

# Hyperparameter tuning

August 1, 2021

```
[ ]: #pip install pycountry
```

```
[ ]: #pip install catboost
```

```
[ ]: #pip install lightgbm
```

```
[1]: pip install folium
```

Requirement already satisfied: folium in /opt/conda/lib/python3.7/site-packages (0.12.1)

Requirement already satisfied: numpy in /opt/conda/lib/python3.7/site-packages (from folium) (1.18.4)

Requirement already satisfied: jinja2>=2.9 in /opt/conda/lib/python3.7/site-packages (from folium) (2.11.2)

Requirement already satisfied: requests in /opt/conda/lib/python3.7/site-packages (from folium) (2.23.0)

Requirement already satisfied: branca>=0.3.0 in /opt/conda/lib/python3.7/site-packages (from folium) (0.4.2)

Requirement already satisfied: MarkupSafe>=0.23 in /opt/conda/lib/python3.7/site-packages (from jinja2>=2.9->folium) (1.1.1)

Requirement already satisfied: idna<3,>=2.5 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (2.9)

Requirement already satisfied: urllib3!=1.25.0,!1.25.1,<1.26,>=1.21.1 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (1.25.9)

Requirement already satisfied: certifi>=2017.4.17 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (2020.4.5.2)

Requirement already satisfied: chardet<4,>=3.0.2 in /opt/conda/lib/python3.7/site-packages (from requests->folium) (3.0.4)

Note: you may need to restart the kernel to use updated packages.

```
[2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import missingno as msno
import xgboost as xgb
import pycountry as pc
import warnings
```

```
warnings.filterwarnings('ignore')
from pandas import DataFrame, read_csv
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import LabelEncoder
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 xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn.ensemble import ExtraTreesClassifier
from lightgbm import LGBMClassifier
from sklearn.ensemble import VotingClassifier
import matplotlib.ticker as mtick
pd.options.display.max_columns = None

import folium
from folium.plugins import HeatMap
#import plotly.express as px
%matplotlib inline

plt.style.use('fivethirtyeight')
%matplotlib inline
pd.set_option('display.max_columns', 32)
```

```
[3]: df = pd.read_csv('hotel_bookings.csv')
      #print(df)
```

```
[4]: df.head()
```

```
[4]:
```

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_month	\
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	

	arrival_date_week_number	arrival_date_day_of_month	\
0	27	1	
1	27	1	

2		27		1	
3		27		1	
4		27		1	

	stays_in_weekend_nights	stays_in_week_nights	adults	children	babies	\
0	0	0	2	0.0	0	
1	0	0	2	0.0	0	
2	0	1	1	0.0	0	
3	0	1	1	0.0	0	
4	0	2	2	0.0	0	

	meal	country	market_segment	distribution_channel	is_repeated_guest	\
0	BB	PRT	Direct	Direct	0	
1	BB	PRT	Direct	Direct	0	
2	BB	GBR	Direct	Direct	0	
3	BB	GBR	Corporate	Corporate	0	
4	BB	GBR	Online TA	TA/TO	0	

	previous_cancellations	previous_bookings_not_canceled	reserved_room_type	\
0	0	0	C	
1	0	0	C	
2	0	0	A	
3	0	0	A	
4	0	0	A	

	assigned_room_type	booking_changes	deposit_type	agent	company	\
0	C	3	No Deposit	NaN	NaN	
1	C	4	No Deposit	NaN	NaN	
2	C	0	No Deposit	NaN	NaN	
3	A	0	No Deposit	304.0	NaN	
4	A	0	No Deposit	240.0	NaN	

	days_in_waiting_list	customer_type	adr	required_car_parking_spaces	\
0	0	Transient	0.0	0	
1	0	Transient	0.0	0	
2	0	Transient	75.0	0	
3	0	Transient	75.0	0	
4	0	Transient	98.0	0	

	total_of_special_requests	reservation_status	reservation_status_date
0	0	Check-Out	2015-07-01
1	0	Check-Out	2015-07-01
2	0	Check-Out	2015-07-02
3	0	Check-Out	2015-07-02
4	1	Check-Out	2015-07-03

```
[5]: df.shape
```

[5]: (119390, 32)

[6]: df.describe()

```
[6]:
```

	is_canceled	lead_time	arrival_date_year	\
count	119390.000000	119390.000000	119390.000000	
mean	0.370416	104.011416	2016.156554	
std	0.482918	106.863097	0.707476	
min	0.000000	0.000000	2015.000000	
25%	0.000000	18.000000	2016.000000	
50%	0.000000	69.000000	2016.000000	
75%	1.000000	160.000000	2017.000000	
max	1.000000	737.000000	2017.000000	

	arrival_date_week_number	arrival_date_day_of_month	\
count	119390.000000	119390.000000	
mean	27.165173	15.798241	
std	13.605138	8.780829	
min	1.000000	1.000000	
25%	16.000000	8.000000	
50%	28.000000	16.000000	
75%	38.000000	23.000000	
max	53.000000	31.000000	

	stays_in_weekend_nights	stays_in_week_nights	adults	\
count	119390.000000	119390.000000	119390.000000	
mean	0.927599	2.500302	1.856403	
std	0.998613	1.908286	0.579261	
min	0.000000	0.000000	0.000000	
25%	0.000000	1.000000	2.000000	
50%	1.000000	2.000000	2.000000	
75%	2.000000	3.000000	2.000000	
max	19.000000	50.000000	55.000000	

	children	babies	is_repeated_guest	\
count	119386.000000	119390.000000	119390.000000	
mean	0.103890	0.007949	0.031912	
std	0.398561	0.097436	0.175767	
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	10.000000	10.000000	1.000000	

	previous_cancellations	previous_bookings_not_canceled	\
count	119390.000000	119390.000000	
mean	0.087118	0.137097	

std	0.844336	1.497437
min	0.000000	0.000000
25%	0.000000	0.000000
50%	0.000000	0.000000
75%	0.000000	0.000000
max	26.000000	72.000000

	booking_changes	agent	company	days_in_waiting_list \
count	119390.000000	103050.000000	6797.000000	119390.000000
mean	0.221124	86.693382	189.266735	2.321149
std	0.652306	110.774548	131.655015	17.594721
min	0.000000	1.000000	6.000000	0.000000
25%	0.000000	9.000000	62.000000	0.000000
50%	0.000000	14.000000	179.000000	0.000000
75%	0.000000	229.000000	270.000000	0.000000
max	21.000000	535.000000	543.000000	391.000000

	adr	required_car_parking_spaces	total_of_special_requests
count	119390.000000	119390.000000	119390.000000
mean	101.831122	0.062518	0.571363
std	50.535790	0.245291	0.792798
min	-6.380000	0.000000	0.000000
25%	69.290000	0.000000	0.000000
50%	94.575000	0.000000	0.000000
75%	126.000000	0.000000	1.000000
max	5400.000000	8.000000	5.000000

37 % of the people have cancelled their booking as per the dataset. Avg. lead time is 104 days, that is almost 3.5 months. Each booking has on an average 1.8 adults and 0.1 children. Only 3% of the guests are repeated. Median lead time is 69 days.

