Hotel Booking Data

Importing Liabraries

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

Loading Dataset

```
data= pd.read_csv("/content/hotel_bookings 2.csv")
data
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_da
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	
3	Resort Hotel	0	13	2015	July	
4	Resort Hotel	0	14	2015	July	
...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	
119386	City Hotel	0	102	2017	August	
119387	City Hotel	0	34	2017	August	
119388	City Hotel	0	109	2017	August	
119389	City Hotel	0	205	2017	August	

119390 rows × 32 columns

EDA And Data Cleaning

```
data.head()
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_w
0	Resort Hotel	0	342	2015	July	
1	Resort Hotel	0	737	2015	July	
2	Resort Hotel	0	7	2015	July	

```
data.shape
```

```
(119390, 32)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   hotel                           119390 non-null  object 
 1   is_canceled                     119390 non-null  int64  
 2   lead_time                       119390 non-null  int64  
 3   arrival_date_year               119390 non-null  int64  
 4   arrival_date_month              119390 non-null  object 
 5   arrival_date_week_number        119390 non-null  int64  
 6   arrival_date_day_of_month       119390 non-null  int64  
 7   stays_in_weekend_nights         119390 non-null  int64  
 8   stays_in_week_nights            119390 non-null  int64  
 9   adults                          119390 non-null  int64  
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64  
 12  meal                            119390 non-null  object 
 13  country                         118902 non-null  object 
 14  market_segment                  119390 non-null  object 
 15  distribution_channel            119390 non-null  object 
 16  is_repeated_guest               119390 non-null  int64  
 17  previous_cancellations          119390 non-null  int64  
 18  previous_bookings_not_canceled  119390 non-null  int64  
 19  reserved_room_type              119390 non-null  object 
 20  assigned_room_type              119390 non-null  object 
 21  booking_changes                 119390 non-null  int64  
 22  deposit_type                    119390 non-null  object 
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64  
 26  customer_type                   119390 non-null  object 
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64  
 29  total_of_special_requests       119390 non-null  int64  
 30  reservation_status              119390 non-null  object 
 31  reservation_status_date         119390 non-null  object 
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

```
data['reservation_status_date'] = pd.to_datetime(data['reservation_status_date'])
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
 #   Column                          Non-Null Count   Dtype  
---  ------                          --------------   -----  
 0   hotel                           119390 non-null  object 
 1   is_canceled                     119390 non-null  int64  
 2   lead_time                       119390 non-null  int64  
 3   arrival_date_year               119390 non-null  int64  
 4   arrival_date_month              119390 non-null  object 
 5   arrival_date_week_number        119390 non-null  int64  
 6   arrival_date_day_of_month       119390 non-null  int64  
 7   stays_in_weekend_nights         119390 non-null  int64  
 8   stays_in_week_nights            119390 non-null  int64
 9   adults                          119390 non-null  int64
 10  children                        119386 non-null  float64
 11  babies                          119390 non-null  int64
 12  meal                            119390 non-null  object
 13  country                         118902 non-null  object
 14  market_segment                  119390 non-null  object
 15  distribution_channel            119390 non-null  object
 16  is_repeated_guest               119390 non-null  int64
 17  previous_cancellations          119390 non-null  int64
 18  previous_bookings_not_canceled  119390 non-null  int64
 19  reserved_room_type              119390 non-null  object
 20  assigned_room_type              119390 non-null  object
 21  booking_changes                 119390 non-null  int64
 22  deposit_type                    119390 non-null  object
 23  agent                           103050 non-null  float64
 24  company                         6797 non-null    float64
 25  days_in_waiting_list            119390 non-null  int64
 26  customer_type                   119390 non-null  object
 27  adr                             119390 non-null  float64
 28  required_car_parking_spaces     119390 non-null  int64
 29  total_of_special_requests       119390 non-null  int64
 30  reservation_status              119390 non-null  object
 31  reservation_status_date         119390 non-null  datetime64[ns]
dtypes: datetime64[ns](1), float64(4), int64(16), object(11)
memory usage: 29.1+ MB
```

```
data.describe(include='object')
```

	hotel	arrival_date_month	meal	country	market_segment	distribution_channe
count	119390	119390	119390	118902	119390	11939
unique	2	12	5	177	8	
top	City Hotel	August	BB	PRT	Online TA	TA/T
freq	79330	13877	92310	48590	56477	9787

```
#check the unique values
for col in data.describe(include='object').columns:
    print(col)
    print(data[col].unique())
    print('-'*50)
```

