Random Forest classifier and its comparison on Hotel booking pridictions

In [70]:

```
# importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix, clas
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import folium #new lib for geo maps
from folium.plugins import HeatMap #new lib
import plotly.express as px #new lib for interactive ploting

#plt.style.use('fivethirtyeight')
#%matplotlib inline
pd.set_option('display.max_columns', 30)
```

In [71]:
```
pip install folium

```

```
Requirement already satisfied: folium in ./opt/anaconda3/lib/python3.
7/site-packages (0.14.0)
Requirement already satisfied: jinja2>=2.9 in ./opt/anaconda3/lib/pyt
hon3.7/site-packages (from folium) (2.10.3)
Requirement already satisfied: branca>=0.6.0 in ./opt/anaconda3/lib/p
ython3.7/site-packages (from folium) (0.6.0)
Requirement already satisfied: requests in ./opt/anaconda3/lib/python
3.7/site-packages (from folium) (2.22.0)
Requirement already satisfied: numpy in ./opt/anaconda3/lib/python3.7
/site-packages (from folium) (1.17.2)
Requirement already satisfied: MarkupSafe>=0.23 in ./opt/anaconda3/li
b/python3.7/site-packages (from jinja2>=2.9->folium) (1.1.1)
Requirement already satisfied: idna<2.9,>=2.5 in ./opt/anaconda3/lib/
python3.7/site-packages (from requests->folium) (2.8)
Requirement already satisfied: chardet<3.1.0,>=3.0.2 in ./opt/anacond
a3/lib/python3.7/site-packages (from requests->folium) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in ./opt/anaconda3/
lib/python3.7/site-packages (from requests->folium) (2019.9.11)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.
1 in ./opt/anaconda3/lib/python3.7/site-packages (from requests->foli
um) (1.24.2)
Note: you may need to restart the kernel to use updated packages.
```

In [72]:
```
pip install plotly
```

```
Requirement already satisfied: plotly in ./opt/anaconda3/lib/python3.
7/site-packages (5.15.0)
Requirement already satisfied: packaging in ./opt/anaconda3/lib/pytho
n3.7/site-packages (from plotly) (19.2)
Requirement already satisfied: tenacity>=6.2.0 in ./opt/anaconda3/lib
/python3.7/site-packages (from plotly) (8.2.2)
Requirement already satisfied: pyparsing>=2.0.2 in ./opt/anaconda3/li
b/python3.7/site-packages (from packaging->plotly) (2.4.2)
Requirement already satisfied: six in ./opt/anaconda3/lib/python3.7/s
ite-packages (from packaging->plotly) (1.12.0)
Note: you may need to restart the kernel to use updated packages.
```

In [73]:
```
# reading data
df = pd.read_csv('/Users/macbookpro/Desktop/hotel_bookings.csv')
df.head(15)
```

Out[73]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_num
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	
5	Resort Hotel	0	14	2015	July	
6	Resort Hotel	0	0	2015	July	
7	Resort Hotel	0	9	2015	July	
8	Resort Hotel	1	85	2015	July	
9	Resort Hotel	1	75	2015	July	
10	Resort Hotel	1	23	2015	July	
11	Resort Hotel	0	35	2015	July	
12	Resort Hotel	0	68	2015	July	
13	Resort Hotel	0	18	2015	July	
14	Resort Hotel	0	37	2015	July	

15 rows × 32 columns

In [74]:
```
1 df.describe()
```

Out[74]:

	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_
count	119390.000000	119390.000000	119390.000000	119390.000000	1
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	

In [6]:
```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 119390 entries, 0 to 119389
Data columns (total 32 columns):
hotel                           119390 non-null object
is_canceled                     119390 non-null int64
lead_time                       119390 non-null int64
arrival_date_year               119390 non-null int64
arrival_date_month              119390 non-null object
arrival_date_week_number        119390 non-null int64
arrival_date_day_of_month       119390 non-null int64
stays_in_weekend_nights         119390 non-null int64
stays_in_week_nights            119390 non-null int64
adults                          119390 non-null int64
children                        119386 non-null float64
babies                          119390 non-null int64
meal                            119390 non-null object
country                         118902 non-null object
market_segment                  119390 non-null object
distribution_channel            119390 non-null object
is_repeated_guest               119390 non-null int64
previous_cancellations          119390 non-null int64
previous_bookings_not_canceled  119390 non-null int64
reserved_room_type              119390 non-null object
assigned_room_type              119390 non-null object
booking_changes                 119390 non-null int64
deposit_type                    119390 non-null object
agent                           103050 non-null float64
company                         6797 non-null float64
days_in_waiting_list            119390 non-null int64
customer_type                   119390 non-null object
adr                             119390 non-null float64
required_car_parking_spaces     119390 non-null int64
total_of_special_requests       119390 non-null int64
reservation_status              119390 non-null object
reservation_status_date         119390 non-null object
dtypes: float64(4), int64(16), object(12)
memory usage: 29.1+ MB
```

In [75]:
```
1 #checking for null values
2
3 null = pd.DataFrame({'Null Values' : df.isna().sum(), 'Percentage Null
4 null
```

Out[75]:

	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
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
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 [76]:
```
# filling null values with zero

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

In [77]:
```
df.shape
```

Out[77]: (119390, 32)

In [78]:
```
# adults, babies and children cant be zero at same time, so droppi

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

Out[78]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week
2224	Resort Hotel	0	1	2015	October	
2409	Resort Hotel	0	0	2015	October	
3181	Resort Hotel	0	36	2015	November	
3684	Resort Hotel	0	165	2015	December	
3708	Resort Hotel	0	165	2015	December	
...	...	...	...	...	...	
115029	City Hotel	0	107	2017	June	
115091	City Hotel	0	1	2017	June	
116251	City Hotel	0	44	2017	July	
116534	City Hotel	0	2	2017	July	
117087	City Hotel	0	170	2017	July	

180 rows × 32 columns

In [79]:
```
df = df[~filter]
df
```

Out[79]:

weekend_nights	stays_in_week_nights	adults	children	babies	meal	country	market_segment
0	0	2	0.0	0	BB	PRT	Direct
0	0	2	0.0	0	BB	PRT	Direct
0	1	1	0.0	0	BB	GBR	Direct
0	1	1	0.0	0	BB	GBR	Corporate
0	2	2	0.0	0	BB	GBR	Online TA
...	...	...	...	...	...	...	...
2	5	2	0.0	0	BB	BEL	Offline TA/TO
2	5	3	0.0	0	BB	FRA	Online TA
2	5	2	0.0	0	BB	DEU	Online TA
2	5	2	0.0	0	BB	GBR	Online TA
2	7	2	0.0	0	HB	DEU	Online TA

In [12]:
```

df[filter]
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:1: UserWarning: Boolean Series key will be reindexed to
match DataFrame index.
  """Entry point for launching an IPython kernel.
```

Out[12]:

hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_numbe

0 rows × 32 columns

In [13]:
```
1 df.shape
```

Out[13]: (119210, 32)

- Exploratory Data Analysis (EDA)

From where the most guests are coming.....?????

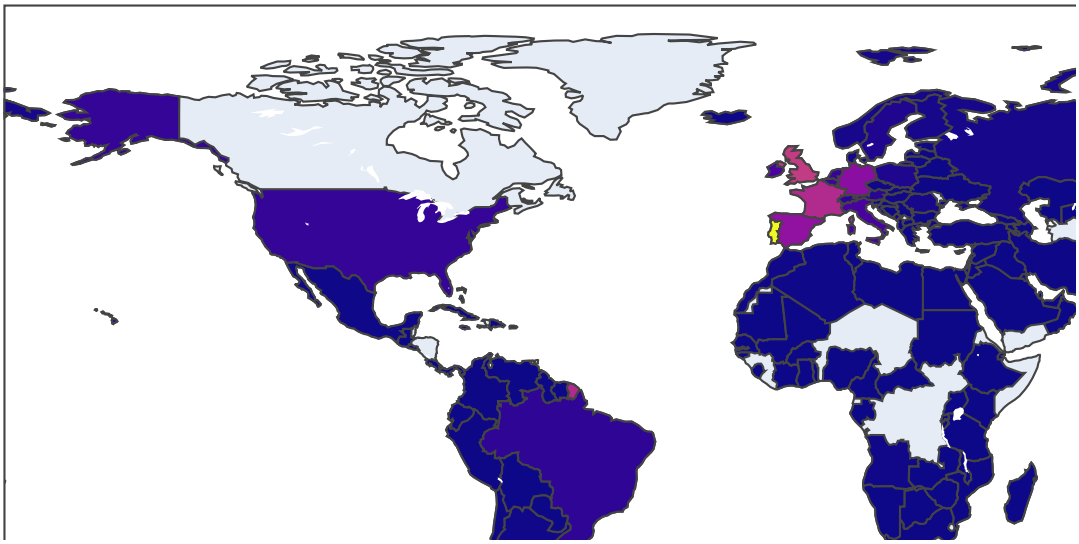
In [80]:
```
1 country_wise_guests = df[df['is_canceled'] == 0]['country'].value_
2 country_wise_guests.columns = ['country', 'No of guests']
3 country_wise_guests
```

Out[80]:

	country	No of guests
0	PRT	20977
1	GBR	9668
2	FRA	8468
3	ESP	6383
4	DEU	6067
...	...	...
161	DJI	1
162	PLW	1
163	SYC	1
164	PYF	1
165	BHS	1

166 rows × 2 columns

In [82]:
```
basemap = folium.Map()
guests_map = px.choropleth(country_wise_guests, locations = countr
             color = country_wise_guests['No of guests'], hover_nam
guests_map.show()
```



People from all over the world are staying in these two hotels. Most guests are from Portugal and other countries in Europe.