## MAJOR OBSERVATIONS:

- 1.Number of bookings made were highest in the month of July and August and lowest in January.
- 2.Bookings were more for the City hotel than the Resort hotel. 3.41.7% of the total bookings were cancelled for City hotel and 21.7% for the Resort hotel.
- 4.Number of days that elapsed between the entering date of the booking and the arrival date is less for the people who cancelled.
- 5.As the hotels are in Portugal Europe, the bookings are mostly with European countries, Highest is Portugal with 48.59k bookings.
- 6.77% of the bookings are made with bed and breakfast.
- 7.Only 3% are repeated guests.

## EXPLORATORY DATA ANALYSIS

```
[7]: 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

```

```

[8]: # dealing with null values
null = pd.DataFrame({'Count of Missing values' : df.isna().sum(), 'Percentage_
↳of missing values' : (df.isna().sum()) / (df.shape[0]) * (100)})
null

```

```

[8]:
Count of Missing values \
hotel                    0
is_canceled              0
lead_time                0
arrival_date_year        0
arrival_date_month       0

```

arrival_date_week_number	0
arrival_date_day_of_month	0
stays_in_weekend_nights	0
stays_in_week_nights	0
adults	0
children	4
babies	0
meal	0
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

Percentage of missing values

hotel	0.000000
is_canceled	0.000000
lead_time	0.000000
arrival_date_year	0.000000
arrival_date_month	0.000000
arrival_date_week_number	0.000000
arrival_date_day_of_month	0.000000
stays_in_weekend_nights	0.000000
stays_in_week_nights	0.000000
adults	0.000000
children	0.003350
babies	0.000000
meal	0.000000
country	0.408744
market_segment	0.000000
distribution_channel	0.000000
is_repeated_guest	0.000000
previous_cancellations	0.000000

previous_bookings_not_canceled	0.000000
reserved_room_type	0.000000
assigned_room_type	0.000000
booking_changes	0.000000
deposit_type	0.000000
agent	13.686238
company	94.306893
days_in_waiting_list	0.000000
customer_type	0.000000
adr	0.000000
required_car_parking_spaces	0.000000
total_of_special_requests	0.000000
reservation_status	0.000000
reservation_status_date	0.000000

There are 32 columns, 12 were Categorical and 20 Numerical There are 4 columns with the missing values namely- country, agent, company, children 'company' column has maximum null values which is 94

```
[9]: hotel = df.drop(columns=['company'])
      hotel
```

```
[9]:
```

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

	arrival_date_month	arrival_date_week_number	\
0	July	27	
1	July	27	
2	July	27	
3	July	27	
4	July	27	
...	...	...	
119385	August	35	
119386	August	35	
119387	August	35	
119388	August	35	
119389	August	35	



	arrival_date_day_of_month	stays_in_weekend_nights	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
...	...	...	
119385	30	2	
119386	31	2	
119387	31	2	
119388	31	2	
119389	29	2	

	stays_in_week_nights	adults	children	babies	meal	country	\
0	0	2	0.0	0	BB	PRT	
1	0	2	0.0	0	BB	PRT	
2	1	1	0.0	0	BB	GBR	
3	1	1	0.0	0	BB	GBR	
4	2	2	0.0	0	BB	GBR	
...	...	...	...	...	...	...	
119385	5	2	0.0	0	BB	BEL	
119386	5	3	0.0	0	BB	FRA	
119387	5	2	0.0	0	BB	DEU	
119388	5	2	0.0	0	BB	GBR	
119389	7	2	0.0	0	HB	DEU	

	market_segment	distribution_channel	is_repeated_guest	\
0	Direct	Direct	0	
1	Direct	Direct	0	
2	Direct	Direct	0	
3	Corporate	Corporate	0	
4	Online TA	TA/TO	0	
...	...	...	...	
119385	Offline TA/TO	TA/TO	0	
119386	Online TA	TA/TO	0	
119387	Online TA	TA/TO	0	
119388	Online TA	TA/TO	0	
119389	Online TA	TA/TO	0	

	previous_cancellations	previous_bookings_not_canceled	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
...	...	...	

119385	0	0
119386	0	0
119387	0	0
119388	0	0
119389	0	0

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
0	C	C	3	No Deposit	
1	C	C	4	No Deposit	
2	A	C	0	No Deposit	
3	A	A	0	No Deposit	
4	A	A	0	No Deposit	
...	...	...	...	...	
119385	A	A	0	No Deposit	
119386	E	E	0	No Deposit	
119387	D	D	0	No Deposit	
119388	A	A	0	No Deposit	
119389	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
0	NaN	0	Transient	0.00	
1	NaN	0	Transient	0.00	
2	NaN	0	Transient	75.00	
3	304.0	0	Transient	75.00	
4	240.0	0	Transient	98.00	
...	...	...	...	...	
119385	394.0	0	Transient	96.14	
119386	9.0	0	Transient	225.43	
119387	9.0	0	Transient	157.71	
119388	89.0	0	Transient	104.40	
119389	9.0	0	Transient	151.20	

	required_car_parking_spaces	total_of_special_requests	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	1	
...	...	...	
119385	0	0	
119386	0	2	
119387	0	4	
119388	0	0	
119389	0	2	

	reservation_status	reservation_status_date
0	Check-Out	2015-07-01

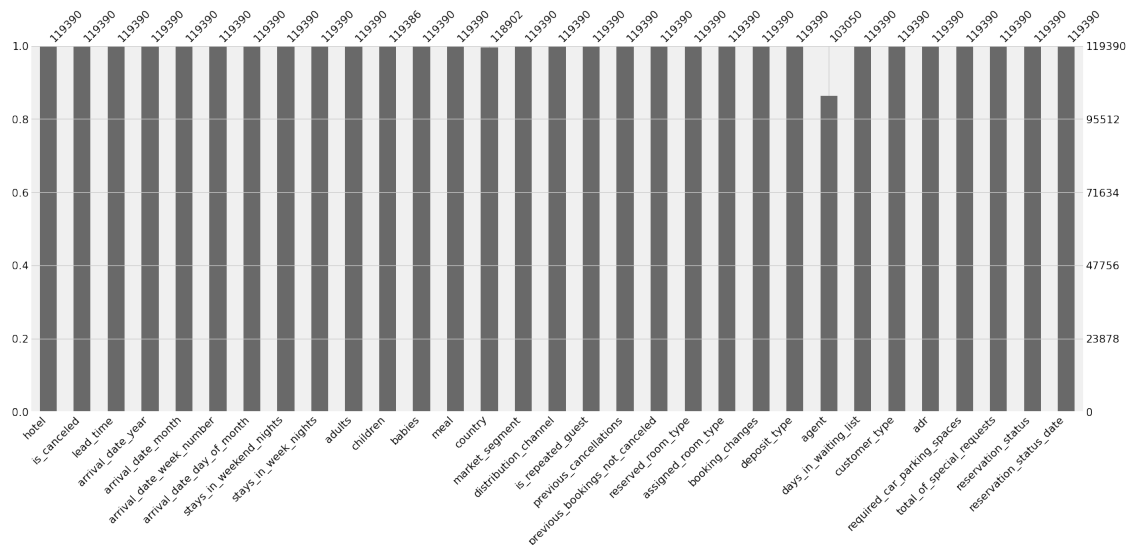
```

1          Check-Out          2015-07-01
2          Check-Out          2015-07-02
3          Check-Out          2015-07-02
4          Check-Out          2015-07-03
...
119385     Check-Out          2017-09-06
119386     Check-Out          2017-09-07
119387     Check-Out          2017-09-07
119388     Check-Out          2017-09-07
119389     Check-Out          2017-09-07

```

```
[119390 rows x 31 columns]
```

```
[10]: #Lets use Missingno library which offers a fair visualization of the
      ↪ distribution of NaN values.
      msno.bar(hotel)
      plt.show()
```



We have almost 120,000 observations, its kind of difficult to make any observation regarding the columns containing NaN values. So, we shall check the distribution of these columns individually.

```
[11]: hotel['children'].value_counts()
```

```

[11]: 0.0      110796
      1.0      4861
      2.0      3652
      3.0        76
      10.0         1
      Name: children, dtype: int64

```

```
[12]: hotel['children'].fillna(0,inplace=True) #In order to deal with the missing
      ↳ information in children's column, we fill it with 0 as we see maximum
      ↳ travellers had 0 children
```

```
[13]: hotel['country'].value_counts()
```

```
[13]: PRT      48590
      GBR      12129
      FRA      10415
      ESP       8568
      DEU       7287
      ...
      LCA         1
      MRT         1
      NCL         1
      SMR         1
      BDI         1
      Name: country, Length: 177, dtype: int64
```

```
[14]: hotel['country'].fillna(hotel['country'].mode()[0], inplace=True) # Since, only
      ↳ 0.4% rows are missing from 'country' column we shall replace it using its
      ↳ mode value
```

```
[15]: hotel['agent'].value_counts()
```

```
[15]: 9.0      31961
      240.0    13922
      1.0      7191
      14.0     3640
      7.0      3539
      ...
      213.0     1
      433.0     1
      197.0     1
      367.0     1
      337.0     1
      Name: agent, Length: 333, dtype: int64
```

```
[16]: hotel['agent'].fillna(0,inplace=True) # For the sake of simplicity, we shall
      ↳ replace the 13% Nan values in column agent with '0'
```

```
[17]: #Rechecking if the null values are handled properly
      missing = pd.DataFrame({'Count of Missing values' : hotel.isna().sum()})
      missing
```

```
[17]:                                     Count of Missing values
      hotel                                                         0
```

```

is_canceled          0
lead_time            0
arrival_date_year    0
arrival_date_month   0
arrival_date_week_number 0
arrival_date_day_of_month 0
stays_in_weekend_nights 0
stays_in_week_nights 0
adults               0
children             0
babies               0
meal                 0
country              0
market_segment       0
distribution_channel  0
is_repeated_guest    0
previous_cancellations 0
previous_bookings_not_canceled 0
reserved_room_type   0
assigned_room_type    0
booking_changes       0
deposit_type          0
agent                0
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

```

```

[18]: # There are a few rows where number of adults is zero, Hence, trying to remove
      ↪ such rows
      filter = (hotel.children == 0) & (hotel.adults == 0) & (hotel.babies == 0)
      hotel[filter]

```

```

[18]:
      hotel  is_canceled  lead_time  arrival_date_year  \
2224  Resort Hotel      0         1         2015
2409  Resort Hotel      0         0         2015
3181  Resort Hotel      0        36         2015
3684  Resort Hotel      0       165         2015
3708  Resort Hotel      0       165         2015
...
115029  City Hotel      0       107         2017
115091  City Hotel      0         1         2017
116251  City Hotel      0        44         2017
116534  City Hotel      0         2         2017

```

117087	City Hotel	0	170	2017
--------	------------	---	-----	------

	arrival_date_month	arrival_date_week_number	\
2224	October	41	
2409	October	42	
3181	November	47	
3684	December	53	
3708	December	53	
...	...	...	
115029	June	26	
115091	June	26	
116251	July	28	
116534	July	28	
117087	July	30	

	arrival_date_day_of_month	stays_in_weekend_nights	\
2224	6	0	
2409	12	0	
3181	20	1	
3684	30	1	
3708	30	2	
...	...	...	
115029	27	0	
115091	30	0	
116251	15	1	
116534	15	2	
117087	27	0	

	stays_in_week_nights	adults	children	babies	meal	country	\
2224	3	0	0.0	0	SC	PRT	
2409	0	0	0.0	0	SC	PRT	
3181	2	0	0.0	0	SC	ESP	
3684	4	0	0.0	0	SC	PRT	
3708	4	0	0.0	0	SC	PRT	
...	...	...	...	...	...	...	
115029	3	0	0.0	0	BB	CHE	
115091	1	0	0.0	0	SC	PRT	
116251	1	0	0.0	0	SC	SWE	
116534	5	0	0.0	0	SC	RUS	
117087	2	0	0.0	0	BB	BRA	

	market_segment	distribution_channel	is_repeated_guest	\
2224	Corporate	Corporate	0	
2409	Corporate	Corporate	0	
3181	Groups	TA/TO	0	
3684	Groups	TA/TO	0	
3708	Groups	TA/TO	0	

...	...	...	...
115029	Online TA	TA/TO	0
115091	Complementary	Direct	0
116251	Online TA	TA/TO	0
116534	Online TA	TA/TO	0
117087	Offline TA/TO	TA/TO	0

	previous_cancellations	previous_bookings_not_canceled	\
2224	0	0	
2409	0	0	
3181	0	0	
3684	0	0	
3708	0	0	

...	...	...
115029	0	0
115091	0	0
116251	0	0
116534	0	0
117087	0	0

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
2224	A	I	1	No Deposit	
2409	A	I	0	No Deposit	
3181	A	C	0	No Deposit	
3684	A	A	1	No Deposit	
3708	A	C	1	No Deposit	
...	...	...	...	...	
115029	A	A	1	No Deposit	
115091	E	K	0	No Deposit	
116251	A	K	2	No Deposit	
116534	A	K	1	No Deposit	
117087	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
2224	0.0	0	Transient-Party	0.00	
2409	0.0	0	Transient	0.00	
3181	38.0	0	Transient-Party	0.00	
3684	308.0	122	Transient-Party	0.00	
3708	308.0	122	Transient-Party	0.00	
...	...	...	...	...	
115029	7.0	0	Transient	100.80	
115091	0.0	0	Transient	0.00	
116251	425.0	0	Transient	73.80	
116534	9.0	0	Transient-Party	22.86	
117087	52.0	0	Transient	0.00	

	required_car_parking_spaces	total_of_special_requests	\
--	-----------------------------	---------------------------	---

2224	0	0
2409	0	0
3181	0	0
3684	0	0
3708	0	0
...	...	...
115029	0	0
115091	1	1
116251	0	0
116534	0	1
117087	0	0

	reservation_status	reservation_status_date
2224	Check-Out	2015-10-06
2409	Check-Out	2015-10-12
3181	Check-Out	2015-11-23
3684	Check-Out	2016-01-04
3708	Check-Out	2016-01-05
...	...	...
115029	Check-Out	2017-06-30
115091	Check-Out	2017-07-01
116251	Check-Out	2017-07-17
116534	Check-Out	2017-07-22
117087	Check-Out	2017-07-29

[180 rows x 31 columns]

```
[19]: #Removing these rows with 0 adults, 0 children and babies
hotel = hotel[~filter]
hotel
```

```
[19]:
```

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

	arrival_date_month	arrival_date_week_number	\
0	July	27	
1	July	27	



2	July	27
3	July	27
4	July	27
...	...	...
119385	August	35
119386	August	35
119387	August	35
119388	August	35
119389	August	35

	arrival_date_day_of_month	stays_in_weekend_nights	\
0	1	0	
1	1	0	
2	1	0	
3	1	0	
4	1	0	
...	...	...	
119385	30	2	
119386	31	2	
119387	31	2	
119388	31	2	
119389	29	2	

	stays_in_week_nights	adults	children	babies	meal	country	\
0	0	2	0.0	0	BB	PRT	
1	0	2	0.0	0	BB	PRT	
2	1	1	0.0	0	BB	GBR	
3	1	1	0.0	0	BB	GBR	
4	2	2	0.0	0	BB	GBR	
...	...	...	...	...	...	...	
119385	5	2	0.0	0	BB	BEL	
119386	5	3	0.0	0	BB	FRA	
119387	5	2	0.0	0	BB	DEU	
119388	5	2	0.0	0	BB	GBR	
119389	7	2	0.0	0	HB	DEU	

	market_segment	distribution_channel	is_repeated_guest	\
0	Direct	Direct	0	
1	Direct	Direct	0	
2	Direct	Direct	0	
3	Corporate	Corporate	0	
4	Online TA	TA/TO	0	
...	...	...	...	
119385	Offline TA/TO	TA/TO	0	
119386	Online TA	TA/TO	0	
119387	Online TA	TA/TO	0	
119388	Online TA	TA/TO	0	

119389	Online TA	TA/TO	0
--------	-----------	-------	---

	previous_cancellations	previous_bookings_not_canceled	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	0	
...	...	...	
119385	0	0	
119386	0	0	
119387	0	0	
119388	0	0	
119389	0	0	

	reserved_room_type	assigned_room_type	booking_changes	deposit_type	\
0	C	C	3	No Deposit	
1	C	C	4	No Deposit	
2	A	C	0	No Deposit	
3	A	A	0	No Deposit	
4	A	A	0	No Deposit	
...	...	...	...	...	
119385	A	A	0	No Deposit	
119386	E	E	0	No Deposit	
119387	D	D	0	No Deposit	
119388	A	A	0	No Deposit	
119389	A	A	0	No Deposit	

	agent	days_in_waiting_list	customer_type	adr	\
0	0.0	0	Transient	0.00	
1	0.0	0	Transient	0.00	
2	0.0	0	Transient	75.00	
3	304.0	0	Transient	75.00	
4	240.0	0	Transient	98.00	
...	...	...	...	...	
119385	394.0	0	Transient	96.14	
119386	9.0	0	Transient	225.43	
119387	9.0	0	Transient	157.71	
119388	89.0	0	Transient	104.40	
119389	9.0	0	Transient	151.20	

	required_car_parking_spaces	total_of_special_requests	\
0	0	0	
1	0	0	
2	0	0	
3	0	0	
4	0	1	

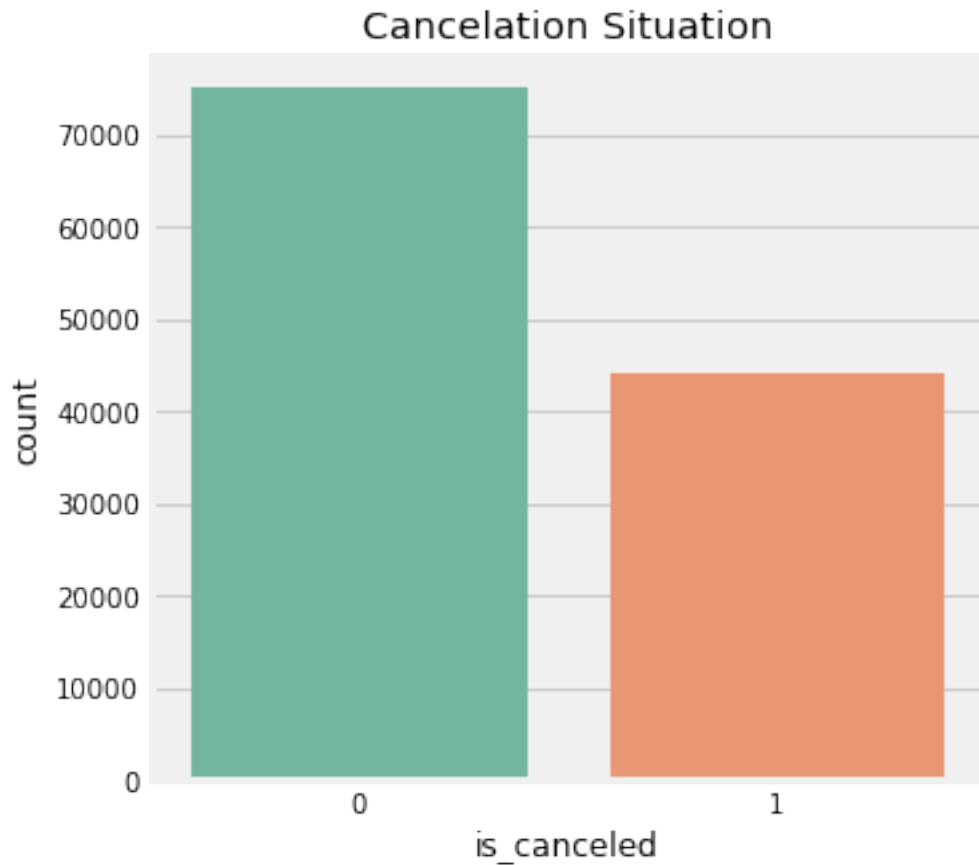
...	...	...
119385	0	0
119386	0	2
119387	0	4
119388	0	0
119389	0	2

	reservation_status	reservation_status_date
0	Check-Out	2015-07-01
1	Check-Out	2015-07-01
2	Check-Out	2015-07-02
3	Check-Out	2015-07-02
4	Check-Out	2015-07-03
...	...	...
119385	Check-Out	2017-09-06
119386	Check-Out	2017-09-07
119387	Check-Out	2017-09-07
119388	Check-Out	2017-09-07
119389	Check-Out	2017-09-07

[119210 rows x 31 columns]

After dealing with the null values and dropping few unwanted rows the new shape of our dataset is (119210,31)

```
[20]: #Checking the cancelation status
plt.figure(figsize=(5,5))
sns.countplot(x='is_canceled' , data = hotel, palette = 'Set2')
plt.title('Cancelation Situation')
plt.show()
```



```
[21]: hotel['is_canceled'].value_counts()/hotel.shape[0]*100
```

```
[21]: 0    62.923412
      1    37.076588
      Name: is_canceled, dtype: float64
```

```
[22]: ## Converting Datatype: Children are listed as float datatype but in reality
      → its interger, so needs to be changed
hotel['is_canceled'] = hotel['is_canceled'].astype('object')
hotel['children'] = hotel['children'].astype('int64')
hotel['agent'] = hotel['agent'].astype('int64')
hotel['country'] = hotel['country'].astype('str')
hotel['reservation_status_date'] = hotel['reservation_status_date'].
      → astype('datetime64')
# looking at the reservation_status_date we can see it doesnt have correct
→ Dtype, hence we need to change it to datetime 64
hotel.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
```

Data columns (total 31 columns):

#	Column	Non-Null Count	Dtype
0	hotel	119210 non-null	object
1	is_canceled	119210 non-null	object
2	lead_time	119210 non-null	int64
3	arrival_date_year	119210 non-null	int64
4	arrival_date_month	119210 non-null	object
5	arrival_date_week_number	119210 non-null	int64
6	arrival_date_day_of_month	119210 non-null	int64
7	stays_in_weekend_nights	119210 non-null	int64
8	stays_in_week_nights	119210 non-null	int64
9	adults	119210 non-null	int64
10	children	119210 non-null	int64
11	babies	119210 non-null	int64
12	meal	119210 non-null	object
13	country	119210 non-null	object
14	market_segment	119210 non-null	object
15	distribution_channel	119210 non-null	object
16	is_repeated_guest	119210 non-null	int64
17	previous_cancellations	119210 non-null	int64
18	previous_bookings_not_canceled	119210 non-null	int64
19	reserved_room_type	119210 non-null	object
20	assigned_room_type	119210 non-null	object
21	booking_changes	119210 non-null	int64
22	deposit_type	119210 non-null	object
23	agent	119210 non-null	int64
24	days_in_waiting_list	119210 non-null	int64
25	customer_type	119210 non-null	object
26	adr	119210 non-null	float64
27	required_car_parking_spaces	119210 non-null	int64
28	total_of_special_requests	119210 non-null	int64
29	reservation_status	119210 non-null	object
30	reservation_status_date	119210 non-null	datetime64[ns]

dtypes: datetime64[ns](1), float64(1), int64(17), object(12)

memory usage: 34.1+ MB

```
[23]: hotel = hotel.drop(['reservation_status_date'], axis = 1)
```

```
[24]: from sklearn import preprocessing

label_encoder = preprocessing.LabelEncoder()

# Encode labels in all the categorical columns
hotel['hotel'] = label_encoder.fit_transform(hotel['hotel'])
hotel['arrival_date_month'] = label_encoder.
    ↳fit_transform(hotel['arrival_date_month'])
```

```

hotel['meal']= label_encoder.fit_transform(hotel['meal'])
hotel['country']= label_encoder.fit_transform(hotel['country'])
hotel['market_segment']= label_encoder.fit_transform(hotel['market_segment'])
hotel['distribution_channel']= label_encoder.
    ↳fit_transform(hotel['distribution_channel'])
hotel['is_repeated_guest']= label_encoder.
    ↳fit_transform(hotel['is_repeated_guest'])
hotel['reserved_room_type']= label_encoder.
    ↳fit_transform(hotel['reserved_room_type'])
hotel['assigned_room_type']= label_encoder.fit_transform(hotel['deposit_type'])
hotel['deposit_type']= label_encoder.fit_transform(hotel['is_repeated_guest'])
hotel['agent']= label_encoder.fit_transform(hotel['agent'])
hotel['customer_type']= label_encoder.fit_transform(hotel['customer_type'])
hotel['reservation_status']= label_encoder.
    ↳fit_transform(hotel['reservation_status'])

```

[25]: hotel.info()

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 30 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   hotel                                119210 non-null  int64
1   is_canceled                          119210 non-null  object
2   lead_time                            119210 non-null  int64
3   arrival_date_year                    119210 non-null  int64
4   arrival_date_month                  119210 non-null  int64
5   arrival_date_week_number             119210 non-null  int64
6   arrival_date_day_of_month            119210 non-null  int64
7   stays_in_weekend_nights              119210 non-null  int64
8   stays_in_week_nights                 119210 non-null  int64
9   adults                               119210 non-null  int64
10  children                             119210 non-null  int64
11  babies                               119210 non-null  int64
12  meal                                 119210 non-null  int64
13  country                              119210 non-null  int64
14  market_segment                       119210 non-null  int64
15  distribution_channel                  119210 non-null  int64
16  is_repeated_guest                     119210 non-null  int64
17  previous_cancellations                 119210 non-null  int64
18  previous_bookings_not_canceled         119210 non-null  int64
19  reserved_room_type                    119210 non-null  int64
20  assigned_room_type                    119210 non-null  int64
21  booking_changes                       119210 non-null  int64
22  deposit_type                          119210 non-null  int64
23  agent                                119210 non-null  int64

```

```

24  days_in_waiting_list      119210 non-null  int64
25  customer_type             119210 non-null  int64
26  adr                       119210 non-null  float64
27  required_car_parking_spaces 119210 non-null  int64
28  total_of_special_requests  119210 non-null  int64
29  reservation_status        119210 non-null  int64
dtypes: float64(1), int64(28), object(1)
memory usage: 33.2+ MB

```

```
[26]: Hotel_data = hotel.drop(['reservation_status', 'stays_in_week_nights',
    ↪ 'stays_in_weekend_nights'], axis=1)
```

```
[27]: Hotel_data.info()
```

```

<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 27 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   hotel                                119210 non-null  int64
 1   is_canceled                          119210 non-null  object
 2   lead_time                            119210 non-null  int64
 3   arrival_date_year                    119210 non-null  int64
 4   arrival_date_month                   119210 non-null  int64
 5   arrival_date_week_number             119210 non-null  int64
 6   arrival_date_day_of_month            119210 non-null  int64
 7   adults                               119210 non-null  int64
 8   children                             119210 non-null  int64
 9   babies                               119210 non-null  int64
10   meal                                 119210 non-null  int64
11   country                              119210 non-null  int64
12   market_segment                       119210 non-null  int64
13   distribution_channel                 119210 non-null  int64
14   is_repeated_guest                    119210 non-null  int64
15   previous_cancellations               119210 non-null  int64
16   previous_bookings_not_canceled       119210 non-null  int64
17   reserved_room_type                   119210 non-null  int64
18   assigned_room_type                   119210 non-null  int64
19   booking_changes                      119210 non-null  int64
20   deposit_type                         119210 non-null  int64
21   agent                                119210 non-null  int64
22   days_in_waiting_list                 119210 non-null  int64
23   customer_type                        119210 non-null  int64
24   adr                                  119210 non-null  float64
25   required_car_parking_spaces          119210 non-null  int64
26   total_of_special_requests            119210 non-null  int64
dtypes: float64(1), int64(25), object(1)
memory usage: 30.5+ MB

```

```
[28]: hotel_model = Hotel_data.drop(['babies', 'deposit_type'], axis = 1)
      hotel_model.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 119210 entries, 0 to 119389
Data columns (total 25 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   hotel                                119210 non-null  int64
 1   is_canceled                          119210 non-null  object
 2   lead_time                            119210 non-null  int64
 3   arrival_date_year                    119210 non-null  int64
 4   arrival_date_month                   119210 non-null  int64
 5   arrival_date_week_number             119210 non-null  int64
 6   arrival_date_day_of_month            119210 non-null  int64
 7   adults                               119210 non-null  int64
 8   children                             119210 non-null  int64
 9   meal                                 119210 non-null  int64
10   country                             119210 non-null  int64
11   market_segment                       119210 non-null  int64
12   distribution_channel                 119210 non-null  int64
13   is_repeated_guest                    119210 non-null  int64
14   previous_cancellations               119210 non-null  int64
15   previous_bookings_not_canceled       119210 non-null  int64
16   reserved_room_type                   119210 non-null  int64
17   assigned_room_type                   119210 non-null  int64
18   booking_changes                      119210 non-null  int64
19   agent                               119210 non-null  int64
20   days_in_waiting_list                 119210 non-null  int64
21   customer_type                        119210 non-null  int64
22   adr                                  119210 non-null  float64
23   required_car_parking_spaces          119210 non-null  int64
24   total_of_special_requests            119210 non-null  int64
dtypes: float64(1), int64(23), object(1)
memory usage: 28.6+ MB
```

```
[29]: y = hotel.iloc[:,1]
      y=y.astype('int')
      X = pd.concat([hotel.iloc[:,0],hotel.iloc[:,2:24]], axis=1)
```

```
[30]: # Splitting the X and Y into Training set and Testing set
      from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
      ↪random_state = 0)
```



```
[31]: standardScalerX = StandardScaler()
X_train = standardScalerX.fit_transform(X_train)
X_test = standardScalerX.fit_transform(X_test)
```

## Model Building

```
[32]: from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import VotingClassifier

from sklearn.metrics import accuracy_score
from sklearn.model_selection import cross_val_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

```
[33]: # Random Forest Model Building

rfc_model = RandomForestClassifier (min_samples_leaf = 6, min_samples_split = 6,
    ↳ n_estimators = 100)

#fitting and prediction of model
rfc_model.fit(X_train, y_train)
predict_rfc = rfc_model.predict(X_test)
```

```
[34]: # Decision Tree Model Building

dtc_model = DecisionTreeClassifier (criterion = 'gini', min_samples_leaf = 4,
    ↳ min_samples_split = 8, max_features = 'auto')

#fitting and prediction of model
dtc_model.fit(X_train, y_train)
predict_dtc = dtc_model.predict(X_test)
```

```
[35]: # Extreme Gradient Boosting Model Building

xgb_model = XGBClassifier (criterion = 'gini', learning_rate = 0.01, max_depth = 5,
    ↳ n_estimators = 100, objective = 'binary:logistic', subsample = 1.000)

#fitting and prediction of model
xgb_model.fit(X_train, y_train)
predict_xgb = xgb_model.predict(X_test)
```

```
[20:04:16] WARNING: ../src/learner.cc:573:
Parameters: { "criterion" } might not be used.
```

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

[20:04:16] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval\_metric if you'd like to restore the old behavior.

```
[36]: # Extra Trees Classifier Model Building

etc_model = ExtraTreesClassifier ( min_samples_leaf = 7, min_samples_split = 2,
    ↪n_estimators = 200 )

#fitting and prediction of model
etc_model.fit(X_train, y_train)
predict_etc = etc_model.predict(X_test)
```

## Classification Report

```
[37]: print("RF", classification_report(y_test, predict_rfc))
print("DTC", classification_report(y_test, predict_dtc))
print("XGB", classification_report(y_test, predict_xgb))
print("ETC", classification_report(y_test, predict_etc))
```

RF		precision	recall	f1-score	support
	0	0.77	0.96	0.86	14919
	1	0.89	0.52	0.66	8923
	accuracy			0.80	23842
	macro avg	0.83	0.74	0.76	23842
	weighted avg	0.81	0.80	0.78	23842

DTC		precision	recall	f1-score	support
	0	0.80	0.90	0.85	14919
	1	0.79	0.61	0.69	8923
	accuracy			0.79	23842
	macro avg	0.79	0.76	0.77	23842
	weighted avg	0.79	0.79	0.79	23842

XGB		precision	recall	f1-score	support
-----	--	-----------	--------	----------	---------

	0	0.73	1.00	0.84	14919
	1	0.99	0.37	0.54	8923
accuracy				0.76	23842
macro avg		0.86	0.69	0.69	23842
weighted avg		0.83	0.76	0.73	23842

ETC		precision	recall	f1-score	support
	0	0.77	0.98	0.87	14919
	1	0.95	0.52	0.67	8923
accuracy				0.81	23842
macro avg		0.86	0.75	0.77	23842
weighted avg		0.84	0.81	0.79	23842

## Confusion Matrix

```
[38]: RF_matrix = confusion_matrix(y_test, predict_rfc)
      DTC_matrix = confusion_matrix(y_test, predict_dtc)
      XGB_matrix = confusion_matrix(y_test, predict_xgb)
      ETC_matrix = confusion_matrix(y_test, predict_etc)

fig, ax = plt.subplots(1, 2, figsize=(15, 8))

sns.heatmap(RF_matrix,annot=True, fmt="d", cbar=False, cmap="Pastel2", ax =_
    ↪ax[0]).set_ylim([0,2])
ax[0].set_title("Random Forest", weight='bold')
ax[0].set_xlabel('Predicted Labels')
ax[0].set_ylabel('Actual Labels')

sns.heatmap(DTC_matrix,annot=True, fmt="d" ,cbar=False, cmap="tab20", ax =_
    ↪ax[1]).set_ylim([0,2])
ax[1].set_title("Decision Tree", weight='bold')
ax[1].set_xlabel('Predicted Labels')
ax[1].set_ylabel('Actual Labels')

#####

fig, axe = plt.subplots(1, 2, figsize=(15, 8))

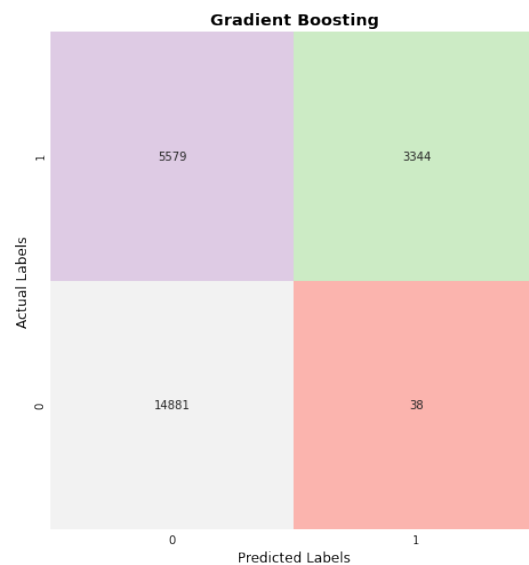
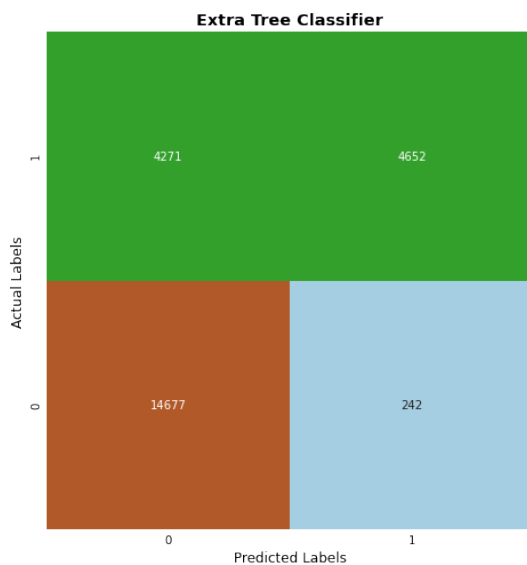
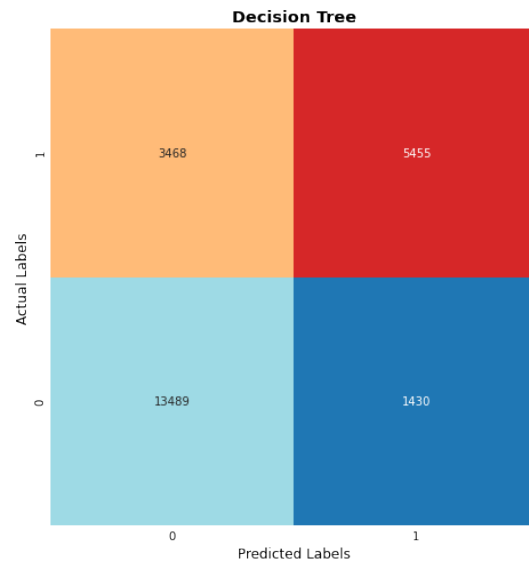
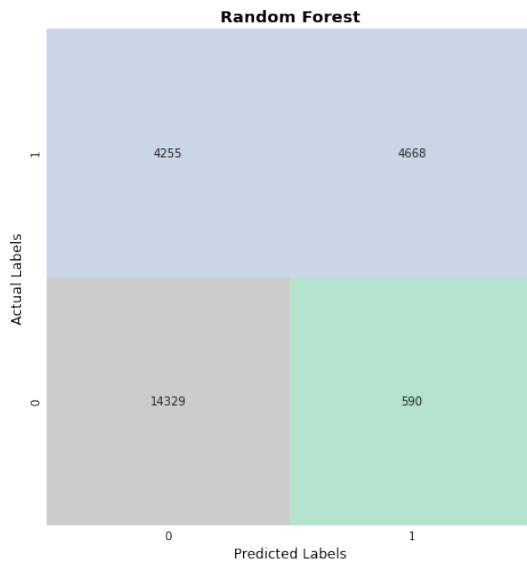
sns.heatmap(XGB_matrix,annot=True, fmt="d", cbar=False, cmap="Pastel1", ax =_
    ↪axe[1]).set_ylim([0,2])
axe[1].set_title("Gradient Boosting", weight='bold')
axe[1].set_xlabel('Predicted Labels')
axe[1].set_ylabel('Actual Labels')
```

```

sns.heatmap(ETC_matrix,annot=True,fmt="d", cbar=False, cmap="Paired", ax =_
→ax[0]).set_ylim([0,2])
ax[0].set_title("Extra Tree Classifier", weight='bold')
ax[0].set_xlabel('Predicted Labels')
ax[0].set_ylabel('Actual Labels')

```

[38]: Text(68.9, 0.5, 'Actual Labels')



SMOTE OVERSAMPLING

```
[39]: pip install -U imbalanced-learn
```

```
Requirement already up-to-date: imbalanced-learn in
/opt/conda/lib/python3.7/site-packages (0.8.0)
Requirement already satisfied, skipping upgrade: numpy>=1.13.3 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn) (1.18.4)
Requirement already satisfied, skipping upgrade: joblib>=0.11 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn) (0.15.1)
Requirement already satisfied, skipping upgrade: scipy>=0.19.1 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn) (1.4.1)
Requirement already satisfied, skipping upgrade: scikit-learn>=0.24 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn) (0.24.2)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from scikit-learn>=0.24->imbalanced-
learn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[40]: pip install imblearn
```

```
Requirement already satisfied: imblearn in /opt/conda/lib/python3.7/site-
packages (0.0)
Requirement already satisfied: imbalanced-learn in
/opt/conda/lib/python3.7/site-packages (from imblearn) (0.8.0)
Requirement already satisfied: scikit-learn>=0.24 in
/opt/conda/lib/python3.7/site-packages (from imbalanced-learn->imblearn)
(0.24.2)
Requirement already satisfied: numpy>=1.13.3 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (1.18.4)
Requirement already satisfied: joblib>=0.11 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (0.15.1)
Requirement already satisfied: scipy>=0.19.1 in /opt/conda/lib/python3.7/site-
packages (from imbalanced-learn->imblearn) (1.4.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from scikit-learn>=0.24->imbalanced-
learn->imblearn) (2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[41]: from collections import Counter
```

```
[42]: pip install delayed
```

```
Requirement already satisfied: delayed in /opt/conda/lib/python3.7/site-packages
(0.11.0b1)
Requirement already satisfied: hiredis in /opt/conda/lib/python3.7/site-packages
(from delayed) (2.0.0)
Requirement already satisfied: redis in /opt/conda/lib/python3.7/site-packages
(from delayed) (3.5.3)
Note: you may need to restart the kernel to use updated packages.
```

```
[43]: from imblearn.over_sampling import SMOTE
```

```
[44]: sm = SMOTE(random_state = 10)
X_sm, y_sm = sm.fit_resample(X, y)
```

```
[45]: X_sm.shape
```

```
[45]: (150022, 23)
```

```
[46]: y_sm.shape
```

```
[46]: (150022,)
```

```
[47]: print('After OverSampling, the shape of train_X: {}'.format(X_sm.shape))
print('After OverSampling, the shape of train_y: {} \n'.format(y_sm.shape))

print("After OverSampling, counts of label '1': {}".format(sum(y_sm == 1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_sm == 0)))
```

After OverSampling, the shape of train\_X: (150022, 23)

After OverSampling, the shape of train\_y: (150022,)

After OverSampling, counts of label '1': 75011

After OverSampling, counts of label '0': 75011

```
[48]: X_train_res, X_test_res, y_train_res, y_test_res = train_test_split(X_sm, y_sm,
    ↪test_size=0.25, random_state=27, stratify=None)
```

```
[49]: standardScalerX = StandardScaler()
X_train_res = standardScalerX.fit_transform(X_train_res)
X_test_res = standardScalerX.fit_transform(X_test_res)
```

```
[50]: from sklearn.model_selection import StratifiedKFold
kfold_cv1 = StratifiedKFold( n_splits = 5, random_state = 27, shuffle = True)

for train_index1, test_index1 in kfold_cv1.split(X_sm, y_sm):
    X_train_res, X_test_res = X_sm.iloc[train_index1], X_sm.iloc[test_index1]
    y_train_res, y_test_res = y_sm.iloc[train_index1], y_sm.iloc[test_index1]
```

```
[51]: # Random Forest Model Building

rfc_model1 = RandomForestClassifier (min_samples_leaf = 6, min_samples_split =
    ↪6, n_estimators = 100)

#fitting and prediction of model
rfc_model1.fit(X_train_res, y_train_res)
predict_rfc1 = rfc_model1.predict(X_test_res)
```

```
[52]: # Decision Tree Model Building

dtc_model1 = DecisionTreeClassifier (criterion = 'gini', min_samples_leaf = 4,
    ↳ min_samples_split = 8, max_features = 'auto')

#fitting and prediction of model
dtc_model1.fit(X_train_res, y_train_res)
predict_dtc1 = dtc_model1.predict(X_test_res)
```

```
[53]: # Extreme Gradient Boosting Model Building

xgb_model1 = XGBClassifier (criterion = 'gini', learning_rate = 0.01, max_depth=
    ↳ 5, n_estimators = 100, objective = 'binary:logistic', subsample = 1.000)

#fitting and prediction of model
xgb_model1.fit(X_train_res, y_train_res)
predict_xgb1 = xgb_model1.predict(X_test_res)
```

[20:07:56] WARNING: ../src/learner.cc:573:  
Parameters: { "criterion" } might not be used.

This may not be accurate due to some parameters are only used in language bindings but passed down to XGBoost core. Or some parameters are not used but slip through this verification. Please open an issue if you find above cases.

[20:07:56] WARNING: ../src/learner.cc:1095: Starting in XGBoost 1.3.0, the default evaluation metric used with the objective 'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set eval\_metric if you'd like to restore the old behavior.

```
[54]: # Extra Trees Classifier Model Building

etc_model1 = ExtraTreesClassifier ( min_samples_leaf = 7, min_samples_split =
    ↳ 2, n_estimators = 200 )

#fitting and prediction of model
etc_model1.fit(X_train_res, y_train_res)
predict_etc1 = etc_model1.predict(X_test_res)
```

```
[55]: print("RF1", classification_report(y_test_res, predict_rfc1))
print("DTC1", classification_report(y_test_res, predict_dtc1))
print("XGB1", classification_report(y_test_res, predict_xgb1))
print("ETC1", classification_report(y_test_res, predict_etc1))
```

RF1		precision	recall	f1-score	support
	0	0.87	0.85	0.86	15002
	1	0.86	0.87	0.86	15002
	accuracy			0.86	30004
	macro avg	0.86	0.86	0.86	30004
	weighted avg	0.86	0.86	0.86	30004
DTC1		precision	recall	f1-score	support
	0	0.81	0.82	0.81	15002
	1	0.82	0.81	0.81	15002
	accuracy			0.81	30004
	macro avg	0.81	0.81	0.81	30004
	weighted avg	0.81	0.81	0.81	30004
XGB1		precision	recall	f1-score	support
	0	0.83	0.72	0.77	15002
	1	0.75	0.85	0.80	15002
	accuracy			0.79	30004
	macro avg	0.79	0.79	0.78	30004
	weighted avg	0.79	0.79	0.78	30004
ETC1		precision	recall	f1-score	support
	0	0.82	0.89	0.85	15002
	1	0.88	0.80	0.84	15002
	accuracy			0.84	30004
	macro avg	0.85	0.84	0.84	30004
	weighted avg	0.85	0.84	0.84	30004

[ ]: