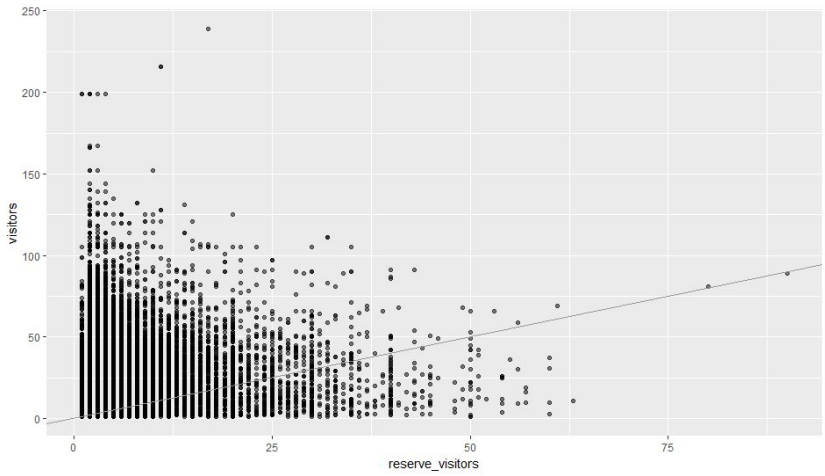


Figure 3: Reserved vs Actual Visitors

4. Data Transformation

Although the data consists of hourly observations, the goal is to predict the total daily visitors. To do this, a new variable, “day_of_week” was created to represent the day of each record, and the response variable was then summed for each day. The resulting object contains a date, the number of daily visitors, the day of the week, and an indicator of whether there was a holiday on that day or not. The following derived variables were created from “visitors” response variable: min visitors, mean visitors, median visitors, max visitors and count visitors.

We also provided the option of specifying a specific restaurant, genre, or area, and then aggregating the data to show daily visitors to the specific criteria. The resulting object would, therefore, contain the total number of visitors on a given day only for the specified restaurant, genre, or area, the date, and day of the the week which they visited, and the holiday flag indicator.

5. Model Development and Validation

5.1 Model Development

The model development started with understanding the basic model building process:

1. Model selection
2. Model fitting
3. Model validation

Figure 4 below explains the basic model methodology we used for this project. We split the data into training and validation set to develop the model and validate against the model. We used RMSLE to measure our accuracy of prediction.

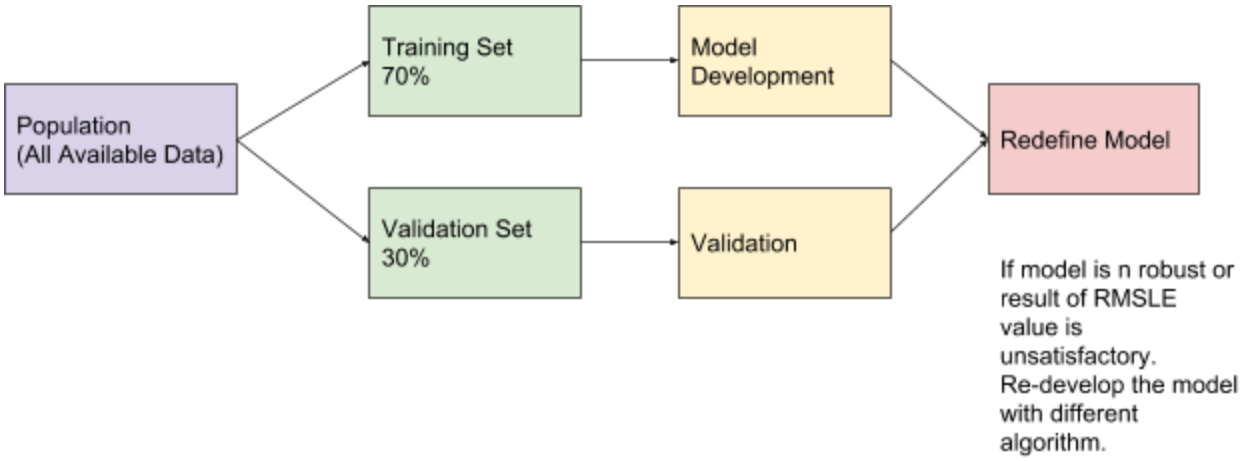


Figure 4: Model Development Methodology

Table 5 below shows the time series and machine learning model/techniques the development team explored to predict the number of visitors.

Model/Technique	**Description**
XGBoost	Extreme Gradient Boosting with XGBoost
H2O	H2O is open-source software for big-data analysis
LightGBM	Light GBM is a boosting framework based on decision tree algorithm
K-Means with XGBoost	k-means clustering is a method to partition n observations into k clusters
ETS	Econometric Time Series
NNETAR	Neural Network Model
ARIMA	AutoRegressive Integrated Moving Average
Random Forest	Decision Tree, ensemble approach

Table 5: Model algorithm details

The development team used Random Forest model as a baseline model. A random forest is an ensemble approach that can also be thought of as a form of nearest neighbor predictor. The random forest starts with a standard machine learning technique called a “decision tree” which, in ensemble terms, corresponds to our weak learner. In a decision tree, an input is entered at the top and as it traverses down the tree, the data gets bucketed into smaller and smaller sets.

We set parameter “number of trees (ntree)” at 500 and “number of parameter sample (mtry)” at 2, which should represent a relatively simple model. Due to the large size of the dataset, it took almost 12 hours to completely train the model, thus it might not be feasible to implement this model. Even though the Random Forest model may be infeasible, it revealed which variables are important (Figure 5, below). This information helps explain which variables carry the most weight in more complex models.

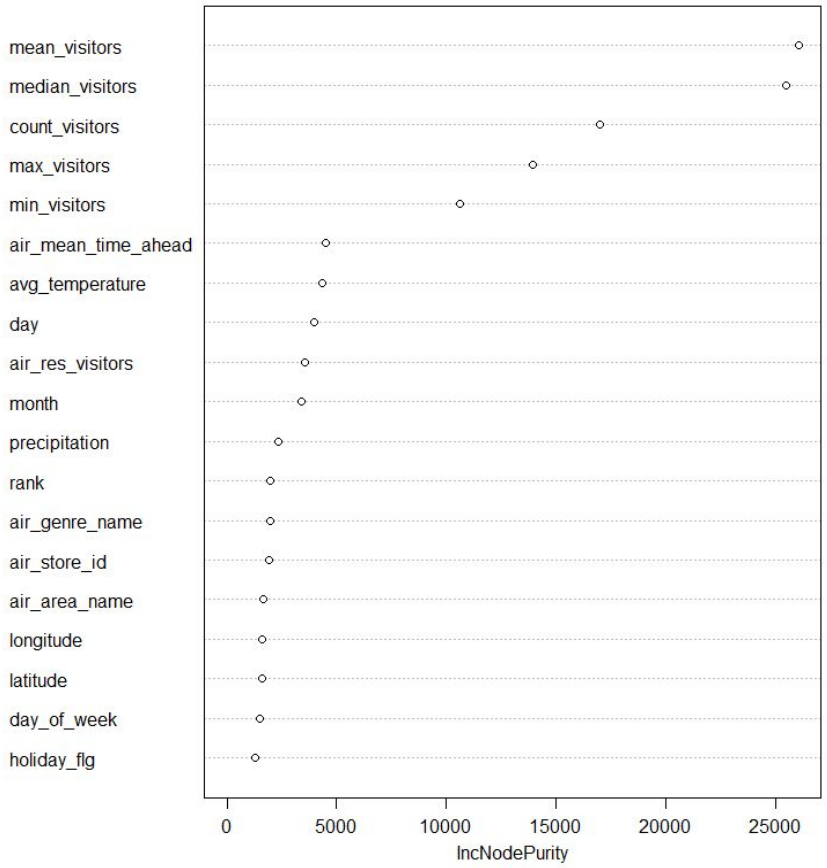


Figure 5: Variable importance plot of predictor variables

Additionally, the large amount of customer data suggested the use of machine learning technology because of the following reasons:

1. Since the number of records is close to 1 million, we can use machine learning for better results.
2. There is no linear relationship between response and predictor variables, except we see a low level of correlation between reserved visitors and actual visitors (as shown in Figure 3 above). The correlation map in Figure 6 below suggests there is no correlation between variables other than derived variables from “visitors” response variable.
3. The big advantage of using machine learning is to capture all patterns beyond any boundaries of linearity or even continuity of boundaries.

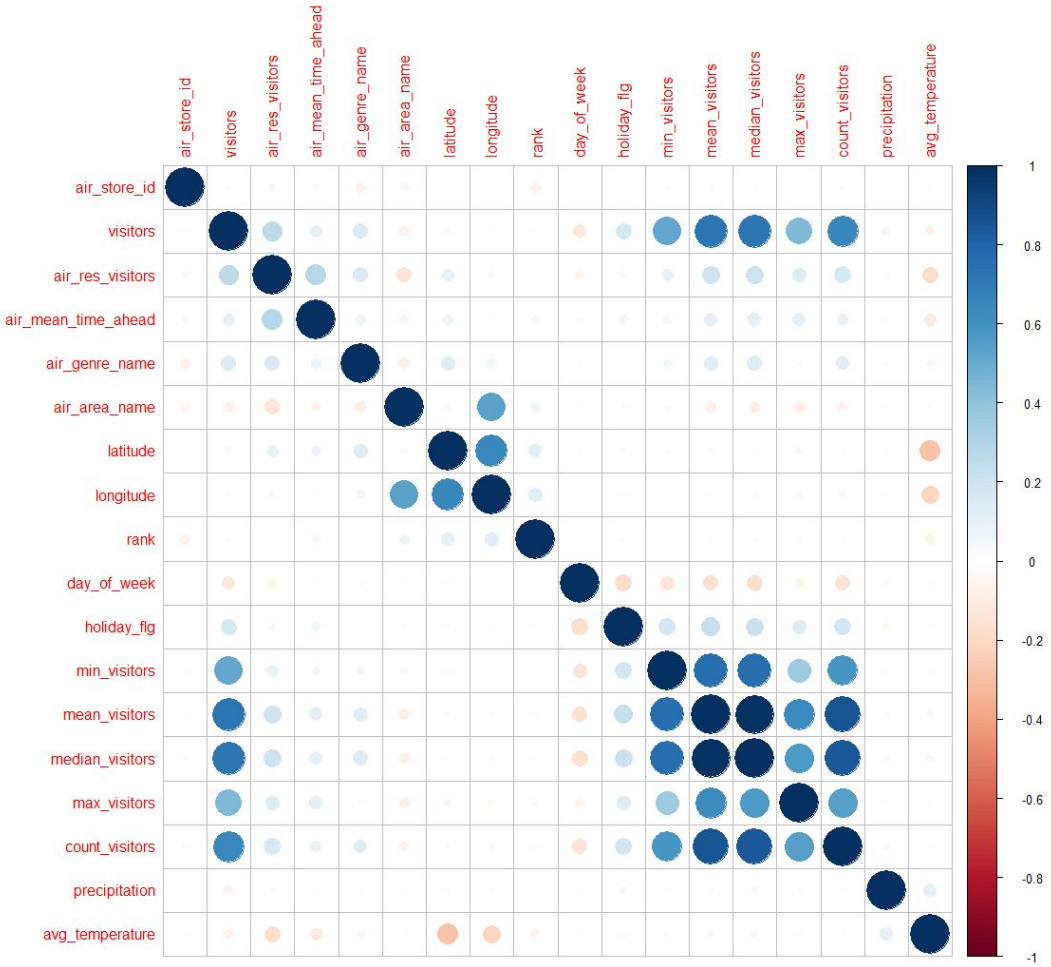


Figure 6: Correlation matrix

For final model development, we focused on 5 types of machine learning models/techniques. These models/techniques were tested, and multiple variations of each were created using different parameter combinations and values. The following sections explain each of these models in more detail.

5.1.1 eXtreme Gradient Boosting (XGBoost)

The development team developed an XGBoost model, which uses the extreme gradient boosting technique. It is a relatively new technique in the data science toolkit and is proving to be dominant in many complex problems with large datasets. The development team chose XGBoost for many of the same reasons why it has become widely popular – it is computationally efficient and it is less likely to overfit the data. This dataset is quite complex due to having many variables, which appeals to the use of XGBoost. Boosting is a statistical method to improve the predictive power of machine learning models using an iterative approach to model building. Table 6 below shows the parameters which were used to fine tune the accuracy of the model.

Parameter Name	Description	Value
objective	"reg:linear" --linear regression	"reg:linear"
eta	Step size shrinkage used in update to prevents overfitting. After each boosting step, we can directly get the weights of new features. and eta actually shrinks the feature weights to make the boosting process more conservative.	0.1
max_depth	Maximum depth of a tree, increase this value will make the model more complex/likely to be overfitting. 0 indicates no limit, the limit is required for depth-wise grow policy.	10
subsample	Subsample ratio of the training instance. Setting it to 0.5 means that XGBoost randomly collected half of the data instances to grow trees and this will prevent overfitting.	0.886
min_child_weight	Minimum sum of instance weight (hessian) needed in a child. If the tree partition step results in a leaf node with the sum of instance weight less than min_child_weight, then the building process will give up further partitioning. In a linear regression model, this simply corresponds to the minimum number of instances needed to be in each node. The larger, the more conservative the algorithm will be.	5
colsample_bytree	Subsample ratio of columns when constructing each tree.	0.886
scale_pos_weight	Control the balance of positive and negative weights, useful for unbalanced classes. A typical value to consider: sum(negative cases) / sum(positive cases)	10
alpha	L1 regularization term on weights, increase this value will make the model more conservative. Normalized to the number of training examples.	10
gamma	Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger, the more conservative the algorithm will be.	30
lambda	L2 regularization term on weights, increase this value will make the model more conservative.	50
silent	0 means printing running messages, 1 means silent mode.	1

Table 6: Model parameters for XGBoost

5.1.2 H2O

H2O is an open-source machine-learning technique for processing big data. Its faster speed and scale enable processing for real-time analytics instead of having to process the data in batches or just processing samples of the data at a time. It is one among many tool sets which enable data scientists to build models for production from the same platform used for modeling. H2O is an easy to use, web-based, interactive, open-source interface for H2O. Its interactive features enable combined code execution, text, mathematics, plots, and rich media in one package. One exceptional feature about the H2O is that one may point and click for building, validation, and testing models. The development team implemented the H2O technique to call the other models and compare their efficiency. Because H2O is easy to use and relatively fast to test and re-run models, the development team recommends it be added to the modeling toolbox at the enterprise level.

5.1.3 LightGBM

Light Gradient Boosting Machine (LightGBM) is a gradient boosting technique which uses a tree-based learning algorithm. LightGBM grows trees vertically (leaf-wise) while other algorithms grow trees horizontally (level-wise). It will choose the leaf with maximum delta loss to grow. When growing the same leaf, the leaf-wise algorithms can reduce more loss than a level-wise algorithm. The size of data is increasing day by day and is becoming difficult for traditional data science algorithms to give faster results. LightGBM is prefixed as ‘Light’ because of its high speed. LightGBM can handle the large size of data and takes less memory to run. Another reason why LightGBM is popular is that it focuses on the accuracy of the results. LightGBM also supports GPU learning processing which includes video analysis, image classification, video analytics, speech recognition and natural language processing and thus data scientists are widely using LightGBM for data science application development. The development team used this algorithm because of it ease, performance, and predictive accuracy. Figure 7 below shows the tunable parameters of LightGBM.

Parameter Name	Description	Value
objective	"reg:linear" --linear regression	"reg:linear"
max_depth	Tree depth explicitly	7
feature_fraction	For faster speed	0.7
bagging_fraction	For faster speed	0.8
min_data_in_leaf	This is a very important parameter to deal with overfitting in leaf-wise tree. Its value depends on the number of training data and num_leaves. Setting it to a large value can avoid growing too deep a tree, but may cause under-fitting. In practice, setting it to hundreds or thousands is enough for a large dataset.	30
learning_rate	For better accuracy	0.02

Table 7: Model parameters for LightGBM

5.1.4 K-Means Cluster with XGBoost

K-Means Clustering is an unsupervised learning algorithm which tries to cluster data based on their similarity. Unsupervised learning means there is no outcome to be predicted, and the algorithm just tries to find patterns in the data. In K-Means clustering, we have to specify the number of clusters we want the data to be grouped into. The algorithm randomly assigns each observation to a cluster, and then finds the centroid of the cluster.

We used the “elbow” method to find the appropriate number of clusters for this dataset, which identifies the point where adding a new cluster does not greatly increase the predictive accuracy. Figure 7 below shows a plot to identify the best number of K-means cluster.

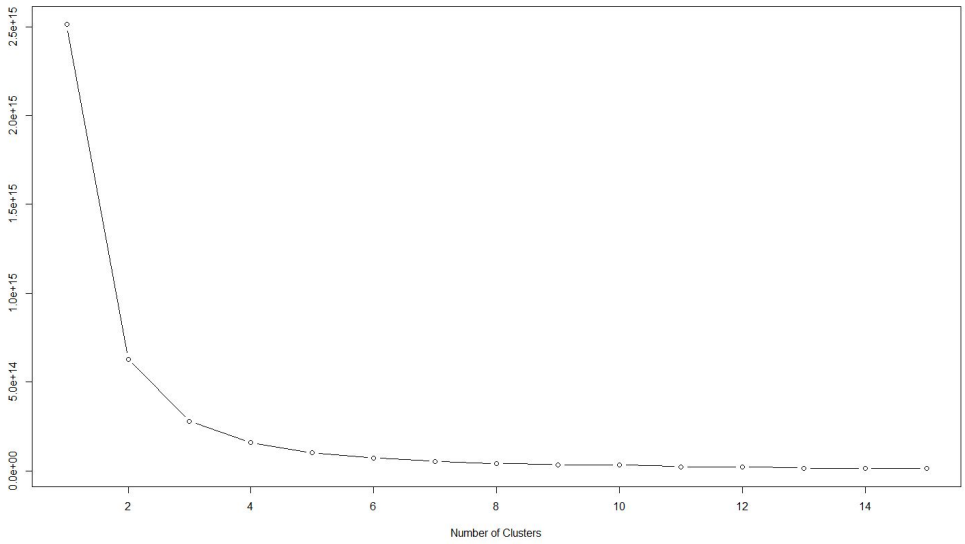


Figure 7: Scree plot to identify the best number of K-Means cluster

As mentioned above, one of the key decisions to be made when performing K-Means clustering is to decide on the number of clusters to use. In practice, there is no easy answer and it’s important to try different ways and numbers of clusters to decide which option is the most useful, applicable or interpretable solution. We think that setting the number of clusters at 6 will give an optimum predictive accuracy for this dataset.

Figure 8 below explains how the training data is first split into 6 clusters, the XGBoost algorithm generates a prediction for each cluster and then the results are merged as one linear classification.

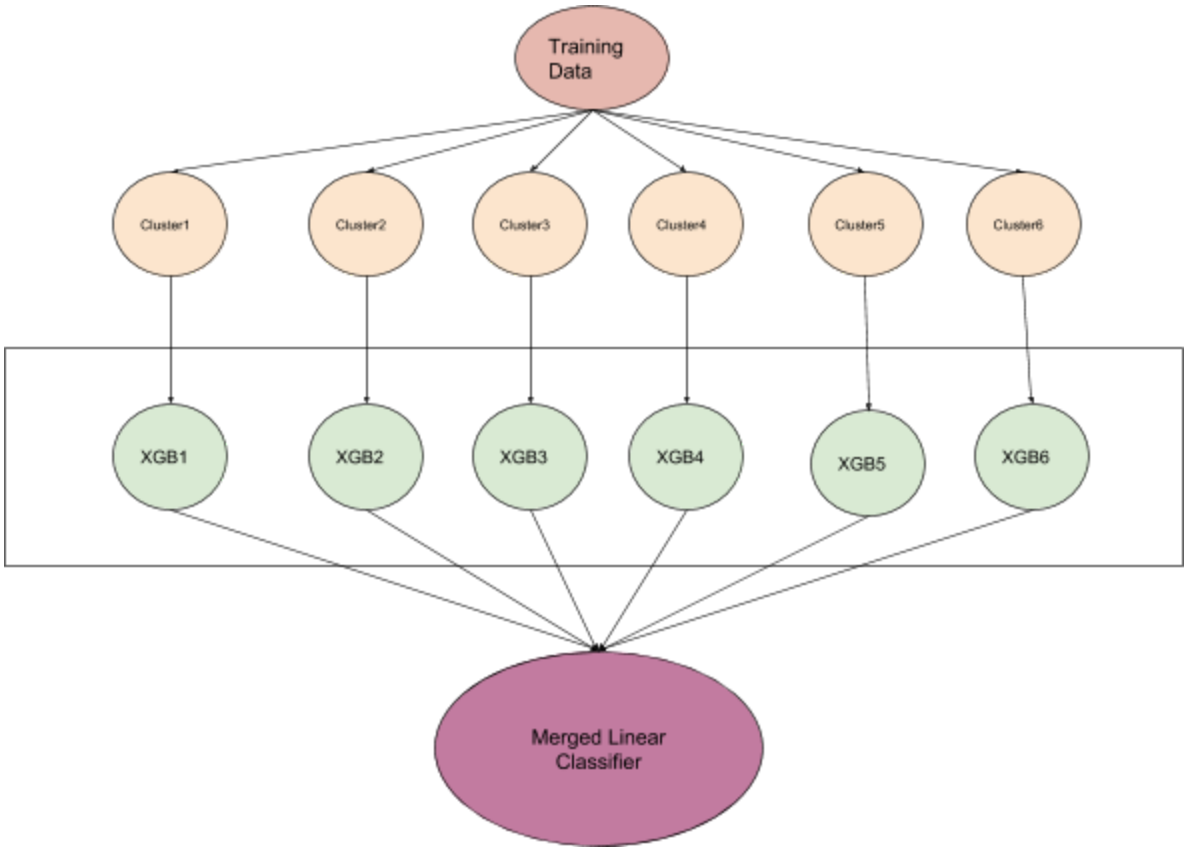


Figure 8: Diagram showing the combination of clustering and XGBoost to improve predictive accuracy

5.2 Model Validation /Testing

Our model development approach involves multiple levels of validation. We split the data into a training set (from 01-01-2016 to 03-08-2017 : 213,510 records) and a validation set (from 03-09-2017 to 04-22-2017: 44,321 records). We used the same data with different models/techniques to measure our prediction validation via RMSLE (Root Mean Squared Logarithmic Error). Table 8 below shows the RMSLE values for 5 best performing model/techniques, used to predict the total number of visitors to a restaurant from 04-23-2017 to 05-30-2017 (test set).

Model/Technique	RMSLE (Validation Set)	RMSLE (Test Set)
XGBoost	0.52	0.52
H2O	0.512	0.512
LightGBM	0.525	0.525
K-Means with XGBoost	0.47	0.47
Random Forest	0.48	0.48

Table 8: Accuracy metrics summary

Based on the model validation results, the development team has chosen K-Means with XGBoost to be the final model for this project. K-Means with XGBoost has shown the best results due to the following reasons:

1. K-Means attempts to find discrete groupings within the data, where members of a group are as similar as possible to one another and as different as possible from members of the other groups. Restaurants visitors are highly volatile based on holidays so we need to cluster data based on high volume of visitors.
2. XGBoost “continue training” feature helped to further boost an already fitted model on new data. This is an iterative process where the first set of data is analyzed and then it is merged with next set of data for further analysis.
3. We used the “elbow” method to find out the appropriate number of clusters for this dataset, which identifies the point where adding a new cluster does not greatly increase the prediction accuracy.

Please refer to section 12.3 in the appendix for model development source code and binary distribution information.

6. Dashboard Development

The intent of the Foodie Analytics dashboard is to give Recruit Holdings the power to understand what is happening within the competitive market, as well as understand future visitor trends for any particular restaurant. This is accomplished through three fully interactive pages within the dashboard, providing different levels of aggregations from a prefecture/national view down to the single restaurant level.

The first and second pages of the dashboard provide aggregate metrics meant to inform the user of general business trends. The first page (section 6.1) is the Location Dashboard, which provides data at the prefecture and genre level. The second page (section 6.2) is Location Detail Dashboard, which examines individual restaurants within a selected prefecture. By leveraging the first and second pages, the user will have guidance to which prefectures or genres hold the most promise for future development as well as which restaurants within the prefecture are the most competitive.

The third page (section 6.3) operates as a restaurant owner’s view or restaurant-detail competitive view, providing restaurant specific forecasting data. This page is ideal for either a restaurant owner to manage their business, or to get insights on how a competitor restaurant is projected to perform.

6.1 Location Dashboard

Below, figure 9 displays the Location Dashboard. This dashboard presents a visual of Japan as a whole, rolling up customer metrics to the prefecture level. In the map of Japan, each prefecture is marked by a blue circle. The size of the circle designates the number of historical visitors, such that the larger the circle is the greater the number of visitors. Intuitively, Tokyo is much larger than the surrounding areas per the below screenshot. However, the total visitors metric fails to tell the whole story. Higher visitor counts are associated with higher populations, so the metric by itself does not describe how many visitors the average restaurant might receive. Hence, the markers are also color-coded based on historical daily visitors per store. This attribute measures the average customer inflow by location, with darker colors designating higher traffic. With this

additional metric, it is clear that restaurants in other prefectures have more visitors dining with them. For example, Shizuoka is averaging closer to 17 visitors per restaurant per day, versus closer to 12 for Tokyo.

This page also provides genre level data at the prefecture aggregate. Besides measuring total historical visitor count and average daily visitor count, it also provides insights into the predicted future daily visitor count. This metric helps to understand how a particular genre is trending in the aggregate. Every genre measured is predicted to have increasing customer traffic, with the exception of Karaoke/Party.

By selecting either a prefecture marker or a particular genre, the page updates to reflect the filtered selection. Hence, by selecting a particular genre the prefecture markers update their size and color to reflect historical visitor count and daily visitor count based on that genre. Similarly, by selecting a prefecture marker the genre measurements will reflect trends for that prefecture.

By selecting a prefecture marker, there is an option to “Drill Down to Area Detail”. This selection brings you down to the next level of the dashboard, based on which prefecture was selected.

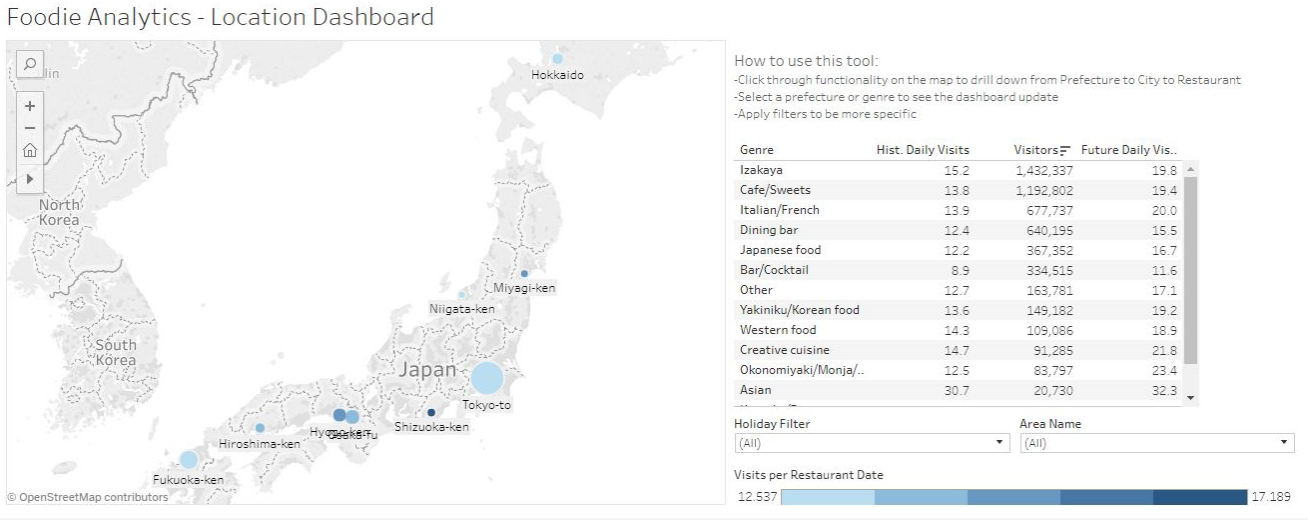


Figure 9: Prefecture view of the dashboard

6.2 Location Detail Dashboard

In the Location Detail page, below in figure 10, individual restaurants now populate the map. The map is focused on a particular prefecture, whichever was selected in the prior dashboard. Given the dataset was anonymized, restaurant locations are approximate and hence overlap with one another. Similar to the Location dashboard, the color and size convention of the restaurant markers match that of the prefecture markers. Additionally, visibility is given to the historical total and daily visitors and future predicted daily visitors at a restaurant level.

Since there can be a large number of restaurants for any given prefecture, filters are provided to drill down to a smaller subset of restaurants in order to define what is most relevant for the user. By selecting a genre, a user can understand which areas within the prefecture are over or underserved. By selecting a region within the prefecture, a user can understand who his nearby competitors are and how they are performing. Ultimately, this page examines what restaurants are present in a particular prefecture, where they are, and how they are performing.

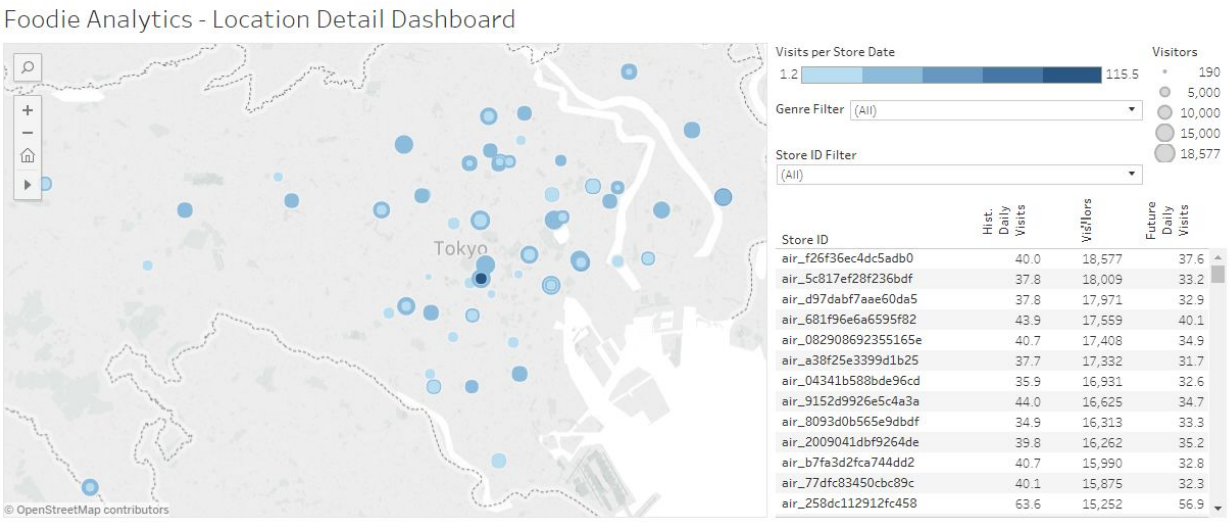


Figure 10: Location Detail view of the dashboard

6.3 Restaurant Detail Dashboard

The Restaurant Detail dashboard, seen in figure 11, focuses on a single restaurant based on the selection in the previous dashboard. The intent for the Restaurant Detail page is to provide specific historical and future insights on customer traffic flows, allowing a manager to manage labor and resource allocation based on the Foodie Analytics model or to examine traffic patterns at a competing restaurant.

First, the Restaurant Detail page provides day level predictions of customer flows for the entirety of the predicted dataset. This can be used directly for planning operations and scheduling for the near-term future. Next, there is a bar chart comparison of historical versus predicted daily customer visits by day of the week. This visualization informs how visitor trends for particular days of the week may change in the future. For example, perhaps weekends will become busier and weekdays will become less busy, which could influence how management operates the business. Finally, a line chart visualizes predicted visitors by date. On top of illustrating the day of the week volatility in customer visits, it also visualizes how the restaurant visitor rate is trending based off a simple linear regression. If the line is positive, the restaurant is predicted to have an increasing customer traffic flow. If the line is negative, the opposite is true.

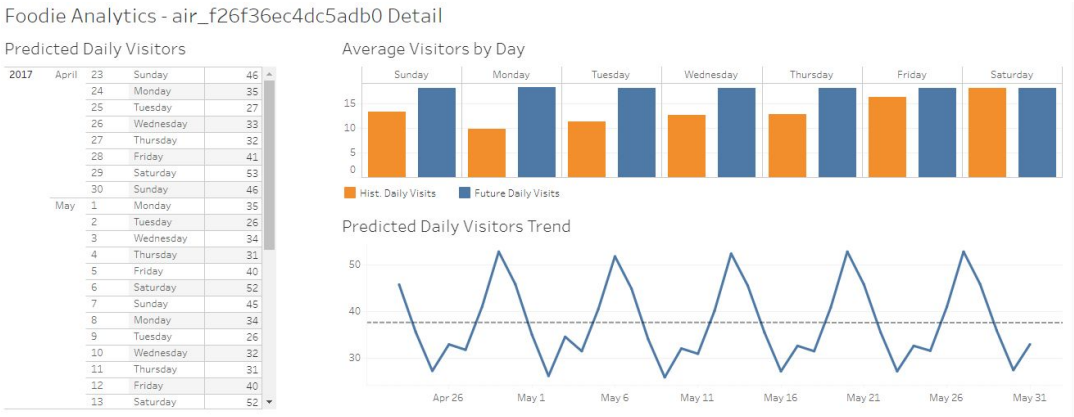


Figure 11: Restaurant Detail view of the dashboard

Please refer to section 12.1 in the appendix for details on how to access the Tableau dashboard.

7. Mobile and Web Application Development

The mobile and web applications are the delivery channels for data insights. One of our project goals is to provide a single view of forecasted data as well as an aggregate view of forecasted data on-the-go. The mobile and web applications were developed based on the idea of satisfying both single and aggregate data. The mobile and web users can select an individual restaurant (single) or area (aggregate) or genre (aggregate) to see the forecast. The start and end dates can be selected from 1 to 39 days i.e. 4/23/2017 to 5/30/2017. The final screen will be a composite view of restaurant/location/genre with start and end dates and a bar chart with the forecasted visitation.

This design helps meet the customer needs to be outlined in our project goals:

1. Customer traffic flow: How many customers are expected to visit a restaurant on any given day? What factors (such as visitation seasonality and holidays) impact the number of visitors?
2. Location: Where should the next restaurant be opened to generate the highest amount of foot traffic? Which restaurants should be scaled back given the lack of demand?
3. Competition: What's the competitive landscape for any existing or new restaurant?
4. Preference: What type of restaurants are most popular? Is there any trend indicating shifts in customer dining preference?
5. Delivery Platform: How can data insights be delivered on the go?

The development of the mobile application was performed in two phases: first, we built a native application for Android devices and second, we developed a responsive-design web application for iOS devices. A native application for iPhone users can be developed as part of future enhancement.

The basic prototypes for both the Android and the responsive web application provide the same information to an end-user.

Figure 12 below shows the main view (home page) of the mobile screen. It is split into three components. The first component has following three menus:

1. "Link to Dashboard" - A link to the Tableau dashboard
2. "Forecast" - Provides user a choice to check the visitation forecast in 3 different ways:
    a. Forecast by Restaurant
    b. Forecast by Area
    c. Forecast by Genre
3. "About Us" - Provides information about Foodie Analytics

The second component is for users to select values for restaurant/area/genre, the start date, and the end date. Currently, for this application, a user can select only up to 39 days starting from 4/23/2017. All remaining dates are disabled for selection because we only had predictor variable information available until 5/30/2017.

The third component will display a bar chart based on the user's inputs, visualizing projected visitor numbers per day.

Figure 12: Mockup of the mobile application

7.1 Mobile Application

7.1.1 Android Mobile Application Technical Details

The following table describes the specification and Android Software Development Kit versions. We used Nexus mobile phone simulator for testing this application.

compileSdkVersion	26
applicationId	com.foodieanalytics
minsdkversion	19
targetsdkversion	26
testedadbname	Nexus 5X

Table 9: Mobile Application Details

7.1.2 Android Mobile Application Screenshots

The screenshots are taken from Nexus Android device 5X screen (5.2 inch display). The look and feel will be the same for all Android device but some larger phones will give extra container space for graphs and input elements. The screenshots start with the home page, user input screen, and final result screen. We recommend to use landscape mode for tableau dashboard but the screenshot taken in portrait mode for consistency purpose. We are utilizing an open source MPAndroidChart package for bar chart generation.

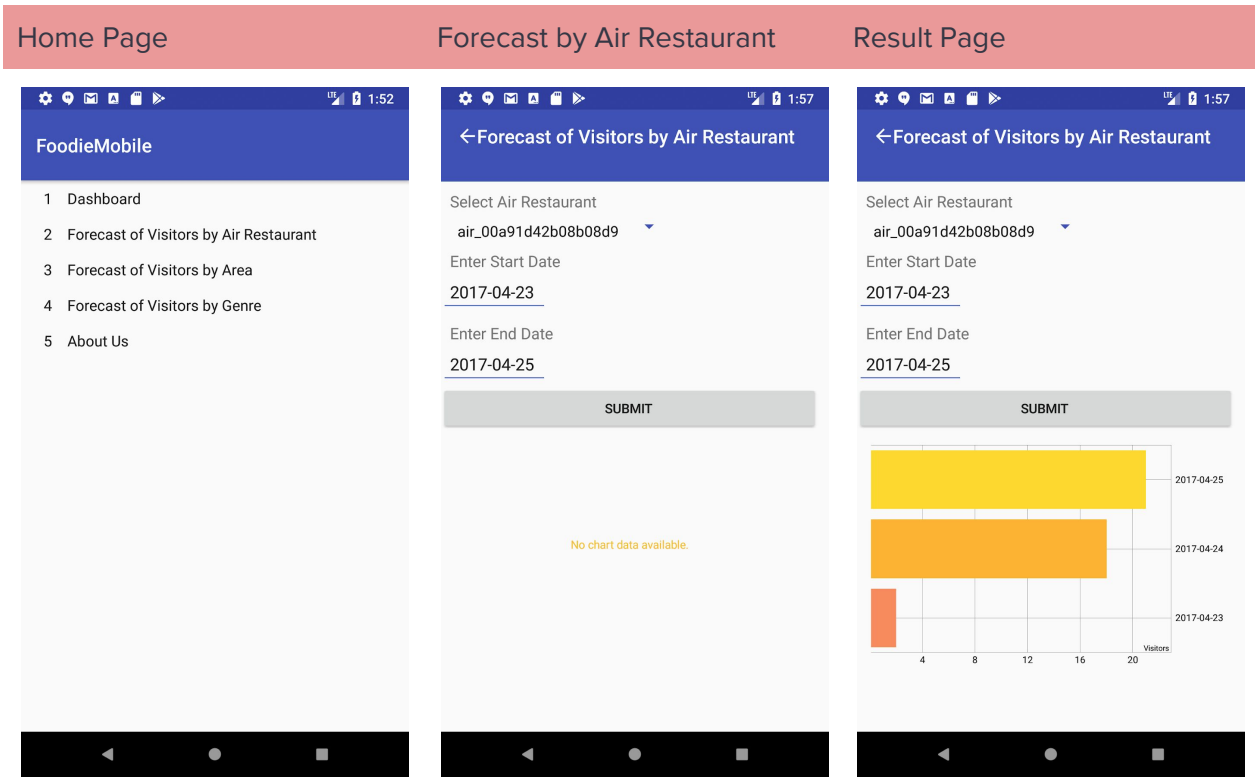


Figure 13: “Forecast by Restaurant” example

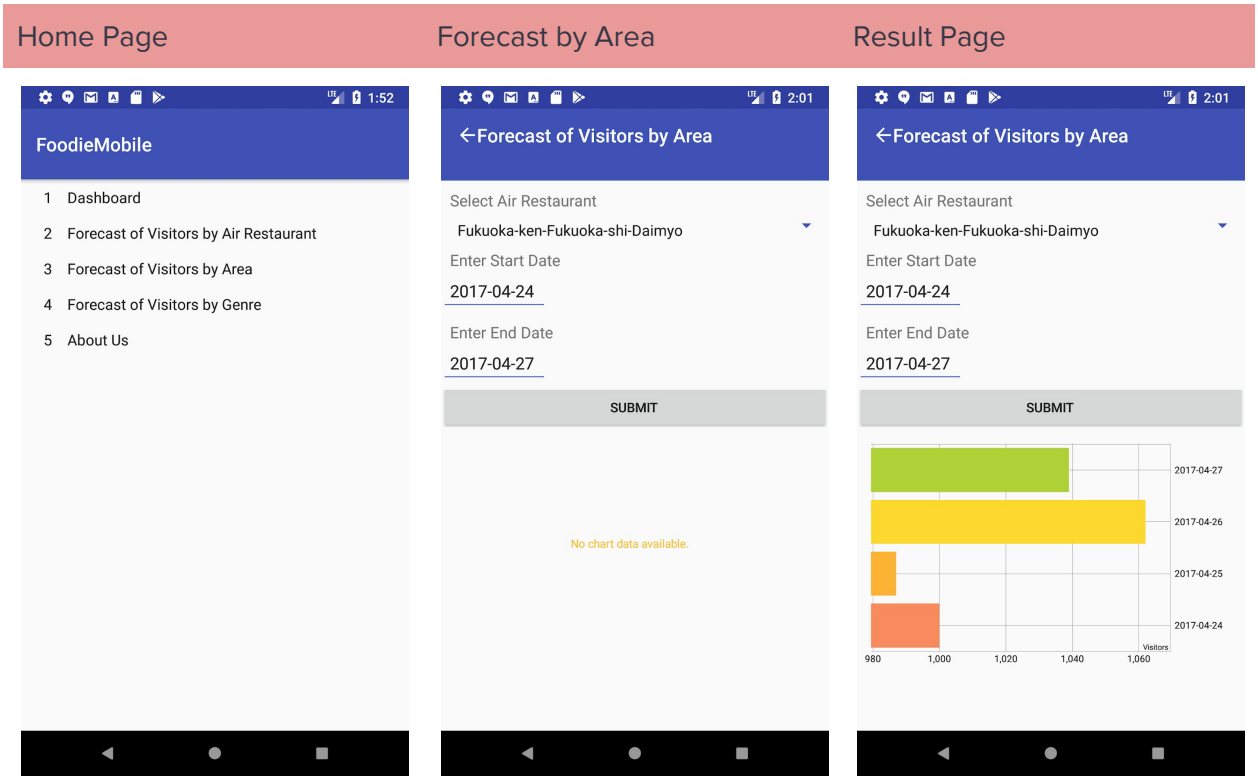


Figure 14: “Forecast by Area” example

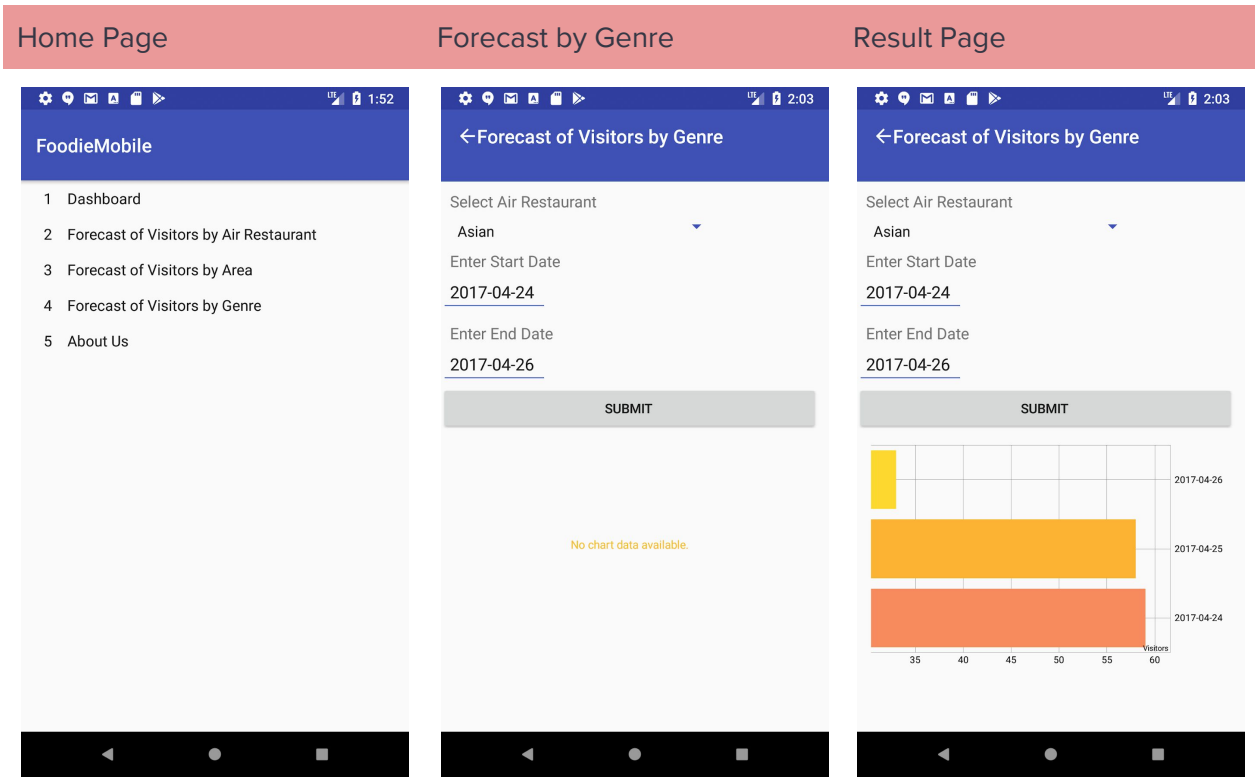


Figure 15: “Forecast by Genre” example

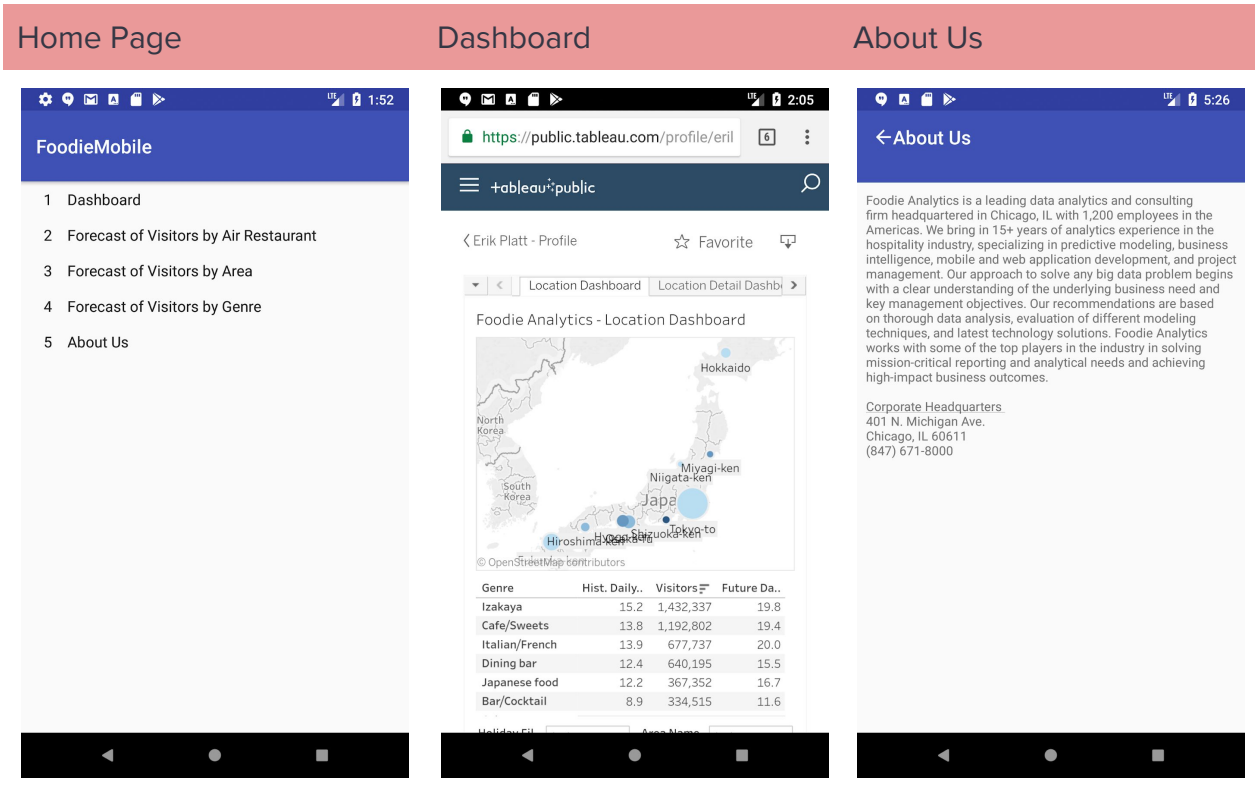


Figure 16: “Dashboard” example from mobile application

Please refer to section 12.3 in the appendix for details on how to download the android application and view the for source code.

7.2 Web Application

7.2.1 Responsive Application Details

iOS utilizes a responsive design to render a web page on any size of mobile screen. The idea behind a responsive web site is it has to seamlessly load across all platforms, devices and desktops. A good responsive design will consider screens ranging from desktop monitors to new mobile phones on the market. The trick is to utilize technologies like CSS (Cascade Style Sheets), CSS3 Introduced Media Queries, and more. For example, the media queries in CSS3 will look for the capability of the device instead of looking at the type of the device. Media queries can be used to check many things, such as width and height of the viewport. Please refer figure 17 below for CSS rendering mechanism.

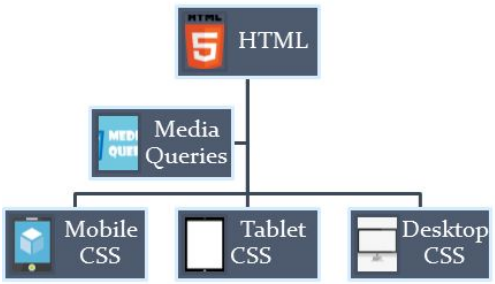


Figure 17: Responsive design development methodology

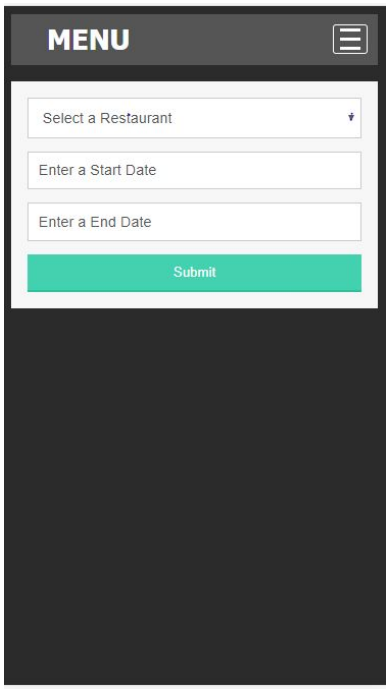
The responsive design is achieved with paramount advantages. The HTML page will load CSS values based on media queries' viewport configuration. The viewport is then determined based on the size of screen.

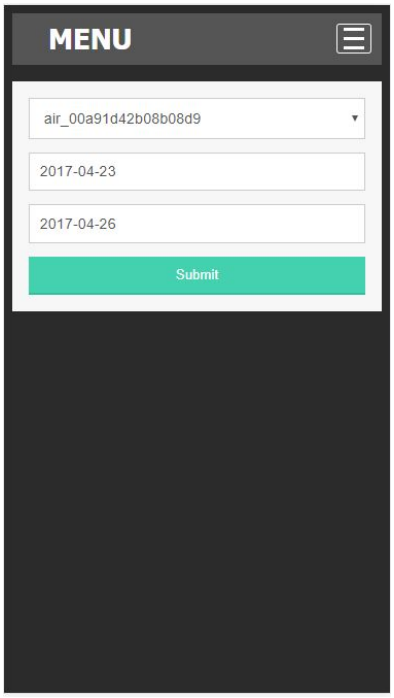
The web application is hosted on Amazon tomcat environment with Java 1.7. Please see appendix 12.2 for URL to access the application through iOS devices. We are recommending it for an iOS device but any web browsers and Android devices as well as can access this web application.

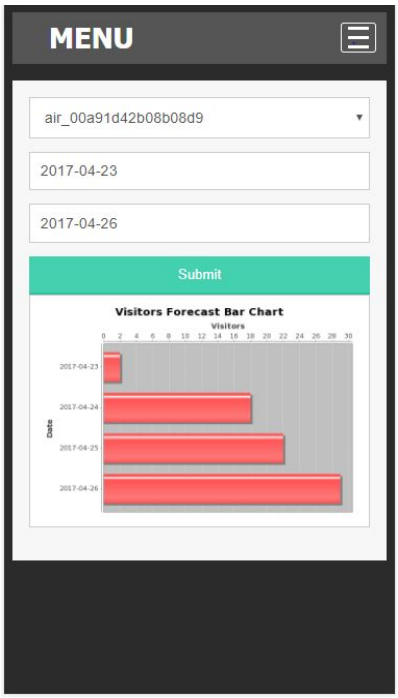
7.2.2 Responsive Application Screenshots

The screenshots are taken from iPhone X (5.8 inches). The look and feel will be same for all iOS device but the Menu view will change based on mobile device or browser screen size due to responsive design. The screenshots start with the home page and user input screen and final result screen. Please refer figure 18-21 for detailed screenshots.

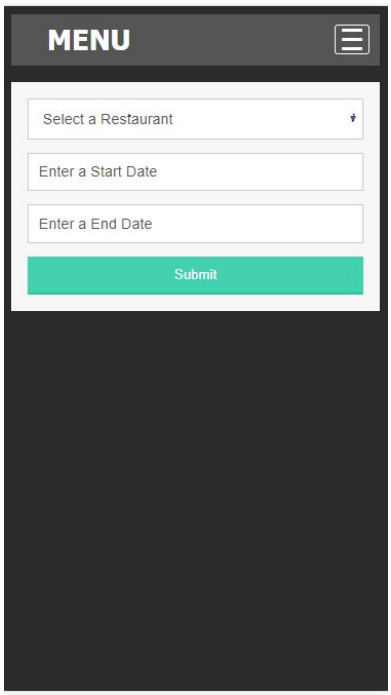
Home Page	Forecast by Air Restaurant	Result Page

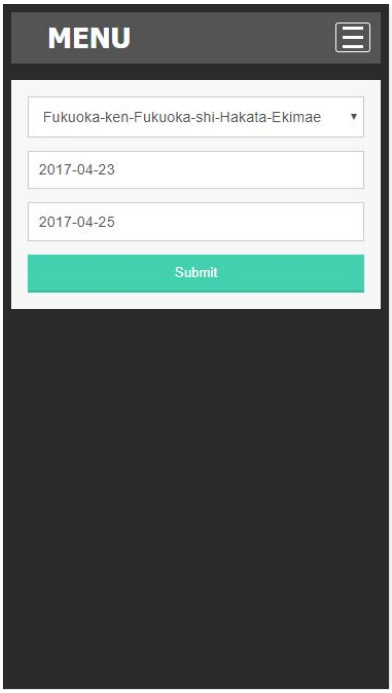


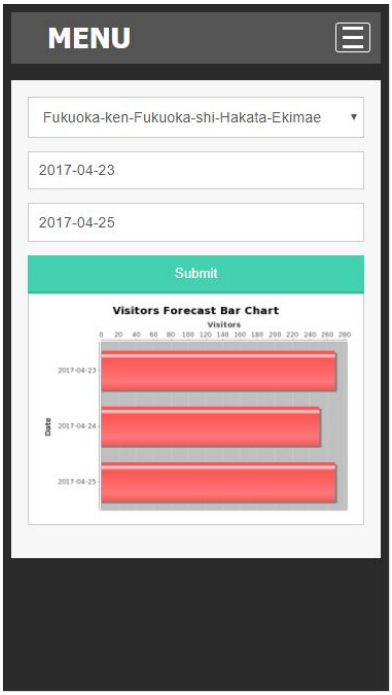




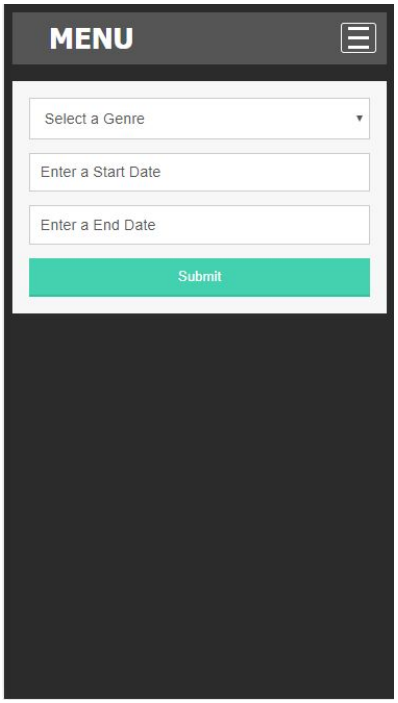
Home Page	Forecast by Area	Result Page



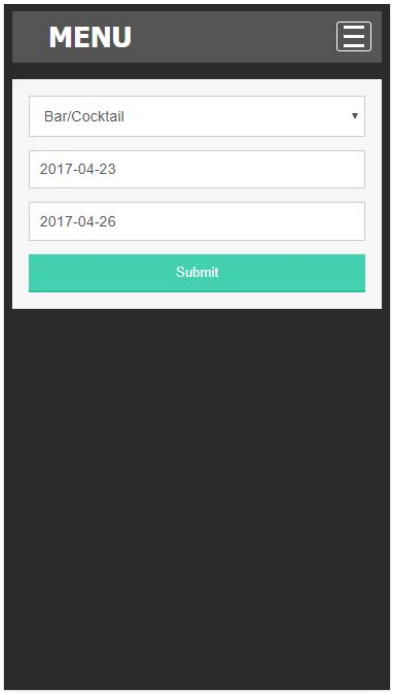




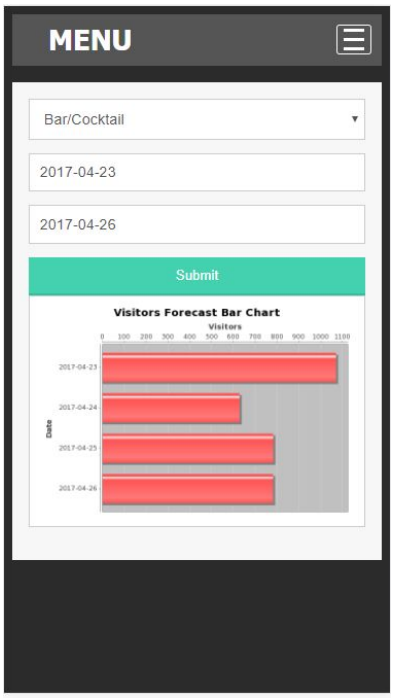
Home Page



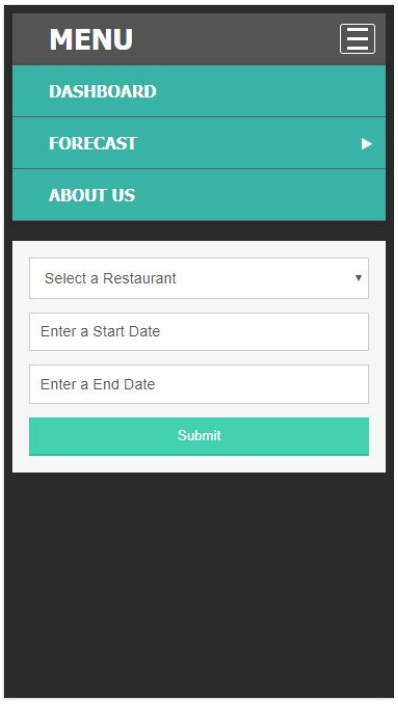
Forecast by Genre



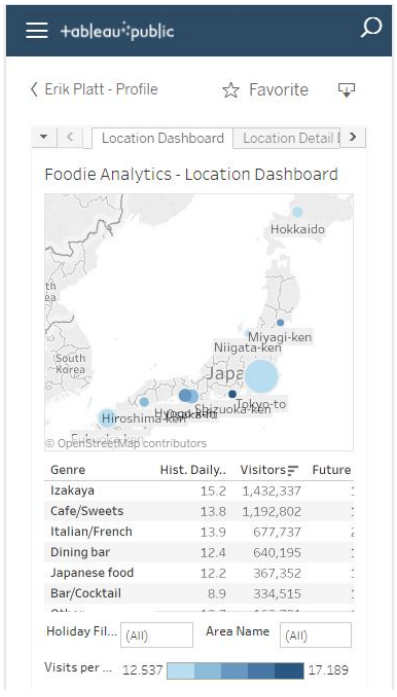
Result Page



Home Page



Dashboard



About Us

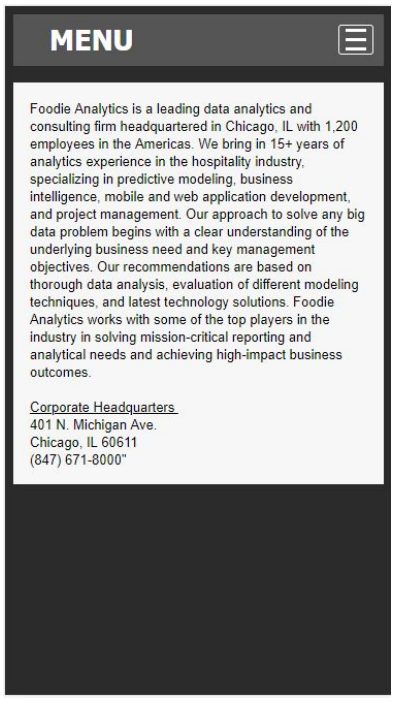


Figure 18-21: Example screenshots of the responsive design application

Please refer to section 12.2 in the appendix for details on how to access the web application from a mobile device.

8. Assumptions and Limitations

8.1 Assumptions

- iOS users will leverage the web application until an iOS native application is developed (see Future Enhancements)
- Tableau will be accessed from the web application as well as the mobile application using the same URLs
- The look and feel will not be the same for the web application as well as the mobile application through a responsive design. The menu will change based on screen size.
- There will not be any authorization or authentication for the web application or the mobile application and web application at this time

8.2 Limitations

- Since we don't have restaurant names, restaurant IDs will be supplied instead
- Predictions can be made for up to 39 days following the last day of the dataset, 04-23-2017 since the predicted data is only available until 05-30-2017.
- Given the 10 week engagement, we will only be developing the mobile application for Android-based mobile phones at this time
- The web application and mobile application did not go through negative testing on test cases.
- Even though the web application can be accessed through web browsers and Android devices we did not test it on Android device and Internet Explorer / Firefox browsers. However, it was tested on Chrome browser.

9. Conclusions

Given the highly competitive nature of the restaurant industry, there are many challenges in maintaining a successful enterprise. Foodie Analytics hopes to resolve some of these challenges through leveraging the vast and detailed data collected by Recruit Holdings via Hot Pepper Gourmet and AirREGI platforms.

Foodie Analytics completed an in-depth exploratory data analysis to ensure all the intricacies and nuances of the data have been captured and understood. We identified and analyzed the response variable and each predictor variable used for data modeling. We identified commonalities within the datasets, merged them into training and testing datasets, and iterated through several rounds of modeling. After identifying 5 promising models/techniques, we chose to proceed with a K-Means with XGBoost model given its high accuracy of the Root Mean Squared Logarithmic Error. Finally, Foodie Analytics developed a dashboard and mobile application to help Recruit Holdings derive immediate value from our efforts.

Furthermore, identifying Japanese holidays provided key insights into visitor predictions. In addition, derivations of actual visitors (such as mean and median visitors), and bringing in external

data on temperature and precipitation further improved our ability to predict. With a complex time-series problem such as this, we found the inputs needed for accurate predictions were relatively simple. These findings will likely help Foodie Analytics in producing actionable results on related future projects.

10. Recommendations

Our Tableau dashboard and mobile/web application will provide direct insight into current and potential restaurant owners. Although the final decision for starting/optimizing a restaurant should take into account the professional expertise of the restaurant owner, we suggest using the tool as follows:

For new restaurant owners:

- Identify locations where your particular genre would fit in best. Consider whether a restaurant should be opened in an area where the potential owner's preferred genre is already popular because there is an existing demand for it or whether the restaurant should be started in an area with low density because it is missing in the local market.
- Specify the genre in the dashboard or mobile/web application and look at predicted visitors across competitors. Are there one or two restaurants monopolizing the market in the genre? If so, it may be more advantageous to avoid this area.
- Take into account overall visitor predictions over time to ensure consistent visitation and analyze the impact of visitation on holidays.

For current restaurant owners:

- Analyze average visitation rates vs predictions for the owner's current restaurant. Is it competitive with the average restaurant? If not, why?
- Use the predicted visitor rates over time to determine on which days competitors are receiving the most visitors.
- Compare predicted visitors for the restaurant owner's preferred genre vs predicted visitors for other genres to see if the restaurant would be more competitive by adding some more trendy options to the menu.

Combining these recommendations with professional knowledge of the restaurant business will help restaurant owners make more money, be more responsive to changes in restaurant trends, and better analyze the competition.

11. Future Enhancements

If given an opportunity, Foodie Analytics believes that the following enhancements could be made to our final deliverable as part of our next project with Recruit Holdings.

- Build a native iPhone application
- Provide training to Recruit Holdings' development team for ongoing maintenance of project deliverables and predictive models
- Add the H2O technique to the modeling toolbox at the enterprise level
- Enhance the current model to allow for predictions further into the future, as more data becomes available
- Perform extensive testing of mobile and web interfaces to execute negative test cases and display error messages where needed

- Add authorization or authentication for the web application and mobile application

12. Appendix

12.1 Dashboard URL

To access the Foodie Analytics dashboard, click the link below:
https://public.tableau.com/profile/erik.platt#!/vizhome/FoodieAnalyticsDashboardv2/LocationDashboard

12.2 Web Application URL

To access the web application, click the link below:
http://foodieapp-env.us-east-2.elasticbeanstalk.com/

12.3 Final Deliverables and Source Code

The development code used for this engagement has been provided in a zip file called "Final_Submission.zip". The code has been separated into multiple files corresponding to different parts of the project. Please refer to the "readme.txt" file for detailed information. The zip file has 3 sub folders:

1. Mobile Application Download (for Android Users) - APK binary distribution
2. Web Application Deployment File - Web binary distribution
3. Dashboard - Tableau File (.twbx)
4. R source code for model development
5. Readme File

Web Application Source Code:
https://github.com/nw498/finalproject498/blob/master/FoodieWebApp.zip

Mobile Application Source Code:
https://github.com/nw498/finalproject498/blob/master/FoodieAnalytics.zip

12.4 Data Files

Air Reserve Details: https://github.com/nw498/finalproject498/blob/master/air_reserve.csv.zip

Air Store Info: https://github.com/nw498/finalproject498/blob/master/air_store_info.csv.zip

Air Visit Details: https://github.com/nw498/finalproject498/blob/master/air_visit_data.csv.zip

HPG Reserve Details: https://github.com/nw498/finalproject498/blob/master/hpg_reserve.csv.zip

HPG Store Info: https://github.com/nw498/finalproject498/blob/master/hpg_store_info.csv.zip

Air-HPG Store Mapping:
https://github.com/nw498/finalproject498/blob/master/store_id_relation.csv.zip

Weather Details:
https://github.com/nw498/finalproject498/blob/master/1-1-16_5-31-17_Weather.zip

12.5 References

Walia, A. S. (2017, July 24). Random Forests in R. Retrieved March 04, 2018, from

https://datascienceplus.com/random-forests-in-r/

Annual events. (n.d.). Retrieved March 04, 2018, from

https://www.japan-guide.com/e/e2062.html

Golden week. (n.d.). Retrieved March 04, 2018, from https://www.japan-guide.com/e/e2282.html

Parameters Tuning. (n.d.). Retrieved March 04, 2018, from

http://lightgbm.readthedocs.io/en/latest/Parameters-Tuning.html

Development Guide¶. (n.d.). Retrieved March 04, 2018, from

http://lightgbm.readthedocs.io/en/latest/Development-Guide.html

XGBoost Hyperparameters. (n.d.). Retrieved March 04, 2018, from

https://docs.aws.amazon.com/sagemaker/latest/dg/xgboost_hyperparameters.html

Mandot, P. (2017, August 17). What is LightGBM, How to implement it? How to fine tune the

parameters? Retrieved March 04, 2018, from
https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

Blackwell, P. A. (2017, February 13). CitizenNet Blog: A Gentle Introduction to Random

Forests, Ensembles, and Performance Metrics in a... Retrieved March 04, 2018, from
https://medium.com/@arshavir.blackwell/citizennet-blog-a-gentle-introduction-to-random-forests-ensembles-and-performance-metrics-in-a-17b6e196b477

Kodali, T. (2015, December 28). K Means Clustering in R. Retrieved March 04, 2018, from

https://datascienceplus.com/k-means-clustering-in-r/

Rego, F. (n.d.). Retrieved March 04, 2018, from

https://rstudio-pubs-static.s3.amazonaws.com/92318_20357e6dd99742eb90232c60c626fa90.html

D. (n.d.). Dmlc/xgboost. Retrieved March 04, 2018, from

https://github.com/dmlc/xgboost/blob/master/doc/parameter.md

M. (n.d.). Microsoft/LightGBM. Retrieved March 04, 2018, from

https://github.com/Microsoft/LightGBM/blob/master/docs/Parameters-Tuning.rst