# Golden Week Restaurant Forecasting

Heidi E. Schmidt

Image attribution https://www.tofugu.com/japan/golden-week/

# Golden Week - Restaurant visitor forecasting in Japan

https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting

## Goal

- Forecast future visitors to Air and HPG restaurants all over Japan
- Use RMSLE - root mean squared log error - this accounts for wide data sets (lots of features) and reduces the error in a order of magnitude. Instead of 0 to infinity - a visually understood smaller numeric range

## Metrics and assumptions.

- 2 data sets - only  valid one air
- Log is great at reducing noise like dummy variables are great for breaking out details
- Permutation testing is also possible using Random Forest with less heart ache

## Random Forest Model pursued after LR tried

- LR  showed base intercepts were widely off as more features added (extreme + and -)
- Random Forest allowed me to use the permutations of the many dummy variables created and not have to track them all down.

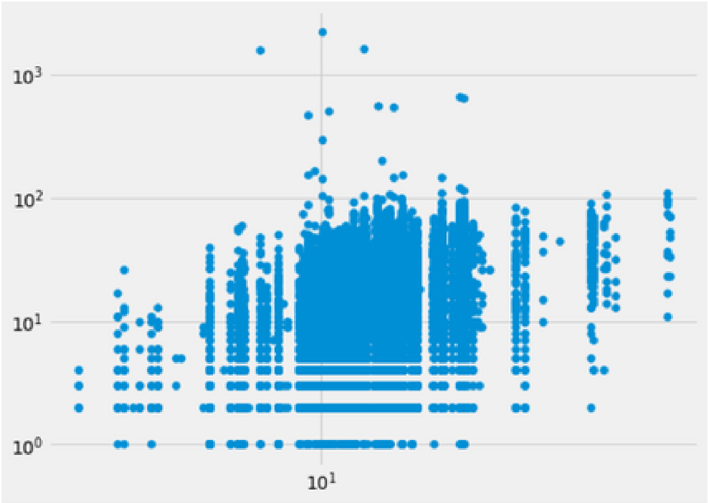
## Learning what I don't know as I went

- Wanting more information the data set didn't have hours, for ex.
- Spending too much time trying to merge two data systems that don't have a good mapping file. Not needed.
- Having too many features: trying to get at the ideal information by day of week, month, genre, location and hanging my notebook :(

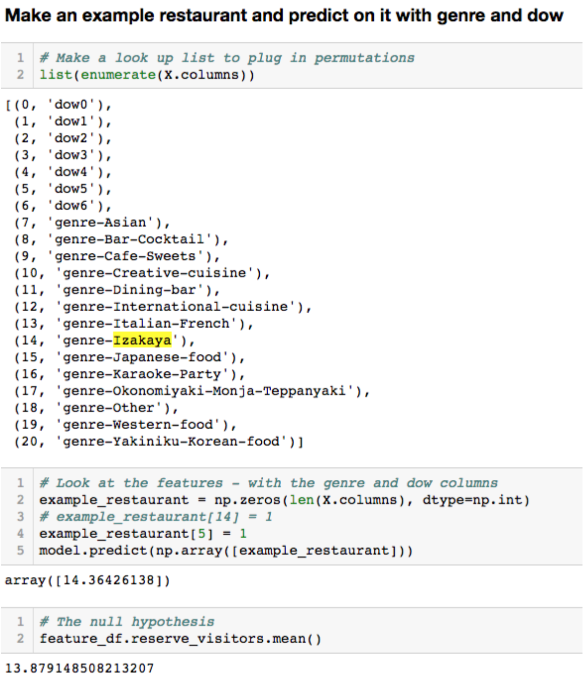
# Impact of findings - Some of the high lights

- Avg (mean) visitors is 13.87
- RMSLE with dow, genre, is_holiday is 0.8382072682672774
- Predicted base visitors just on dow = 10.69 visitors
- Best days of the week to have the most visitors based on dow, genre, is_holiday are Thursdays and Fridays respectively 16.04 - 16.6
- If a Holiday, base visitors is 9.40
- Best months of the year are Jan-May in 2017 and Oct-Dec 2016
- Best food options to draw in visitors are bar dining and Izayaka (pub food), Teppanyaki (savory pancakes), Italian-French, Western
- Best locations are Tokyo, Osaka, Hiroshima, and Fukuoka

# Linear Regression vs Random Forest




Plotting predicted Visitors logarithmically visually shows a relationship (with stratification of low and high end outliers)



**Make an example restaurant and predict on it with genre and dow**

```
# Make a look up list to plug in permutations
list(enumerate(X.columns))
```

```
[(0, 'dow0'),
 (1, 'dow1'),
 (2, 'dow2'),
 (3, 'dow3'),
 (4, 'dow4'),
 (5, 'dow5'),
 (6, 'dow6'),
 (7, 'genre-Asian'),
 (8, 'genre-Bar-Cocktail'),
 (9, 'genre-Cafe-Sweets'),
 (10, 'genre-Creative-cuisine'),
 (11, 'genre-Dining-bar'),
 (12, 'genre-International-cuisine'),
 (13, 'genre-Italian-French'),
 (14, 'genre-Izakaya'),
 (15, 'genre-Japanese-food'),
 (16, 'genre-Karaoke-Party'),
 (17, 'genre-Okonomiyaki-Monja-Teppanyaki'),
 (18, 'genre-Other'),
 (19, 'genre-Western-food'),
 (20, 'genre-Yakiniku-Korean-food')]
```

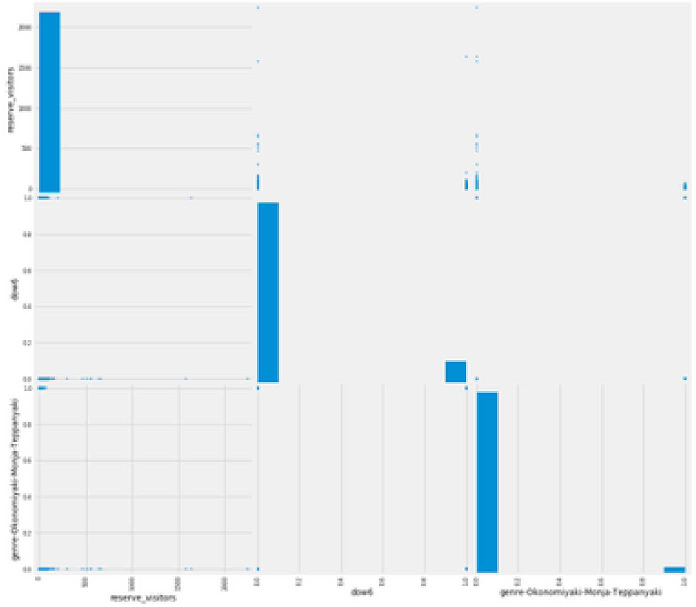
```
# Look at the features - with the genre and dow columns
example_restaurant = np.zeros(len(X.columns), dtype=np.int)
# example_restaurant[14] = 1
example_restaurant[5] = 1
model.predict(np.array([example_restaurant]))
```

```
array([14.36426138])
```

```
# The null hypothesis
feature_df.reserve_visitors.mean()
```

```
13.879148508213207
```

Being able to use the RF predicted value of visitors with ea individual feature brought RMSLE to 0.838 from 0.839




Versus a LR scatter plot matrix of visitors to key features

# Recommendations for future development

- Roll up more data and have data sets spanning full years for comparison.
- Drop the fake lat/lon and use ward, chome, ku (like state, city, city block level) to generalize locations. Map those to numbers.
- Make the two types of stores, air and hpg, genre's normalized and mapped. Their information for even ones that mapped together were not the same. Left big holes of information - i.e.  missing data that may have been useful.
- Take into account weather temperature
- Add Random Forest  R graphs to show which decision trees had the most value.

# Lessons Learned

## The tales of the notebook

- Iteration through > 22 features to determine the right fit hosed my notebook and caused me to have to go back to an old copy and reverse engineer my code snippets back in

- There is never enough time when you don't understand that all your features are not in the same units that you've practiced in

- Learned a bunch in class and will keep on working on tutorial kaggles. This is a marathon, not a sprint.

- The ocean of data science is deep, wide, nifty and will take considerable swimming in to get proficient over time.