In [16]:
```
df.head()
```

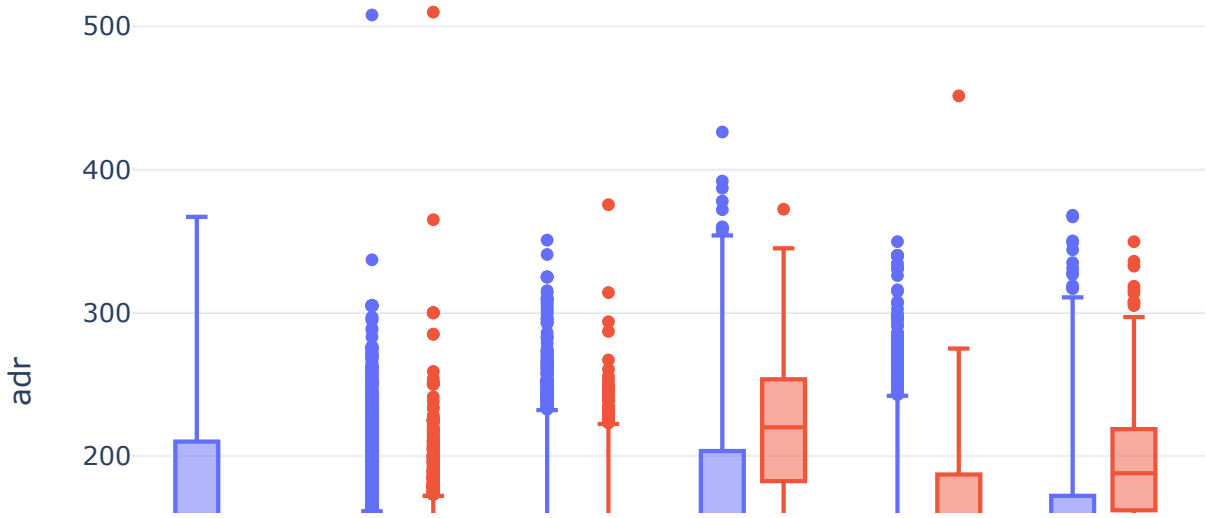
Out[16]:

hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
Resort Hotel	0	342	2015	July	27
Resort Hotel	0	737	2015	July	27
Resort Hotel	0	7	2015	July	27
Resort Hotel	0	13	2015	July	27
Resort Hotel	0	14	2015	July	27

ows × 32 columns

In [83]:
```
#adr = average daily rate (price)
data = df[df['is_canceled'] == 0]

px.box(data_frame = data, x = 'reserved_room_type', y = 'adr', col
```



The figure shows that the average price per room depends on its type and the standard deviation.

How does the price vary per night over the year.....?

In [18]:
```
data_resort = df[(df['hotel'] == 'Resort Hotel') & (df['is_cancele
data_city = df[(df['hotel'] == 'City Hotel') & (df['is_canceled']
```

In [19]:
```
1 data_resort
```

Out[19]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_
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	
...	...	...	...	...	...	
40055	Resort Hotel	0	212	2017	August	
40056	Resort Hotel	0	169	2017	August	
40057	Resort Hotel	0	204	2017	August	
40058	Resort Hotel	0	211	2017	August	
40059	Resort Hotel	0	161	2017	August	

28927 rows × 32 columns

In [84]:
```
1 data_city
```

Out[84]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_number
30	City Hotel	0	6	2015	July	27
36	City Hotel	0	3	2015	July	27
70	City Hotel	0	43	2015	July	27
71	City Hotel	0	43	2015	July	27
72	City Hotel	0	43	2015	July	27
...	...	...	...	...	...	...
35	City Hotel	0	23	2017	August	35
36	City Hotel	0	102	2017	August	35
37	City Hotel	0	34	2017	August	35
38	City Hotel	0	109	2017	August	35
39	City Hotel	0	205	2017	August	35

4 rows × 32 columns

In [21]:
```
resort_hotel = data_resort.groupby(['arrival_date_month'])['adr'].
resort_hotel
```

Out[21]:

	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 [22]:
```
city_hotel=data_city.groupby(['arrival_date_month'])['adr'].mean()
city_hotel
```

Out[22]:

	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 [23]:
```
final_hotel = resort_hotel.merge(city_hotel, on = 'arrival_date_mo
final_hotel
```

Out[23]:

	arrival_date_month	adr_x	adr_y
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
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 [24]:
```
final_hotel.columns = ['month', 'price_for_resort', 'price_for_cit
final_hotel
```

Out[24]:

	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
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

Assignment task: months in above DF is not sorted, sort the DF

In [85]:
```
plt.figure(figsize = (20,10))

px.line(final_hotel, x = 'month', y = ['price_for_resort','price_f
        title = 'Room price per night over the Months', template =
```


```
<Figure size 1440x720 with 0 Axes>
```

Assigment Task: pass the sorted dataframe in above plot

This plot clearly shows that prices in the Resort Hotel are much higher during the summer and prices of city hotel varies less and is most expensive during Spring and Autumn .

Which are the most busy months?

In [26]:
```
resort_guests = data_resort['arrival_date_month'].value_counts().r
resort_guests.columns=['month','no of guests']
resort_guests
```

Out[26]:

	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 [27]:
```
1 city_guests = data_city['arrival_date_month'].value_counts().reset
2 city_guests.columns=['month','no of guests']
3 city_guests
```

Out[27]:

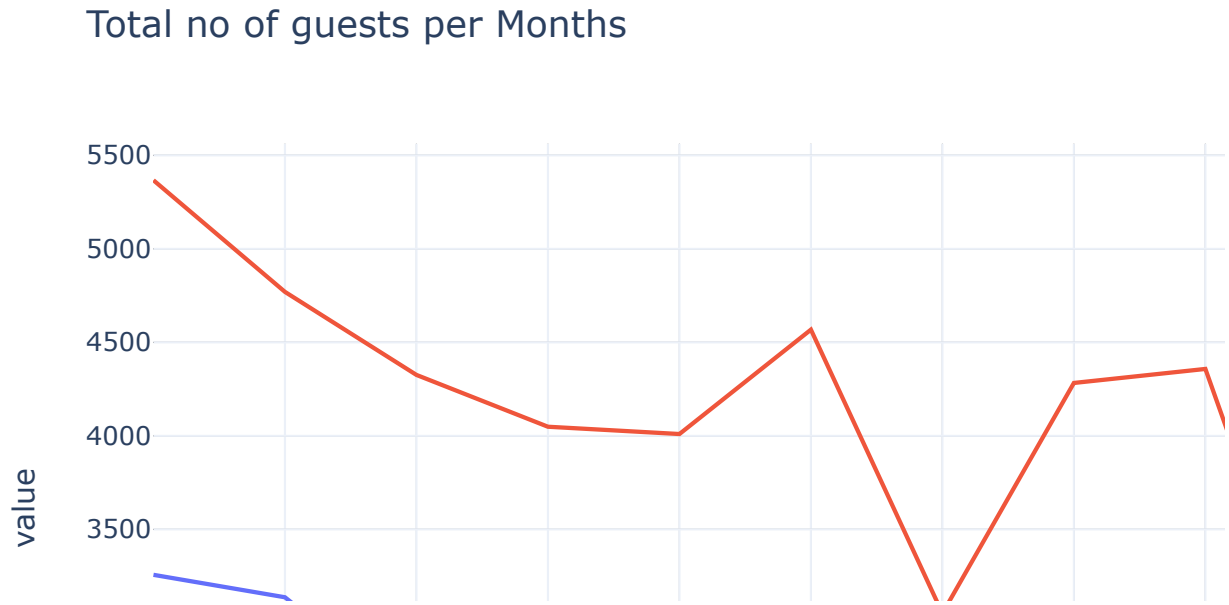
	month	no of guests
0	August	5367
1	July	4770
2	May	4568
3	June	4358
4	October	4326
5	September	4283
6	March	4049
7	April	4010
8	February	3051
9	November	2676
10	December	2377
11	January	2249

In [28]:
```
final_guests = resort_guests.merge(city_guests,on='month')
final_guests.columns=['month','no of guests in resort','no of gues
final_guests
```

Out[28]:

	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
8	June	2037	4358
9	December	2014	2377
10	November	1975	2676
11	January	1866	2249

In [29]:
```
1 px.line(final_guests, x = 'month', y = ['no of guests in resort',
2         title='Total no of guests per Months', template = 'plotly_
```



How long do people stay at the hotels....?

In [30]:
```
filter = df['is_canceled'] == 0
data = df[filter]
data.head()
```

Out[30]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	arrival_date_week_numb
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	

5 rows × 32 columns

In [86]:
```
1 data['total_nights'] = data['stays_in_weekend_nights'] + data['sta
2 data.head()
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)
```

Out[86]:

reserved_room_type	assigned_room_type	booking_changes	deposit_type	agent	company	day
C	C	3	No Deposit	0.0	0.0	
C	C	4	No Deposit	0.0	0.0	
A	C	0	No Deposit	0.0	0.0	
A	A	0	No Deposit	304.0	0.0	
A	A	0	No Deposit	240.0	0.0	

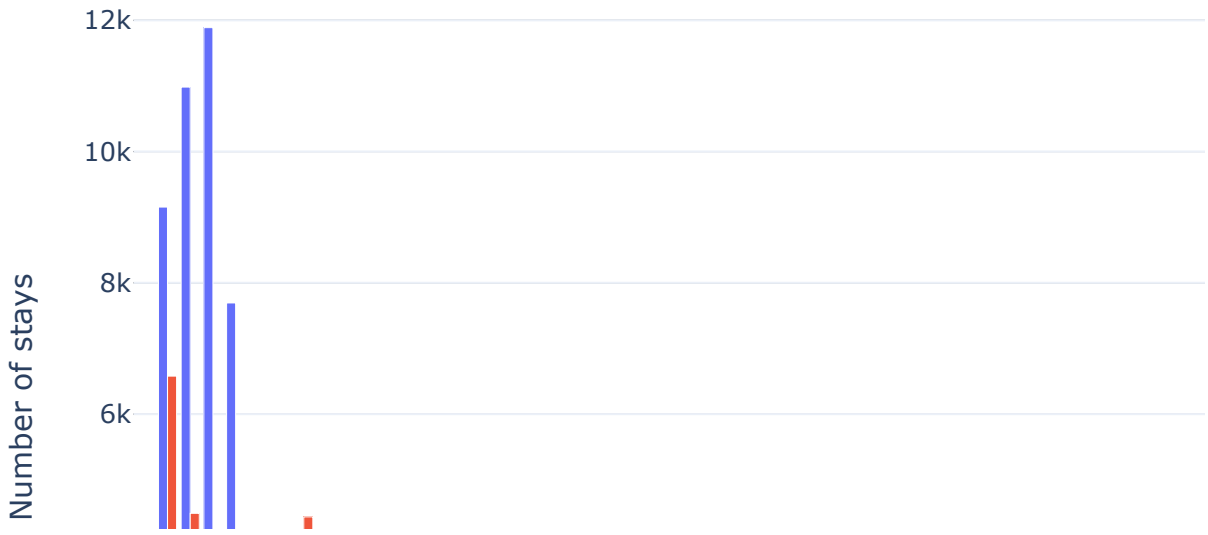
In [32]:
```
stay = data.groupby(['total_nights', 'hotel']).agg('count').reset_
stay = stay.iloc[:, :3]
stay = stay.rename(columns={'is_canceled':'Number of stays'})
stay
```

Out[32]:

	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 [33]:
```
px.bar(data_frame = stay, x = 'total_nights', y = 'Number of stays
        template = 'plotly_white')
```

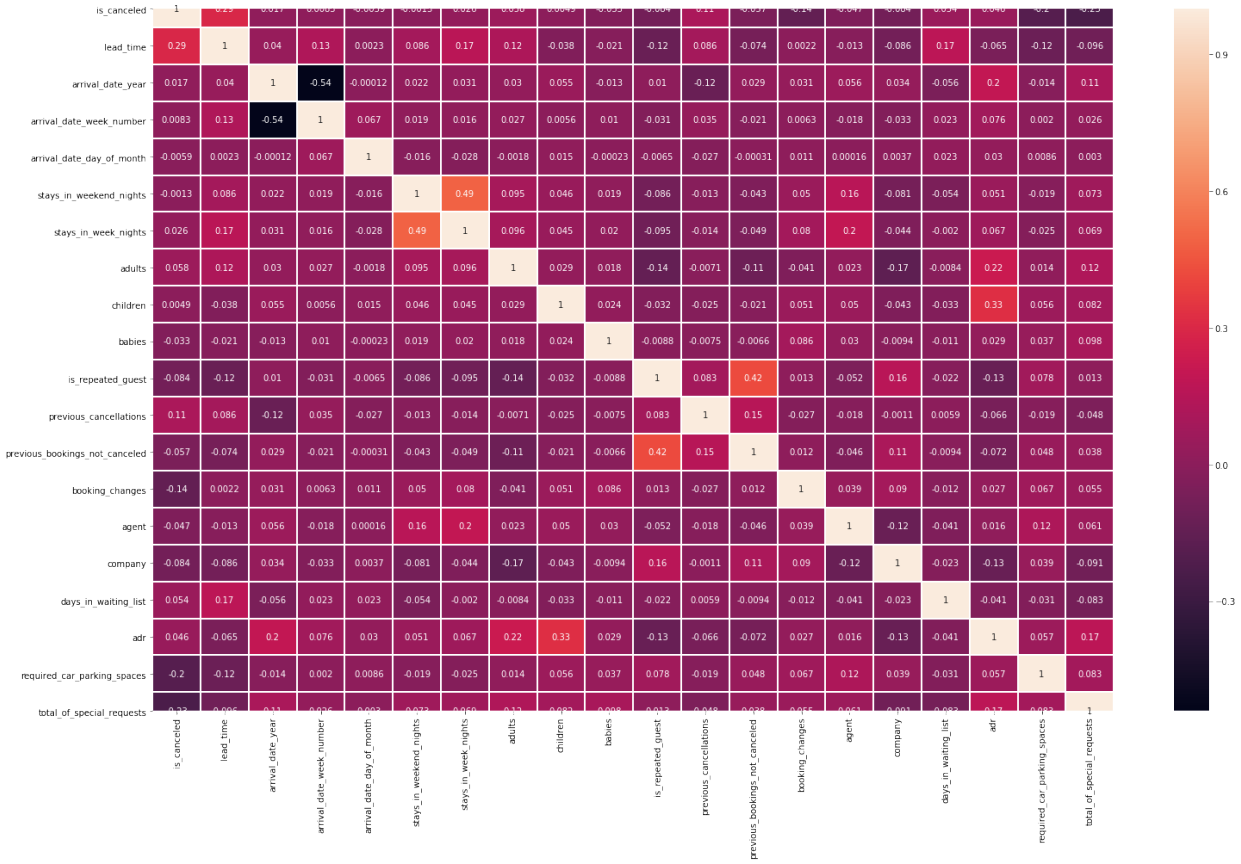


Data Pre Processing

In [34]:
```
plt.figure(figsize = (25, 15))

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

```



In [35]:
```
df.shape
```

Out[35]: (119210, 32)

In [36]:
```
correlation = df.corr()['is_canceled'].abs().sort_values(ascending
correlation
```

Out[36]:
```
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 [37]:
```
# dropping columns that are not useful

useless_col = ['days_in_waiting_list', 'arrival_date_year', 'arriv
               'reservation_status', 'country', 'days_in_waiting_l

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

In [38]:
```
df.head()
```

Out[38]:

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

In [39]:
```
cat_cols=list(df.select_dtypes(['object']).columns)
cat_cols
```

Out[39]:
```
['hotel',
 'arrival_date_month',
 'meal',
 'market_segment',
 'distribution_channel',
 'reserved_room_type',
 'deposit_type',
 'customer_type',
 'reservation_status_date']
```

In [40]:
```
cat_df = df[cat_cols]
cat_df.head()
```

Out[40]:

	hotel	arrival_date_month	meal	market_segment	distribution_channel	reserved_room_type
0	Resort Hotel	July	BB	Direct	Direct	C
1	Resort Hotel	July	BB	Direct	Direct	C
2	Resort Hotel	July	BB	Direct	Direct	A
3	Resort Hotel	July	BB	Corporate	Corporate	A
4	Resort Hotel	July	BB	Online TA	TA/TO	A

In [41]:
```
cat_df['reservation_status_date'] = pd.to_datetime(cat_df['reserva

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
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:1: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:4: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:5: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)
```

In [42]:
```
1 cat_df.drop(['reservation_status_date','arrival_date_month'] , axi
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/pandas/co
re/frame.py:4102: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)
```

In [43]:
```
1 cat_df.head(15)
```

Out[43]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	cus
0	Resort Hotel	BB	Direct	Direct	C	No Deposit	
1	Resort Hotel	BB	Direct	Direct	C	No Deposit	
2	Resort Hotel	BB	Direct	Direct	A	No Deposit	
3	Resort Hotel	BB	Corporate	Corporate	A	No Deposit	
4	Resort Hotel	BB	Online TA	TA/TO	A	No Deposit	
5	Resort Hotel	BB	Online TA	TA/TO	A	No Deposit	
6	Resort Hotel	BB	Direct	Direct	C	No Deposit	
7	Resort Hotel	FB	Direct	Direct	C	No Deposit	
8	Resort Hotel	BB	Online TA	TA/TO	A	No Deposit	
9	Resort Hotel	HB	Offline TA/TO	TA/TO	D	No Deposit	
10	Resort Hotel	BB	Online TA	TA/TO	E	No Deposit	
11	Resort Hotel	HB	Online TA	TA/TO	D	No Deposit	
12	Resort Hotel	BB	Online TA	TA/TO	D	No Deposit	
13	Resort Hotel	HB	Online TA	TA/TO	G	No Deposit	
14	Resort Hotel	BB	Online TA	TA/TO	E	No Deposit	

In [44]:
```
# 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' 'Gr
oups'
 '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 [45]:
```
# encoding categorical variables

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

cat_df['meal'] = cat_df['meal'].map({'BB' : 0, 'FB': 1, 'HB': 2, 'SC

cat_df['market_segment'] = cat_df['market_segment'].map({'Direct': 0
                                                         'Compleme

cat_df['distribution_channel'] = cat_df['distribution_channel'].map(
```

```
11
12
13cat_df['reserved_room_type'] = cat_df['reserved_room_type'].map({'C'
14
15
16cat_df['deposit_type'] = cat_df['deposit_type'].map({'No Deposit': 0
17
18cat_df['customer_type'] = cat_df['customer_type'].map({'Transient':
19
20cat_df['year'] = cat_df['year'].map({2015: 0, 2014: 1, 2016: 2, 2017
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:3: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:5: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:8: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:11: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:14: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:16: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:18: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas
-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html
#returning-a-view-versus-a-copy)

/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/ipykernel
_launcher.py:20: SettingWithCopyWarning:


A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (http://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

Assignment Task: Please perform the above encoding using other coding techniques (Label encoder, one hot etc)

In [46]:
```
1 cat_df.head(15)
```

Out[46]:

	hotel	meal	market_segment	distribution_channel	reserved_room_type	deposit_type	cust
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	1	0	
3	0	0	1	1	1	0	
4	0	0	2	2	1	0	
5	0	0	2	2	1	0	
6	0	0	0	0	0	0	
7	0	1	0	0	0	0	
8	0	0	2	2	1	0	
9	0	2	3	2	2	0	
10	0	0	2	2	3	0	
11	0	2	2	2	2	0	
12	0	0	2	2	2	0	
13	0	2	2	2	4	0	
14	0	0	2	2	3	0	

Now creating Numerical Datafram

In [47]:
```
num_df = df.drop(columns = cat_cols, axis = 1)
num_df.drop('is_canceled', axis = 1, inplace = True)
num_df
```

Out[47]:

	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nigh
0	342	27	1	
1	737	27	1	
2	7	27	1	
3	13	27	1	
4	14	27	1	
...	...	...	...	
119385	23	35	30	
119386	102	35	31	
119387	34	35	31	
119388	109	35	31	
119389	205	35	29	

119210 rows × 16 columns

In [48]:
```
num_df.var()
```

Out[48]:
```
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 [49]:
```
# 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_v
num_df['arrival_date_day_of_month'] = np.log(num_df['arrival_date_
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)
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/pandas/co
re/series.py:853: RuntimeWarning:

invalid value encountered in log
```

In [50]:
```
num_df.var()
```

Out[50]:
```
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 [51]:
```
# checking for null values

null = pd.DataFrame({'Null Values' : num_df.isna().sum(), 'Percent
null
```

Out[51]:

	Null Values	Percentage Null Values
lead_time	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	0	0.000000
babies	0	0.000000
is_repeated_guest	0	0.000000
previous_cancellations	0	0.000000
previous_bookings_not_canceled	0	0.000000
agent	0	0.000000
company	0	0.000000
adr	1	0.000839
required_car_parking_spaces	0	0.000000
total_of_special_requests	0	0.000000

In [52]:
```
num_df['adr'] = num_df['adr'].fillna(value = num_df['adr'].mean())
```

In [53]:
```
1 num_df.head(15)
```

Out[53]:

	lead_time	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	s
0	5.837730	3.332205	0.693147	0	
1	6.603944	3.332205	0.693147	0	
2	2.079442	3.332205	0.693147	0	
3	2.639057	3.332205	0.693147	0	
4	2.708050	3.332205	0.693147	0	
5	2.708050	3.332205	0.693147	0	
6	0.000000	3.332205	0.693147	0	
7	2.302585	3.332205	0.693147	0	
8	4.454347	3.332205	0.693147	0	
9	4.330733	3.332205	0.693147	0	
10	3.178054	3.332205	0.693147	0	
11	3.583519	3.332205	0.693147	0	
12	4.234107	3.332205	0.693147	0	
13	2.944439	3.332205	0.693147	0	
14	3.637586	3.332205	0.693147	0	

In [54]:
```
1 X = pd.concat([cat_df, num_df], axis = 1)
2 y = df['is_canceled']
```

In [55]:
```
1 X.shape, y.shape
```

Out[55]: ((119210, 26), (119210,))

In [58]:
```
1 # splitting data into training set and test set
2
3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_siz
```

In [61]:
```
1 X_train.shape
```

Out[61]: (95368, 26)

In [62]:
```
1 X_test.shape
```

Out[62]: (23842, 26)

In [63]:
```
1 y_train.head(), y_test.head()
```

Out[63]:
```
(43178     0
 62123     1
 97743     0
 5719      0
 110104    0
 Name: is_canceled, dtype: int64, 58357    1
 90360    0
 80845    1
 66861    1
 48085    0
 Name: is_canceled, dtype: int64)
```

Models Training

- Logistic Regression

In [64]:
```
lr = LogisticRegression()
lr.fit(X_train, y_train)

y_pred_lr = lr.predict(X_test)

acc_lr = accuracy_score(y_test, y_pred_lr)
conf = confusion_matrix(y_test, y_pred_lr)
clf_report = classification_report(y_test, y_pred_lr)

print(f"Accuracy Score of Logistic Regression is : {acc_lr}")
print(f"Confusion Matrix : \n{conf}")
print(f"Classification Report : \n{clf_report}")
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/sklearn/l
inear_model/logistic.py:432: FutureWarning:

Default solver will be changed to 'lbfgs' in 0.22. Specify a solver t
o silence this warning.


Accuracy Score of Logistic Regression is : 0.8078181360624108
Confusion Matrix :
[[14188   852]
 [ 3730  5072]]
Classification Report :
              precision    recall  f1-score   support

           0       0.79      0.94      0.86     15040
           1       0.86      0.58      0.69      8802

    accuracy                           0.81     23842
   macro avg       0.82      0.76      0.77     23842
weighted avg       0.82      0.81      0.80     23842
```

- KNN

In [65]:
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

y_pred_knn = knn.predict(X_test)

acc_knn = accuracy_score(y_test, y_pred_knn)
conf = confusion_matrix(y_test, y_pred_knn)
clf_report = classification_report(y_test, y_pred_knn)

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

```
Accuracy Score of KNN is : 0.892123144031541
Confusion Matrix :
[[14487   553]
 [ 2019  6783]]
Classification Report :
              precision    recall  f1-score   support

           0       0.88      0.96      0.92     15040
           1       0.92      0.77      0.84      8802

    accuracy                           0.89     23842
   macro avg       0.90      0.87      0.88     23842
weighted avg       0.90      0.89      0.89     23842
```

- Decision Tree Classifier

In [66]:
```
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 : 0.9500461370690378
Confusion Matrix :
[[14438   602]
 [  589  8213]]
Classification Report :
              precision    recall  f1-score   support

           0       0.96      0.96      0.96     15040
           1       0.93      0.93      0.93      8802

    accuracy                           0.95     23842
   macro avg       0.95      0.95      0.95     23842
weighted avg       0.95      0.95      0.95     23842
```

- Random Forest Classifier

In [67]:
```
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}")
```

```
/Users/macbookpro/opt/anaconda3/lib/python3.7/site-packages/sklearn/ensemble/forest.py:245: FutureWarning:

The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.


Accuracy Score of Random Forest is : 0.9437966613539133
Confusion Matrix :
[[14824   216]
 [ 1124  7678]]
Classification Report :
              precision    recall  f1-score   support

           0       0.93      0.99      0.96     15040
           1       0.97      0.87      0.92      8802

    accuracy                           0.94     23842
   macro avg       0.95      0.93      0.94     23842
weighted avg       0.95      0.94      0.94     23842
```

- Models Comparison

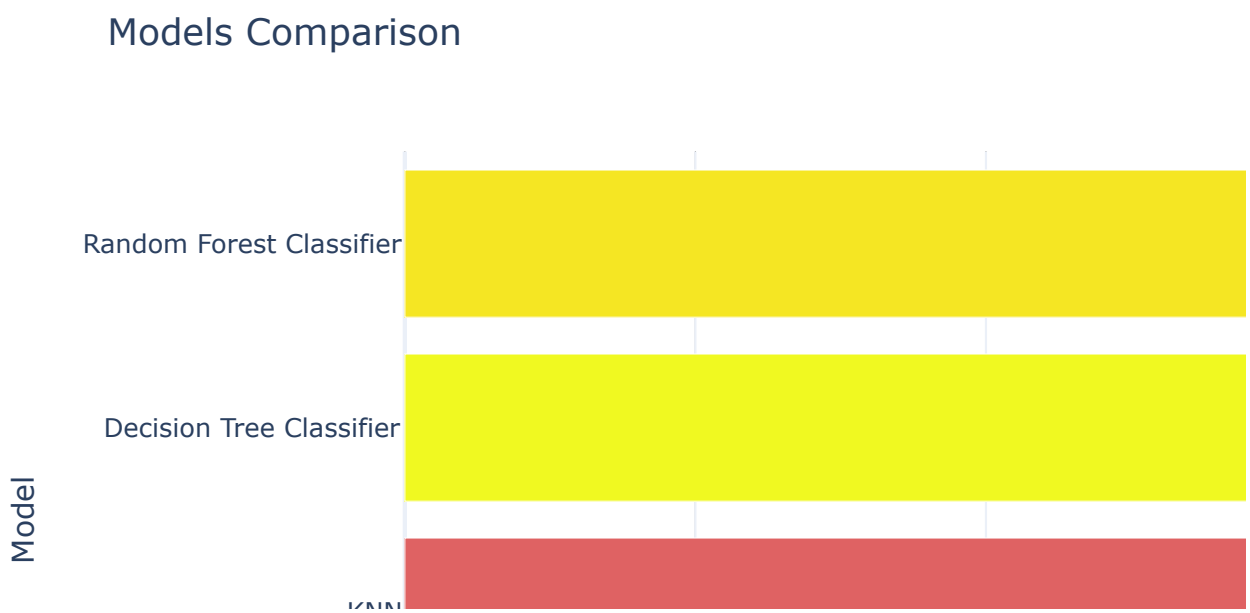
In [68]:
```
models = pd.DataFrame({
    'Model' : ['Logistic Regression', 'KNN', 'Decision Tree Classi
    'Score' : [acc_lr, acc_knn, acc_dtc, acc_rd_clf]
})


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

Out[68]:

	Model	Score
2	Decision Tree Classifier	0.950046
3	Random Forest Classifier	0.943797
1	KNN	0.892123
0	Logistic Regression	0.807818

In [69]:
```
1 px.bar(data_frame = models, x = 'Score', y = 'Model', color = 'Scor
```



In [ ]:
```
1
```