```
hotel
['Resort Hotel' 'City Hotel']
--------------------------------------------------
arrival_date_month
['July' 'August' 'September' 'October' 'November' 'December' 'January'
 'February' 'March' 'April' 'May' 'June']
--------------------------------------------------
meal
['BB' 'FB' 'HB' 'SC' 'Undefined']
--------------------------------------------------
country
['PRT' 'GBR' 'USA' 'ESP' 'IRL' 'FRA' nan 'ROU' 'NOR' 'OMN' 'ARG' 'POL'
 'DEU' 'BEL' 'CHE' 'CN' 'GRC' 'ITA' 'NLD' 'DNK' 'RUS' 'SWE' 'AUS' 'EST'
 'CZE' 'BRA' 'FIN' 'MOZ' 'BWA' 'LUX' 'SVN' 'ALB' 'IND' 'CHN' 'MEX' 'MAR'
 'UKR' 'SMR' 'LVA' 'PRI' 'SRB' 'CHL' 'AUT' 'BLR' 'LTU' 'TUR' 'ZAF' 'AGO'
 'ISR' 'CYM' 'ZMB' 'CPV' 'ZWE' 'DZA' 'KOR' 'CRI' 'HUN' 'ARE' 'TUN' 'JAM'
 'HRV' 'HKG' 'IRN' 'GEO' 'AND' 'GIB' 'URY' 'JEY' 'CAF' 'CYP' 'COL' 'GGY'
 'KWT' 'NGA' 'MDV' 'VEN' 'SVK' 'FJI' 'KAZ' 'PAK' 'IDN' 'LBN' 'PHL' 'SEN'
 'SYC' 'AZE' 'BHR' 'NZL' 'THA' 'DOM' 'MKD' 'MYS' 'ARM' 'JPN' 'LKA' 'CUB'
 'CMR' 'BIH' 'MUS' 'COM' 'SUR' 'UGA' 'BGR' 'CIV' 'JOR' 'SYR' 'SGP' 'BDI'
 'SAU' 'VNM' 'PLW' 'QAT' 'EGY' 'PER' 'MLT' 'MWI' 'ECU' 'MDG' 'ISL' 'UZB'
 'NPL' 'BHS' 'MAC' 'TGO' 'TWN' 'DJI' 'STP' 'KNA' 'ETH' 'IRQ' 'HND' 'RWA'
 'KHM' 'MCO' 'BGD' 'IMN' 'TJK' 'NIC' 'BEN' 'VGB' 'TZA' 'GAB' 'GHA' 'TMP'
 'GLP' 'KEN' 'LIE' 'GNB' 'MNE' 'UMI' 'MYT' 'FRO' 'MMR' 'PAN' 'BFA' 'LBY'
 'MLI' 'NAM' 'BOL' 'PRY' 'BRB' 'ABW' 'AIA' 'SLV' 'DMA' 'PYF' 'GUY' 'LCA'
 'ATA' 'GTM' 'ASM' 'MRT' 'NCL' 'KIR' 'SDN' 'ATF' 'SLE' 'LAO']
--------------------------------------------------
market_segment
['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'
 'Undefined' 'Aviation']
```

```
-------------------------------------------------
distribution_channel
['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']
-------------------------------------------------
reserved_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'P' 'B']
-------------------------------------------------
assigned_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'P' 'L' 'K']
-------------------------------------------------
deposit_type
['No Deposit' 'Refundable' 'Non Refund']
-------------------------------------------------
customer_type
['Transient' 'Contract' 'Transient-Party' 'Group']
-------------------------------------------------
reservation_status
['Check-Out' 'Canceled' 'No-Show']
-------------------------------------------------
```

```
#checking the null values
data.isnull().sum()
```

```
hotel                                0
is_canceled                          0
lead_time                            0
arrival_date_year                    0
arrival_date_month                   0
arrival_date_week_number             0
arrival_date_day_of_month            0
stays_in_weekend_nights              0
stays_in_week_nights                 0
adults                               0
children                             4
babies                               0
meal                                 0
country                            488
market_segment                       0
distribution_channel                 0
is_repeated_guest                    0
previous_cancellations               0
previous_bookings_not_canceled       0
reserved_room_type                   0
assigned_room_type                   0
booking_changes                      0
deposit_type                         0
agent                            16340
company                         112593
days_in_waiting_list                 0
customer_type                        0
adr                                  0
required_car_parking_spaces          0
total_of_special_requests            0
reservation_status                   0
reservation_status_date              0
dtype: int64
```

```
data.isnull().sum()
```

```
hotel                                0
is_canceled                          0
lead_time                            0
arrival_date_year                    0
arrival_date_month                   0
arrival_date_week_number             0
arrival_date_day_of_month            0
stays_in_weekend_nights              0
stays_in_week_nights                 0
adults                               0
children                             4
babies                               0
meal                                 0
country                            488
market_segment                       0
```

```
distribution_channel                  0
is_repeated_guest                     0
previous_cancellations                0
previous_bookings_not_canceled        0
reserved_room_type                    0
assigned_room_type                    0
booking_changes                       0
deposit_type                          0
agent                             16340
company                          112593
days_in_waiting_list                  0
customer_type                         0
adr                                   0
required_car_parking_spaces           0
total_of_special_requests             0
reservation_status                    0
reservation_status_date               0
dtype: int64
```

```
data.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival
count	119390.000000	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	27.165173	
std	0.482918	106.863097	0.707476	13.605138	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	
50%	0.000000	69.000000	2016.000000	28.000000	
75%	1.000000	160.000000	2017.000000	38.000000	
max	1.000000	737.000000	2017.000000	53.000000	

```
data = data[data['adr']<5000]
```

```
data.describe()
```

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival
count	119389.000000	119389.000000	119389.000000	119389.000000	
mean	0.370411	104.011994	2016.156555	27.165292	
std	0.482917	106.863358	0.707479	13.605134	
min	0.000000	0.000000	2015.000000	1.000000	
25%	0.000000	18.000000	2016.000000	16.000000	
50%	0.000000	69.000000	2016.000000	28.000000	
75%	1.000000	160.000000	2017.000000	38.000000	
max	1.000000	737.000000	2017.000000	53.000000	

```
data.head().T
```

	0	1	2	3	4
hotel	Resort Hotel	Resort Hotel	Resort Hotel	Resort Hotel	Resort Hotel
is_canceled	0	0	0	0	0
lead_time	342	737	7	13	14
arrival_date_year	2015	2015	2015	2015	2015
arrival_date_month	July	July	July	July	July
arrival_date_week_number	27	27	27	27	27
arrival_date_day_of_month	1	1	1	1	1
stays_in_weekend_nights	0	0	0	0	0
stays_in_week_nights	0	0	1	1	2
adults	2	2	1	1	2
children	0.0	0.0	0.0	0.0	0.0
babies	0	0	0	0	0
meal	BB	BB	BB	BB	BB
country	PRT	PRT	GBR	GBR	GBR
market_segment	Direct	Direct	Direct	Corporate	Online TA
distribution_channel	Direct	Direct	Direct	Corporate	TA/TO
is_repeated_guest	0	0	0	0	0
previous_cancellations	0	0	0	0	0
previous_bookings_not_canceled	0	0	0	0	0
reserved_room_type	C	C	A	A	A
assigned_room_type	C	C	C	A	A
booking_changes	3	4	0	0	0
deposit_type	No Deposit	No Deposit	No Deposit	No Deposit	No Deposit
agent	NaN	NaN	NaN	304.0	240.0
company	NaN	NaN	NaN	NaN	NaN
days_in_waiting_list	0	0	0	0	0

Model Building

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.linear_model import LinearRegression


data = data.drop(columns=['country'])

#check the unique values
for col in data.describe(include='object').columns:
    print(col)
    print(data[col].unique())
    print('-'*50)
```

```
hotel
['Resort Hotel' 'City Hotel']
--------------------------------------------------
arrival_date_month
['July' 'August' 'September' 'October' 'November' 'December' 'January'
 'February' 'March' 'April' 'May' 'June']
```

```
--------------------------------------------------
meal
['BB' 'FB' 'HB' 'SC' 'Undefined']
--------------------------------------------------
market_segment
['Direct' 'Corporate' 'Online TA' 'Offline TA/TO' 'Complementary' 'Groups'
 'Undefined' 'Aviation']
--------------------------------------------------
distribution_channel
['Direct' 'Corporate' 'TA/TO' 'Undefined' 'GDS']
--------------------------------------------------
reserved_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'H' 'L' 'P' 'B']
--------------------------------------------------
assigned_room_type
['C' 'A' 'D' 'E' 'G' 'F' 'I' 'B' 'H' 'P' 'L' 'K']
--------------------------------------------------
deposit_type
['No Deposit' 'Refundable' 'Non Refund']
--------------------------------------------------
customer_type
['Transient' 'Contract' 'Transient-Party' 'Group']
--------------------------------------------------
reservation_status
['Check-Out' 'Canceled' 'No-Show']
--------------------------------------------------
```

