October 4, 2020

Hotel booking demand

Intrudction

Here is a data set about hotel booking demand which I would like to explore and to use machine learning methodology.
This data set includes features about the type of hotel booked, the lead time before the vacation, the number of stays and etc.

This is the order of my work:

- First exploring
- Preprocessing
- Supervised learning
- Unsupervised learning

In [2]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

Let's read the data set.

In [3]:
```
try:
    hotel_data = pd.read_csv('C:/Users/Matan/Documents/Python/hotel_bookings.csv')
except: print("Something got wrong")
```

In [4]:
```
hotel_data
```

Out[4]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_
0	Resort Hotel	0	342	2015	July	27	1	
1	Resort Hotel	0	737	2015	July	27	1	
2	Resort Hotel	0	7	2015	July	27	1	
3	Resort Hotel	0	13	2015	July	27	1	
4	Resort Hotel	0	14	2015	July	27	1	
...	...	...	...	...	...	...	...	
119385	City Hotel	0	23	2017	August	35	30	
119386	City Hotel	0	102	2017	August	35	31	
119387	City Hotel	0	34	2017	August	35	31	
119388	City Hotel	0	109	2017	August	35	31	
119389	City Hotel	0	205	2017	August	35	29	

119390 rows × 32 columns

The data is ready.

First exploring

In [5]:
```
print("The number number of rows in hotel data are", hotel_data.shape[0])
print("The number of columns in hotel data are", hotel_data.shape[1])
```

```
The number number of rows in hotel data are 119390
The number of columns in hotel data are 32
```

In [6]:
```
hotel_data.columns
```

Out[6]:
```
Index(['hotel', 'is_canceled', 'lead_time', 'arrival_date_year',
       'arrival_date_month', 'arrival_date_week_number',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'children', 'babies', 'meal',
       'country', 'market_segment', 'distribution_channel',
       'is_repeated_guest', 'previous_cancellations',
       'previous_bookings_not_canceled', 'reserved_room_type',
       'assigned_room_type', 'booking_changes', 'deposit_type', 'agent',
       'company', 'days_in_waiting_list', 'customer_type', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests',
       'reservation_status', 'reservation_status_date'],
      dtype='object')
```

In [7]:
```
hotel_data.dtypes
```

Out[7]:
```
hotel                              object
is_canceled                         int64
lead_time                           int64
arrival_date_year                   int64
arrival_date_month                 object
arrival_date_week_number            int64
arrival_date_day_of_month           int64
stays_in_weekend_nights             int64
stays_in_week_nights                int64
adults                              int64
children                          float64
babies                              int64
meal                               object
country                            object
market_segment                     object
distribution_channel               object
is_repeated_guest                   int64
previous_cancellations              int64
previous_bookings_not_canceled      int64
reserved_room_type                 object
assigned_room_type                 object
booking_changes                     int64
deposit_type                       object
agent                             float64
company                           float64
days_in_waiting_list                int64
customer_type                      object
adr                               float64
required_car_parking_spaces         int64
total_of_special_requests           int64
reservation_status                 object
reservation_status_date            object
dtype: object
```

In [8]:
```
hotel_data['country'].drop_duplicates().count()
```

Out[8]: 177

In [9]:
```
hotel_data['distribution_channel'].drop_duplicates().count()
```

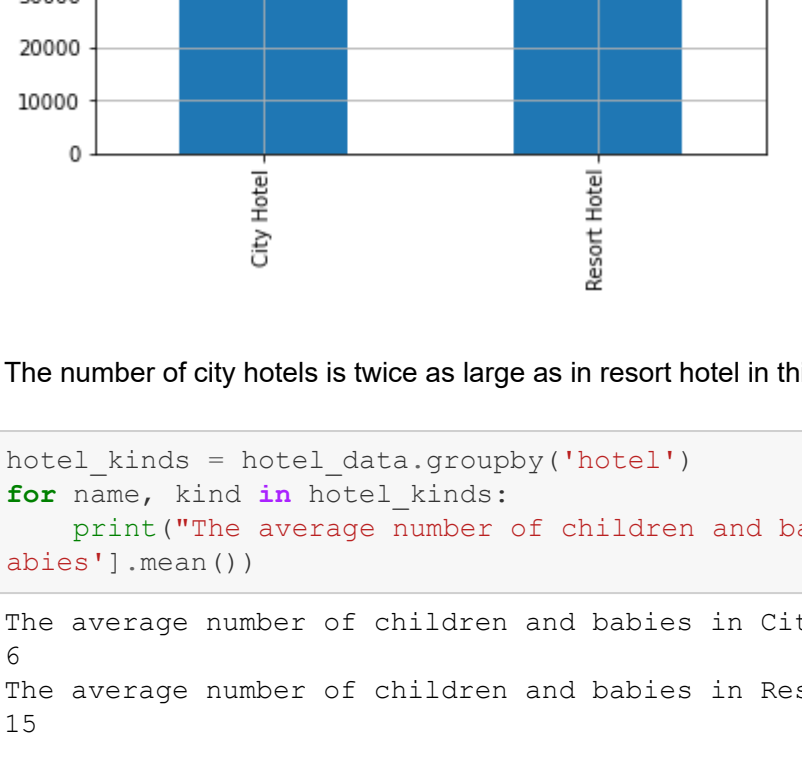
Out[9]: 5

In [10]:
```
hotel_data['hotel'].drop_duplicates()
```

Out[10]:
```
0        Resort Hotel
40060      City Hotel
Name: hotel, dtype: object
```

There are only two types of hotels.

In [11]:
```
hotel_data['hotel'].value_counts().plot(kind="bar", title="Number of Resort and City Hotels")
plt.grid()
```



The number of city hotels is twice as large as in resort hotel in this data set.

In [12]:
```
hotel_kinds = hotel_data.groupby('hotel')
for name, kind in hotel_kinds:
    print("The average number of children and babies in",name, "are:", kind['children'].mean(), kind['b
abies'].mean())
```

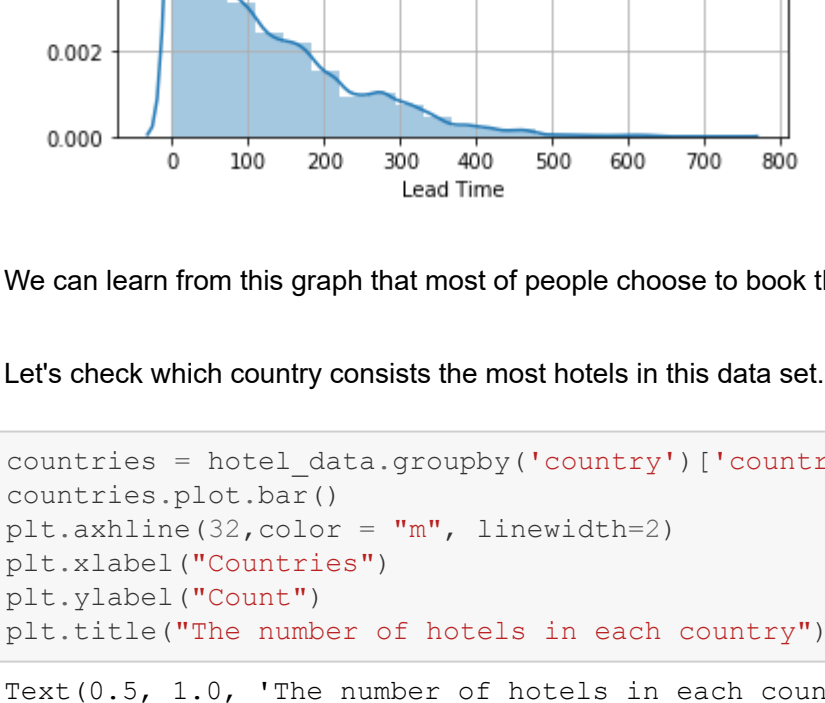
```
The average number of children and babies in City Hotel are: 0.09136979048483473 0.004941384091768561
6
The average number of children and babies in Resort Hotel are: 0.12868197703444834 0.0139041437843235
15
```

We can learn from this result that children and babies are more likely to go to resort hotels.

Now let's check the distribution of the number of days that elapsed between the entering date of the booking and the arrival date.

In [13]:
```
sns.distplot(hotel_data['lead_time'], bins=20)
plt.grid()
plt.xlabel('Lead Time')
plt.ylabel("Count")
plt.title("The Distribution Of The Days Before The Vacation")
```

Out[13]: Text(0.5, 1.0, 'The Distribution Of The Days Before The Vacation')

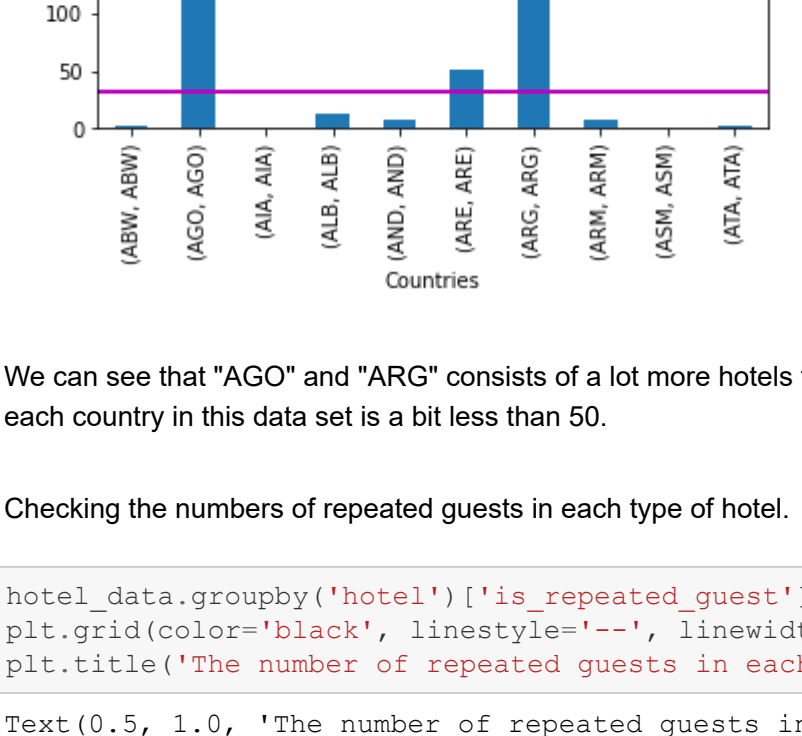


We can learn from this graph that most of people choose to book their hotel up to 100 days from their arrival date.

Let's check which country consists the most hotels in this data set.

In [14]:
```
countries = hotel_data.groupby('country')['country'].value_counts().head(10)
countries.plot.bar()
plt.axhline(32,color = "m", linewidth=2)
plt.xlabel("Countries")
plt.ylabel("Count")
plt.title("The number of hotels in each country")
```

Out[14]: Text(0.5, 1.0, 'The number of hotels in each country')

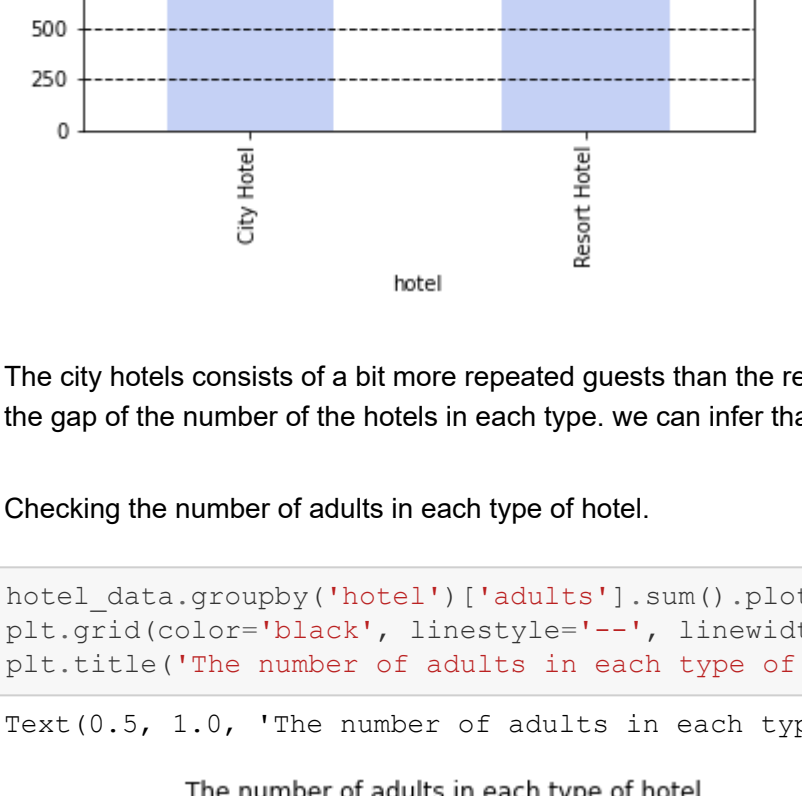


We can see that "AGO" and "ARG" consists of a lot more hotels than the others in this data set. Moreover, The average number of hotels in each country in this data set is a bit less than 50.

Checking the numbers of repeated guests in each type of hotel.

In [15]:
```
hotel_data.groupby('hotel')['is_repeated_guest'].sum().plot(kind='bar',color='royalblue', alpha=0.3)
plt.grid(color='black', linestyle='--', linewidth=1, axis='y', alpha=0.9)
plt.title('The number of repeated guests in each type of hotel')
```

Out[15]: Text(0.5, 1.0, 'The number of repeated guests in each type of hotel')

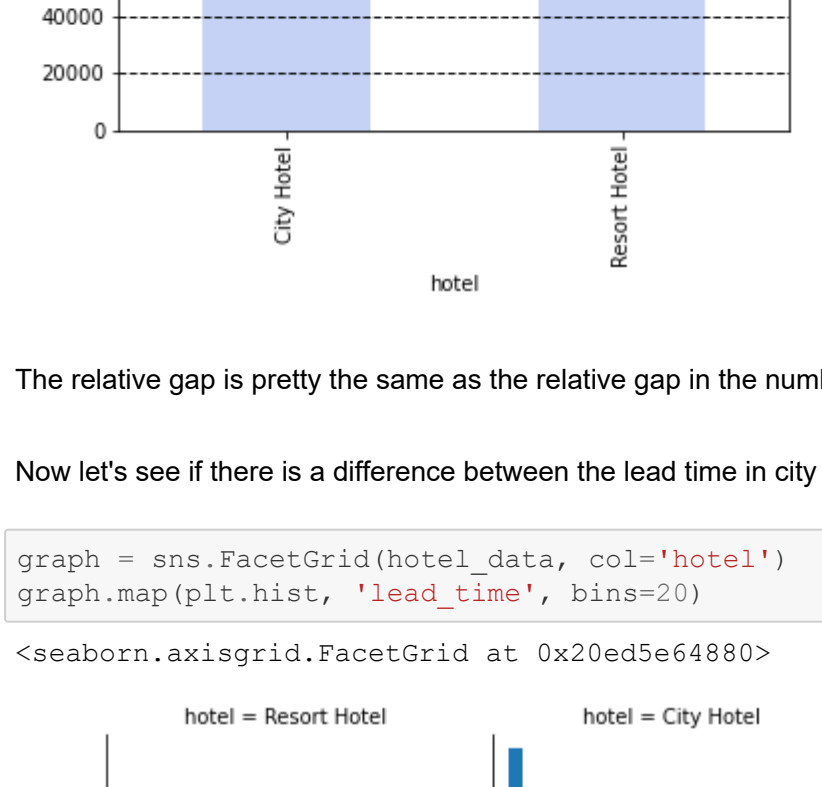


The city hotels consists of a bit more repeated guests than the resort hotel. However, the gap between those kinds of hotel is smaller than the gap of the number of the hotels in each type. we can infer that guests are more likely visit again in resort hotels and not in city hotels.

Checking the number of adults in each type of hotel.

In [16]:
```
hotel_data.groupby('hotel')['adults'].sum().plot(kind='bar',color='royalblue', alpha=0.3)
plt.grid(color='black', linestyle='--', linewidth=1, axis='y', alpha=0.9)
plt.title('The number of adults in each type of hotel')
```

Out[16]: Text(0.5, 1.0, 'The number of adults in each type of hotel')

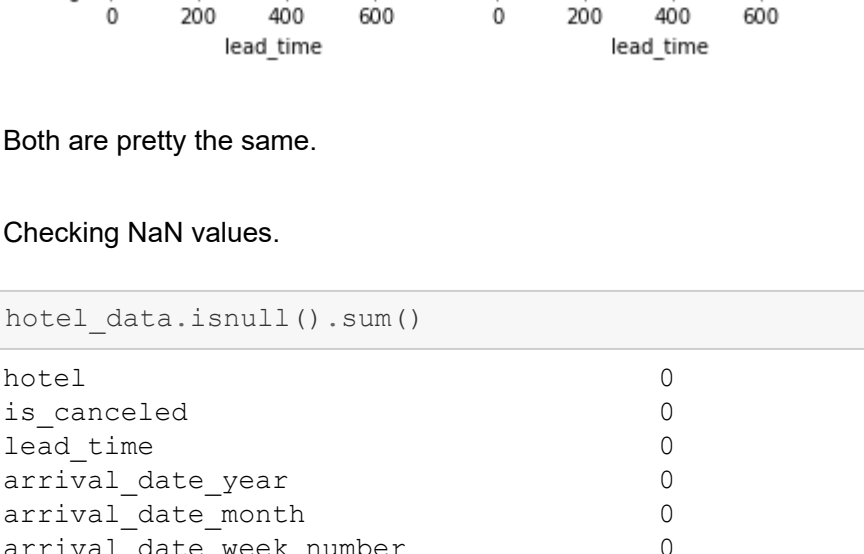


The relative gap is pretty the same as the relative gap in the number of hotels consists of the data set.

Now let's see if there is a difference between the lead time in city hotels to resort hotels.

In [17]:
```
graph = sns.FacetGrid(hotel_data, col='hotel')
graph.map(plt.hist, 'lead_time', bins=20)
```

Out[17]: <seaborn.axisgrid.FacetGrid at 0x20ed5e64880>



Both are pretty the same.

Checking NaN values.

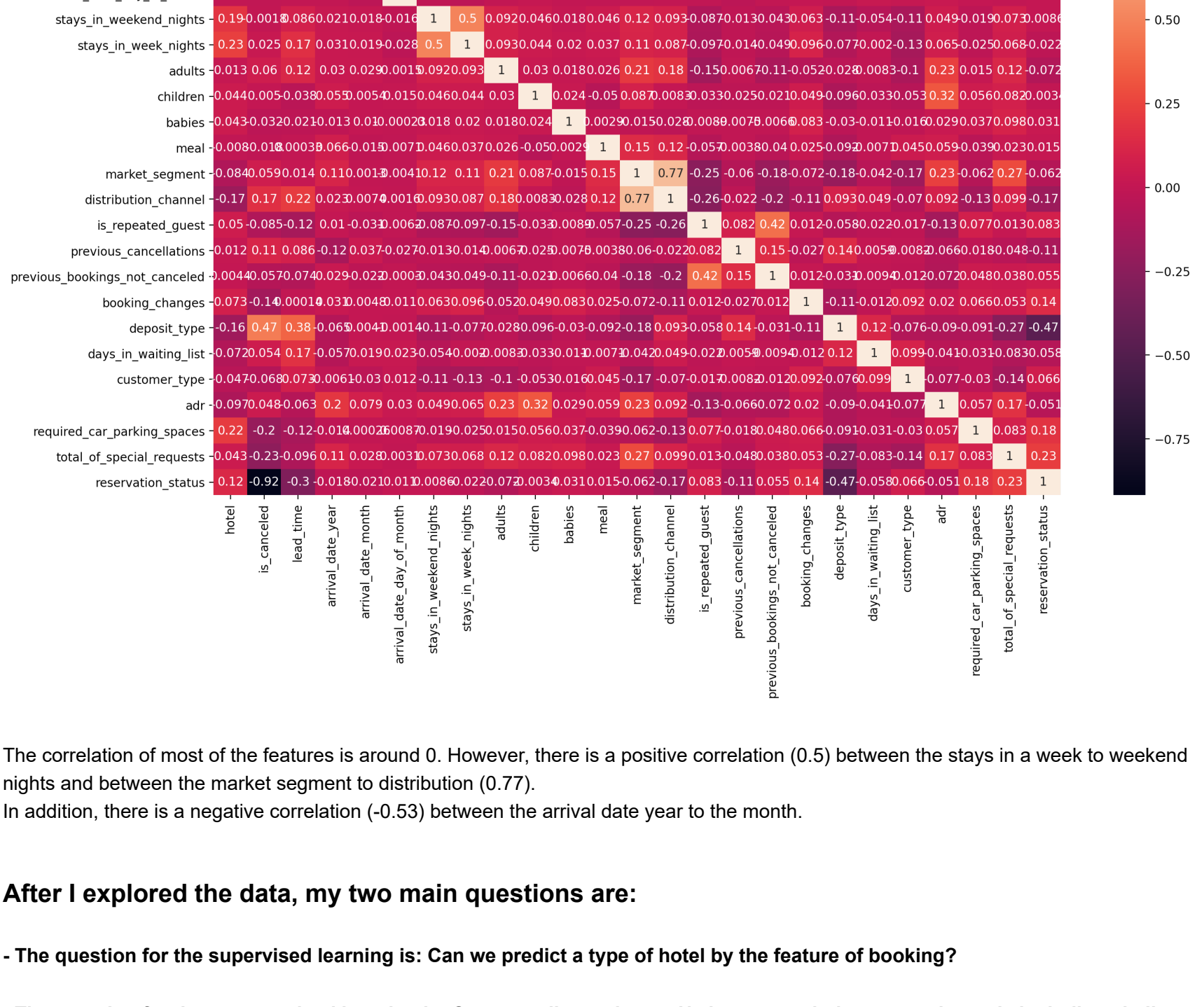
In [18]:
```
hotel_data.isnull().sum()
```

Out[18]:
```
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
```

Most of the values in the company column are missing.
13.68% of the values in the agent column are missing as well.

Now let's check if there is correlation between the features.

In [81]:
```
sns.heatmap(hotel_data.corr(), annot = True)
plt.title('Heatmap', fontsize = 20)
plt.rcParams['figure.figsize'] = [16, 12]
plt.rcParams['figure.dpi'] = 200
```



The correlation of most of the features is around 0. However, there is a positive correlation (0.5) between the stays in a week to weekend nights and between the market segment to distribution (0.77).
In addition, there is a negative correlation (-0.53) between the arrival date year to the month.

After I explored the data, my two main questions are:

- The question for the supervised learning is: Can we predict a type of hotel by the feature of booking?

- The question for the unsupervised learning is: Can we split our data to N clusters such that every cluster is including similar booking features?

Preprocessing

In [19]:
```
from sklearn import preprocessing
```

First, I will delete unnecessary columns and rows which consists of NaN values.

In [20]:
```
hotel_data = hotel_data.drop(columns=['arrival_date_week_number','country','reserved_room_type','assign
ed_room_type','agent','company','reservation_status_date'])
hotel_data = hotel_data.dropna(subset=['children'])
```

Deleting explantion

Columns:

- Arrival date week number - Two many different values that couldn't help to solve our main questions.
- Country - Two many different countries which can cause us to overfitting.
- Reserved and assigned room type - not belong to our main questions.
- Agent - Missing values.
- Company - Missing values.
- Reservation status date - not belong to our main questions.

Changing the name of the months to their numbers.

In [21]:
```
hotel_data['arrival_date_month'] = hotel_data.arrival_date_month.apply(lambda x: x[:3])
months = {'Jan': 1, 'Feb': 2, 'Mar': 3, 'Apr': 4, 'May': 5, 'Jun': 6, 'Jul': 7, 'Aug': 8, 'Sep': 9, 'Oc
t': 10, 'Nov': 11, 'Dec': 12}
hotel_data.arrival_date_month = hotel_data.arrival_date_month.map(months)
```

Now I will do the preprocessing. One data will fit supervised learning and the other to unsupervised.

In [23]:
```
def first_preprocessing(data_frame, supervised_or_unsupervised):
    try:
        new_data = data_frame
        if supervised_or_unsupervised == 'supervised':
            encoders = {'hotel': preprocessing.LabelEncoder(),
                        'meal': preprocessing.LabelEncoder(),
                        'market_segment': preprocessing.LabelEncoder(),
                        'distribution_channel': preprocessing.LabelEncoder(),
                        'deposit_type': preprocessing.LabelEncoder(),
                        'customer_type': preprocessing.LabelEncoder(),
                        'reservation_status': preprocessing.LabelEncoder()}
            encoder_columns_list = ['hotel','meal','market_segment','distribution_channel','deposit_typ
e','customer_type','reservation_status']
            for column in encoder_columns_list:
                new_data[column] = encoders[column].fit_transform(new_data[column].astype(str))
        elif supervised_or_unsupervised == 'unsupervised':
            new_data = pd.get_dummies(new_data)
            columns = new_data.columns
            min_max_scaler = preprocessing.MinMaxScaler()
            new_data = min_max_scaler.fit_transform(new_data)
            new_data = pd.DataFrame(new_data)
            new_data.columns = columns
        return new_data
    except: print("Somthing got wrong - first_preprocessing")
```

Preprocessing explantion:

Supervised data:

- Changing the columns which contain object to int values. #### Unsupervised data:
- Using "One Hot Vector" to the columns that contain object values.
- Normalize the whole data set.

In [24]:
```
hotel_supervised = first_preprocessing(hotel_data,'supervised')
hotel_supervised
```

Out[24]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_day_of_month	stays_in_weekend_nights	stays_ir
0	1	0	342	2015	7	1	0	
1	1	0	737	2015	7	1	0	
2	1	0	7	2015	7	1	0	
3	1	0	13	2015	7	1	0	
4	1	0	14	2015	7	1	0	
...	...	...	...	...	...	...	...	
119385	0	0	23	2017	8	30	2	
119386	0	0	102	2017	8	31	2	
119387	0	0	34	2017	8	31	2	
119388	0	0	109	2017	8	31	2	
119389	0	0	205	2017	8	29	2	

119386 rows × 25 columns

Supervised data is ready.

In [25]:
```
hotel_unsupervised = first_preprocessing(hotel_data,'unsupervised')
hotel_unsupervised
```

Out[25]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_day_of_month	stays_in_weekend_nights	stays_ir
0	1.0	0.0	0.464043	0.0	0.545455	0.000000	0.000000	
1	1.0	0.0	1.000000	0.0	0.545455	0.000000	0.000000	
2	1.0	0.0	0.009498	0.0	0.545455	0.000000	0.000000	
3	1.0	0.0	0.017639	0.0	0.545455	0.000000	0.000000	
4	1.0	0.0	0.018996	0.0	0.545455	0.000000	0.000000	
...	...	...	...	...	...	...	...	
119381	0.0	0.0	0.031208	1.0	0.636364	0.966667	0.105263	
119382	0.0	0.0	0.138399	1.0	0.636364	1.000000	0.105263	
119383	0.0	0.0	0.046133	1.0	0.636364	1.000000	0.105263	
119384	0.0	0.0	0.147897	1.0	0.636364	1.000000	0.105263	
119385	0.0	0.0	0.278155	1.0	0.636364	0.933333	0.105263	

119386 rows × 25 columns

Unsupervised data is ready.

Now let's try to answer the first question:
Can we predict a type of hotel by the feature of booking?
I will use supervised learning with the help of **Naive Bayes Classifier**. This classifier is a statistical classification technique based on Bayes Theorem.
Following the main question, I will try to predict the right type of hotel between the two:

- Resort hotel
- City hotel

In [32]:
```
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, precision_score, recall_score
```

Separating the target column and all the rest features:

In [33]:
```
X = hotel_supervised.drop(columns = ['hotel']).copy()
y = hotel_supervised["hotel"].copy()
```

Building two functions that supposed to split the data for **train** and **test** and create the **Naive Bayes Classifier**:

In [34]:
```
def split_test_train(X, y, test_size):
    try:
        return train_test_split(X, y, test_size=test_size, random_state=0)
    except:
        print("Something got wrong - split_test_train")
```

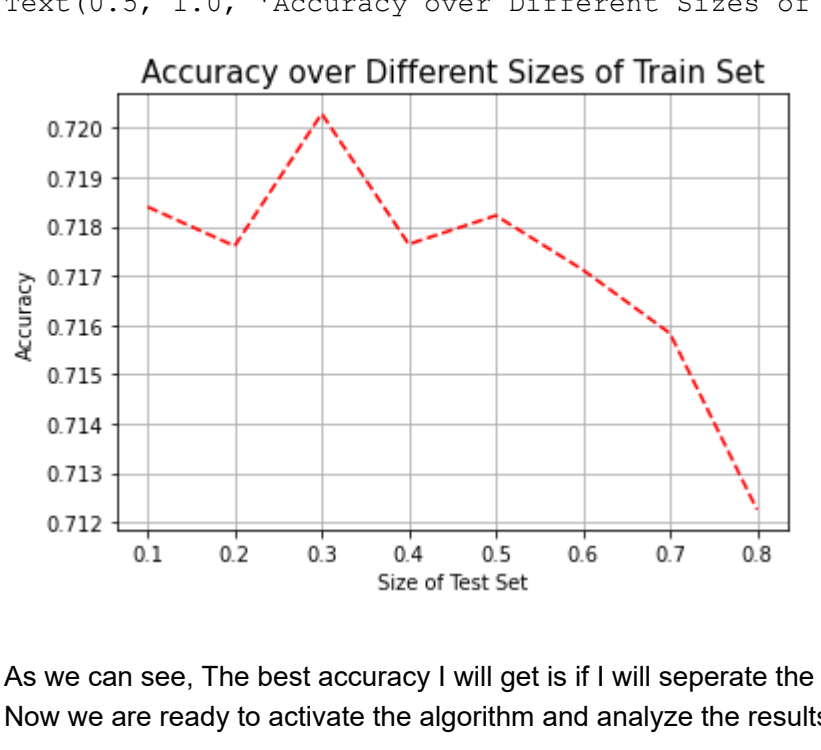
In [35]:
```
def create_naive_bayes_classifier(X, y):
    try:
        model = GaussianNB()
        model.fit(X, y)
        return model
    except:
        print("Something got wrong - create_naive_bayes_classifier")
```

Before I choose the train and the test part, I want to make a plot that can show me what is the best accuracy based on the partition of the train and test.

In [36]:
```
ratios = np.arange(0.1, 0.9, 0.1)
accuracy = []
for ratio in ratios:
    X_train, X_test, y_train, y_test = split_test_train(X, y, ratio)
    model = create_naive_bayes_classifier(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy.append(accuracy_score(y_test, y_pred))
```

In [37]:
```
plt.grid(True)
plt.plot(ratios, accuracy, 'r--')
plt.xlabel('Size of Test Set')
plt.ylabel('Accuracy')
plt.title('Accuracy over Different Sizes of Train Set', fontsize=15)
```

Out[37]: Text(0.5, 1.0, 'Accuracy over Different Sizes of Train Set')



As we can see, The best accuracy I will get is if I will seperate the data to 0.3 test group and 0.7 train group.
Now we are ready to activate the algorithm and analyze the results.

In [38]:
```
X_train, X_test, y_train, y_test = split_test_train(X, y, 0.3)
```

In [39]:
```
model = create_naive_bayes_classifier(X_train, y_train)
```

In [40]:
```
y_pred = model.predict(X_test)
```

In [41]:
```
accuracy_score(y_test, y_pred)
```

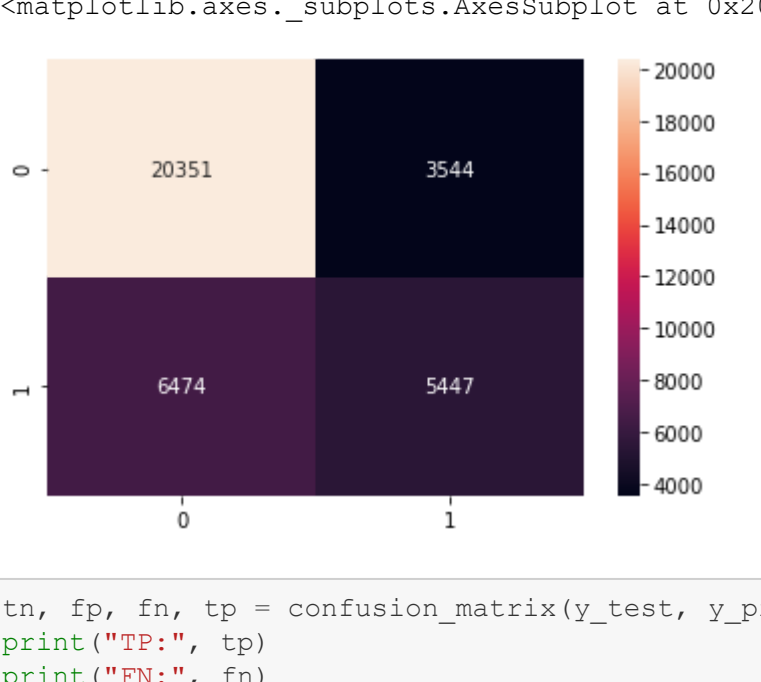
Out[41]: 0.7202926066562431

We received not bad accuracy depends on the starting point which our chances were 50/50.

Let's use a confusion matrix to see if we can find something significant.

In [42]:
```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt="d")
```

Out[42]: <matplotlib.axes._subplots.AxesSubplot at 0x20ed5d45af0>



In [43]:
```
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print("TP:", tp)
print("FN:", fn)
print("FP:", fp)
print("TN:", tn)
```

```
TP: 5447
FN: 6474
FP: 3544
TN: 20351
```

Based of the confusion matrix we can see that it was easier to the algorithm to predict the city hotel(0).

To dig more into the result, I will use two rates:

- **Precision** - this rate using all the points predicted to positive to how many of them are actually positive. In our cases, the positive is the resort hotel.
- **Recall** - this is the ratio of the number correctly predicted positive class (resort hotel) to the total number of positive classes.

In [44]:
```
precision_score(y_test, y_pred)
```

Out[44]: 0.6058280502724948

From the precision score, we learn that when the algorithm predicts resort hotel, in most of the cases he was right (60.5%).

In [45]:
```
recall_score(y_test, y_pred)
```

Out[45]: 0.4569247546346783

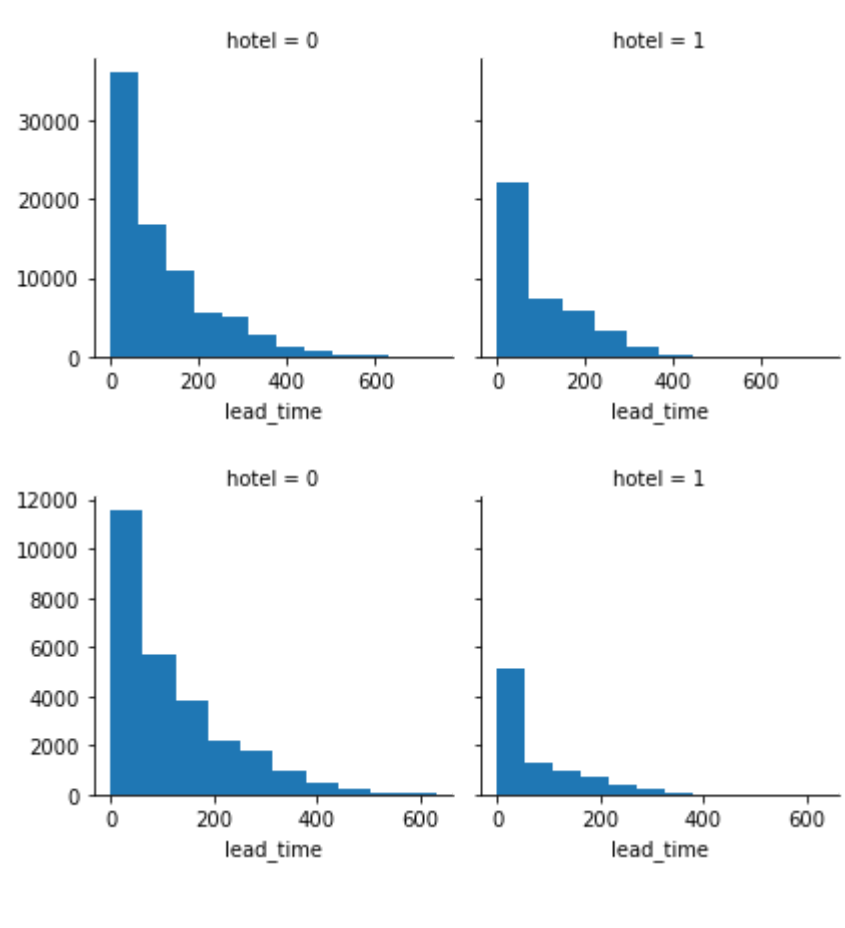
From the recall score we learn that most of the predictions of the actual resort hotel was false (45.6%).

Below are two plots about the difference between the actual data to the predicted one.

In [46]:
```
predicted_data = X_test.copy()
predicted_data["hotel"] = y_pred
```

In [47]:
```
graph = sns.FacetGrid(hotel_supervised, col='hotel')
graph.map(plt.hist, 'lead_time', bins=10)
graph = sns.FacetGrid(predicted_data, col='hotel')
graph.map(plt.hist, 'lead_time', bins=10)
```

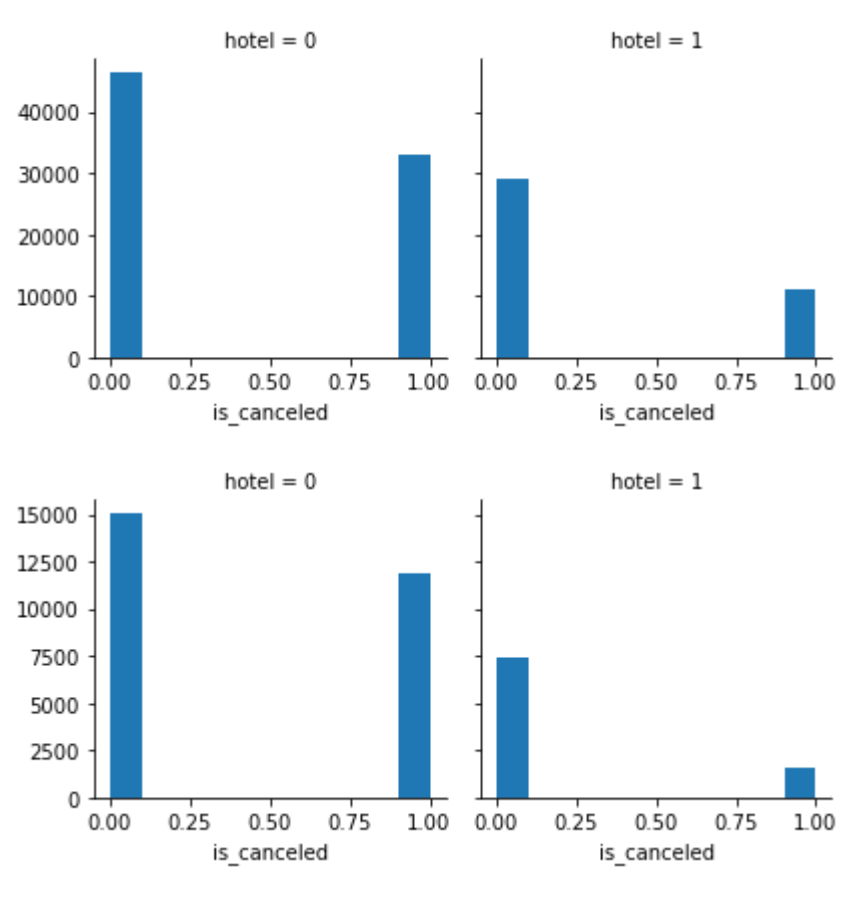
Out[47]: <seaborn.axisgrid.FacetGrid at 0x20ed5dc7a60>



We can see that the distribution of the lead time between the actual data to the predicted data is pretty the same.

In [48]:
```
graph = sns.FacetGrid(hotel_supervised, col='hotel')
graph.map(plt.hist, 'is_canceled', bins=10)
graph = sns.FacetGrid(predicted_data, col='hotel')
graph.map(plt.hist, 'is_canceled', bins=10)
```

Out[48]: <seaborn.axisgrid.FacetGrid at 0x20eda70f040>



The relative gap between the cancellation of the actual data is a bit less than the predicted one.

After I got the result and I analyzed it, I wanted to check if I can optimize the algorithm by keeping the features that clarify the most the prediction (city/resort hotel). Therefore, I am going to use two methods about feature selection and after I will keep only the most important features, I will activate the algorithm one more time to check if the prediction was more accurate.

In [49]:
```
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import RFE
```

Method 1 - Tree based Feature Selection and Random Forest Classification

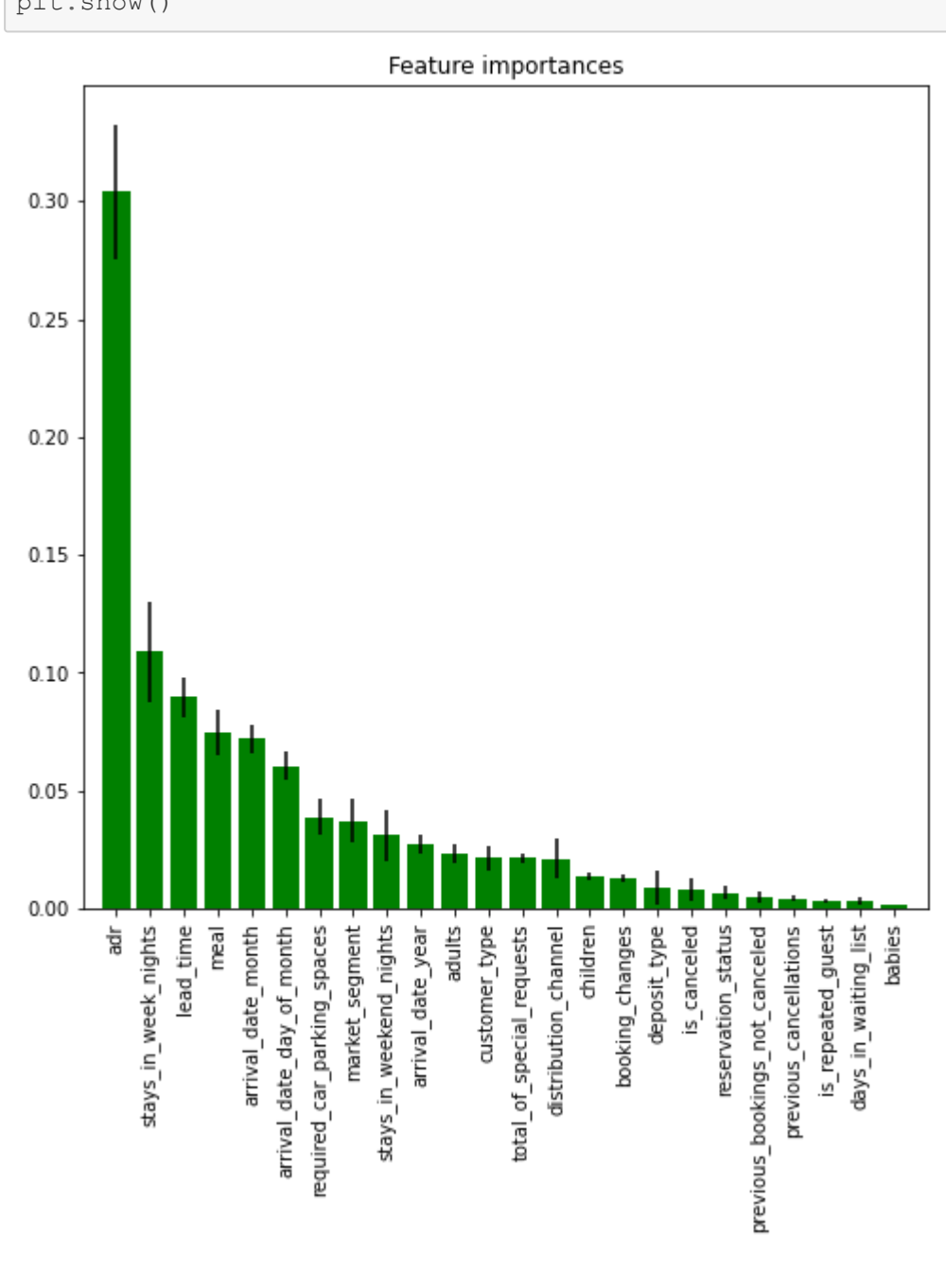
In this method, I will use **Random Forest** algorithm to predict the type of hotel. After that, I will use "feature*importances*" to check the contribution of any feature to the prediction and in the end, I will display it on bar plot.

In [50]:
```
clf_rf_ = RandomForestClassifier(n_estimators=20)
clf_rf_ = clf_rf_.fit(X_train,y_train)
importances = clf_rf_.feature_importances_
std = np.std([tree.feature_importances_ for tree in clf_rf_.estimators_],axis=0)
indices = np.argsort(importances)[::-1]
```

In [51]:
```
print("Feature ranking:")
for f in range(X_train.shape[1]):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))
```

```
Feature ranking:
1. feature 20 (0.303992)
2. feature 6 (0.109053)
3. feature 1 (0.089587)
4. feature 10 (0.074752)
5. feature 3 (0.072111)
6. feature 4 (0.060585)
7. feature 21 (0.038873)
8. feature 11 (0.037319)
9. feature 5 (0.031120)
10. feature 2 (0.027636)
11. feature 7 (0.023431)
12. feature 19 (0.021380)
13. feature 22 (0.021327)
14. feature 12 (0.021293)
15. feature 8 (0.013677)
16. feature 16 (0.012840)
17. feature 17 (0.009036)
18. feature 0 (0.008365)
19. feature 23 (0.006573)
20. feature 15 (0.004977)
21. feature 14 (0.004363)
22. feature 13 (0.003232)
23. feature 18 (0.002927)
24. feature 9 (0.001552)
```

In [52]:
```
plt.figure(1, figsize=(8, 8))
plt.title("Feature importances")
plt.bar(range(X_train.shape[1]), importances[indices],
color="g", yerr=std[indices], align="center")
plt.xticks(range(X_train.shape[1]), X_train.columns[indices],rotation=90)
plt.xlim([-1, X_train.shape[1]])
plt.show()
```



We can learn that the feature "adr" is significantly more clearly to the prediction in comparison to the other features, for instance, this feature almost 3 times bigger (2.877) than the second most important feature.

Method 2 - Recursive feature elimination (RFE) with Random Forest Classification

The algorithm assign weights to each of features. Whose absolute weights are the smallest are pruned from the current set features. That procedure is recursively repeated on the pruned set until the desired number of features.

In [53]:
```
clf_rf_ = RandomForestClassifier(n_estimators=20)
rfe = RFE(estimator=clf_rf_, n_features_to_select=11, step=1)
rfe = rfe.fit(X_train, y_train)
```

In [54]:
```
print('Chosen best 11 feature by RFE:',X_train.columns[rfe.support_])
```

```
Chosen best 11 feature by RFE: Index(['lead_time', 'arrival_date_year', 'arrival_date_month',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'meal', 'market_segment',
       'distribution_channel', 'adr', 'required_car_parking_spaces'],
      dtype='object')
```

This method brought me the same 11 most important features as the first method, therefore I will use only those 11 features and will activate the algorithm once again.

In [55]:
```
new_hotel_supervised = hotel_supervised.drop(columns = ['deposit_type', 'children', 'booking_changes',
'reservation_status', 'is_canceled', 'previous_bookings_not_canceled', 'previous_cancellations','days_i
n_waiting_list','is_repeated_guest','babies','distribution_channel', 'deposit_type'])
```

In [56]:
```
X = new_hotel_supervised.drop(columns = ['hotel']).copy()
y = new_hotel_supervised["hotel"].copy()
```

In [57]:
```
X_train, X_test, y_train, y_test = split_test_train(X, y, 0.3)
```

In [58]:
```
model = create_naive_bayes_classifier(X_train, y_train)
```

In [59]:
```
y_pred = model.predict(X_test)
```

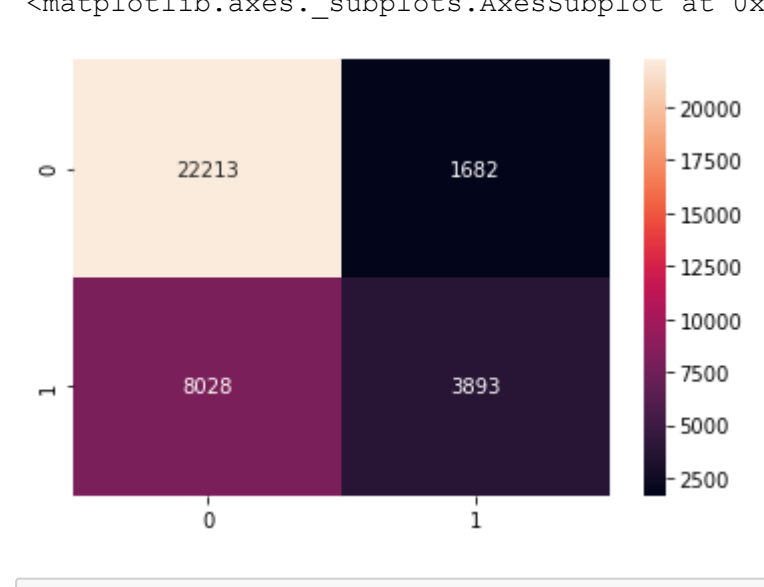
In [60]:
```
accuracy_score(y_test, y_pred)
```

Out[60]:
```
0.7288921152557516
```

I increased the accuracy by 0.008 points - unfortunately not so significantly.

In [61]:
```
sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt="d")
```

Out[61]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20edbdc0d30>
```



In [62]:
```
tn, fp, fn, tp = confusion_matrix(y_test, y_pred).ravel()
print("TP:", tp)
print("FN:", fn)
print("FP:", fp)
print("TN:", tn)
```

```
TP: 3893
FN: 8028
FP: 1682
TN: 22213
```

From the confusion matrix, we can see that the algorithm predicted better the city hotel. However, the prediction of the resulting hotel was less good than before.

In [63]:
```
hotel_type = hotel_supervised.groupby('hotel')[['lead_time', 'arrival_date_year', 'arrival_date_month',
       'arrival_date_day_of_month', 'stays_in_weekend_nights',
       'stays_in_week_nights', 'adults', 'meal', 'adr',
       'required_car_parking_spaces', 'total_of_special_requests']].mean()
hotel_type
```

Out[63]:

	lead_time	arrival_date_year	arrival_date_month	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adu
hotel							
0	109.741106	2016.174344	6.556400	15.787094	0.795187	2.182954	1.850
1	92.675686	2016.121443	6.544583	15.821243	1.189815	3.128732	1.867

In conclusion, in both tests, the accuracy of the algorithm was around 0.72. The feature that has the most clarity about the type of the hotel is the "adr" which is the daily rate, the "adr" of the city hotel is significantly higher than the resort hotel (on average of 10.348 higher). Except this feature, I can say that people book for a longer vacation in resort hotel since the average of the stays in the week and the weekend night is longer in resort hotel than the city hotel. Moreover, the lead time of the booking for the city hotel is longer than the resort hotel (on average of 17.07 longer).
Finally, although the accuracy of the algorithm was above 0.5 (our starting point) the algorithm got difficulties to predict the resort hotel since the number of the city hotel is much higher in this data set than the resort hotel, and the range of almost every feature was very alike to the other type of hotel.

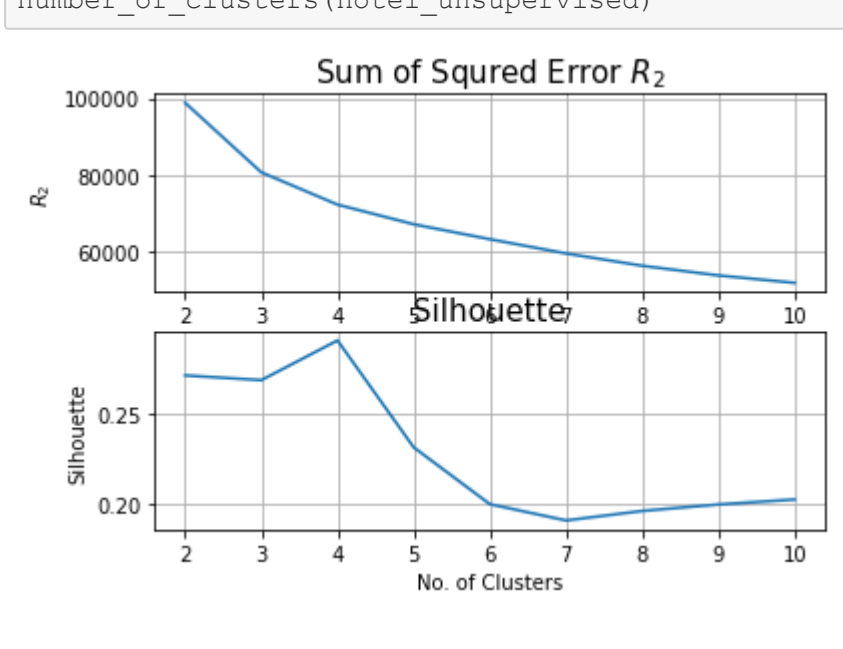
Now let's move one to the next the unsupervised question:
Can we split our data to N clusters such that every cluster is including similar booking features?
To answer that, I will use **KMeans** algorithm. This algorithm is a method of vector quantization, originally from signal processing, that aims to partition **n** observations into **k** clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster.

In [66]:
```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
```

Firstly, I need to find out how many clusters do I need to divide into each data I used. In order to deal with this dilemma, I will build a function which called 'number_of_clusters()'. This function will return us two graphs. The first graph is called "Sum of square error ", in this graph we will see what is the sum of square error as long as the number of clusters increases. Now comes the question of what is the number of clusters that are "suspicious" to be optimal. For answering this question, I will use the "Elbow Method". In this method, when we see a break in some number clusters, we will take that number as an option about how many clusters should we use in this data set. The second graph I will use is called "Silhouette", the silhouette is a measure about how good our algorithm work, his range is between -1 to 1. As long as the value is close to 1 we can say that the clustering was good. In our graph, I will wish to choose the number of clusters by the best silhouette I got. I need to find the best cluster by a combination of those two graphs, a number of clusters which has a good "Elbow break" and ones that has the best silhouette.

In [67]:
```
def number_of_clusters(data_frame):
    try:
        sum_squared = []
        silhouette = []
        for i in range(2, 11):
            kmeans = KMeans(n_clusters=i, init='k-means++')
            kmeans.fit(data_frame)
            sum_squared.append(kmeans.inertia_)
            silhouette.append(silhouette_score(data_frame, kmeans.labels_))
        x1 = (range(2, 11))
        x2 = (range(2, 11))
        y1 = sum_squared
        y2 = silhouette
        plt.subplot(2, 1, 1)
        plt.plot(x1, y1)
        plt.title(r'Sum of Squred Error ${R_2}$', fontsize=15)
        plt.grid()
        plt.ylabel(r'${R_2}$')
        plt.subplot(2, 1, 2)
        plt.plot(x2, y2)
        plt.title('Silhouette', fontsize=15)
        plt.xlabel('No. of Clusters')
        plt.ylabel('Silhouette')
        plt.grid()
        plt.show()
    except:
        print("Something got wrong - number_of_clusters")
```

In [68]:
```
number_of_clusters(hotel_unsupervised)
```



As we can see, in the first plot, the sharper elbow break is when I divide the data into 3 clusters. However, when we look at the second plot, the highest silhouette is when I divide the data into 4 clusters. In the second plot, we see that although the best silhouette is to divide the data into 4 clusters, the silhouette of 3 clusters is not much lower than the 4. Moreover, the silhouette score is low in those both option, so I will choose to divide the data into 3 clusters.

Below is two functions:

- **create_kmeans_classifier** - creating KMeans algorithm.
- **clusters_information** - display each cluster and his statistic details.

In [69]:
```
def create_kmeans_calssifier(k):
    try:
        return KMeans(n_clusters=k, init='k-means++')
    except:
        print("Something got wrong - create_kmeans_calssifier")
```

In [70]:
```
def clusters_information(data_frame):
    try:
        clusters = data_frame.groupby("label")
        for name, group in clusters:
            print(name)
            print(group)
            print(group.describe())
    except:
        print("Something got wrong - clusters_information")
```

Activating the functions:

In [71]:
```
kmeans = create_kmeans_calssifier(3)
kmeans.fit(hotel_unsupervised)
silhouette_score(hotel_unsupervised, kmeans.labels_)
```

Out[71]:
```
0.26888207260811636
```

In [72]:
```
hotel_unsupervised["label"] = pd.Series(kmeans.labels_)
clusters_information(hotel_unsupervised)
```

```
0
       hotel  is_canceled  lead_time  arrival_date_year  arrival_date_month  \
0        1.0          0.0   0.464043                0.0            0.545455
1        1.0          0.0   1.000000                0.0            0.545455
2        1.0          0.0   0.009498                0.0            0.545455
3        1.0          0.0   0.017639                0.0            0.545455
4        1.0          0.0   0.018996                0.0            0.545455
...      ...          ...        ...                ...                 ...
40055    1.0          0.0   0.287653                1.0            0.636364
40056    1.0          0.0   0.229308                1.0            0.636364
40057    1.0          0.0   0.276798                1.0            0.636364
40058    1.0          0.0   0.286296                1.0            0.636364
40059    1.0          0.0   0.218453                1.0            0.636364

       arrival_date_day_of_month  stays_in_weekend_nights  \
0                       0.000000                 0.000000
1                       0.000000                 0.000000
2                       0.000000                 0.000000
3                       0.000000                 0.000000
4                       0.000000                 0.000000
...                          ...                      ...
40055                   1.000000                 0.105263
40056                   0.966667                 0.105263
40057                   0.933333                 0.210526
40058                   1.000000                 0.210526
40059                   1.000000                 0.210526

       stays_in_week_nights    adults  children  ...  \
0                      0.00  0.036364       0.0  ...
1                      0.00  0.036364       0.0  ...
2                      0.02  0.018182       0.0  ...
3                      0.02  0.018182       0.0  ...
4                      0.04  0.036364       0.0  ...
...                     ...       ...       ...  ...
40055                  0.16  0.036364       0.1  ...
40056                  0.18  0.036364       0.0  ...
40057                  0.20  0.036364       0.0  ...
40058                  0.20  0.036364       0.0  ...
40059                  0.20  0.036364       0.0  ...

       previous_bookings_not_canceled  booking_changes  deposit_type  \
0                                 0.0         0.142857           0.0
1                                 0.0         0.190476           0.0
2                                 0.0         0.000000           0.0
3                                 0.0         0.000000           0.0
4                                 0.0         0.000000           0.0
...                               ...              ...           ...
40055                             0.0         0.047619           0.0
40056                             0.0         0.000000           0.0
40057                             0.0         0.000000           0.0
40058                             0.0         0.000000           0.0
40059                             0.0         0.000000           0.0

       days_in_waiting_list  customer_type       adr  \
0                       0.0       0.666667  0.001180
1                       0.0       0.666667  0.001180
2                       0.0       0.666667  0.015053
3                       0.0       0.666667  0.015053
4                       0.0       0.666667  0.019307
...                     ...            ...       ...
40055                   0.0       0.666667  0.017781
40056                   0.0       1.000000  0.038593
40057                   0.0       0.666667  0.029585
40058                   0.0       0.000000  0.022044
40059                   0.0       0.666667  0.019503

       required_car_parking_spaces  total_of_special_requests  \
0                              0.0                        0.0
1                              0.0                        0.0
2                              0.0                        0.0
3                              0.0                        0.0
4                              0.0                        0.2
...                            ...                        ...
40055                          0.0                        0.0
40056                          0.0                        0.2
40057                          0.0                        0.6
40058                          0.0                        0.2
40059                          0.0                        0.0

       reservation_status  label
0                     0.5      0
1                     0.5      0
2                     0.5      0
3                     0.5      0
4                     0.5      0
...                   ...    ...
40055                 0.5      0
40056                 0.5      0
40057                 0.5      0
40058                 0.5      0
40059                 0.5      0

[29228 rows x 26 columns]
         hotel   is_canceled     lead_time  arrival_date_year  \
count  29228.0  29228.000000  29228.000000       29228.000000
mean       1.0      0.009922      0.106652           0.551132
std        0.0      0.099116      0.126125           0.360262
min        1.0      0.000000      0.000000           0.000000
25%        1.0      0.000000      0.006784           0.500000
50%        1.0      0.000000      0.050204           0.500000
75%        1.0      0.000000      0.176391           1.000000
max        1.0      1.000000      1.000000           1.000000

       arrival_date_month  arrival_date_day_of_month  stays_in_weekend_nights  \
count        29228.000000               29228.000000             29228.000000
mean             0.499381                   0.495889                 0.059584
std              0.298264                   0.295200                 0.060173
min              0.000000                   0.000000                 0.000000
25%              0.272727                   0.233333                 0.000000
50%              0.545455                   0.500000                 0.052632
75%              0.727273                   0.766667                 0.105263
max              1.000000                   1.000000                 1.000000

       stays_in_week_nights        adults      children  ...  \
count          29228.000000  29228.000000  29228.000000  ...
mean               0.060060      0.033304      0.010682  ...
std                0.048973      0.008485      0.040265  ...
min                0.000000      0.000000      0.000000  ...
25%                0.020000      0.036364      0.000000  ...
50%                0.040000      0.036364      0.000000  ...
75%                0.100000      0.036364      0.000000  ...
max                1.000000      0.072727      1.000000  ...

       previous_bookings_not_canceled  booking_changes  deposit_type  \
count                    29228.000000     29228.000000  29228.000000
mean                         0.002684         0.016164      0.005594
std                          0.015848         0.036912      0.069417
min                          0.000000         0.000000      0.000000
25%                          0.000000         0.000000      0.000000
50%                          0.000000         0.000000      0.000000
75%                          0.000000         0.000000      0.000000
max                          0.416667         0.809524      1.000000

       days_in_waiting_list  customer_type           adr  \
count          29228.000000   29228.000000  29228.000000
mean               0.001760       0.698508      0.017954
std                0.021741       0.220483      0.010966
min                0.000000       0.000000      0.000000
25%                0.000000       0.666667      0.010058
50%                0.000000       0.666667      0.014498
75%                0.000000       0.666667      0.023006
max                0.473146       1.000000      0.095143

       required_car_parking_spaces  total_of_special_requests  \
count                 29228.000000               29228.000000
mean                      0.023655                   0.134015
std                       0.049872                   0.166359
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.000000
75%                       0.000000                   0.200000
max                       1.000000                   1.000000

       reservation_status    label
count        29228.000000  29228.0
mean             0.504961      0.0
std              0.049558      0.0
min              0.500000      0.0
25%              0.500000      0.0
50%              0.500000      0.0
75%              0.500000      0.0
max              1.000000      0.0

[8 rows x 26 columns]
1
        hotel  is_canceled  lead_time  arrival_date_year  arrival_date_month  \
8         1.0          1.0   0.115332                0.0            0.545455
9         1.0          1.0   0.101764                0.0            0.545455
10        1.0          1.0   0.031208                0.0            0.545455
27        1.0          1.0   0.081411                0.0            0.545455
32        1.0          1.0   0.130258                0.0            0.545455
...       ...          ...        ...                ...                 ...
110276    0.0          1.0   0.179104                1.0            0.272727
111351    0.0          1.0   0.005427                1.0            0.454545
111920    0.0          1.0   0.009498                1.0            0.363636
111921    0.0          1.0   0.008141                1.0            0.545455
117291    0.0          1.0   0.000000                1.0            0.636364

        arrival_date_day_of_month  stays_in_weekend_nights  \
8                        0.000000                 0.000000
9                        0.000000                 0.000000
10                       0.000000                 0.000000
27                       0.000000                 0.105263
32                       0.000000                 0.105263
...                           ...                      ...
110276                   0.800000                 0.000000
111351                   0.133333                 0.052632
111920                   1.000000                 0.000000
111921                   0.533333                 0.052632
117291                   0.033333                 0.000000

        stays_in_week_nights    adults  children  ...  \
8                       0.06  0.036364       0.0  ...
9                       0.06  0.036364       0.0  ...
10                      0.08  0.036364       0.0  ...
27                      0.10  0.036364       0.0  ...
32                      0.16  0.036364       0.0  ...
...                      ...       ...       ...  ...
110276                  0.00  0.000000       0.0  ...
111351                  0.00  0.018182       0.0  ...
111920                  0.02  0.018182       0.0  ...
111921                  0.00  0.018182       0.0  ...
117291                  0.04  0.018182       0.0  ...

        previous_bookings_not_canceled  booking_changes  deposit_type  \
8                             0.000000         0.000000           0.0
9                             0.000000         0.000000           0.0
10                            0.000000         0.000000           0.0
27                            0.000000         0.000000           0.0
32                            0.000000         0.000000           0.0
...                                ...              ...           ...
110276                        0.000000         0.047619           0.0
111351                        0.055556         0.000000           0.0
111920                        0.013889         0.000000           0.0
111921                        0.013889         0.000000           0.0
117291                        0.000000         0.000000           0.0

        days_in_waiting_list  customer_type       adr  \
8                        0.0       0.666667  0.016347
9                        0.0       0.666667  0.020694
10                       0.0       0.666667  0.023931
27                       0.0       0.666667  0.020972
32                       0.0       0.666667  0.021212
...                      ...            ...       ...
110276                   0.0       0.666667  0.001180
111351                   0.0       0.666667  0.013203
111920                   0.0       0.666667  0.013203
111921                   0.0       0.666667  0.013203
117291                   0.0       0.666667  0.013203

        required_car_parking_spaces  total_of_special_requests  \
8                               0.0                        0.2
9                               0.0                        0.0
10                              0.0                        0.0
27                              0.0                        0.4
32                              0.0                        0.4
...                             ...                        ...
110276                          0.0                        0.6
111351                          0.0                        0.0
111920                          0.0                        0.0
111921                          0.0                        0.0
117291                          0.0                        0.2

        reservation_status  label
8                      0.0      1
9                      0.0      1
10                     0.0      1
27                     0.0      1
32                     0.0      1
...                    ...    ...
110276                 0.0      1
111351                 0.0      1
111920                 0.0      1
111921                 1.0      1
117291                 1.0      1

[43930 rows x 26 columns]
              hotel  is_canceled     lead_time  arrival_date_year  \
count  43930.000000      43930.0  43930.000000       43930.000000
mean       0.246574          1.0      0.197359           0.586126
std        0.431022          0.0      0.160946           0.357054
min        0.000000          1.0      0.000000           0.000000
25%        0.000000          1.0      0.066486           0.500000
50%        0.000000          1.0      0.154681           0.500000
75%        0.000000          1.0      0.291723           1.000000
max        1.000000          1.0      0.853460           1.000000

       arrival_date_month  arrival_date_day_of_month  stays_in_weekend_nights  \
count        43930.000000               43930.000000             43930.000000
mean             0.508961                   0.490849                 0.048690
std              0.270483                   0.292786                 0.053018
min              0.000000                   0.000000                 0.000000
25%              0.272727                   0.233333                 0.000000
50%              0.545455                   0.500000                 0.052632
75%              0.727273                   0.733333                 0.105263
max              1.000000                   1.000000                 0.842105

       stays_in_week_nights        adults      children  ...  \
count          43930.000000  43930.000000  43930.000000  ...
mean               0.051260      0.034592      0.010615  ...
std                0.037526      0.012348      0.040854  ...
min                0.000000      0.000000      0.000000  ...
25%                0.020000      0.036364      0.000000  ...
50%                0.040000      0.036364      0.000000  ...
75%                0.060000      0.036364      0.000000  ...
max                0.800000      1.000000      0.300000  ...

       previous_bookings_not_canceled  booking_changes  deposit_type  \
count                    43930.000000     43930.000000  43930.000000
mean                         0.000342         0.004616      0.165582
std                          0.009427         0.021290      0.236188
min                          0.000000         0.000000      0.000000
25%                          0.000000         0.000000      0.000000
50%                          0.000000         0.000000      0.000000
75%                          0.000000         0.000000      0.500000
max                          0.805556         0.761905      1.000000

       days_in_waiting_list  customer_type           adr  \
count          43930.000000   43930.000000  43930.000000
mean               0.009176       0.695425      0.020626
std                0.055137       0.167656      0.009713
min                0.000000       0.000000      0.001180
25%                0.000000       0.666667      0.014683
50%                0.000000       0.666667      0.018992
75%                0.000000       0.666667      0.024819
max                1.000000       1.000000      1.000000

       required_car_parking_spaces  total_of_special_requests  \
count                      43930.0               43930.000000
mean                           0.0                   0.065404
std                            0.0                   0.129558
min                            0.0                   0.000000
25%                            0.0                   0.000000
50%                            0.0                   0.000000
75%                            0.0                   0.000000
max                            0.0                   1.000000

       reservation_status    label
count        43930.000000  43930.0
mean             0.020874      1.0
std              0.142965      0.0
min              0.000000      1.0
25%              0.000000      1.0
50%              0.000000      1.0
75%              0.000000      1.0
max              1.000000      1.0

[8 rows x 26 columns]
2
        hotel  is_canceled  lead_time  arrival_date_year  arrival_date_month  \
40060     0.0          0.0   0.008141                0.0            0.545455
40066     0.0          0.0   0.004071                0.0            0.545455
40070     0.0          0.0   0.058345                0.0            0.545455
40071     0.0          0.0   0.058345                0.0            0.545455
40072     0.0          0.0   0.058345                0.0            0.545455
...       ...          ...        ...                ...                 ...
119381    0.0          0.0   0.031208                1.0            0.636364
119382    0.0          0.0   0.138399                1.0            0.636364
119383    0.0          0.0   0.046133                1.0            0.636364
119384    0.0          0.0   0.147897                1.0            0.636364
119385    0.0          0.0   0.278155                1.0            0.636364

        arrival_date_day_of_month  stays_in_weekend_nights  \
40060                    0.000000                 0.000000
40066                    0.033333                 0.000000
40070                    0.066667                 0.000000
40071                    0.066667                 0.000000
40072                    0.066667                 0.000000
...                           ...                      ...
119381                   0.966667                 0.105263
119382                   1.000000                 0.105263
119383                   1.000000                 0.105263
119384                   1.000000                 0.105263
119385                   0.933333                 0.105263

        stays_in_week_nights    adults  children  ...  \
40060                   0.04  0.018182       0.0  ...
40066                   0.06  0.018182       0.0  ...
40070                   0.04  0.036364       0.0  ...
40071                   0.04  0.036364       0.0  ...
40072                   0.04  0.036364       0.0  ...
...                      ...       ...       ...  ...
119381                  0.10  0.036364       0.0  ...
119382                  0.10  0.054545       0.0  ...
119383                  0.10  0.036364       0.0  ...
119384                  0.10  0.036364       0.0  ...
119385                  0.14  0.036364       0.0  ...

        previous_bookings_not_canceled  booking_changes  deposit_type  \
40060                              0.0         0.000000           0.0
40066                              0.0         0.047619           0.0
40070                              0.0         0.000000           0.0
40071                              0.0         0.047619           0.0
40072                              0.0         0.000000           0.0
...                                ...              ...           ...
119381                             0.0         0.000000           0.0
119382                             0.0         0.000000           0.0
119383                             0.0         0.000000           0.0
119384                             0.0         0.000000           0.0
119385                             0.0         0.000000           0.0

        days_in_waiting_list  customer_type       adr  \
40060                    0.0       0.666667  0.001180
40066                    0.0       1.000000  0.012032
40070                    0.0       1.000000  0.017087
40071                    0.0       1.000000  0.009134
40072                    0.0       1.000000  0.017087
...                      ...            ...       ...
119381                   0.0       0.666667  0.018963
119382                   0.0       0.666667  0.042877
119383                   0.0       0.666667  0.030351
119384                   0.0       0.666667  0.020491
119385                   0.0       0.666667  0.029147

        required_car_parking_spaces  total_of_special_requests  \
40060                           0.0                        0.0
40066                           0.0                        0.0
40070                           0.0                        0.0
40071                           0.0                        0.0
40072                           0.0                        0.0
...                             ...                        ...
119381                          0.0                        0.0
119382                          0.0                        0.4
119383                          0.0                        0.8
119384                          0.0                        0.0
119385                          0.0                        0.4

        reservation_status  label
40060                  0.5      2
40066                  0.5      2
40070                  0.5      2
40071                  0.5      2
40072                  0.5      2
...                    ...    ...
119381                 0.5      2
119382                 0.5      2
119383                 0.5      2
119384                 0.5      2
119385                 0.5      2

[46228 rows x 26 columns]
         hotel  is_canceled     lead_time  arrival_date_year  \
count  46228.0      46228.0  46228.000000       46228.000000
mean       0.0          0.0      0.109502           0.588031
std        0.0          0.0      0.121931           0.345418
min        0.0          0.0      0.000000           0.000000
25%        0.0          0.0      0.016282           0.500000
50%        0.0          0.0      0.067843           0.500000
75%        0.0          0.0      0.164179           1.000000
max        0.0          0.0      0.702849           1.000000

       arrival_date_month  arrival_date_day_of_month  stays_in_weekend_nights  \
count        46228.000000               46228.000000             46228.000000
mean             0.504187                   0.493954                 0.042141
std              0.279415                   0.290996                 0.045343
min              0.000000                   0.000000                 0.000000
25%              0.272727                   0.233333                 0.000000
50%              0.545455                   0.500000                 0.052632
75%              0.727273                   0.733333                 0.105263
max              1.000000                   1.000000                 0.842105

       stays_in_week_nights        adults      children  ...  \
count          46228.000000  46228.000000  46228.000000  ...
mean               0.042459      0.033238      0.009990  ...
std                0.028016      0.009747      0.038614  ...
min                0.000000      0.000000      0.000000  ...
25%                0.020000      0.036364      0.000000  ...
50%                0.040000      0.036364      0.000000  ...
75%                0.060000      0.036364      0.000000  ...
max                0.820000      0.072727      0.300000  ...

       previous_bookings_not_canceled  booking_changes  deposit_type  \
count                    46228.000000     46228.000000  46228.000000
mean                         0.002896         0.012589      0.000389
std                          0.029499         0.033796      0.016107
min                          0.000000         0.000000      0.000000
25%                          0.000000         0.000000      0.000000
50%                          0.000000         0.000000      0.000000
75%                          0.000000         0.000000      0.000000
max                          1.000000         1.000000      1.000000

       days_in_waiting_list  customer_type           adr  \
count          46228.000000   46228.000000  46228.000000
mean               0.005499       0.737396      0.020740
std                0.044953       0.192554      0.007509
min                0.000000       0.000000      0.001180
25%                0.000000       0.666667      0.015977
50%                0.000000       0.666667      0.019658
75%                0.000000       1.000000      0.024486
max                0.969309       1.000000      0.095513

       required_car_parking_spaces  total_of_special_requests  \
count                 46228.000000               46228.000000
mean                      0.005227                   0.148218
std                       0.025142                   0.166770
min                       0.000000                   0.000000
25%                       0.000000                   0.000000
50%                       0.000000                   0.200000
75%                       0.000000                   0.200000
max                       0.375000                   1.000000

       reservation_status    label
count             46228.0  46228.0
mean                  0.5      2.0
std                   0.0      0.0
min                   0.5      2.0
25%                   0.5      2.0
50%                   0.5      2.0
75%                   0.5      2.0
max                   0.5      2.0

[8 rows x 26 columns]
```

As we can see here, the silhouette score that we got is very low. An explanation for this is because the values of the features are very similar that the algorithm had difficulties to handle those samples and to separate them into groups that have the same behavior.