EDA for Hotel Booking Cancelation Analysis

October 13, 2022

1 Project - Hotel Booking Cancellation Prediction

1.1 Problem Statement

1.1.1 Context

A significant number of hotel bookings are called off due to cancellations or no-shows. Typical reasons for cancellations include change of plans, scheduling conflicts, etc. This is often made easier by the option to do so free of charge or preferably at a low cost. This may be beneficial to hotel guests, but it is a less desirable and possibly revenue-diminishing factor for hotels to deal with. Such losses are particularly high on last-minute cancellations.

The new technologies involving online booking channels have dramatically changed customers' booking possibilities and behavior. This adds a further dimension to the challenge of how hotels handle cancellations, which are no longer limited to traditional booking and guest characteristics.

This pattern of cancellations of bookings impacts a hotel on various fronts: 1. **Loss of resources (revenue)** when the hotel cannot resell the room. 2. **Additional costs of distribution channels** by increasing commissions or paying for publicity to help sell these rooms. 3. **Lowering prices last minute**, so the hotel can resell a room, resulting in reducing the profit margin. 4. **Human resources to make arrangements** for the guests.

1.1.2 Objective

This increasing number of cancellations calls for a Machine Learning based solution that can help in predicting which booking is likely to be canceled. INN Hotels Group has a chain of hotels in Portugal - they are facing problems with this high number of booking cancellations and have reached out to your firm for data-driven solutions. You, as a Data Scientist, have to analyze the data provided to find which factors have a high influence on booking cancellations, build a predictive model that can predict which booking is going to be canceled in advance, and help in formulating profitable policies for cancellations and refunds.

1.1.3 Data Description

The data contains the different attributes of customers' booking details. The detailed data dictionary is given below:

Data Dictionary

- **Booking_ID:** Unique identifier of each booking

- **no_of_adults:** Number of adults
- **no_of_children:** Number of children
- **no_of_weekend_nights:** Number of weekend nights (Saturday or Sunday) the guest stayed or booked to stay at the hotel
- **no_of_week_nights:** Number of weekday nights (Monday to Friday) the guest stayed or booked to stay at the hotel
- **type_of_meal_plan:** Type of meal plan booked by the customer:
  - Not Selected – No meal plan selected
  - Meal Plan 1 – Breakfast
  - Meal Plan 2 – Half board (breakfast and one other meal)
  - Meal Plan 3 – Full board (breakfast, lunch, and dinner)
- **required_car_parking_space:** Does the customer require a car parking space? (0 - No, 1- Yes)
- **room_type_reserved:** Type of room reserved by the customer. The values are ciphered (encoded) by INN Hotels.
- **lead_time:** Number of days between the date of booking and the arrival date
- **arrival_year:** Year of arrival date
- **arrival_month:** Month of arrival date
- **arrival_date:** Date of the month
- **market_segment_type:** Market segment designation.
- **repeated_guest:** Is the customer a repeated guest? (0 - No, 1- Yes)
- **no_of_previous_cancellations:** Number of previous bookings that were canceled by the customer prior to the current booking
- **no_of_previous_bookings_not_canceled:** Number of previous bookings not canceled by the customer prior to the current booking
- **avg_price_per_room:** Average price per day of the reservation; prices of the rooms are dynamic. (in euros)
- **no_of_special_requests:** Total number of special requests made by the customer (e.g. high floor, view from the room, etc)
- **booking_status:** Flag indicating if the booking was canceled or not.

1.2 Importing the libraries required

[1]:
```
# Importing the basic libraries we will require for the project

# Libraries to help with reading and manipulating data
import pandas as pd
import numpy as np

# Libaries to help with data visualization
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()

# Importing the Machine Learning models we require from Scikit-Learn
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
```

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier

# Importing the other functions we may require from Scikit-Learn
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import MinMaxScaler, LabelEncoder, OneHotEncoder

# To get diferent metric scores
from sklearn.metrics import␣
 ↪confusion_matrix,classification_report,roc_auc_score,plot_confusion_matrix,precision_recall

# Code to ignore warnings from function usage
import warnings;
import numpy as np
warnings.filterwarnings('ignore')
```

1.3 Loading the dataset

[2]:
```
hotel = pd.read_csv("INNHotelsGroup.csv")
```

[3]:
```
# Copying data to another variable to avoid any changes to original data
data = hotel.copy()
```

1.4 Overview of the dataset

1.4.1 Viewing the first and last 5 rows of the dataset

Let's **view the first few rows and last few rows** of the dataset in order to understand its structure a little better.

We will use the head() and tail() methods from Pandas to do this.

[4]:
```
data.head()
```

[4]:

	Booking_ID	no_of_adults	no_of_children	no_of_weekend_nights	\
0	INN00001	2	0	1	
1	INN00002	2	0	2	
2	INN00003	1	0	2	
3	INN00004	2	0	0	
4	INN00005	2	0	1	

	no_of_week_nights	type_of_meal_plan	required_car_parking_space	\
0	2	Meal Plan 1	0	
1	3	Not Selected	0	
2	1	Meal Plan 1	0	
3	2	Meal Plan 1	0	
4	1	Not Selected	0	

```
  room_type_reserved  lead_time  arrival_year  arrival_month  arrival_date  \
0        Room_Type 1        224          2017             10             2
1        Room_Type 1          5          2018             11             6
2        Room_Type 1          1          2018              2            28
3        Room_Type 1        211          2018              5            20
4        Room_Type 1         48          2018              4            11

  market_segment_type  repeated_guest  no_of_previous_cancellations  \
0             Offline               0                             0
1              Online               0                             0
2              Online               0                             0
3              Online               0                             0
4              Online               0                             0

   no_of_previous_bookings_not_canceled  avg_price_per_room  \
0                                     0               65.00
1                                     0              106.68
2                                     0               60.00
3                                     0              100.00
4                                     0               94.50

   no_of_special_requests booking_status
0                       0   Not_Canceled
1                       1   Not_Canceled
2                       0       Canceled
3                       0       Canceled
4                       0       Canceled
```

[5]:
```
data.tail()
```

[5]:
```
      Booking_ID  no_of_adults  no_of_children  no_of_weekend_nights  \
36270   INN36271             3               0                     2
36271   INN36272             2               0                     1
36272   INN36273             2               0                     2
36273   INN36274             2               0                     0
36274   INN36275             2               0                     1

       no_of_week_nights type_of_meal_plan  required_car_parking_space  \
36270                  6       Meal Plan 1                           0
36271                  3       Meal Plan 1                           0
36272                  6       Meal Plan 1                           0
36273                  3      Not Selected                           0
36274                  2       Meal Plan 1                           0

      room_type_reserved  lead_time  arrival_year  arrival_month  \
36270        Room_Type 4         85          2018              8
```

```
36271        Room_Type 1        228          2018             10
36272        Room_Type 1        148          2018              7
36273        Room_Type 1         63          2018              4
36274        Room_Type 1        207          2018             12

       arrival_date market_segment_type  repeated_guest  \
36270             3              Online               0
36271            17              Online               0
36272             1              Online               0
36273            21              Online               0
36274            30             Offline               0

       no_of_previous_cancellations  no_of_previous_bookings_not_canceled  \
36270                             0                                     0
36271                             0                                     0
36272                             0                                     0
36273                             0                                     0
36274                             0                                     0

       avg_price_per_room  no_of_special_requests booking_status
36270              167.80                       1   Not_Canceled
36271               90.95                       2       Canceled
36272               98.39                       2   Not_Canceled
36273               94.50                       0       Canceled
36274              161.67                       0   Not_Canceled
```

1.4.2 Understanding the shape of the dataset

[6]:
```
data.shape
```

[6]:
```
(36275, 19)
```

- The dataset has 36275 rows and 19 columns.

1.4.3 Checking the data types of the columns for the dataset

[7]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
```

```
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  object
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  object
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  object
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  object
dtypes: float64(1), int64(13), object(5)
memory usage: 5.3+ MB
```

- `Booking_ID`, `type_of_meal_plan`, `room_type_reserved`, `market_segment_type`, and `booking_status` are of object type while rest columns are numeric in nature.
- There are no null values in the dataset.

1.4.4 Dropping duplicate valuesmif any

[8]:
```
# checking for duplicate values
data.duplicated().sum()
```

[8]:
```
0
```

- There are **no duplicate values** in the data.

1.4.5 Dropping the unique values column

Let's drop the Booking_ID column first before we proceed forward, as a column with unique values will have almost no predictive power for the Machine Learning problem at hand.

[9]:
```
data = data.drop(["Booking_ID"], axis=1)
```

[10]:
```
data.head()
```

[10]:
```
   no_of_adults  no_of_children  no_of_weekend_nights  no_of_week_nights  \
0             2               0                     1                  2
1             2               0                     2                  3
2             1               0                     2                  1
3             2               0                     0                  2
4             2               0                     1                  1

  type_of_meal_plan  required_car_parking_space room_type_reserved  lead_time  \
```

```
0       Meal Plan 1                       0        Room_Type 1        224
1      Not Selected                       0        Room_Type 1          5
2       Meal Plan 1                       0        Room_Type 1          1
3       Meal Plan 1                       0        Room_Type 1        211
4      Not Selected                       0        Room_Type 1         48

   arrival_year  arrival_month  arrival_date market_segment_type  \
0          2017             10             2             Offline
1          2018             11             6              Online
2          2018              2            28              Online
3          2018              5            20              Online
4          2018              4            11              Online

   repeated_guest  no_of_previous_cancellations  \
0               0                             0
1               0                             0
2               0                             0
3               0                             0
4               0                             0

   no_of_previous_bookings_not_canceled  avg_price_per_room  \
0                                     0               65.00
1                                     0              106.68
2                                     0               60.00
3                                     0              100.00
4                                     0               94.50

   no_of_special_requests booking_status
0                       0   Not_Canceled
1                       1   Not_Canceled
2                       0       Canceled
3                       0       Canceled
4                       0       Canceled
```

1.4.6 Checking the summary statistics of the dataset

[11]:
```
data.describe().T
```

[11]:
```
                                 count         mean        std     min  \
no_of_adults                   36275.0     1.844962   0.518715     0.0
no_of_children                 36275.0     0.105279   0.402648     0.0
no_of_weekend_nights           36275.0     0.810724   0.870644     0.0
no_of_week_nights              36275.0     2.204300   1.410905     0.0
required_car_parking_space     36275.0     0.030986   0.173281     0.0
lead_time                      36275.0    85.232557  85.930817     0.0
arrival_year                   36275.0  2017.820427   0.383836  2017.0
arrival_month                  36275.0     7.423653   3.069894     1.0
```

arrival_date	36275.0	15.596995	8.740447	1.0
repeated_guest	36275.0	0.025637	0.158053	0.0
no_of_previous_cancellations	36275.0	0.023349	0.368331	0.0
no_of_previous_bookings_not_canceled	36275.0	0.153411	1.754171	0.0
avg_price_per_room	36275.0	103.423539	35.089424	0.0
no_of_special_requests	36275.0	0.619655	0.786236	0.0

	25%	50%	75%	max
no_of_adults	2.0	2.00	2.0	4.0
no_of_children	0.0	0.00	0.0	10.0
no_of_weekend_nights	0.0	1.00	2.0	7.0
no_of_week_nights	1.0	2.00	3.0	17.0
required_car_parking_space	0.0	0.00	0.0	1.0
lead_time	17.0	57.00	126.0	443.0
arrival_year	2018.0	2018.00	2018.0	2018.0
arrival_month	5.0	8.00	10.0	12.0
arrival_date	8.0	16.00	23.0	31.0
repeated_guest	0.0	0.00	0.0	1.0
no_of_previous_cancellations	0.0	0.00	0.0	13.0
no_of_previous_bookings_not_canceled	0.0	0.00	0.0	58.0
avg_price_per_room	80.3	99.45	120.0	540.0
no_of_special_requests	0.0	0.00	1.0	5.0

Observations

- The number of adults per room is an average of 2.
- The number of children is maximum of 10. This is very unlikely.
- Some of the room don't have adults in them which is very unlikely.
- For the number of previous cancellation, the maximum is 13
- The average price per room is about 103.4. Some of the room has no price which is unlikely or are complimentary or promotional campaign by the hotel to their guests.
- The number of previous bookings not cancelled is more than those cancelled with maximum number of 58 as against 13 for the previously cancelled bookings

1.5 Exploratory Data Analysis

1.5.1 Univariate Analysis

Let's explore these variables in some more depth by observing their distributions.

