In [1]:
```
# importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import warnings
warnings.filterwarnings('ignore')
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.ensemble import VotingClassifier
import folium
from folium.plugins import HeatMap
import plotly.express as px
plt.style.use('fivethirtyeight')
%matplotlib inline
pd.set_option('display.max_columns', 32)
```

In [2]:
```
# reading data
df = pd.read_csv('hotel_bookings.csv')
df.head()
```

Out[2]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
2	Resort Hotel	0	7	2015	July	27	1	0	

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

EDA

In [3]:
```
df.describe()
```

Out[3]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
count	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000	119390.000000
mean	0.370416	104.011416	2016.156554	27.165173	15.798241	0.927599	2.500302
std	0.482918	106.863097	0.707476	13.605138	8.780829	0.998613	1.908286
min	0.000000	0.000000	2015.000000	1.000000	1.000000	0.000000	0.000000
25%	0.000000	18.000000	2016.000000	16.000000	8.000000	0.000000	1.000000
50%	0.000000	69.000000	2016.000000	28.000000	16.000000	1.000000	2.000000
75%	1.000000	160.000000	2017.000000	38.000000	23.000000	2.000000	3.000000
max	1.000000	737.000000	2017.000000	53.000000	31.000000	19.000000	50.000000

In [4]:
```
df.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
```

In [5]:
```
# checking for null values

null = pd.DataFrame({'Null Values' : df.isna().sum(), 'Percentage Null Values' : (df.isna().sum()) / (df.shape[0]) * (100)})
null
```

Out[5]:

	Null Values	Percentage Null Values
hotel	0	0.000000
is_canceled	0	0.000000
lead_time	0	0.000000
arrival_date_year	0	0.000000

	Null Values	Percentage Null Values
arrival_date_month	0	0.000000
arrival_date_week_number	0	0.000000
arrival_date_day_of_month	0	0.000000
stays_in_weekend_nights	0	0.000000
stays_in_week_nights	0	0.000000
adults	0	0.000000
children	4	0.003350
babies	0	0.000000
meal	0	0.000000
country	488	0.408744
market_segment	0	0.000000
distribution_channel	0	0.000000
is_repeated_guest	0	0.000000
previous_cancellations	0	0.000000
previous_bookings_not_canceled	0	0.000000
reserved_room_type	0	0.000000
assigned_room_type	0	0.000000
booking_changes	0	0.000000
deposit_type	0	0.000000
agent	16340	13.686238
company	112593	94.306893
days_in_waiting_list	0	0.000000
customer_type	0	0.000000
adr	0	0.000000

	Null Values	Percentage Null Values
required_car_parking_spaces	0	0.000000
total_of_special_requests	0	0.000000
reservation_status	0	0.000000
reservation_status_date	0	0.000000

In [6]:
```
# filling null values with zero

df.fillna(0, inplace = True)
```

In [7]:
```
# adults, babies and children cant be zero at same time, so dropping the rows having all these zero at same time

filter = (df.children == 0) & (df.adults == 0) & (df.babies == 0)
df[filter]
```

Out[7]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
2224	Resort Hotel	0	1	2015	October	41	6	0
2409	Resort Hotel	0	0	2015	October	42	12	0
3181	Resort Hotel	0	36	2015	November	47	20	1
3684	Resort Hotel	0	165	2015	December	53	30	1
3708	Resort Hotel	0	165	2015	December	53	30	2
...	...	...	...	...	...	...	...	...
115029	City Hotel	0	107	2017	June	26	27	0
115091	City Hotel	0	1	2017	June	26	30	0

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
116251	City Hotel	0	44	2017	July	28	15	1
116534	City Hotel	0	2	2017	July	28	15	2
117087	City Hotel	0	170	2017	July	30	27	0

180 rows × 32 columns

In [8]:
```
df = df[~filter]
df
```

Out[8]:

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

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights
119388	City Hotel	0	109	2017	August	35	31	2
119389	City Hotel	0	205	2017	August	35	29	2

119210 rows × 32 columns

In [9]:
```
country_wise_guests = df[df['is_canceled'] == 0]['country'].value_counts().reset_index()
country_wise_guests.columns = ['country', 'No of guests']
country_wise_guests
```

Out[9]:

	country	No of guests
0	PRT	20977
1	GBR	9668
2	FRA	8468
3	ESP	6383
4	DEU	6067
...	...	...
161	BFA	1
162	MRT	1
163	KIR	1
164	ZMB	1
165	DMA	1

166 rows × 2 columns

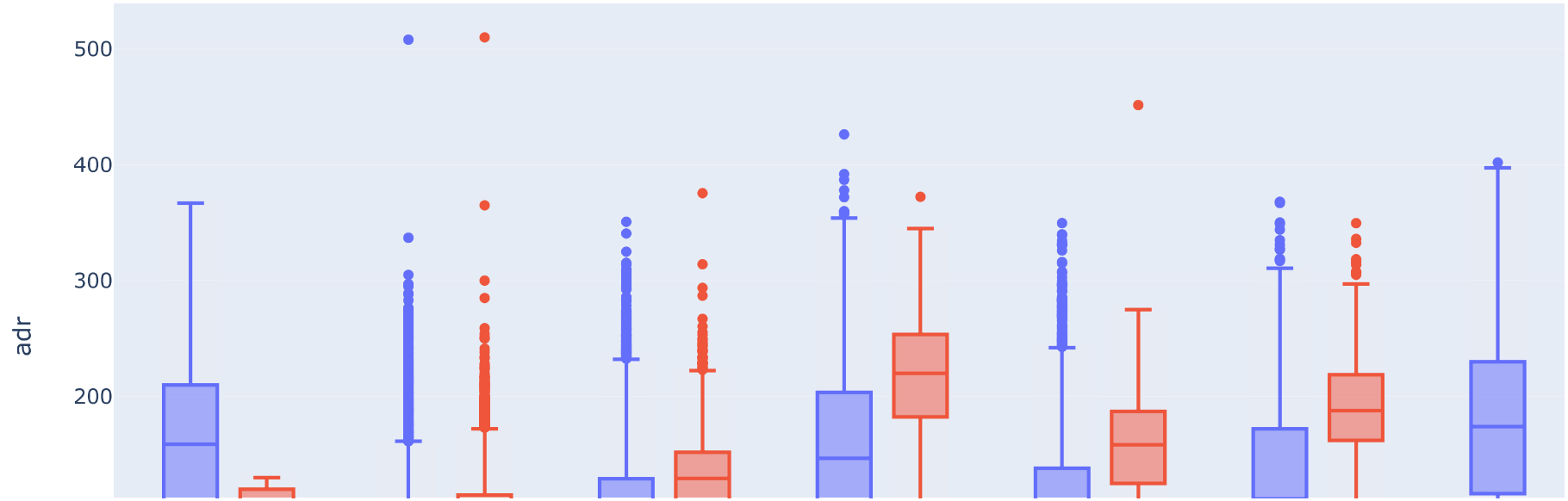
In [10]:
```
df.head()
```

Out[10]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
2	Resort Hotel	0	7	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

In [11]:
```
data = df[df['is_canceled'] == 0]

px.box(data_frame = data, x = 'reserved_room_type', y = 'adr', color = 'hotel')
```



In [12]:
```
#Variation of per nignt price over the year
data_resort = df[(df['hotel'] == 'Resort Hotel') & (df['is_canceled'] == 0)]
data_city = df[(df['hotel'] == 'City Hotel') & (df['is_canceled'] == 0)]
resort_hotel = data_resort.groupby(['arrival_date_month'])['adr'].mean().reset_index()
resort_hotel
```

Out[12]:

	arrival_date_month	adr
0	April	75.867816
1	August	181.205892
2	December	68.410104
3	February	54.147478
4	January	48.761125
5	July	150.122528
6	June	107.974850
7	March	57.056838
8	May	76.657558
9	November	48.706289
10	October	61.775449
11	September	96.416860

In [13]:
```
city_hotel=data_city.groupby(['arrival_date_month'])['adr'].mean().reset_index()
```

```
city_hotel
```

Out[13]:

	arrival_date_month	adr
0	April	111.962267
1	August	118.674598
2	December	88.401855
3	February	86.520062
4	January	82.330983
5	July	115.818019
6	June	117.874360
7	March	90.658533
8	May	120.669827
9	November	86.946592
10	October	102.004672
11	September	112.776582

In [14]:

```
final_hotel = resort_hotel.merge(city_hotel, on = 'arrival_date_month')
final_hotel.columns = ['month', 'price_for_resort', 'price_for_city_hotel']
final_hotel
```

Out[14]:

	month	price_for_resort	price_for_city_hotel
0	April	75.867816	111.962267
1	August	181.205892	118.674598
2	December	68.410104	88.401855
3	February	54.147478	86.520062
4	January	48.761125	82.330983
5	July	150.122528	115.818019
6	June	107.974850	117.874360

	month	price_for_resort	price_for_city_hotel
7	March	57.056838	90.658533
8	May	76.657558	120.669827
9	November	48.706289	86.946592
10	October	61.775449	102.004672
11	September	96.416860	112.776582

In [15]:
```
# most busy months
resort_guests = data_resort['arrival_date_month'].value_counts().reset_index()
resort_guests.columns=['month','no of guests']
resort_guests
```

Out[15]:

	month	no of guests
0	August	3257
1	July	3137
2	October	2575
3	March	2571
4	April	2550
5	May	2535
6	February	2308
7	September	2102
8	June	2037
9	December	2014
10	November	1975
11	January	1866

In [16]:
```
city_guests = data_city['arrival_date_month'].value_counts().reset_index()
city_guests.columns=['month','no of guests']
city_guests
```

Out[16]:

	month	no of guests
0	August	5367
1	July	4770
2	May	4568
3	June	4358
4	October	4326
5	September	4283
6	March	4049
7	April	4010
8	February	3051
9	November	2676
10	December	2377
11	January	2249

In [17]:
```
final_guests = resort_guests.merge(city_guests,on='month')
final_guests.columns=['month','no of guests in resort','no of guest in city hotel']
final_guests
```

Out[17]:

	month	no of guests in resort	no of guest in city hotel
0	August	3257	5367
1	July	3137	4770
2	October	2575	4326
3	March	2571	4049
4	April	2550	4010
5	May	2535	4568
6	February	2308	3051
7	September	2102	4283

	month	no of guests in resort	no of guest in city hotel
8	June	2037	4358
9	December	2014	2377
10	November	1975	2676
11	January	1866	2249

In [18]:
```
filter = df['is_canceled'] == 0
data = df[filter]
data.head()
```

Out[18]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	
2	Resort Hotel	0	7	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

In [19]:
```
data['total_nights'] = data['stays_in_weekend_nights'] + data['stays_in_week_nights']
data.head()
```

Out[19]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
0	Resort Hotel	0	342	2015	July	27	1	0	
1	Resort Hotel	0	737	2015	July	27	1	0	

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stay
2	Resort Hotel	0	7	2015	July	27	1	0	
3	Resort Hotel	0	13	2015	July	27	1	0	
4	Resort Hotel	0	14	2015	July	27	1	0	

5 rows × 33 columns

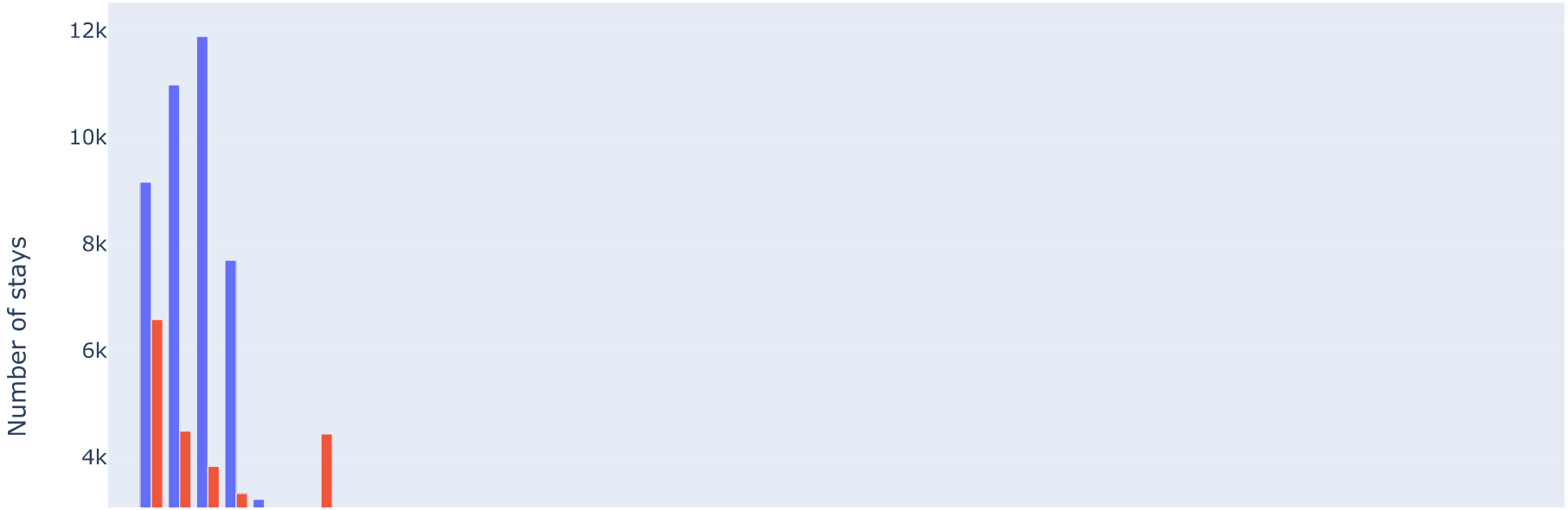
In [20]:
```
stay = data.groupby(['total_nights', 'hotel']).agg('count').reset_index()
stay = stay.iloc[:, :3]
stay = stay.rename(columns={'is_canceled':'Number of stays'})
stay
```

Out[20]:

	total_nights	hotel	Number of stays
0	0	City Hotel	251
1	0	Resort Hotel	371
2	1	City Hotel	9155
3	1	Resort Hotel	6579
4	2	City Hotel	10983
...	...	...	...
57	46	Resort Hotel	1
58	48	City Hotel	1
59	56	Resort Hotel	1
60	60	Resort Hotel	1
61	69	Resort Hotel	1

62 rows × 3 columns

In [21]:
```
px.bar(data_frame = stay, x = 'total_nights', y = 'Number of stays', color = 'hotel', barmode = 'group')
```

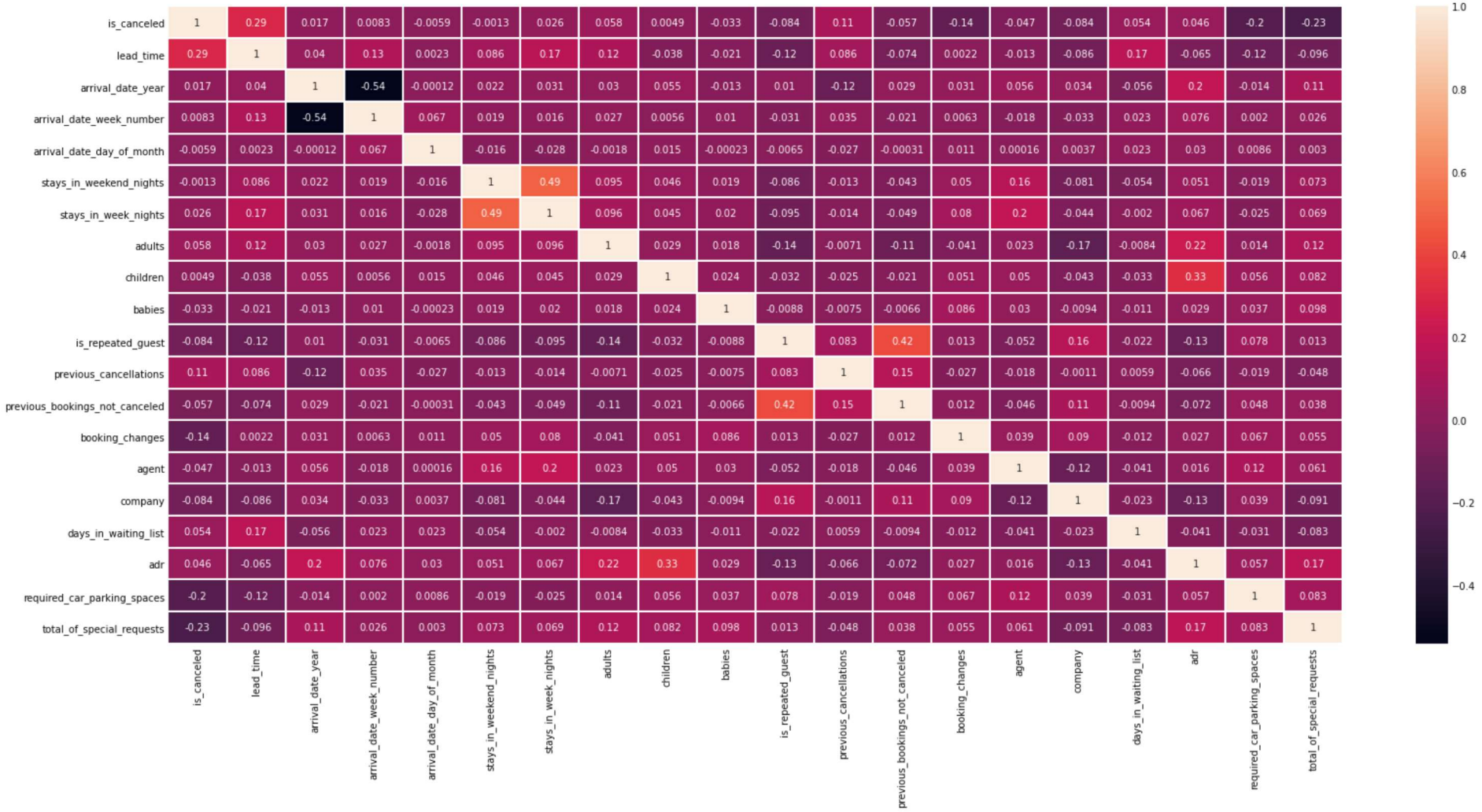


Data Pre Processing

In [22]:
```
plt.figure(figsize = (24, 12))

corr = df.corr()
```

```
sns.heatmap(corr, annot = True, linewidths = 1)
plt.show()
```



In [23]:
```
correlation = df.corr()['is_canceled'].abs().sort_values(ascending = False)
correlation
```

Out[23]:
```
is_canceled                     1.000000
lead_time                       0.292876
total_of_special_requests       0.234877
required_car_parking_spaces     0.195701
booking_changes                 0.144832
```

```
previous_cancellations            0.110139
is_repeated_guest                 0.083745
company                           0.083594
adults                            0.058182
previous_bookings_not_canceled    0.057365
days_in_waiting_list              0.054301
agent                             0.046770
adr                               0.046492
babies                            0.032569
stays_in_week_nights              0.025542
arrival_date_year                 0.016622
arrival_date_week_number          0.008315
arrival_date_day_of_month         0.005948
children                          0.004851
stays_in_weekend_nights           0.001323
Name: is_canceled, dtype: float64
```

In [24]:
```
# dropping columns that are not useful

useless_col = ['days_in_waiting_list', 'arrival_date_year', 'arrival_date_year', 'assigned_room_type', 'booking_changes',
               'reservation_status', 'country', 'days_in_waiting_list']

df.drop(useless_col, axis = 1, inplace = True)
```

In [25]:
```
df.head()
```

Out[25]:

	hotel	is_canceled	lead_time	arrival_date_month	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights
0	Resort Hotel	0	342	July	27	1	0	0
1	Resort Hotel	0	737	July	27	1	0	0
2	Resort Hotel	0	7	July	27	1	0	1
3	Resort Hotel	0	13	July	27	1	0	1
4	Resort Hotel	0	14	July	27	1	0	2

In [26]:
```
# creating numerical and categorical dataframes

cat_cols = [col for col in df.columns if df[col].dtype == 'O']
cat_cols
```

Out[26]:
```
['hotel',
 'arrival_date_month',
 'meal',
 'market_segment',
 'distribution_channel',
 'reserved_room_type',
 'deposit_type',
 'customer_type',
 'reservation_status_date']
```

In [27]:
```
cat_df = df[cat_cols]
cat_df.head()
```

Out[27]:

	hotel	arrival_date_month	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	reservation_status_date
0	Resort Hotel	July	BB	Direct	Direct	C	No Deposit	Transient	2015-07-01
1	Resort Hotel	July	BB	Direct	Direct	C	No Deposit	Transient	2015-07-01
2	Resort Hotel	July	BB	Direct	Direct	A	No Deposit	Transient	2015-07-02
3	Resort Hotel	July	BB	Corporate	Corporate	A	No Deposit	Transient	2015-07-02
4	Resort Hotel	July	BB	Online TA	TA/TO	A	No Deposit	Transient	2015-07-03

In [28]:
```
cat_df['reservation_status_date'] = pd.to_datetime(cat_df['reservation_status_date'])

cat_df['year'] = cat_df['reservation_status_date'].dt.year
cat_df['month'] = cat_df['reservation_status_date'].dt.month
cat_df['day'] = cat_df['reservation_status_date'].dt.day
```

In [29]:
```
cat_df.drop(['reservation_status_date','arrival_date_month'] , axis = 1, inplace = True)
```

In [30]:
```
cat_df.head()
```

Out[30]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day
0	Resort Hotel	BB	Direct	Direct	C	No Deposit	Transient	2015	7	1
1	Resort Hotel	BB	Direct	Direct	C	No Deposit	Transient	2015	7	1
2	Resort Hotel	BB	Direct	Direct	A	No Deposit	Transient	2015	7	2
3	Resort Hotel	BB	Corporate	Corporate	A	No Deposit	Transient	2015	7	2
4	Resort Hotel	BB	Online TA	TA/TO	A	No Deposit	Transient	2015	7	3

In [31]:
```
# printing unique values of each column
for col in cat_df.columns:
    print(f"{col}: \n{cat_df[col].unique()}\n")
```

```
hotel:
['Resort Hotel' 'City Hotel']

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' 'B']

deposit_type:
['No Deposit' 'Refundable' 'Non Refund']

customer_type:
['Transient' 'Contract' 'Transient-Party' 'Group']

year:
[2015 2014 2016 2017]

month:
[ 7  5  4  6  3  8  9  1 11 10 12  2]

day:
[ 1  2  3  6 22 23  5  7  8 11 15 16 29 19 18  9 13  4 12 26 17 10 20 14
```

```
30 28 25 21 27 24 31]
```

In [32]:
```
# encoding categorical variables

cat_df['hotel'] = cat_df['hotel'].map({'Resort Hotel' : 0, 'City Hotel' : 1})

cat_df['meal'] = cat_df['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC': 3, 'Undefined': 4})

cat_df['market_segment'] = cat_df['market_segment'].map({'Direct': 0, 'Corporate': 1, 'Online TA': 2, 'Offline TA/TO': 3,
                                                         'Complementary': 4, 'Groups': 5, 'Undefined': 6, 'Aviation': 7})

cat_df['distribution_channel'] = cat_df['distribution_channel'].map({'Direct': 0, 'Corporate': 1, 'TA/TO': 2, 'Undefined': 3,
                                                             'GDS': 4})

cat_df['reserved_room_type'] = cat_df['reserved_room_type'].map({'C': 0, 'A': 1, 'D': 2, 'E': 3, 'G': 4, 'F': 5, 'H': 6,
                                                         'L': 7, 'B': 8})

cat_df['deposit_type'] = cat_df['deposit_type'].map({'No Deposit': 0, 'Refundable': 1, 'Non Refund': 3})

cat_df['customer_type'] = cat_df['customer_type'].map({'Transient': 0, 'Contract': 1, 'Transient-Party': 2, 'Group': 3})

cat_df['year'] = cat_df['year'].map({2015: 0, 2014: 1, 2016: 2, 2017: 3})
```

In [33]:
```
cat_df.head()
```

Out[33]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day
0	0	0	0	0	0	0	0	0	7	1
1	0	0	0	0	0	0	0	0	7	1
2	0	0	0	0	1	0	0	0	7	2
3	0	0	1	1	1	0	0	0	7	2
4	0	0	2	2	1	0	0	0	7	3

In [34]:
```
num_df = df.drop(columns = cat_cols, axis = 1)
num_df.drop('is_canceled', axis = 1, inplace = True)
num_df
```

Out[34]:

	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeat

	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeat
0	342	27	1	0	0	2	0.0	0	
1	737	27	1	0	0	2	0.0	0	
2	7	27	1	0	1	1	0.0	0	
3	13	27	1	0	1	1	0.0	0	
4	14	27	1	0	2	2	0.0	0	
...	...	...	...	...	...	...	...	...	
119385	23	35	30	2	5	2	0.0	0	
119386	102	35	31	2	5	3	0.0	0	
119387	34	35	31	2	5	2	0.0	0	
119388	109	35	31	2	5	2	0.0	0	
119389	205	35	29	2	7	2	0.0	0	

119210 rows × 16 columns

In [35]:
```
num_df.var()
```

Out[35]:
```
lead_time                       11422.361808
arrival_date_week_number          184.990111
arrival_date_day_of_month          77.107192
stays_in_weekend_nights             0.990258
stays_in_week_nights                3.599010
adults                              0.330838
children                            0.159070
babies                              0.009508
is_repeated_guest                   0.030507
previous_cancellations              0.713887
previous_bookings_not_canceled      2.244415
agent                           11485.169679
company                          2897.684308
adr                              2543.589039
required_car_parking_spaces         0.060201
total_of_special_requests           0.628652
dtype: float64
```

In [36]:
```
# normalizing numerical variables

num_df['lead_time'] = np.log(num_df['lead_time'] + 1)
num_df['arrival_date_week_number'] = np.log(num_df['arrival_date_week_number'] + 1)
num_df['arrival_date_day_of_month'] = np.log(num_df['arrival_date_day_of_month'] + 1)
num_df['agent'] = np.log(num_df['agent'] + 1)
num_df['company'] = np.log(num_df['company'] + 1)
num_df['adr'] = np.log(num_df['adr'] + 1)
```

In [37]:
```
num_df.var()
```

Out[37]:
```
lead_time                       2.582757
arrival_date_week_number        0.440884
arrival_date_day_of_month       0.506325
stays_in_weekend_nights         0.990258
stays_in_week_nights            3.599010
adults                          0.330838
children                        0.159070
babies                          0.009508
is_repeated_guest               0.030507
previous_cancellations          0.713887
previous_bookings_not_canceled  2.244415
agent                           3.535793
company                         1.346883
adr                             0.515480
required_car_parking_spaces     0.060201
total_of_special_requests       0.628652
dtype: float64
```

In [38]:
```
num_df['adr'] = num_df['adr'].fillna(value = num_df['adr'].mean())
```

In [39]:
```
num_df.head()
```

Out[39]:

	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	is_repeated_gue
0	5.837730	3.332205	0.693147	0	0	2	0.0	0	
1	6.603944	3.332205	0.693147	0	0	2	0.0	0	
2	2.079442	3.332205	0.693147	0	1	1	0.0	0	
3	2.639057	3.332205	0.693147	0	1	1	0.0	0	
4	2.708050	3.332205	0.693147	0	2	2	0.0	0	

In [40]:
```
X = pd.concat([cat_df, num_df], axis = 1)
y = df['is_canceled']
```

In [41]:
```
X.shape, y.shape
```

Out[41]:
```
((119210, 26), (119210,))
```

In [42]:
```
# splitting data into training set and test set

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.30)
```

In [43]:
```
X_train.head()
```

Out[43]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day	lead_time	arrival_date_w
41492	1	0	5	2	1	0	2	0	8	19	3.850148	
71269	1	0	2	2	1	0	0	3	4	15	5.529429	
49637	1	3	2	2	1	0	0	2	3	20	4.553877	
48404	1	0	5	2	1	3	0	0	10	16	5.529429	
35855	0	0	2	2	1	0	0	3	5	3	1.098612	

In [44]:
```
X_test.head()
```

Out[44]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	customer_type	year	month	day	lead_time	arrival_date_
110761	1	0	2	2	2	0	0	3	5	1	2.890372	
102401	1	0	0	0	1	0	0	2	11	27	4.882802	
33189	0	2	5	2	1	0	2	3	2	17	5.337538	
112501	1	0	2	2	1	0	2	3	5	25	2.639057	
61031	1	0	2	2	1	0	0	2	10	25	4.465908	

In [45]:
```
y_train.head(), y_test.head()
```

Out[45]:
```
(41492    0
 71269    1
 49637    1
 48404    1
 35855    0
 Name: is_canceled, dtype: int64,
 110761    0
 102401    0
 33189     0
 112501    0
 61031     1
 Name: is_canceled, dtype: int64)
```

Decision Tree Classifier

In [46]:
```
dtc = DecisionTreeClassifier()
dtc.fit(X_train, y_train)

y_pred_dtc = dtc.predict(X_test)

acc_dtc = accuracy_score(y_test, y_pred_dtc)
conf = confusion_matrix(y_test, y_pred_dtc)
clf_report = classification_report(y_test, y_pred_dtc)

print(f"Accuracy Score of Decision Tree is : {acc_dtc}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
Accuracy Score of Decision Tree is : 0.9455023348153119
Confusion Matrix :
[[21598   943]
 [ 1006 12216]]
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     22541
           1       0.93      0.92      0.93     13222

    accuracy                           0.95     35763
   macro avg       0.94      0.94      0.94     35763
weighted avg       0.95      0.95      0.95     35763
```

Random Forest Classifier

In [47]:
```
rd_clf = RandomForestClassifier()
rd_clf.fit(X_train, y_train)

y_pred_rd_clf = rd_clf.predict(X_test)

acc_rd_clf = accuracy_score(y_test, y_pred_rd_clf)
conf = confusion_matrix(y_test, y_pred_rd_clf)
clf_report = classification_report(y_test, y_pred_rd_clf)

print(f"Accuracy Score of Random Forest is : {acc_rd_clf}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
Accuracy Score of Random Forest is : 0.954869557922993
Confusion Matrix :
[[22355   186]
 [ 1428 11794]]
Classification Report :
              precision    recall  f1-score   support

           0       0.94      0.99      0.97     22541
           1       0.98      0.89      0.94     13222

    accuracy                           0.95     35763
   macro avg       0.96      0.94      0.95     35763
weighted avg       0.96      0.95      0.95     35763
```

Models Comparison

In [48]:
```
models = pd.DataFrame({
    'Model' : ['Decision Tree Classifier', 'Random Forest Classifier'],
    'Score' : [ acc_dtc, acc_rd_clf]
})


models.sort_values(by = 'Score', ascending = False)
```

Out[48]:

	Model	Score

	Model	Score
1	Random Forest Classifier	0.954870
0	Decision Tree Classifier	0.945502

In [ ]: