Trip Advisory Rating Impact on Hotel Popularity

About the Data

Kaggle Link: https://www.kaggle.com/datasets/jocelyndumlao/tripadvisor-rating-impact-on-hotel-popularity

We use detailed data on 4,599 hotels located in Rome collected from TripAdvisor, the world's largest travel platform, to examine the effects of bubble ratings (detailed to half-bubbles) on hotel popularity measured by the number of people viewing the hotel's page. By using a regression discontinuity design, we find that the bubble presentation of ratings does not create any significant jumps at cutoffs. This result is different from those obtained in previous studies of similarly designed rating systems from other industries. We provide possible explanations and implications of this result. Another finding is that web users tend to shortlist hotels with a bubble rating of at least 3. Despite that, there is no compelling evidence of review manipulation around the cutoff of 2.75 to make a transition from the 2.5-bubble rating to the 3-bubble rating. Potential uses of the number of views as a proxy of demand in hospitality research are outlined.

Fields:

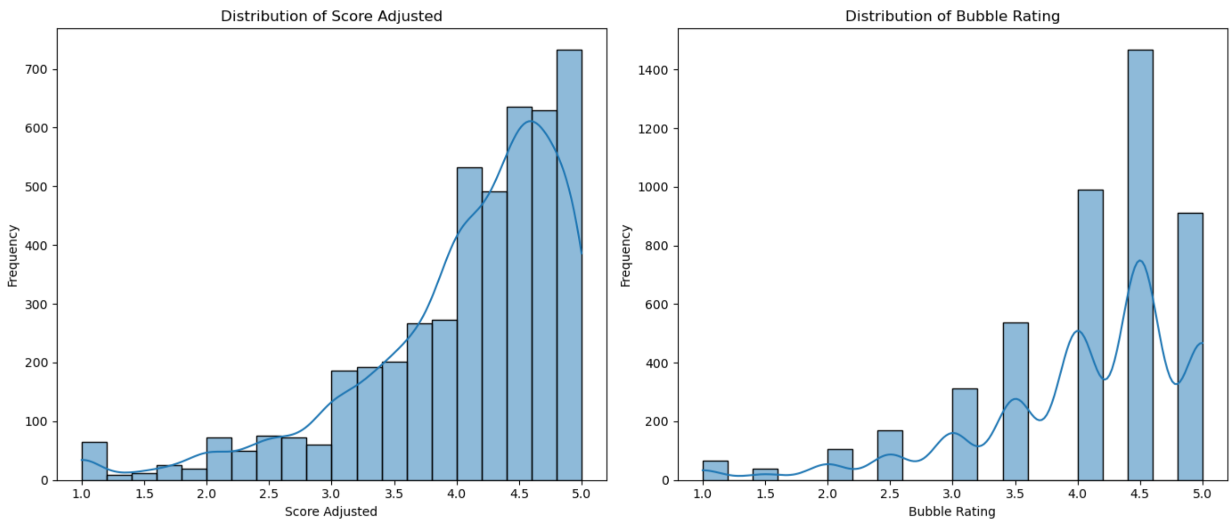
Tourism, Discontinuity, Demand Estimation, Multiple Regression, Causal Inference

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the dataset with ISO-8859-1 encoding
file_path = 'data_rdd.csv'  # Replace with the actual path
df = pd.read_csv(file_path, encoding='ISO-8859-1')

# Basic information about the dataset
print(df.info())

# Descriptive statistics
desc_stats = df.describe()
print(desc_stats)

# Check for missing values
missing_values = df.isnull().sum()
missing_values = missing_values[missing_values > 0]
print("Columns with missing values:", missing_values)

# Visualization
plt.figure(figsize=(14, 6))
```

```
# Plot histograms for 'score_adjusted'
plt.subplot(1, 2, 1)
sns.histplot(df['score_adjusted'], bins=20, kde=True)
plt.title('Distribution of Score Adjusted')
plt.xlabel('Score Adjusted')
plt.ylabel('Frequency')

# Plot histograms for 'bubble_rating'
plt.subplot(1, 2, 2)
sns.histplot(df['bubble_rating'], bins=20, kde=True)
plt.title('Distribution of Bubble Rating')
plt.xlabel('Bubble Rating')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4599 entries, 0 to 4598
Columns: 272 entries, Unnamed: 0 to amenities_Yoga classes
dtypes: float64(8), int64(261), object(3)
memory usage: 9.5+ MB
None
        Unnamed: 0        views  views_binary  score_adjusted  bubble_rating
\
count  4599.000000  4599.000000   4599.000000     4599.000000    4599.000000
mean   2300.000000     0.274625      0.037399        4.058395       4.067841
std    1327.761274     2.231304      0.189759        0.837018       0.850739
min       1.000000     0.000000      0.000000        1.000000       1.000000
25%    1150.500000     0.000000      0.000000        3.673771       3.500000
50%    2300.000000     0.000000      0.000000        4.282051       4.500000
75%    3449.500000     0.000000      0.000000        4.666667       4.500000
max    4599.000000    88.000000      1.000000        5.000000       5.000000

       category_hotel  category_inn  category_specialty    class_4_5  \
count     4599.000000   4599.000000         4599.000000  4599.000000
mean         0.190041      0.619482            0.190476     0.111763
std          0.392376      0.485567            0.392719     0.315109
min          0.000000      0.000000            0.000000     0.000000
25%          0.000000      0.000000            0.000000     0.000000
50%          0.000000      1.000000            0.000000     0.000000
75%          0.000000      1.000000            0.000000     0.000000
max          1.000000      1.000000            1.000000     1.000000

       class_3_4_5  ...  amenities_Wardrobe / closet  \
count  4599.000000  ...                  4599.000000
mean      0.295934  ...                     0.025440
std       0.456511  ...                     0.157475
min       0.000000  ...                     0.000000
25%       0.000000  ...                     0.000000
50%       0.000000  ...                     0.000000
75%       1.000000  ...                     0.000000
max       1.000000  ...                     1.000000

       amenities_Washing machine  amenities_Water park  \
count                4599.000000           4599.000000
mean                    0.020874              0.000435
std                     0.142978              0.020851
min                     0.000000              0.000000
25%                     0.000000              0.000000
50%                     0.000000              0.000000
75%                     0.000000              0.000000
max                     1.000000              1.000000

       amenities_Water park offsite  amenities_Waterslide  \
count                   4599.000000           4599.000000
mean                       0.000870              0.000217
std                        0.029482              0.014746
min                        0.000000              0.000000
25%                        0.000000              0.000000
50%                        0.000000              0.000000
75%                        0.000000              0.000000
max                        1.000000              1.000000

       amenities_Waxing services  amenities_Whirlpool bathtub  amenities_Wifi
\
count                4599.000000                  4599.000000     4599.000000
```

```
mean                    0.002609                        0.004784         0.648619
std                     0.051020                        0.069006         0.477454
min                     0.000000                        0.000000         0.000000
25%                     0.000000                        0.000000         0.000000
50%                     0.000000                        0.000000         1.000000
75%                     0.000000                        0.000000         1.000000
max                     1.000000                        1.000000         1.000000

       amenities_Wine / champagne  amenities_Yoga classes
count                 4599.000000             4599.000000
mean                     0.072624                0.001522
std                      0.259547                0.038988
min                      0.000000                0.000000
25%                      0.000000                0.000000
50%                      0.000000                0.000000
75%                      0.000000                0.000000
max                      1.000000                1.000000

[8 rows x 269 columns]
Columns with missing values: location_grade     218
price_curr_min    2224
price_min         1125
price_max         1127
photos             577
dtype: int64
```



Initial analysis: It contains 4599 rows and 272 columns. The columns consist of various data types, including float64 (8 columns), int64 (261 columns), and object (3 columns).

Histogram & Scores

- Distribution of Score Adjusted: The distribution is slightly left-skewed, indicating that most of the hotels have higher adjusted scores. The peak occurs around a score of 4.5, which suggests that a large number of hotels have ratings close to this value.
- Distribution of Bubble Rating: This distribution is also left-skewed. The majority of hotels have bubble ratings around 4.5, which corroborates the findings from the score_adjusted distribution.

Data Discrprencies

- Fields: Missing data values

- location_grade: 218 missing values

- price_curr_min: 2224 missing values
- price_min: 1125 missing values
- price_max: 1127 missing values
- photos: 577 missing values

Variable Inspection - Marketing Insights

Customer Ratings (score_adjusted, bubble_rating)

- Understand customer satisfaction and identify areas for improvement.

Views (views, views_binary)

- Hotels with fewer views might need more aggressive marketing.

Categories (category_hotel, category_inn, category_specialty)

- Tailor different marketing strategies for hotels, inns, and specialty lodgings.

Amenities

- Promote unique amenities to attract specific customer segments.

Price (price_curr_min, price_min, price_max)

- Dynamic pricing strategies can be developed.

Location (location_grade)

- Use this information for geo-targeted advertising

Possible Marketing Strategies to Deploy

Customer Segmentation

- Use clustering algorithms to segment hotels based on a combination of features like ratings, views, and amenities.

Promotions and Discounts

- Target hotels with lower views or ratings for special promotions to increase visibility and customer engagement.

Upselling and Cross-selling

- Utilize amenities data to create packages that can be upsold to existing bookings.

Geo-Targeting

- Use location data to target customers in specific regions with high-rated but lesser-known hotels.

Dynamic Pricing

- Implement dynamic pricing strategies based on real-time data analytics to maximize revenue.

Reputation Management

- Focus marketing efforts on improving online ratings and reviews for hotels that are lagging in these areas.

Strategy Selection - Dynamic Pricing

Dynamic pricing is a strategy that allows businesses to set flexible prices for products or services based on current market demands. For hotels, factors like time of booking, occupancy rates, special events, and competitor prices could all influence room rates.

Let's create a simple dynamic pricing model based on some of these factors. Since we don't have all the variables typically used in a dynamic pricing model (like competitor prices, real-time demand, etc.), we will use what we have:

- Customer Ratings (score_adjusted, bubble_rating)
- Views (views)
- Amenities (We will count the number of amenities offered by each hotel)
- Location Grade (location_grade)
- Time to Booking (A simulated variable for demonstration)

Dynamic Price - Model Formula

Dynamic Price = Base Price * (1 + $(R.F. + V.F. + A)/10$)

- R.F. = Rating Factor
- V.F. = View Factor
- A = Amentity

In [3]:
```
# Handle missing values in our selected variables.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
# Load dataset and handle missing values
df['location_grade'].fillna(df['location_grade'].median(), inplace=True)

# Create 'amenity_count' and 'time_to_booking' columns
amenities_cols = [col for col in df.columns if col.startswith('amenities_')]
df['amenity_count'] = df[amenities_cols].sum(axis=1)
np.random.seed(0)
df['time_to_booking'] = np.random.randint(1, 31, df.shape[0])

# Set base price and calculate factors
base_price = 100
rating_factor = (df['score_adjusted'] / df['score_adjusted'].max()) * 100
view_factor = (df['views'] / df['views'].max()) * 100
amenity_factor = (df['amenity_count'] / df['amenity_count'].max()) * 100
location_factor = (df['location_grade'] / df['location_grade'].max()) * 100
time_factor = (df['time_to_booking'] / df['time_to_booking'].max()) * 100

# Calculate dynamic price
df['dynamic_price'] = base_price * (1 + (rating_factor + view_factor + amenity_
df.head(5)
```

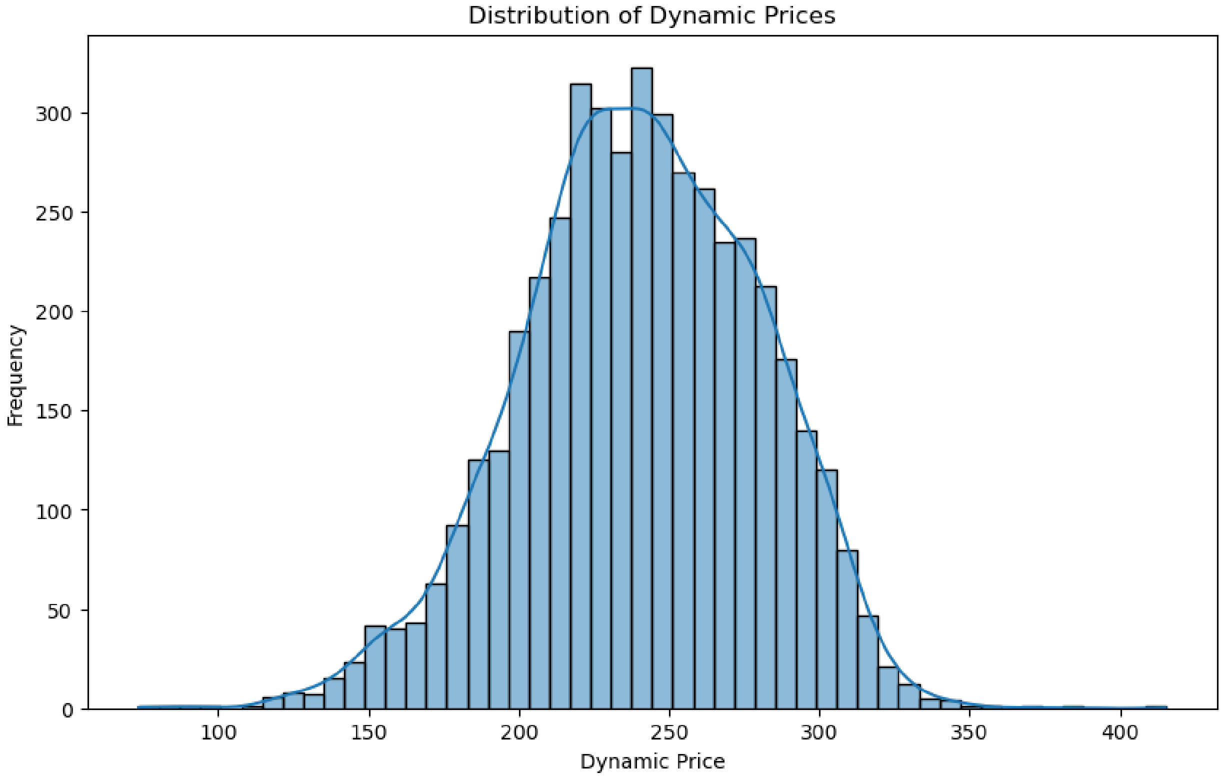
Out[3]:

	Unnamed: 0	hotel_url	name	views	views_binary	score_ac
0	1	https://www.tripadvisor.com/Hotel_Review-g1877...	Casa Mia in Trastevere	0	0	4.4
1	2	https://www.tripadvisor.com/Hotel_Review-g1877...	Hotel Artemide	88	1	4.
2	3	https://www.tripadvisor.com/Hotel_Review-g1877...	A.Roma Lifestyle Hotel	32	1	4.6
3	4	https://www.tripadvisor.com/Hotel_Review-g1877...	iQ Hotel Roma	17	1	4.6
4	5	https://www.tripadvisor.com/Hotel_Review-g1877...	The Guardian	0	0	4.6

5 rows × 275 columns

In [4]:

```
## Plot
import matplotlib.pyplot as plt
import seaborn as sns

# Plot the distribution of the dynamic prices
plt.figure(figsize=(10, 6))
sns.histplot(df['dynamic_price'], bins=50, kde=True)
plt.title('Distribution of Dynamic Prices')
plt.xlabel('Dynamic Price')
plt.ylabel('Frequency')
plt.show()
```



Our Kernel Denisy estimation gives us the smooth curve over our dynamic price. The histogram shows that most prices are clustered around $250 - \$350 dollar range. This variation reflects the factors included such as customer rating, views, amentity counts, and location grades.

Customer Segmentation - Decision Tree Classifier

Customer segmentation in the context of hotels can help identify different groups of hotels that share similar characteristics. This can be useful for targeted marketing, among other applications.Since we are dealing with a segmentation task, we first need to define the categories or segments into which we wish to divide the hotels.

- Low-Price Hotels: Hotels with dynamic prices in the lower 33% percentile.
- Mid-Price Hotels: Hotels with dynamic prices between the 33% and 66% percentiles.
- High-Price Hotels: Hotels with dynamic prices in the upper 33% percentile.

In [5]:
```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report

