Dexter Nguyen

Forecasting Store Traffic Using Feature Selection with sklearn and Pandas in Python

April 1, 2021

Contents

Business Understanding

My findings will encompass what may be contributing to the fluctuation of the foot traffic rates of restaurants/stores across Japan. This information will benefit corporate management levels within the F&B industry in optimizing category management, store selection, and associated resource allocation, leading to higher store visits, higher revenue, and a potential better reputation resulting from better customer service. The factors explored will also show a better picture of the Japanese market in terms of customer's tastes in consideration of other local factors like holidays, weather, and seasonality.

Data Understanding

We have three main data sets as below. As the raw data was very heavy and complicated, these three datasets were already the result of a pre-processing process.

- **Store visits data** of 2016 (train_restaurant_visitors.csv): provides such data as date of visits, type of restaurant/store, latitude and longitude of the store, and the number of visitors to each store by date.
- **Holiday information** in Japan (holidays_data.csv): provides different holiday names in Japan by date.
- **Weather data** in 2016 (new_weather_data2.csv): provides a wide variety of weather attributes in each day, in terms of key categories: temperature, humidity, rain, sunshine, snow, and wind.

Data Preparation & Exploration

Data Cleaning

After going through three data sets, I acknowledged that there were many missing values. I classified the levels of missing values in three: > 60% missing, < 20% missing, 100% missing before offering solutions for each category. For the first solution, I decided to remove all the variables with more than 60% missing values. To deal with features below 20% of missing values, I used the median approach for numerical weather attribute values: Temp, Sun, Rain_avg_year, Sun_avg_year Total_precip. In terms of the Holiday information, I replaced missing values with Normal_day since the data with holiday information accounts for a small percentage during the year.

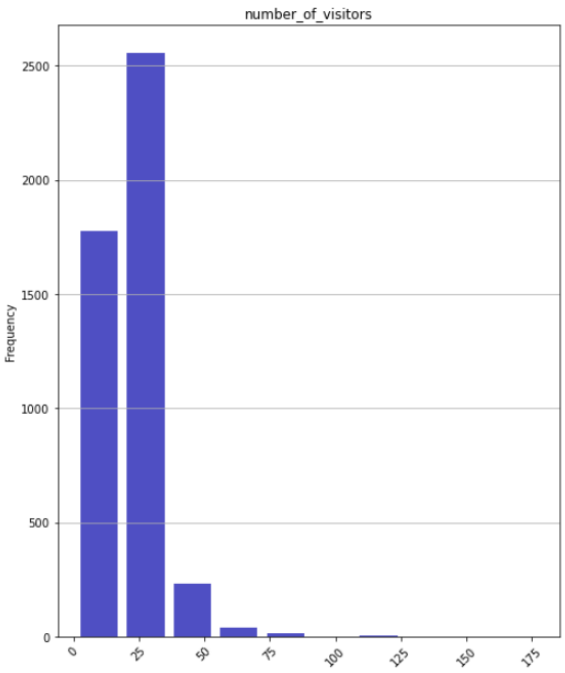
As a result of three cleaned data sets without any missing values, the next step should be data merging. Using the same Date column (Primary Key) from each data set for joining, I ended up with a master data set. This data set includes information about store visits to each store by date, the holiday information (if yes) by date, and the weather information.

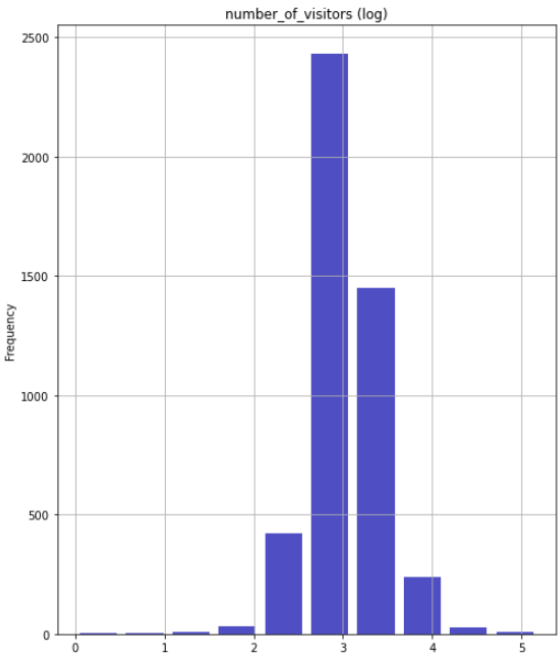
Last, I removed other irrelevant data features from our master data set during the cleaning and merging process, resulting in a final data set that is to be used for Transformation, EDA, and Modeling parts. The final data set has the below variables:

1. Date
2. Month
3. Number_of_visitors
4. Type
5. Holiday_Name
6. Temp
7. Total_Precip
8. Rain_avg_year
9. Sun
10. Sun_avg_year

Data Transformation

Looking at the distribution of our dependent variable number_of_visitors, I concluded that our data is not normally distributed, but it is slightly skewed. After running a QQ plot, we needed to transform our data to be normally distributed. The new QQ plot demonstrates that our log(number_of_visitors) values are much more normally distributed, allowing us to move forward with our analysis.

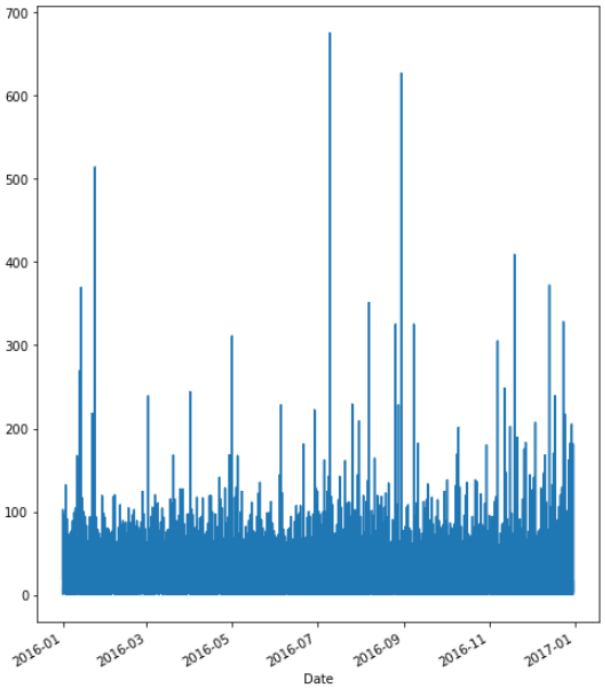


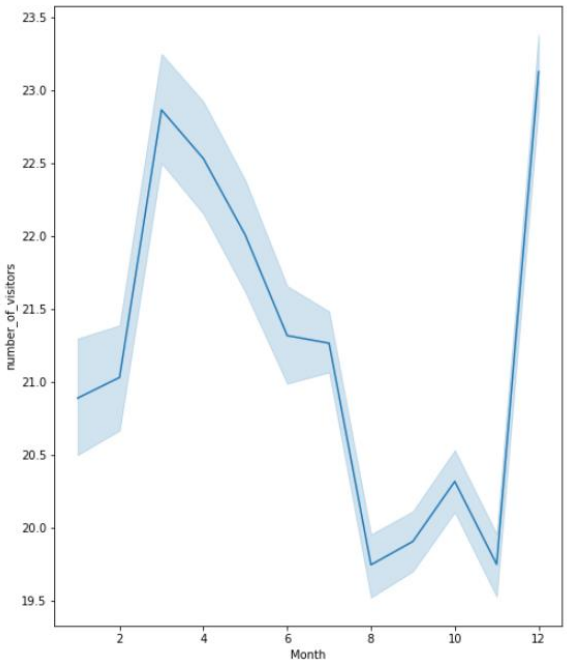


Another transformation I did was converting all our string columns to factor/categorical variables: Type, Holiday_Name, and Month.

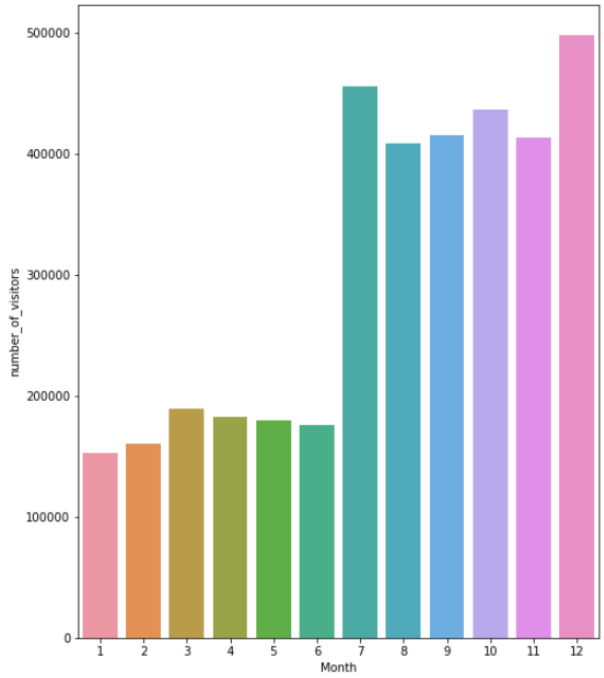
Data Exploratory

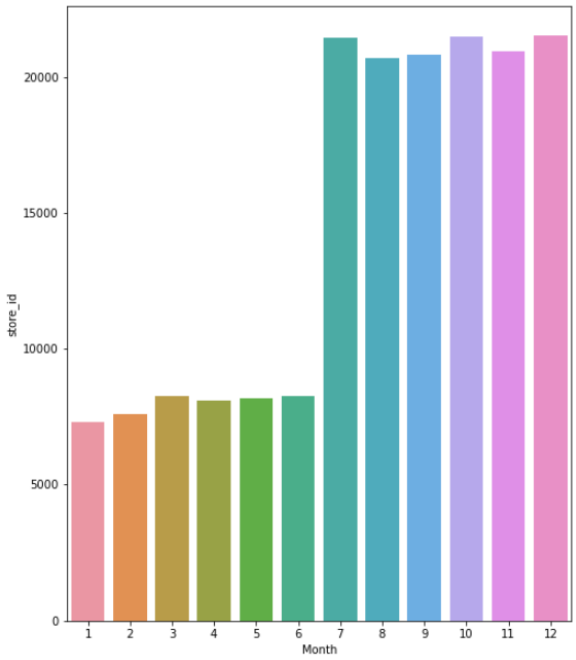
Moving to EDA, first, I wanted to identify any trend or seasonality pattern over the 2016 period. Looking at the number of visitors, we saw that the average number of visits per store is higher during the March-May period and increasing again in December. August-November period, meanwhile, underwent the lowest average number of visits.



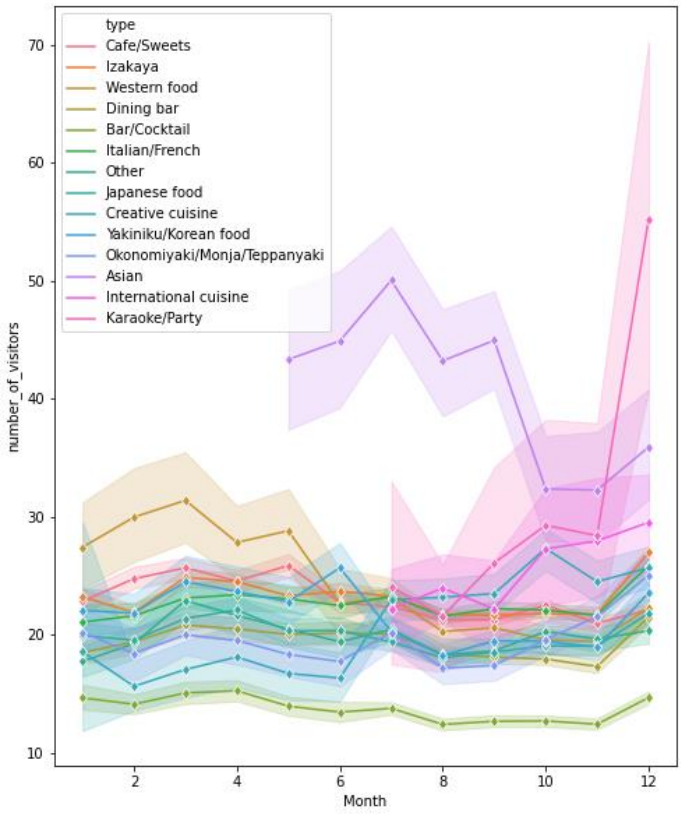


However, assuming that the average number of visits per store was affected by the number of stores opened during 2016, I used two visualizations to confirm my hypothesis. As a result, a significant number of stores opened during the second half of the year, increasing the total number of store visits.