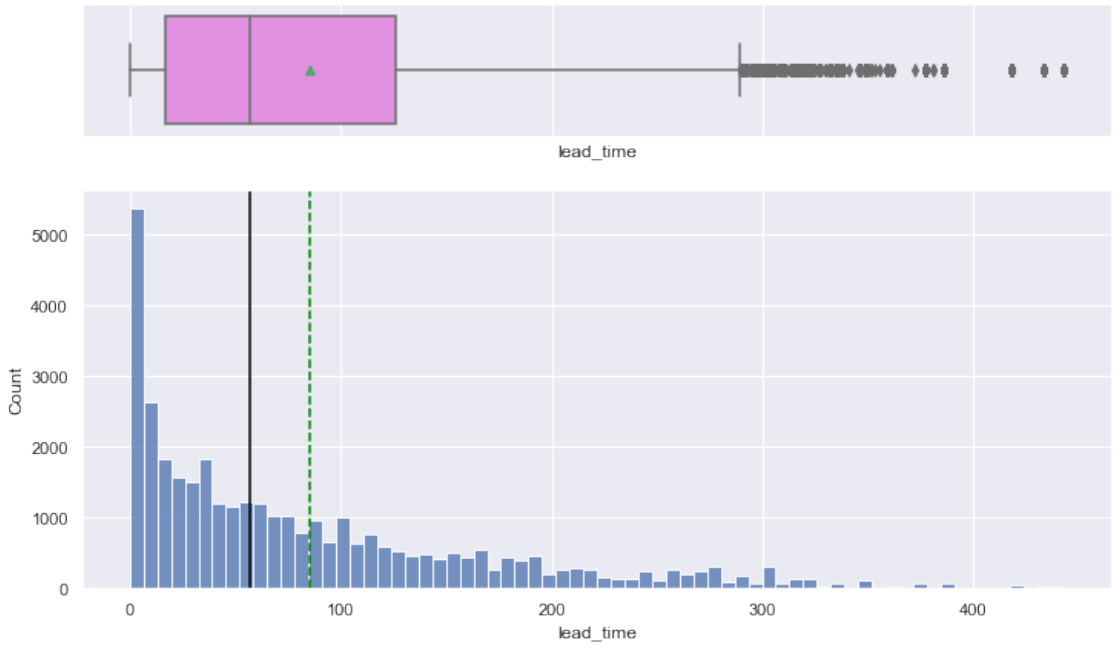
We will first define a **hist_box() function** that provides both a boxplot and a histogram in the same visual, with which we can perform univariate analysis on the columns of this dataset.

[12]:
```
# Function to plot a boxplot and a histogram along the same scale.


def hist_box(data, feature, figsize=(12, 7), kde=False, bins=None):
    """
    Boxplot and histogram combined
```

```
    data: dataframe
    feature: dataframe column
    figsize: size of figure (default (12,7))
    kde: whether to the show density curve (default False)
    bins: number of bins for histogram (default None)
    """
    f2, (ax_box2, ax_hist2) = plt.subplots(
        nrows=2,  # Number of rows of the subplot grid= 2
        sharex=True,  # x-axis will be shared among all subplots
        gridspec_kw={"height_ratios": (0.25, 0.75)},
        figsize=figsize,
    )  # creating the 2 subplots
    sns.boxplot(
        data=data, x=feature, ax=ax_box2, showmeans=True, color="violet"
    )  # boxplot will be created and a star will indicate the mean value of the␣
↪column
    sns.histplot(
        data=data, x=feature, kde=kde, ax=ax_hist2, bins=bins, palette="winter"
    ) if bins else sns.histplot(
        data=data, x=feature, kde=kde, ax=ax_hist2
    )  # For histogram
    ax_hist2.axvline(
        data[feature].mean(), color="green", linestyle="--"
    )  # Add mean to the histogram
    ax_hist2.axvline(
        data[feature].median(), color="black", linestyle="-"
    )  # Add median to the histogram
```

Plotting the histogram and box plot for the variable `Lead Time` using the hist_box function

[13]:
```
hist_box(data,'lead_time')
```



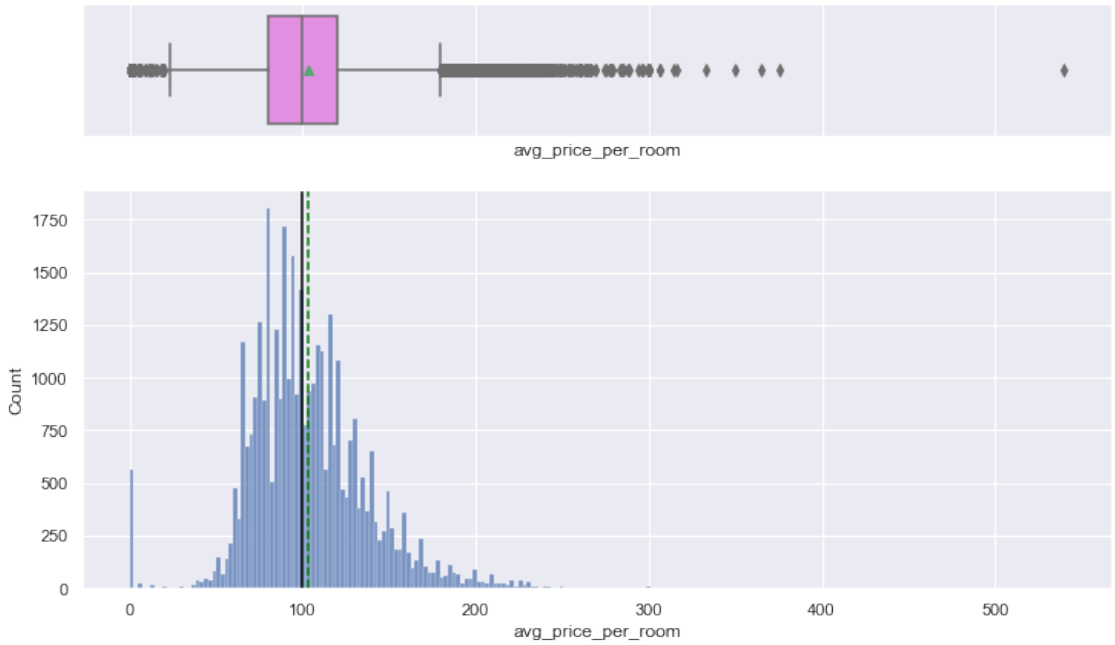
Observations - The distribution of lead time is right skewed - Number of days between the date of booking and the arrival date is in smaller increases.

Plotting the histogram and box plot for the variable `Average Price per Room` using the hist_box function.

[14]:
```
data['avg_price_per_room'].skew()
```

[14]:
```
0.6671328746979995
```

[15]:
```
hist_box(data,"avg_price_per_room")
```



Observation - The average price per room is almost normally distributed although the box plot shows outliers towards the right making it right skewed. Let us check these data.

[17]:
```
data[data['avg_price_per_room']>300]
```

[17]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	\
4150	2	2	1	2	
9461	3	0	0	2	
13944	2	2	0	1	
14773	2	2	0	3	
20900	2	1	1	2	
25670	2	2	1	2	
33114	2	0	0	1	
33955	2	0	1	2	
34306	2	2	0	3	

	type_of_meal_plan	required_car_parking_space	room_type_reserved	\
4150	Meal Plan 1	0	Room_Type 7	
9461	Meal Plan 1	0	Room_Type 4	
13944	Meal Plan 2	1	Room_Type 6	
14773	Meal Plan 1	0	Room_Type 6	
20900	Meal Plan 2	0	Room_Type 1	
25670	Meal Plan 2	0	Room_Type 6	
33114	Meal Plan 1	0	Room_Type 1	
33955	Meal Plan 2	0	Room_Type 4	
34306	Meal Plan 2	0	Room_Type 6	

	lead_time	arrival_year	arrival_month	arrival_date	\
4150	4	2018	7	8	
9461	21	2018	12	30	
13944	6	2018	8	13	
14773	28	2018	6	2	
20900	173	2018	7	25	
25670	11	2018	9	16	
33114	35	2018	3	25	
33955	57	2018	12	30	
34306	43	2018	12	29	

	market_segment_type	repeated_guest	no_of_previous_cancellations	\
4150	Online	0	0	
9461	Online	0	0	
13944	Online	0	0	
14773	Online	0	0	
20900	Offline	0	0	
25670	Online	0	0	
33114	Offline	0	0	
33955	Online	0	0	
34306	Online	0	0	

	no_of_previous_bookings_not_canceled	avg_price_per_room	\
4150	0	306.00	
9461	0	375.50	
13944	0	316.00	
14773	0	332.57	
20900	0	365.00	
25670	0	306.00	
33114	0	540.00	
33955	0	314.10	
34306	0	349.63	

	no_of_special_requests	booking_status
4150	3	Not_Canceled
9461	0	Not_Canceled
13944	0	Canceled
14773	1	Not_Canceled
20900	1	Canceled
25670	0	Canceled
33114	0	Canceled
33955	0	Not_Canceled
34306	1	Not_Canceled

- There are about 9 rooms with average price per room above 300, 5 of which were not canceled.

Interestingly some rooms have a price equal to 0. Let's check them.

[18]:
```
data[data["avg_price_per_room"] == 0]
```

[18]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	\
63	1	0	0	1	
145	1	0	0	2	
209	1	0	0	0	
266	1	0	0	2	
267	1	0	2	1	
…	…	…	…	…	
35983	1	0	0	1	
36080	1	0	1	1	
36114	1	0	0	1	
36217	2	0	2	1	
36250	1	0	0	2	

	type_of_meal_plan	required_car_parking_space	room_type_reserved	\
63	Meal Plan 1	0	Room_Type 1	
145	Meal Plan 1	0	Room_Type 1	
209	Meal Plan 1	0	Room_Type 1	
266	Meal Plan 1	0	Room_Type 1	
267	Meal Plan 1	0	Room_Type 1	
…	…	…	…	
35983	Meal Plan 1	0	Room_Type 7	
36080	Meal Plan 1	0	Room_Type 7	
36114	Meal Plan 1	0	Room_Type 1	
36217	Meal Plan 1	0	Room_Type 2	
36250	Meal Plan 2	0	Room_Type 1	

	lead_time	arrival_year	arrival_month	arrival_date	\
63	2	2017	9	10	
145	13	2018	6	1	
209	4	2018	2	27	
266	1	2017	8	12	
267	4	2017	8	23	
…	…	…	…	…	
35983	0	2018	6	7	
36080	0	2018	3	21	
36114	1	2018	3	2	
36217	3	2017	8	9	
36250	6	2017	12	10	

	market_segment_type	repeated_guest	no_of_previous_cancellations	\
63	Complementary	0	0	
145	Complementary	1	3	
209	Complementary	0	0	
266	Complementary	1	0	
267	Complementary	0	0	

```
…                    …              …                                …
35983       Complementary              1                                4
36080       Complementary              1                                3
36114              Online              0                                0
36217              Online              0                                0
36250              Online              0                                0

       no_of_previous_bookings_not_canceled  avg_price_per_room  \
63                                        0                 0.0
145                                       5                 0.0
209                                       0                 0.0
266                                       1                 0.0
267                                       0                 0.0
…                                       …                   …
35983                                    17                 0.0
36080                                    15                 0.0
36114                                     0                 0.0
36217                                     0                 0.0
36250                                     0                 0.0

       no_of_special_requests booking_status
63                          1   Not_Canceled
145                         1   Not_Canceled
209                         1   Not_Canceled
266                         1   Not_Canceled
267                         1   Not_Canceled
…                         …            …
35983                       1   Not_Canceled
36080                       1   Not_Canceled
36114                       0   Not_Canceled
36217                       2   Not_Canceled
36250                       0   Not_Canceled

[545 rows x 18 columns]
```

- There are quite a few hotel rooms which have a price equal to 0.
- In the market segment column, it looks like many values are complementary.

[19]:
```
data.loc[data["avg_price_per_room"] == 0, "market_segment_type"].value_counts()
```

[19]:
```
Complementary    354
Online           191
Name: market_segment_type, dtype: int64
```

- It makes sense that most values with room prices equal to 0 are the rooms given as complimentary service from the hotel.
- The rooms booked online must be a part of some promotional campaign done by the hotel.

[18]:
```
# Calculating the 25th quantile
Q1 = data["avg_price_per_room"].quantile(0.25)

# Calculating the 75th quantile
Q3 = data["avg_price_per_room"].quantile(0.75)

# Calculating IQR
IQR = Q3 - Q1

# Calculating value of upper whisker
Upper_Whisker = Q3 + 1.5 * IQR
Upper_Whisker
```

[18]:
```
179.55
```

[19]:
```
# assigning the outliers the value of upper whisker
data.loc[data["avg_price_per_room"] >= 500, "avg_price_per_room"] =␣
 ↪Upper_Whisker
```

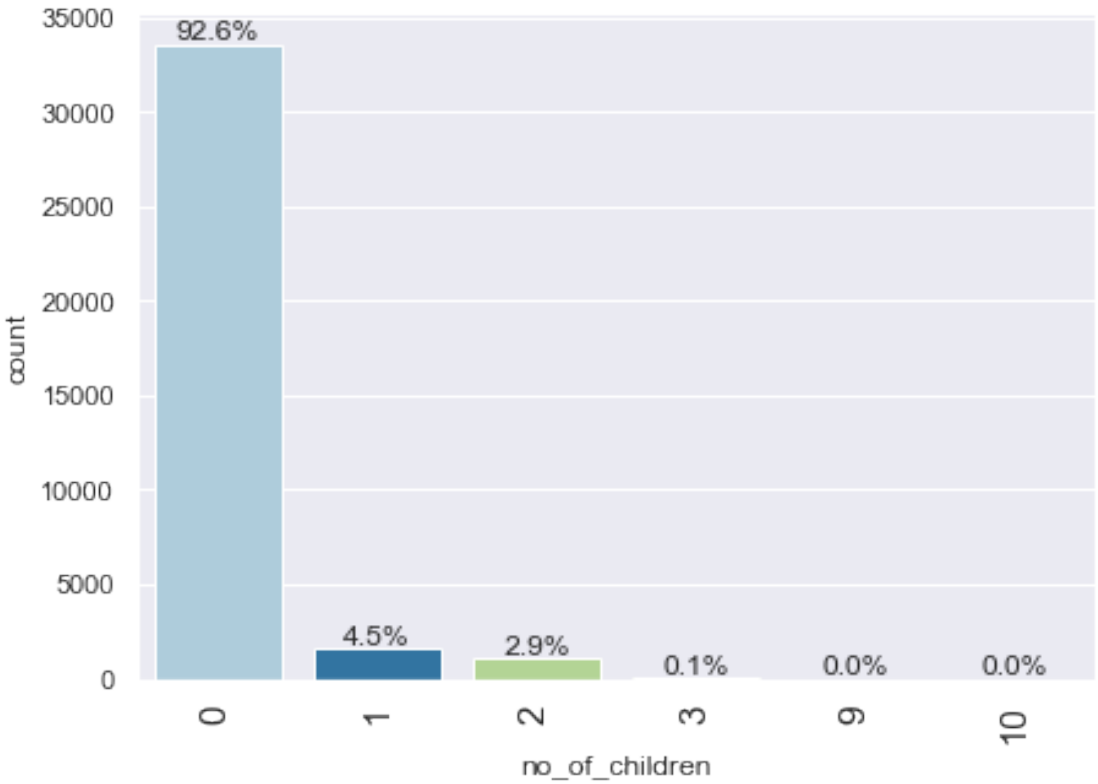
Let's understand the distribution of the categorical variables

[22]:
```
# Function to create labeled barplots


def labeled_barplot(data, feature, perc=True, n=None):
    """
    Barplot with percentage at the top

    data: dataframe
    feature: dataframe column
    perc: whether to display percentages instead of count (default is False)
    n: displays the top n category levels (default is None, i.e., display all␣
 ↪levels)
    """

    total = len(data[feature])  # length of the column
    count = data[feature].nunique()
    if n is None:
        plt.figure(figsize=(count + 1, 5))
    else:
        plt.figure(figsize=(n + 1, 5))

    plt.xticks(rotation=90, fontsize=15)
    ax = sns.countplot(
        data=data,
        x=feature,
        palette="Paired",
```

```
        order=data[feature].value_counts().index[:n].sort_values(),
    )

    for p in ax.patches:
        if perc == True:
            label = "{:.1f}%".format(
                100 * p.get_height() / total
            )  # percentage of each class of the category
        else:
            label = p.get_height()  # count of each level of the category

        x = p.get_x() + p.get_width() / 2  # width of the plot
        y = p.get_height()  # height of the plot
        ax.annotate(
            label,
            (x, y),
            ha="center",
            va="center",
            size=12,
            xytext=(0, 5),
            textcoords="offset points",
        )  # annotate the percentage

    plt.show()  # show the plot
```

Number of Children

[23]:
```
labeled_barplot(data,'no_of_children')
```

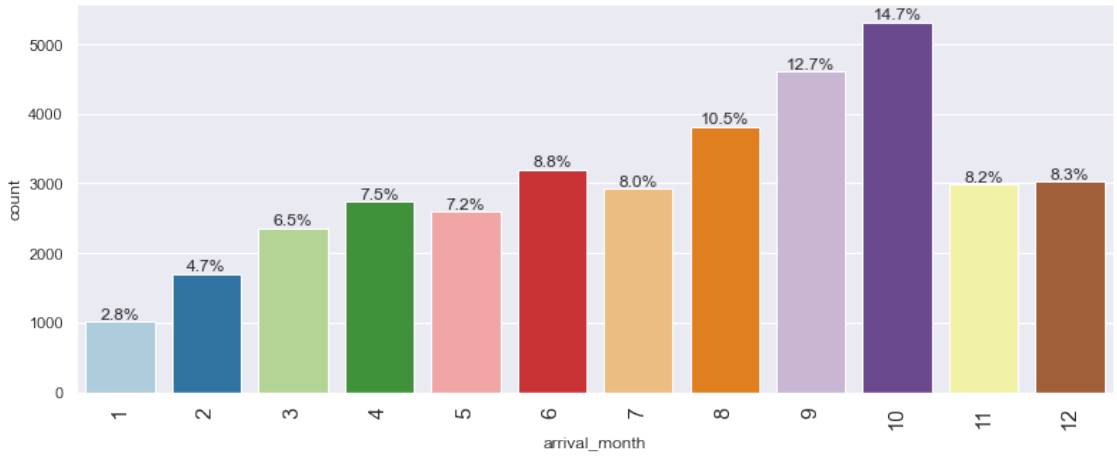


- Customers were not travelling with children in 93% of cases.
- There are some values in the data where the number of children is 9 or 10, which is highly unlikely.
- We will replace these values with the maximum value of 3 children.

[24]:
```
# replacing 9, and 10 children with 3
data["no_of_children"] = data["no_of_children"].replace([9, 10], 3)
```

Arrival Month

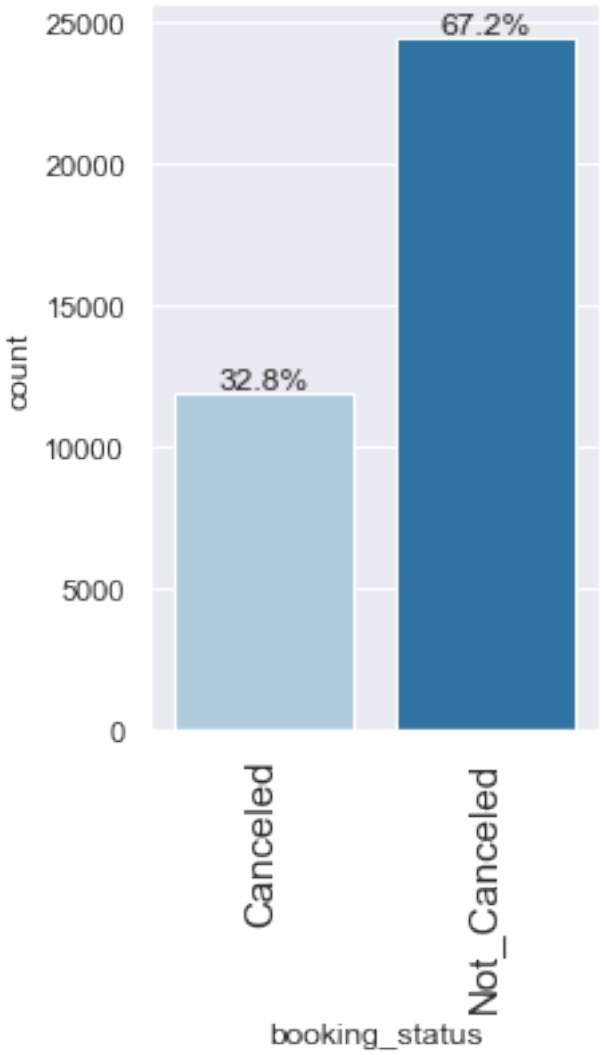
[25]:
```
labeled_barplot(data,"arrival_month")
```



- October is the busiest month for hotel arrivals followed by September and August. **Over 35% of all bookings**, as we see in the above table, were for one of these three months.
- Around 14.7% of the bookings were made for an October arrival.
- More bookings are done in August, September and October. This is most likely as there are more vacations during summer.

Booking Status

[26]:
```
labeled_barplot(data,"booking_status")
```



- 32.8% of the bookings were canceled by the customers whilst 67.2% were not cancelled.

Let's encode Canceled bookings to 1 and Not_Canceled as 0 for further analysis

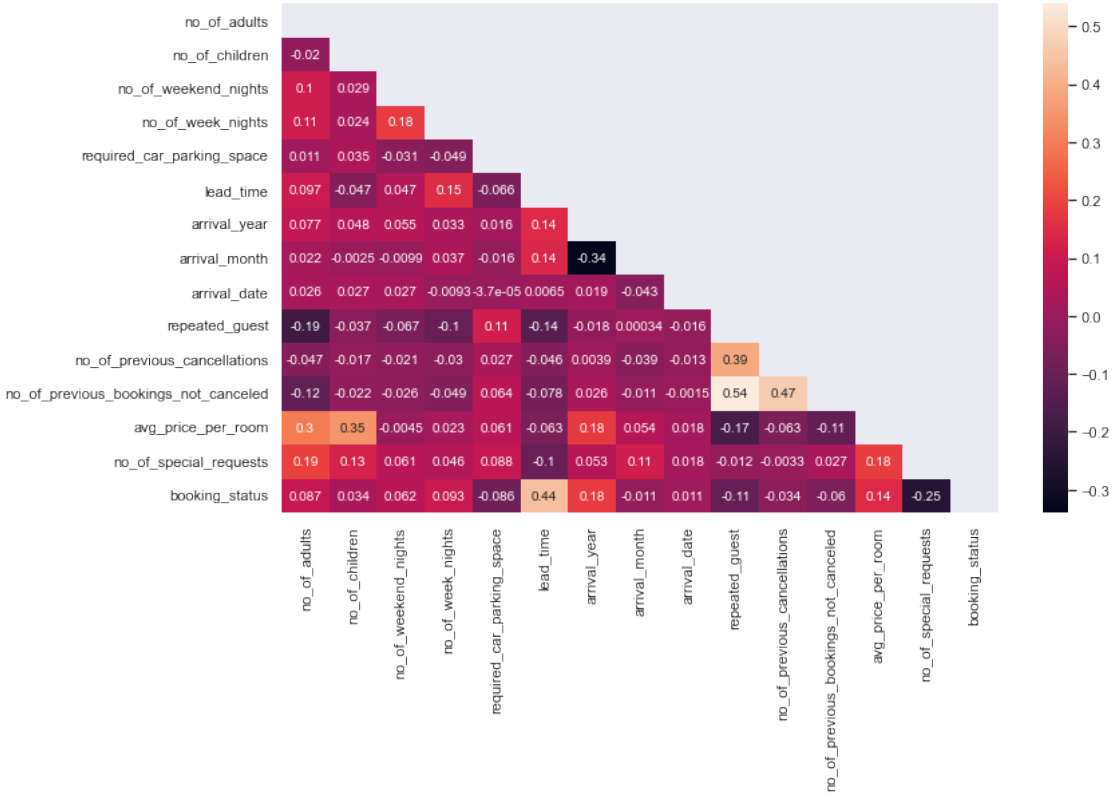
[27]:
```
data["booking_status"] = data["booking_status"].apply(
    lambda x: 1 if x == "Canceled" else 0
)
```

1.5.2 Bivariate Analysis

Finding and visualizing the correlation matrix using a heatmap

[32]:
```
cols_list = data.select_dtypes(include=np.number).columns.tolist()
plt.figure(figsize=(12, 7))
matrix = np.triu(data.corr())
sns.heatmap(data.corr(), annot=True, mask=matrix)
```

```
plt.show()
```



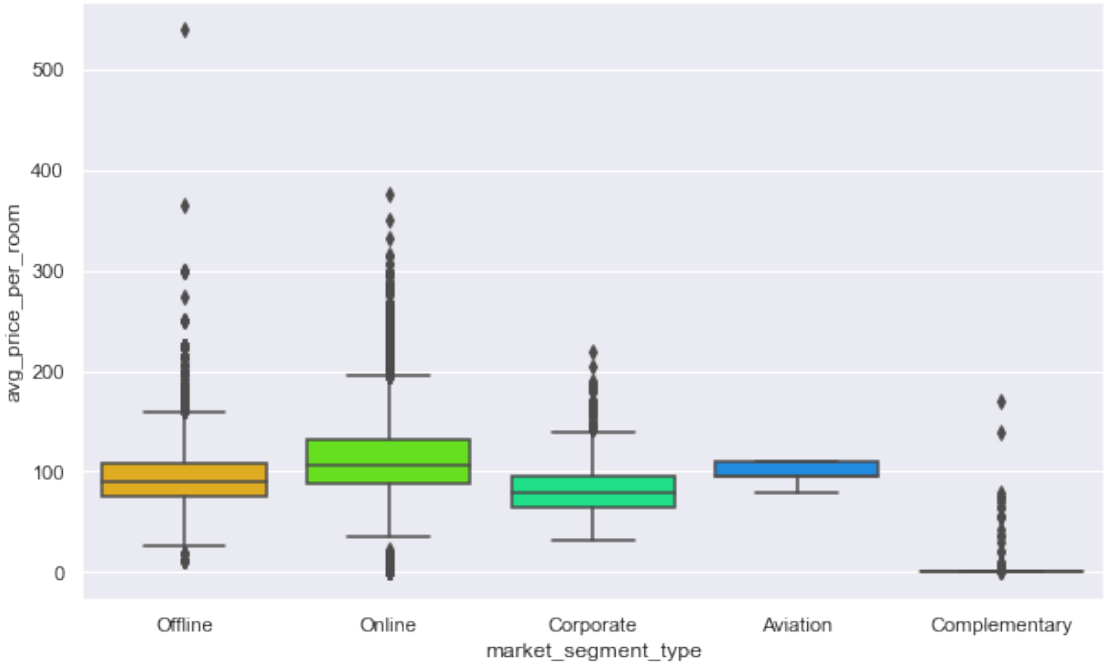
Observations

- There is a high correlation between repeated guests and the number if previously booked not cancelled. This is possible as some guest that had stayed in the hotel previously had booked again and don't need to cancel maybe due to the great service they enjoyed and would like to stay at the hotel again.
- There is a high correlation between number of children and avg price per room. This is expected as the price of a room is higher with the number of its occupants.
- There is a hig correlation between lead time and the booking status. This may be so as the client has ample time to make the bookings. They may or may not cancel before the arrival date.
- Most of the variables are independent.

Hotel rates are dynamic and change according to demand and customer demographics. Let's see how prices vary across different market segments

[33]:
```
plt.figure(figsize=(10, 6))
sns.boxplot(
    data=data, x="market_segment_type", y="avg_price_per_room",␣
 ↪palette="gist_rainbow"
)
```

```
plt.show()
```



- Rooms booked online have high variations in prices.
- The offline and corporate room prices are almost similar.
- Complementary market segment gets the rooms at very low prices, which makes sense.

- We will define a **stacked barplot()** function to help analyse how the target variable varies across predictor categories.
- For Booking status, 0 means not cancelled, 1 means cancelled

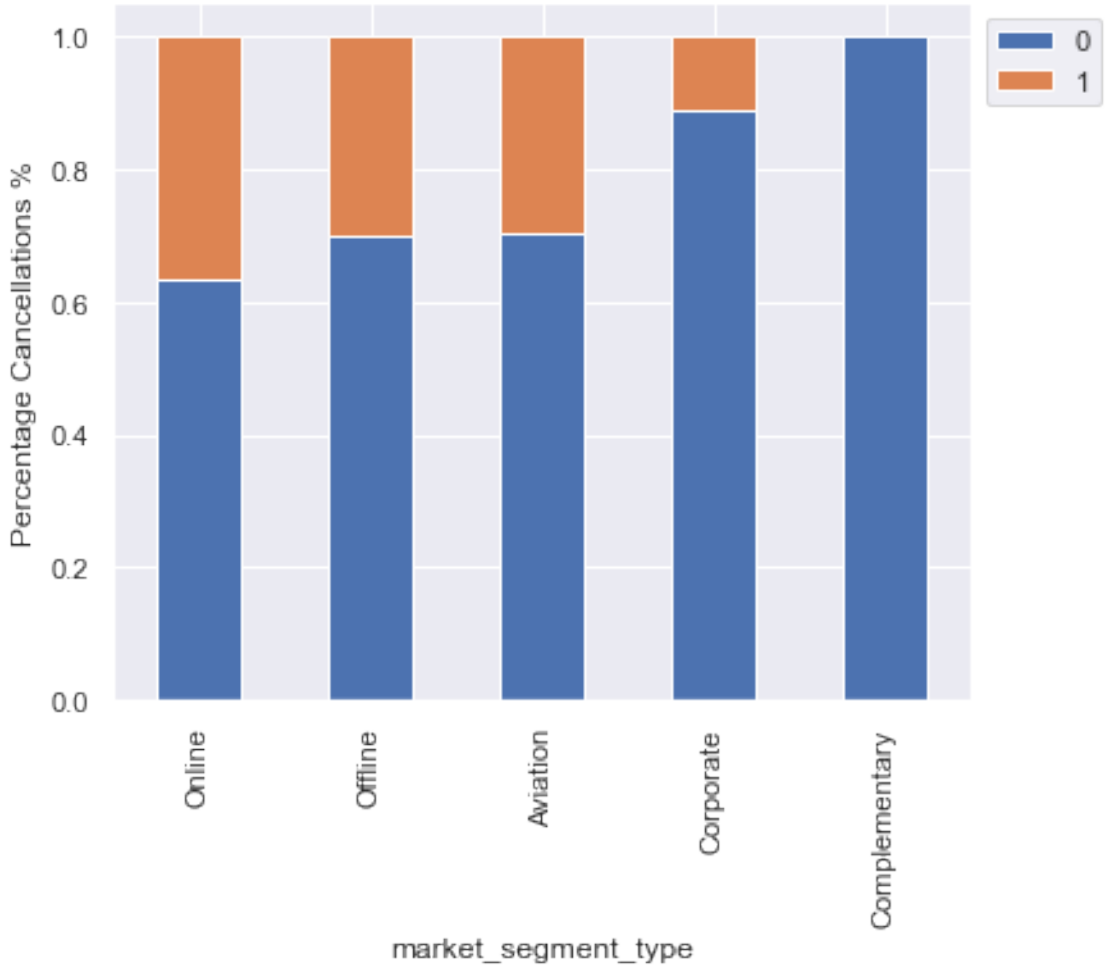
[34]:
```
# Defining the stacked_barplot() function


def stacked_barplot(data, predictor, target, figsize=(10,6)):
    """
    Print the category counts and plot a stacked bar chart

    data: dataframe
    predictor: independent variable
    target: target variable
    """
    count = data[predictor].nunique()
    sorter = data[target].value_counts().index[-1]
    tab1 = pd.crosstab(data[predictor], data[target], margins=True).sort_values(
        by=sorter, ascending=False
    )
```

```
    print(tab1)
    print("-" * 120)
    tab = pd.crosstab(data[predictor], data[target], normalize="index").
↪sort_values(
        by=sorter, ascending=False
    )
    tab.plot(kind="bar", stacked=True, figsize=(count + 1, 5))
    plt.legend(
        loc="lower left",
        frameon=False,
    )
    plt.legend(loc="upper left", bbox_to_anchor=(1, 1))
    plt.ylabel('Percentage Cancellations %')
    plt.show()
```

Plotting the stacked barplot for the variable `Market Segment Type` against the target variable `Booking Status` using the stacked_barplot function

[35]:
```
stacked_barplot(data,'market_segment_type','booking_status' )
```

```
booking_status           0      1    All
market_segment_type
All                  24390  11885  36275
Online               14739   8475  23214
Offline               7375   3153  10528
Corporate             1797    220   2017
Aviation                88     37    125
Complementary          391      0    391
------------------------------------------------------------------------------------
----------------------------------------
```



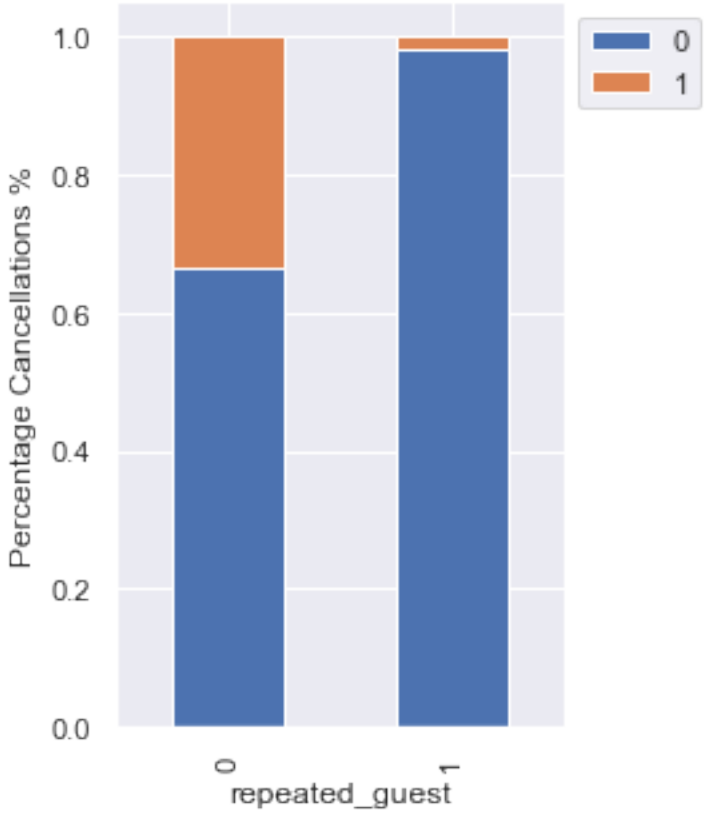
Observations - There are no cancellation in the complimentary market segment type. This is a complimentary service which is free and so there may be no cancellation here - Amongst the market segment with cancellation, the percentage of cancellation for the online market segment is the highest while corporate market segment is the lowest. Cancellation of booking in Corporate organizations are not frequent unless situations beyond their control such as change of dates of their events.

Plotting the stacked barplot for the variable `Repeated Guest` against the target variable `Booking Status` using the stacked_barplot function** Repeating guests are the guests who stay in the hotel often and are important to brand equity.

[36]:
```
stacked_barplot(data, 'repeated_guest','booking_status')
```

```
booking_status      0      1    All
repeated_guest
All             24390  11885  36275
0               23476  11869  35345
```

```
1                   914      16     930
------------------------------------------------------------------------------
--------------------------------------
```



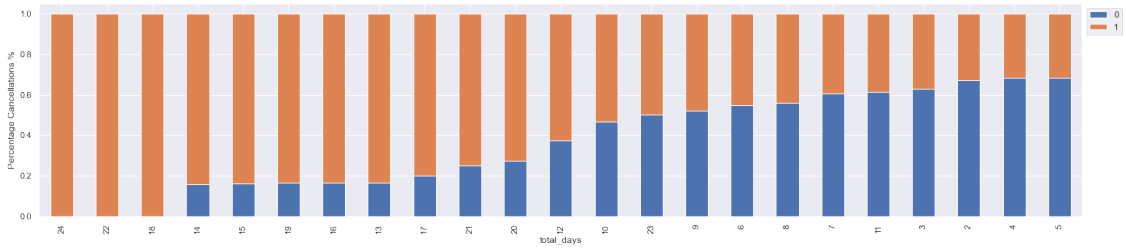
Observations - Repeated guest rarely cancel their reservation and if they do, it may due to situations beyond their control. - This is because guest they had stayed in the hotel previously and had enjoyed the service of the hotel. They are comfortable with the brand of INN Hotels Group - First time guest tends to cancel more frequently as they have not experience the service of the hotel before.

Let's analyze the customer who stayed for at least a day at the hotel.

[37]:
```
stay_data = data[(data["no_of_week_nights"] > 0) &␣
 ↪(data["no_of_weekend_nights"] > 0)]
stay_data["total_days"] = (stay_data["no_of_week_nights"] +␣
 ↪stay_data["no_of_weekend_nights"])

stacked_barplot(stay_data, "total_days", "booking_status",figsize=(15,6))
```

```
booking_status      0     1    All
total_days
All             10979  6115  17094
```

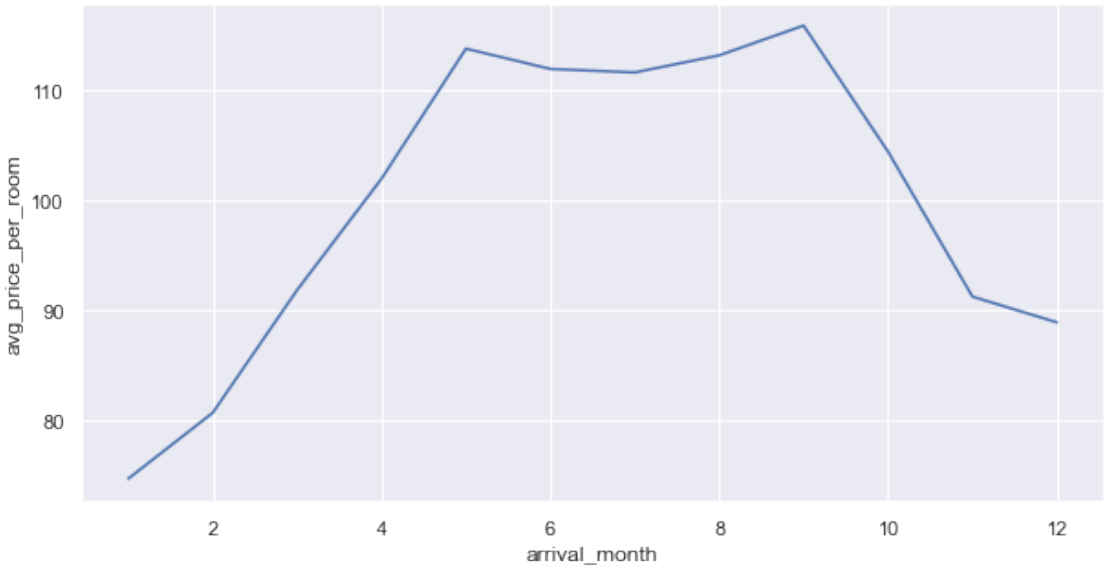
```
3                3689  2183   5872
4                2977  1387   4364
5                1593   738   2331
2                1301   639   1940
6                 566   465   1031
7                 590   383    973
8                 100    79    179
10                 51    58    109
9                  58    53    111
14                  5    27     32
15                  5    26     31
13                  3    15     18
12                  9    15     24
11                 24    15     39
20                  3     8     11
19                  1     5      6
16                  1     5      6
17                  1     4      5
18                  0     3      3
21                  1     3      4
22                  0     2      2
23                  1     1      2
24                  0     1      1
--------------------------------------------------------------------------------
----------------------------------------
```



- The general trend is that the chances of cancellation increase as the number of days the customer planned to stay at the hotel increases.

As hotel room prices are dynamic, Let's see how the prices vary across different months

[38]:
```
plt.figure(figsize=(10, 5))
sns.lineplot(y=data["avg_price_per_room"], x=data["arrival_month"], ci=None)
plt.show()
```



- The price of rooms is highest in May to September - around 115 euros per room.

Recommendations - From our analysis, the average price per room is one of the reasons why hotels bookings are cancelled. The company should try to reduce the prices of the rooms when it is not the peak periods and make these prices during peak periods (between May and October) reasonably lower than their competitors to retain hotel bookings. If their prices are considerably lower with low impact on their profit margin, they tend to have more bookings that are not cancelled and have higher number of repeated guests. - The company should carry out frequent campaigns and complimentary services to attract more clients in each market segment especially the online segment. Rooms booked online have high variations in prices and they also tend to cancel their bookings more frequently than other segments. Lower prices and frequent online campaigns can help retain the bookings of the online segments. - To make first time guests a repeated guest, INN Hotel should make their customers feel amazing and special by frequently communicating with them. This may be in form of emails, paid questionnaires and surveys, discounts on hotel room for the client and their referrals. As the number of hotels booking cancellation from repeated guest is quite low, ensuring the first-time guest are converted to frequent guest will help reduce the booking cancellations and retain more customers. - The general trend is that the chances of cancellation increase as the number of days the customer planned to stay at the hotel increases. For bookings with longer stay, INN Group should propose more services. The INN Group can propose complimentary dinner for booking that tend to stay for more than 5 days while others enjoy only complimentary breakfast. They might also propose discount for longer stays which would tend to sway the client in INN Group's favor to not cancel their bookings and enjoy these complimentary services.