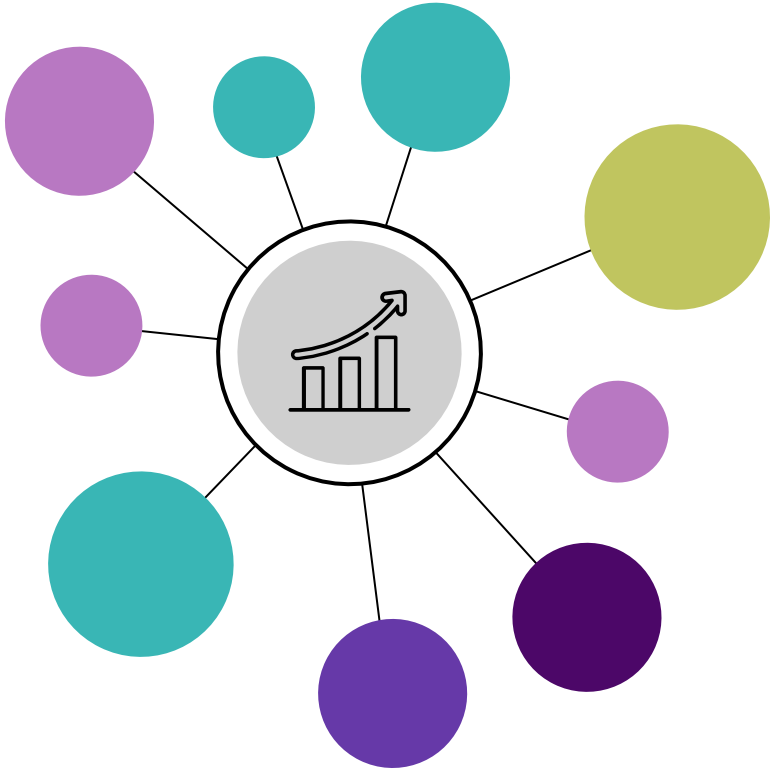


Business Growth Strategy for Hosts on Airbnb Platform

PURDUE UNIVERSITY

Meet the Team

Priyam Sarkar

Samridhi Vats

Priyanka Shinde

Nagraj Deshmukh

PURDUE UNIVERSITY

Executive Summary of Findings

Airbnb's business model embodies a quintessential platform-based structure, capitalizing on the network effect where the value of its service increases with each additional user, aligning with Metcalfe's Law. Our comprehensive analysis bolstered by predictive and prescriptive modelling provided strategic insights leading to value extraction for hosts of Airbnb platform. We undertook an exhaustive study of the potential new entrants and existing hosts to understand the competitive landscape better.

Superhost Program & Revenue Impact: Data from the Superhost Certification Program highlighted a stark contrast in earnings, with Superhosts significantly outperforming regular hosts. The rigorous statistical tests underscore the program's influence on revenue, customer satisfaction, and repeat bookings.

City Selection & Market Entry: Focused analysis on city and tract selection criteria revealed key indicators that guide potential new entrants in identifying lucrative city and tracts for market entry. The study demonstrated that certain tracts within chosen cities present substantial revenue-generating potential.

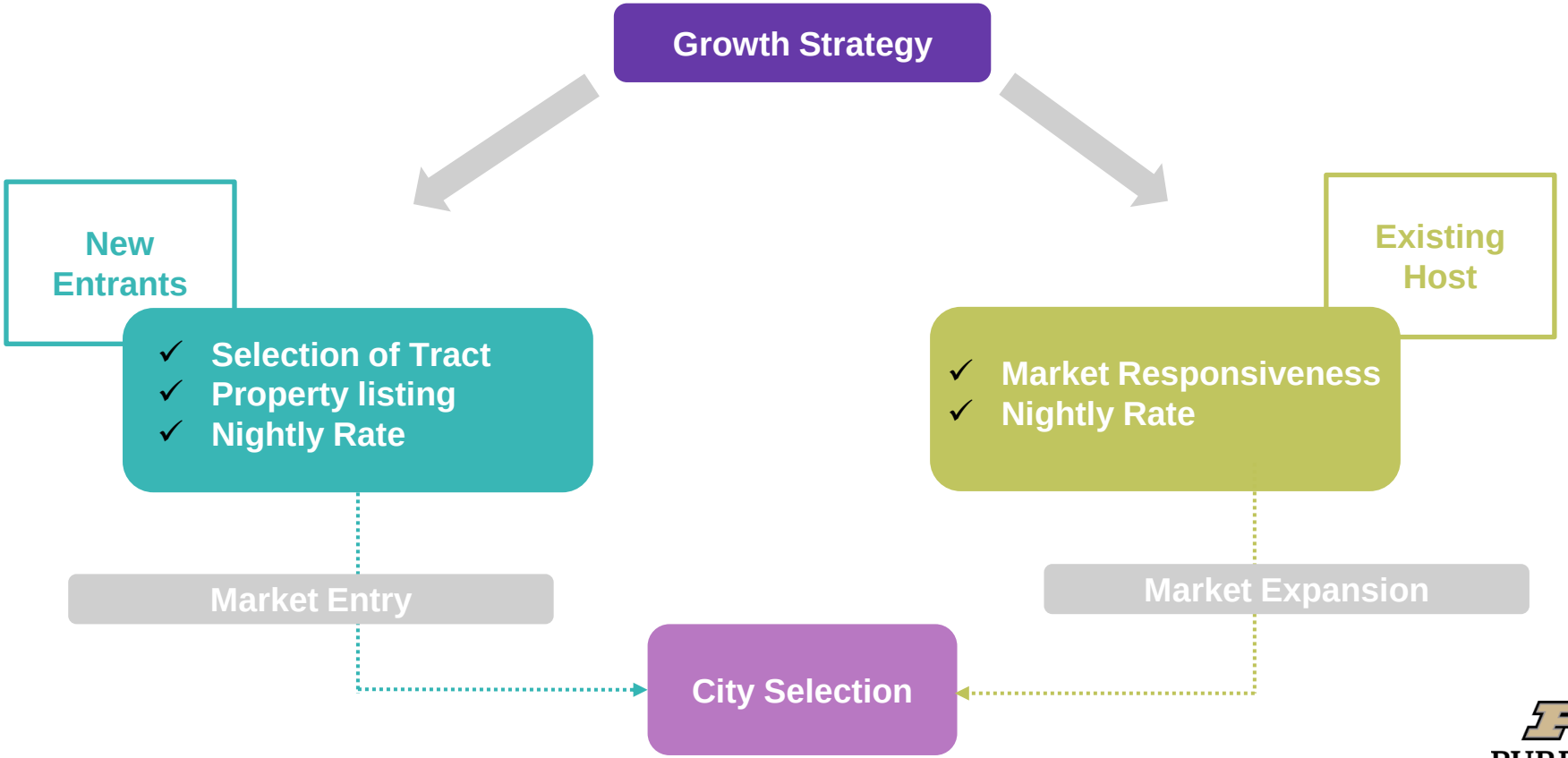
Market Expansion Strategy: For existing hosts, the model pinpoints factors in hosts' control that affect revenue on basis of seasonality demand. These controllable factors, along with responsive strategies to seasonality and market demands, are pivotal in driving revenue growth.

Challenges and Future Scope: Our project navigated through data inconsistencies and missing values to ensure robust model performance. The final recommendation underscores the vital role of data-driven strategies in maximizing value extraction for hosts of Airbnb platform to ultimately boost revenue growth. Due to time constraint, we have analyzed value extraction for hosts but for future scope, we can incorporate value creation for guests as well.

By integrating these findings, hosts can leverage key features for differentiation and competitive advantage, ultimately leading to sustained growth and value extraction on Airbnb platform.

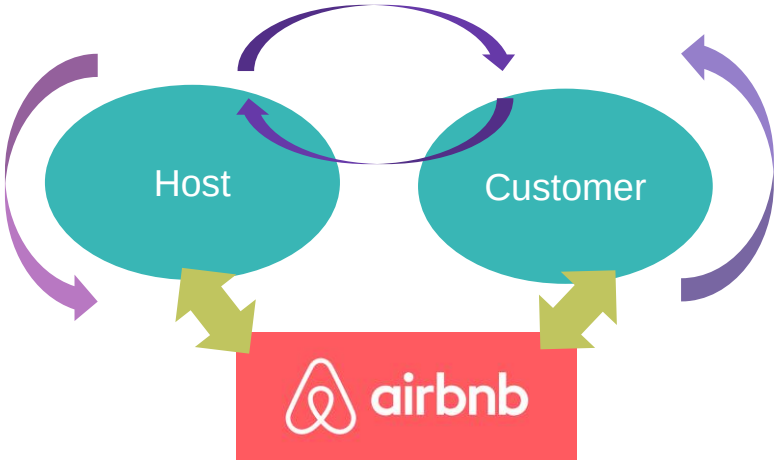
PURDUE UNIVERSITY

Problem Statement

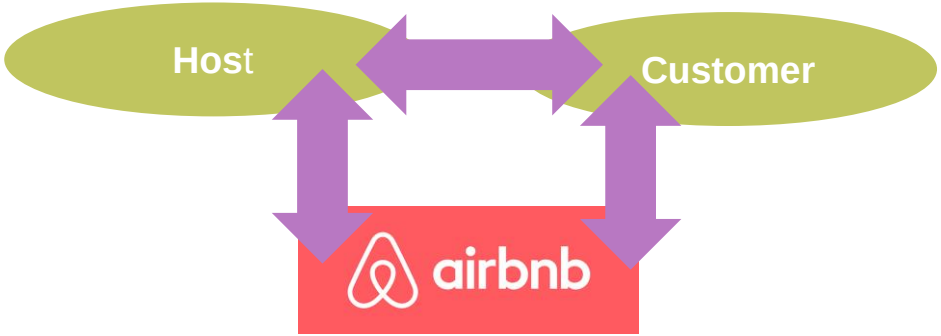


PURDUE UNIVERSITY

Network Effect of Airbnb



Multi layered competition for Host in Airbnb platform



Correlation Matrix:

	Number of Hosts	Number of Superhosts	Rating Overall
Number of Hosts	1.000000	-0.109705	-0.896112
Number of Superhosts	-0.109705	1.000000	0.264264
Rating Overall	-0.896112	0.264264	1.000000
numReserv_pastYear	-0.868152	-0.063267	0.734875

With intense competition, **value creation for host is challenging.**

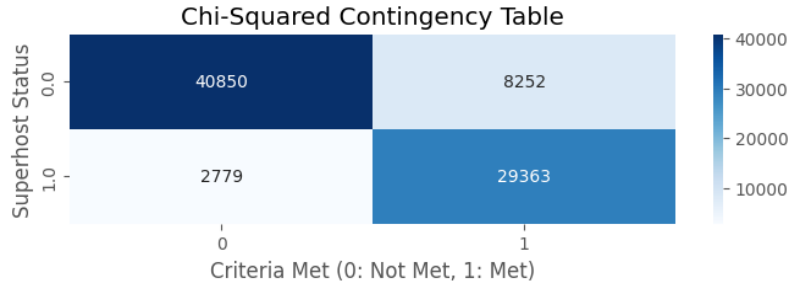
Because of the Project Limitation, we will analyze one side of Network effect and we have chosen **Host**.

However, **data driven business recommendation and growth strategy** can solve the problem.

PURDUE UNIVERSITY

Super Host Certification Program

```
# Criteria for Superhost
min_trips = 10
min_response_rate = 90
max_cancellations = 0
min_5_star_prop = 80
```

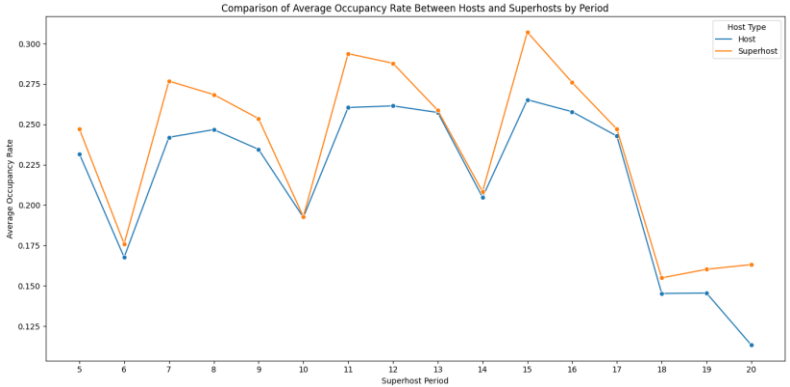


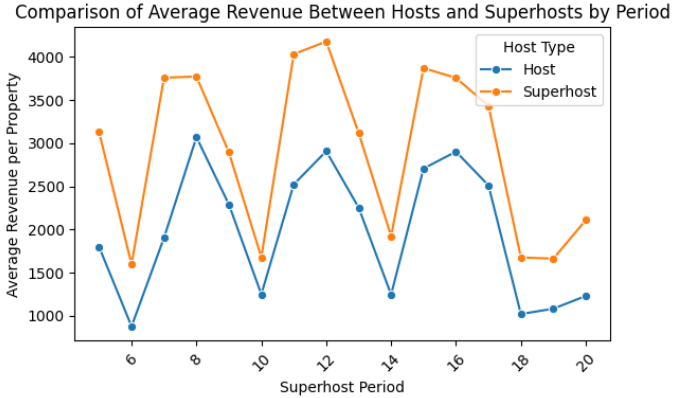

```
Revenue
t-test :  12.39, p : 0.0000

Customer Satisfaction (average rating)
t-test :  127.10, p : 0.0000

Repeat bookings
t-test :  -37.58, p: 0.0000
```

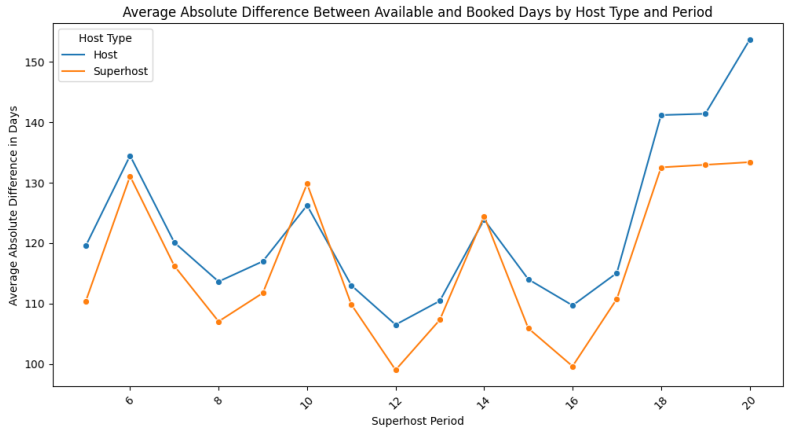
1. **Superhosts, on average, generate higher revenue than non-Superhosts.**
2. **Superhosts have higher customer satisfaction ratings.**
3. **Non-Superhosts have a higher number of reservations on average compared to Super hosts.**

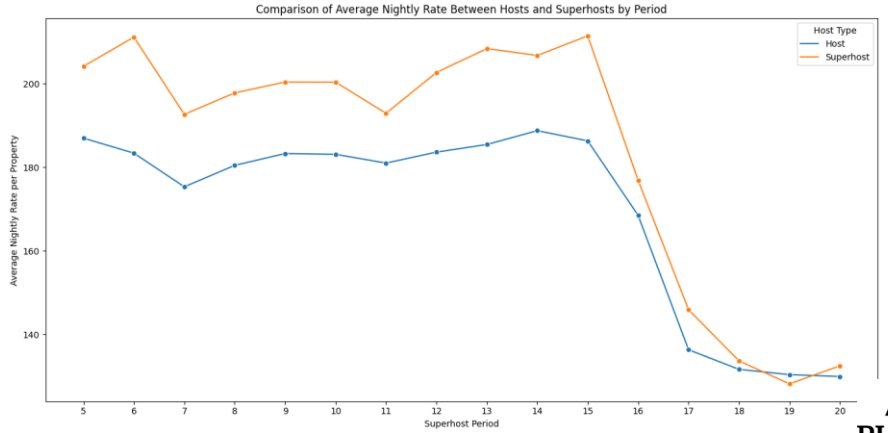
PURDUE UNIVERSITY

Super Value Creation!



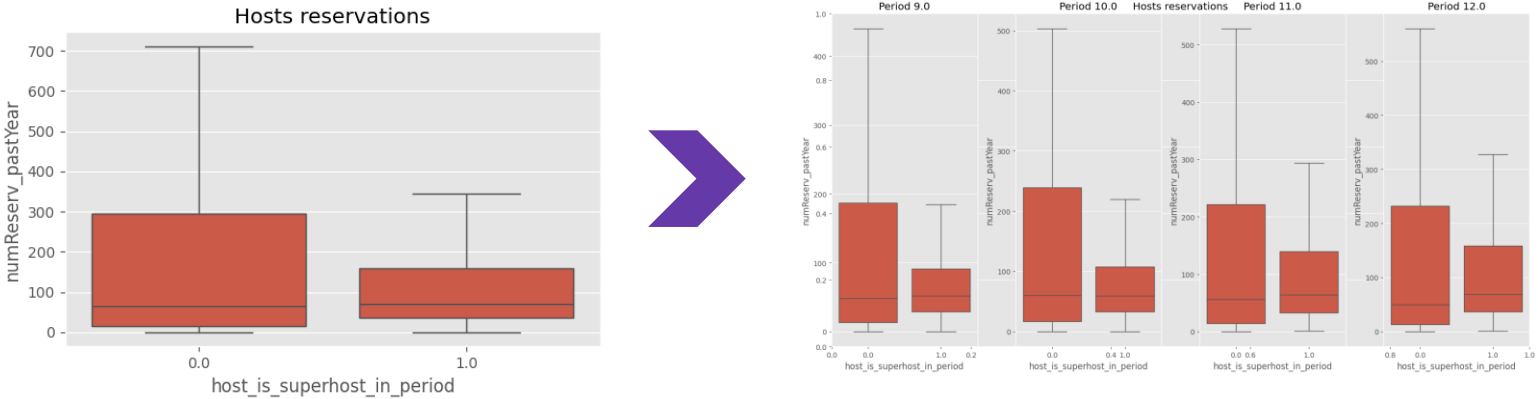






PURDUE UNIVERSITY

Superhost Status Boosts New Entrants' Business



1. Easier for new entrants to achieve superhost status initially.
2. Challenges in retaining superhost status for more than 3 evaluation periods.

Superhost status might provide an initial reservation boost but leads to increased cancellations, potentially resulting in status loss within a few periods.

Avg. consecutive period for which a host retains its superhost status ~ 3

PURDUE UNIVERSITY

Market Entry Strategy of New host

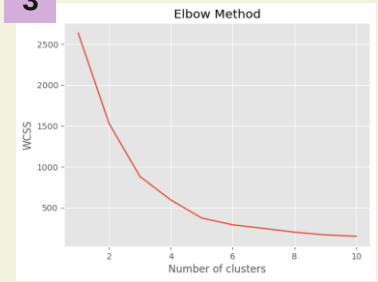
1

```
# List of new variables for clustering
new_vars_for_clustering = [
    'booking_rate_per_host',
    'revenue_per_host',
    'superhosts_per_host',
    'reservations_per_host',
    'cancellations_per_host'
]
```

2

```
# Plot the elbow graph
plt.plot(range(1, 11), wcss)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

3



4

Cluster centers (in the original feature space):

	booking_rate_per_host	revenue_per_host	superhosts_per_host	\
0	1.028510	0.432229	38.175886	
1	100.000000	100.000000	100.000000	
2	46.662498	12.742170	92.132505	

	reservations_per_host	cancellations_per_host
0	0.028495	0.512315
1	100.000000	13.333333
2	0.238468	30.740741

5

```
Cluster 0:
[17031010100, 17031010201, 17031010202, 1703
Cluster 1:
[17031300500]
Cluster 2:
[17031020302, 17031030104, 17031140500, 1703
```

← Selected tract

For a chosen city, which tract(s) to venture into?

1. Tract Analysis & Feature Selection

(Per total hosts)

2. Tract Clustering
3. Optimal no. of Cluster Determination
4. Competitive Landscape in Clustered Tracts
5. Final Tract Selection

PURDUE UNIVERSITY

Revenue Drivers in the “Chosen Tract”

Analyzing Competitive environment within High potential Tracts

Finalized Tract in Chicago

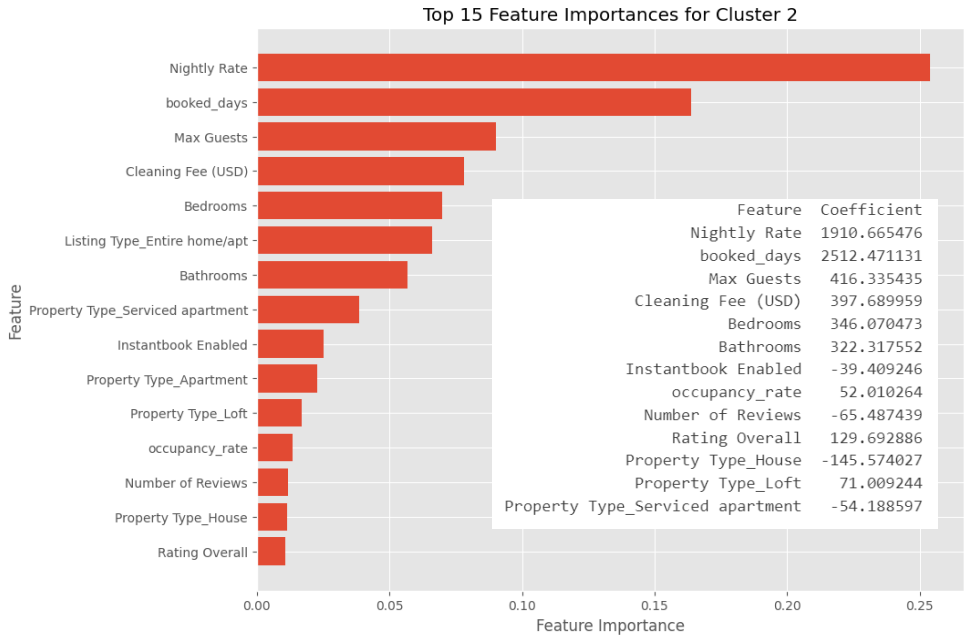
```
##Cluster Mapping
tracts_cluster_2 = [17031300500]
```

XGBoost Model

```
# Define the Gradient Boosting model
model = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('regressor', xgb.XGBRegressor(objective='reg:squarederror',
        n_estimators=100, learning_rate=0.1, max_depth=5))
])
```

Predictive analysis to help hosts understand key features for differentiation and competitive positioning

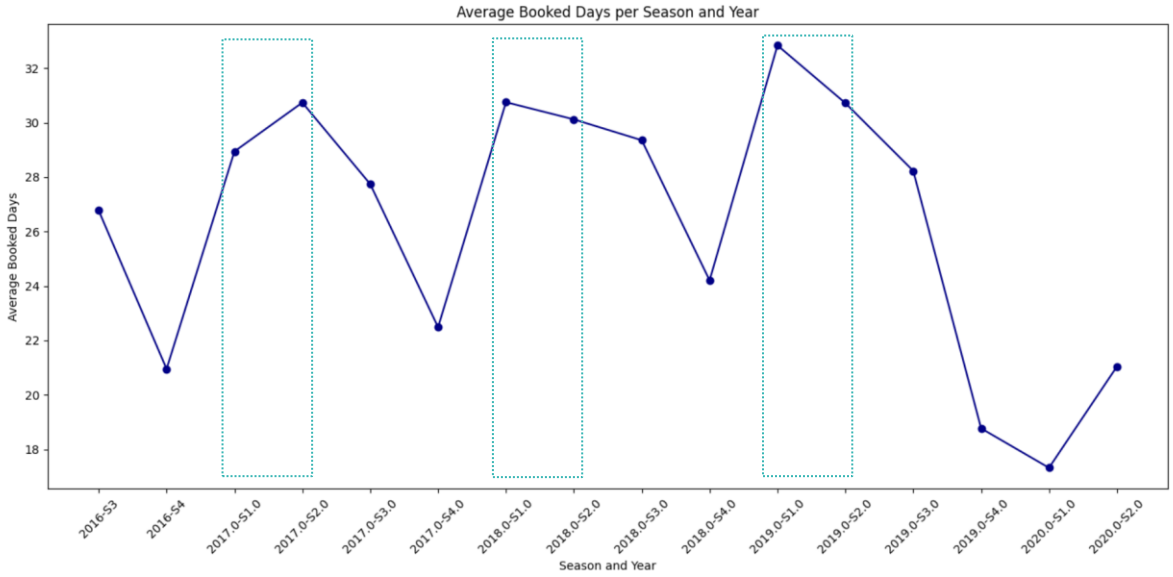
Important Feature for Tract 2

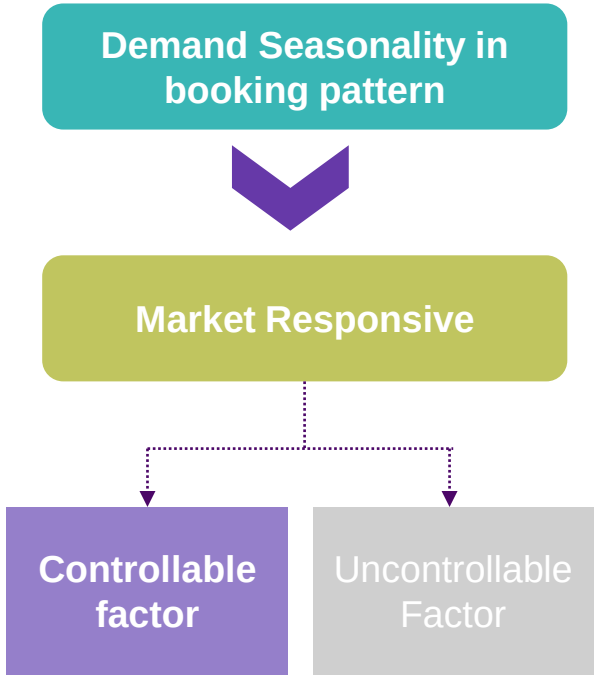



```
Root Mean Squared Error: 1802.19
R-squared: 0.84
Mean Absolute Percentage Error (MAPE): 34.15%
```

PURDUE UNIVERSITY

Market Responsiveness
Assessing Demand Seasonality Needs





PURDUE UNIVERSITY

Impact of Controllable Factors and Seasonality on Revenue

OLS Regression Results

Dep. Variable:	revenue	R-squared:	0.612
Model:	OLS	Adj. R-squared:	0.612
Method:	Least Squares	F-statistic:	3062.
Date:	Fri, 08 Dec 2023	Prob (F-statistic):	0.00
Time:	21:19:05	Log-Likelihood:	-3.8232e+05
No. Observations:	44703	AIC:	7.647e+05
Df Residuals:	44679	BIC:	7.649e+05
Df Model:	23		
Covariance Type:	nonrobust		

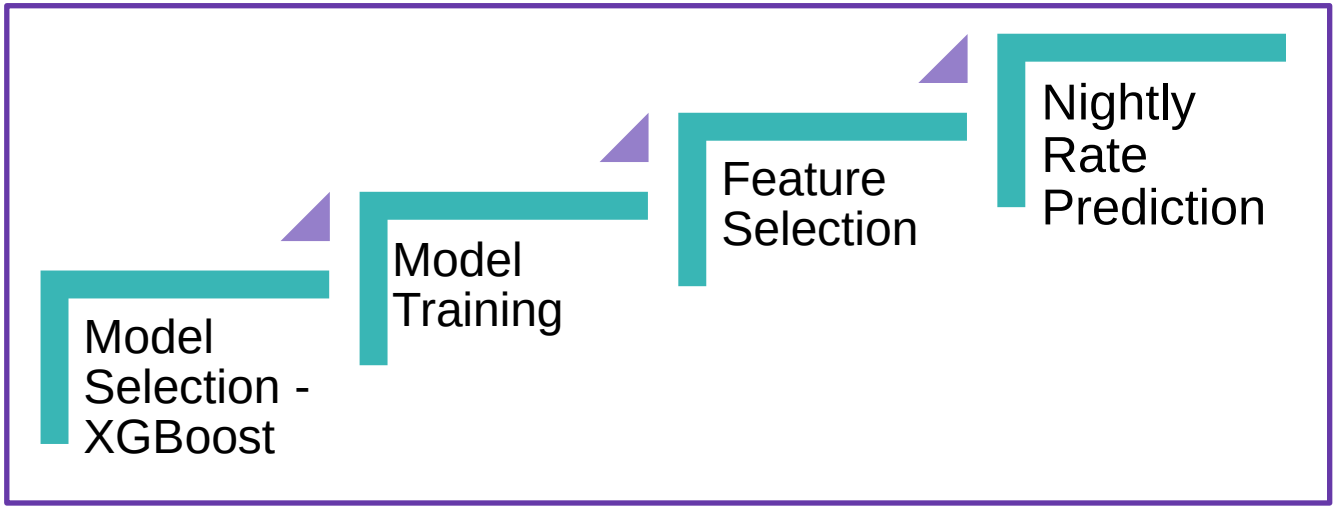
	coef	std err	t	P>|t|
const	-2344.9933	172.678	-13.580	0.000
season_Nightly Rate_interaction	-0.3974	0.063	-6.318	0.000
season_Cleaning Fee (USD)_interaction	-0.3973	0.183	-2.167	0.030
season_Max Guests_interaction	-29.9202	3.106	-9.634	0.000
season_Minimum Stay_interaction	2.5334	0.722	3.507	0.000
season_Pets Allowed_interaction	28.7440	14.093	2.040	0.041
season_Instantbook Enabled_interaction	-33.3185	10.800	-3.085	0.002
season_available_days_interaction	-0.2573	0.083	-3.109	0.002
season_Number of Photos_interaction	-2.1301	0.551	-3.869	0.000
season_hostResponseAverage_pastYear_interaction	1.9046	0.184	10.348	0.000

Controllable Factors

1. Nightly Rate
2. Cleaning Fee (USD)
3. Maximum Guests
4. Minimum Stay
5. Pets Allowed
6. Instant Book Enabled
7. Available Days
8. Number of Photos
9. Response Time

PURDUE UNIVERSITY

Nightly Rate Recommendation for Both Existing Host and New Host



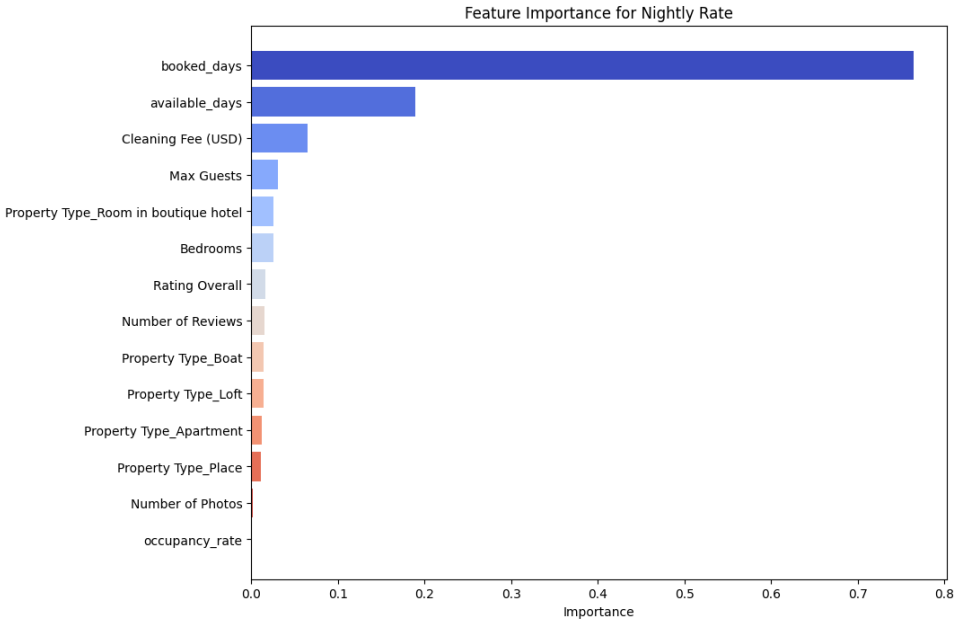
PURDUE UNIVERSITY

Nightly Rate Recommendation for Both Existing Host and New Host

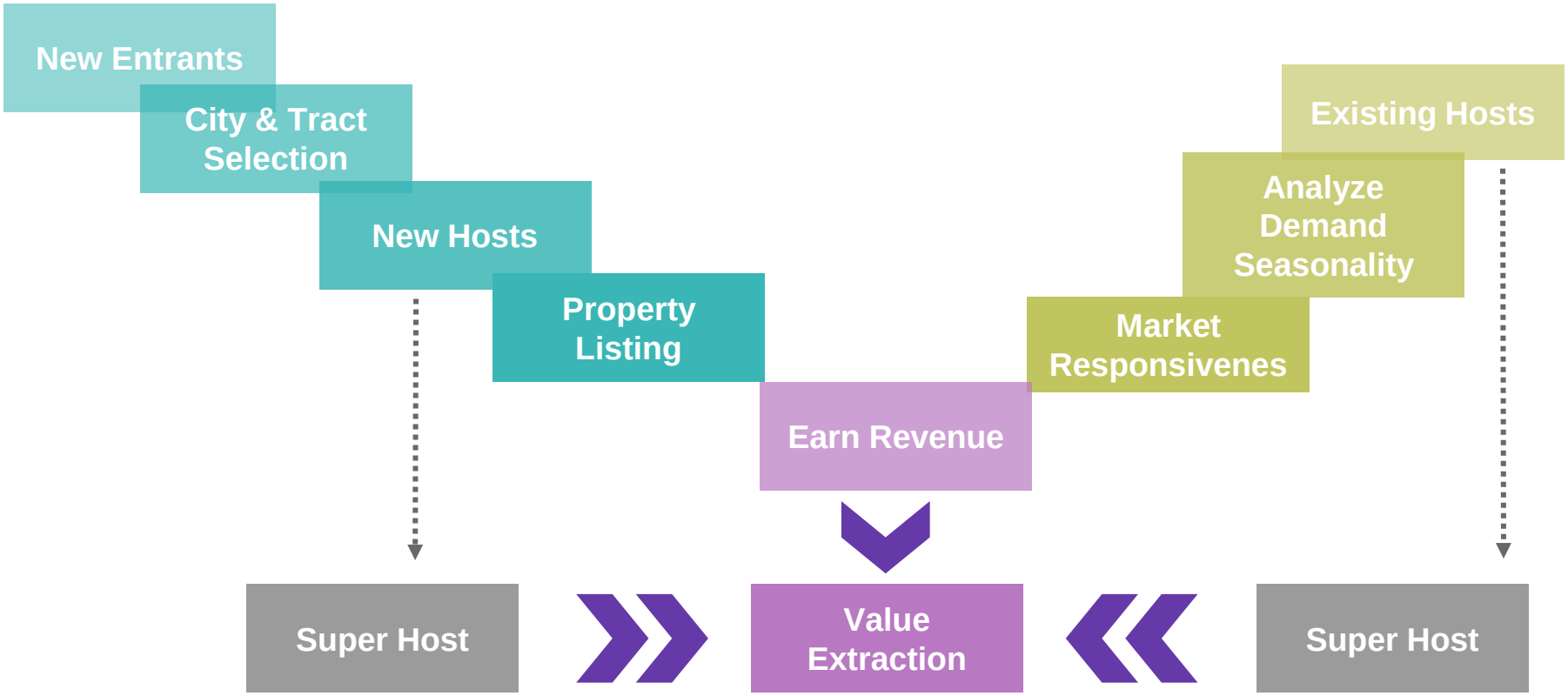
```
Training - RMSE: 95.75188150425886, MAPE: 47.12527571068168%, R-Squared: 0.6168818766374641
Validation - RMSE: 100.09587339162654, MAPE: 47.40330429732502%, R-Squared: 0.5841805267888589
```

```
Input Values to predict Nightly Rate:
Cleaning Fee (USD): [35]
Bathrooms: [3]
Max Guests: [4]
Bedrooms: [3]
Rating Overall: [4]
census_tract: [17031671800]
Number of Reviews: [10]
Number of Photos: [20]
Listing Type_Entire home/apt: [1]
Property Type_Apartment: [1]
Property Type_Loft: [1]
Property Type_Place: [1]
Property Type_Boat: [1]
```

```
Predicted Nightly Rate: 466.2528
```



Conclusion



PURDUE UNIVERSITY

Basis of City Selection and Outcome

Selection Criteria

```
# Combined score per host
city_metrics['combined_score_per_host'] = (
    city_metrics['booking_rate_per_host'] +
    city_metrics['revenue_per_host'] +
    city_metrics['superhosts_per_host'] +
    city_metrics['reservations_per_host'] -
    city_metrics['cancellations_per_host']
)
```

Outcome

	City	combined_score_per_host
7	Oakland	241.477518
1	Chicago	230.376751
2	Dallas	168.390218
9	Washington	75.956511
0	Boston	73.689445
3	Houston	66.607554
8	Philadelphia	65.699838
4	Los Angeles	39.032313
6	New York	21.995084
5	Miami	-17.080418

PURDUE UNIVERSITY

Challenges and Future Scope

- Lots of Missing data in the dataset
- Difficulty in determining between impute and remove
- Categorical variable with too many levels, which causes linear dependency on each other.

- Binary column contains non-binary value
- pre_year_superhost column contains values other than 0 and 1

- Few Columns definition are missing in dataset
- Example : Response Rate is an important parameter for super host classification and missing in data description

Due to the time constraint , we have analyzed value creation for host. For future scope we can incorporate value creation for guest as well.

PURDUE UNIVERSITY

Thank You!

APPENDIX

Being a Super host enhances value creation!

Performed T test and got **there is a significant difference between Host and Super host.**

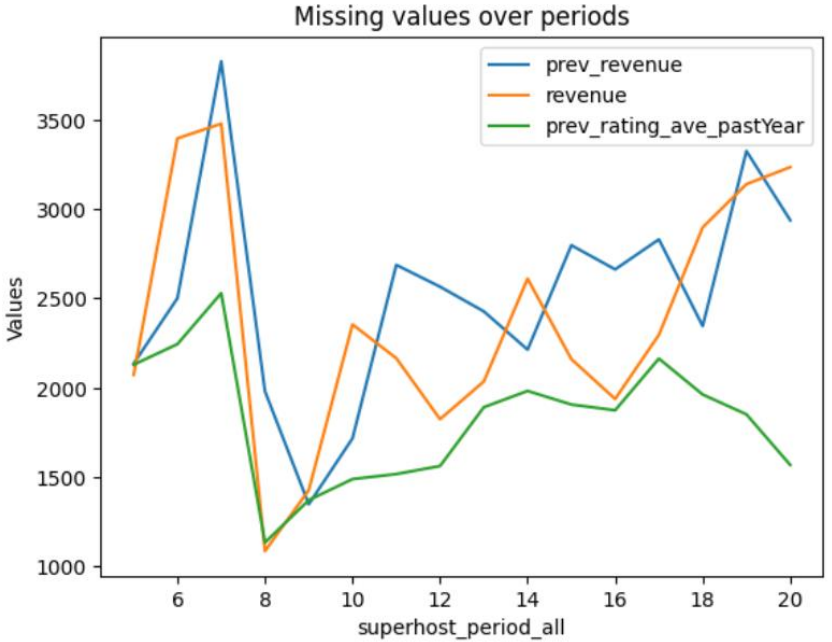
- ✓ Available days
- ✓ Booked days
- ✓ Occupancy rate
- ✓ Nightly rate

```
available_days: t-statistic=-11.001631081022754, p-value=3.926537237891909e-28
booked_days: t-statistic=31.52849585023777, p-value=1.1346973937782663e-216
revenue: t-statistic=29.06908271622486, p-value=1.3675373394434613e-184
occupancy_rate: t-statistic=17.387549648509406, p-value=1.3853068796072642e-67
Nightly Rate: t-statistic=4.146682330442327, p-value=3.377214642270404e-05
```

PURDUE UNIVERSITY

Handling Missing Values

Column Name	Missing Values	Missing Percent
prev_booked_days_avePrice	40294	0.335177
prev_booked_days	40294	0.335177
prev_occupancy_rate	40294	0.335177
prev_revenue	40294	0.335177
revenue	38108	0.316993
booked_days_avePrice	38108	0.316993
booked_days	38108	0.316993
occupancy_rate	38108	0.316993
prev_rating_ave_pastYear	29172	0.242661
prev_prop_5_StarReviews_pastYear	29172	0.242661
prev_numReviews_pastYear	28296	0.235374



We saw a very high number of missing values in our data – upto 34% in some variables

Also the missing values were distributed pretty evenly over the periods

Handling Missing Values

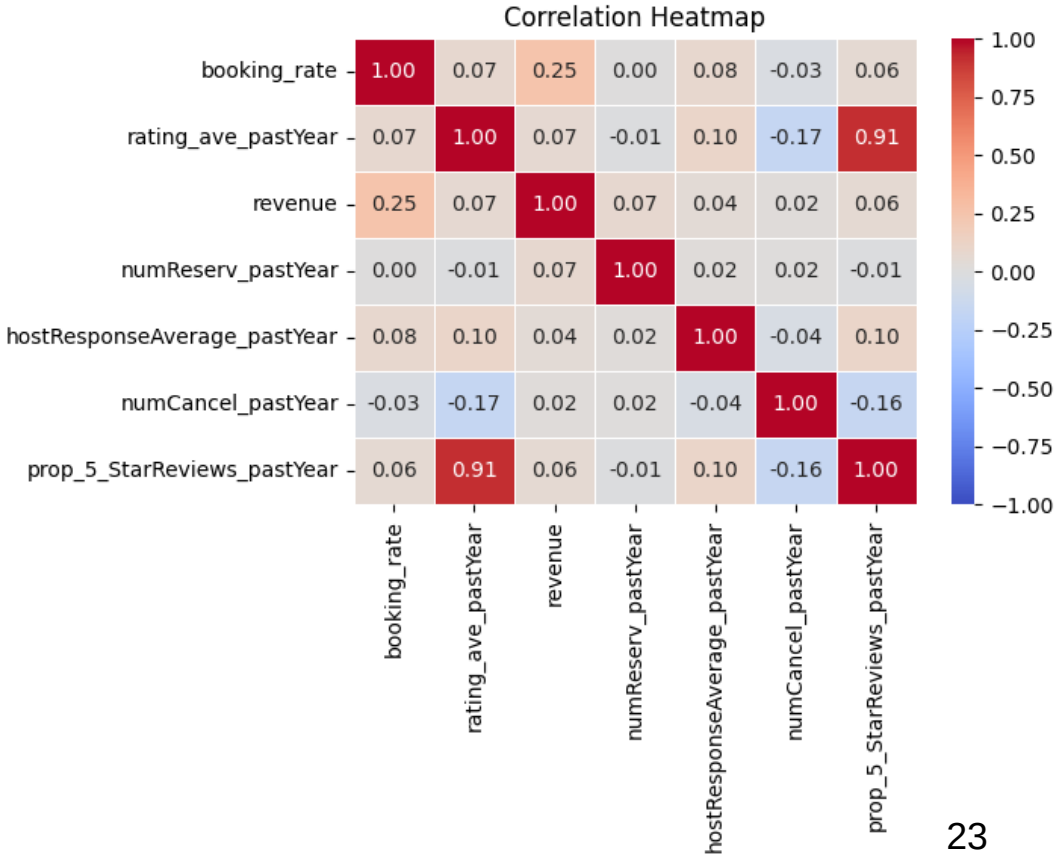
```
# 2. Imputing missing values for other columns

numeric_cols = airbnb_data_cleaned.select_dtypes(include=['float64', 'int64']).columns
categorical_cols = airbnb_data_cleaned.select_dtypes(include=['object']).columns

# Grouping by census tract and imputing numeric columns with the median of their respective tract
for col in numeric_cols:
    airbnb_data_cleaned[col] = airbnb_data_cleaned.groupby('census_tract')[col].transform(lambda x: x.fillna(x.median()))

# Grouping by census tract and imputing categorical columns with the mode of their respective tract
for col in categorical_cols:
    airbnb_data_cleaned[col] = airbnb_data_cleaned.groupby('census_tract')[col].transform(lambda x: x.fillna(x.mode()[0] if
    not x.mode().empty else "Unknown"))
```

We dropped the rows which had missing values for revenue

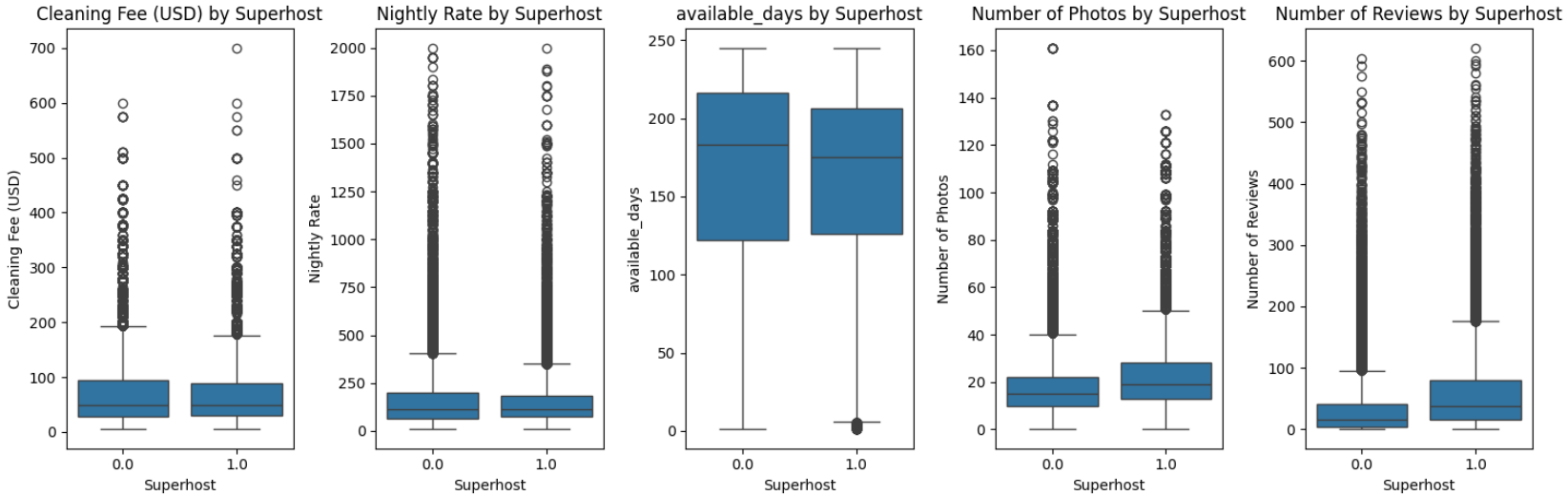
Properties in the same geographical area have similar metric values, hence for other variables we used the tract mean for the numerical variables and the tract median for categorical variables.

Interdependence of variables

1. Negative correlation of reservations with ratings and 5starreviews is surprising.
2. More booking rate results into more revenue, is understandable.
3. Positive correlation between reservations and cancellations suggests that even when the reservations increase, cancellations increase.
4. Avg-rating and 5stars high correlation is obvious
5. More cancellations negatively impact the average rating.



Distribution of different variables based on superhost status



1. Most notable differences are in the number of photos and the number of reviews.
2. Superhosts tend to get higher reviews on average

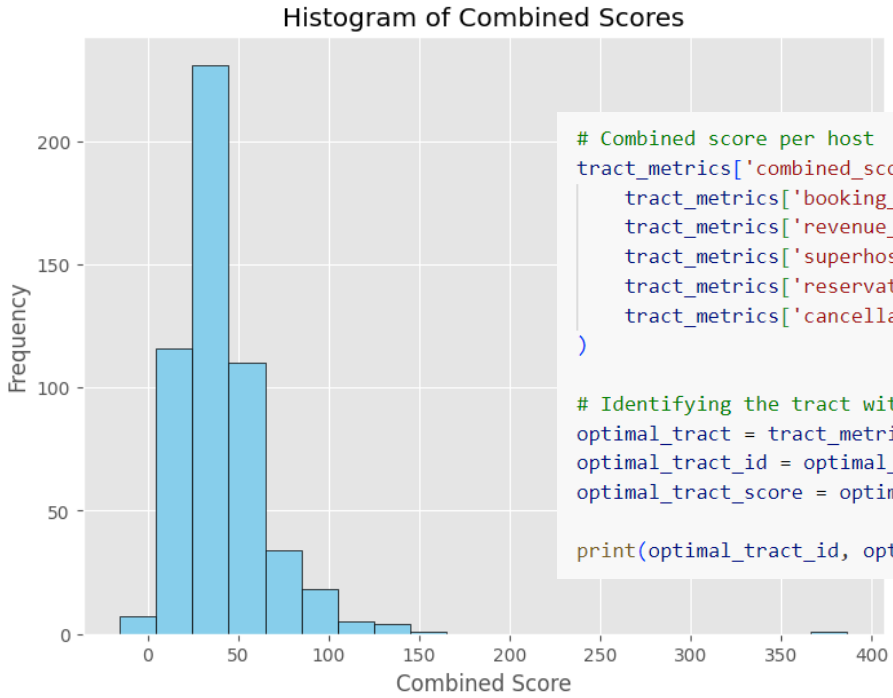
Ability of different variables to predict the superhost status

Logit Regression Results

Dep. Variable:	host_is_superhost_in_period	No. Observations:	56870
Model:	Logit	Df Residuals:	56865
Method:	MLE	Df Model:	4
Date:	Fri, 08 Dec 2023	Pseudo R-squ.:	0.2757
Time:	13:46:14	Log-Likelihood:	-27628.
converged:	True	LL-Null:	-38144.
Covariance Type:	nonrobust	LLR p-value:	0.000

	coef	std err	z	P>|z|	[0.025	0.975]
const	-21.6302	0.356	-60.822	0.000	-22.327	-20.933
numReserv_pastYear	-0.0002	9.41e-06	-21.334	0.000	-0.000	-0.000
hostResponseAverage_pastYear	0.1403	0.003	41.342	0.000	0.134	0.147
numCancel_pastYear	-0.9597	0.026	-36.464	0.000	-1.011	-0.908
prop_5_StarReviews	0.0911	0.001	85.123	0.000	0.089	0.093

Optimal Tract Selection Method




```python
# Combined score per host
tract_metrics['combined_score_per_host'] = (
    tract_metrics['booking_rate_per_host'] +
    tract_metrics['revenue_per_host'] +
    tract_metrics['superhosts_per_host'] +
    tract_metrics['reservations_per_host'] -
    tract_metrics['cancellations_per_host']
)

# Identifying the tract with the highest combined score
optimal_tract = tract_metrics.sort_values(by='combined_score_per_host', ascending=False).iloc[0]
optimal_tract_id = optimal_tract['census_tract']
optimal_tract_score = optimal_tract['combined_score_per_host']

print(optimal_tract_id, optimal_tract_score)
```

Another Tract Selection Method

```
# Applying K-Means clustering to each new variable
# We use 2 clusters: one for high and one for low values for each parameter
kmeans_results = {}
for var in new_vars_for_clustering:
    kmeans = KMeans(n_clusters=2, random_state=0).fit(new_data_normalized[[var]])
    tract_metrics[var + '_cluster'] = kmeans.labels_
    kmeans_results[var] = kmeans

# Identify the high value cluster for each variable
# Assuming that for booking rate, revenue, superhosts, and reservations, the higher cluster is better (higher mean is better)
# For cancellations, the lower cluster is better (lower mean is better)
high_value_clusters = {}
for var in new_vars_for_clustering:
    if var == 'cancellations_per_host':
        # For cancellations, we want the cluster with the lower mean
        high_value_clusters[var] = tract_metrics.groupby(var + '_cluster')[var].mean().idxmin()
    else:
        # For all other variables, we want the cluster with the higher mean
        high_value_clusters[var] = tract_metrics.groupby(var + '_cluster')[var].mean().idxmax()

# Finding the common tracts that fall into the desirable clusters for all variables
optimal_tracts = tract_metrics.copy()
for var, cluster in high_value_clusters.items():
    optimal_tracts = optimal_tracts[optimal_tracts[var + '_cluster'] == cluster]

# Select only the census_tract and the new variables for clustering
optimal_tracts = optimal_tracts[['census_tract'] + new_vars_for_clustering]

optimal_tracts
```