Import Dataset

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
df=pd.read_csv("Hotel Reservations.csv")
df.head()
```

Out[2]:

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

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

Out[3]:
```
Booking_ID  no_of_adults  no_of_children  no_of_weekend_nights  no_of_we
ek_nights  type_of_meal_plan  required_car_parking_space  room_type_rese
rved  lead_time  arrival_year  arrival_month  arrival_date  market_segme
nt_type  repeated_guest  no_of_previous_cancellations  no_of_previous_bo
okings_not_canceled  avg_price_per_room  no_of_special_requests  booking
_status
INN00001    2             0               1                     2
Meal Plan 1        0                          Room_Type 1         224
2017          10             2             Offline              0
0                              0                                     65.0
0               0                       Not_Canceled      1
INN24187    1             0               0                     2
Meal Plan 1        0                          Room_Type 4         13
2018          6              7             Corporate            0
0                              0                                     110.
00              0                       Not_Canceled      1
INN24181    2             0               0                     5
Meal Plan 1        0                          Room_Type 4         83
2018          7              5             Online               0
0                              0                                     127.
80              0                       Canceled          1
INN24182    2             0               1                     2
Meal Plan 1        0                          Room_Type 1         4
2017          12             4             Offline              0
0                              0                                     58.0
0               0                       Not_Canceled      1
INN24183    1             0               2                     1
Meal Plan 1        0                          Room_Type 1         72
2018          4              3             Online               0
0                              0                                     87.3
0               0                       Canceled          1

..
INN12086    3             0               2                     2
Meal Plan 1        0                          Room_Type 4         34
2018          8              28            Online               0
0                              0                                     150.
98              2                       Not_Canceled      1
INN12085    2             0               2                     1
Not Selected       0                          Room_Type 1         67
2018          3              19            Online               0
0                              0                                     85.5
0               0                       Canceled          1
INN12084    1             0               1                     2
Not Selected       0                          Room_Type 1         2
2018          2              5             Online               0
0                              0                                     57.9
8               1                       Not_Canceled      1
INN12083    2             1               0                     3
Meal Plan 1        0                          Room_Type 1         38
2018          3              24            Online               0
0                              0                                     133.
80              0                       Canceled          1
INN36275    2             0               1                     2
Meal Plan 1        0                          Room_Type 1         207
2018          12             30            Offline              0
0                              0                                     161.
```

```
67              0                        Not_Canceled      1
Length: 36275, dtype: int64
```

In [4]:
```
df["type_of_meal_plan"].value_counts()
```

Out[4]:
```
Meal Plan 1     27835
Not Selected     5130
Meal Plan 2      3305
Meal Plan 3         5
Name: type_of_meal_plan, dtype: int64
```

In [7]:
```
df[['Booking_ID',"type_of_meal_plan"]]
```

Out[7]:

	Booking_ID	type_of_meal_plan
0	INN00001	Meal Plan 1
1	INN00002	Not Selected
2	INN00003	Meal Plan 1
3	INN00004	Meal Plan 1
4	INN00005	Not Selected
...	...	...
36270	INN36271	Meal Plan 1
36271	INN36272	Meal Plan 1
36272	INN36273	Meal Plan 1
36273	INN36274	Not Selected
36274	INN36275	Meal Plan 1

36275 rows × 2 columns

In [8]:
```
df[['Booking_ID',"room_type_reserved"]]
```

Out[8]:

	Booking_ID	room_type_reserved
0	INN00001	Room_Type 1
1	INN00002	Room_Type 1
2	INN00003	Room_Type 1
3	INN00004	Room_Type 1
4	INN00005	Room_Type 1
...	...	...
36270	INN36271	Room_Type 4
36271	INN36272	Room_Type 1
36272	INN36273	Room_Type 1
36273	INN36274	Room_Type 1
36274	INN36275	Room_Type 1

36275 rows × 2 columns

In [10]:
```
df["room_type_reserved"].value_counts()
```

Out[10]:
```
Room_Type 1    28130
Room_Type 4     6057
Room_Type 6      966
Room_Type 2      692
Room_Type 5      265
Room_Type 7      158
Room_Type 3        7
Name: room_type_reserved, dtype: int64
```

In [12]:
```
df[['Booking_ID',"booking_status"]]
```

Out[12]:

	Booking_ID	booking_status
0	INN00001	Not_Canceled
1	INN00002	Not_Canceled
2	INN00003	Canceled
3	INN00004	Canceled
4	INN00005	Canceled
...	...	...
36270	INN36271	Not_Canceled
36271	INN36272	Canceled
36272	INN36273	Not_Canceled
36273	INN36274	Canceled
36274	INN36275	Not_Canceled

36275 rows × 2 columns

In [13]:
```
df["booking_status"].value_counts()
```

Out[13]:
```
Not_Canceled    24390
Canceled        11885
Name: booking_status, dtype: int64
```

Cleaning the data make it numberic value

In [14]:
```
data = df.replace({"type_of_meal_plan":{"Not Selected":0,"Meal Plan 1":1,'
                   "room_type_reserved":{"Room_Type 1":1,"Room_Type 2
                   "market_segment_type":{"Offline":0,"Online":1, "Co
                   "booking_status":{"Canceled":0,"Not_Canceled":1}}}
```

In [20]:
```
data.head()
```

Out[20]:

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

In [16]:
```
data[['Booking_ID',"type_of_meal_plan"]]
```

Out[16]:

	Booking_ID	type_of_meal_plan
0	INN00001	1
1	INN00002	0
2	INN00003	1
3	INN00004	1
4	INN00005	0
...	...	...
36270	INN36271	1
36271	INN36272	1
36272	INN36273	1
36273	INN36274	0
36274	INN36275	1

36275 rows × 2 columns

In [17]:
```
data[['Booking_ID',"booking_status"]]
```

Out[17]:

	Booking_ID	booking_status
0	INN00001	1
1	INN00002	1
2	INN00003	0
3	INN00004	0
4	INN00005	0
...	...	...
36270	INN36271	1
36271	INN36272	0
36272	INN36273	1
36273	INN36274	0
36274	INN36275	1

36275 rows × 2 columns

In [18]:
```
data[['Booking_ID',"room_type_reserved"]]
```

Out[18]:

	Booking_ID	room_type_reserved
0	INN00001	1
1	INN00002	1
2	INN00003	1
3	INN00004	1
4	INN00005	1
...	...	...
36270	INN36271	4
36271	INN36272	1
36272	INN36273	1
36273	INN36274	1
36274	INN36275	1

36275 rows × 2 columns

In [15]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 36275 entries, 0 to 36274
Data columns (total 19 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Booking_ID                            36275 non-null  object
 1   no_of_adults                          36275 non-null  int64
 2   no_of_children                        36275 non-null  int64
 3   no_of_weekend_nights                  36275 non-null  int64
 4   no_of_week_nights                     36275 non-null  int64
 5   type_of_meal_plan                     36275 non-null  int64
 6   required_car_parking_space            36275 non-null  int64
 7   room_type_reserved                    36275 non-null  int64
 8   lead_time                             36275 non-null  int64
 9   arrival_year                          36275 non-null  int64
 10  arrival_month                         36275 non-null  int64
 11  arrival_date                          36275 non-null  int64
 12  market_segment_type                   36275 non-null  int64
 13  repeated_guest                        36275 non-null  int64
 14  no_of_previous_cancellations          36275 non-null  int64
 15  no_of_previous_bookings_not_canceled  36275 non-null  int64
 16  avg_price_per_room                    36275 non-null  float64
 17  no_of_special_requests                36275 non-null  int64
 18  booking_status                        36275 non-null  int64
dtypes: float64(1), int64(17), object(1)
memory usage: 5.3+ MB
```

In [5]:
```
data.describe()
```

Out[5]:

	no_of_adults	no_of_children	no_of_weekend_nights	no_of_week_nights	type_of_m
count	36275.000000	36275.000000	36275.000000	36275.000000	3627
mean	1.844962	0.105279	0.810724	2.204300	
std	0.518715	0.402648	0.870644	1.410905	
min	0.000000	0.000000	0.000000	0.000000	
25%	2.000000	0.000000	0.000000	1.000000	
50%	2.000000	0.000000	1.000000	2.000000	
75%	2.000000	0.000000	2.000000	3.000000	
max	4.000000	10.000000	7.000000	17.000000	

In [6]:
```
data.drop(["booking_status"],axis=1).corrwith(data["booking_status"])
```

Out[6]:
```
no_of_adults                          -0.086920
no_of_children                        -0.033078
no_of_weekend_nights                  -0.061563
no_of_week_nights                     -0.092996
type_of_meal_plan                     -0.049374
required_car_parking_space             0.086185
room_type_reserved                    -0.022986
lead_time                             -0.438538
arrival_year                          -0.179529
arrival_month                          0.011233
arrival_date                          -0.010629
market_segment_type                    0.077877
repeated_guest                         0.107287
no_of_previous_cancellations           0.033728
no_of_previous_bookings_not_canceled   0.060179
avg_price_per_room                    -0.142569
no_of_special_requests                 0.253070
dtype: float64
```

Build the data training and Test Set

In [7]:
```
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
X = data.drop(["Booking_ID","arrival_month","arrival_date", "booking_statu
y = data["booking_status"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
```

Make model pipeline

In [8]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import GaussianNB

model_pipeline = []
model_pipeline.append(LogisticRegression(solver='liblinear'))
model_pipeline.append(SVC())
model_pipeline.append(KNeighborsClassifier())
model_pipeline.append(DecisionTreeClassifier())
model_pipeline.append(RandomForestClassifier())
model_pipeline.append(GaussianNB())
```

In [9]:
```
from sklearn import metrics
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix

model_list = ["Logistic Regression", "SVM", "KNN", "Decision Tree", "Rando
acc_list =[]
auc_list = []
cm_list = []

for model in model_pipeline:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    acc_list.append(metrics.accuracy_score(y_test,y_pred))
    fpr, tpr ,_tresholds = metrics.roc_curve(y_test,y_pred)
    auc_list.append(round(metrics.auc(fpr,tpr),2))
    cm_list.append(confusion_matrix(y_test,y_pred))
```

Make heatmap result from test data set

Finding the Best to find the best model

In [10]:
```
result_df = pd.DataFrame({"Model":model_list,"Accuracy" :acc_list, "AUC":a
result_df
```

Out[10]:

	Model	Accuracy	AUC
0	Logistic Regression	0.789324	0.73
1	SVM	0.762509	0.67
2	KNN	0.805530	0.76
3	Decision Tree	0.857572	0.84
4	Random Forest	0.886726	0.86
5	Naive Bayes	0.451508	0.58

the result say that Random Forest are the best model for the data set to make predicion

Make a prediction

In [11]:
```
from sklearn.ensemble import RandomForestClassifier

clf=RandomForestClassifier(n_estimators=100)

clf.fit(X_train,y_train)

y_pred=clf.predict(X_test)
```

In [12]:
```
#checking the accuracy pf the model
from sklearn import metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8868933255367137
```

Preparing the data, here I use data from the updateed version of the original data

In [13]:
```
guest = 36273 # can change by the guest Booking ID
data1=data.iloc[[guest]].drop(["Booking_ID","arrival_month","arrival_date'
#limiting the input data cause random forest just eccept 15 feature. so I
data1.values.tolist()
```

Out[13]:
```
[[2.0,
  0.0,
  0.0,
  3.0,
  0.0,
  0.0,
  1.0,
  63.0,
  2018.0,
  1.0,
  0.0,
  0.0,
  0.0,
  94.5,
  0.0]]
```

In [15]:
```
code = clf.predict(data1)
if code == 0:
    print("Not Cancel")
else:
    print("cancel")
```

```
Not Cancel
```

In [ ]: