Capstone Project 1: Milestone Report

Problem Statement

Running a thriving business can be more challenging than it seems. For many businesses, accurately forecasting the number of customers on a given day may be the difference between turning a profit and losing money. This is especially true of restaurants where inventory has a short shelf life and over- or under-staffing can be costly.

Predicting customer turnout at restaurants can be challenging due to unforeseen circumstances like weather and changing local competition. This is even harder for newer restaurants with little historical data.

Recruit Holdings has access to unique datasets that may make automated future customer prediction possible. Specifically, Recruit Holdings owns AirREGI (a restaurant point of sales service), and Restaurant Board (reservation log management software).

In this project, I draw from these and other data sources to predict the total number of visitors to a restaurant on a given date. This information will help restaurants be more efficient with regard to both purchasing and staffing. It will also free up resources so that restaurants can focus on creating an enjoyable dining experience for their customers.

Data

Data for this project came from AirREGI/Restaurant Board (AIR) system. AIR is a reservation and cash register system that provides users increased convenience and flexibility as they dine out. Files from the AIR system include reservations, visits, and other information from restaurants across Japan. These time-series data span from January, 2016 through April, 2017.

In these data, the following is provided for each restaurant:

Air_store_id - the restaurant's id in the air system
Visit_datetime - the time of the reservation
Reserve_datetime - the time the reservation was made
Reserve_visitors - the number of visitors for that reservation
Air_genre_name - type of restaurant
Air_area_name - region where the store is located
Latitude - approximate latitude of the restaurant
Longitude - approximate longitude of the restaurant
Visitors - the number of visitors to the restaurant on the date

In addition to the restaurant information described above, this project uses date-specific information (i.e. day of the week, holiday, etc.) as well as weather information linked to each restaurant. These include:

Calendar_date - date of each reservation
Day_of_week - day of the week for each calendar date
Holiday_flg - indicates that the day is a holiday in Japan
Avg_temperature - average daily temperature at the weather station closest to each restaurant
High_temperature - daily high temperature at the weather station closest to each restaurant
Low_temperature - daily low temperature at the weather station closest to each restaurant
Precipitation - daily precipitation total at the weather station closest to each restaurant
Hours_sunlight - daily hours of sunlight at the weather station closest to each restaurant

Data Wrangling

I began by downloading the relevant AIR files from the Kaggle API and reading them into my local Python environment. These included reservation data, visitor data, restaurant information, calendar information, and weather data. The visitor data contained the variable of interest: number of visitors for each transaction. These data were reported at the transaction-level and, therefore, contained multiple cases per restaurant, per day. Because the outcome of interest is the number of visitors per day, I chose to group the data by restaurant, resampled the data by day, and summed the number of visitors at each restaurant on each day. I repeated this process with the reservation data, but chose to count the number of reservations made for each day.

Because the visitor data contained the outcome of interest, I used a left join on the visitor data to bring in the reservation data. This ensured that all of the relevant visitor data was retained and irrelevant reservation data was excluded. I then merged the restaurant information and calendar information files using the same method.

Wrangling the weather data was more complicated because these data were stored in separate files for each weather station. Additionally, weather station ids were provided in the file names, but were not contained as data in the files. To address this, I wrote a function that read each file into an empty list while simultaneously creating an id variable generated from the id in the filename. I then concatenated these data frames to create a single weather data frame where each observation contained unique weather information for each day and for each weather station. Using a crosswalk provided by Kaggle, I then linked the restaurant data to the weather data from the nearest weather station for each day. This final data frame contained 296,279 unique observations.

Once the final data frame was created, I examined the percentage of values that were missing for each variable. These are presented in the following table:

Variable	Percent Missing
air_store_id	0%
visit_date	0%
visitors	0%
number_of_reservations	76.2%
air_genre_name	0%
air_area_name	0%
latitude	0%
longitude	0%
day_of_week	0%
holiday_flg	0%
station_id	0%
avg_temperature	9.9%
high_temperature	9.9%
low_temperature	9.9%
precipitation	30.4%
hours_sunlight	14.1%

Because the vast majority of data was missing from number_of_reservations, I chose to exclude it from the analysis. The only other variables with missing values were from the weather files. With each of these weather variables, I began by creating a missing flag to keep track of the observations that would contain imputed values. I then calculated the global daily median value for each variable and replaced missing values with the median value for that day. I chose the median because it will be robust to non-normal distributions.

As a final step, I examined outliers. Using descriptive statistics (i.e. min, max, mean, median, std, etc.), I determined that there were no extreme outliers in any of the predictors. However, the outcome (i.e. number of daily visitors) contained several extreme outliers. I dealt with these

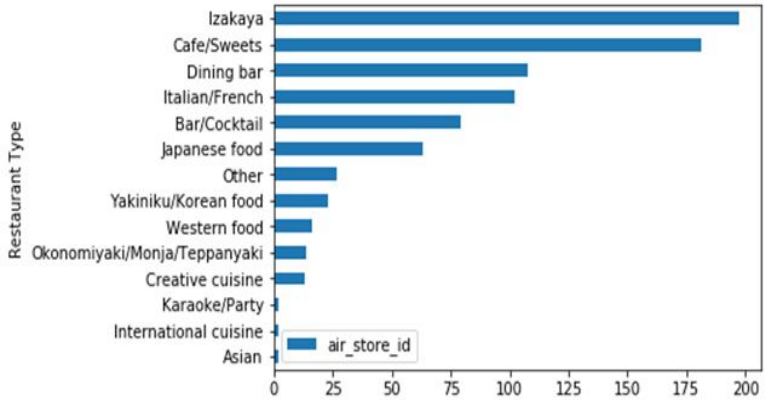
by first identifying observations with values greater than 3.5 standard deviations above the restaurant-specific mean. I then replaced these values with 3.5 standard deviations above the restaurant-specific mean to create a maximum value 'cap' for each restaurant. This eliminated the large residual values that would be created by these observations while still capturing the fact that their values were in the extremes of the distribution.

Exploratory Data Analysis: Descriptive Statistics and Graphical Analysis

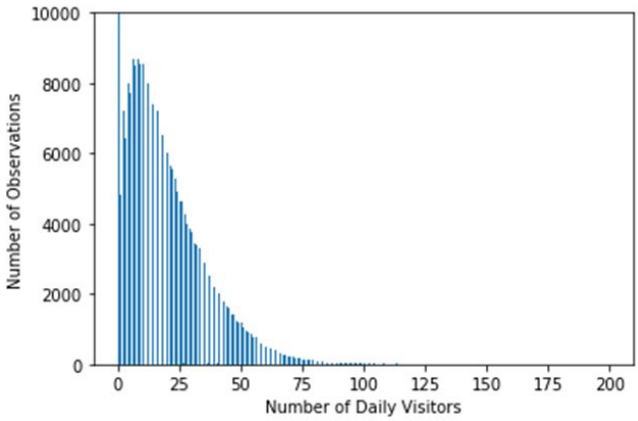
As an initial step, I conducted exploratory data analysis (EDA) to gain a better understanding of the data. In total, the data contained 296,279 unique day-restaurant observations that spanned from Friday, January 1st, 2016 through Saturday, April 22nd, 2017. These included information for 829 restaurants in 103 neighborhoods in Japan. The number of restaurants were not equally distributed across neighborhoods. The neighborhoods with the largest number of restaurants are included the following table:

Neighborhood	**Number of Restaurants**
Fukuoka-ken Fukuoka-shi Daimyō	64
Tōkyō-to Shibuya-ku Shibuya	58
Tōkyō-to Minato-ku Shibakōen	51
Tōkyō-to Shinjuku-ku Kabukichō	39
Tōkyō-to Setagaya-ku Setagaya	30
Tōkyō-to Chūō-ku Tsukiji	29
Ōsaka-fu Ōsaka-shi Ōgimachi	25
Hiroshima-ken Hiroshima-shi Kokutaijimachi	23
Tōkyō-to Meguro-ku Kamimeguro	22
Hokkaidō Sapporo-shi Minami 3 Jōnishi	21

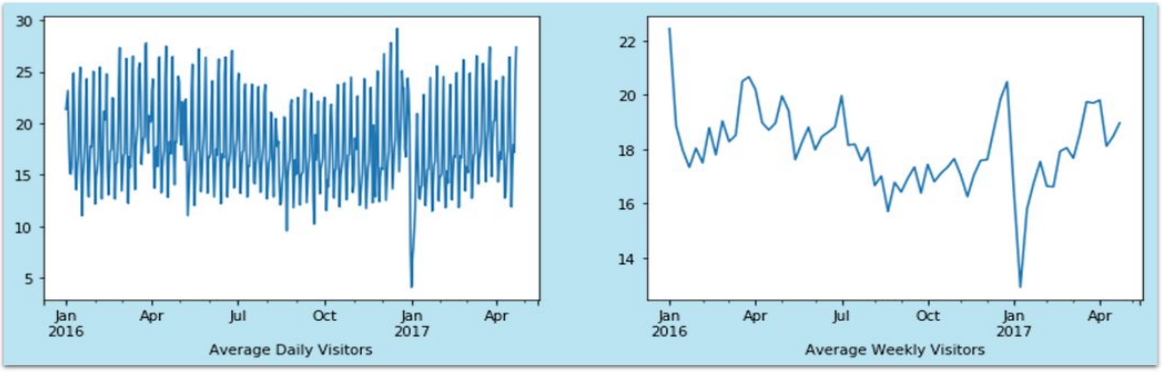
The restaurants in these data represented fourteen ‘genres’ the most frequent of which were ‘Izakaya’ and ‘Cafe/Sweets.’



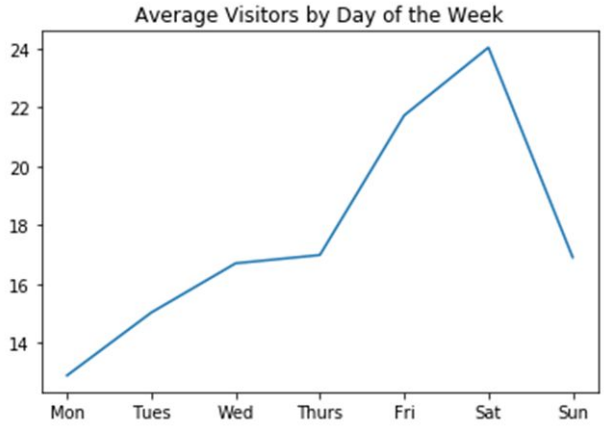
The average number of visitors across all days and all restaurants was 17.8 with a standard deviation of 16.6. The minimum value for the number of visitors was 0, the median value was 14, and the maximum value was 199.9. In this dataset, there were many observations with zero visitors (44,171).



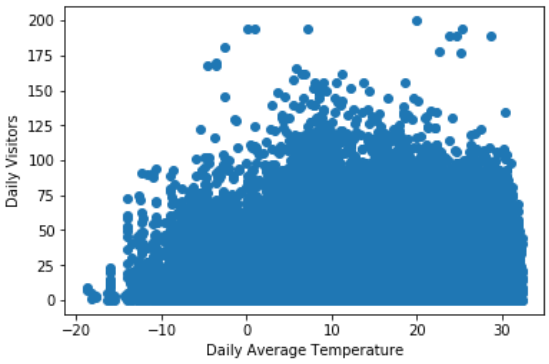
Looking at trends in the data, there are seasonal trends both within and across months. The number of visitors is highest in April through July, December, and January. The months between August and November seem to be slowest for these restaurants.

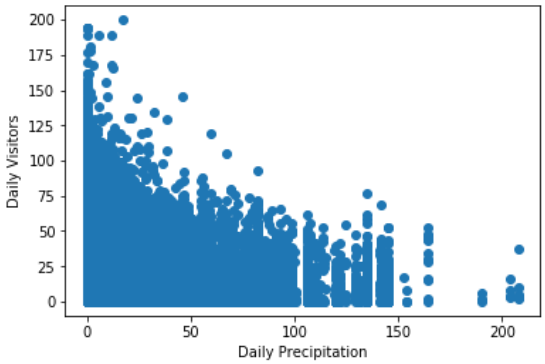


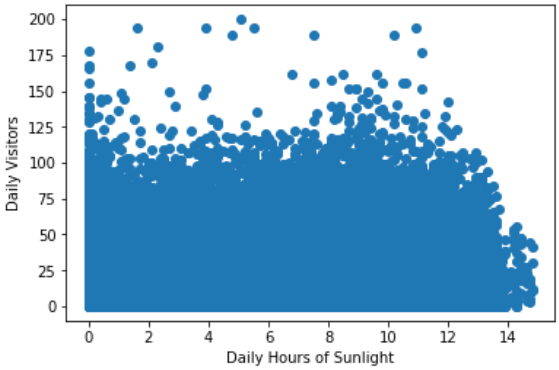
Looking at differences in average daily visitors by day of the week, Monday is the slowest day with the average number of visitors increasing steadily throughout the week and dropping sharply on Sunday.



There appears to be a slight positive relationship between average daily visitors and average daily temperature and a negative relationship between average daily visitors and daily precipitation. There does not appear to be a relationship between average daily visitors and hours of sunlight. Despite these graphical insights, formal statistical tests are needed to confirm these results







Exploratory Data Analysis: Statistical Analyses

I completed further EDA by conducting formal statistical tests. In all of these analyses, I examined relationships with the outcome of interest: the number of daily visitors at each restaurant. To do this, several preparatory steps were necessary. Specifically, the time-series nature of the data and the non-normal distribution of the outcome (and resulting non-normal residual distribution) needed to be addressed. Because the data consisted of multiple observations from the same restaurants over time, the assumption of independent observations was violated. To address this, I used Scipy's implementation of autocorrelation-robust standard errors. Additionally, the fact that the data consisted of counts of visitors meant that that the assumption of normally-distributed residuals was also violated. While a generalized linear model (GLM) with a negative binomial link function would have been most appropriate for these data, Scipy does not yet combine GLM and autocorrelation robust standard errors. Therefore, I chose to transform the outcome by taking the natural logarithm of the number of visitors and use OLS regression. To avoid infinite values, I added 1 to each value prior to this transformation.

The first question I sought to answer was whether the observed differences in average visitors by day of the week were significantly different from one another. To do this, I estimated a linear regression model with dummy variables for each day of the week as predictors and average number of visitors as the outcome. In this model, I used Monday as the reference category since the graphical analysis showed that Mondays have the fewest visitors, on average. As the table below shows, each day of the week had significantly more visitors, on average, compared to Mondays. The pattern of the coefficients reflects the pattern shown in the graphical analysis.

Day of Week	Coefficient/Std. Err
Tuesday	0.30*** (0.01)
Wednesday	0.42*** (0.01)
Thursday	0.49*** (0.01)
Friday	0.80*** (0.01)
Saturday	0.83*** (0.01)
Sunday	0.10*** (0.01)
R-squared = .05***	
* = $p < .05$, ** = $p < .01$, *** = $p < .001$	

I also conducted three additional analyses to answer the following questions:

1. Are the number of daily visitors at each restaurant related to the daily average temperature?
2. Are the number of daily visitors at each restaurant related to the daily precipitation?
3. Are the number of daily visitors at each restaurant related to the daily hours of sunlight?

To do this, I estimated three separate regression models with average daily temperature, precipitation, and hours of sunlight as predictors and daily visitors was the outcome. As the table below shows, only daily precipitation and daily hours of sunlight were statistically significant. Specifically, as daily precipitation increased, the number of visitors decreased with the opposite being true for daily hours of sunlight. Despite these statistically significant results, the extremely small coefficient estimates and R-squared values suggest that these relationships are not practically significant.

Variable	**Model 1**	**Model 2**	**Model 3**
Daily Average Temperature	0.00 (0.00)		
Daily Precipitation		-0.003*** (0.00)	
Daily Hours of Sunlight			0.007*** (0.00)
R-squared	0.00	0.001***	0.001***
* = $p < .05$, ** = $p < .01$, *** = $p < .001$			