Hotel Revenue Analysis Project

Agenda

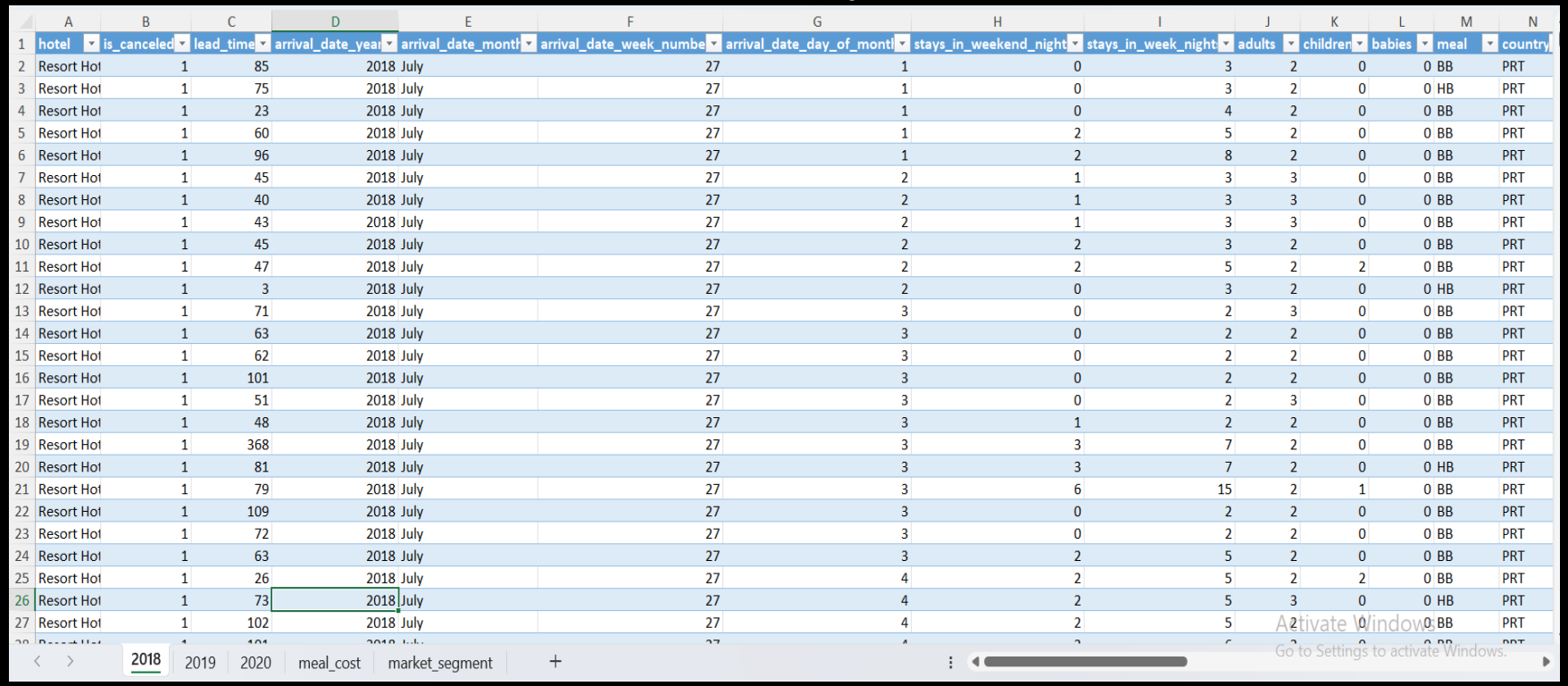
1. Introduction
2. Dataset Background
3. Dataset glossary
4. Dataset Preparation in Excel
5. SQL Data Pre-Processing
6. Hotel revenue data analysis
7. Hotel Revenue Dashboard
8. Recommendations

Introduction

Welcome to my presentation on **Hotel Revenue Analysis**, a project that combines data analytics tools to uncover key insights and optimize revenue strategies in the hospitality industry. The analysis was conducted using a comprehensive approach involving **Excel** for data cleaning and exploratory analysis, **SQL** for efficient data querying, **Python** for advanced analysis and automation, and **Power BI** for interactive visualization.

This project focuses on understanding revenue trends, identifying factors impacting performance, and generating actionable insights to support strategic decision-making. With data-driven methodologies, this analysis provides valuable recommendations for enhancing profitability and operational efficiency in the hotel industry.

Dataset Background



hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	meal	country
Resort Hot	1	85	2018	July	27	1	0	3	2	0	0	BB	PRT
Resort Hot	1	75	2018	July	27	1	0	3	2	0	0	HB	PRT
Resort Hot	1	23	2018	July	27	1	0	4	2	0	0	BB	PRT
Resort Hot	1	60	2018	July	27	1	2	5	2	0	0	BB	PRT
Resort Hot	1	96	2018	July	27	1	2	8	2	0	0	BB	PRT
Resort Hot	1	45	2018	July	27	2	1	3	3	0	0	BB	PRT
Resort Hot	1	40	2018	July	27	2	1	3	3	0	0	BB	PRT
Resort Hot	1	43	2018	July	27	2	1	3	3	0	0	BB	PRT
Resort Hot	1	45	2018	July	27	2	2	3	2	0	0	BB	PRT
Resort Hot	1	47	2018	July	27	2	2	5	2	2	0	BB	PRT
Resort Hot	1	3	2018	July	27	2	0	3	2	0	0	HB	PRT
Resort Hot	1	71	2018	July	27	3	0	2	3	0	0	BB	PRT
Resort Hot	1	63	2018	July	27	3	0	2	2	0	0	BB	PRT
Resort Hot	1	62	2018	July	27	3	0	2	2	0	0	BB	PRT
Resort Hot	1	101	2018	July	27	3	0	2	2	0	0	BB	PRT
Resort Hot	1	51	2018	July	27	3	0	2	3	0	0	BB	PRT
Resort Hot	1	48	2018	July	27	3	1	2	2	0	0	BB	PRT
Resort Hot	1	368	2018	July	27	3	3	7	2	0	0	BB	PRT
Resort Hot	1	81	2018	July	27	3	3	7	2	0	0	HB	PRT
Resort Hot	1	79	2018	July	27	3	6	15	2	1	0	BB	PRT
Resort Hot	1	109	2018	July	27	3	0	2	2	0	0	BB	PRT
Resort Hot	1	72	2018	July	27	3	0	2	2	0	0	BB	PRT
Resort Hot	1	63	2018	July	27	3	2	5	2	0	0	BB	PRT
Resort Hot	1	26	2018	July	27	4	2	5	2	2	0	BB	PRT
Resort Hot	1	73	2018	July	27	4	2	5	3	0	0	HB	PRT
Resort Hot	1	102	2018	July	27	4	2	5	2	0	0	BB	PRT

The dataset used for this project was gathered from a hotel booking website and provides comprehensive information on hotel reservations over three years: **2018**, **2019**, and **2020**. It includes detailed records for two types of hotels **Resort Hotel** and **City Hotel** capturing various aspects of the booking process and guest preferences

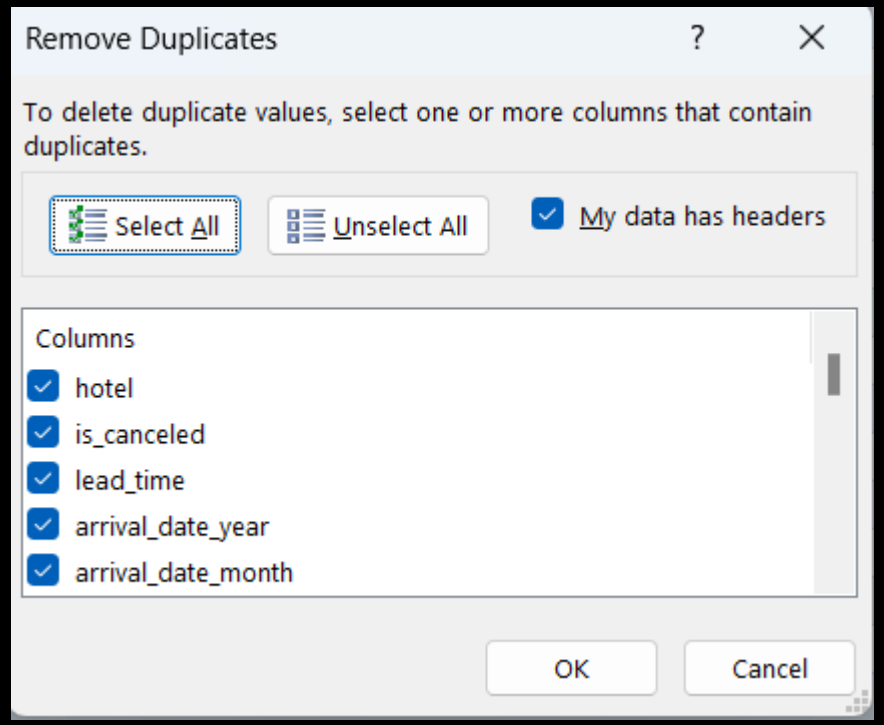
Dataset glossary

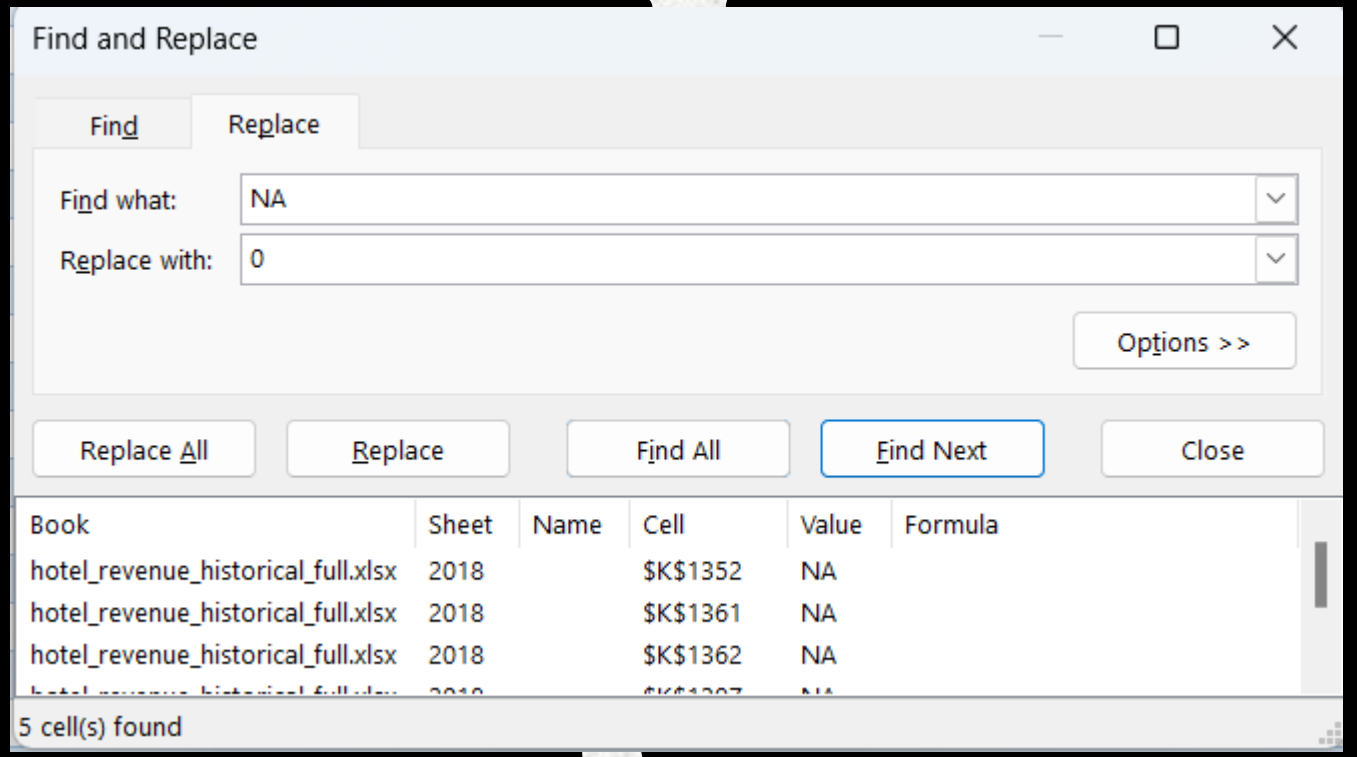
Field Name	Description
Hotel	Hotel (H1 = Resort Hotel or H2 = City Hotel)
is_canceled	Value indicating if the booking was canceled (1) or not (0)
lead_time	Number of days that elapsed between the entering date of the booking into the PMS and the arrival date
Arrival_date_year	Year of arrival date
Arrival_date_month	Month of arrival date
Arrival_date_week_number	Week number of year for arrival date
Arrival_date_day_of_month	Day of arrival date
Stays_in_weekend_nights	Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
Stays_in_week_nights	Number of weeknights (Monday to Friday) the guest stayed or booked to stay at the hotel
Adults	Number of adults
Children	Number of children
Babies	Number of babies
Cost	Cost incurred by the hotel company related to the associated meal package provided to the customer

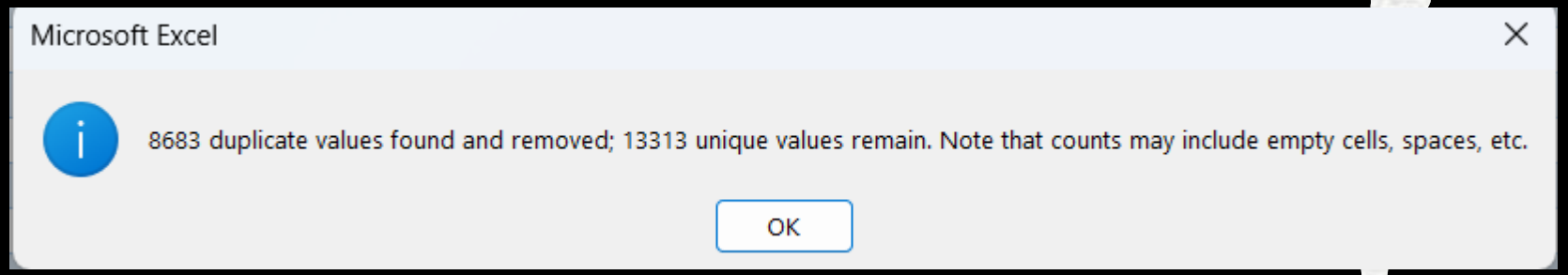
Dataset glossary

Field Name	Description
Market_segment	Market segment designation. In categories, the term “TA” means “Travel Agents” and “TO” means “Tour Operators”
Distribution_channel	Booking distribution channel. The term “TA” means “Travel Agents” and “TO” means “Tour Operators”
Is_repeated_guest	Value indicating if the booking name was from a repeated guest (1) or not (0)
Previous_cancellations	Number of previous bookings that were cancelled by the customer prior to the current booking
Previous_bookings_not_cancelled	Number of previous bookings not cancelled by the customer prior to the current booking
Reserved_room_type	Code of room type reserved. Code is presented instead of designation for anonymity reasons.
Agent	ID of the travel agency that made the booking

Dataset Preparation in Excel







SQL Data Pre-Processing

i1.sql - DESKTOP-...N4A1LD8\ASUS (59))

```sql
/*
    Baseline Query to consolidate data across all tables
*/
-- Given that we'd like to analyze trends, revenue growth across each year, and that the names of
-- columns remain the same across each year, UNION ALL was performed across all years to consolidate
-- data. We'll use CTE and refer to the consolidated table as Hotel. At the same time, the foreign keys
-- market_segment, meal was used to JOIN market_segment$ & meal_cost$ tables together:

WITH hotels AS (
    SELECT * FROM [dbo].[2018]
    UNION ALL
    SELECT * FROM [dbo].[2019]
    UNION ALL
    SELECT * FROM [dbo].[2020]
)
SELECT *
FROM hotels AS h
LEFT JOIN [dbo].[market_segment] AS m ON h.market_segment = m.market_segment
LEFT JOIN [dbo].[meal_cost] AS c ON h.meal = c.meal
```

- Raw .csv file is ingested into Microsoft SQL Server Management Studio (SSMS) and SQL queries were iteratively built upon the baseline query below to obtain the final query to import into SSMS

- The baseline query utilizes UNION ALL, CTE & LEFT JOINs to aggregate all sheets within the excel spreadsheet

SQL Data Pre-Processing

i2.sql - DESKTOP-...N4A1LD8\ASUS (63)) | i1.sql - DESKTOP-...N4A1LD8\ASUS (59))

