SHRIRAM **HANGARGEKAR**

# RESTAURANT'S VISITOR FORECASTING

IBM ADVANCED DATA SCIENCE CAPSTONE PROJECT

# VISITOR FORECASTING

- What is the problem?
- Why is it an important problem?
- So, what is the solution?
- What is needed for solution?
- How is the solution achieved?

# EXTRACT – TRANSFORM - LOAD




Transform




- Connect to Data Source
- Download csv's in file
- Load in Pandas data-frame

- Handle missing data
- Adjusting the ranges
- Matching data types

- Merge all data
- Connect to server
- Store as csv

# ETL - DATA SOURCE

② **AIR RESERVATION SYSTEM STORE INFO DATA**

| AIR System ID | string |
|---|---|
| Genres | string |
| Area Location | string |
| Longitude | float |
| Latitude | float |

① **AIR RESERVATION SYSTEM RESERVATION DATA**

| Reservation ID | int |
|---|---|
| Reservation Date | datetime |
| Reservation Store | string |
| Visit Date | datetime |
| Reserved Visitors | string |

④ **NATIONAL HOLIDAY DATA**

| Date | datetime |
|---|---|
| Day | string |
| Flag | boolean |

③ **HISTORIC VISITORS' DATA**

| Visit Date | datetime |
|---|---|
| Reservation Store | string |
| Total Visitors | int |

⑦ **STORE ID RELATIONSHIP**

| Store ID | int |
|---|---|
| AIR System ID | string |
| HPG System ID | string |

⑥ **HPG RESERVATION SYSTEM STORE INFO DATA**

| HPG System ID | string |
|---|---|
| Genres | string |
| Area Location | string |
| Longitude | float |
| Latitude | float |

⑤ **HPG RESERVATION SYSTEM RESERVATION DATA**

| Reservation ID | int |
|---|---|
| Reservation Date | datetime |
| Reservation Store | string |
| Visit Date | datetime |
| Reserved Visitors | string |

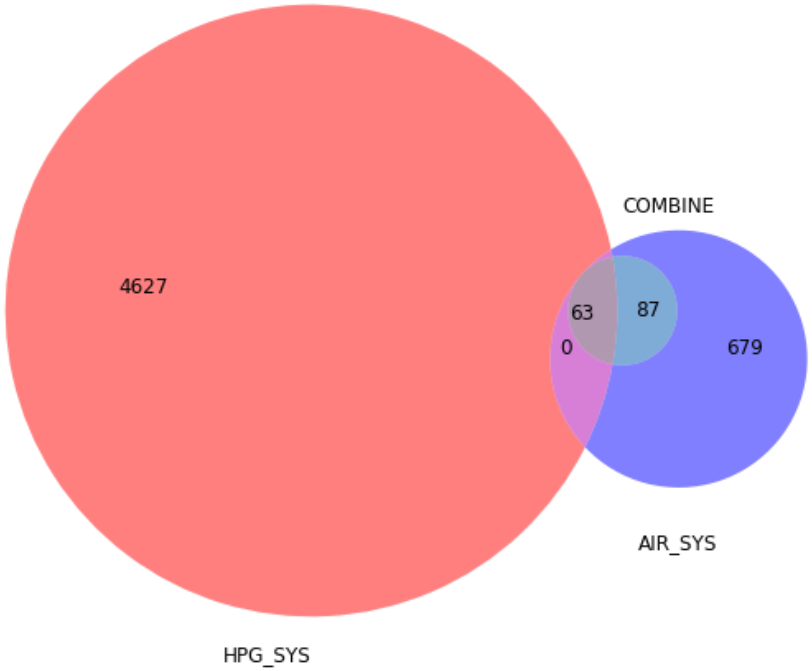
| | Name | Date modified |
|---|---|---|
| ① | air_reserve | 11/21/2017 7:31 AM |
| ② | air_store_info | 11/21/2017 7:31 AM |
| ③ | air_visit_data | 11/21/2017 7:32 AM |
| ④ | date_info | 11/21/2017 7:32 AM |
| ⑤ | hpg_reserve | 11/21/2017 7:32 AM |
| ⑥ | hpg_store_info | 11/21/2017 7:31 AM |
| ⑦ | store_id_relation | 11/21/2017 7:32 AM |

*Source:* [https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting/data](https://www.kaggle.com/c/recruit-restaurant-visitor-forecasting/data)

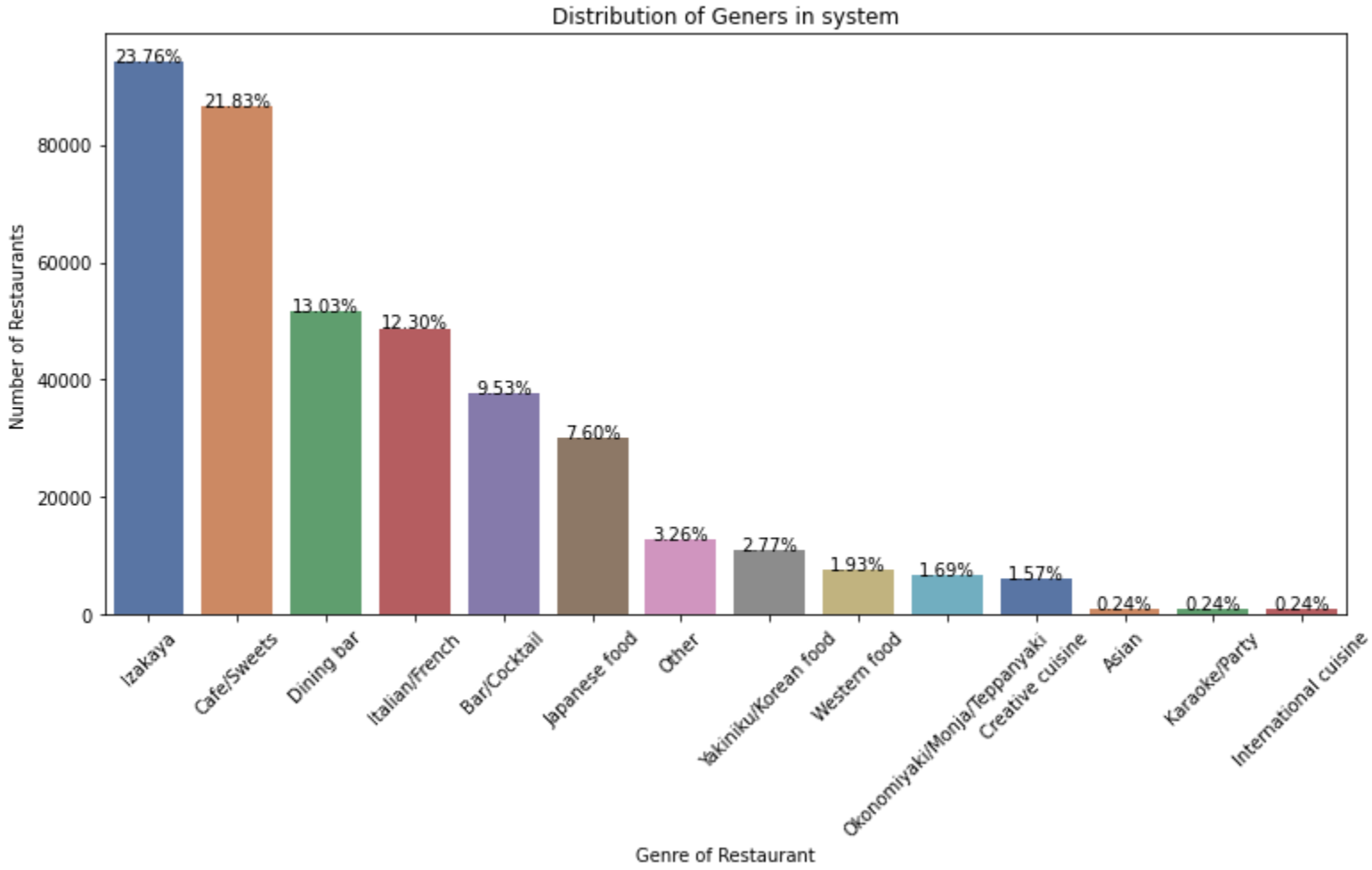
# RESTAURANTS BY SYSTEMS

| - | AIR_SYS | HPG_SYS |
|---|---|---|
| Unique | 679 | 4627 |
| Combine | 150 | 63 |
| Explicit | 0 | 87 |

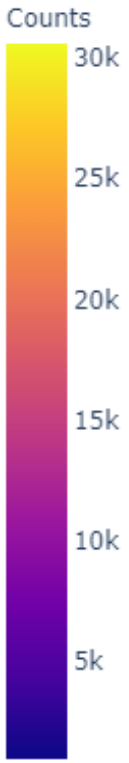
System-wise distribution of hotels



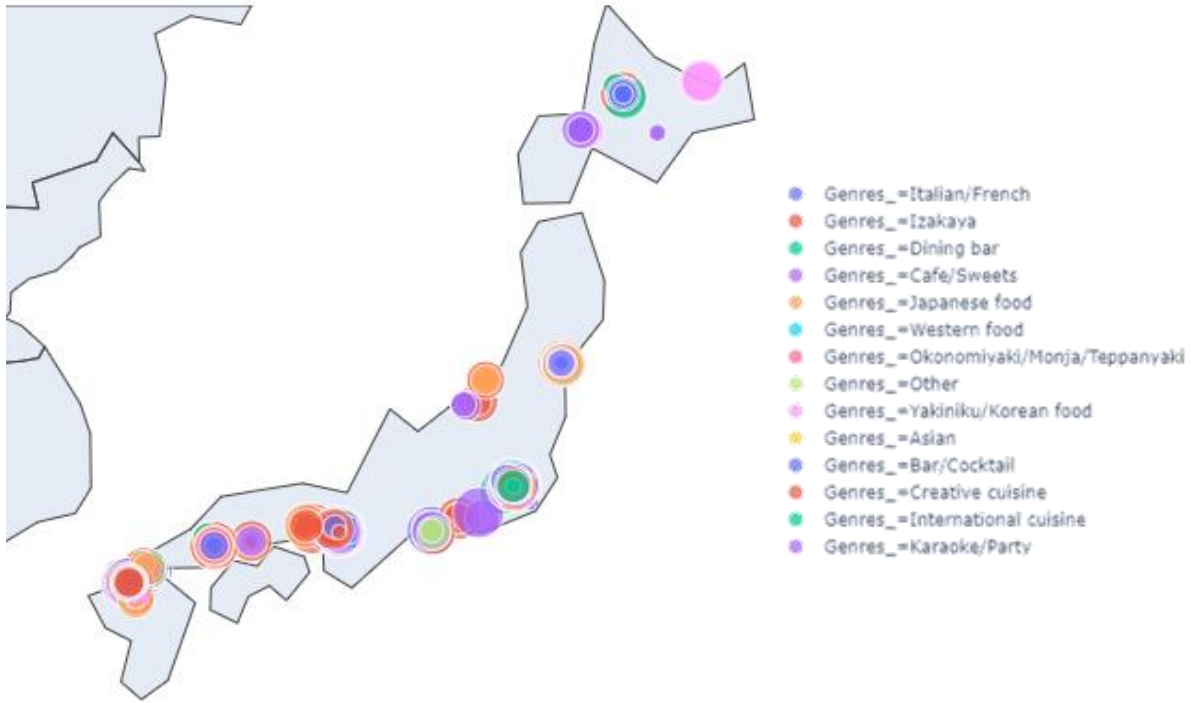
# RESTAURANTS BY GENRES



# RESTAURANTS AND VISITORS BY LOCATION

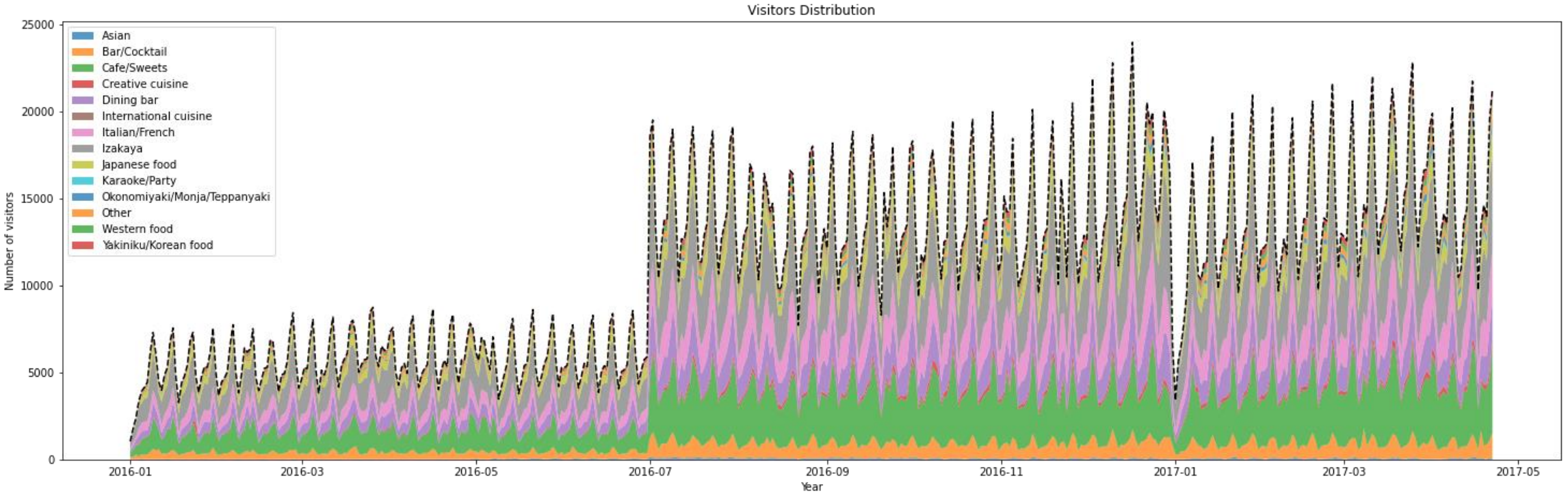




Restaurants by Area

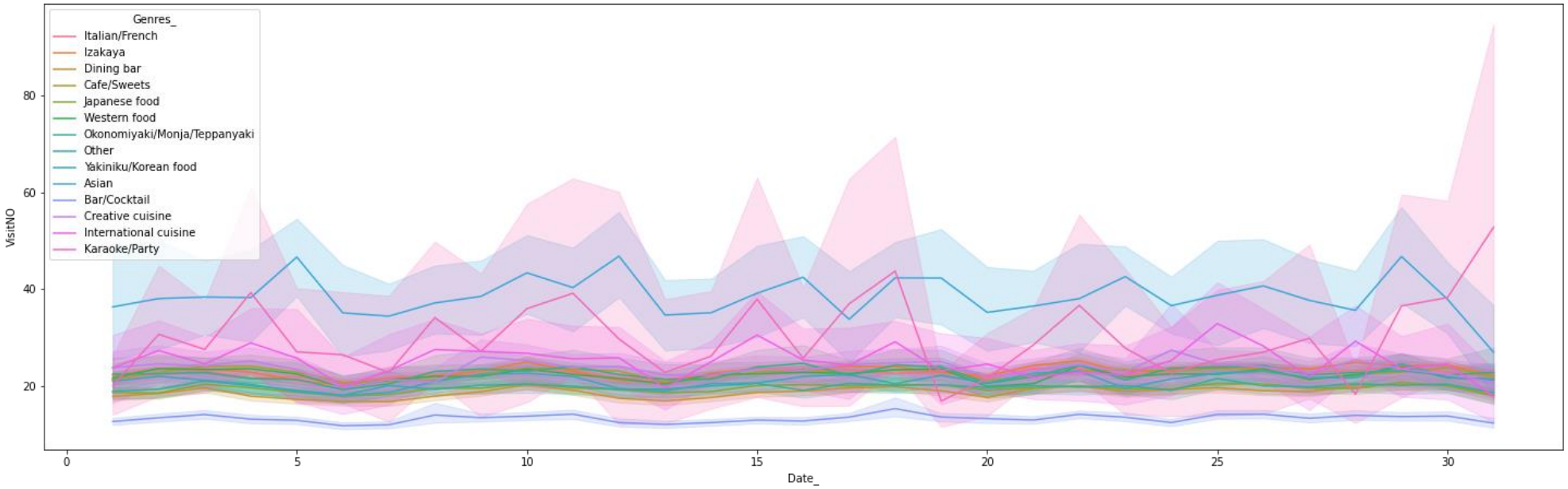



Visitors by Area

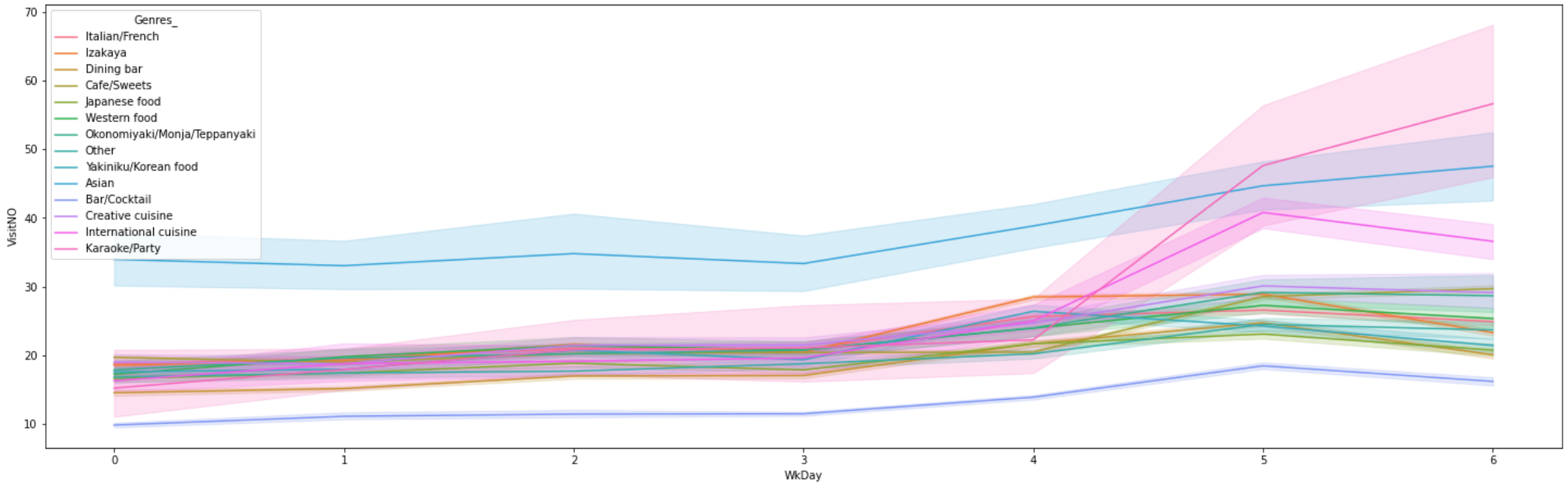
# VISITORS TIMESERIES PLOT



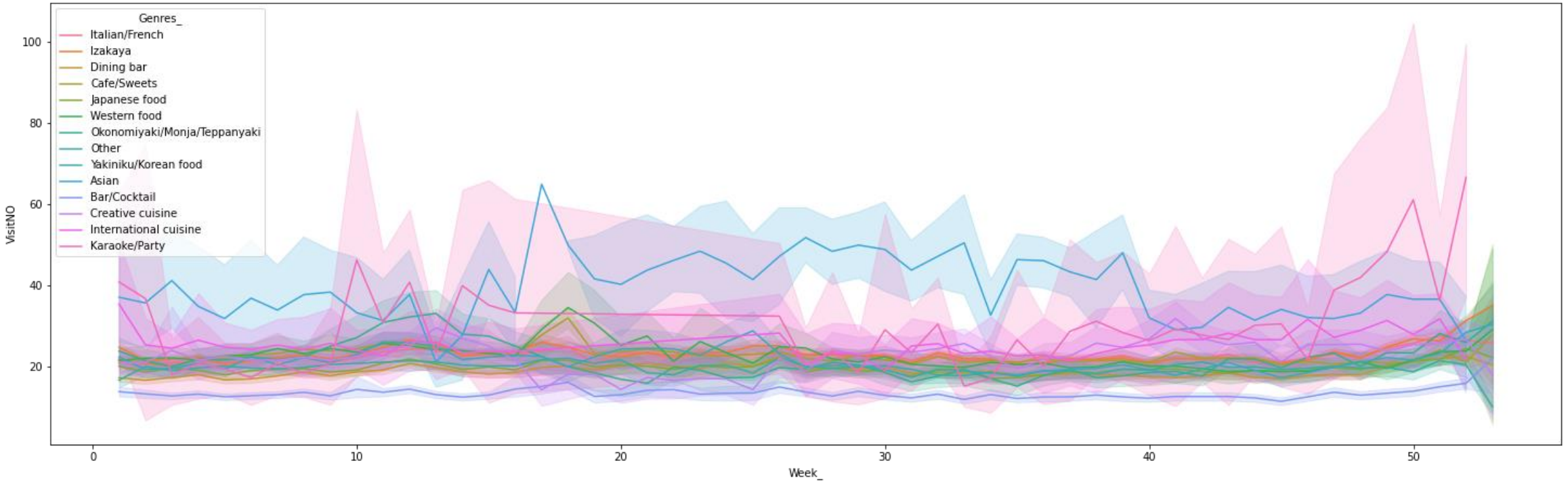
# TRENDS IN VISITORS – OVER MONTH



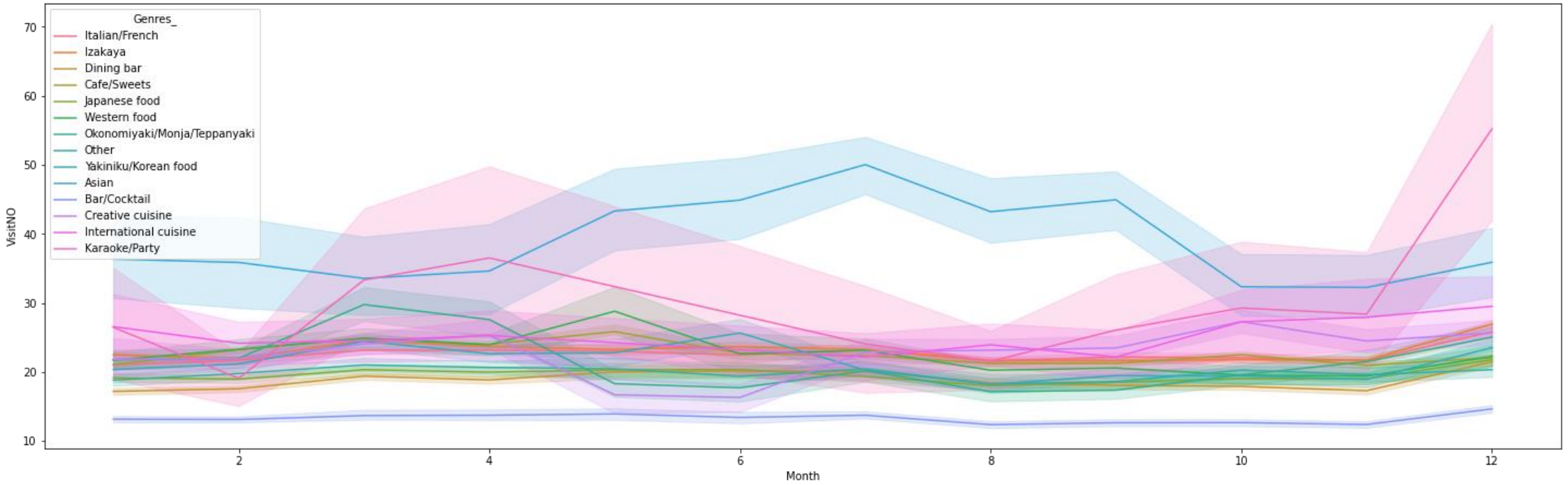
# TRENDS IN VISITORS – OVER WEEK



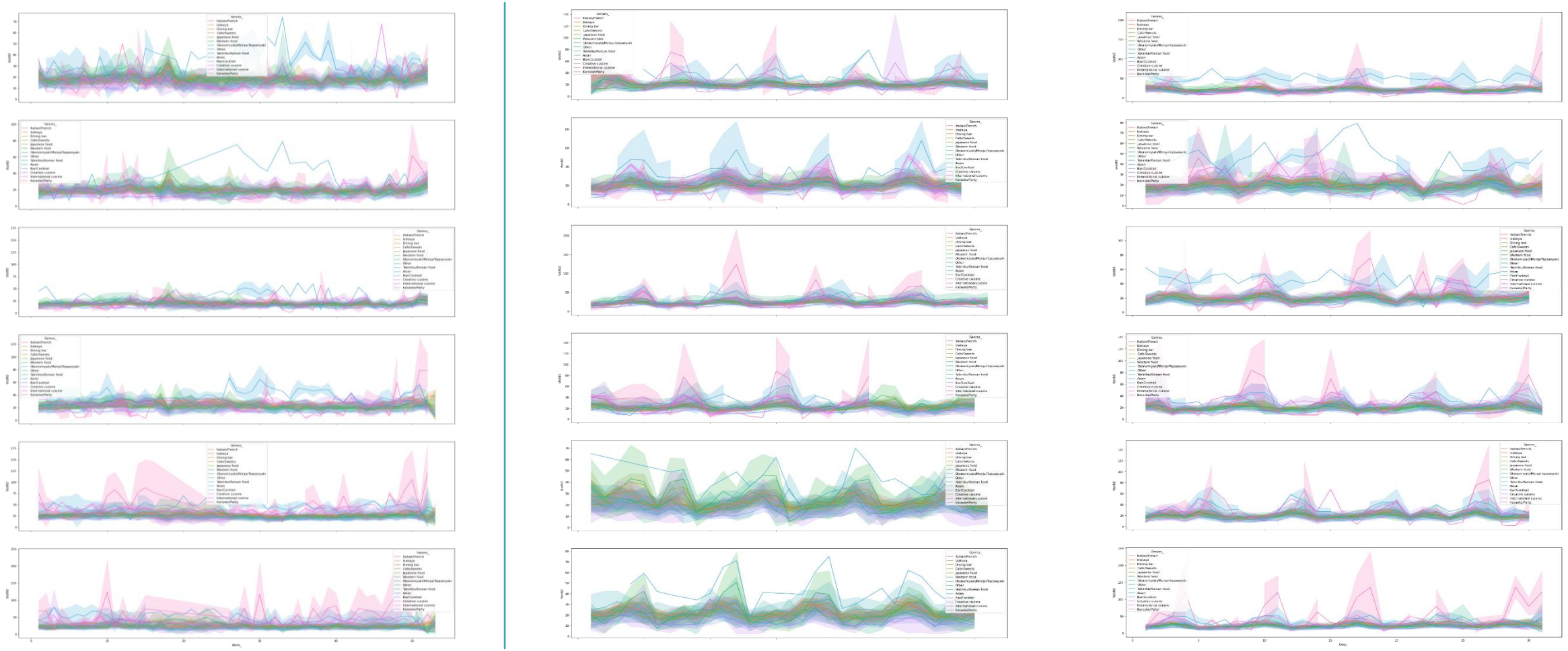
# TRENDS IN VISITORS – OVER YEAR



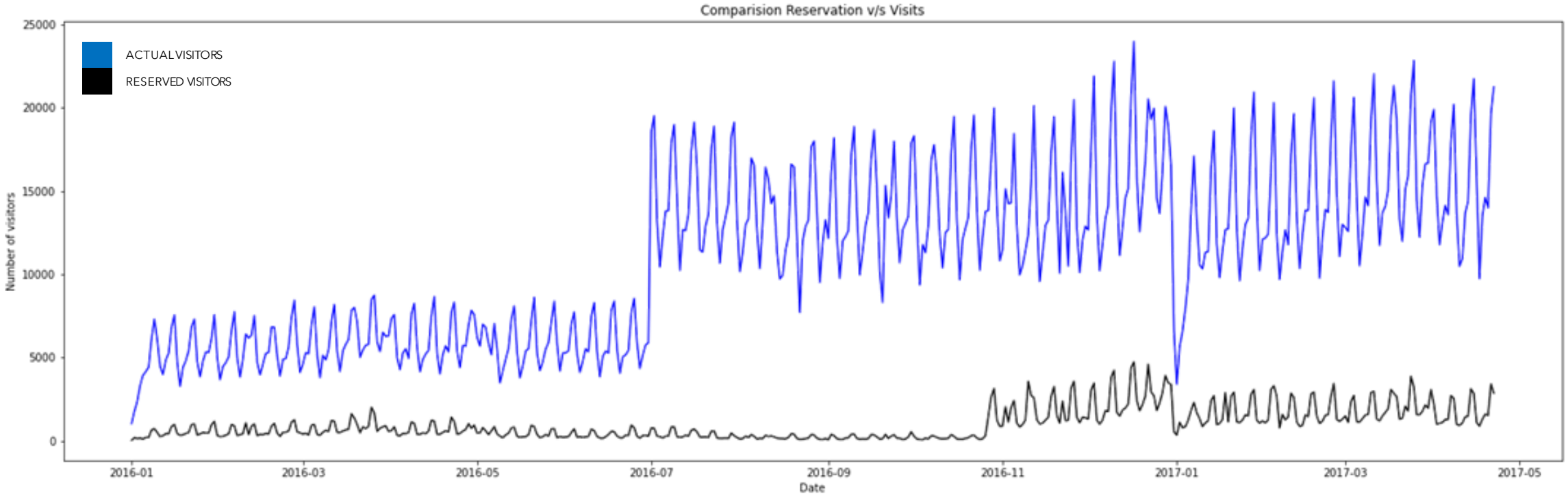
# TRENDS IN VISITORS – OVER YEAR



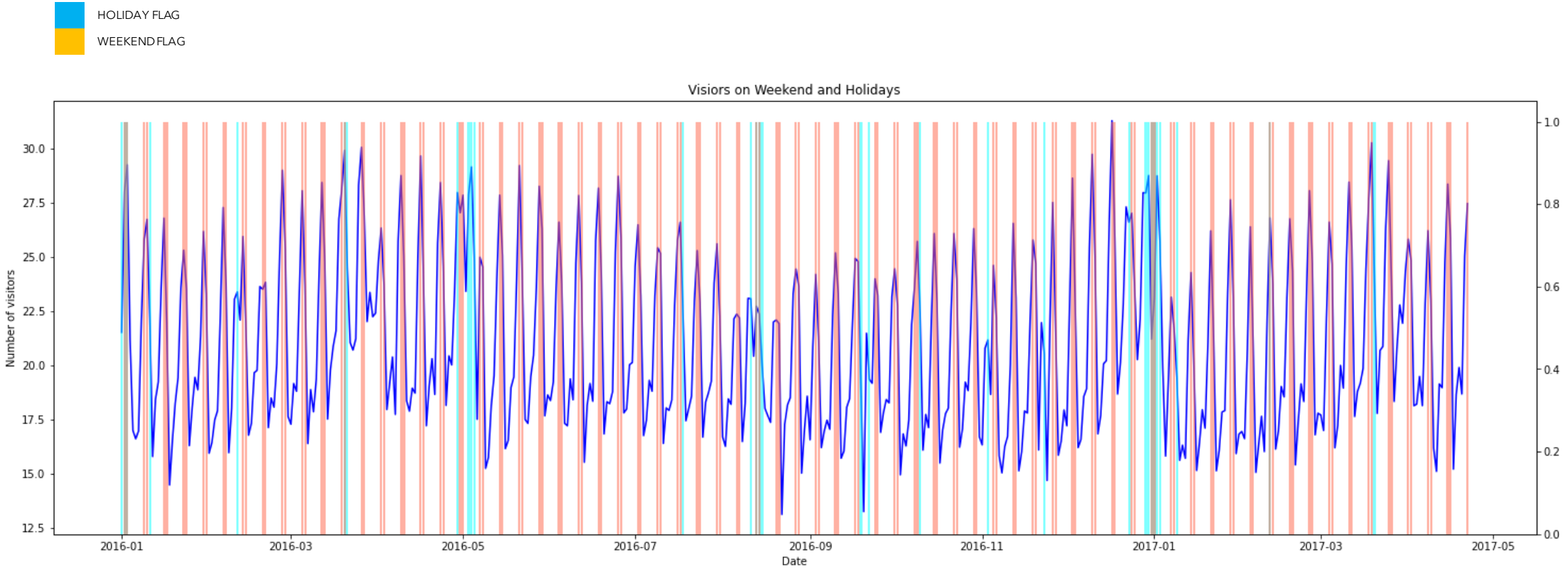
# VISITORS VISITING PATTERN ACROSS YEAR



# VISITORS – ACTUAL V/S RESERVED

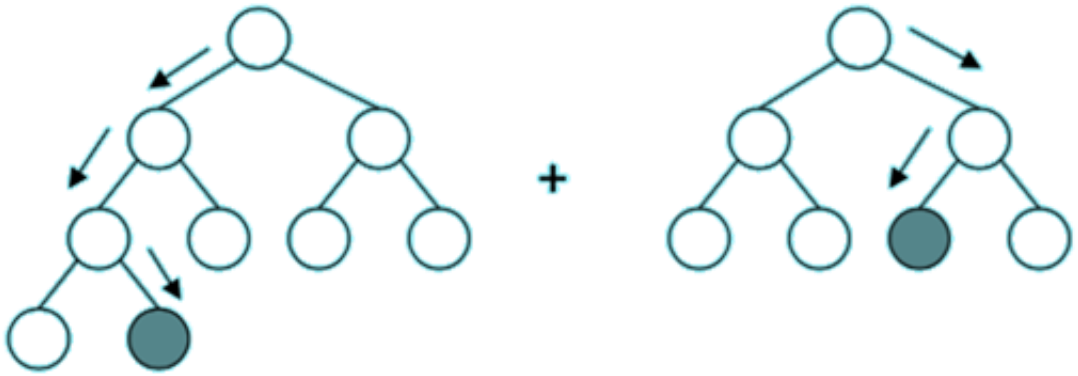


# VISITORS – ON HOLIDAYS & WEEKENDS

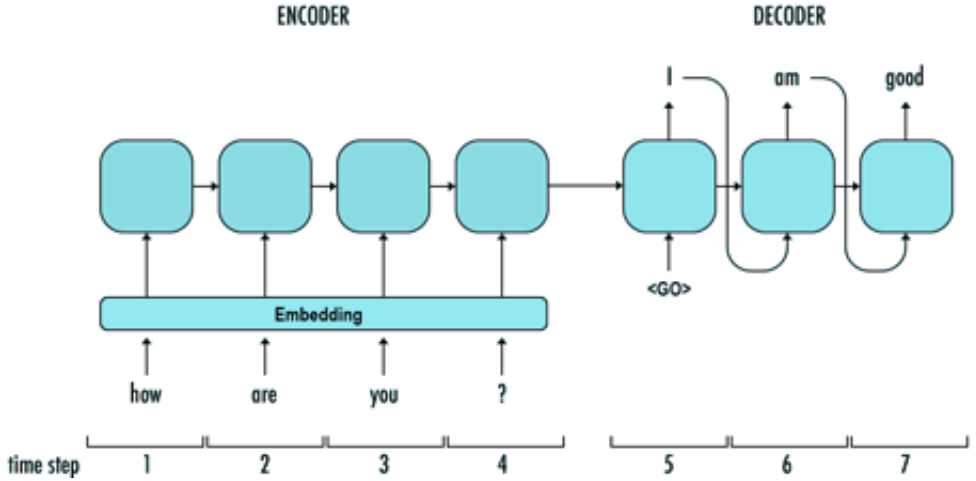


# MODEL DEFINITION

- CLASSICAL ML MODEL
- Gradient Booster Regressor



- DEEP LEARNING MODEL
- Seq2Seq LSTM Encode-Decoder



# FEATURE ENGINEERING

- Features Derived:
  - From visit date: year, month, date, day, weekend, holiday
  - From area: City, ward, Street
  - From reservation: reservation days, visitors reserved
  - From visitors: statistical features – min, max. mean, median, std
- Features Transformed:
  - Categorical : Label encoder
  - Numeric: Min Max Scaler
- Previous 7 days visitors (only for LSTM)

# MODEL EVALUATION

- Data split 80:10:10 train-test-validation split
- Root mean squared logarithmic error as metric

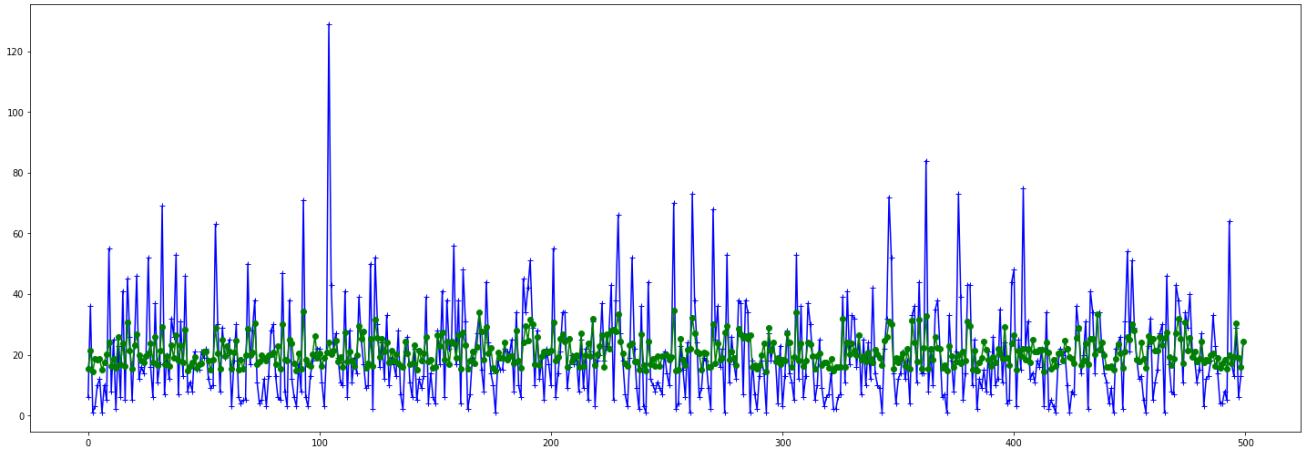
$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\log(1+p) - \log(1+a))^2}$$

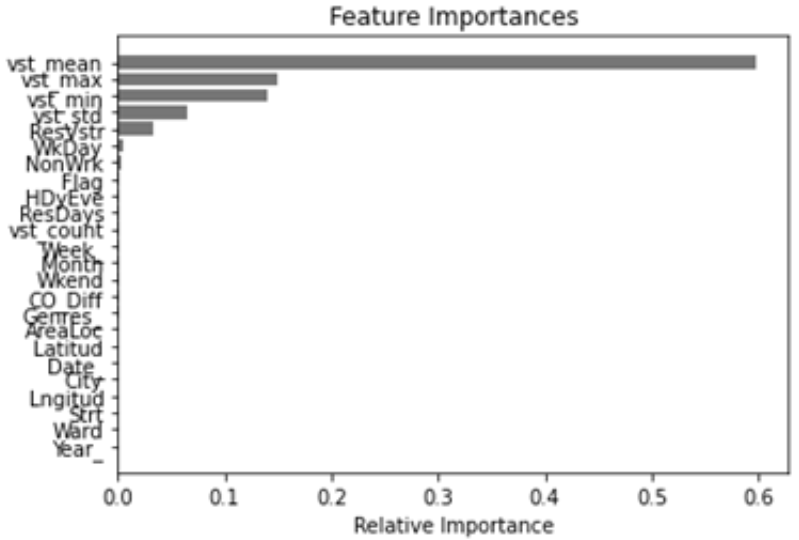
$n$ is the total number of observations | $p_i$ is your prediction of target | $a_i$ is the actual target for $i$.

  - Robustness to the effect of the outliers
  - Measurement of relative error
  - Biased penalty for overestimation

# GRADIENT BOOST MODEL

- K-fold cross-validation and training
- Prediction is averaged over 5 folds
- Hyper Parameter tuning performed



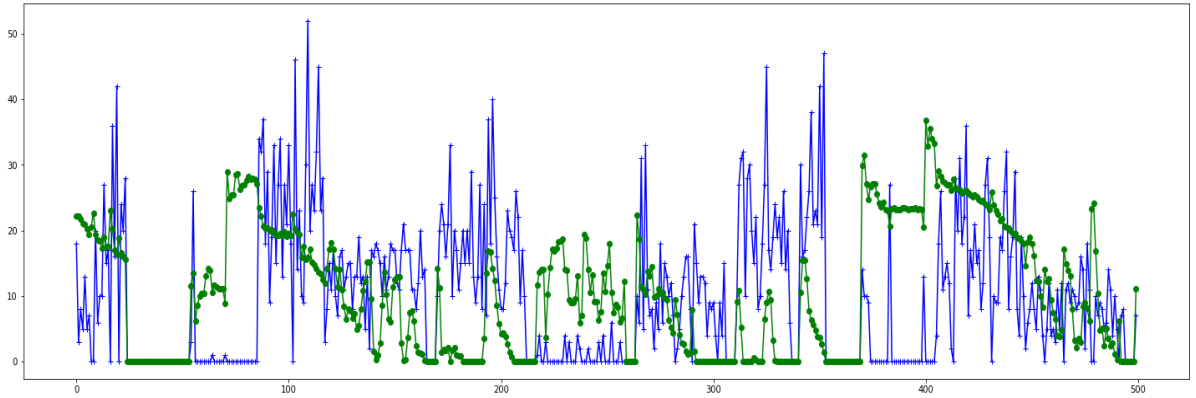

```
--------------------------
Performance of base model:
0.71
--------------------------
Performance of fine model:
0.72
--------------------------
```

# LSTM ENCODE-DECODER

- One layer of encoder
- Two layers of decoder units
- Two iterations with different features

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| input_5 (InputLayer) | [(None, None, 32)] | 0 | [] |
| input_6 (InputLayer) | [(None, None, 32)] | 0 | [] |
| lstm_3 (LSTM) | [(None, 64), (None, 64), (None, 64)] | 24832 | ['input_5[0][0]'] |
| lstm_4 (LSTM) | (None, None, 64) | 24832 | ['input_6[0][0]', 'lstm_3[0][1]', 'lstm_3[0][2]'] |
| lstm_5 (LSTM) | [(None, None, 64), (None, 64), (None, 64)] | 33024 | ['lstm_4[0][0]'] |
| time_distributed_1 (TimeDistributed) | (None, None, 1) | 65 | ['lstm_5[0][0]'] |

Total params: 82,753
Trainable params: 82,753
Non-trainable params: 0




```
------------------------------
Performance of first model:
1.61
------------------------------
Performance of second model:
2.09
------------------------------
```

# SUMMARY

- GBM works better
- Further tasks: Tuning LSTM for better performance
  - Activation Function
  - Number of layers
  - Number of hidden units in each layer
  - Optimizer
- Links below:
  - Architectural decision document :

Recruite_Restaurants_Visitors_Forecasting_ADD_Document.pdf

  - Entity relationship diagram:

Database Documentation.pdf

  - Jupyter Notebook:

IBM_Capstone.ipynb

| Algorithm | Variation | RSMLE | Visual |
|---|---|---|---|
| Gradient Boost | Before tuning | 0.7174 | -- |
| | After tuning | **0.7204** | **OK** |
| Encoder-Decoder | With 3 prev days | 1.6358 | --- |
| | With 7 prev days | 2.0922 | --- |