```
from sklearn.preprocessing import LabelEncoder

# Create a LabelEncoder for each categorical feature
le_hotel = LabelEncoder()
le_arrival_date_month = LabelEncoder()
le_meal = LabelEncoder()
le_market_segment = LabelEncoder()
le_distribution_channel = LabelEncoder()
le_reserved_room_type = LabelEncoder()
le_assigned_room_type = LabelEncoder()
le_deposit_type = LabelEncoder()
le_customer_type = LabelEncoder()
le_reservation_status = LabelEncoder()

# Fit and transform each feature
data['hotel'] = le_hotel.fit_transform(data['hotel'])
data['arrival_date_month'] = le_arrival_date_month.fit_transform(data['arrival_date_month'])
data['meal'] = le_meal.fit_transform(data['meal'])
data['market_segment'] = le_market_segment.fit_transform(data['market_segment'])
data['distribution_channel'] = le_distribution_channel.fit_transform(data['distribution_channel'])
data['reserved_room_type'] = le_reserved_room_type.fit_transform(data['reserved_room_type'])
data['assigned_room_type'] = le_assigned_room_type.fit_transform(data['assigned_room_type'])
data['deposit_type'] = le_deposit_type.fit_transform(data['deposit_type'])
data['customer_type'] = le_customer_type.fit_transform(data['customer_type'])
data['reservation_status'] = le_reservation_status.fit_transform(data['reservation_status'])
```

```
"""# Convert categorical variables to numerical using Label Encoding
label_encoders = {}
categorical_cols = ['hotel', 'arrival_date_month', 'meal', 'country', 'market_segment', 'distribution_channel',
                    'reserved_room_type', 'assigned_room_type', 'deposit_type', 'customer_type', 'reservation_status']
for col in categorical_cols:
    le = LabelEncoder()
    data[col] = le.fit_transform(data[col])
    label_encoders[col] = le"""
```

```
'# Convert categorical variables to numerical using Label Encoding\nlabel_encoders = {}
\ncategorical_cols = ['hotel', 'arrival_date_month', 'meal', 'country', 'market_segmen
t', 'distribution_channel',\n                    'reserved_room_type', 'assigned_room_t
ype', 'deposit_type', 'customer_type', 'reservation_status']\nfor col in categorical_co
ls:\n    le = LabelEncoder()\n    data[col] = le.fit_transform(data[col])\n    label_en
```

```
data.head()
```

.ting_list	customer_type	adr	required_car_parking_spaces	total_of_special_requests	r
0	2	0.0	0	0	
0	2	0.0	0	0	
0	2	75.0	0	0	
0	2	75.0	0	0	
0	2	98.0	0	1	

```
missing_values = data.isnull().sum()
print(missing_values)
```

```
hotel                                0
is_canceled                          0
lead_time                            0
arrival_date_year                    0
arrival_date_month                   0
arrival_date_week_number             0
arrival_date_day_of_month            0
stays_in_weekend_nights              0
stays_in_week_nights                 0
adults                               0
children                             4
babies                               0
meal                                 0
market_segment                       0
distribution_channel                 0
is_repeated_guest                    0
previous_cancellations               0
previous_bookings_not_canceled       0
reserved_room_type                   0
assigned_room_type                   0
booking_changes                      0
deposit_type                         0
agent                            16340
company                         112592
days_in_waiting_list                 0
customer_type                        0
adr                                  0
required_car_parking_spaces          0
total_of_special_requests            0
reservation_status                   0
reservation_status_date              0
dtype: int64
```

```
data.drop(['agent', 'company','children'], axis=1, inplace=True)
```

```
data.drop(['reservation_status_date'], axis=1, inplace=True)
```

```
missing_values = data.isnull().sum()
print(missing_values)
```

```
hotel                             0
is_canceled                       0
lead_time                         0
arrival_date_year                 0
arrival_date_month                0
arrival_date_week_number          0
arrival_date_day_of_month         0
stays_in_weekend_nights           0
stays_in_week_nights              0
adults                            0
babies                            0
meal                              0
market_segment                    0
distribution_channel              0
is_repeated_guest                 0
previous_cancellations            0
previous_bookings_not_canceled    0
```

```
reserved_room_type               0
assigned_room_type               0
booking_changes                  0
deposit_type                     0
days_in_waiting_list             0
customer_type                    0
adr                              0
required_car_parking_spaces      0
total_of_special_requests        0
reservation_status               0
dtype: int64
```

```
x = data.drop("is_canceled",axis=1)
y = data['is_canceled']
#split in train and test
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=0)
```

```
lr_clf = LogisticRegression()
lr_clf.fit(x_train,y_train)
```

```
▾ LogisticRegression
LogisticRegression()
```

```
y_pred = lr_clf.predict(x_test)
y_pred[20:25] # Y predicted
```

```
array([0, 0, 0, 0, 0])
```

```
y_test[20:25] # y actual
```

```
23861     0
11674     1
17890     0
33804     0
104569    0
Name: is_canceled, dtype: int64
```

```
cnf_matrix = confusion_matrix(y_test,y_pred)
print("Confusion matrix:\n",cnf_matrix)
```

```
Confusion matrix:
 [[19540  2861]
 [ 3915  9501]]
```

```
clf_report = classification_report(y_test,y_pred)
print("classification_report is :\n",clf_report)
```

```
classification_report is :
               precision    recall  f1-score   support

           0       0.83      0.87      0.85     22401
           1       0.77      0.71      0.74     13416

    accuracy                           0.81     35817
   macro avg       0.80      0.79      0.79     35817
weighted avg       0.81      0.81      0.81     35817
```

```
y_pred_prob = lr_clf.predict_proba(x_test)
y_pred_prob
```

```
array([[0.64741984, 0.35258016],
       [0.22668723, 0.77331277],
       [0.11508764, 0.88491236],
       ...,
       [0.903019  , 0.096981  ],
           [0.9930613 , 0.0069387 ],
           [0.36011693, 0.63988307]])
```

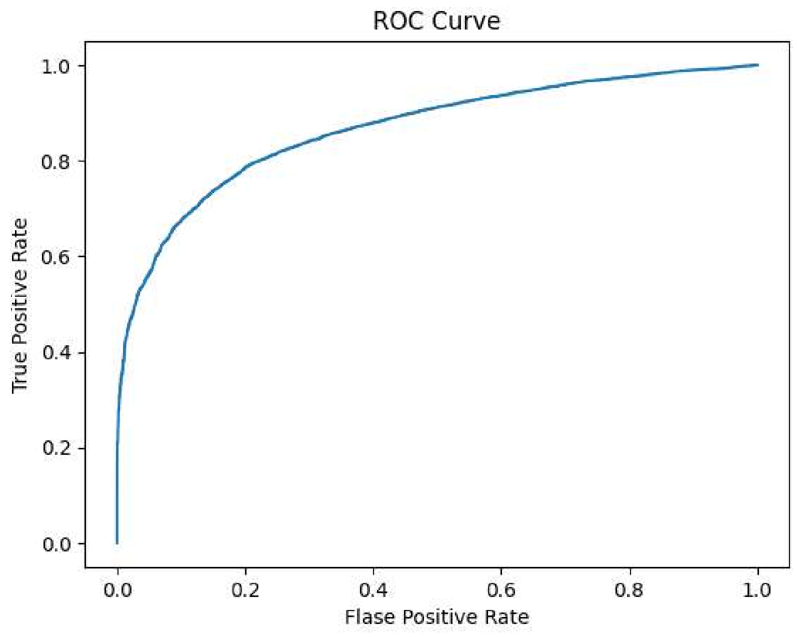
```
fpr,tpr,thresh = roc_curve(y_test,y_pred_prob[:,1])
```

```
thresh
```

```
     array([1.99889973e+00, 9.98899727e-01, 9.95337658e-01, ...,
            1.18493472e-04, 1.15576312e-04, 2.37022228e-06])
```

```
plt.title("ROC Curve")
plt.plot(fpr,tpr)
plt.xlabel("Flase Positive Rate")
plt.ylabel("True Positive Rate")
```

```
     Text(0, 0.5, 'True Positive Rate')
```



```
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Train a Random Forest Classifier
clf = RandomForestClassifier(random_state=42)
clf.fit(x_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(x_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
confusion = confusion_matrix(y_test, y_pred)
report = classification_report(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print("Confusion Matrix:\n", confusion)
print("Classification Report:\n", report)
```

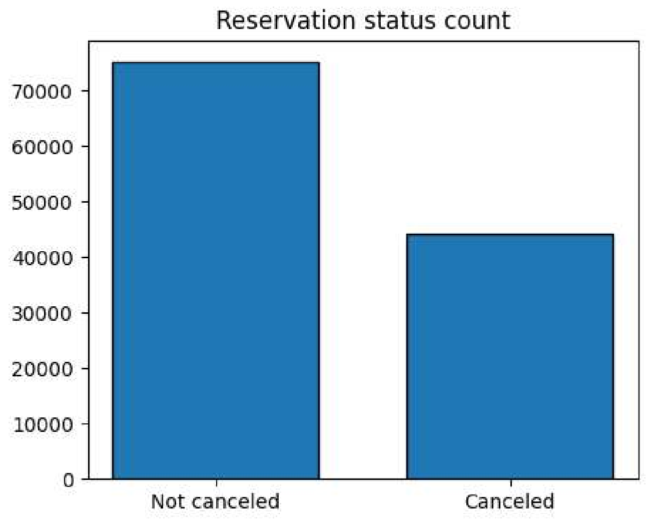
```
     Accuracy: 1.00
     Confusion Matrix:
      [[22401     0]
      [    1 13415]]
```

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

           0       1.00      1.00      1.00     22401
           1       1.00      1.00      1.00     13416

    accuracy                           1.00     35817
   macro avg       1.00      1.00      1.00     35817
weighted avg       1.00      1.00      1.00     35817
```

Data Analysis and Visualization

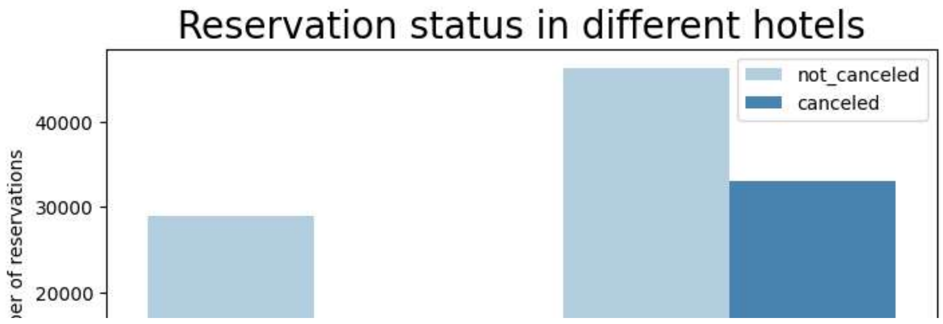
```
data['is_canceled'].value_counts()
cancelled_perce=data['is_canceled'].value_counts(normalize=True)
print(cancelled_perce)

plt.figure(figsize = (5,4))
plt.title('Reservation status count')
plt.bar(['Not canceled','Canceled'],data['is_canceled'].value_counts(),edgecolor='k',width=0.7)
plt.show()
```

```
0    0.629589
1    0.370411
Name: is_canceled, dtype: float64
```



here is clear that 37% people canceled the booking that is high percentage

```
plt.figure(figsize = (8,4))
ax = sns.countplot(x = 'hotel', hue = 'is_canceled',data=data, palette= 'Blues')
legend_labels,_ = ax. get_legend_handles_labels()
plt.title('Reservation status in different hotels', size=20)
plt.xlabel('hotel')
plt.ylabel('number of reservations')
plt.legend(['not_canceled','canceled'])
plt.show()
```



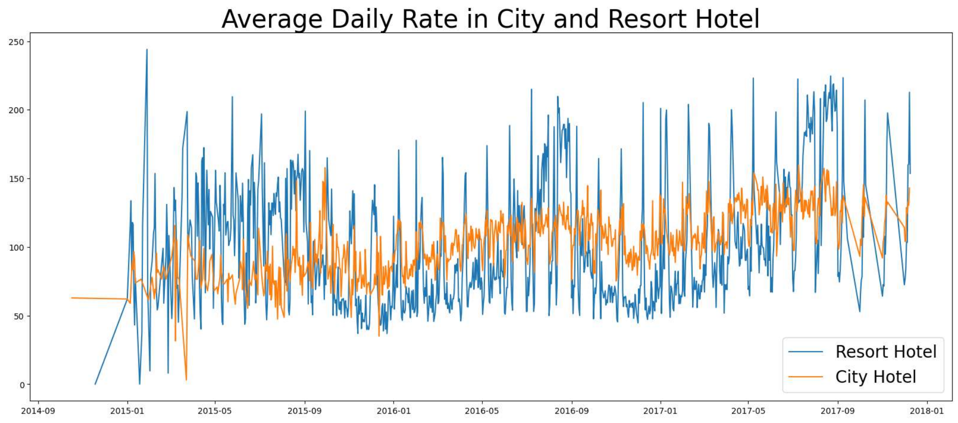
```
resort_hotel = data[data['hotel'] == 'Resort Hotel']
resort_hotel['is_canceled'].value_counts(normalize = True)

     0    0.722366
     1    0.277634
     Name: is_canceled, dtype: float64
```

hotel

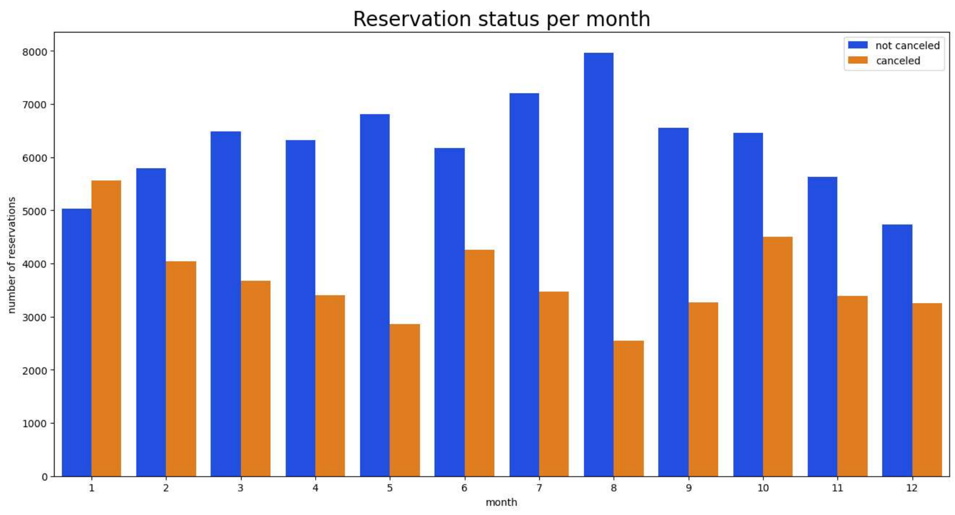
```
city_hotel = data[data['hotel'] == 'City Hotel']
city_hotel['is_canceled'].value_counts(normalize = True)

     0    0.582738
     1    0.417262
     Name: is_canceled, dtype: float64
```

```
resort_hotel = resort_hotel.groupby('reservation_status_date')[['adr']].mean()
city_hotel = city_hotel.groupby('reservation_status_date')[['adr']].mean()
```

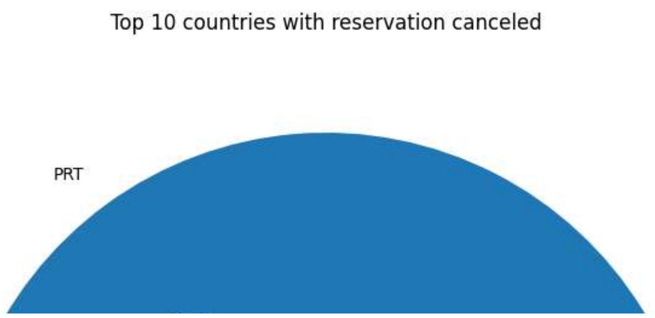
```
plt.figure(figsize = (20,8))
plt.title('Average Daily Rate in City and Resort Hotel', fontsize=30)
plt.plot(resort_hotel.index, resort_hotel['adr'], label = 'Resort Hotel')
plt.plot(city_hotel.index, city_hotel['adr'], label = 'City Hotel')
plt.legend(fontsize = 20)
plt.show()
```



```
data['month'] = data['reservation_status_date'].dt.month
plt.figure(figsize = (16,8))
ax = sns.countplot(x='month',hue='is_canceled',data=data,palette='bright')
legend_labels,_ = ax.get_legend_handles_labels()
ax.legend(bbox_to_anchor = (1,1))
plt.title('Reservation status per month',size=20)
plt.xlabel('month')
plt.ylabel('number of reservations')
plt.legend(['not canceled','canceled'])
plt.show()
```



```
cancelled_data = data[data['is_canceled'] == 1]
top_10_country = cancelled_data['country'].value_counts()[:10]
plt.figure(figsize=(10,10))
plt.title('Top 10 countries with reservation canceled')
plt.pie(top_10_country, autopct='%.2f',labels = top_10_country.index)
plt.show()
```



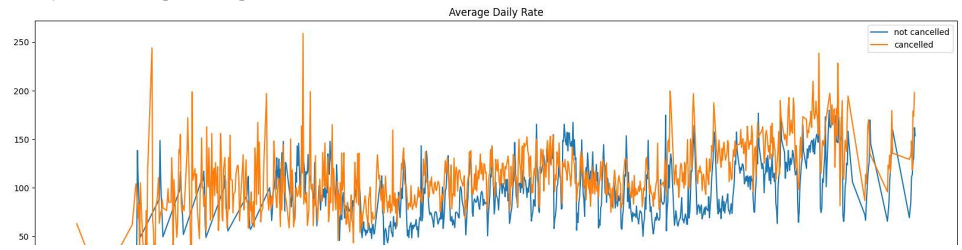
```
data['market_segment'].value_counts()

    Online TA        56477
    Offline TA/TO    24218
    Groups           19811
    Direct           12606
    Corporate         5295
    Complementary      743
    Aviation           237
    Undefined            2
    Name: market_segment, dtype: int64
```

```
data['market_segment'].value_counts(normalize=True)

    Online TA        0.473050
    Offline TA/TO    0.202850
    Groups           0.165937
    Direct           0.105588
    Corporate        0.044351
    Complementary    0.006223
    Aviation         0.001985
    Undefined        0.000017
    Name: market_segment, dtype: float64
```

```
cancelled_data['market_segment'].value_counts(normalize=True)

    Online TA        0.468964
    Groups           0.273545
    Offline TA/TO    0.187911
    Direct           0.043733
    Corporate        0.022432
    Complementary    0.002193
    Aviation         0.001176
    Undefined        0.000045
    Name: market_segment, dtype: float64
```

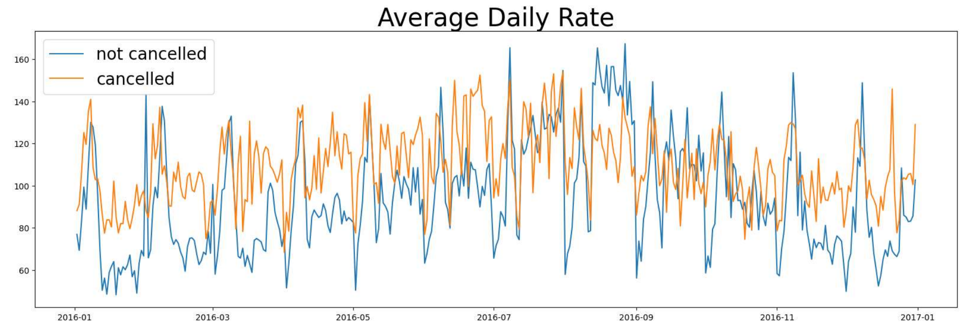
```
cancelled_data_adr = cancelled_data.groupby('reservation_status_date')[['adr']].mean()
cancelled_data_adr.reset_index(inplace=True)
cancelled_data_adr.sort_values('reservation_status_date',inplace=True)

not_cancelled_data = data[data['is_canceled'] == 0]
not_cancelled_data_adr = not_cancelled_data.groupby('reservation_status_date')[['adr']].mean()
not_cancelled_data_adr.reset_index(inplace=True)
not_cancelled_data_adr.sort_values('reservation_status_date',inplace=True)

plt.figure(figsize = (20,6))
plt.title('Average Daily Rate')
plt.plot(not_cancelled_data_adr['reservation_status_date'],not_cancelled_data_adr['adr'],label='not cancelled')
plt.plot(cancelled_data_adr['reservation_status_date'],cancelled_data_adr['adr'],label='cancelled')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x7fe5956ec730>
```



```
cancelled_data_adr = cancelled_data_adr[(cancelled_data_adr['reservation_status_date']>'2016') & (cancelled_data_adr['reservation_sta
not_cancelled_data_adr = not_cancelled_data_adr[(not_cancelled_data_adr['reservation_status_date']>'2016') & (not_cancelled_data_adr[


plt.figure(figsize = (20,6))
plt.title('Average Daily Rate',fontsize=30)
plt.plot(not_cancelled_data_adr['reservation_status_date'],not_cancelled_data_adr['adr'],label='not cancelled')
plt.plot(cancelled_data_adr['reservation_status_date'],cancelled_data_adr['adr'],label='cancelled')
plt.legend(fontsize=20)
```

```
<matplotlib.legend.Legend at 0x7fe59566fdc0>
```


Average Daily Rate
not cancelled
cancelled