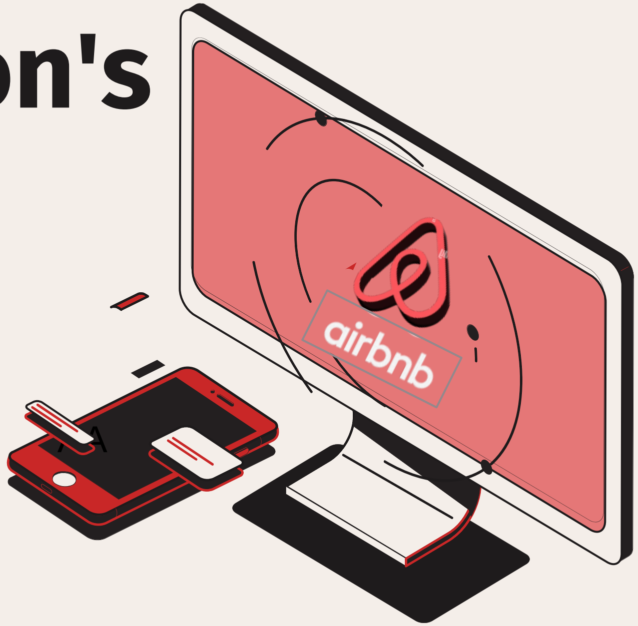
MGMT 683: IT Innov And Comp Advantage

Capitalizing on Houston's AirBnb momentum

Team 22

Pooja Udayanjali Kannuri
Ann Mathew
Archita Ray
Abithaa Shree Venkatesh



Contents

Executive Summary

To propose a strategic roadmap for elevating business performance within the hospitality sector on Airbnb. This involves a concerted effort to bolster **overall ratings** through an unwavering commitment to service excellence, coupled with a tactical approach to augmenting **occupancy rates.** By adeptly managing reservations, cultivating positive reviews, and optimizing pricing strategies, hosts can significantly impact guest satisfaction and maximize revenue potential.

Assumptions

- ✓ Only the properties which generate revenue impact our analysis
- ✓ The analysis currently excludes considerations for seasonality, focusing on general patterns and factors influencing overall ratings.

Business Problem

Airbnb's strategic initiatives aim to focus on two key areas to enhance brand recognition and maintain market competitiveness:

- ✓ **Predicting overall guest ratings to improve satisfaction**

  To predict overall ratings and identify key factors to enhance host satisfaction and boost revenue

- ✓ **Predicting Occupancy Rates for Enhanced Market Competitiveness**

  This predictive model forecasts occupancy rates using key factors like the number of reviews, reservations, and nightly rates to help hosts optimize revenue.

This dual-pronged strategy aims to empower hosts with actionable insights, enhancing service quality and boosting platform competitiveness.

Data Source

airbnb_Houston.csv

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Airbnb_Host_ID	INPUT	62915837	46727844	100000	0	4844	55848197	2.7821E8	0.641465	-0.14997
Airbnb_Property_ID	INPUT	16014533	5553962	100000	0	3816	16687261	28752314	-0.26242	0.449922
Bathrooms	INPUT	1.518815	0.793633	99976	24	0	1	10.5	1.92089	6.741808
Bedrooms	INPUT	1.606364	0.992958	99996	4	0	1	20	2.1094	15.3711
Cleaning_Fee__USD_	INPUT	75.47839	62.60453	65070	34930	5	65	999	2.606023	17.10971
Instantbook_Enabled	INPUT	0.5258	0.499336	100000	0	0	1	1	-0.10334	-1.98936
Latitude	INPUT	29.74286	0.066982	100000	0	29.53411	29.74024	30.03549	1.130828	3.639843
Longitude	INPUT	-95.4262	0.086516	100000	0	-95.7185	-95.4069	-95.0618	-0.33568	2.58371
Max_Guests	INPUT	4.067776	2.516134	99991	9	1	4	16	1.407957	2.825913
Minimum_Stay	INPUT	5.62287	21.03468	100000	0	1	2	1124	13.8115	314.7265
Nightly_Rate	INPUT	341.922	418.2686	100000	0	1	160	1999	1.912019	3.014123
Nightly_Rate_tractQuartile	INPUT	1.483692	1.165921	95937	4063	0	1	3	0.032506	-1.46431
Number_of_Photos	INPUT	15.07596	12.44424	99999	1	0	12	239	3.021175	20.55715
Number_of_Reviews	INPUT	13.87502	35.90509	99998	2	0	1	787	6.058804	59.09142
Rating_Overall	INPUT	92.99618	16.15617	55425	44575	0	98	100	-4.58544	22.40229
Superhost	INPUT	0.18261	0.386348	100000	0	0	0	1	1.643058	0.699653
VAR94	INPUT	1.479676	1.169299	78330	21670	0	1	3	0.038312	-1.47102
available_days	INPUT	169.9919	73.28711	79001	20999	0	190	245	-0.69007	-0.78808
available_days_aveListedPrice	INPUT	277.3311	390.8738	79000	21000	1	107	9999	2.962954	20.91378
available_days_aveListedPrice_tr	INPUT	1.469163	1.169468	75202	24798	0	1	3	0.05066	-1.47034
booked_days	INPUT	20.84138	18.13564	39099	60901	1	16	158	1.190395	1.737699
booked_days_avePrice	INPUT	132.8318	192.578	39099	60901	1	85	6000	5.649188	53.02579
booked_days_period_city	INPUT	71200.94	21115.22	100000	0	30734	72582	106875	0.197252	-0.61723
booked_days_period_tract	INPUT	673.6401	917.8605	100000	0	0	409	10221	4.785591	37.80528
census_tract	INPUT	4.82E10	2345174	100000	0	4.816E10	4.82E10	4.82E10	-23.0522	229817.6
hostResponseAverage_pastYear	INPUT	91.28517	22.11066	69041	30959	0	100	100	-3.327	10.28678
hostResponseNumber_pastYear	INPUT	51.50499	77.96644	69041	30959	1	21	394	2.545333	6.402837
host_is_superhost_in_period	INPUT	0.18261	0.386348	100000	0	0	0	1	1.643058	0.699653
numCancel_pastYear	INPUT	0.532909	1.51799	53891	46109	0	0	61	12.28798	335.8144

Data
~146

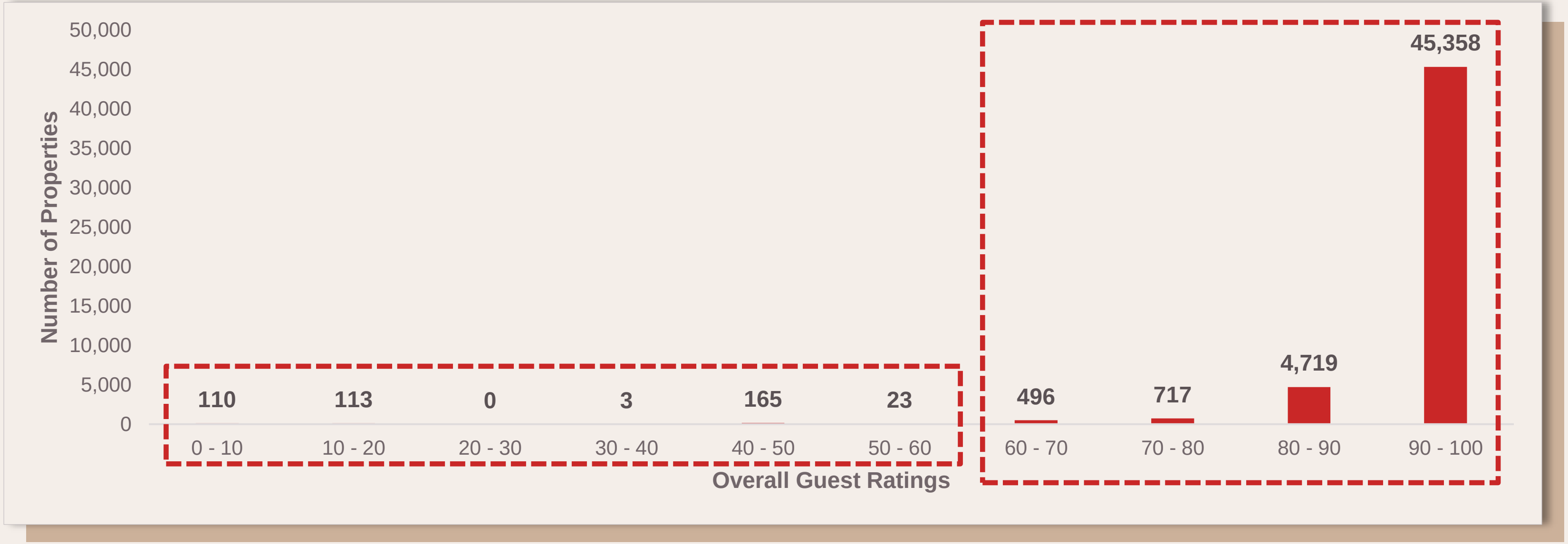
Tourist Attractions

Data Insi

Nu ns: 103
Nu mns: 7

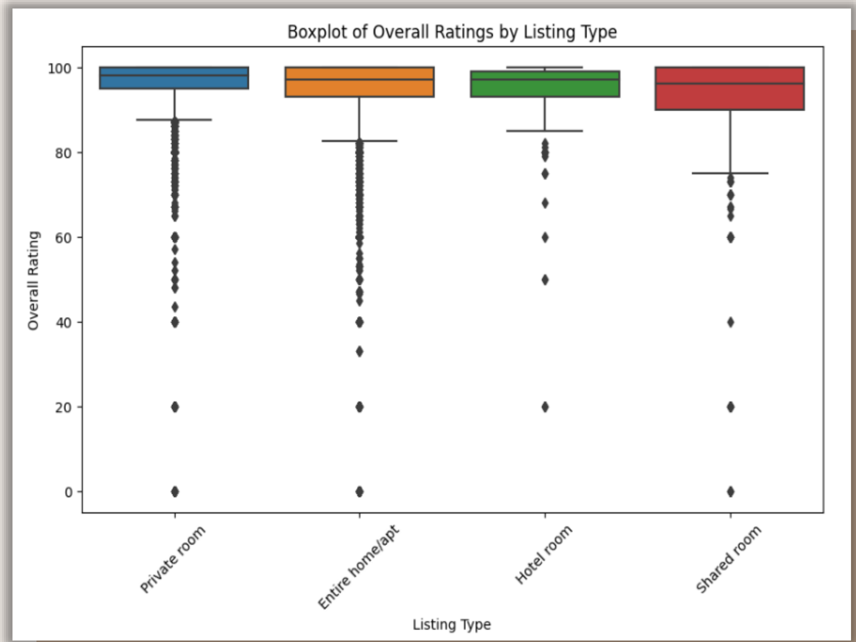
C Categorical)

Overall Guest Ratings



- ✓ Showcases **extremely positive** guest experience on Airbnb, with a staggering 88% of the properties scoring sky-high ratings of 90-100
- ✓ Only a **handful fall below the 20 rating mark**, making a stay in an Airbnb in this area almost a sure bet for a fantastic experience.

Outlier Handling



- **Outliers Reflect Reality:** Highlight the extremes of guest experiences, essential for understanding overall property performance
- **Valuable Data Points**: Genuine outliers provide crucial insights for predictive models, spotlighting issues like cleanliness and host responsiveness.
- **Robust Modeling:** Training with outliers prepares the model for unusual cases, enhancing its robustness.

Overfitting risk: Models might be overfitted to the outliers, learning noise rather than the signal, which could reduce generalizability.

Handling missing data

