In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
raw_data = pd.read_csv('airline_passenger_satisfaction.csv')
```

In [4]:
```
raw_data.head(400)
```

Out[4]:

	ID	Gender	Age	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Dep and Conve
0	1	Male	48	First-time	Business	Business	821	2	5.0	
1	2	Female	35	Returning	Business	Business	821	26	39.0	
2	3	Male	41	Returning	Business	Business	853	0	0.0	
3	4	Male	50	Returning	Business	Business	1905	0	0.0	
4	5	Female	49	Returning	Business	Business	3470	0	1.0	
...	...	...	...	...	...	...	...	...	...	
395	396	Male	35	Returning	Business	Business	3838	0	3.0	
396	397	Female	26	First-time	Business	Business	158	0	1.0	
397	398	Male	52	Returning	Business	Business	134	0	0.0	
398	399	Male	38	First-time	Business	Business	164	0	0.0	
399	400	Male	23	First-time	Business	Economy	134	0	0.0	

400 rows × 24 columns

In [5]:
```
pd.options.display.max_rows= None
pd.options.display.max_columns = None
```

In [6]:
```
raw_data.head()
```

Out[6]:

	ID	Gender	Age	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Departu and Arri Ti Convenier
0	1	Male	48	First-time	Business	Business	821	2	5.0	
1	2	Female	35	Returning	Business	Business	821	26	39.0	
2	3	Male	41	Returning	Business	Business	853	0	0.0	
3	4	Male	50	Returning	Business	Business	1905	0	0.0	
4	5	Female	49	Returning	Business	Business	3470	0	1.0	

DROP 'ID'

In [8]:
```
raw_data_two = raw_data.copy()
```

In [9]:
```
raw_data_two = raw_data_two.drop(['ID','Age'], axis = 1)
```

In [10]:
```
raw_data_two.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 22 columns):
 #   Column                                  Non-Null Count   Dtype  
---  ------                                  --------------   -----  
 0   Gender                                  129880 non-null  object 
 1   Customer Type                           129880 non-null  object 
 2   Type of Travel                          129880 non-null  object 
 3   Class                                   129880 non-null  object 
 4   Flight Distance                         129880 non-null  int64  
 5   Departure Delay                         129880 non-null  int64  
 6   Arrival Delay                           129487 non-null  float64
 7   Departure and Arrival Time Convenience  129880 non-null  int64  
 8   Ease of Online Booking                  129880 non-null  int64  
 9   Check-in Service                        129880 non-null  int64  
 10  Online Boarding                         129880 non-null  int64  
 11  Gate Location                           129880 non-null  int64  
 12  On-board Service                        129880 non-null  int64  
 13  Seat Comfort                            129880 non-null  int64  
 14  Leg Room Service                        129880 non-null  int64  
 15  Cleanliness                             129880 non-null  int64  
 16  Food and Drink                          129880 non-null  int64  
 17  In-flight Service                       129880 non-null  int64  
 18  In-flight Wifi Service                  129880 non-null  int64  
 19  In-flight Entertainment                 129880 non-null  int64  
 20  Baggage Handling                        129880 non-null  int64  
 21  Satisfaction                            129880 non-null  object 
dtypes: float64(1), int64(16), object(5)
memory usage: 21.8+ MB
```

'Arrival Delay' type change to INT

In [12]:
```
raw_data_two['Arrival Delay'].unique()
```

Out[12]:
```
array([5.000e+00, 3.900e+01, 0.000e+00, 1.000e+00, 3.000e+00, 2.100e+01,
       3.000e+01, 7.600e+01, 2.000e+00, 1.500e+01, 2.400e+01, 5.200e+01,
       1.200e+01, 2.800e+01, 1.700e+01, 2.600e+01, 7.000e+01, 8.000e+00,
       1.400e+01, 1.300e+01, 4.100e+01, 2.900e+01, 1.000e+01, 7.100e+01,
       3.400e+01, 1.860e+02, 6.700e+01, 5.900e+01, 2.960e+02, 4.000e+00,
       7.000e+00, 2.540e+02, 7.800e+01, 1.230e+02, 3.500e+01, 1.100e+01,
       1.800e+01, 3.600e+01, 9.500e+01, 2.700e+01, 5.100e+01, 1.450e+02,
       1.220e+02, 4.000e+01, 2.500e+01, 1.920e+02, 6.000e+00, 1.170e+02,
       8.400e+01, 3.700e+01, 4.900e+01, 1.160e+02, 4.400e+01, 5.700e+01,
       1.600e+01, 2.600e+02, 6.100e+01, 1.360e+02, 6.200e+01, 5.000e+01,
       1.850e+02, 1.900e+01, 4.200e+01, 9.000e+00, 6.300e+01, 1.330e+02,
       5.600e+01, 4.300e+01, 8.500e+01, 3.100e+01, 5.500e+01, 2.930e+02,
       8.900e+01, 1.400e+02,       nan, 2.050e+02, 2.200e+01, 7.900e+01,
       2.130e+02, 9.900e+01, 1.670e+02, 7.400e+01, 1.200e+02, 3.200e+01,
       9.600e+01, 6.600e+01, 4.500e+01, 2.300e+01, 2.000e+01, 7.300e+01,
       5.800e+01, 5.300e+01, 3.300e+01, 9.800e+01, 1.190e+02, 6.000e+01,
       2.080e+02, 1.350e+02, 1.290e+02, 4.700e+01, 4.800e+01, 4.600e+01,
       3.800e+01, 1.740e+02, 1.240e+02, 1.710e+02, 2.950e+02, 1.650e+02,
       1.020e+02, 1.110e+02, 8.200e+01, 7.500e+01, 2.010e+02, 1.410e+02,
       4.170e+02, 3.170e+02, 2.360e+02, 1.340e+02, 5.400e+01, 1.420e+02,
       6.400e+01, 1.790e+02, 8.800e+01, 8.700e+01, 6.900e+01, 1.000e+02,
       2.830e+02, 9.300e+01, 1.080e+02, 9.700e+01, 2.170e+02, 1.210e+02,
       8.300e+01, 6.500e+01, 1.150e+02, 1.060e+02, 1.180e+02, 1.270e+02,
       1.930e+02, 9.400e+01, 1.750e+02, 8.220e+02, 8.000e+01, 2.720e+02,
       2.390e+02, 6.800e+01, 1.800e+02, 1.090e+02, 1.380e+02, 2.210e+02,
       1.140e+02, 2.980e+02, 7.200e+01, 1.720e+02, 1.040e+02, 1.760e+02,
       8.100e+01, 1.070e+02, 4.090e+02, 9.100e+01, 1.010e+02, 1.440e+02,
       2.790e+02, 1.870e+02, 1.580e+02, 2.560e+02, 7.700e+01, 5.240e+02,
       8.600e+01, 2.060e+02, 1.960e+02, 2.480e+02, 1.890e+02, 1.130e+02,
       2.970e+02, 2.440e+02, 2.290e+02, 1.500e+02, 1.840e+02, 9.200e+01,
       3.150e+02, 2.550e+02, 1.610e+02, 2.400e+02, 1.560e+02, 4.730e+02,
       2.570e+02, 1.310e+02, 9.000e+01, 2.300e+02, 1.490e+02, 1.690e+02,
       1.250e+02, 1.680e+02, 1.640e+02, 1.370e+02, 2.780e+02, 1.390e+02,
       9.520e+02, 1.950e+02, 1.570e+02, 3.420e+02, 1.120e+02, 1.530e+02,
       1.100e+02, 1.880e+02, 1.940e+02, 2.250e+02, 1.620e+02, 1.480e+02,
       3.560e+02, 1.050e+02, 3.380e+02, 2.030e+02, 1.011e+03, 1.550e+02,
       1.600e+02, 1.030e+02, 2.160e+02, 7.020e+02, 1.260e+02, 2.860e+02,
       1.540e+02, 2.240e+02, 3.780e+02, 2.420e+02, 1.280e+02, 3.470e+02,
       7.050e+02, 2.450e+02, 1.730e+02, 3.480e+02, 3.030e+02, 1.300e+02,
       1.830e+02, 3.520e+02, 1.990e+02, 2.090e+02, 7.950e+02, 1.590e+02,
       1.660e+02, 2.870e+02, 2.270e+02, 2.370e+02, 2.460e+02, 2.000e+02,
       1.510e+02, 2.610e+02, 2.100e+02, 3.630e+02, 1.430e+02, 2.750e+02,
       2.140e+02, 1.780e+02, 4.910e+02, 1.320e+02, 1.470e+02, 4.010e+02,
       2.220e+02, 9.200e+02, 4.580e+02, 1.810e+02, 3.300e+02, 4.320e+02,
       4.550e+02, 3.260e+02, 3.580e+02, 2.680e+02, 1.460e+02, 1.900e+02,
       1.980e+02, 2.850e+02, 4.460e+02, 2.730e+02, 1.520e+02, 1.820e+02,
       2.260e+02, 1.770e+02, 1.700e+02, 2.530e+02, 2.470e+02, 2.760e+02,
       3.330e+02, 2.180e+02, 3.370e+02, 2.620e+02, 2.150e+02, 4.850e+02,
       1.630e+02, 2.330e+02, 6.240e+02, 2.380e+02, 3.290e+02, 5.160e+02,
       3.240e+02, 3.350e+02, 3.810e+02, 2.700e+02, 2.810e+02, 2.880e+02,
       4.060e+02, 4.840e+02, 2.640e+02, 2.510e+02, 2.430e+02, 4.160e+02,
       2.650e+02, 2.320e+02, 2.120e+02, 4.070e+02, 3.950e+02, 3.450e+02,
       3.110e+02, 2.190e+02, 3.920e+02, 2.840e+02, 3.360e+02, 3.070e+02,
       5.890e+02, 2.200e+02, 2.230e+02, 2.580e+02, 3.460e+02, 3.800e+02,
       2.280e+02, 1.970e+02, 4.100e+02, 3.970e+02, 2.020e+02, 2.910e+02,
       3.500e+02, 3.100e+02, 5.180e+02, 2.520e+02, 3.620e+02, 3.200e+02,
```