# Load dataset and handle missing values
df['location_grade'].fillna(df['location_grade'].median(), inplace=True)

# Create 'amenity_count' and 'time_to_booking' columns
amenities_cols = [col for col in df.columns if col.startswith('amenities_')]
```

```
df['amenity_count'] = df[amenities_cols].sum(axis=1)
np.random.seed(0)
df['time_to_booking'] = np.random.randint(1, 31, df.shape[0])

# Create target variable based on dynamic_price percentiles
low_price_threshold = df['dynamic_price'].quantile(0.33)
mid_price_threshold = df['dynamic_price'].quantile(0.66)
df['price_segment'] = 'High-Price Hotels'
df.loc[df['dynamic_price'] <= mid_price_threshold, 'price_segment'] = 'Mid-Pric
df.loc[df['dynamic_price'] <= low_price_threshold, 'price_segment'] = 'Low-Pric

# Select features and target variable
features = ['score_adjusted', 'bubble_rating', 'views', 'location_grade', 'amen
target = 'price_segment'
X = df[features]
y = df[target]

# Split data into training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, randor

# Initialize and train Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)
clf.fit(X_train, y_train)

# Predict on the test set and evaluate the model
y_pred = clf.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Print evaluation metrics
print(f'Accuracy: {accuracy}')
print(f'Classification Report: \n{classification_rep}')
```

```
Accuracy: 0.9065217391304348
Classification Report:
                   precision    recall  f1-score   support

High-Price Hotels       0.92      0.95      0.93       294
 Low-Price Hotels       0.92      0.94      0.93       311
 Mid-Price Hotels       0.88      0.84      0.86       315

         accuracy                           0.91       920
        macro avg       0.91      0.91      0.91       920
     weighted avg       0.91      0.91      0.91       920
```

The model performs well across all three categories, with F1-scores ranging from 0.86 for Mid-Price Hotels to 0.93 for both High-Price and Low-Price Hotels.

This suggests that the model is fairly good at segmenting hotels into the defined price categories based on the selected features.

Customer Segmentation - Naive Bayes Approach

In [6]:

```
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, classification_report
```

```
# Initialize and train the Gaussian Naive Bayes Classifier
gnb = GaussianNB()
gnb.fit(X_train, y_train)

# Predict on the test set and evaluate
y_pred_nb = gnb.predict(X_test)
accuracy_nb = accuracy_score(y_test, y_pred_nb)
classification_rep_nb = classification_report(y_test, y_pred_nb)

# Print evaluation metrics
print(f'Naive Bayes Accuracy: {accuracy_nb}')
print(f'Naive Bayes Classification Report: \n{classification_rep_nb}')
```

```
Naive Bayes Accuracy: 0.8043478260869565
Naive Bayes Classification Report:
                   precision    recall  f1-score   support

High-Price Hotels       0.96      0.80      0.87       294
 Low-Price Hotels       0.74      0.99      0.85       311
 Mid-Price Hotels       0.76      0.63      0.69       315

         accuracy                           0.80       920
        macro avg       0.82      0.80      0.80       920
     weighted avg       0.82      0.80      0.80       920
```

The model performs reasonably well but not as accurately as the Decision Tree Classifier. It is especially good at identifying "Low-Price Hotels," with a recall of 0.99, but less effective at identifying "Mid-Price Hotels," with a recall of 0.63.

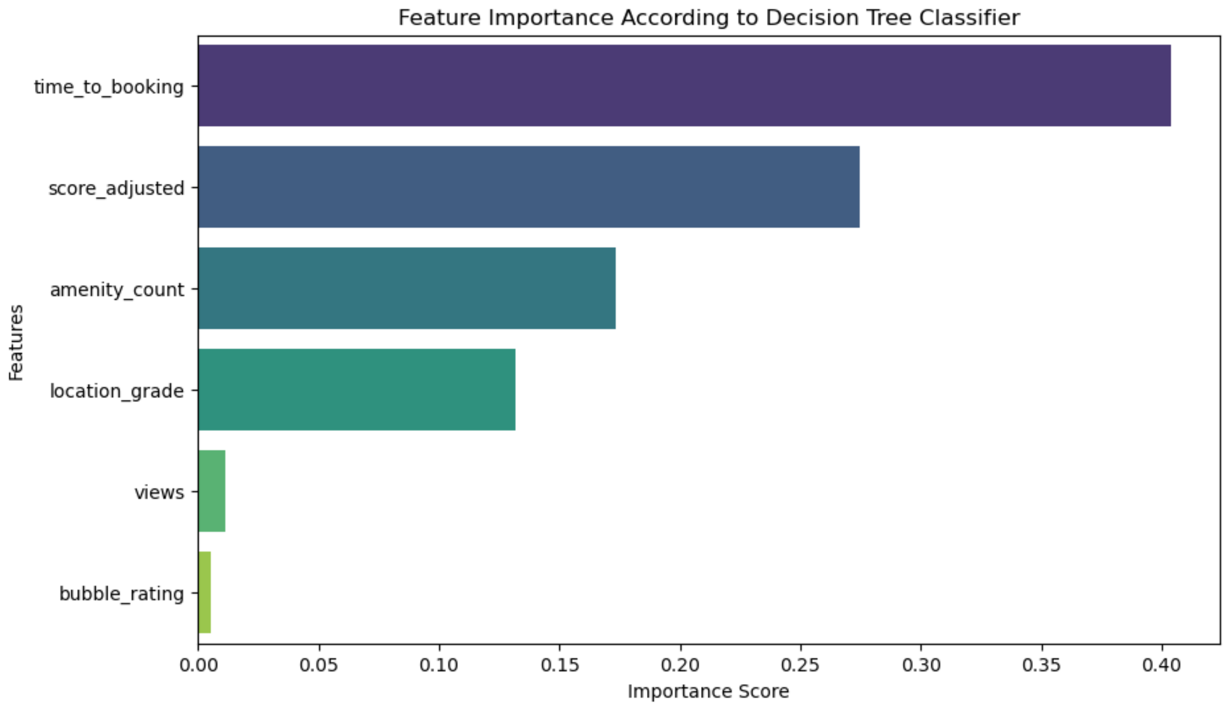
Summary Proposal

Our data-driven approach suggests that the hotel industry can benefit from targeted marketing strategies based on customer segmentation. Utilizing machine learning classifiers like Decision Trees and Naive Bayes, we have effectively categorized hotels into "Low-Price," "Mid-Price," and "High-Price" segments with high accuracy. This categorization is instrumental for tailoring marketing efforts to specific customer bases. For instance, "Low-Price Hotels" could be marketed to budget-conscious travelers, while "High-Price Hotels" can be promoted to luxury seekers. The classifiers also identify key features like customer ratings, views, amenities, and location grade that significantly influence pricing, thus providing avenues for further optimization.

To support this reasoning, let's visualize the most important features that influence hotel pricing according to the Decision Tree Classifier.

In [8]:
```
# Feature Importance from Decision Tree Classifier
feature_importance = clf.feature_importances_

# Create a DataFrame to hold feature names and their importance scores
feature_importance_df = pd.DataFrame({'Feature': features, 'Importance': featu

# Sort the features by importance
feature_importance_df = feature_importance_df.sort_values(by='Importance', asce
```

```
# Plot the feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df, palette='v
plt.title('Feature Importance According to Decision Tree Classifier')
plt.xlabel('Importance Score')
plt.ylabel('Features')
plt.show()
```



The bar graph above displays the feature importance scores according to the Decision Tree Classifier. Among the considered features, score_adjusted (adjusted customer rating) and location_grade emerge as the most influential factors in determining hotel pricing. These key features provide actionable insights for marketing strategies. For example, a focus on improving customer ratings and leveraging location advantages could contribute to a higher dynamic price, and thus potentially higher revenue.

In [ ]: