EDA

Hotel Booking Dataset

using Python

by

Nikita Rajpurohit

Table of content:-

Introduction:

This project involves performing EDA on a hotel booking dataset to identify patterns, trends, and key insights. The analysis focuses on understanding booking behavior, cancellations, customer preferences, and other important factors that influence hotel operations. Visualizations and statistical summaries are used to make the findings clear and actionable.

The dataset consists of detailed hotel booking records from two types of properties: city hotels and resort hotels. It includes information about reservations, customer profiles, booking timelines, duration of stay, meal preferences, and cancellations. The data spans multiple years, enabling analysis of seasonal trends, customer behavior, and overall booking patterns.

Problem Statement:

Understanding booking patterns and customer behavior is crucial for the hospitality industry to improve revenue and reduce cancellations. This analysis aims to explore hotel booking data to identify trends, cancellation reasons, and customer preferences, providing insights that can help in better decision-making and operational planning.

Data Description:

1. Booking Information

- **Hotel** – Type of hotel (City or Resort).
- **Is_cancelled** – Whether the booking was canceled (1) or not (0).
- **Lead Time** – Days between booking date and arrival.
- **Arrival Year** – Year of arrival.
- **Arrival Month** – Month of arrival.
- **Arrival Week Number** – Week number of arrival date.
- **Reservation Status** – Final booking status (Checked-Out, Canceled, No-Show).
- **Reservation Status Date** – Date when status was last updated.

2. Customer Details

- **Weekend Nights** – Number of weekend nights stayed.
- **Week Nights** – Number of weekday nights stayed.
- **Adults** – Number of adults in the booking.
- **Children** – Number of children in the booking.
- **Babies** – Number of babies in the booking.
- **Parking Spaces Required** – Number of parking spaces requested.

3. Booking Behavior

- **Repeated Guest** – Indicates if the guest has booked before.
- **Previous Cancellations** – Number of past canceled bookings.
- **Previous Non-Canceled Bookings** – Number of successful past bookings.
- **Customer Type** – Category of customer (Transient, Group, etc.).
- **Booking Changes** – Number of modifications made to the booking.
- **Special Requests** – Number of extra requests made by the customer.

4. Preferences and Financial Details

- **Meal** – Meal plan booked (BB, HB, FB, etc.).
- **Market Segment** – Category based on booking channel (Online, Offline).
- **Distribution Channel** – Source of booking (e.g., TA/TO, Direct).
- **Reserved Room Type** – Initially booked room type.
- **Assigned Room Type** – Room type assigned at check-in.
- **Deposit Type** – Type of deposit made.
- **Agent ID** – Booking agent ID.
- **Company ID** – Corporate account ID.
- **Average Daily Rate (ADR)** – Price per night of stay.
- **Waiting List Days** – Days on the waiting list.

Exploratory Data Analysis:

- Analyzed the distribution of bookings across hotel types (City Hotel vs Resort Hotel).
- Examined monthly and yearly booking trends to identify seasonality.
- Investigated cancellation patterns and calculated cancellation rates.
- Studied lead time distribution to understand booking behavior.
- Compared customer preferences for meals and market segments.
- Calculated retention rate to measure repeat customer bookings.
- Visualized relationships between stay duration, special requests, and booking outcomes.

Data Wrangling:

Data cleaning and pre-processing steps performed before analysis:

1. Data Quality Checks

- Checked **total number of records** and dataset dimensions.

```
[255]: num_rows = len(df.index)
       print(num_rows)
```
```
119390
```

- Verified **data types** of all columns.
- Identified and **removed duplicate records** to ensure uniqueness.

```
[256]: df.drop_duplicates(inplace=True)

[257]: num_rows = len(df.index)#Count number of rows
       print(num_rows)
```
```
87396
```

- Checked for **missing values** and handled them appropriately.

```
: df['Total_people']=df['Total_people'].replace(0,'No people')

: df.drop(df[df['Total_people']=='No people'].index,inplace=True)

: num_rows = len(df.index)#Count number of rows
  print(num_rows)
```
```
87226
```

2. Feature Engineering

- **Merged year, month, and day** into a single `Arrival Date` column for better date handling.

```
[261]: df['Arrival_date']=pd.to_datetime(df['arrival_date_year'].astype(str) + '-' +
                                         df['arrival_date_month'].astype(str) + '-' +
                                         df['arrival_date_day_of_month'].astype(str))#merging date,month and year columns
```

- **Removed week number column** as it was redundant.
- Combined **weekend nights** and **week nights** into a new column `Total Stay` for total nights stayed.

```
[262]: df['Total_stay']=df['stays_in_weekend_nights']+df['stays_in_week_nights']
```

- Dropped the original **weekend nights** and **week nights** columns after merging.

```
[265]: df.drop(['stays_in_weekend_nights','stays_in_week_nights'],axis=1,inplace=True)
```

- Combined **adults, children, and babies** into a single `Total People` column for guest count.

```
7]: df['Total_people']=df['adults']+df['children']+df['babies']
```

- Created **new index column** as dataset lacked a unique identifier.

```
df['booking_id']=range(1, len(df) + 1)
```

3. Data Cleaning

- Removed **invalid or erroneous values** from the `children` column (e.g., negative values).

```
[266]: missing_values=df['children'].isnull().sum()
       print(missing_values)
       df.dropna(subset=['children'],inplace=True)

       4
```

- Standardized **categorical values** for consistency.
- Replaced **missing country values** with "Other"

```
df.fillna({'country':'Other'},inplace=True)
```

- Filled **agent missing values** as "Direct Booking"

```
[274]: df.fillna({'agent':'Direct Booking'},inplace=True)

[275]: print(df['agent'].head())

       0    Direct Booking
       1    Direct Booking
       2    Direct Booking
       3             304.0
       4             240.0
       Name: agent, dtype: object
```

- Removed **company column** due to excessive missing values.
- Ensured **reserved room type** = **assigned room type** where applicable.

```
]: df['is_satisfied']=df['reserved_room_type']==df['assigned_room_type']

]: df['is_satisfied']=df['is_satisfied'].replace(True,'Satisfied')
   df['is_satisfied']=df['is_satisfied'].replace(False,'Not satisfied')
```

- Removed unnecessary or redundant columns after transformations.

KPIs:

1. **Total Bookings** – Total number of reservations made.

```
[178]: Total_Bookings=df['booking_id'].count()
       print('Total Bookings=',Total_Bookings)

       Total Bookings= 87226
```

2. **Bookings by Hotel Type** – Distribution between City Hotel and Resort Hotel.

```
[186]: hotel_counts = df['hotel'].value_counts()
       print(hotel_counts)

       hotel
       City Hotel      53270
       Resort Hotel    33956
       Name: count, dtype: int64
```

3. **Average Lead Time-lead time** - The number of days between customer books their room to arriving in hotel.

```
[367]: Avg_lead_time=df['lead_time'].median()
       print('Average Lead Time =',round(Avg_lead_time),'days')

       Average Lead Time = 49 days
```

4. **Overall Cancellation Rate (%)** – Percentage of bookings canceled.

```
total_cancellations=df['is_canceled'].mean()*100
print('Total cancellations=',round(total_cancellations,2),'%')

Total cancellations= 27.52 %
```

1. **Retention Rate (%)** – Percentage of repeated guests.

```
]:
Total_guests = len(df)
repeated_guests = df['is_repeated_guest'].sum()

retention_rate = (repeated_guests / total_guests) * 100
print("Retention Rate=",round(retention_rate,2),'%')
```

```
Retention Rate= 3.86 %
```

2. **Average Stay Duration** – Average total nights per booking.

```
:
Avg_stay=df['Total_stay'].mean()
print('Average Stay =',round(Avg_stay),'days ')
```

```
Average Stay = 4 days
```

3. **Average Number of Guests per Booking**

```
:
Avg_guests=df['Total_people'].mean()
print('Average count of guests=',round(Avg_guests))
```

```
Average count of guests= 2
```

4. **Average Daily Rate (ADR)** – Average price per night.

```
:
Avg_adr=df['adr'].mean()
print("Average Adr=",round(Avg_adr,2))
```