```
       2.070e+02, 2.590e+02, 1.910e+02, 3.060e+02, 3.440e+02, 2.500e+02,
       3.340e+02, 4.700e+02, 3.860e+02, 3.120e+02, 2.900e+02, 3.490e+02,
       3.190e+02, 3.140e+02, 3.510e+02, 2.630e+02, 3.570e+02, 3.010e+02,
       3.210e+02, 3.220e+02, 2.110e+02, 4.570e+02, 3.930e+02, 2.710e+02,
       5.000e+02, 6.000e+02, 4.930e+02, 3.640e+02, 2.740e+02, 4.400e+02,
       2.410e+02, 3.720e+02, 5.020e+02, 3.020e+02, 2.350e+02, 4.030e+02,
       4.860e+02, 4.020e+02, 4.710e+02, 5.610e+02, 3.820e+02, 5.930e+02,
       4.380e+02, 3.700e+02, 2.820e+02, 4.480e+02, 4.270e+02, 3.710e+02,
       4.040e+02, 3.910e+02, 2.890e+02, 5.860e+02, 4.180e+02, 2.040e+02,
       3.050e+02, 4.450e+02, 3.890e+02, 3.660e+02, 3.550e+02, 2.660e+02,
       3.310e+02, 3.000e+02, 3.830e+02, 3.770e+02, 2.770e+02, 6.080e+02,
       2.990e+02, 3.590e+02, 4.340e+02, 3.230e+02, 6.040e+02, 4.290e+02,
       2.800e+02, 4.000e+02, 9.400e+02, 3.180e+02, 6.910e+02, 6.150e+02,
       3.690e+02, 3.530e+02, 8.600e+02, 1.280e+03, 3.740e+02, 2.490e+02,
       6.380e+02, 1.584e+03, 1.115e+03, 3.160e+02, 4.440e+02, 3.540e+02,
       4.430e+02, 4.330e+02, 2.310e+02, 7.290e+02, 9.700e+02, 4.350e+02,
       3.410e+02, 3.390e+02, 3.250e+02, 8.230e+02, 3.080e+02, 4.600e+02,
       3.040e+02, 3.270e+02, 9.240e+02, 3.990e+02, 3.130e+02, 4.200e+02,
       3.090e+02, 5.800e+02, 2.340e+02, 3.680e+02, 2.940e+02, 2.690e+02,
       7.170e+02, 4.310e+02, 7.480e+02, 4.590e+02, 7.200e+02, 5.430e+02,
       5.670e+02, 5.550e+02, 4.540e+02, 2.920e+02, 4.250e+02, 3.880e+02,
       4.220e+02, 4.360e+02, 3.600e+02, 5.070e+02, 2.670e+02, 4.240e+02,
       3.850e+02, 4.130e+02, 5.030e+02, 3.790e+02, 4.120e+02])
```

In [13]:
```
#for i in range(raw_data_two['Arrival Delay'].shape[0]):
    #if np.isnan(raw_data_two['Arrival Delay'][i]):
        #raw_data_two['Arrival Delay'][i] = 0.0
```

In [14]:
```
raw_data_two = raw_data_two.fillna(value=0)
```

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

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 22 columns):
 #   Column                                  Non-Null Count   Dtype  
---  ------                                  --------------   -----  
 0   Gender                                  129880 non-null  object 
 1   Customer Type                           129880 non-null  object 
 2   Type of Travel                          129880 non-null  object 
 3   Class                                   129880 non-null  object 
 4   Flight Distance                         129880 non-null  int64  
 5   Departure Delay                         129880 non-null  int64  
 6   Arrival Delay                           129880 non-null  float64
 7   Departure and Arrival Time Convenience  129880 non-null  int64  
 8   Ease of Online Booking                  129880 non-null  int64  
 9   Check-in Service                        129880 non-null  int64  
 10  Online Boarding                         129880 non-null  int64  
 11  Gate Location                           129880 non-null  int64  
 12  On-board Service                        129880 non-null  int64  
 13  Seat Comfort                            129880 non-null  int64  
 14  Leg Room Service                        129880 non-null  int64  
 15  Cleanliness                             129880 non-null  int64  
 16  Food and Drink                          129880 non-null  int64  
 17  In-flight Service                       129880 non-null  int64  
 18  In-flight Wifi Service                  129880 non-null  int64  
 19  In-flight Entertainment                 129880 non-null  int64  
 20  Baggage Handling                        129880 non-null  int64  
 21  Satisfaction                            129880 non-null  object 
dtypes: float64(1), int64(16), object(5)
memory usage: 21.8+ MB
```

In [16]:
```
raw_data_two['Arrival Delay'] = raw_data_two['Arrival Delay'].astype(int)
```

In [17]:
```
raw_data_two.head(20)
```

Out[17]:

	Gender	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ea O Boo
0	Male	First-time	Business	Business	821	2	5	3	
1	Female	Returning	Business	Business	821	26	39	2	
2	Male	Returning	Business	Business	853	0	0	4	
3	Male	Returning	Business	Business	1905	0	0	2	
4	Female	Returning	Business	Business	3470	0	1	3	
5	Male	Returning	Business	Business	3788	0	0	4	
6	Male	Returning	Business	Business	1963	0	0	3	
7	Female	Returning	Business	Business	853	0	3	3	
8	Male	Returning	Business	Business	2607	0	0	1	
9	Female	Returning	Business	Business	2822	13	0	2	
10	Female	First-time	Business	Business	821	0	5	1	
11	Female	First-time	Business	Business	421	20	21	2	
12	Male	First-time	Business	Economy	453	16	30	2	
13	Male	Returning	Personal	Business	853	68	76	5	
14	Male	Returning	Personal	Economy	853	0	0	4	
15	Male	Returning	Personal	Economy	821	0	0	5	
16	Female	Returning	Personal	Economy	821	0	0	3	
17	Female	Returning	Personal	Economy	821	0	0	5	
18	Female	Returning	Personal	Economy	853	0	0	5	
19	Female	Returning	Personal	Economy	821	4	0	3	

DATA MAPPING

In [19]:
```
satisfaction_mapping = {'Neutral or Dissatisfied':0,'Satisfied':1}
```

In [20]:
```
raw_data_two['Satisfaction'] = raw_data_two['Satisfaction'].map(satisfaction_mappin
```

In [21]:
```
gender_mapping = {'Male':0,'Female':1}
```

In [22]:
```
raw_data_two['Gender'] = raw_data_two['Gender'].map(gender_mapping)
```

In [23]:
```
customer_type_mapping = {'First-time':0,'Returning':1}
```

In [24]:
```
 raw_data_two['Customer Type'] = raw_data_two['Customer Type'].map(customer_type_ma
```

In [25]:
```
type_travel_mapping = {'Business':0,'Personal':1}
```

In [26]:
```
raw_data_two['Type of Travel'] = raw_data_two['Type of Travel'].map(type_travel_map
```

Dummies from 'Class'

In [28]:
```
class_columns  = pd.get_dummies(raw_data_two['Class'])
```

In [29]:
```
class_columns.head()
```

Out[29]:

	Business	Economy	Economy Plus
0	True	False	False
1	True	False	False
2	True	False	False
3	True	False	False
4	True	False	False

In [30]:
```
class_columns.shape
```

Out[30]:
```
(129880, 3)
```

In [31]:
```
class_columns['check'] = class_columns.sum(axis = 1)
class_columns.head()
```

Out[31]:

	Business	Economy	Economy Plus	check
0	True	False	False	1
1	True	False	False	1
2	True	False	False	1
3	True	False	False	1
4	True	False	False	1

In [32]:
```
class_columns['check'].sum(axis = 0)
```

Out[32]:
```
129880
```

In [33]:
```
class_columns = class_columns.drop(['check'], axis = 1)
```

In [34]:
```
class_columns.head()
```

Out[34]:

	Business	Economy	Economy Plus
0	True	False	False
1	True	False	False
2	True	False	False
3	True	False	False
4	True	False	False

In [35]:
```
class_columns.columns = ['Business Class','Economy Class', 'Economy Plus Class']
```

In [36]:
```
class_columns.head()
```

Out[36]:

	Business Class	Economy Class	Economy Plus Class
0	True	False	False
1	True	False	False
2	True	False	False
3	True	False	False
4	True	False	False

CONCATENATE CLASS COLUMNS

In [38]:
```
raw_data_two = pd.concat([raw_data_two,class_columns],axis = 1)
raw_data_two.head()
```

Out[38]:

	Gender	Customer Type	Type of Travel	Class	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ease o Onlin Booking
0	0	0	0	Business	821	2	5	3	
1	1	1	0	Business	821	26	39	2	
2	0	1	0	Business	853	0	0	4	
3	0	1	0	Business	1905	0	0	2	
4	1	1	0	Business	3470	0	1	3	

In [39]:
```
raw_data_two = raw_data_two.drop(['Class'],axis = 1 )
```

In [40]:
```
raw_data_two.head()
```

Out[40]:

	Gender	Customer Type	Type of Travel	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ease of Online Booking	Check-in Service
0	0	0	0	821	2	5	3	3	4
1	1	1	0	821	26	39	2	2	3
2	0	1	0	853	0	0	4	4	4
3	0	1	0	1905	0	0	2	2	3
4	1	1	0	3470	0	1	3	3	3

COLUMNS REORDER

In [42]:
```
raw_data_two.columns.values
```

Out[42]:
```
array(['Gender', 'Customer Type', 'Type of Travel', 'Flight Distance',
       'Departure Delay', 'Arrival Delay',
       'Departure and Arrival Time Convenience', 'Ease of Online Booking',
       'Check-in Service', 'Online Boarding', 'Gate Location',
       'On-board Service', 'Seat Comfort', 'Leg Room Service',
       'Cleanliness', 'Food and Drink', 'In-flight Service',
       'In-flight Wifi Service', 'In-flight Entertainment',
       'Baggage Handling', 'Satisfaction', 'Business Class',
       'Economy Class', 'Economy Plus Class'], dtype=object)
```

In [43]:
```
columns_order = ['Gender', 'Customer Type', 'Type of Travel','Business Class',
       'Economy Class', 'Economy Plus Class',
       'Flight Distance', 'Departure Delay', 'Arrival Delay',
       'Departure and Arrival Time Convenience', 'Ease of Online Booking',
       'Check-in Service', 'Online Boarding', 'Gate Location',
       'On-board Service', 'Seat Comfort', 'Leg Room Service',
       'Cleanliness', 'Food and Drink', 'In-flight Service',
       'In-flight Wifi Service', 'In-flight Entertainment',
       'Baggage Handling', 'Satisfaction',]
```

In [44]:
```
raw_data_two = raw_data_two[columns_order]
```

In [45]:
```
raw_data_two.head()
```

Out[45]:

	Gender	Customer Type	Type of Travel	Business Class	Economy Class	Economy Plus Class	Flight Distance	Departure Delay	Arrival Delay
0	0	0	0	True	False	False	821	2	5
1	1	1	0	True	False	False	821	26	39
2	0	1	0	True	False	False	853	0	0
3	0	1	0	True	False	False	1905	0	0
4	1	1	0	True	False	False	3470	0	1

In [46]:
```
raw_data_cmod = raw_data_two.copy()
```

In [47]:
```
raw_data_cmod.head()
```

Out[47]:

	Gender	Customer Type	Type of Travel	Business Class	Economy Class	Economy Plus Class	Flight Distance	Departure Delay	Arrival Delay
0	0	0	0	True	False	False	821	2	5
1	1	1	0	True	False	False	821	26	39
2	0	1	0	True	False	False	853	0	0
3	0	1	0	True	False	False	1905	0	0
4	1	1	0	True	False	False	3470	0	1

REMOVING ROWS WITH SATISFACTION LEVEL = 0 ('not applicable') <h1

>

In [49]:
```
columns_to_check = ['Ease of Online Booking',
       'Check-in Service', 'Online Boarding', 'Gate Location',
       'On-board Service', 'Seat Comfort', 'Leg Room Service',
       'Cleanliness', 'Food and Drink', 'In-flight Service',
       'In-flight Wifi Service', 'In-flight Entertainment',
       'Baggage Handling']
```

In [50]:
```
for column in columns_to_check:
    zero_count = (raw_data_cmod[column] == 0).sum()
    print(f" {column}: {zero_count}")
```

```
Ease of Online Booking: 5682
Check-in Service: 1
Online Boarding: 3080
Gate Location: 1
On-board Service: 5
Seat Comfort: 1
Leg Room Service: 598
Cleanliness: 14
Food and Drink: 132
In-flight Service: 5
In-flight Wifi Service: 3916
In-flight Entertainment: 18
Baggage Handling: 0
```

In [51]:
```
def remove_rows_with_zero_in_columns(df, columns_to_check):

    mask = df[columns_to_check].ne(0).all(axis=1)

    df_filtered = df[mask]

    return df_filtered
```

In [52]:
```
raw_data_cmod = remove_rows_with_zero_in_columns(raw_data_cmod, columns_to_check)
```

In [53]:
```
print(f"Rozmiar po: {raw_data_cmod.shape[0]}")
```

```
Rozmiar po: 123878
```

In [54]:
```
for column in columns_to_check:
    zero_count = (raw_data_cmod[column] == 0).sum()
    print(f" {column}: {zero_count}")
```

```
Ease of Online Booking: 0
Check-in Service: 0
Online Boarding: 0
Gate Location: 0
On-board Service: 0
Seat Comfort: 0
Leg Room Service: 0
Cleanliness: 0
Food and Drink: 0
In-flight Service: 0
In-flight Wifi Service: 0
In-flight Entertainment: 0
Baggage Handling: 0
```

FINAL CHECKPOINT

In [56]:
```
data_preprocessed = raw_data_cmod.copy()
```

In [57]:
```
contains_nan = data_preprocessed.isna().any().any()

if contains_nan:
    print("DataFrame zawiera wartości NaN.")
else:
    print("DataFrame nie zawiera wartości NaN.")
```

```
DataFrame nie zawiera wartości NaN.
```

In [58]:
```
data_preprocessed.head()
```

Out[58]:

	Gender	Customer Type	Type of Travel	Business Class	Economy Class	Economy Plus Class	Flight Distance	Departure Delay	Arrival Delay
0	0	0	0	True	False	False	821	2	5
1	1	1	0	True	False	False	821	26	39
2	0	1	0	True	False	False	853	0	0
3	0	1	0	True	False	False	1905	0	0
4	1	1	0	True	False	False	3470	0	1

In [59]:
```
data_preprocessed.to_csv('flight_satisfaction_preprocessed.csv',index = False)
```