COME VISIT AGAIN

Terminal Stack

Abhishek Tandon

(MT2021005)

Chandrakant Sahu

(MT2021034)

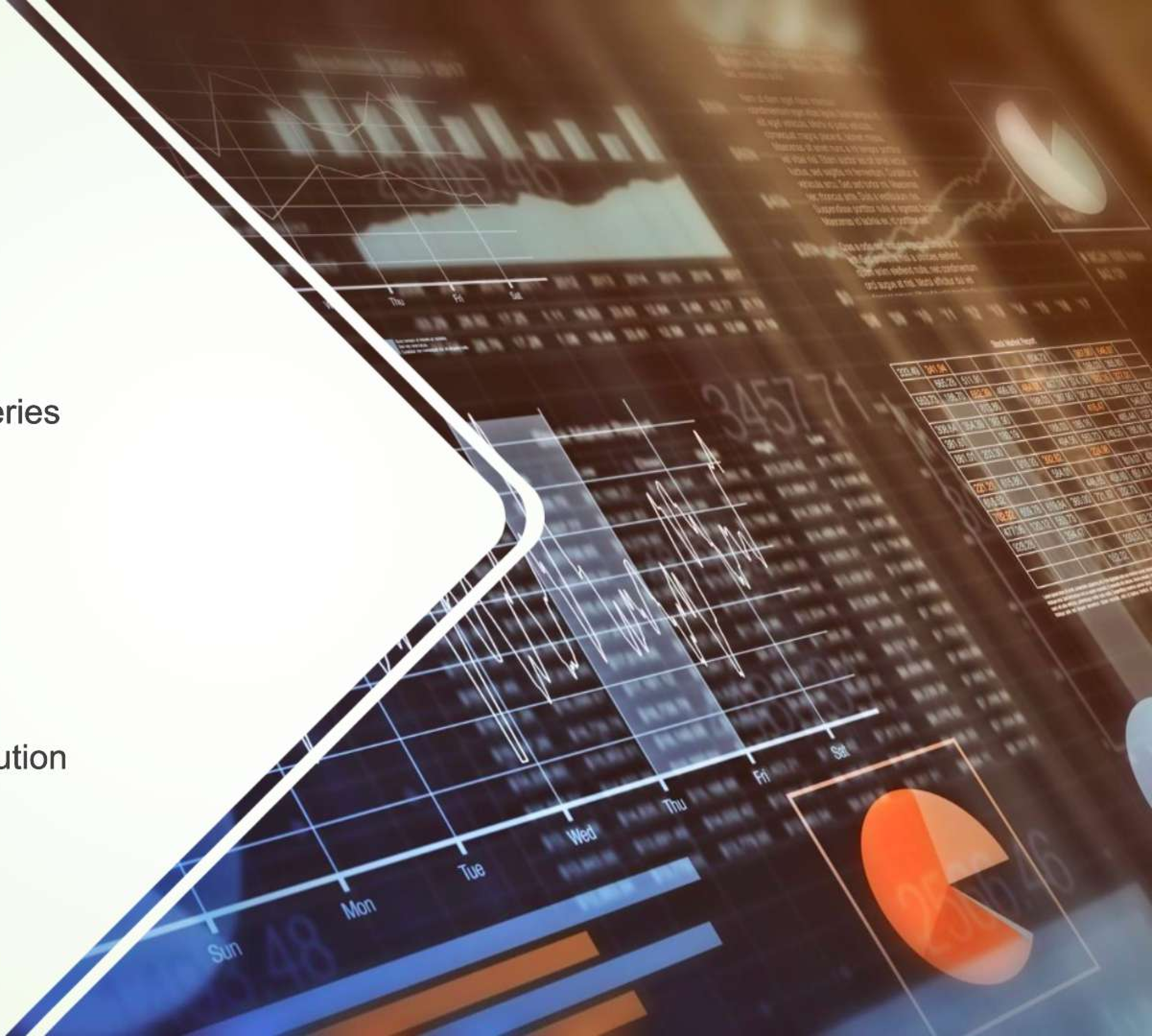
INDEX

INTRODUCTION AND PROBLEM STATEMENT

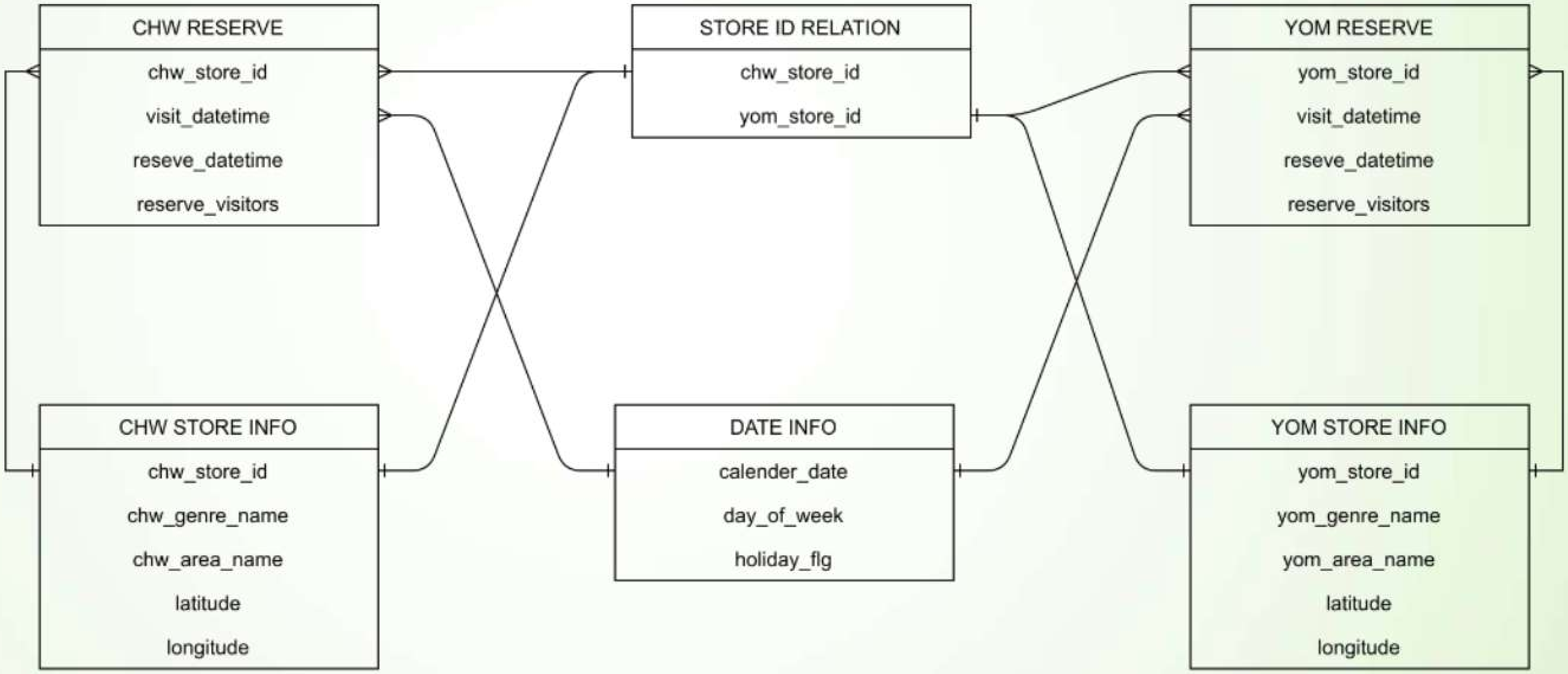
Recruit holdings owns Hot Pepper Gourmet (a restaurant review service), ChwREGI (a restaurant point of sales service), and Restaurant Board (reservation log management software). We were challenged to use reservation and visitation data to predict the total number of visitors to a restaurant for future dates. This information would help restaurants be more efficient in resource management and it will allow them to create much more effective dining experience for their customers.

Exploratory Data Analysis

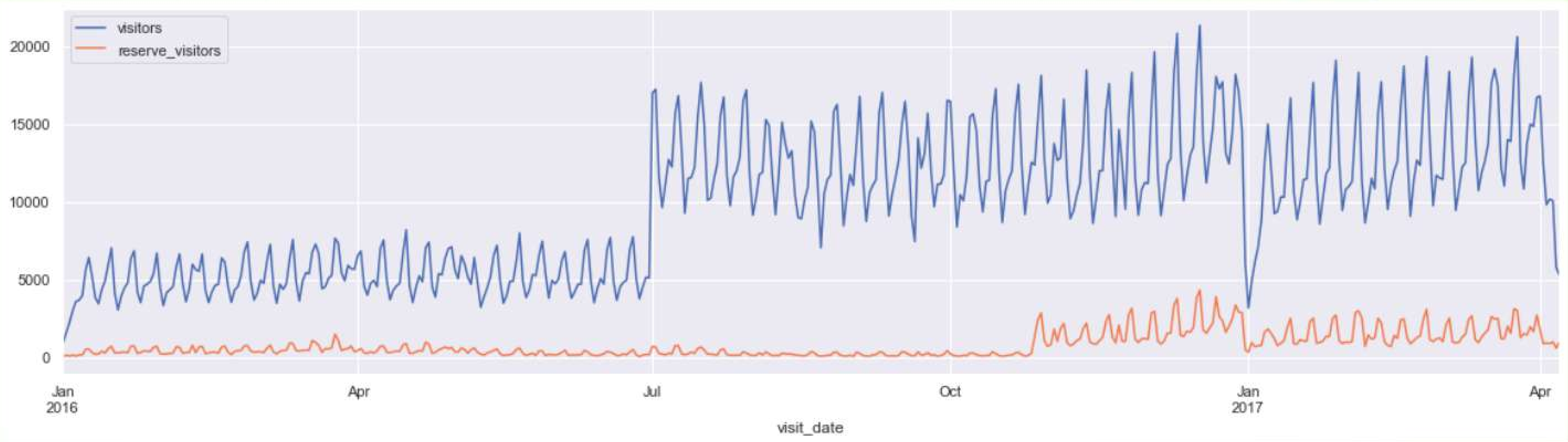
- How metadata is linked ?
- Total visitors and reservation timeseries analysis
- Yearly reservation activity analysis
- Overall Reservations Insights
- Weekly Activity analysis
- Activity based on day of week
- Hourly Visit and Reservation distribution
- And many more...



How metadata is linked?

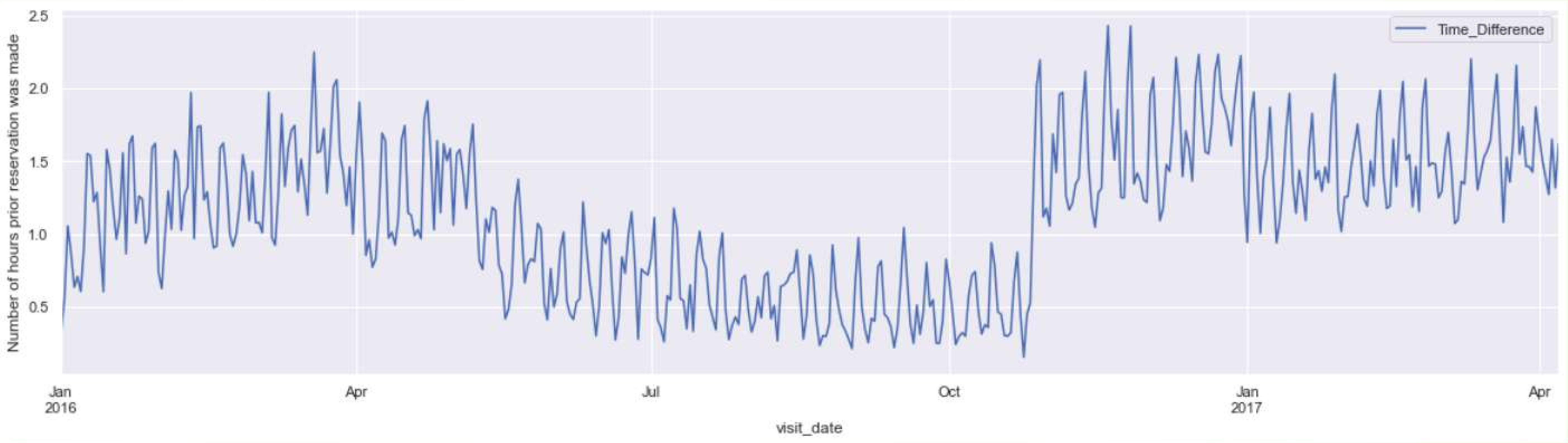


Total visitors and reservation timeseries analysis



- Most of the restaurants have no prior reservations.
- Starting from November 2016, number of reservations started to grow. Which implies that new restaurants were onboarded to take reservations in advance.
- The dip during new year is caused because people prefer to spend new year with their families at home. First week of January sees less visitors.
- The reason of sudden increase in number of visitor in July 2016 is because many new restaurants were
- added in the database.

Yearly reservation activity



- Higher number of reservations are seen after November 2016 and hence we can clearly see that people need to book the restaurants much earlier than previously.
- People tend to reserve less during July to October. This can be a seasonal thing.

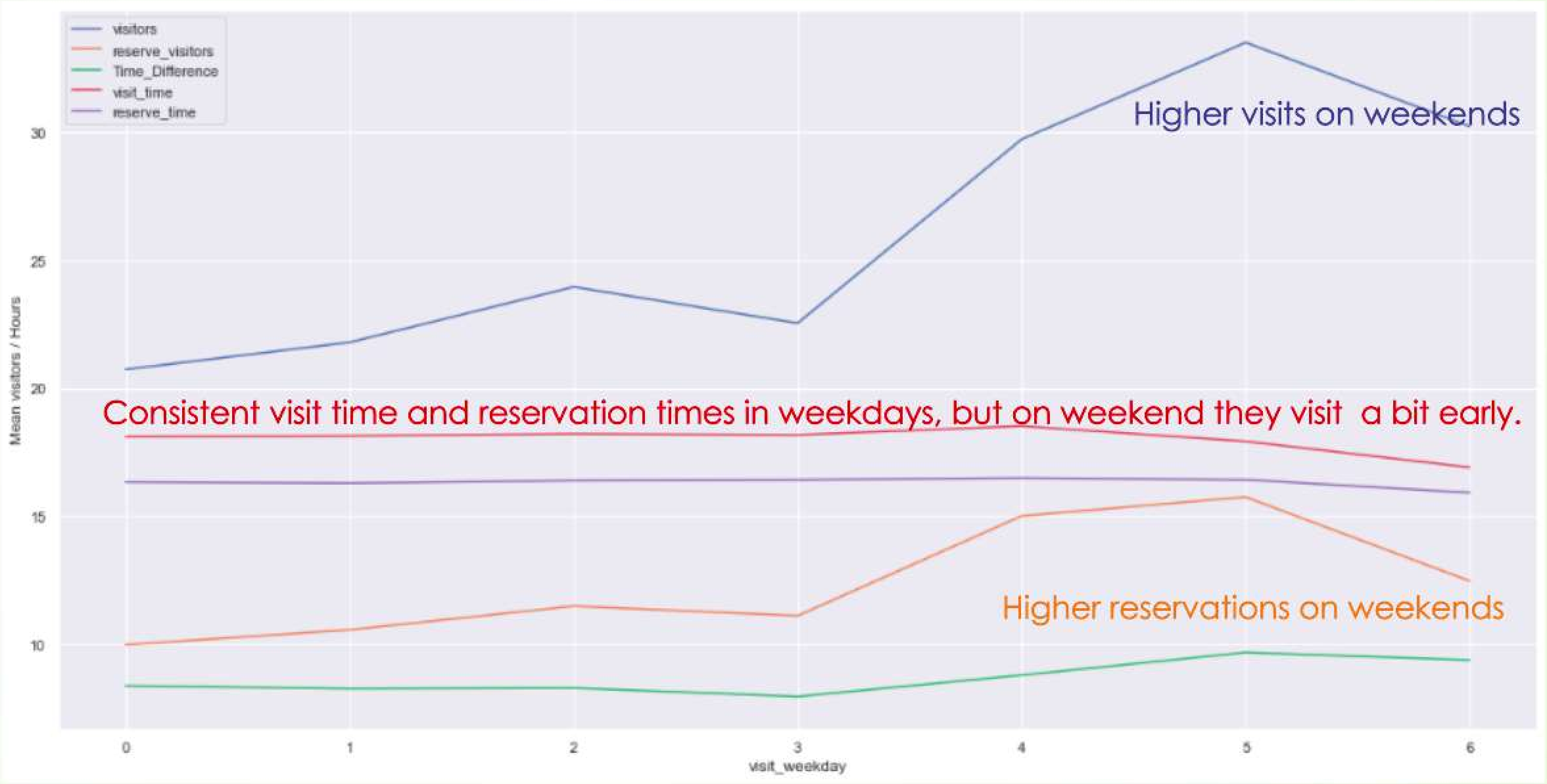
Overall Reservations Insights

- For most of the restaurants we do not have reservations data. Hence most of the entries are 0
- Which means many restaurant do not provide reservation facility.
- After November, prior reservations were made even 24 hours before in many restaurants.

Total Visitors per Restaurant per Day

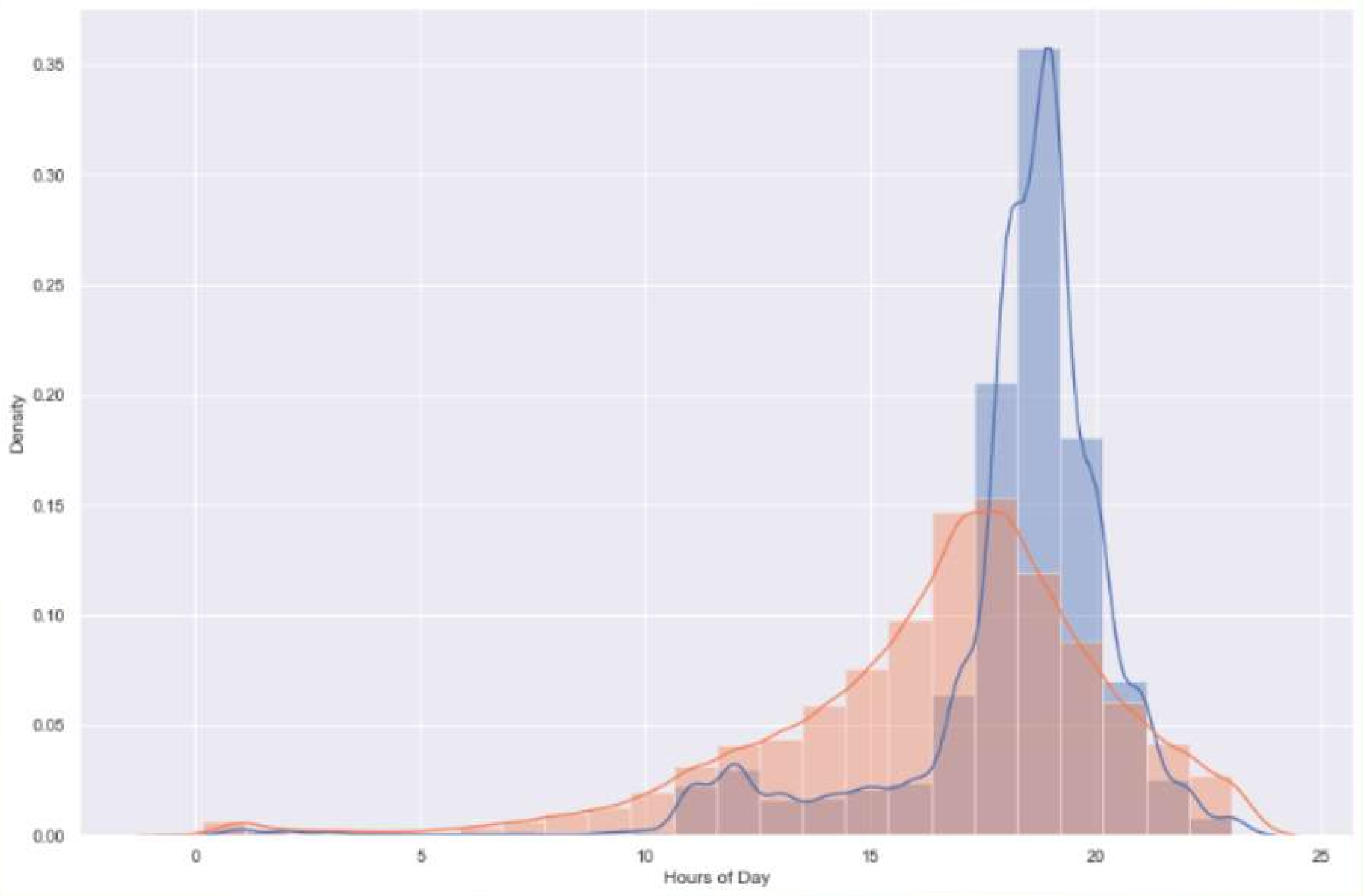
- We see the outliers here, where in a single day restaurants have more than 400 visitors.
- We can filter out outliers if needed and focus more on restaurants that have less than 200 visitors.

Activity based on day of week



Hourly Visit and Reservation distribution

- People tend to visit the restaurants in the evening between 6 PM to 8 PM
- Most reservations are made between 3 PM and 7 PM
- Hence most people tend to have dinner rather than lunch.



And many more...

City wise analysis

Food Genre wise analysis

Year Wise analysis

Holiday Wise analysis

We performed all the above analysis then we move towards feature engineering

FEATURE ENGINEERING

Final set of features :

#	Column	Non-Null Count	Dtype
0	chw_genre_name	215705 non-null	int64
1	latitude	215705 non-null	float64
2	visit_year	215705 non-null	int64
3	visit_month	215705 non-null	int64
4	visit_weekday	215705 non-null	int64
5	city	215705 non-null	int64
6	ward	215705 non-null	int64
7	neighborhood	215705 non-null	int64
8	holiday_flg_0	215705 non-null	uint8
9	holiday_flg_1	215705 non-null	uint8
10	mean_visitors	215705 non-null	float64
11	median_visitors	215705 non-null	float64
12	min_visitors	215705 non-null	float64
13	max_visitors	215705 non-null	float64

Models Used

- Simple Linear Regression
- KNeighbors Regression
- Random Forest Regression

Model 1: Linear Regression

```
#Trying simple Linear Regression model

from sklearn.linear_model import LinearRegression
lr_model = LinearRegression(normalize=True)
lr_model.fit(X_train, y_train)
y_preds=lr_model.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5363602058458732
```

Model 2: KNeighbors Regression

```
#Trying KNeighbors Regression model

from sklearn.neighbors import KNeighborsRegressor
knr_model = KNeighborsRegressor(n_jobs=-1, n_neighbors=5)
knr_model.fit(X_train, y_train)
y_preds=knr_model.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5547749615379268
```

Model 3: Random Forest Regression

```
#Trying Random Forest Regressor Regression model

from sklearn.ensemble import RandomForestRegressor

rfrmodel = RandomForestRegressor(n_estimators=500, n_jobs=-1,
                                 max_samples=None)

rfrmodel.fit(X_train, y_train)
y_preds=rfrmodel.predict(X_test)
rmsle(y_test, y_preds)
```

```
0.5649699218265354
```

Table of models and their scores

Models	Score
Simple Linear Regression	0.54540
KNeighbors Regression	0.53479
Random Forest Regression	0.52696

Individual contributions

- There is nothing like "Individual contributions" , we both work collectively.

CONCLUSION

With the given dataset we think the result is satisfactory. We could have included external data set such as weather data for better prediction. Also, we could have used Time series forecasting methods such as ARIMA for better prediction. However, from curriculum perspective and to learn various regression methods, we preferred Standard regression models that were available in Scikit learn library.

REFERENCE

- https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html
- https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- https://www.kaggle.com/c/recruit-restaurant-visitorforecasting/overview
- Learning from Data, A short course by Yaser S Abu. Mostafa, MalikMagdon Ismail, Hsuan-Tein Lin.