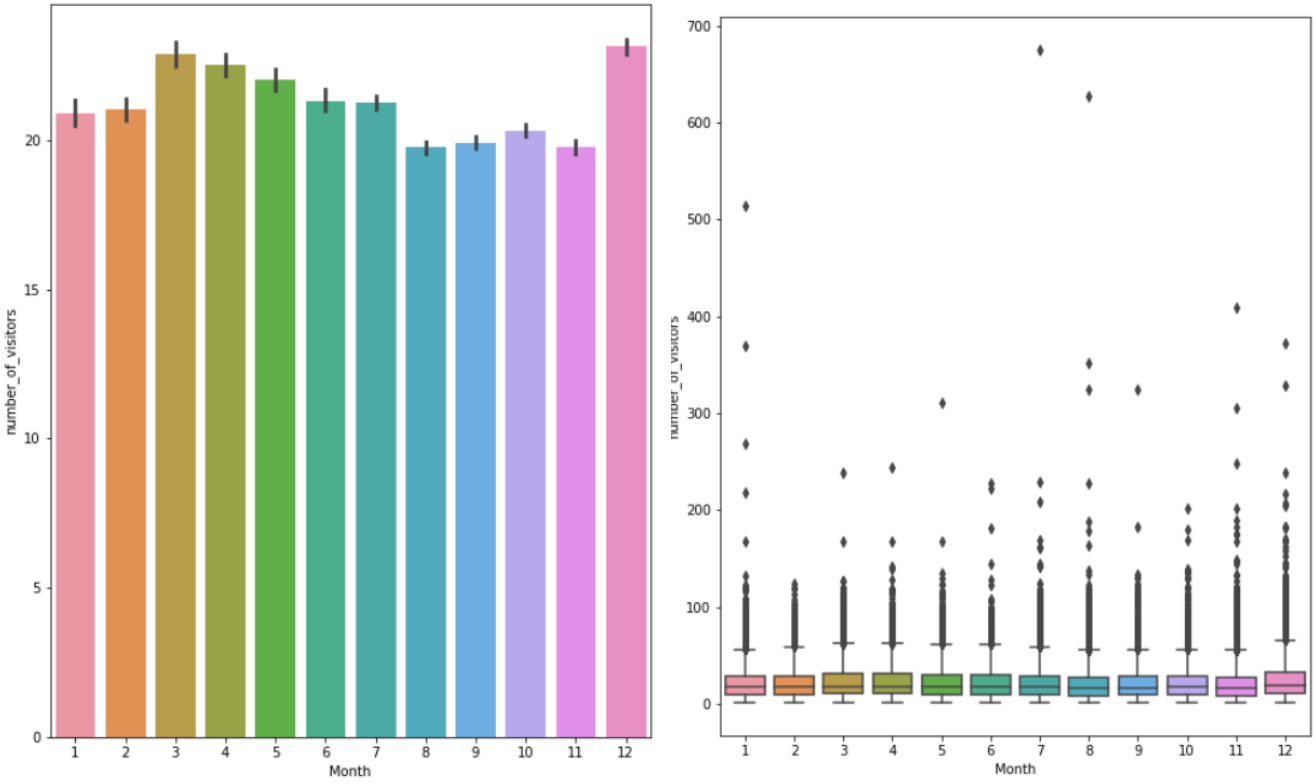




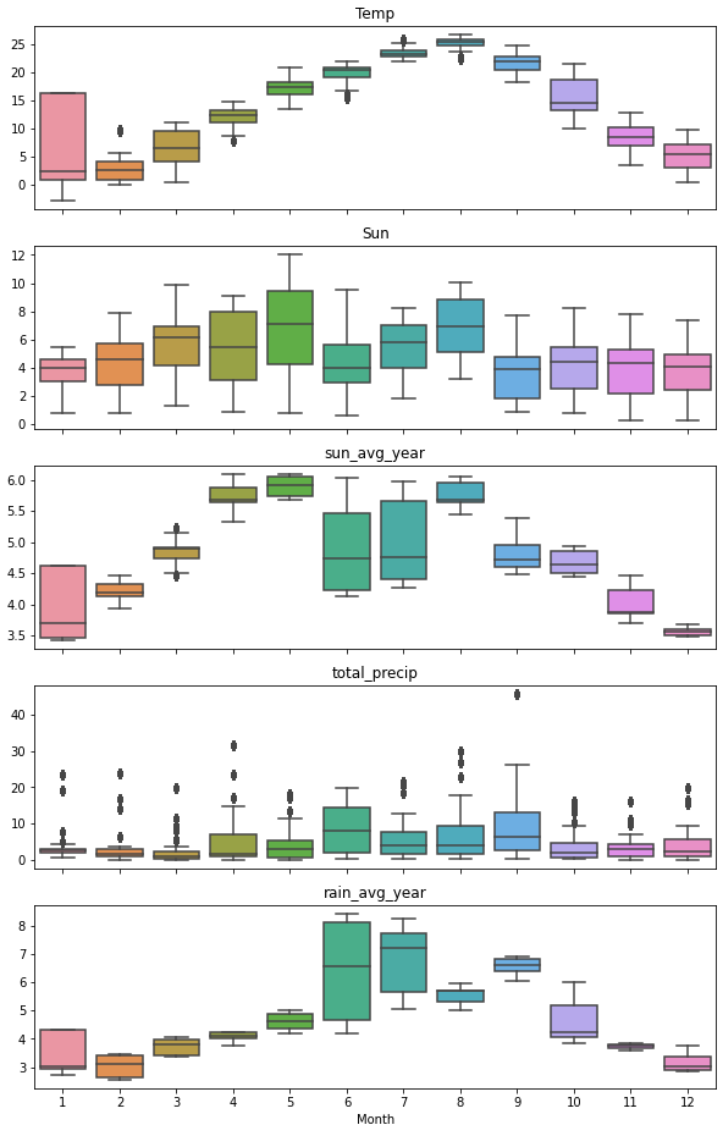
Considering Restaurant/Store types in the time analysis, we also saw that some of the new stores opened in the second half of 2016 were offering new categories that had a higher average number of visitors.



To dive deep into the other statistics of the number of visitors like variance, I also used boxplots as below. We saw that even the average number of visits per store didn't differ much; there were more variances since June due to the expansion of new stores.



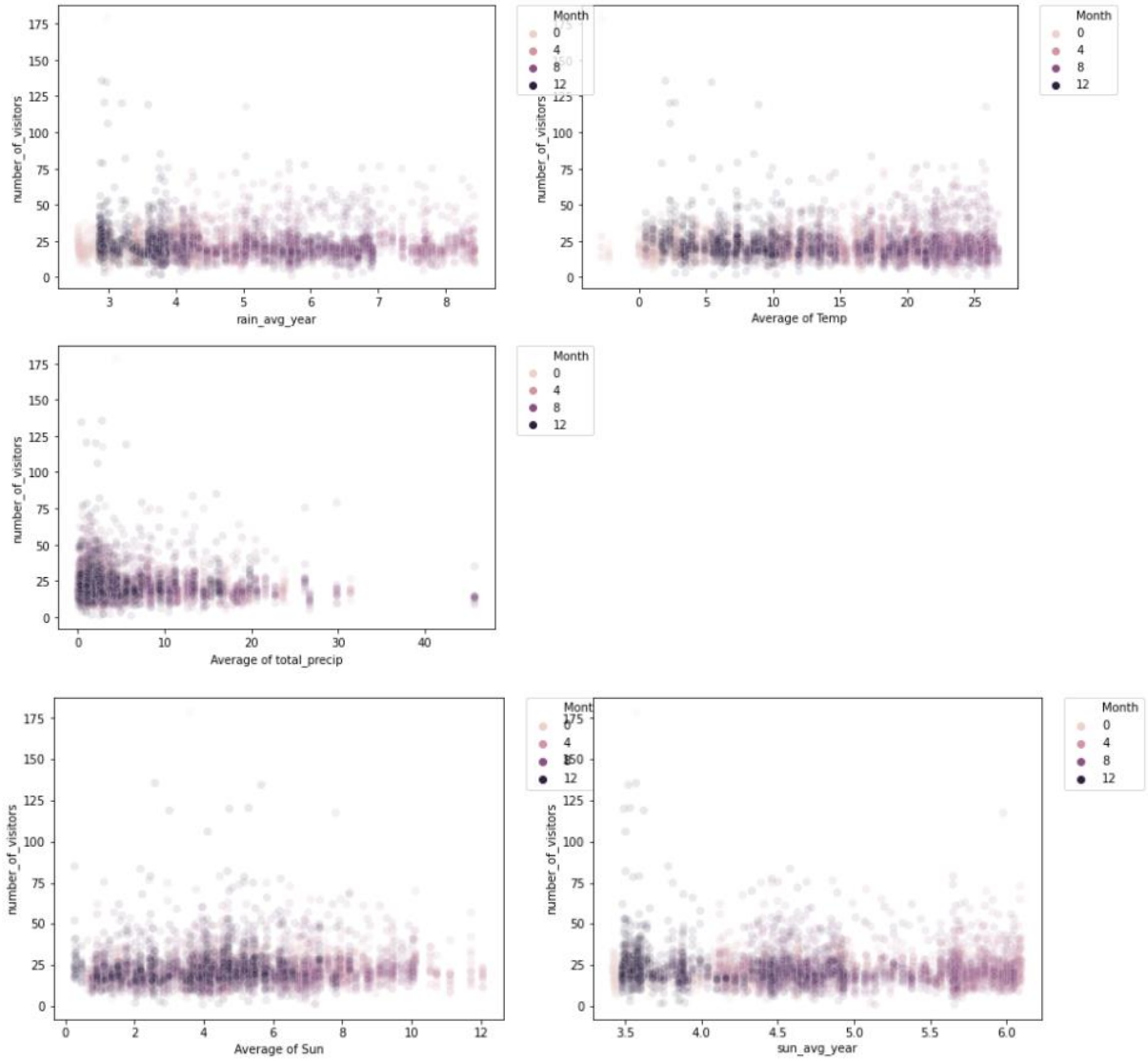
Using boxplot in the monthly period, I also researched the patterns of other numerical independent variables, most of which are weather features. There are no new findings that the temperature was higher in summer and the rain season took place during June and September. The coldest month was January, with the lowest temperature, while the hottest month was August.



To understand the characteristics and interactions of different variables of interest, I also used histograms to get an idea of other independent variables' frequency.

Next, I build 2-D boxplots to visualize the relationship between the number of visitors and independent variables. Taking into account the Holiday date, it is interesting that people least went out to store or restaurant on New Year's Day. Meanwhile, other occasions like The Emperor's Birthday or Showa day attracted a lot of people to visit stores. In terms of store/restaurant category, we saw the dominance of Asian food, which might be most accessible in an Asian country like Japan. Reversely, not many people went to other high-end places like Bar/Cocktail over the same period.

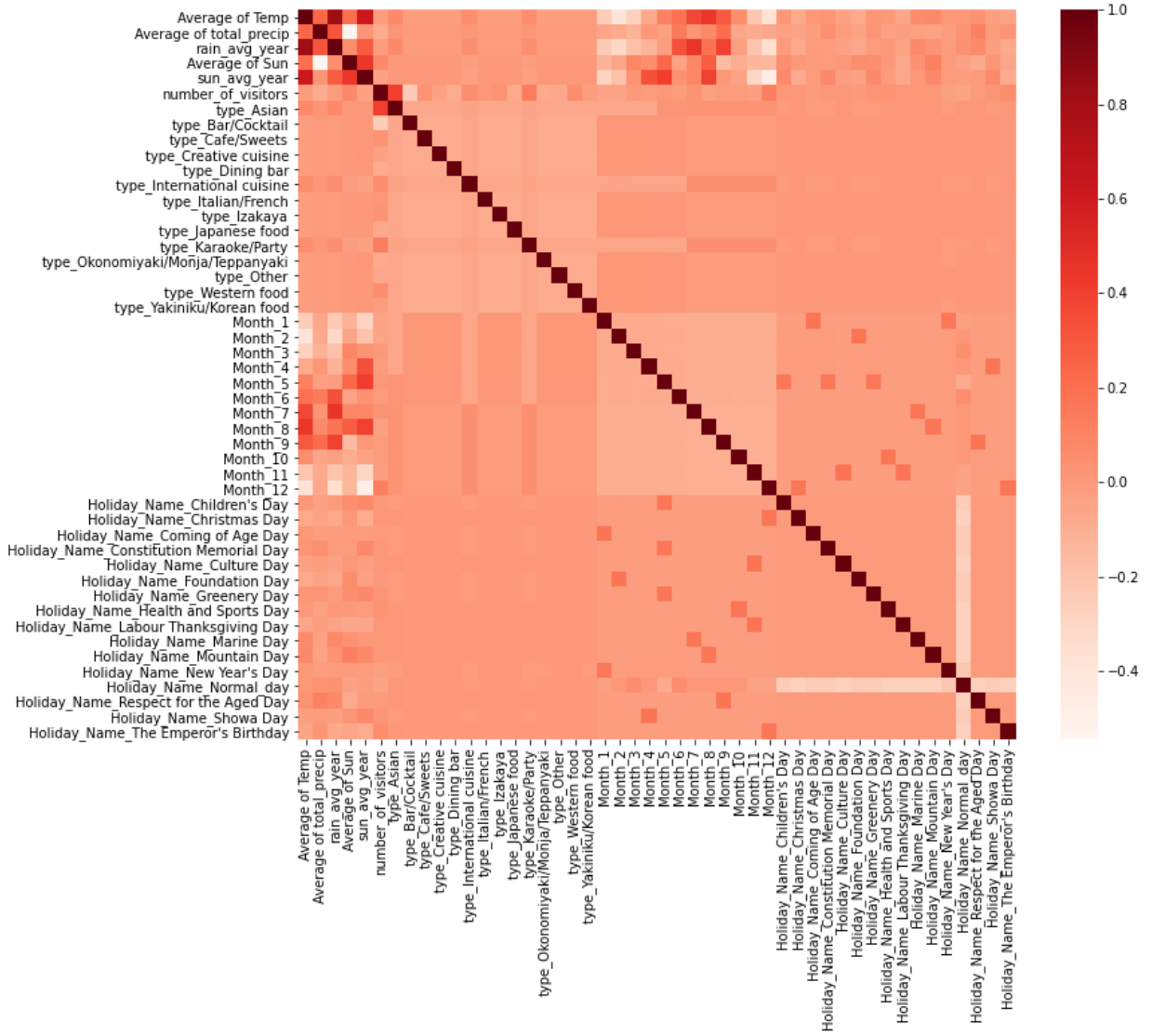
I also considered other numerical independent variables in the relationship analysis with the number of visitors. Building 3-D scatter plots with Month as the third attribute, we may agree that the only pattern worth mentioning is the Average of total Precip, which infers a negative relationship: lower precip, a higher number of visitors to stores. Besides talking about the number of visitors, we also knew that there were fewer sunny days and rainy days in December, partly explaining why people go out most often during the same period.



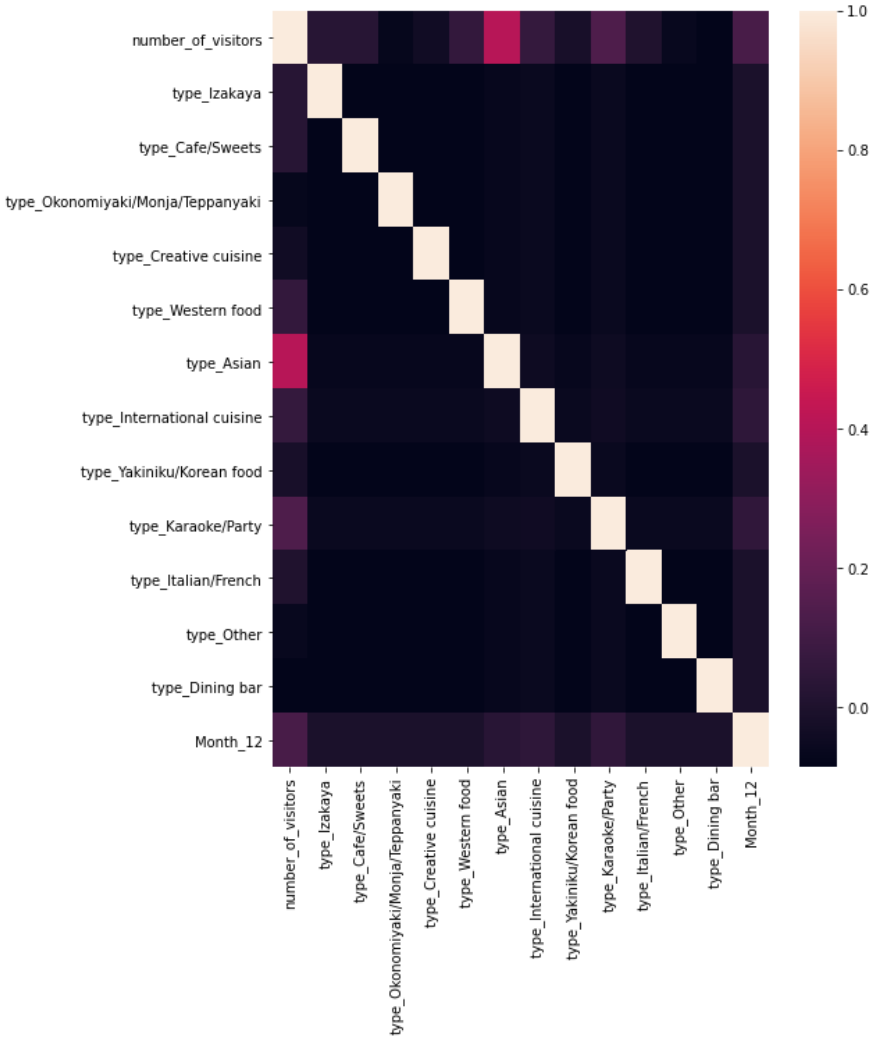
Correlation Analysis

To see which variables are likely to affect the number of visitors, I ran a correlation analysis of all selected independent variables against our dependent variable. Before doing this, I

converted all the categorical variables to dummy variables to calculate the correlation. Once this was completed, I chose any variable with more than 10% correlation with the number of visitors. This group of variables is called the top 13 variables of interest.



Last, I considered if the collinearity problem existed in our data with the current top 13 variables, using another correlation matrix:



We will take advantage of such regularization techniques as LASSO in the next modeling part to deal with other highly correlated variables.

Feature Selection and Modeling

For the Feature Selection and Modeling part, I decided to use three different methods. I will dive deep into each of these later in the analysis.

Method 1: Filter Method

As the name suggests, I used filter and take only the subset of the relevant features in this method. The model is built after selecting the features. The filtering here is done using a correlation matrix, and it is most commonly done using Pearson correlation.

We know that the number of visitors is highly correlated with a number of variables (our "Top 13"). We employed multi-linear regression to build an optimal prediction model for the number of visitors. I call this "Model 1", which included our "Top 13" explanatory variables (variables highly correlated with the number of visitors). Using the Train-Test Split method, we have Model 1 summary as below:

```
# Model Evaluation
from sklearn import metrics

# Re-Print the model R2 score:
print(lm1.score(X_train1, np.log(y_train1)))

# Print result of MAE
print(metrics.mean_absolute_error(np.log(y_test1), (y_pred1)))

# Print result of MSE
print(metrics.mean_squared_error(np.log(y_test1), (y_pred1)))

# Print result of RMSE
print(np.sqrt(metrics.mean_squared_error(np.log(y_test1), (y_pred1))))
```

```
0.2156589663674543
0.25719778903575413
0.12216487904994613
0.3495209279141181
```

In Model 1, all identified variables are highly correlated with our target variable – the number of visitors. All the variables have a positive relationship with the target variable. Type_Asian has the most substantial marginal impact on the target variable.