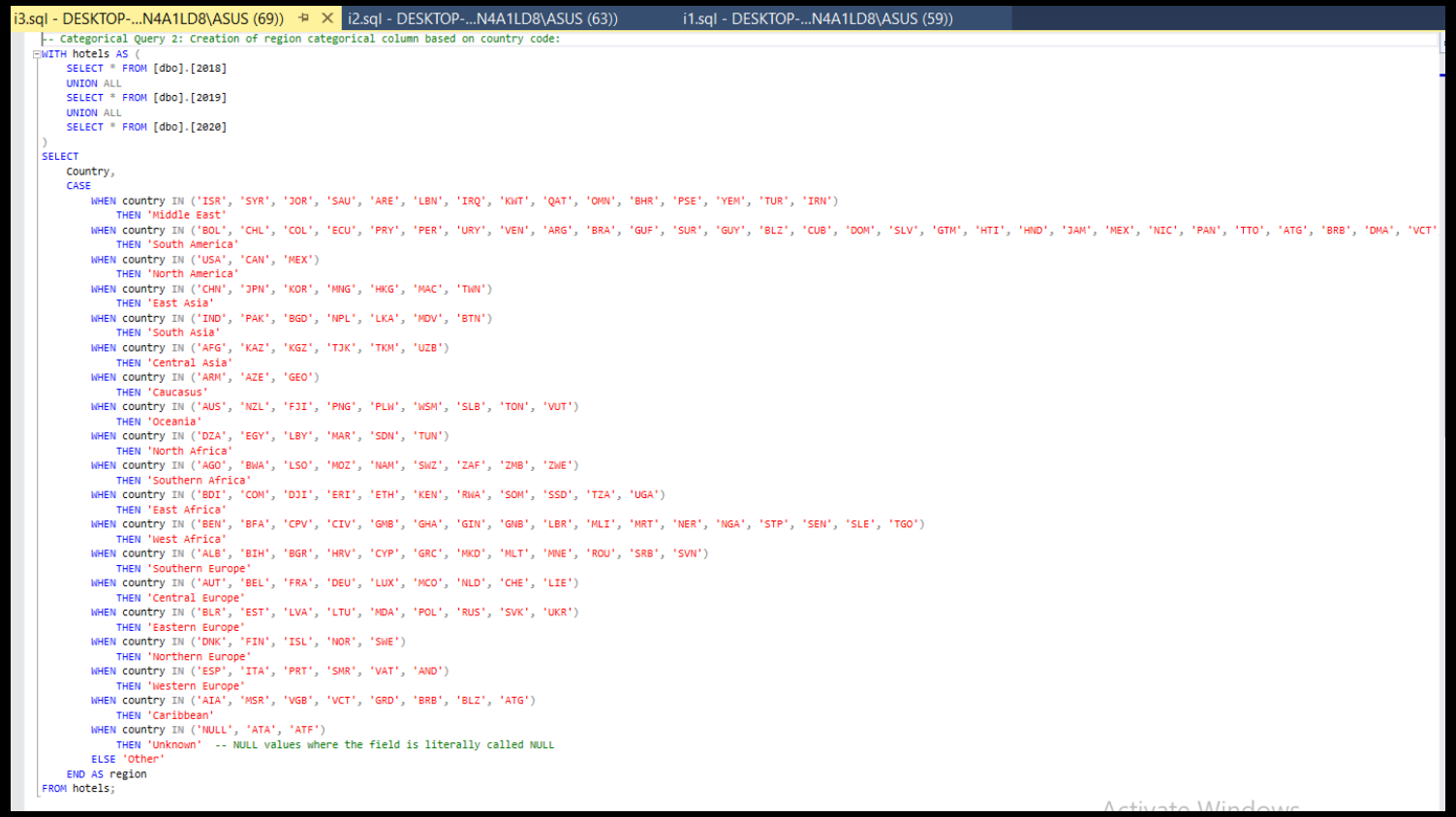
```sql
WITH hotels AS (
    SELECT * FROM [dbo].[combined data]

),
hotel_all_columns_with_guest_type AS (
    SELECT
        TRY_CAST(NULLIF(adults, 'NA') AS INT) AS adults,
        TRY_CAST(NULLIF(children, 'NA') AS INT) AS children,
        TRY_CAST(NULLIF(babies, 'NA') AS INT) AS babies,
        CASE
            WHEN TRY_CAST(NULLIF(adults, 'NA') AS INT) = 1 AND TRY_CAST(NULLIF(children, 'NA') AS INT) = 0 AND TRY_CAST(NULLIF(babies, 'NA') AS INT) = 0 THEN 'Single'
            WHEN TRY_CAST(NULLIF(adults, 'NA') AS INT) = 2 AND TRY_CAST(NULLIF(children, 'NA') AS INT) = 0 AND TRY_CAST(NULLIF(babies, 'NA') AS INT) = 0 THEN 'Couple'
            WHEN TRY_CAST(NULLIF(adults, 'NA') AS INT) > 2 AND TRY_CAST(NULLIF(children, 'NA') AS INT) = 0 AND TRY_CAST(NULLIF(babies, 'NA') AS INT) = 0 THEN 'Group'
            ELSE 'Family'
        END AS guest_type
    FROM hotels AS h
    LEFT JOIN [dbo].[market_segment] AS m ON h.market_segment = m.market_segment
    LEFT JOIN [dbo].[meal_cost] AS c ON h.meal = c.meal
)
SELECT *
FROM hotel_all_columns_with_guest_type
```

- One of the requirements was to investigate if there's any guest-related seasonality or trends Noticing that in the dataset, there are 'adults', 'children' & 'babies' information for each booking made
- I want to create a single column that categorizes each booking by the type of guests, they fall into 4 categories
    1. Single
    2. Couple
    3. Group
    4. Family

SQL Data Pre-Processing

- I observed that the dataset stores the guests' countries in a three-letter country code format.
- To gain deeper insights into customer behavior based on their geographical location, I plan to create a new categorical column called **'region'**.
- This will classify the guests into broader regions, allowing for a more comprehensive analysis of trends and patterns across different parts of the world.



SQL Data Pre-Processing

Given that the month of arrival of guests are provided in the dataset, I'd like to categorize them by seasons:

i4.sql - DESKTOP-...N4A1LD8\ASUS (70)) | i3.sql - DESKTOP-...N4A1LD8\ASUS (69)) | i2.sql - DESKTOP-...N4A1LD8\ASUS (63)) | i1.sql - DESKTOP-...N4A1LD8\ASUS (59))

```sql
-- Categorical Query 3: creation of seasons categorization code:
WITH hotels AS (
    SELECT * FROM [dbo].[2018]
    UNION ALL
    SELECT * FROM [dbo].[2019]
    UNION ALL
    SELECT * FROM [dbo].[2020]
),
hotel_all_columns_with_seasons AS (
    SELECT arrival_date_month,
           CASE
               WHEN arrival_date_month IN ('December', 'January', 'February') THEN 'Winter'
               WHEN arrival_date_month IN ('March', 'April', 'May') THEN 'Spring'
               WHEN arrival_date_month IN ('June', 'July', 'August') THEN 'Summer'
               WHEN arrival_date_month IN ('September', 'October', 'November') THEN 'Autumn'
           END AS seasons
    FROM hotels AS h
    LEFT JOIN [dbo].[market_segment] AS m ON h.market_segment = m.market_segment
    LEFT JOIN [dbo].[meal_cost] AS c ON h.meal = c.meal
)
SELECT *
FROM hotel_all_columns_with_seasons;
```

SQL Data Pre-Processing

The SQL query below utilizes DATEPART & CASE WHEN statements, as well as converting and concatenating each columns from the arrival date to extract out the corresponding day of the week in which the guests arrive in.

i5.sql - DESKTOP-...N4A1LD8\ASUS (68)) | i4.sql - DESKTOP-...N4A1LD8\ASUS (70)) | i3.sql - DESKTOP-...N4A1LD8\ASUS (69)) | i2.sql - DESKTOP-...N4A1LD8\ASUS (63))

```sql
-- Categorical Query 4: creation of day_of_week categorization code
WITH hotels AS (
    SELECT * FROM [dbo].[2018]
    UNION ALL
    SELECT * FROM [dbo].[2019]
    UNION ALL
    SELECT * FROM [dbo].[2020]
)
SELECT
    arrival_date_day_of_month,
    arrival_date_month,
    arrival_date_year,
    CONVERT(DATE, CONVERT(VARCHAR, arrival_date_day_of_month) + ' ' + arrival_date_month + ' ' + CONVERT(VARCHAR, arrival_date_year)) AS arrival_date,
    CASE DATEPART(dw, CONVERT(DATE, CONVERT(VARCHAR, arrival_date_day_of_month) + ' ' + arrival_date_month + ' ' + CONVERT(VARCHAR, arrival_date_year)))
        WHEN 1 THEN 'Sunday'
        WHEN 2 THEN 'Monday'
        WHEN 3 THEN 'Tuesday'
        WHEN 4 THEN 'Wednesday'
        WHEN 5 THEN 'Thursday'
        WHEN 6 THEN 'Friday'
        WHEN 7 THEN 'Saturday'
    END AS day_of_week
FROM hotels;
```

SQL Data Pre-Processing

i6.sql - DESKTOP-...N4A1LD8\ASUS (73)) | i5.sql - DESKTOP-...N4A1LD8\ASUS (68)) | i4.sql - DESKTOP-...N4A1LD8\ASUS (70)) | i3.sql - DESKTOP-...N4A1LD8\ASUS (69))

```sql
-- Categorical Query 5: creation of revenue categorization code:
WITH hotels AS (
    SELECT * FROM [dbo].[2018]
    UNION ALL
    SELECT * FROM [dbo].[2019]
    UNION ALL
    SELECT * FROM [dbo].[2020]
),
hotel_all_columns_with_revenue AS (
    SELECT
        reservation_status,
        CAST(adr AS FLOAT) AS adr_numeric, -- Convert ADR to numeric
        CAST(discount AS FLOAT) AS discount_numeric, -- Convert Discount to numeric
        CAST(c.Cost AS FLOAT) AS cost_numeric, -- Convert Cost to numeric
        CAST(stays_in_week_nights AS INT) + CAST(stays_in_weekend_nights AS INT) AS Number_of_Rooms_Sold, -- Ensure stays are integers
        CASE
            WHEN reservation_status = 'Canceled' THEN 0 -- Validated it's 0. No revenue generated from canceled transactions.
            WHEN CAST(adr AS FLOAT) < 0 THEN 0 -- ADR cannot be negative.
            WHEN (reservation_status = 'Check-Out' OR reservation_status = 'No-Show') AND (CAST(adr AS FLOAT) = 0 OR CAST(adr AS FLOAT) < 0) THEN 0
            ELSE
                (((CAST(stays_in_week_nights AS INT) + CAST(stays_in_weekend_nights AS INT)) *
                (CAST(adr AS FLOAT)) *
                (1 - CAST(discount AS FLOAT))) - CAST(c.Cost AS FLOAT))
        END AS revenue
    FROM hotels AS h
    LEFT JOIN [dbo].[market_segment] AS m ON h.market_segment = m.market_segment
    LEFT JOIN [dbo].[meal_cost] AS c ON h.meal = c.meal
)
SELECT *
FROM hotel_all_columns_with_revenue;
```

Since we're given the:

1. ADR value
2. # of weekdays and weekends stayed from a particular booking
3. % Discount corresponding for guests belong to a particular market_segment
4. Meal Cost of food provided to the guest

We can manipulate the ADR formula to obtain the corresponding revenue generated by each booking, this is given by:

$$Revenue\ per\ booking = [\times stays\ in\ week\ nights + stays\ in\ weekend\ nights \times ADR\ 1 - Discount\ ] - Meal\ Cos\text{?}$$

Hotel revenue data analysis

The model used in this task is a **Random Forest Regressor**, a powerful supervised machine learning algorithm designed for regression tasks.

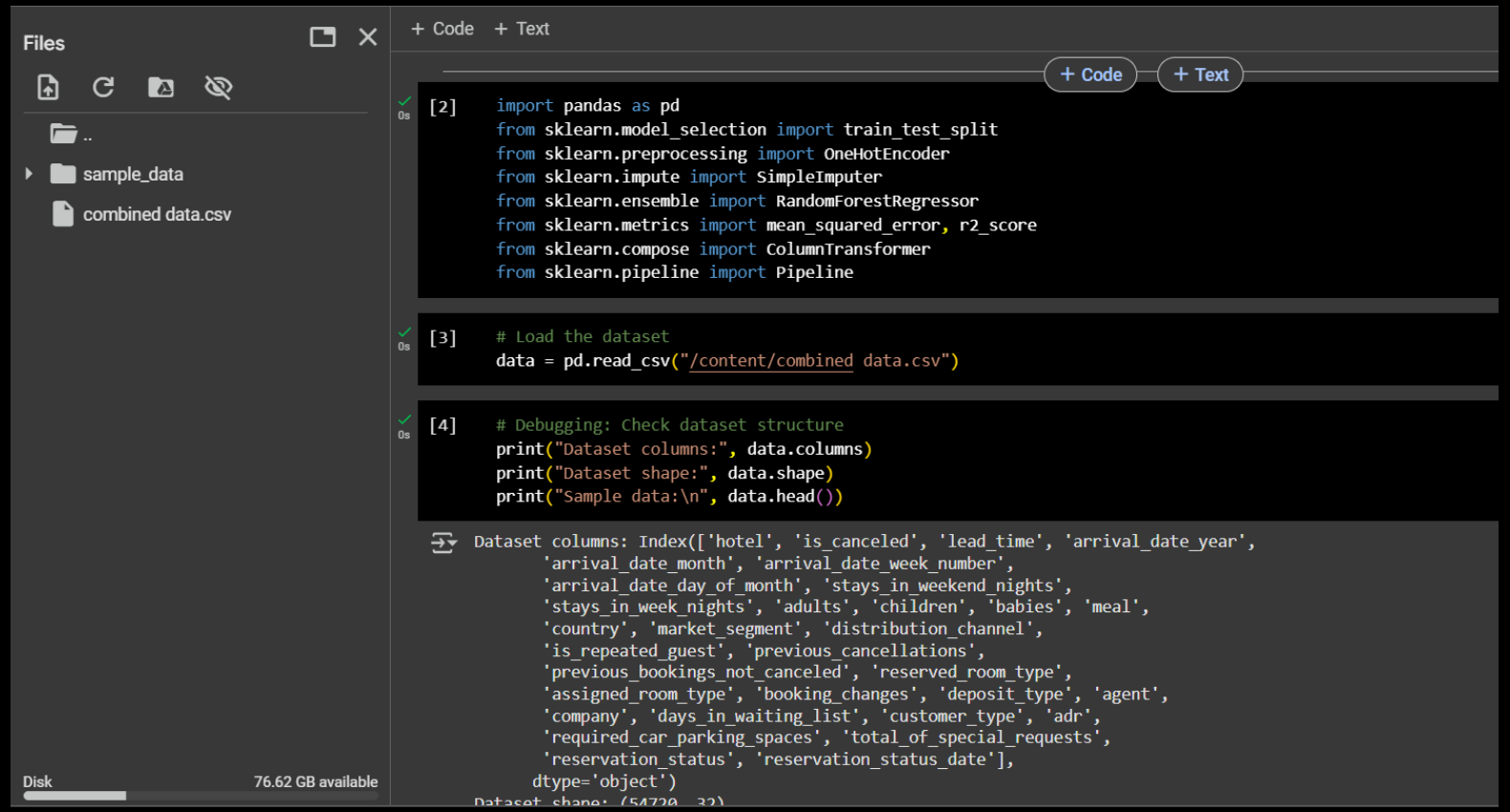
It operates by constructing multiple decision trees during the training phase and aggregating their predictions to produce a final output.

This ensemble method enhances accuracy by reducing variance, making it robust against overfitting and capable of handling complex, non-linear relationships in data.

In this pipeline, the Random Forest Regressor predicts the **Average Daily Rate (ADR)** of hotel bookings based on features such as total nights stayed, meal types, and customer demographics.

The model is integrated with preprocessing steps to handle missing values and encode categorical variables, ensuring smooth and efficient handling of the dataset's mixed data types.

Random Forest’s ability to manage noisy data, provide feature importance insights, and scale efficiently makes it an ideal choice for this task, delivering reliable and interpretable predictions for a critical business metric in the hospitality industry.



1. Importing Libraries

The code imports essential libraries:
- ***pandas***: For handling tabular data.
- ***sklearn.model_selection***: To split data into training and testing sets.
- ***sklearn.preprocessing***: For preprocessing tasks like encoding categorical data.
- ***sklearn.impute***: To handle missing values.
- ***sklearn.ensemble***: To use the Random Forest Regressor.
- ***sklearn.metrics***: To evaluate model performance using metrics like Mean Squared Error (MSE) and R^2 score.
- ***sklearn.compose***: For combining preprocessing steps.
- ***sklearn.pipeline***: To build a streamlined machine learning workflow.

```
# Create new features and drop unnecessary columns
data['total_nights'] = data['stays_in_weekend_nights'] + data['stays_in_week_nights']
data.drop(['reservation_status', 'reservation_status_date', 'arrival_date_month',
           'arrival_date_year', 'arrival_date_week_number',
           'arrival_date_day_of_month', 'stays_in_weekend_nights',
           'stays_in_week_nights'], axis=1, inplace=True)
```

```
# Debugging: Check 'adr' column
print("'adr' in dataset columns:", 'adr' in data.columns)
print("Number of missing values in 'adr':", data['adr'].isnull().sum())
```

```
'adr' in dataset columns: True
Number of missing values in 'adr': 0
```

```
# Identify numerical and categorical columns
numerical_cols = data.select_dtypes(include=['int64', 'float64']).columns
categorical_cols = data.select_dtypes(include=['object']).columns
```

```
# Exclude 'adr' from numerical_cols
numerical_cols = data.select_dtypes(include=['int64', 'float64']).columns.drop('adr', errors='ignore')
categorical_cols = data.select_dtypes(include=['object']).columns
```

```
# Preprocessing for numerical and categorical data
num_preprocessor = SimpleImputer(strategy='mean')
cat_preprocessor = Pipeline(steps=[
    ('imputer', SimpleImputer(strategy='most_frequent')),
    ('onehot', OneHotEncoder(handle_unknown='ignore'))
])
```

```
# Column transformer
preprocessor = ColumnTransformer(
    transformers=[
        ('num', num_preprocessor, numerical_cols),
        ('cat', cat_preprocessor, categorical_cols)
    ])
```

```
# Define the pipeline
pipeline = Pipeline(steps=[
    ('preprocessor', preprocessor),
    ('model', RandomForestRegressor(random_state=42))
])
```

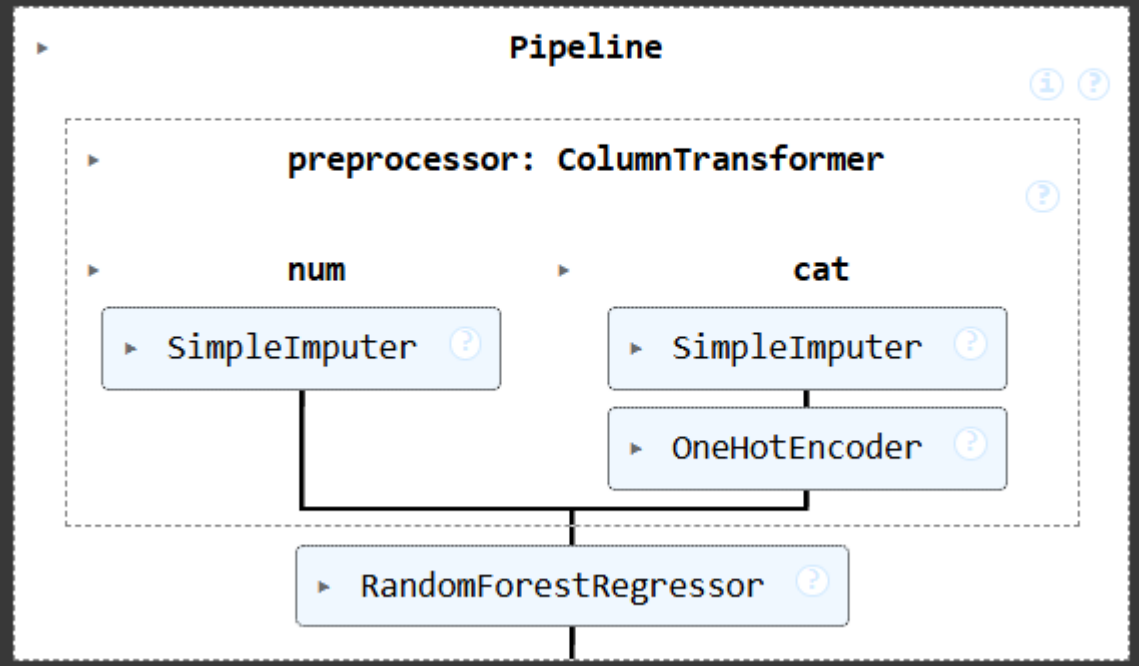
```
# Define features and target
X = data.drop(columns=['adr'])  # Features
y = data['adr']                 # Target (Average Daily Rate)
```

```
# Debugging: Check feature and target shapes
print("Shape of feature matrix (X):", X.shape)
print("Shape of target vector (y):", y.shape)
```

```
Shape of feature matrix (X): (54720, 24)
Shape of target vector (y): (54720,)
```

```
# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
[ ]  # Train the pipeline
     pipeline.fit(X_train, y_train)
```




```
[ ]  # Make predictions
     y_pred = pipeline.predict(X_test)
```

```
     # Evaluate the model
     mse = mean_squared_error(y_test, y_pred)
     r2 = r2_score(y_test, y_pred)
```

```
[ ]  print(f"\nModel Performance:")
     print(f"Mean Squared Error (MSE): {mse:.2f}")
     print(f"R-squared (R2): {r2:.2f}")
```

```
Model Performance:
Mean Squared Error (MSE): 1084.89
R-squared (R2): 0.55
```

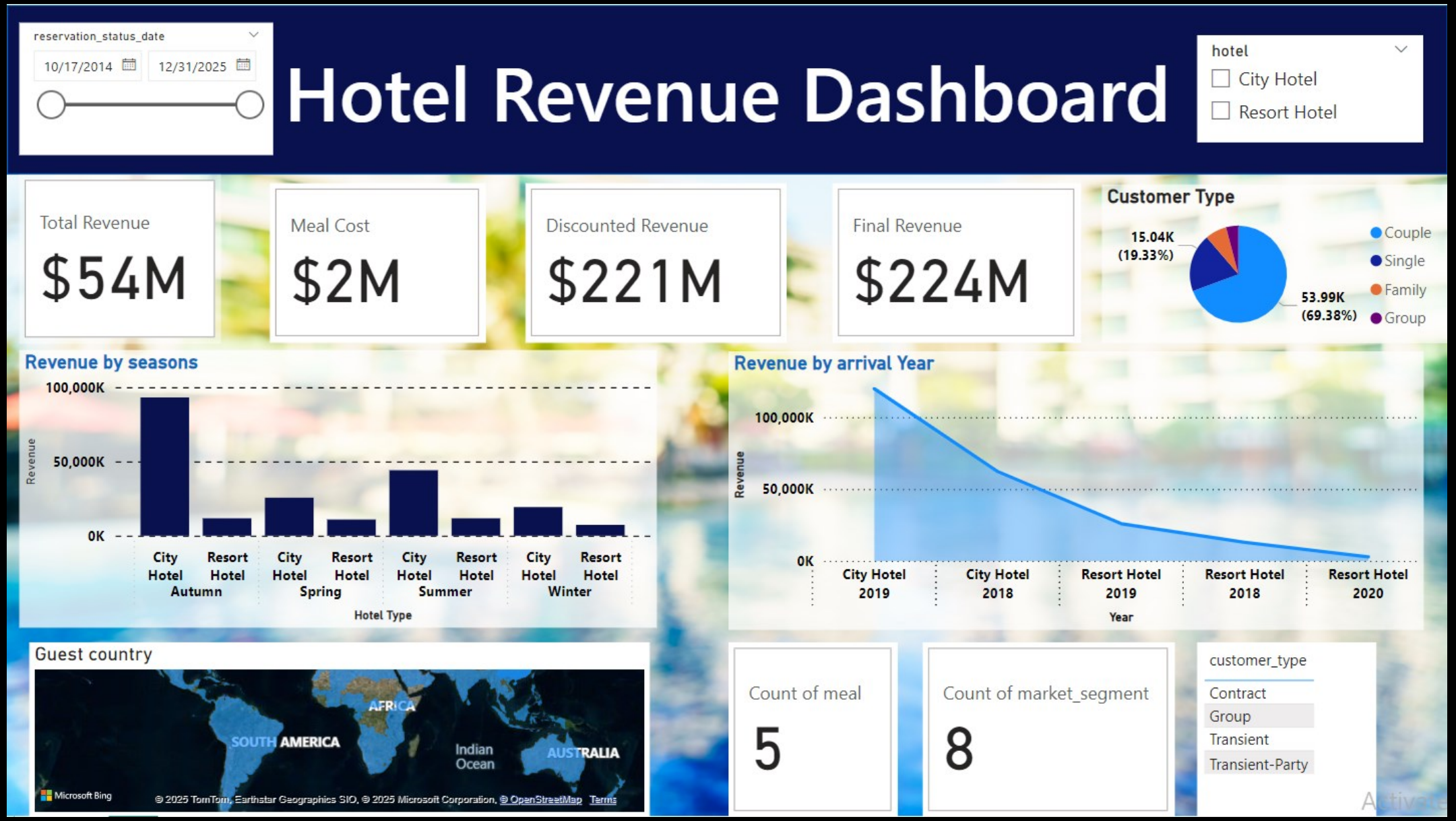
•**Mean Squared Error (MSE)**: Measures the average squared difference between predicted and actual values. Lower values indicate better performance.

MSE = 1084.89.

R-squared (R^2): Indicates the proportion of variance in the target variable explained by the model. Higher values (closer to 1) indicate better performance.

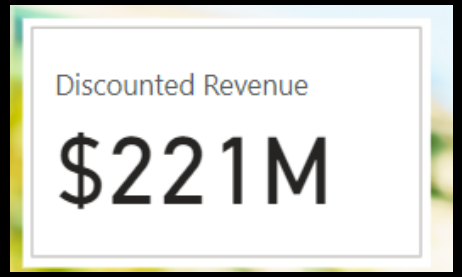
$R^2 = 0.55$ (indicates 55% of the variance in the data is explained by the model).

Hotel Revenue Dashboard




```
Total Revenue =
SUMX(
    'combined data (2)',
    'combined data (2)'[adr] *
    ('combined data (2)'[stays_in_weekend_nights] + 'combined data (2)'[stays_in_week_nights]) *
    ('combined data (2)'[adults] + COALESCE('combined data (2)'[children], 0))
)
```

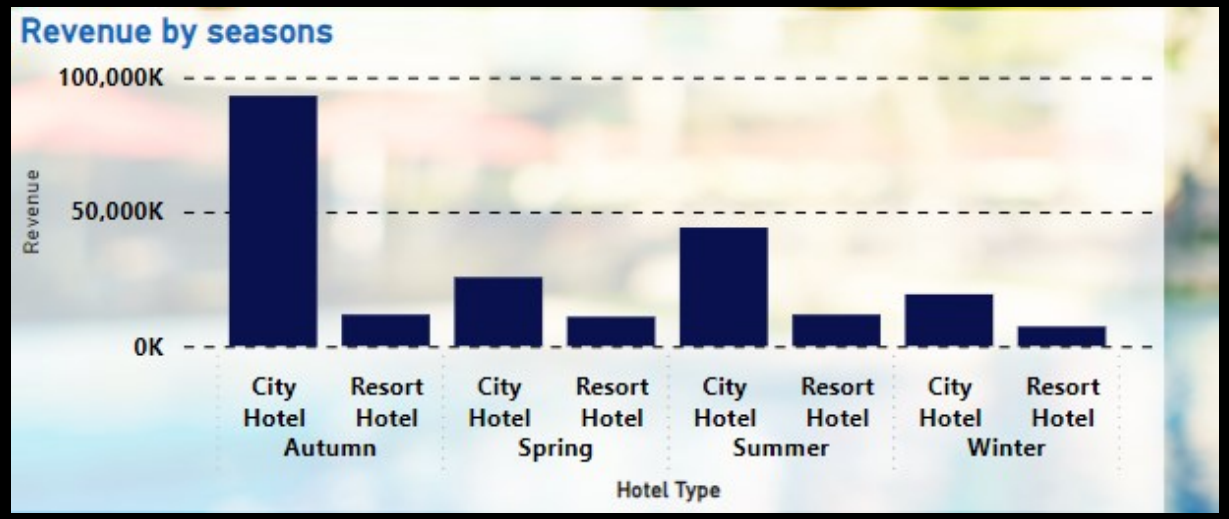

```
Meal Cost =
SUMX(
    'combined data (2)',
    COALESCE(RELATED('meal_cost'[Cost]), 0) *
    ('combined data (2)'[adults] + COALESCE('combined data (2)'[children], 0))
)
```




```
Discounted Revenue =
SUMX(
    'combined data (2)',
    [Total Revenue] *
    (1 - COALESCE(RELATED('market_segment'[Discount]), 0))
)
```

Power BI Data Pre-Processing

Revenue by Seasons



Autumn:
- **City Hotels: Generate the highest revenue compared to all other seasons and hotel types, with a noticeable spike above $100K.**
- **Resort Hotels: Revenue is significantly lower in Autumn compared to City Hotels.**

Spring:
- **Both City and Resort Hotels generate relatively low revenue, with a slight edge for Resort Hotels.**

Summer:
- **City Hotels again lead in revenue, with Resort Hotels closely following. This season shows better performance than Spring and Winter for both hotel types.**

Winter:
- **Revenue is lowest for both City and Resort Hotels, with Resort Hotels performing marginally better than City Hotels.**

Predictions Based on the Data

•City Hotels Dominate in Autumn:

•The spike in City Hotel revenue during Autumn suggests higher demand in urban areas during this time. This could be linked to business travel, conferences, or events commonly scheduled in Autumn.

•**Prediction:** This trend is likely to continue unless there's a major change in market conditions, as the seasonal preference for City Hotels in Autumn seems consistent.

•Resort Hotels Perform Well in Summer:

•Summer months align with vacation periods, making it a strong season for leisure-oriented Resort Hotels.

•**Prediction:** Resort Hotels will likely see consistent demand in summer seasons, particularly if they target families or tourists during holidays.

•Low Demand in Winter:

•Both hotel types experience their lowest revenue in Winter, possibly due to reduced travel during the off-peak season or harsh weather conditions.

•**Prediction:** Winter will continue to be a weak season unless promotional campaigns or events drive more bookings.

•Growth Potential for Spring:

•Spring shows modest revenue but could grow with targeted marketing (e.g., focusing on festivals, spring break tourism, or outdoor events).

•**Prediction:** Efforts to attract customers during Spring (e.g., discounted packages, seasonal offerings) might slightly boost revenue.

Revenue by Year

•**City Hotels:**
•**Peak in 2019:** Revenue for City Hotels reached its highest point in 2019, indicating strong demand for urban accommodations that year.
•**Decline After 2019:** After the peak in 2019, there is a sharp decline in revenue for City Hotels in subsequent years. This could be attributed to external factors like reduced business travel, global economic shifts, or the impact of the COVID-19 pandemic (2020).

•**Resort Hotels:**
•**Steady Decline (2018–2020):** Resort Hotels show a consistent decline in revenue over the years. Unlike City Hotels, there's no evident peak, suggesting stagnation or a lack of growth in demand for Resort Hotels.
•**Lower Revenue Overall:** Resort Hotels have consistently generated less revenue compared to City Hotels, which could indicate a smaller target market or less efficient operations.

Predictions Based on Trends:

City Hotels:

1. If the decline seen post-2019 is tied to temporary factors (e.g., pandemic-related disruptions), revenue might recover as travel normalizes.
2. However, long-term trends like increased remote work or virtual meetings might limit growth in business travel, potentially capping revenue recovery for City Hotels.

Resort Hotels:

1. Resort Hotels could see a slower recovery due to their steady decline and competition from alternative accommodations (e.g., vacation rentals).
2. They might experience a rebound if leisure travel and "staycation" trends gain popularity, especially in post-pandemic years.

Revenue Levels (2021+):

1. Without interventions, both hotel types are likely to experience slow revenue growth or even stagnation in the short term.
2. Predictive models using external factors (e.g., tourism statistics, economic recovery rates) could provide more clarity on when and how revenue might recover.

Actionable Recommendations

City Hotels

1. Leverage their dominance in Autumn and Summer by optimizing pricing and promotional packages.
2. Focus on business travelers and events to recover lost revenue since 2019.

Resort Hotels

1. Capitalize on Summer demand by offering family and group vacation packages.
2. Address the steady decline in revenue by diversifying offerings and targeting untapped customer segments.

Seasonal Strategies

1. Boost Winter and Spring performance through targeted discounts, bundled packages, and events (e.g., winter getaways, spring festivals).

Customer Segmentation

1. Focus on retaining and growing the Family customer base, which accounts for the majority of revenue.
2. Develop campaigns for Couples and Groups to diversify the customer mix.

Geographic Focus

1. Identify top-performing countries and regions from the map visualization and develop marketing strategies to attract guests from these areas.

Conclusion

- The dashboard reveals that City Hotels outperform Resort Hotels, with peak revenue in Autumn, while Resort Hotels excel in Summer but face steady decline.
- Families are the largest customer segment, while Groups offer growth potential.
- Revenue dropped significantly post-2019, likely due to global disruptions.
- Winter and Spring are low-performing seasons, presenting opportunities for targeted promotions.
- To drive growth, hotels should focus on seasonal strategies, expand offerings, and tailor packages to customer segments.

Thank you