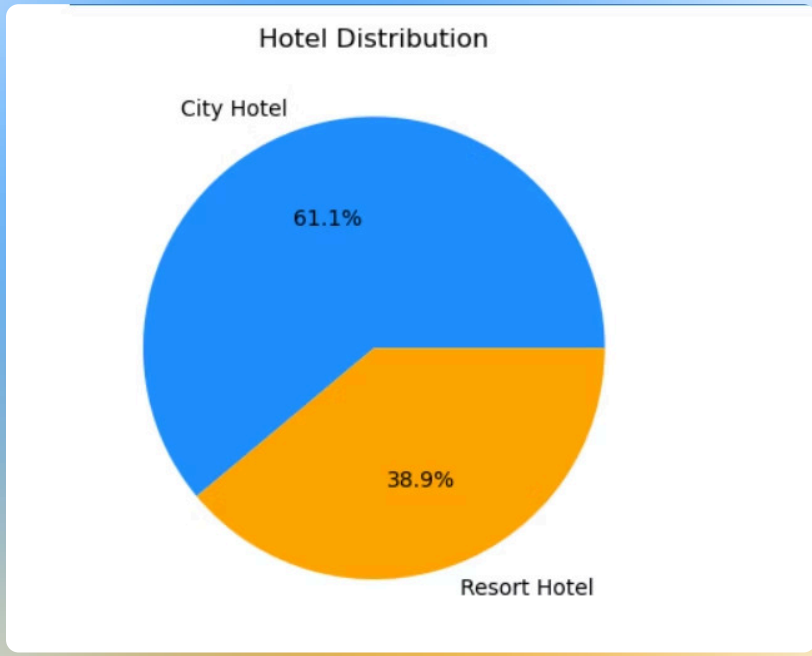
```
Average Adr= 106.52
```

Visualizations:

Hotel Distribution:

The visualization shows the **number of bookings by hotel type**. It is clear that **City Hotels receive a significantly higher number of bookings compared to Resort Hotels**.

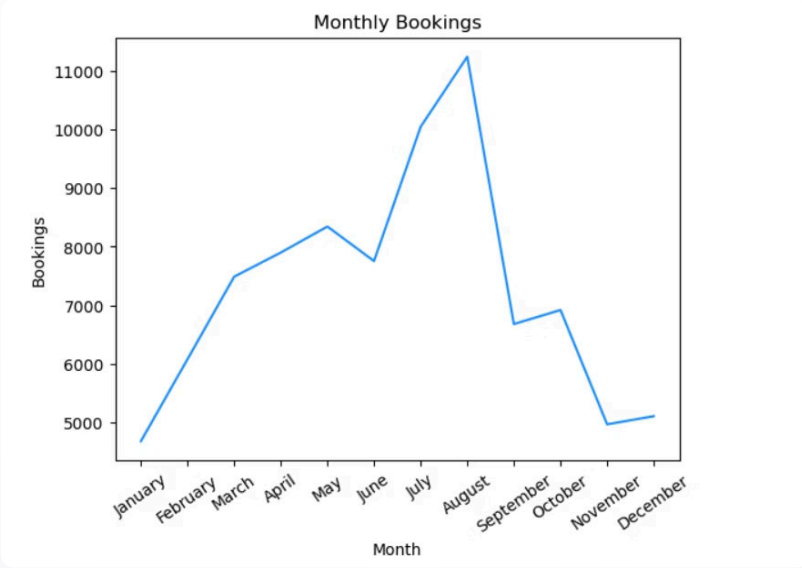
- City Hotels dominate the booking share, indicating strong demand in urban locations.
- Resort Hotels have relatively lower bookings, suggesting potential areas for improvement.
- Seasonal demand could play a role, as resorts often rely on holidays and vacation periods.
- Enhancing customer experience in resorts (packages, leisure activities, family deals) can also increase demand.



Monthly Booking:

The analysis of monthly bookings shows clear **seasonal trends**, with some months experiencing significantly higher demand than others.

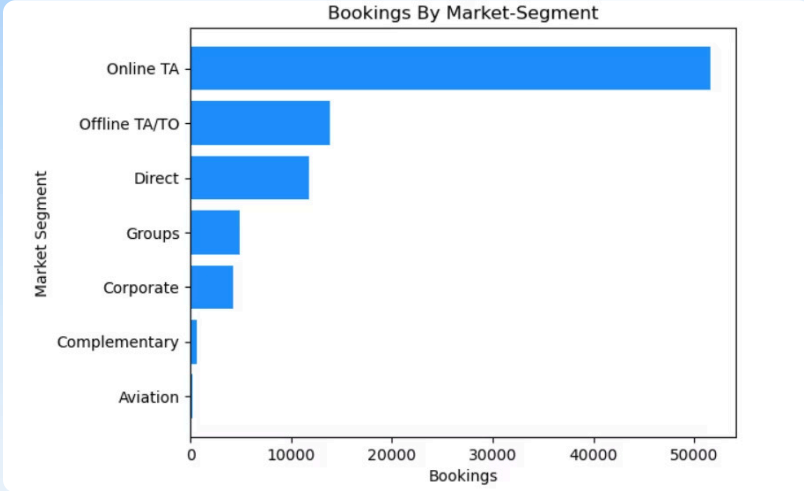
- Identify peak seasons to optimize revenue through dynamic pricing and premium packages.
- Use early-bird offers and promotions to secure bookings ahead of high-demand periods.
- Target deals, special discounts to balance bookings in off-peak months.
- Align staffing and resources with expected demand to maintain service quality.



Market Segment:

The analysis shows that **Online Travel Agents (TA)** contribute the largest share of bookings. OTAs (like MakemyTrip.com, GoIbibo, etc.) act as **third-party platforms** where customers compare hotels and make reservations. While they help attract a large volume of customers, they also charge **high commissions**, which reduces the hotel's profit margin.

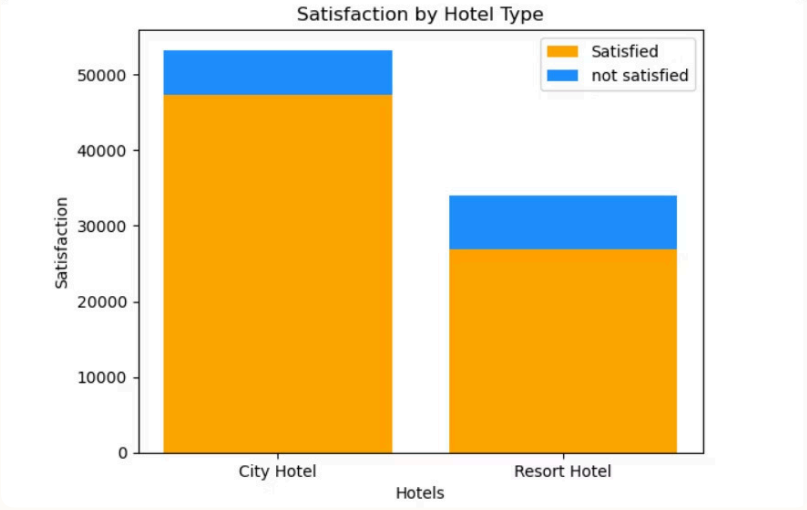
- Promote **direct bookings** through hotel websites and apps by offering incentives such as discounts, free add-ons, or loyalty points.
- Improve **digital presence to compete with Online TAs.**
- Build **customer loyalty programs** to encourage repeat bookings directly with the hotel.



Customer Satisfaction:

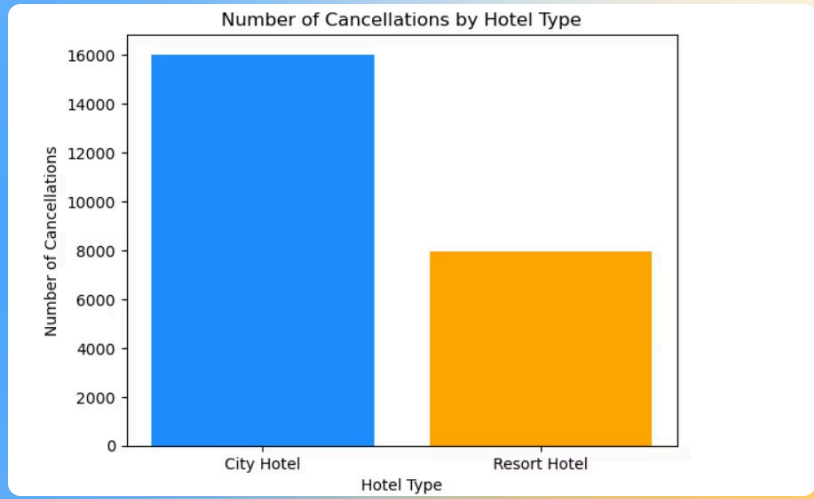
The analysis compares satisfaction levels between **City Hotels** and **Resort Hotels**. Results show that City Hotels receive higher satisfaction rates, while Resort Hotels lag behind.Customer satisfaction was estimated by checking whether the **reserved room type matched the assigned room type**.

- City Hotels benefit from **better accessibility, business-friendly services, and consistent quality**.
- Resort Hotels may face challenges due to **seasonality, service gaps, or facility limitations**.
- To improve satisfaction in resorts, hotels can focus on **enhancing leisure facilities, personalized services, and seasonal offers**.
- Strengthening **guest feedback systems** can help identify pain points and improve service quality.



Cancellations by hotel Type:

The analysis shows that **cancellations are higher in City Hotels compared to Resort Hotels**. Cancellations occur due to reasons such as **flexible travel plans, last-minute changes, long lead times, and seasonal uncertainty**. These cancellations directly impact hotels by causing **loss of revenue, poor resource allocation, and reduced occupancy rates**.



1.City Hotels

- Cause: Frequent among **business travelers** who often reschedule or cancel at short notice.
- Effect: Leads to **sudden drops in occupancy** and unused inventory.
- Solutions:
  - Implement **stricter last-minute cancellation policies**.
  - Offer **non-refundable but discounted rates** to secure bookings.
  - Allow **flexible re-booking** instead of outright cancellations.

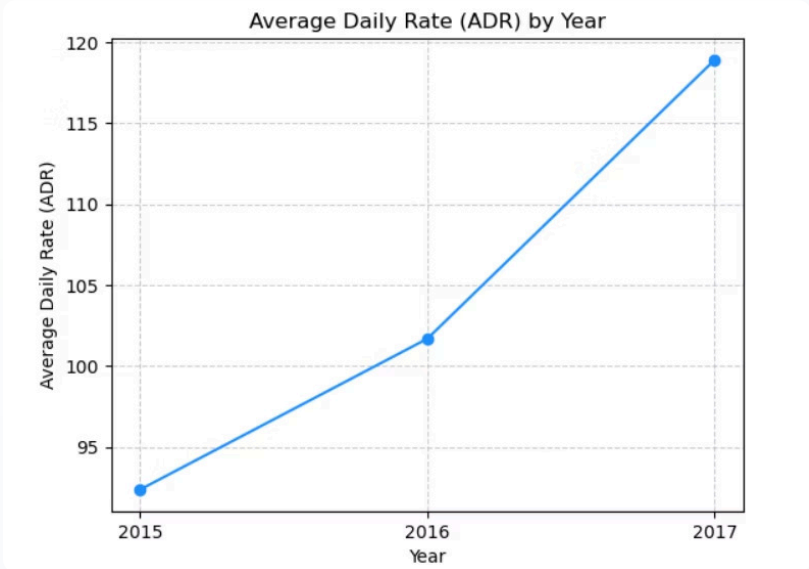
2.Resort Hotels

- Cause: Often linked to **long lead times, seasonal demand, and weather-related uncertainties**.
- Effect: Results in **higher vacancy during off-season** and disrupted forecasting.
- Solutions:
  - Introduce **early discounts** to lock in advance bookings.
  - Promote **seasonal packages with partial refund policies**.
  - Strengthen **guest engagement** with reminders and pre-arrival offers.

Average Daily Rate(ADR):

The analysis shows that the **Average Daily Rate (ADR) has been increasing year by year**. This reflects a positive pricing strategy and growing revenue potential. However, maintaining this trend requires consistent improvements in customer experience, demand management, and operational efficiency.

- **Flexible Pricing** – Adjust room prices based on demand, season, and booking time.
- **Segment Based Pricing** – Offer special prices for families, business travelers, groups, or long stays.
- **Extra Services** – Promote add-ons like better rooms, meals, spa, or activities to increase earnings.
- **Encourage Direct Bookings** – Give discounts or perks to customers who book through the hotel's own website/app.
- **Special Offers in Slow Months** – Create packages to attract guests during low-demand periods.
- **Reward Repeat Customers** – Use loyalty points, discounts, or perks to keep guests coming back.



Conclusion:

This project explored hotel booking data to uncover trends, customer behavior, and key performance indicators. Through data cleaning, visualization, and analysis, several insights were identified that highlight both strengths and areas for improvement in hotel operations.

- The analysis highlights key patterns in **hotel bookings, cancellations, customer preferences, and pricing (ADR)**.
- **City Hotels attract more bookings**, while **Resort Hotels need improvement** through better packages and promotions.
- **Cancellations remain a challenge**, especially in City Hotels, affecting revenue and occupancy.
- **Seasonality plays a big role** – peak months bring high demand, while low months require targeted offers.
- **ADR is steadily increasing**, showing positive growth, but further improvement is possible through better pricing strategies and direct bookings.
- Focusing on **customer satisfaction, loyalty programs, and balanced booking channels** will help hotels grow sustainably.