Method 2: Wrapper Method

A wrapper method needs one machine learning algorithm and uses its performance as evaluation criteria. It means you feed the features to the selected Machine Learning algorithm, and based on the model performance, you add/remove the features. Though it is an iterative and computationally expensive process, it is more accurate than the filter method. I will talk about two wrapper methods in this analysis: Backward Elimination and RFE.

Model 2.1: **Backward Elimination**

I will fit all the possible features into the model at first. After checking the model's performance, it will iteratively remove the worst performing features one by one till the overall performance of

the model reaches an acceptable range (using a p-value which is above 0.05). I call this "Model 2.1", which includes all significant explanatory variables. Using the Train-Test Split method, we have Model 2.1 summary as below:

```
# Model Evaluation
from sklearn import metrics

# Re-print the R2 score
print(lm21.score(X_train21, np.log(y_train21)))

# Print result of MAE
print(metrics.mean_absolute_error(np.log(y_test21), (y_pred21)))

# Print result of MSE
print(metrics.mean_squared_error(np.log(y_test21), (y_pred21)))

# Print result of RMSE
print(np.sqrt(metrics.mean_squared_error(np.log(y_test21), (y_pred21))))
```

```
0.2826141021399394
0.2383622392913331
0.1179616814952667
0.34345550147765386
```

In Model 2.1, we found out some variables have a negative relationship with the target variable – the number of visitors. The most significant one is type_Bar/Cocktail, suggesting that people least go to this place. Like Model 1, Type_Asian has the most positively substantial marginal impact on the target variable.

Model 2.2: **RFE (Recursive Feature Elimination)**

This method recursively removes attributes and builds a model on those attributes that remain. It considers accuracy metrics to rank the feature according to their importance. The RFE method takes the model to be used and the number of required features as input. It then ranks all the variables, with 1 being most important. It also gives its support, True as a relevant feature and False as an irrelevant feature.

After finding out 12 as the Optimum number of features, I fit these 12 variables and call this "Model 2.2". Using the Train-Test Split method, we have Model 2.2 summary as below:

```
# Model Evaluation
from sklearn import metrics

# Re-print R2 score
print(lm22.score(X_train22, np.log(y_train22)))

# Print result of MAE
print(metrics.mean_absolute_error(np.log(y_test22), (y_pred22)))

# Print result of MSE
print(metrics.mean_squared_error(np.log(y_test22), (y_pred22)))

# Print result of RMSE
print(np.sqrt(metrics.mean_squared_error(np.log(y_test22), (y_pred22))))
```

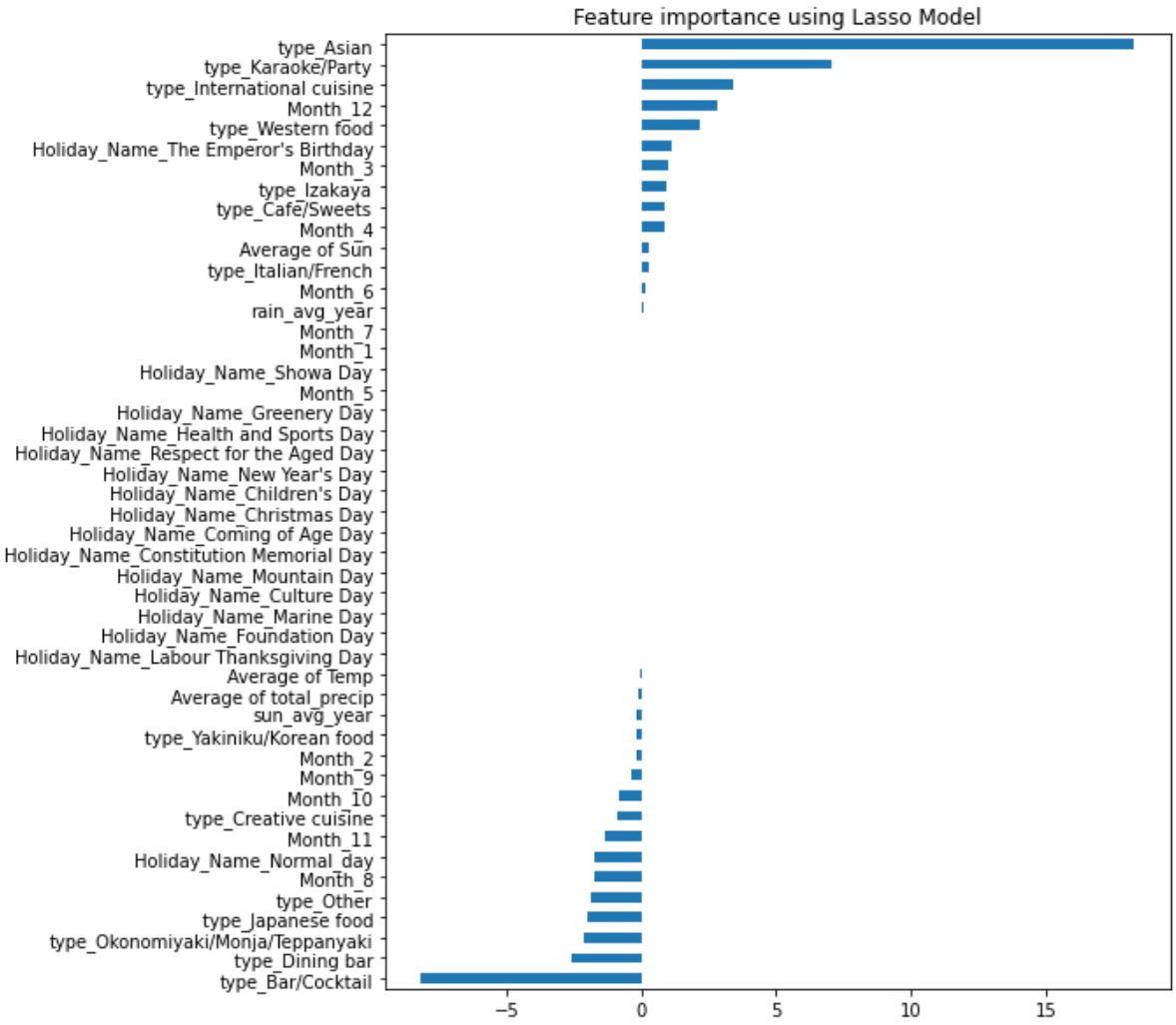
```
0.1869795634258622
0.2509307472009291
0.1263953354590881
0.3555212166089221
```

