**PYTHON FOR DATA SCIENCE**

**Assignment Report**

**on**

**Hotel Booking Analysis**

**By**

**CHETHAN S**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Content** | **Page Number** |
| 1 | **Problem Statement** | 3 |
| 2 | **Introduction** | 3 |
| 3 | **Dataset** | 4-5 |
| 4 | **Implementation** | 6-8 |
| 5 | **Screenshots** | 9-15 |
| 6 | **Conclusion** | 16 |
| 7 | **Code** | 17-19 |

1. **PROBLEM STATEMENT:**

**Objective:**

To develop a predictive model that can accurately forecast hotel booking cancellations using historical booking data.

**Background:**

Hotel booking cancellations can significantly impact the operational and financial planning of hotels. Understanding and predicting cancellations can help hotels optimize their booking strategies, improve customer satisfaction, and increase revenue. By analyzing patterns in historical booking data, we aim to identify key factors that influence booking cancellations.

**Goal:**

The goal is to build a machine learning model that predicts whether a booking will be canceled based on the provided features. By doing so, hotels can take proactive measures to manage bookings and reduce cancellation rates.

1. **INTRODUCTION:**

This hotel booking analysis constitutes an in-depth investigation of a dataset spanning two years, starting from 2015 to 2017. Through a series of steps, including data preprocessing, descriptive statistics, and examining the distribution of guests from various countries, we aim to unearth valuable insights in the hospitality industry. Additionally, the comparison between "assigned" and "reserved" room types will provide insights into different booking practices. In this analysis, we will also identify the most productive market segments in terms of bookings and analyze the average price per night across all segments. Furthermore, by examining guest arrival patterns, we can understand the time dynamics that impact hotel operations. Lastly, by conducting a test of guest arrival distribution, we will undergo a crucial phase in testing the validity of our data. All of this will aid us in making better decisions in managing and marketing this hotel.

In tourism and travel related industries, most of the research on Revenue Management demand forecasting and prediction problems employ data from the aviation industry, in the format known as the Passenger Name Record (PNR). This is a format developed by the aviation industry. However, the remaining tourism and travel industries like hospitality, cruising, theme parks, etc., have different requirements and particularities that cannot be fully explored without industry’s specific data. Hence, two hotel datasets with demand data are shared to help in overcoming this limitation.

The datasets now made available were collected aiming at the development of prediction models to classify a hotel booking’s likelihood to be cancelled. Nevertheless, due to the characteristics of the variables included in these datasets, their use goes beyond this cancellation prediction problem. One of the most important properties in data for prediction models is not to promote leakage of future information. In order to prevent this from happening, the timestamp of the target variable must occur after the input variables’ timestamp. Thus, instead of directly extracting variables from the bookings database table, when available, the variables’ values were extracted from the bookings change log, with a timestamp relative to the day prior to arrival date (for all the bookings created before their arrival date). Not all variables in these datasets come from the bookings or change log database tables. Some come from other tables, and some are engineered from different variables from different tables.

1. **DATASET DESCRIPTION:**

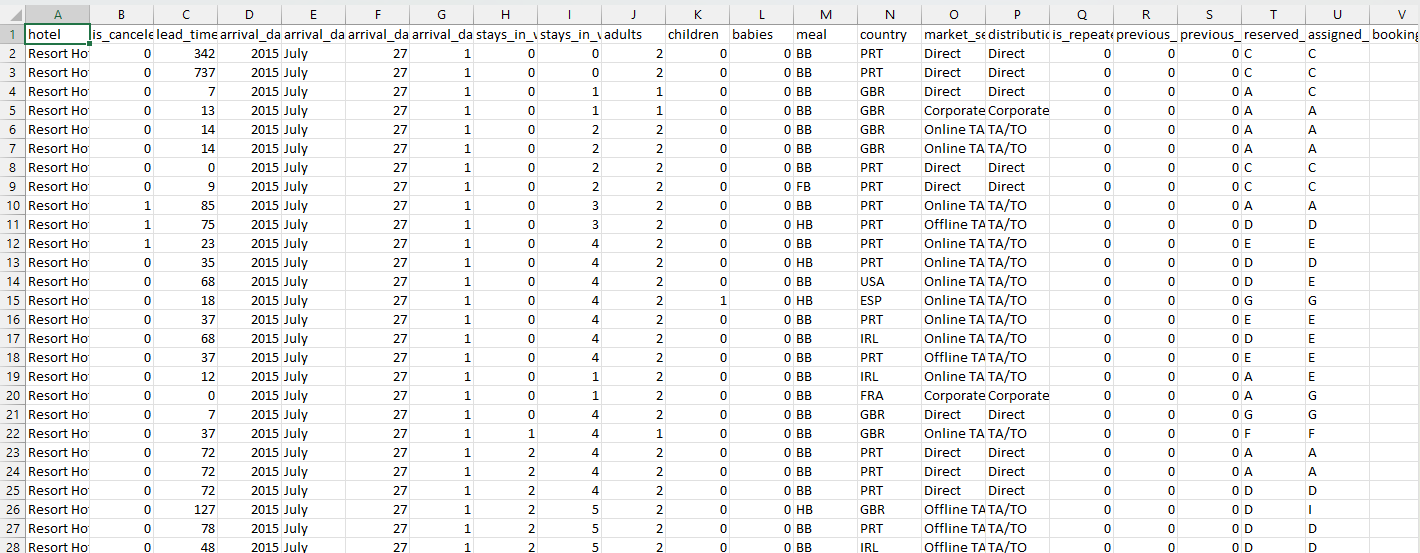
The dataset contains various features related to hotel bookings, including:

* hotel: The type of hotel (e.g., Resort Hotel or City Hotel).
* is\_canceled: Indicates whether the booking was canceled (1) or not (0).
* lead\_time: The number of days between the booking date and the arrival date.
* arrival\_date\_year: The year of arrival.
* arrival\_date\_month: The month of arrival.
* arrival\_date\_week\_number: The week number of the arrival date.
* arrival\_date\_day\_of\_month: The day of the month of arrival.
* stays\_in\_weekend\_nights: Number of weekend nights (Saturday and Sunday) the guest stayed or booked to stay.
* stays\_in\_week\_nights: Number of week nights (Monday to Friday) the guest stayed or booked to stay.
* adults: Number of adults.
* children: Number of children.
* babies: Number of babies.
* meal: Type of meal booked (e.g., BB - Bed & Breakfast, HB - Half board, FB - Full board, SC - Self-catering).
* country: Country of origin of the guest.
* market\_segment: Market segment designation (e.g., Direct, Corporate, Online TA).
* distribution\_channel: Booking distribution channel (e.g., TA/TO - Travel Agents/ Tour Operators, Direct, Corporate).
* is\_repeated\_guest: Indicates if the booking was from a repeated guest (1) or not (0).
* previous\_cancellations: Number of previous bookings that were canceled by the customer prior to the current booking.
* previous\_bookings\_not\_canceled: Number of previous bookings not canceled by the customer prior to the current booking.
* reserved\_room\_type: Code of room type reserved.
* assigned\_room\_type: Code of room type assigned.
* booking\_changes: Number of changes/amendments made to the booking from the moment the booking was created.
* deposit\_type: Type of deposit made for the booking (e.g., No Deposit, Non Refundable, Refundable).
* agent: ID of the travel agent who made the booking.
* company: ID of the company/entity that made the booking.
* days\_in\_waiting\_list: Number of days the booking was on the waiting list before it was confirmed to the customer.
* customer\_type: Type of booking (e.g., Contract, Group, Transient, Transient-Party).
* adr\*\*: Average Daily Rate, calculated by dividing the sum of all lodging transactions by the total number of staying nights.
* required\_car\_parking\_spaces: Number of car parking spaces required by the customer.
* total\_of\_special\_requests: Total number of special requests made by the customer (e.g., high floor, crib).
* reservation\_status: Reservation status (e.g., Canceled, Check-Out, No-Show).
* reservation\_status\_date: Date at which the last status was set.

These features provide comprehensive information about each hotel booking, including guest details, booking specifics, and stay information.

This dataset provides a comprehensive overview of the booking details, guest preferences, and stay characteristics, enabling a thorough analysis for various purposes such as predicting cancellations, understanding guest satisfaction, and optimizing revenue management.

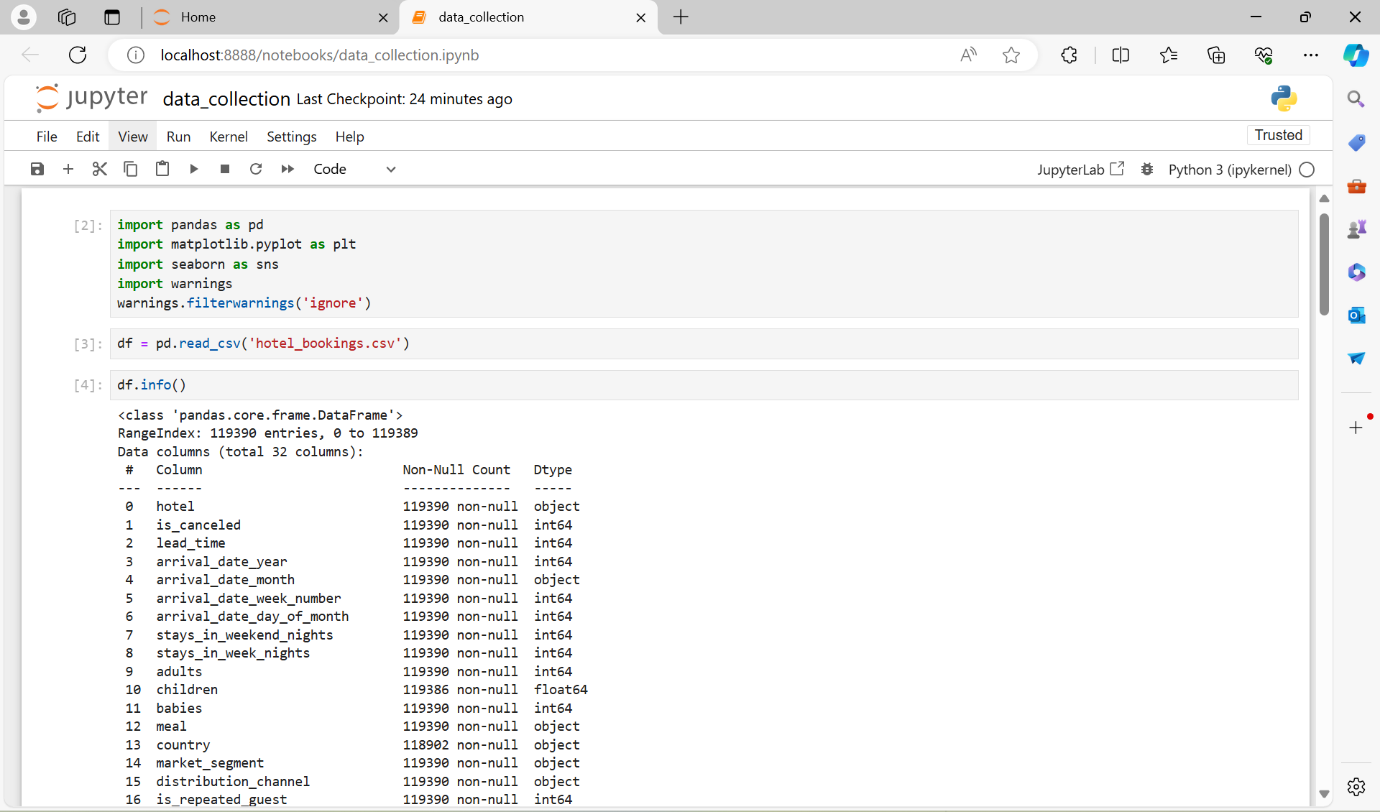
**SCREENSHOT OF DATASET:**

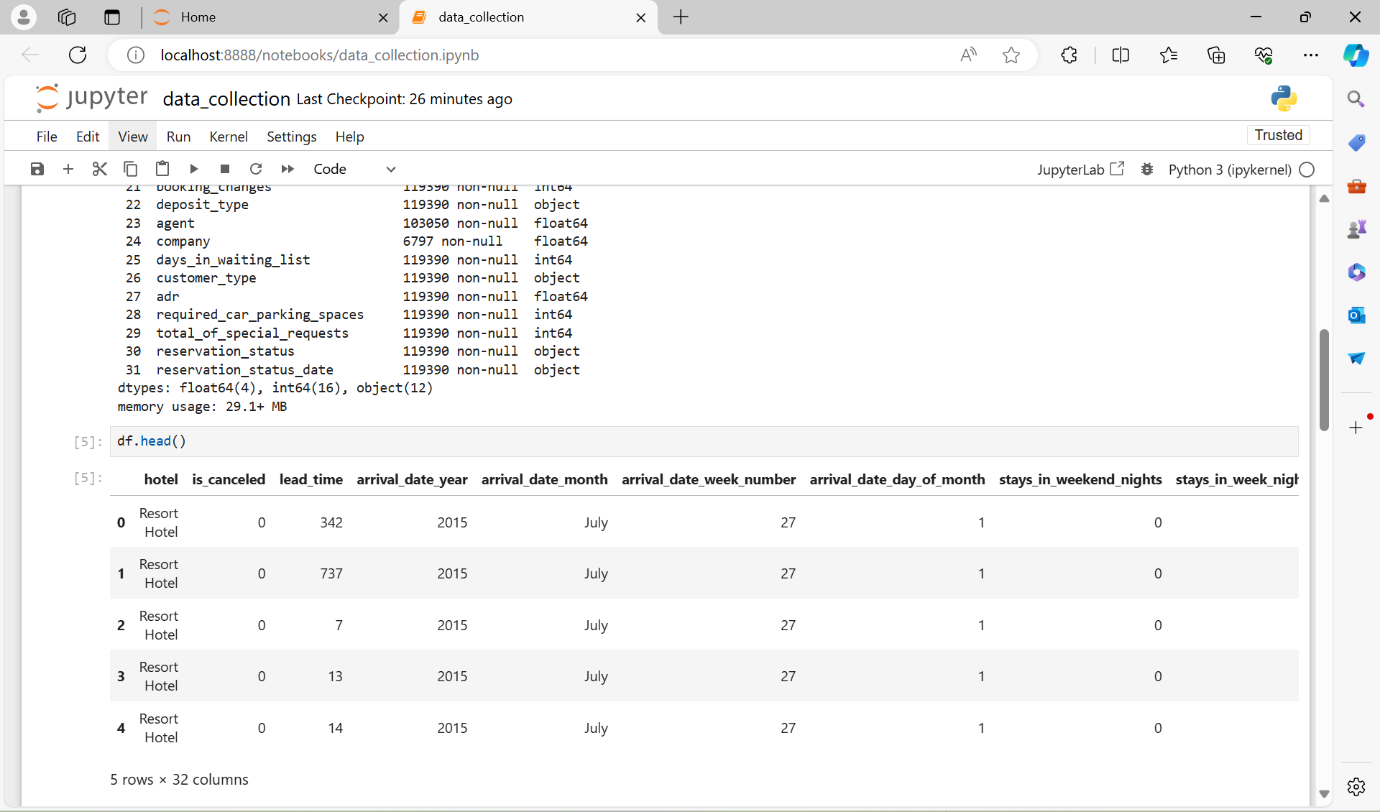


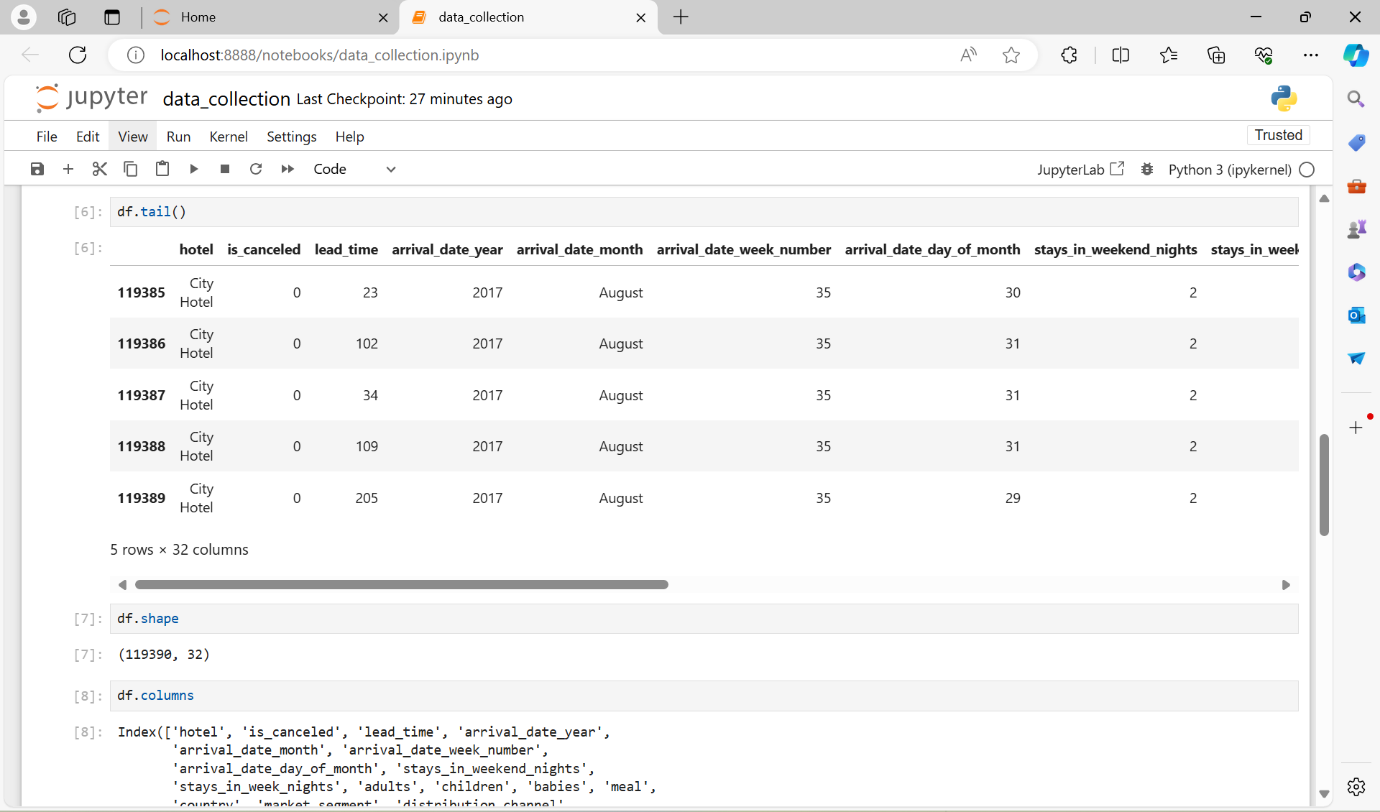
1. **IMPLEMENTATION:**
2. **Data Collection:**
3. Identify Data Sources:
   * Determine where the data is located and how it can be accessed.
   * Possible sources include:
   * Property Management Systems (PMS)
   * Online booking platforms (e.g., Booking.com, Expedia)
   * Customer Relationship Management (CRM) systems
   * Point of Sale (POS) systems
   * Guest feedback tools and surveys
4. Collect Relevant Data:
   * Gather data relevant to the analysis objectives.
   * Ensure data includes key features such as:
   * Booking details (e.g., `is\_canceled`, `lead\_time`)
   * Guest demographics (e.g., `adults`, `children`, `country`)
   * Stay details (e.g., `stays\_in\_weekend\_nights`, `stays\_in\_week\_nights`)
   * Transaction details (e.g., `adr`, `total\_of\_special\_requests`)
5. Data Quality Check:
   * Perform an initial assessment of data quality to identify any issues or inconsistencies.
   * Verify data integrity, completeness, and accuracy.
6. **Data Preprocessing:**
7. Handle Missing Values:
   * Identify missing values in the dataset and decide on a strategy to handle them.
   * Options include:
   * Dropping rows or columns with missing values if they are not significant.
   * Imputing missing values using techniques like mean, median, mode, or predictive modeling.
8. Correct Data Types:
   * Ensure that data types are appropriate for each feature.
   * Convert categorical variables to categorical data types.
   * Convert numerical variables to numeric data types.
9. Handle Outliers:
   * Identify outliers in numerical features using statistical methods.
   * Decide whether to remove outliers, cap/floor them, or transform them using appropriate techniques (e.g., log transformation).
10. Standardize Categorical Values:
    * Standardize categorical values to ensure consistency and remove unnecessary variations.
    * Perform operations such as lowercase conversion, removing leading/trailing whitespaces, and grouping similar categories.
11. Feature Engineering:
    * Create new features that may provide additional insights or improve model performance.
    * Examples include:
    * Calculating total guests (`adults` + `children` + `babies`)
    * Deriving stay duration (`stays\_in\_weekend\_nights` + `stays\_in\_week\_nights`)
12. Encoding Categorical Variables:
    * Convert categorical variables into a numerical representation suitable for machine learning algorithms.
    * Options include one-hot encoding or label encoding.
13. Normalize Numerical Features:
    * Normalize numerical features to ensure they are on a comparable scale.
    * Common techniques include Min-Max scaling or Standardization.
14. Feature Selection:
    * Select relevant features for analysis and modeling to reduce dimensionality and improve model performance.
    * Consider using techniques like correlation analysis, feature importance, or domain knowledge.
15. Split the Dataset:
    * Split the dataset into training and testing sets for model evaluation.
    * Optionally, set aside a validation set for hyperparameter tuning.
16. **Data Analysis:**
17. Descriptive Statistics:
    * Compute summary statistics (mean, median, standard deviation) for numerical features.
    * Count the frequency of categories for categorical features.
18. Univariate Analysis:
    * Use histograms to understand the distribution.
    * Use bar charts to see the distribution of categories.
19. Bivariate Analysis:
    * Use scatter plots to explore relationships (e.g., `lead\_time` vs. `adr`)
    * Use box plots to compare distributions across categories (e.g., `adr` by `hotel` type).
20. Multivariate Analysis:
    * Compute and visualize correlations between multiple features using a heatmap.
21. Time Series Analysis:
    * Analyze trends and seasonality in booking data (e.g., bookings over time, monthly patterns).
22. Segmentation Analysis:
    * Group data by certain features (e.g., `market\_segment`, `customer\_type`) and analyze each segment’s characteristics (e.g., average ADR, cancellation rates).
23. Hypothesis Testing:
    * Formulate and test hypotheses (e.g., "Does `lead\_time` affect cancellation rates?").
    * Use statistical tests to validate hypotheses (e.g., t-tests, chi-square tests).
24. Predictive Analysis:
    * Build predictive models if needed (e.g., predicting cancellations).
    * Train and evaluate models using appropriate metrics (e.g., accuracy, precision, recall).

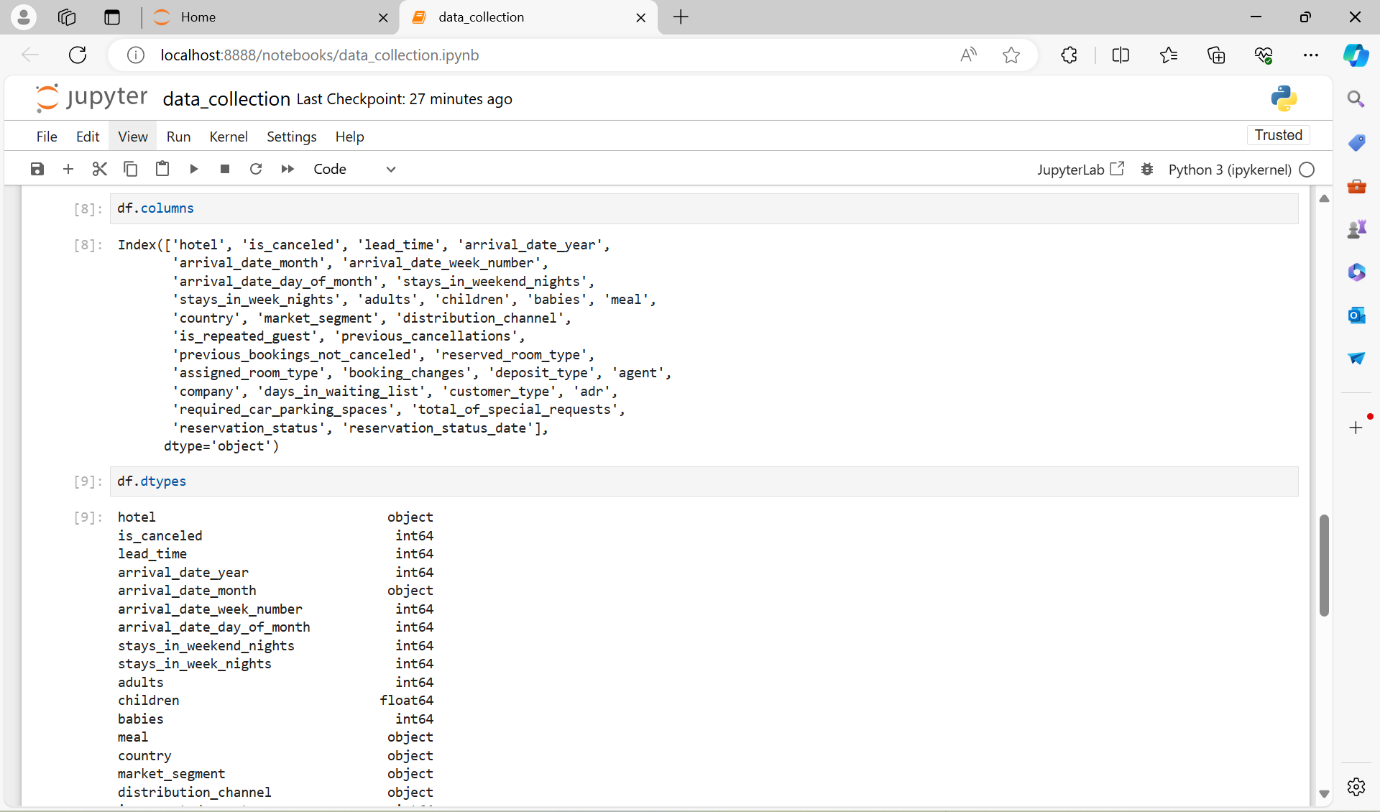
Questions to answer from the data:

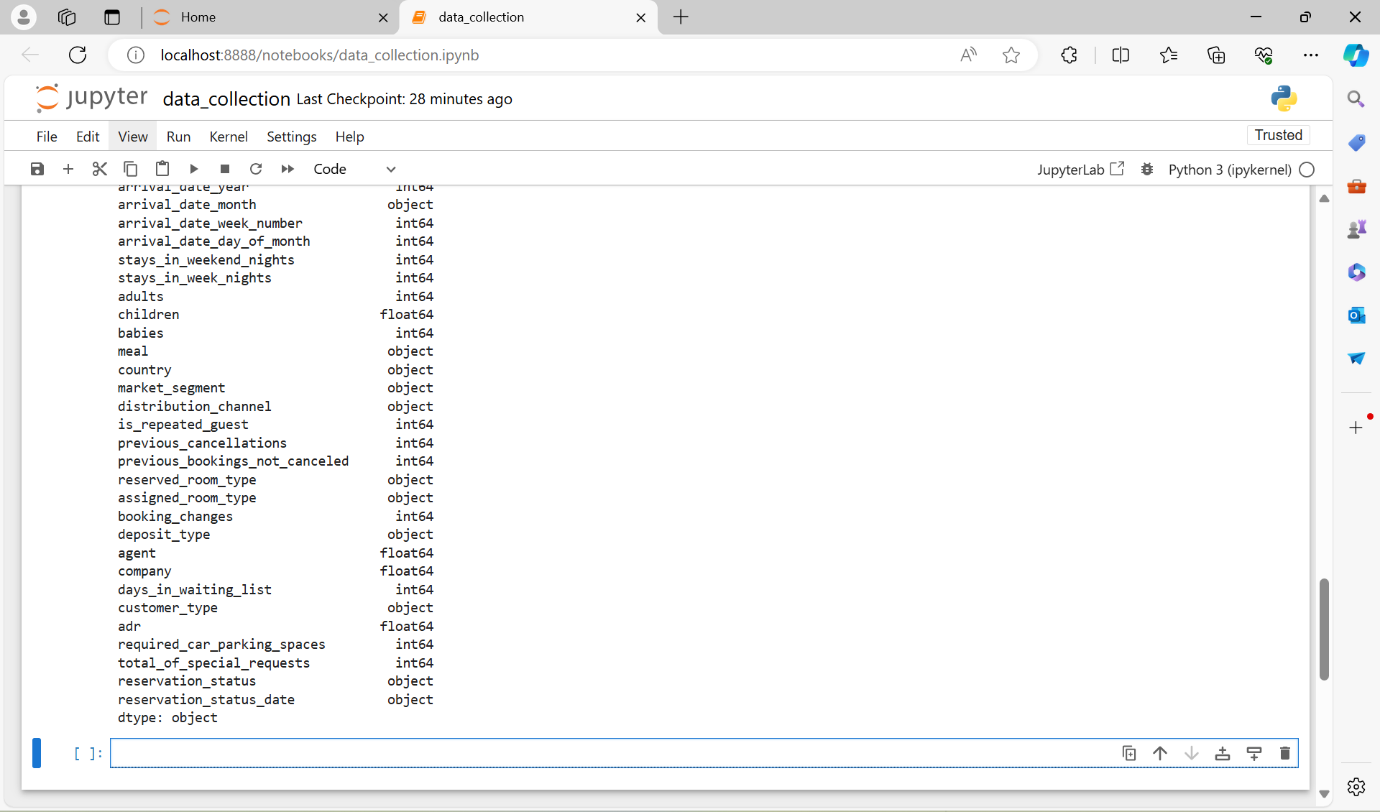
1. Impact of special requests on ADR
2. Booking and Stay Patterns
3. Impact of booking changes on reservation status
4. Scatter plot of lead\_time vs adr, colored by is\_canceled.
5. Lead Time Distribution
6. Calculate the distribution of canceled vs not canceled bookings
7. **Data Modelling**
8. Define the Problem:
   * Clearly define the problem. For example, predicting hotel booking cancellations.
9. Select Features and Target Variable:
   * Choose the features (input variables) that will be used for modeling and the target variable (output) you want to predict.
10. Split Data into Training and Testing Sets:
    * Divide the dataset into training and testing sets to evaluate the model's performance.
11. Choose a Model:
    * Select appropriate machine learning algorithms based on the problem type (e.g., classification, regression).
12. Train the Model:
    * Use the training data to train your model.
13. Evaluate the Model:
    * Evaluate the model's performance using the testing data and metrics appropriate for the problem (e.g., accuracy, precision, recall for classification; RMSE, MAE for regression).
14. Hyperparameter Tuning:
    * Optimize model performance by tuning hyperparameters using techniques like grid search or random search.
15. Validate the Model:
    * Use cross-validation to ensure the model generalizes well to unseen data.
16. Finalize the Model:
    * Once the model is validated, finalize it for deployment.
17. **SCREENSHOTS**

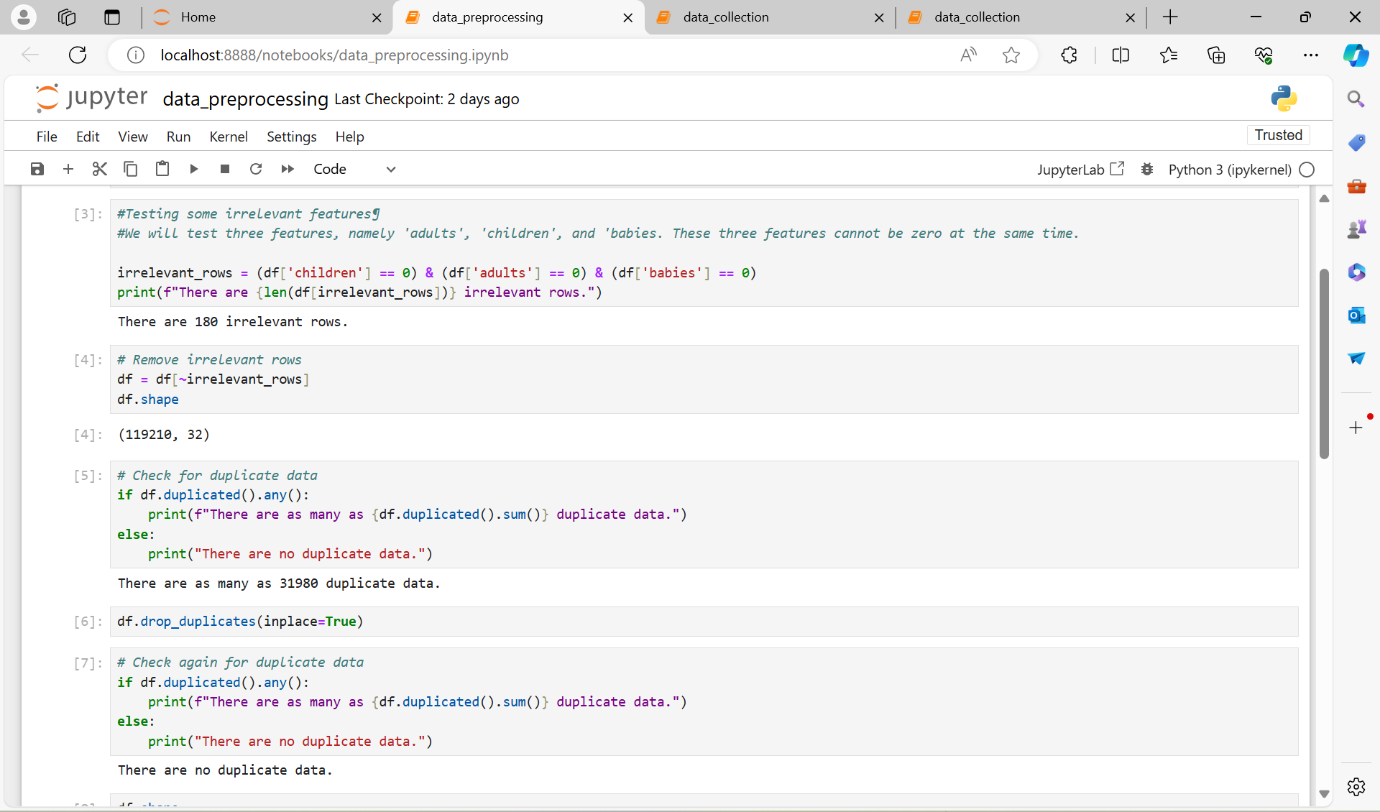


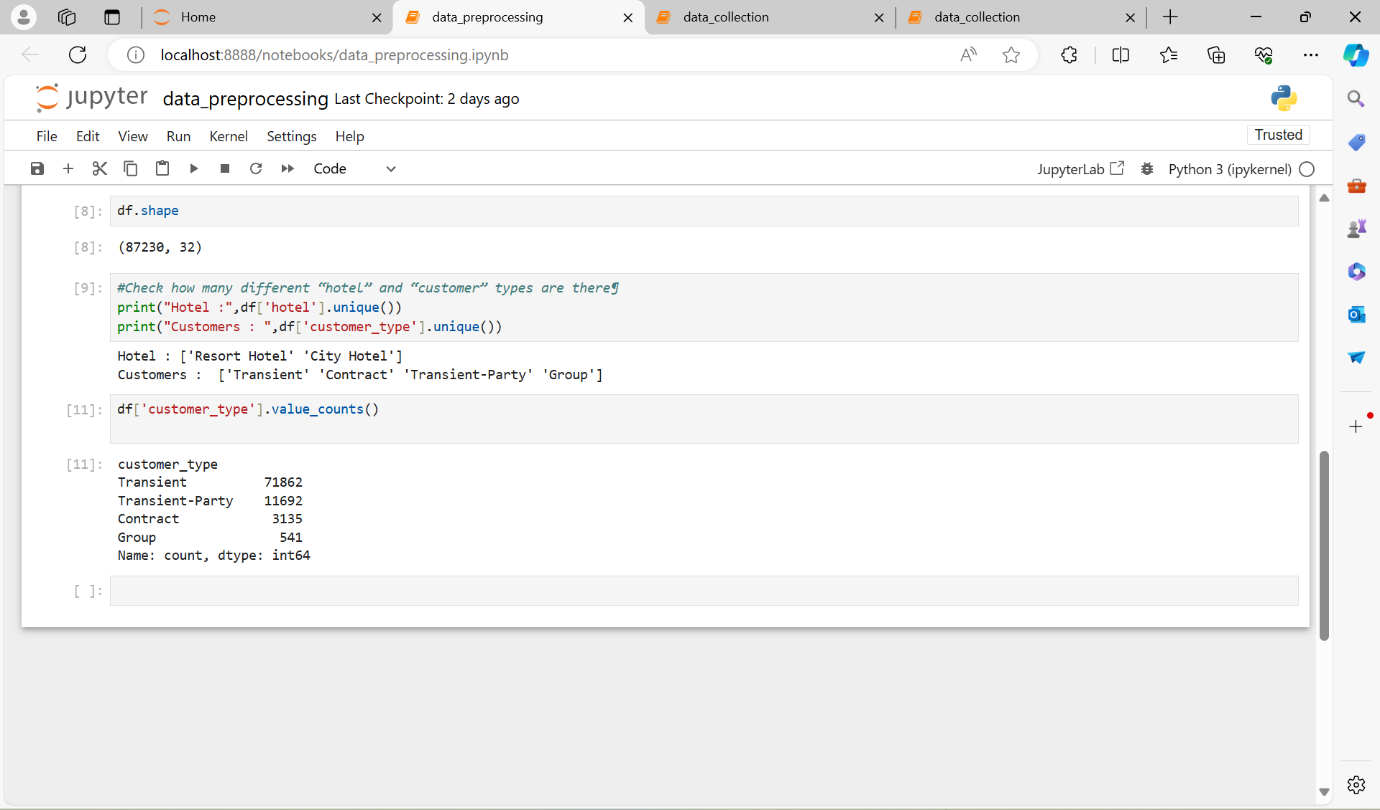




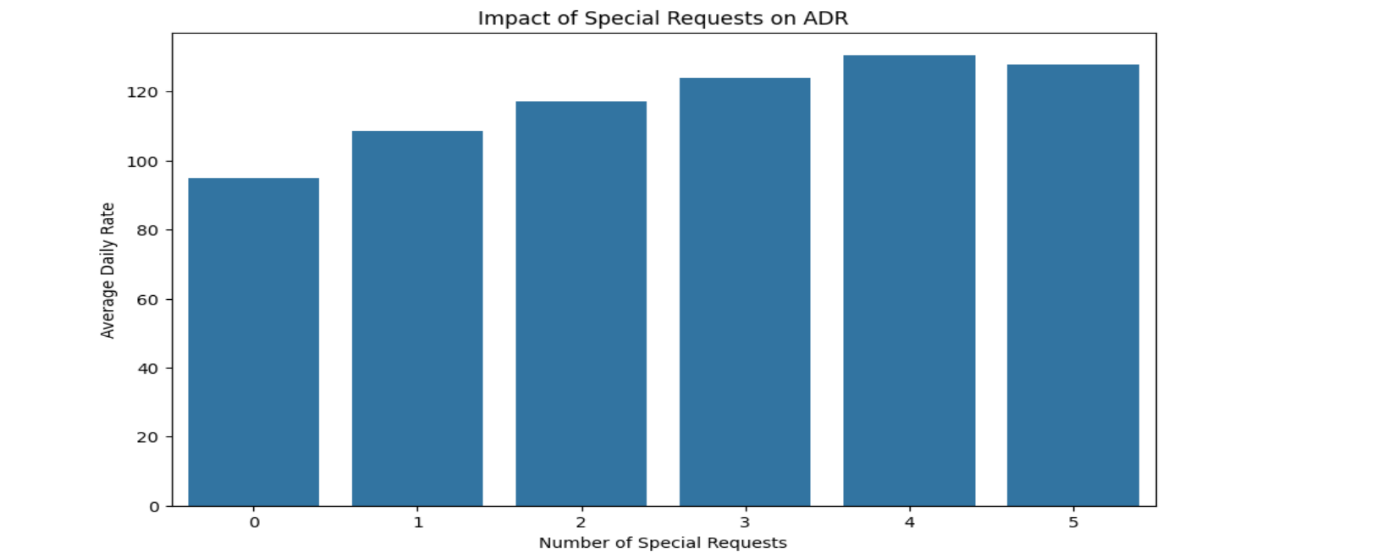




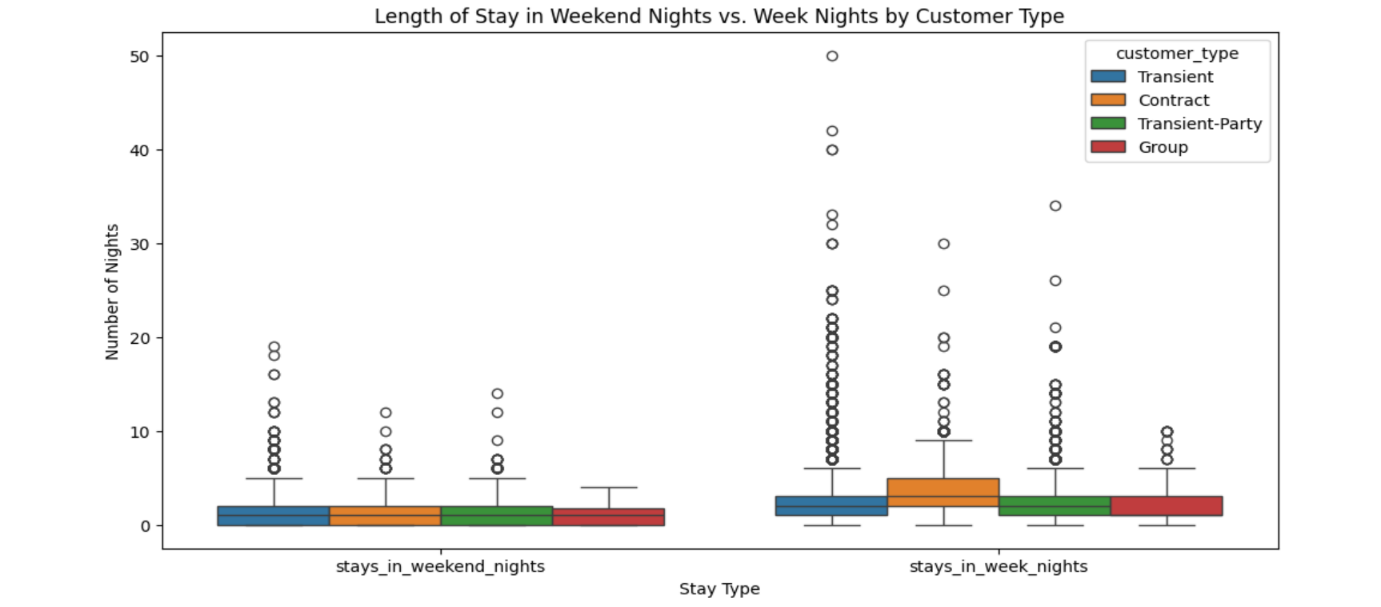




1. **Bar Graph**



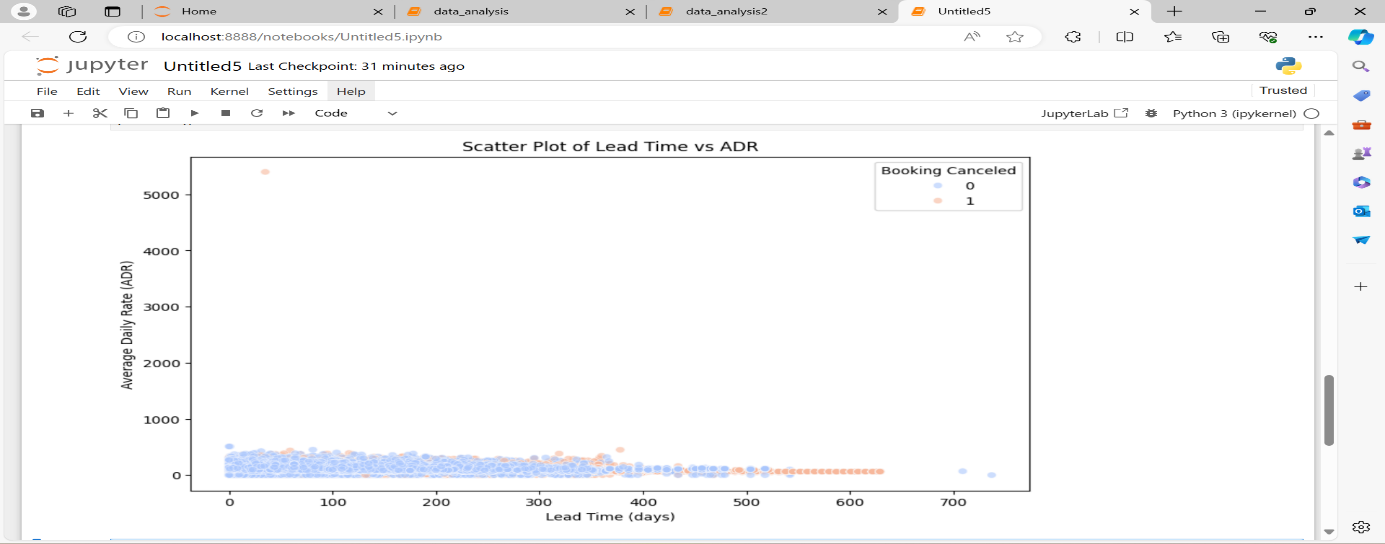
1. **Box Plot**



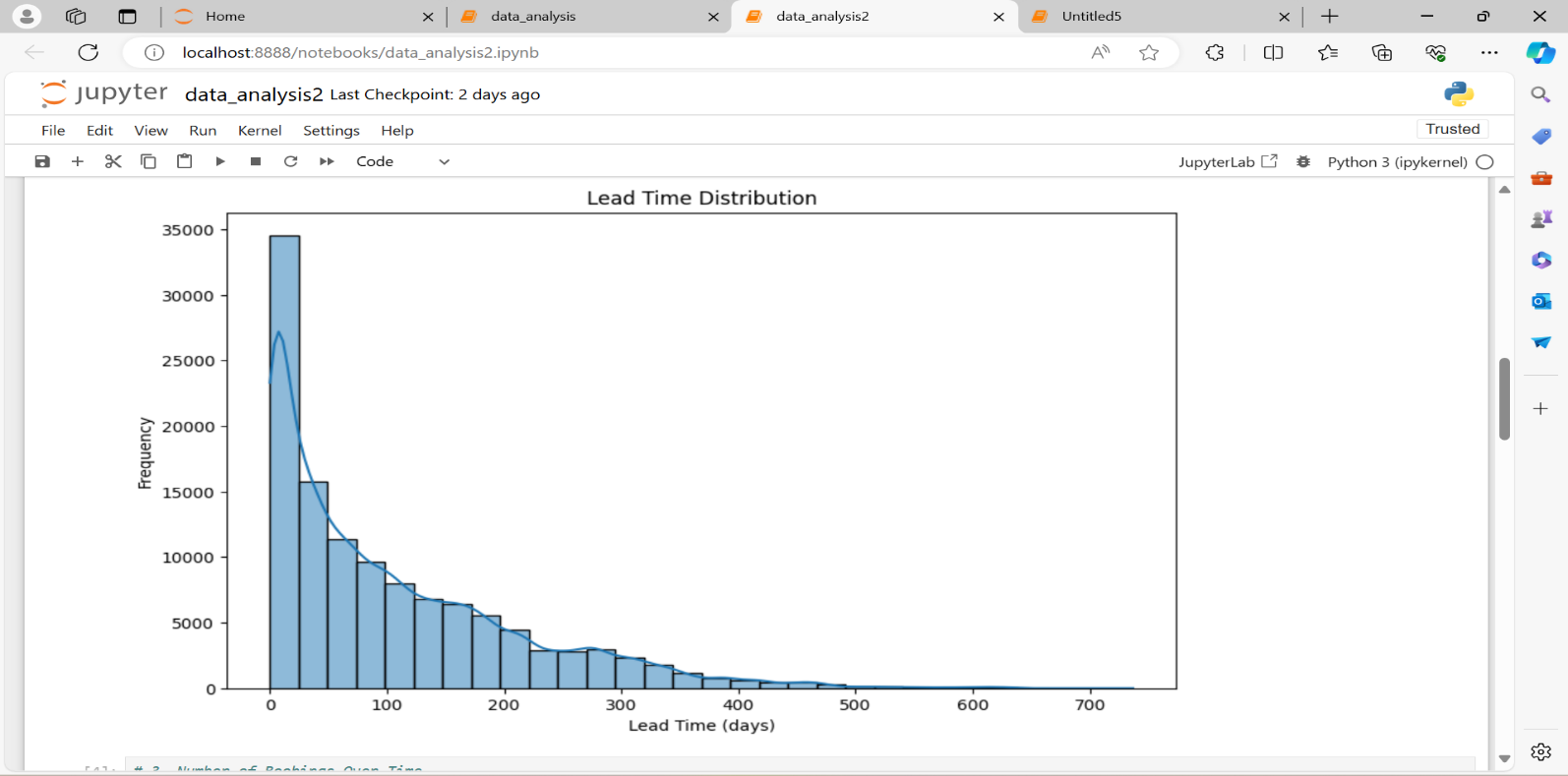
1. **Line Graph**



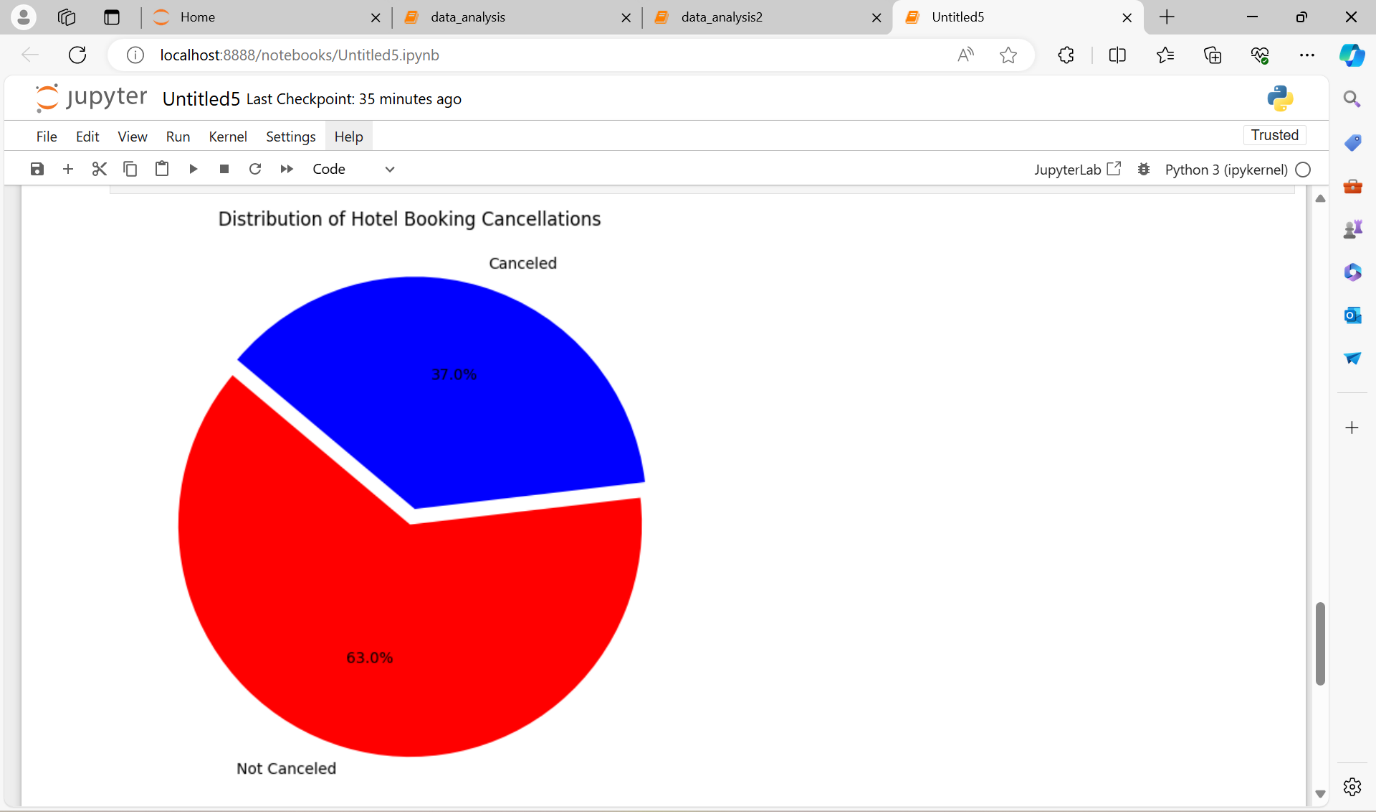
1. **Scatter Graph**

****

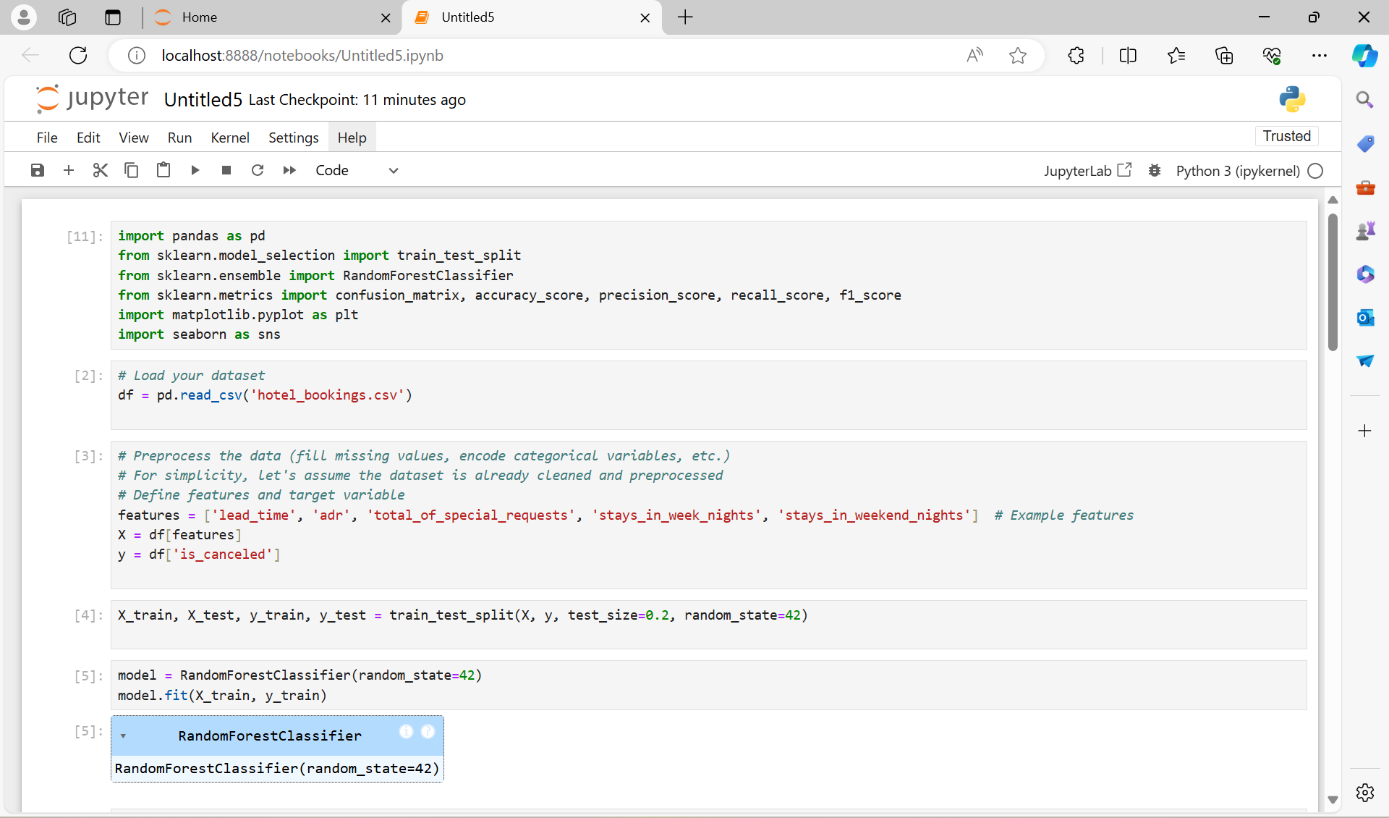
1. **Histogram**

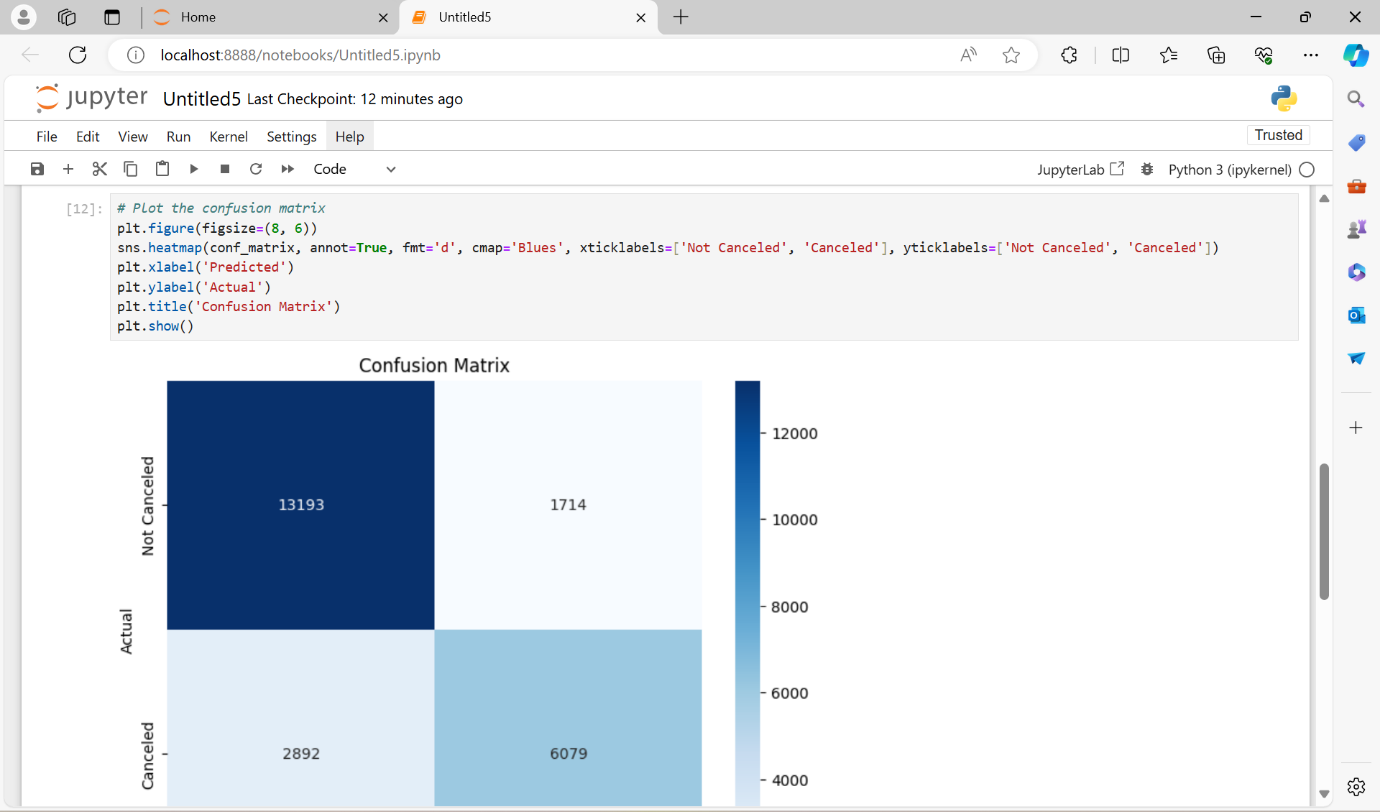
****

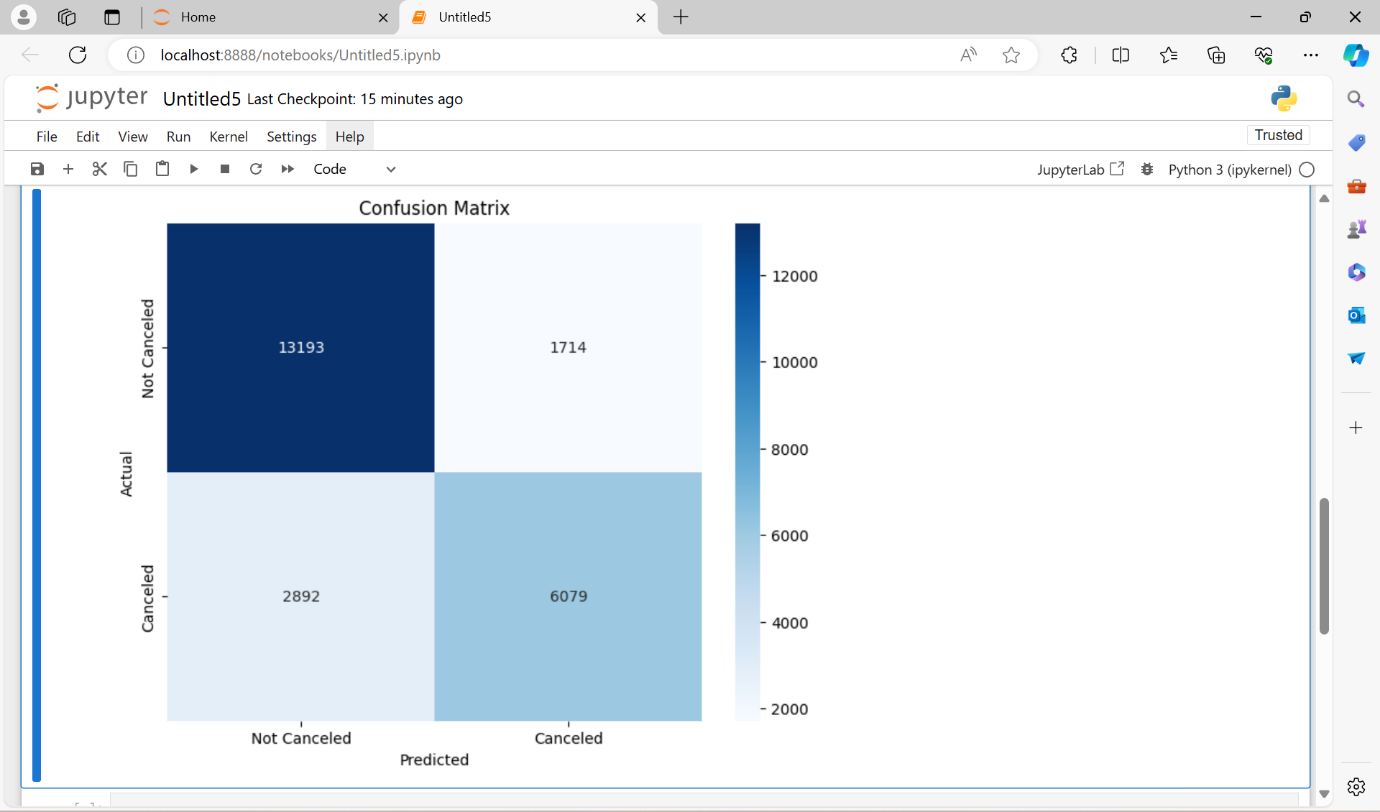
1. **Pie chart**

****

**Confusion Matrix**







Confusion matrices are powerful tools for evaluating the performance of classification models. They not only provide a detailed breakdown of actual versus predicted classifications but also enable the calculation of critical metrics such as accuracy, precision, recall, specificity, and F1 score. These metrics help in understanding the model’s strengths and weaknesses, guiding improvements and informing decision-making processes.

1. **CONCLUSION**

Resort hotels tend to have less bookings in comparison to city hotels so they need to work on their marketing strategy and promote the hotels more, especially on social media. Resort hotels could also reduce prices to increases booking percentages. May-August happens to be the busiest months but so the hotels should target more customers and try to do more business during these times. Although city hotels have more bookings, they also tend to have more cancellations so to prevent this they could take advance money during vacation. This would ensure most bookings to not being cancelled. They could also apply no-refund policies or make the refund policies rather strict so the customers choose not to cancel. It is quite clear most customers travel in pairs and bringing children or babies along are very rare so the hotels could advertise in ways that attract couples more and also business travellers. Most guests do not return but as these customers have already visited once, advertisements should be targeted in such ways so they are bound to return the next time they visit. The customers could also be offered special benefits if they do return to stay.

The analysis and predictive modeling of the hotel booking dataset provide valuable insights into factors influencing booking cancellations. By leveraging these insights, hotels can implement targeted strategies to reduce cancellations, optimize occupancy rates, and enhance overall operational efficiency. The deployment of a predictive model as a web service further empowers hotel management with real-time decision-making tools, ultimately improving customer satisfaction and revenue management.

Delving into hotel booking data through analysis and predictive modeling unlocks a treasure trove of knowledge concerning booking cancellations. By understanding the why behind cancellations, hotels can craft targeted strategies to prevent them in the first place. This translates to maximized occupancy rates, a more efficient operation, and ultimately, a healthier bottom line. But the benefits extend far beyond. Imagine a web service powered by a trusty prediction model – a real-time decision-making tool right at the fingertips of hotel management. With this kind of foresight, hotels can address potential issues proactively, ensuring a smooth and satisfying experience for their guests. This translates to not only increased customer satisfaction but also optimized revenue management, allowing hotels to tailor their offerings to meet the ever-changing needs of their clientele. In short, analyzing and harnessing the power of hotel booking data empowers hotels to not just survive but thrive in today's competitive landscape.

Resort hotels indeed face unique challenges compared to their urban counterparts, often experiencing lower booking rates and thus necessitating a reevaluation of their marketing strategies. Leveraging social media platforms for promotion can amplify visibility and attract more guests, particularly during peak seasons like May to August. Additionally, offering competitive pricing strategies can incentivize bookings and capitalize on heightened demand periods.

To address the issue of cancellations, resort hotels can adopt measures such as securing advance payments during the reservation process or implementing stringent refund policies. By doing so, they can mitigate the risk of last-minute cancellations, ensuring a more stable revenue stream and optimizing occupancy rates.

Understanding guest demographics is crucial for effective marketing. Resort hotels can tailor their advertisements to appeal to couples and business travelers, as these demographics are more prevalent among their customer base. Moreover, focusing on customer retention is paramount. By offering special incentives and benefits to returning guests, hotels can cultivate loyalty and encourage repeat visits, thereby fostering long-term relationships and maximizing revenue potential.

1. **CODE**
2. import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import joblib

from sklearn.model\_selection import train\_test\_split

from sklearn.tree import DecisionTreeClassifier # Add this import statement

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

import warnings

warnings.filterwarnings('ignore')

1. #Load the dataset

df = pd.read\_csv('hotel\_bookings.csv')

1. #datatype info

df.info()

1. df.head()
2. df.tail()
3. df.shape
4. df.columns
5. df.dtypes
6. #Testing some irrelevant features¶

#We will test three features, namely 'adults', 'children', and 'babies. These three features cannot be zero at the same time.

irrelevant\_rows = (df['children'] == 0) & (df['adults'] == 0) & (df['babies'] == 0)

print(f"There are {len(df[irrelevant\_rows])} irrelevant rows.")

1. # Remove irrelevant rows

df = df[~irrelevant\_rows]

df.shape

1. # Check for duplicate data

if df.duplicated().any():

print(f"There are as many as {df.duplicated().sum()} duplicate data.")

else:

print("There are no duplicate data.")

1. #Check how many different “hotel” and “customer” types are there

print("Hotel :",df['hotel'].unique())

print("Customers : ",df['customer\_type'].unique())

1. #Count values

df['customer\_type'].value\_counts()

1. **Bar Graph**

#Question 1: Impact of special requests on ADR

special\_requests\_adr = df.groupby('total\_of\_special\_requests')['adr'].mean().reset\_index()

plt.figure(figsize=(10, 6))

sns.barplot(data=special\_requests\_adr, x='total\_of\_special\_requests', y='adr')

plt.title('Impact of Special Requests on ADR')

plt.xlabel('Number of Special Requests')

plt.ylabel('Average Daily Rate')

plt.show()

1. **Box Plot**

# Question 2: Booking and Stay Patterns

# Length of stay in weekend nights vs. week nights

stay\_patterns = df[['stays\_in\_weekend\_nights', 'stays\_in\_week\_nights', 'customer\_type']].melt(id\_vars='customer\_type')

plt.figure(figsize=(12, 6))

sns.boxplot(data=stay\_patterns, x='variable', y='value', hue='customer\_type')

plt.title('Length of Stay in Weekend Nights vs. Week Nights by Customer Type')

plt.xlabel('Stay Type')

plt.ylabel('Number of Nights')

plt.show()

1. **Line Graph**

#Question 3: Impact of booking changes on reservation status

booking\_changes\_status = df.groupby('booking\_changes')['is\_canceled'].mean().reset\_index()

plt.figure(figsize=(10, 6))

sns.lineplot(data=booking\_changes\_status, x='booking\_changes', y='is\_canceled')

plt.title('Impact of Booking Changes on Cancellation Rate')

plt.xlabel('Number of Booking Changes')

plt.ylabel('Cancellation Rate')

plt.show()

1. **Scatter Graph**

#Question 4:Scatter plot of lead\_time vs adr, colored by is\_canceled

plt.figure(figsize=(10, 6))

sns.scatterplot(x='lead\_time', y='adr', hue='is\_canceled', data=df, palette='coolwarm', alpha=0.6)

plt.title('Scatter Plot of Lead Time vs ADR')

plt.xlabel('Lead Time (days)')

plt.ylabel('Average Daily Rate (ADR)')

plt.legend(title='Booking Canceled')

plt.show()

1. **Histogram**

#Question 5: Lead Time Distribution

plt.figure(figsize=(10, 6))

sns.histplot(data=df, x='lead\_time', bins=30, kde=True)

plt.title('Lead Time Distribution')

plt.xlabel('Lead Time (days)')

plt.ylabel('Frequency')

plt.show()

1. **Pie Chart**

#Question 6:Calculate the distribution of canceled vs not canceled bookings

cancellation\_counts = df['is\_canceled'].value\_counts()

# Create a pie chart

labels = ['Not Canceled', 'Canceled']

colors = ['red', 'blue']

explode = (0, 0.07) # explode the 'Canceled' slice

plt.figure(figsize=(7, 7))

plt.pie(cancellation\_counts, labels=labels, colors=colors, explode=explode, autopct='%1.1f%%', startangle=140)

plt.title('Distribution of Hotel Booking Cancellations')

plt.show()

1. # Preprocess the data (fill missing values, encode categorical variables, etc.)

# For simplicity, let's assume the dataset is already cleaned and preprocessed

# Define features and target variable

features = ['lead\_time', 'adr', 'total\_of\_special\_requests', 'stays\_in\_week\_nights', 'stays\_in\_weekend\_nights'] # Example features

X = df[features]

y = df['is\_canceled']

1. X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)
2. model = RandomForestClassifier(random\_state=42)

model.fit(X\_train, y\_train)

1. # Make predictions on the test set

y\_pred = model.predict(X\_test)

1. # Calculate the confusion matrix

conf\_matrix = confusion\_matrix(y\_test, y\_pred)

print("Confusion Matrix:")

print(conf\_matrix)

1. # Calculate other metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print(f"Accuracy: {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1 Score: {f1:.2f}")

1. # Plot the confusion matrix

plt.figure(figsize=(8, 6))

sns.heatmap(conf\_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=['Not Canceled', 'Canceled'], yticklabels=['Not Canceled', 'Canceled'])

plt.xlabel('Predicted')

plt.ylabel('Actual')

plt.title('Confusion Matrix')

plt.show()