01

Revenue

Dropped all columns with blank revenue

02

Replace with 0

Based on business context replacing blank with 0 for columns like cleaning fee etc.

03

Group by ID

Replacing missing values for numerical columns based on median of property ID group

04

Standardize

Enhancing algorithm performance and fairness across different scales and units.

05

Elbow Method

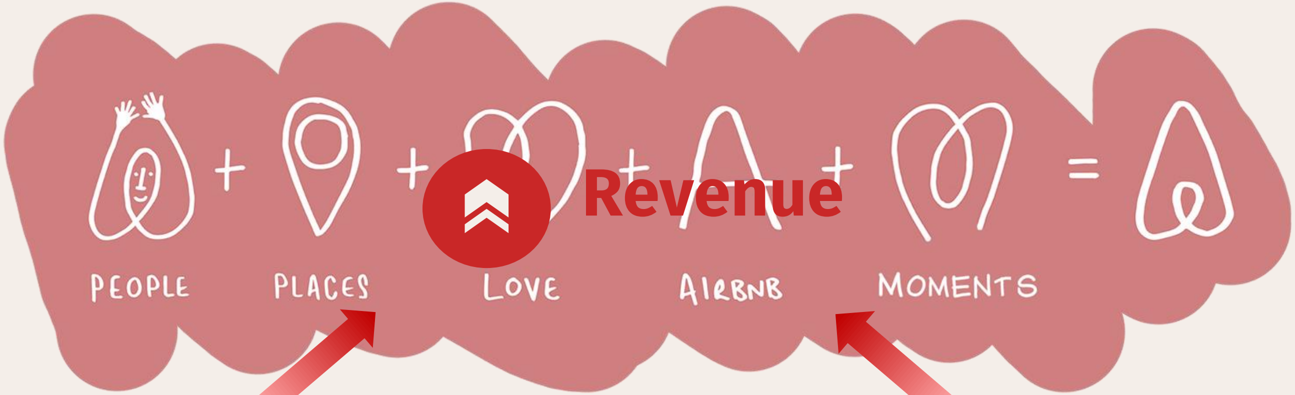
Found 16 as optimal number of centroids using elbow method

06

Cluster

Fill in remaining missing values by k means clustering, cluster mean for numerical and mode for categorical

Objective



Revenue

Overall Rating

Although rating does not have strong direct correlation with Revenue, maintaining a healthy portfolio of listings on the site will boost network effect and customer loyalty

Occupancy Rating

Occupancy Rate has around a 30% correlation with Revenue, a good metric for hosts to consider to boost they’re earnings

Our Approach

Sampling

Stratified sampling based on Property Type to get a healthy mix of population.

Split Train & Test

Utilised both 60:40 & 75:25 split for the data partition.

One hot Encoding

Handled categorical variables to evaluate its impact on target variable.

Transformations

Experimented with different scaler transformations on the numerical columns. ex: MaxAbsScaler(), StandardScaler()

Model 1: *Unveiling Impactful Features for predicting Overall Ratings for Enhanced Host Satisfaction*

1 Feature Selection

- ✓ prev_Rating Overall
- ✓ Number of Reviews,
- ✓ rating_ave_pastYear
- ✓ prev_Number of Reviews
- ✓ Number of Photos
- ✓ Instantbook Enabled
- ✓ Cleaning Fee (USD)

2 Model Selection

Model	Train R2	Valid R2	Train RMSE
Linear Regression	0.5070	0.4682	5.91
Random Forest	0.8489	0.8380	10.29
Lasso Regression	0.1327	0.1109	98.06
Gradient Boosting	0.8289	0.788	11.66
Ensemble	0.2538	0.2176	-7.76

3 Result : Final Model

Random Forest

- Versatile
- Competitive R2 Score
- Feature Importance
- Reduced Sensitivity to Outliers
- Robustness
- Handling Non-linearity

Robustness of results

- R-squared (Train): **0.9617**
- R-squared (Test): **0.8489**

Model 2: *Predicting Occupancy Rates for Enhanced Market Competitiveness*

1 Feature Selection

- ✓ numReviews_pastYear^2
- ✓ numReviews_pastYear
- ✓ numReserv_pastYear
- ✓ booked_days
- ✓ num_5_star_Rev_pastYear
- ✓ Max Guests
- ✓ Prev_available_days
- ✓ Prev_booked_days
- ✓ Nightly Rate
- ✓ numReserv_pastYear

2 Model Selection

Model	Train R2	Valid R2	Train RMSE	Valid RMSE
Polynomial Regression	0.868	0.861	0.063	0.065
Linear Regression	0.680	0.675	0.098	0.100
Random Forest	1.000	0.998	0.003	0.008
Gradient Boosting	0.991	0.989	0.017	0.019
XGBoost	1.000	0.999	0.003	0.006

3 Result : Final Model

Polynomial Regression

- Interpretability
- Competitive R2 Score
- Low Complexity
- Robust Predictive
- Performance
- Balanced Trade-Off

Robustness of results

- Cross-Validation Mean R-squared: **0.8634**
- Standard Deviation of CV R-squared: **0.0047**

Business Insights

Our framework

To analyze the effect of non-controllable and controllable factors on overall ratings and occupancy rate, empowering hosts to enhance satisfaction and optimize revenue.

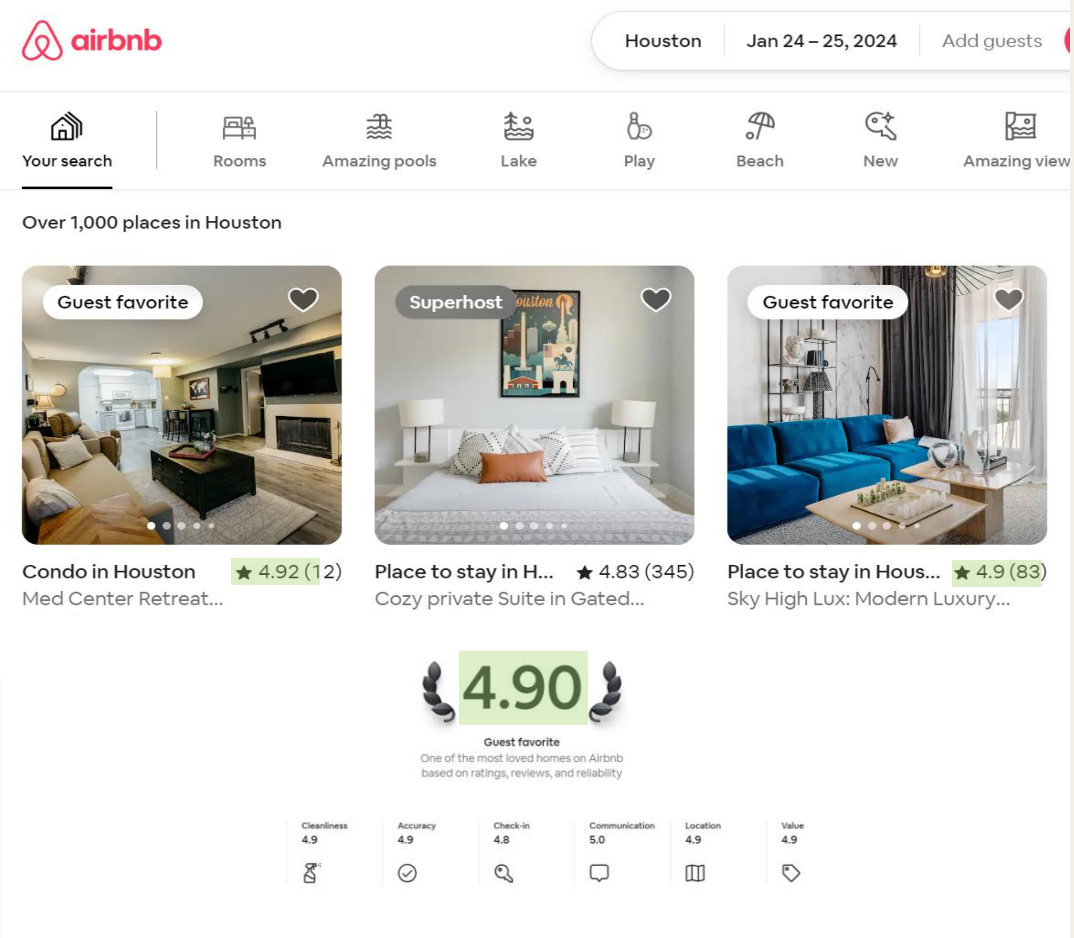
Customer focused approach

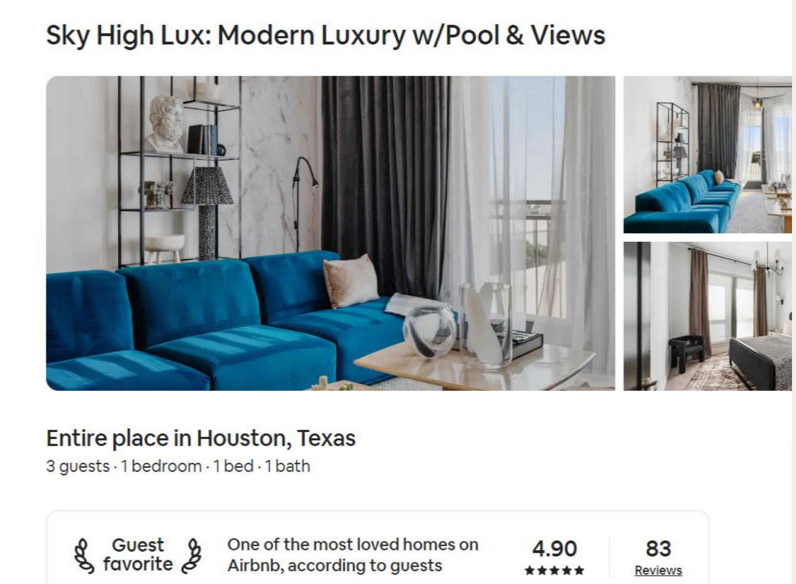
Ratings Overall

Including Instant Book, maintaining consistent ratings, and earning high reviews positively impacts overall ratings, boosting business performance.

Occupancy Rate

5 start reviews, reservations, and booked days, max guest & nightly rate all influence the occupancy rate.





Recommendations

Display predicted high-rate listings in green, making it easier for users to quickly assess the quality of a listing. This visual update aims to enhance the user experience by providing a more intuitive and immediate understanding of a property's reputation even for newer properties.

Learnings from the project

- **Challenge Hypotheses**: While it's natural to form initial hypotheses, we discovered unexpected trends after data analysis.
- **Target-Dependent Modeling**: The choice of the best predictive model is highly dependent on the specific goal, whether it's forecasting ratings or predicting booking likelihood.
- **Importance of Feature Selection**: Identify which features are most predictive for your specific objectives. Different targets might require focusing on different subsets of features.

References

- https://masterhost.ca/airbnb-profitabilityhouston/#:~:text=The%20Houston%20Airbnb%20market%20manifests,Max%20Daily%20Rate%20reaches%20%24204. https://www.mashvisor.com/blog/airbnb-houston/https://www.hostyapp.com/why-airbnb-in-houston-is-a-great-investment-short-review/
- https://chat.openai.com/share/91c12a67-1726-4d4b-bb93-3ec6f0ebc691
- https://chat.openai.com/share/d9498290-7e75-4f06-ad80-8779ad3ac9d3
- https://chat.openai.com/c/07fc29e3-284a-4d9b-bc8f-ecd0b69114de
- Effects of reputation on guest satisfaction: From the perspective of two-sided reviews on Airbnb: https://scholar.google.com/citations?view_op=view_citation&hl=en&user=O0YpdN8AAAAJ&citation_for_view=O0YpdN8AAAAJ:R3hNpaxXUhUC

BACKUP SLIDES

Model performance for occupancy rate prediction

Model 2	Training R2	Validation R2	Training RMSE	Validation RMSE
Polynomial Regression	0.867744	0.861158	0.063051	0.065165
Linear Regression	0.680242	0.675288	0.098039	0.099656
Random Forest	0.999645	0.997681	0.003265	0.008422
Gradient Boosting	0.990638	0.988669	0.016776	0.018616
XGBoost	0.999660	0.998804	0.003196	0.006048

Summary Stats

Variable Summary

Role	Measurement Level	Frequency Count
INPUT	INTERVAL	90
INPUT	NOMINAL	8
REJECTED	NOMINAL	10
TARGET	INTERVAL	1

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	Listing_Type	INPUT	4	0	Entire home/apt	68.31	Private room	29.38
TRAIN	Pets_Allowed	INPUT	2	0	False	79.02	True	20.98
TRAIN	Property_Type	INPUT	65	0	Apartment	36.21	House	33.92
TRAIN	Property_Type_1	INPUT	65	0	Apartment	36.21	House	33.92

(maximum 500 observations printed)

Data Role=TRAIN Type=PEARSON Target=occupancy_rate

Input	Correlation
booked_days	0.60655
prev_occupancy_rate	0.33886
revenue	0.28681
time_to_date_mean	0.26579
prev_booked_days	0.20150
prev_numReserv_pastYear	0.16747
prev_numReservedDays_pastYear	0.16738
numReservedDays_pastYear	0.16721
numReserv_pastYear	0.15974
prev_scrapes_in_period	0.15253
superhost_observed_in_period	0.14863
scrapes_in_period	0.14862
superhost_ratio	0.12957
Superhost	0.12932
host_is_superhost_in_period	0.12932
hostResponseAverage_pastYear	0.11644
prev_time_to_date_mean	0.11555
Number_of_Reviews	0.11478
prev_host_is_superhost	0.11136
prev_host_is_superhost_in_period	0.11136
Instantbook_Enabled	0.11049
prev_year_superhosts	0.10945
prev_hostResponseAverage_pastYea	0.10805
prev_Number_of_Reviews	0.10434
prev_host_is_superhost1	0.09821
hostResponseNumber_pastYear	0.09126
Number_of_Photos	0.08977
prev_Instantbook_Enabled	0.08573
prev_hostResponseNumber_pastYear	0.08288
num_5_star_Rev_pastYear	0.07954
prev_host_is_superhost2	0.07895
prev_num_5_star_Rev_pastYear	0.07883
numReviews_pastYear	0.07414
tract_booking_share	0.07371
booked_days_period_tract	0.07339
rating_ave_pastYear	0.07262
prev_numReviews_pastYear	0.07239
tract_superhosts	0.07118
prev_rating_ave_pastYear	0.06806
tract_prev_superhosts	0.06739

Summary Stats

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Airbnb_Host_ID	INPUT	62915837	46727844	100000	0	4844	55848197	2.7821E8	0.641465	-0.14997
Airbnb_Property_ID	INPUT	16014533	5553962	100000	0	3816	16687261	28752314	-0.26242	0.449922
Bathrooms	INPUT	1.518815	0.793633	99976	24	0	1	10.5	1.92089	6.741808
Bedrooms	INPUT	1.606364	0.992958	99996	4	0	1	20	2.1094	15.3711
Cleaning_Fee__USD_	INPUT	75.47839	62.60453	65070	34930	5	65	999	2.606023	17.10971
Instantbook_Enabled	INPUT	0.5258	0.499336	100000	0	0	1	1	-0.10334	-1.98936
Latitude	INPUT	29.74286	0.066982	100000	0	29.53411	29.74024	30.03549	1.130828	3.639843
Longitude	INPUT	-95.4262	0.086516	100000	0	-95.7185	-95.4069	-95.0618	-0.33568	2.58371
Max_Guests	INPUT	4.067776	2.516134	99991	9	1	4	16	1.407957	2.825913
Minimum_Stay	INPUT	5.62287	21.03468	100000	0	1	2	1124	13.8115	314.7265
Nightly_Rate	INPUT	341.922	418.2686	100000	0	1	160	1999	1.912019	3.014123
Nightly_Rate_tractQuartile	INPUT	1.483692	1.165921	95937	4063	0	1	3	0.032506	-1.46431
Number_of_Photos	INPUT	15.07596	12.44424	99999	1	0	12	239	3.021175	20.55715
Number_of_Reviews	INPUT	13.87502	35.90509	99998	2	0	1	787	6.058804	59.09142
Rating_Overall	INPUT	92.99618	16.15617	55425	44575	0	98	100	-4.58544	22.40229
Superhost	INPUT	0.18261	0.386348	100000	0	0	0	1	1.643058	0.699653
VAR94	INPUT	1.479676	1.169299	78330	21670	0	1	3	0.038312	-1.47102
available_days	INPUT	169.9919	73.28711	79001	20999	0	190	245	-0.69007	-0.78808
available_days_aveListedPrice	INPUT	277.3311	390.8738	79000	21000	1	107	9999	2.962954	20.91378
available_days_aveListedPrice_tr	INPUT	1.469163	1.169468	75202	24798	0	1	3	0.05066	-1.47034
booked_days	INPUT	20.84138	18.13564	39099	60901	1	16	158	1.190395	1.737699
booked_days_avePrice	INPUT	132.8318	192.578	39099	60901	1	85	6000	5.649188	53.02579
booked_days_period_city	INPUT	71200.94	21115.22	100000	0	30734	72582	106875	0.197252	-0.61723
booked_days_period_tract	INPUT	673.6401	917.8605	100000	0	0	409	10221	4.785591	37.80528
census_tract	INPUT	4.82E10	2345174	100000	0	4.816E10	4.82E10	4.82E10	-23.0522	229817.6
hostResponseAverage_pastYear	INPUT	91.28517	22.11066	69041	30959	0	100	100	-3.327	10.28678
hostResponseNumber_pastYear	INPUT	51.50499	77.96644	69041	30959	1	21	394	2.545333	6.402837
host_is_superhost_in_period	INPUT	0.18261	0.386348	100000	0	0	0	1	1.643058	0.699653
numCancel_pastYear	INPUT	0.532909	1.51799	53891	46109	0	0	61	12.28798	335.8144
numReserv_pastYear	INPUT	173.6599	713.2525	90804	9196	0	8	19797	10.7052	185.6187
numReservedDays_pastYear	INPUT	1008.701	4018.815	90804	9196	0	37	58410	5.737686	36.63785
numReviews_pastYear	INPUT	62.19805	156.4643	53891	46109	0	19	3264	8.295648	92.79703
num_5_star_Rev_pastYear	INPUT	50.12336	126.9417	53891	46109	0	15	2616	8.528642	98.5173
prev_Instantbook_Enabled	INPUT	0.50923	0.499917	100000	0	0	1	1	-0.03693	-1.99868
prev_Nightly_Rate	INPUT	357.3829	448.9296	95567	4433	1	165	10000	2.481618	13.04243
prev_Nightly_Rate_tractQuartile	INPUT	1.477038	1.166073	91456	8544	0	1	3	0.041622	-1.46397
prev_Number_of_Reviews	INPUT	12.5076	33.75259	95565	4435	0	1	768	6.297282	63.59965
prev_Rating_Overall	INPUT	93.02761	16.60433	50560	49440	0	98	100	-4.55218	21.71058

Step-by-step code flow

Importing libraries

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures, MaxAbsScaler,
        OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import numpy as np
from sklearn.pipeline import Pipeline
```

In this part, you import the necessary Python libraries for data manipulation, machine learning, and evaluation.

Loading the Dataset

```
# Load the dataset data =
        pd.read_csv('/content/AirbnbHouston_Preprocessed_dataset
        2.csv') # Adjust path as needed
```

Here, you load your dataset from a CSV file into a Pandas DataFrame. The path to the dataset file is specified, and you should adjust it to your file's actual location.

Defining Target and Features

```
# Define the target variable and features
target = 'occupancy_rate'
features = ['host_is_superhost_in_period', 'numReviews_pastYear',
            'booked_days', ...]  # List of feature names
X = data[features]
y = data[target]
```

You specify the target variable ('occupancy_rate') and the list of feature columns that will be used for modeling. X contains the feature data, and y contains the target variable.

Splitting the Dataset

```
# Splitting the dataset into training and validation sets (60:40)
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.4,
        random_state=42)
```

The dataset is split into training and validation sets using a 60:40 ratio. The random_state ensures reproducibility.

Data Preprocessing

```
# Preprocessing
numeric_cols = X.select_dtypes(include=['int64', 'float64']).columns
categorical_cols = X.select_dtypes(include=['object', 'category']).columns

numeric_transformer = Pipeline(steps=[...])  # Numeric data preprocessing
categorical_transformer = Pipeline(steps=[...])  # Categorical data
        preprocessing
preprocessor = ColumnTransformer(transformers=[...])  # Apply
        transformations to numeric and categorical columns

X_train_processed = preprocessor.fit_transform(X_train)
X_val_processed = preprocessor.transform(X_val)
```

This section performs data preprocessing steps, including imputation (filling missing values) and scaling for numeric features and one-hot encoding for categorical features. The ColumnTransformer is used to apply these transformations to the appropriate columns in the dataset.

Creating Polynomial Features

```
# Create polynomial features
poly_features = PolynomialFeatures(degree=2)
X_train_poly = poly_features.fit_transform(X_train_processed)
X_val_poly = poly_features.transform(X_val_processed)
```

Polynomial features of degree 2 are generated from the preprocessed data. This allows the model to capture more complex relationships between features.

Linear Regression Model

```
# Linear Regression Model
linear_reg = LinearRegression()
linear_reg.fit(X_train_poly, y_train)
```

A linear regression model is instantiated and trained on the polynomial features of the training data.

Predictions and Evaluation

```
# Predict and evaluate on training and validation data
y_train_pred = linear_reg.predict(X_train_poly)
y_val_pred = linear_reg.predict(X_val_poly)

train_mse = mean_squared_error(y_train, y_train_pred)
train_r2 = r2_score(y_train, y_train_pred)
val_mse = mean_squared_error(y_val, y_val_pred)
val_r2 = r2_score(y_val, y_val_pred)
# Print model summary
print("Model Summary:")

print("Training MSE:", train_mse)
print("Training R-squared:", train_r2)
print("Validation MSE:", val_mse)
print("Validation R-squared:", val_r2)

print("Intercept:", linear_reg.intercept_)
print("Coefficients:", linear_reg.coef_)
```

The model is used to make predictions on both the training and validation datasets, and various evaluation metrics such as Mean Squared Error (MSE), R-squared

Random Forest Model

```
X = sampled_data[selected_columns]
y = sampled_data['Rating Overall']

        # Identify numeric and categorical features
numeric_features = X.select_dtypes(include=[np.number]).columns
categorical_features = X.select_dtypes(include=[np.object]).columns

        # Create preprocessing pipeline
numeric_transformer = Pipeline(steps=[
  ('imputer', SimpleImputer(strategy='mean')),
  ('scaler', StandardScaler())
])

        categorical_transformer = Pipeline(steps=[
  ('imputer', SimpleImputer(strategy='most_frequent')),
  ('onehot', OneHotEncoder(handle_unknown='ignore'))
])

        preprocessor = ColumnTransformer(
  transformers=[
    ('num', numeric_transformer, numeric_features),
    ('cat', categorical_transformer, categorical_features)
  ])

        # Random Forest
rf_model = Pipeline(steps=[('preprocessor', preprocessor),
                ('regressor', RandomForestRegressor(n_estimators=100,
        random_state=42))])
```

A Random Forest regression model is initialized and fitted to the training dataset, leveraging an ensemble of decision trees to grasp intricate patterns and relationships present in the data.

Predictions and Evaluation

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30,
        random_state=42)

        # Fit the Random Forest model
rf_model.fit(X_train, y_train)

# Predictions
y_pred_rf = rf_model.predict(X_test)
# Evaluate Random Forest
mse_rf = mean_squared_error(y_test, y_pred_rf)

r2_rf = r2_score(y_test, y_pred_rf)

        print("Random Forest:")
print("RMSE:", mse_rf)
print("R-squared:", r2_rf)
```

The model is used to make predictions on both the training and validation datasets, and various evaluation metrics such as Mean Squared Error (MSE), R-squared

Top influencial Features & their contribution (1/2)

Random Forest

Previous Rating Overall: Importance - 0.874963

Number of Reviews: Importance - 0.068726

Average Rating in the Past Year: Importance - 0.016475

Previous Number of Reviews: Importance - 0.007278

Booked Days Average Price: Importance - 0.003343

Tract Housing Units: Importance - 0.003223

Proportion of 5-Star Reviews in the Past Year: Importance - 0.002739

Previous Revenue: Importance - 0.002578

Previous Nightly Rate: Importance - 0.002034

Longitude: Importance - 0.001286

Number of Cancellations in the Past Year: Importance - 0.000948

Top influencial Features & their contribution (2/2)

Polynomial Regression

- Intercept: 1.14×10^{-13}
- `numReviews_pastYear`: -2.70×10^{-5}
- `num_5_star_Rev_pastYear`: 6.54×10^{-5}
- `numReserv_pastYear`: -5.77×10^{-6}
- `available_days`: -0.00523
- `booked_days`: 0.01737
- `Nightly Rate`: -5.47×10^{-6}
- `Max Guests`: -0.00059
- `numReviews_pastYear^2`: 1.12×10^{-8}
- `numReviews_pastYear num_5_star_Rev_pastYear`: -5.69×10^{-8}
- `numReviews_pastYear numReserv_pastYear`: 9.92×10^{-9}