In Model 2.2, we saw that all the variables have a negative relationship with the target variable – the number of visitors. Like Model 2.1, the most significant one is type_Bar/Cocktail, suggesting that people least go to this place.

Method 3: Embedded Method

Regularization methods are the most commonly used embedded methods which penalize a feature given a coefficient threshold. I will do feature selection using Lasso. If the variable is irrelevant, Lasso penalizes coefficients and makes it 0. Hence the features with coefficient = 0 are removed, and the rest are taken.

Using the LASSO regression method, we came up with "Model 3" that performs both variable selection and regularization. Diving deep into variable selection, we have the top 30 predictors most important to the model. It is done by using MDI (Gini Importance or Mean Decrease in Impurity) that calculates each feature's importance as the sum over the number of splits (across all trees) that include the feature, proportionally to the number of samples it splits.



Model 3 has a summary as below:

```
# Model Evaluation
from sklearn import metrics

# Re-print R2 score
print(lm33.score(X_train33, np.log(y_train33)))

# Print result of MAE
print(metrics.mean_absolute_error(np.log(y_test33), (y_pred33)))

# Print result of MSE
print(metrics.mean_squared_error(np.log(y_test33), (y_pred33)))

# Print result of RMSE
print(np.sqrt(metrics.mean_squared_error(np.log(y_test33), (y_pred33))))
```

```
0.279191147244356
0.24296487209289722
0.12109412530975708
0.3479858119374367
```

In comparison with Model 1, Model 2.1, and Model 2.2, we have additional insights from the Month variable group. However, once again, we can confirm the reverse impact on the target variable of Type_Asian (positive) and Type_Bar/Cocktail (negative).

Evaluation

After running our four models, I used four metrics: R-squared, MAE, MSE, RMSE, to evaluate our model prediction performance. As we expected from the table below, Model 2.1 using Backward Elimination is the best in terms of all the metrics. Another model that has a very close performance behind is Model 3 using LASSO Regularization.

It is expected that the Filter method (Model 1) is less accurate, and Wrapper and Embedded methods (Model 2.1 and Model 3) offer more accurate outcomes. However, these two later models will be computationally expensive in practice when dealing with a large number of features.

	Model 1	Model 2.1	Model 2.2	Model 3
R2	0.2157	0.2826	0.1870	0.2792
MAE	0.2572	0.2384	0.2509	0.2430
MSE	0.1222	0.1180	0.1264	0.1211
RMSE	0.3495	0.3435	0.3555	0.3480

Deployment

By analyzing a group of data sets associated with store traffic, I was able to create a model that can help top management team, resource planners, and especially store managers within the F&B industry predict the foot visits to their stores/restaurant and have a better understanding of customer's preference in Japan market. I have found that Model 2.1 using Backward Elimination performed better than others. I also saw an impressive prediction accuracy from Model 3 that uses LASSO Regularization. In general, I found out some variables have a positive relationship with the number of visitors while others have a negative relationship with the target variable. There are fewer people who go to stores/restaurants during the cold season (Month 8, 9, 10, and 11), and Type_Bar/Cocktail is the least popular place. However, the traffic increases again in December and reaches its peak in the summer when the weather is more supporting shopping activities. It is also clear that Type_Asian has the most positively substantial marginal impact on

the number of visitors. Another finding worth mentioning is the group of holiday variables since it does not considerably influence shopping or dining decisions on Japanese customers.

However, this analysis has some limitations. First, the data set had many missing values from some attributes, which can narrow down the accuracy of our prediction. The solution for this is to include more qualified data by working with the data provider for the mining and clean step. Second, based on the agreement with the data provider, I was not able to use variables regarding store/restaurant location, which may be good predictors of store traffic. Another limitation is a lack of data features relevant to store/restaurant, like the store capacity, store rating, and customer service. These factors can impact the power of attracting people's visits. We can deal with this problem by asking for more internal data from the data provider. The last thing is about the data time period. In this analysis, we only had 2016 store traffic data, which may not reflect the business and market's current situation. Proposing a further extension of this analysis to the data provider by including more updated data should be considered. In the future, we also can try other performance measures and other machine learning techniques for better performance and comparison of results.