Data Preprocessing

Following are Data Preprocessing Steps include in this Notebook:

- Standardization
- Encoding
- Missing Values Imputing
- Disecretization
- Normalization

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

Dataset Download from Here (https://www.kaggle.com/jessemostipak/hotel-booking-demand)

In [2]:
```
data = pd.read_csv("hotel_bookings.csv")
```

In [3]:
```
data.head()
```

Out[3]:

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

5 rows × 32 columns

In [4]:
```
data.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
```

Encoding

In [5]:

```
from sklearn.preprocessing import LabelEncoder , OneHotEncoder
```

In [6]:

```
data['hotel'].value_counts()
```

Out[6]:

```
City Hotel      79330
Resort Hotel    40060
Name: hotel, dtype: int64
```

LABEL ENCODER

In [7]:

```
le = LabelEncoder()
data['hotel'] = le.fit_transform(data['hotel'])
```

In [8]:

```
data['hotel'].value_counts()
```

Out[8]:

```
0    79330
1    40060
Name: hotel, dtype: int64
```

In [9]:

```
le.classes_
```

Out[9]:

```
array(['City Hotel', 'Resort Hotel'], dtype=object)
```

ONE HOT ENCODER

In [10]:

```
data['customer_type'].value_counts()
```

Out[10]:

```
Transient          89613
Transient-Party    25124
Contract            4076
Group                577
Name: customer_type, dtype: int64
```

In [11]:

```
one_hot = OneHotEncoder()
transformed_data = one_hot.fit_transform(data['customer_type'].values.reshape(-1,1)).toarray()
```

In [12]:

```
one_hot.categories_
```

Out[12]:

```
[array(['Contract', 'Group', 'Transient', 'Transient-Party'], dtype=object)]
```

In [13]:

```
transformed_data = pd.DataFrame(transformed_data ,
                                columns = ['Contract', 'Group', 'Transient', 'Transient-Party'])
```

In [14]:

```
transformed_data.head()
```

Out[14]:

	Contract	Group	Transient	Transient-Party
0	0.0	0.0	1.0	0.0
1	0.0	0.0	1.0	0.0
2	0.0	0.0	1.0	0.0
3	0.0	0.0	1.0	0.0
4	0.0	0.0	1.0	0.0

In [15]:

```
transformed_data.iloc[90 , ]
```

Out[15]:

```
Contract           0.0
Group              0.0
Transient          1.0
Transient-Party    0.0
Name: 90, dtype: float64
```

In [16]:

```
data['customer_type'][90]
```

Out[16]:

```
'Transient'
```

Normalization & Standardization

In [17]:

```
# consider only numerical columns

numeric_columns = [c for c in data.columns if data[c].dtype ≠ np.dtype('O')]
```

In [18]:

```
len(numeric_columns) , len(data.columns)
```

Out[18]:

```
(21, 32)
```

In [19]:

```
numeric_columns.remove('company')
numeric_columns.remove('agent')
```

In [20]:

```
temp_data = data[numeric_columns]
```

In [21]:

```
temp_data
```

Out[21]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	a
0	1	0	342	2015	27	1	0	0	
1	1	0	737	2015	27	1	0	0	
2	1	0	7	2015	27	1	0	1	
3	1	0	13	2015	27	1	0	1	
4	1	0	14	2015	27	1	0	2	
...	...	...	...	...	...	...	...	...	
119385	0	0	23	2017	35	30	2	5	
119386	0	0	102	2017	35	31	2	5	
119387	0	0	34	2017	35	31	2	5	
119388	0	0	109	2017	35	31	2	5	
119389	0	0	205	2017	35	29	2	7	

119390 rows × 19 columns

In [22]:

```
from sklearn.preprocessing import StandardScaler , MinMaxScaler
```

Normalization

In [23]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [24]:

```
normalizer = MinMaxScaler()
```

In [25]:

```
temp_data.dropna(axis = 1 , inplace = True)
```

In [26]:

```
normalized_data = normalizer.fit_transform(temp_data)
```

In [27]:

```
pd.DataFrame(normalized_data , columns = temp_data.columns)
```

Out[27]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	
0	1.0	0.0	0.464043	0.0	0.500000	0.000000	0.000000	0.00	0
1	1.0	0.0	1.000000	0.0	0.500000	0.000000	0.000000	0.00	0
2	1.0	0.0	0.009498	0.0	0.500000	0.000000	0.000000	0.02	0
3	1.0	0.0	0.017639	0.0	0.500000	0.000000	0.000000	0.02	0
4	1.0	0.0	0.018996	0.0	0.500000	0.000000	0.000000	0.04	0
...	...	...	...	...	...	...	...	...	
119385	0.0	0.0	0.031208	1.0	0.653846	0.966667	0.105263	0.10	0
119386	0.0	0.0	0.138399	1.0	0.653846	1.000000	0.105263	0.10	0
119387	0.0	0.0	0.046133	1.0	0.653846	1.000000	0.105263	0.10	0
119388	0.0	0.0	0.147897	1.0	0.653846	1.000000	0.105263	0.10	0
119389	0.0	0.0	0.278155	1.0	0.653846	0.933333	0.105263	0.14	0

119390 rows × 18 columns

Standardization

In [28]:

```
standard_scaler = StandardScaler()
```

In [30]:

```
standardized_data = standard_scaler.fit_transform(temp_data)
```

In [31]:

```
pd.DataFrame(standardized_data , columns = temp_data.columns)
```

Out[31]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_night
0	1.407224	-0.76704	2.227051	-1.634768	-0.012141	-1.685297	-0.928890	-1.31024
1	1.407224	-0.76704	5.923385	-1.634768	-0.012141	-1.685297	-0.928890	-1.31024
2	1.407224	-0.76704	-0.907814	-1.634768	-0.012141	-1.685297	-0.928890	-0.78620
3	1.407224	-0.76704	-0.851667	-1.634768	-0.012141	-1.685297	-0.928890	-0.78620
4	1.407224	-0.76704	-0.842309	-1.634768	-0.012141	-1.685297	-0.928890	-0.26217
...	...	...	...	...	...	...	...	.
119385	-0.710619	-0.76704	-0.758089	1.192195	0.575875	1.617366	1.073895	1.30992
119386	-0.710619	-0.76704	-0.018822	1.192195	0.575875	1.731251	1.073895	1.30992
119387	-0.710619	-0.76704	-0.655153	1.192195	0.575875	1.731251	1.073895	1.30992
119388	-0.710619	-0.76704	0.046682	1.192195	0.575875	1.731251	1.073895	1.30992
119389	-0.710619	-0.76704	0.945032	1.192195	0.575875	1.503481	1.073895	2.35798

119390 rows × 18 columns

Handling With Missing Values

In [32]:

```
data.isnull().sum()
```

Out[32]:

```
hotel                                0
is_canceled                          0
lead_time                            0
arrival_date_year                    0
arrival_date_month                   0
arrival_date_week_number             0
arrival_date_day_of_month            0
stays_in_weekend_nights              0
stays_in_week_nights                 0
adults                               0
children                             4
babies                               0
meal                                 0
country                            488
market_segment                       0
distribution_channel                 0
is_repeated_guest                    0
previous_cancellations               0
previous_bookings_not_canceled       0
reserved_room_type                   0
assigned_room_type                   0
booking_changes                      0
deposit_type                         0
agent                            16340
company                         112593
days_in_waiting_list                 0
customer_type                        0
adr                                  0
required_car_parking_spaces          0
total_of_special_requests            0
reservation_status                   0
reservation_status_date              0
dtype: int64
```

In [33]:

```
# here I Will show you imputing values in Null columns only for 'agent' column
```

In [34]:

```
data['agent'].isnull().sum()
```

Out[34]:

```
16340
```

Simple Imputer

In [35]:

```
from sklearn.impute import SimpleImputer
```

In [36]:

```
imputer = SimpleImputer(missing_values=np.nan , strategy='mean')
```

In [37]:

```
agent_col = imputer.fit_transform(data['agent'].values.reshape(-1,1))
```

In [41]:

```
pd.DataFrame(agent_col).isnull().sum()
```

```
0    0
dtype: int64
```

In [42]:

```
data['agent'].isnull().sum()
```

Out[42]:

```
16340
```

Discretization

In [46]:

```
from sklearn.preprocessing import KBinsDiscretizer
```

In [47]:

```
temp_data.head()
```

Out[47]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	adults
0	1	0	342	2015	27	1	0	0	2
1	1	0	737	2015	27	1	0	0	2
2	1	0	7	2015	27	1	0	1	1
3	1	0	13	2015	27	1	0	1	1
4	1	0	14	2015	27	1	0	2	2

Quantile Discretization Transform

In [48]:

```
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='quantile')
new_data = trans.fit_transform(temp_data)
```

In [50]:

```
pd.DataFrame(new_data,columns = temp_data.columns )
```

Out[50]:

_guest	previous_cancellations	previous_bookings_not_canceled	booking_changes	days_in_waiting_list	adr	required_car_parking_spaces	total_of_special_requests
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	3.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	5.0	0.0	1.0
...	...	...	...	...	...	...	...
0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	9.0	0.0	2.0
0.0	0.0	0.0	0.0	0.0	8.0	0.0	2.0
0.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0
0.0	0.0	0.0	0.0	0.0	8.0	0.0	2.0

Uniform Discretization Transform

In [51]:

```python
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='uniform')
new_data = trans.fit_transform(temp_data)

pd.DataFrame(new_data,columns = temp_data.columns )
```

Out[51]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	a
0	9.0	0.0	4.0	0.0	5.0	0.0	0.0	0.0	
1	9.0	0.0	9.0	0.0	5.0	0.0	0.0	0.0	
2	9.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	
3	9.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	
4	9.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	
119385	0.0	0.0	0.0	9.0	6.0	9.0	1.0	1.0	
119386	0.0	0.0	1.0	9.0	6.0	9.0	1.0	1.0	
119387	0.0	0.0	0.0	9.0	6.0	9.0	1.0	1.0	
119388	0.0	0.0	1.0	9.0	6.0	9.0	1.0	1.0	
119389	0.0	0.0	2.0	9.0	6.0	9.0	1.0	1.0	

119390 rows × 18 columns

KMeans Discretization Transform

In [52]:

```python
trans = KBinsDiscretizer(n_bins =10 , encode = 'ordinal' , strategy='kmeans')
new_data = trans.fit_transform(temp_data)

pd.DataFrame(new_data,columns = temp_data.columns )
```

Out[52]:

	hotel	is_canceled	lead_time	arrival_date_year	arrival_date_week_number	arrival_date_day_of_month	stays_in_weekend_nights	stays_in_week_nights	a
0	1.0	0.0	6.0	0.0	4.0	0.0	0.0	0.0	
1	1.0	0.0	9.0	0.0	4.0	0.0	0.0	0.0	
2	1.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	
3	1.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	
4	1.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	...	
119385	0.0	0.0	0.0	2.0	6.0	9.0	1.0	1.0	
119386	0.0	0.0	2.0	2.0	6.0	9.0	1.0	1.0	
119387	0.0	0.0	1.0	2.0	6.0	9.0	1.0	1.0	
119388	0.0	0.0	2.0	2.0	6.0	9.0	1.0	1.0	
119389	0.0	0.0	4.0	2.0	6.0	9.0	1.0	2.0	

119390 rows × 18 columns

In [ ]: