INTRODUCTION

Objective

Predicting airline passenger satisfaction using random forest, gradient boosting, and KNN.

Dataset

https://www.kaggle.com/datasets/mysarahmadbhat/airline-passenger-satisfaction?resource=download (https://www.kaggle.com/datasets/mysarahmadbhat/airline-passenger-satisfaction?resource=download)

Data Dictionary:

1. **ID**: Unique passenger identifier
2. **Gender**: Gender of the passenger (Female/Male)
3. **Age**: Age of the passenger
4. **Customer Type**: Type of airline customer (First-time/Returning)
5. **Type of Travel**: Purpose of the flight (Business/Personal)
6. **Class**: Travel class in the airplane for the passenger seat
7. **Flight Distance**: Flight distance in miles
8. **Departure & Arrival Delay**: Flight departure & arrival delay in minutes
9. **Satisfaction**: Overall satisfaction level with the airline (Satisfied/Neutral or unsatisfied)

```
"Satisfaction level from 1 (lowest) to 5 (highest) - 0 means ""not applicable""
```

10. **Departure & Arrival Time Convenience**
11. **Ease of Online Booking**
12. **Check-in Service**
13. **Online Boarding**
14. **Gate Location**
15. **On-board Service**
16. **Seat Comfort**
17. **Leg Room Service**
18. **Cleanliness**
19. **Food and Drink**
20. **In-flight Service**
21. **In-flight Wifi Service**
22. **In-flight Entertainment**
23. **Baggage Handling**

Below are the steps executed in this notebook

1. IMPORT LIBRARIES
2. LOAD DATASET
3. DATA UNDERSTANDING

- 1. Check Data Description
- 2. Check data info
- 3. Check Missing Value

4. DATA PREPARATION

- 1. Handling Missing Value
- 2. Duplicated Data

5. STATISTICAL SUMMARY

- 1. Numerical columns
- 2. Categorical Columns

6. Outlier Detection

7. Encoding Categorical Columns

- 1. One-hot encoding
- 2. Label encoding on Target Column

8. Export encoded data for EDA

1. IMPORT LIBRARIES

In [1]:

```
import warnings
warnings.filterwarnings('ignore')
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. LOAD DATASET

In [2]:

```
df = pd.read_csv('airline_passenger_satisfaction.csv')
print('Total Row : ', len(df))
df.head(5)
```

```
Total Row :  129880
```

Out[2]:

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

5 rows × 24 columns

3. DATA UNDERSTANDING

1. Check Data Description

In [3]:

```
df.describe()
```

Out[3]:

	ID	Age	Flight Distance	Departure Delay	Arrival Delay	Departur Arrival Conven
count	129880.000000	129880.000000	129880.000000	129880.000000	129487.000000	129880.0
mean	64940.500000	39.427957	1190.316392	14.713713	15.091129	3.0
std	37493.270818	15.119360	997.452477	38.071126	38.465650	1.5
min	1.000000	7.000000	31.000000	0.000000	0.000000	0.0
25%	32470.750000	27.000000	414.000000	0.000000	0.000000	2.0
50%	64940.500000	40.000000	844.000000	0.000000	0.000000	3.0
75%	97410.250000	51.000000	1744.000000	12.000000	13.000000	4.0
max	129880.000000	85.000000	4983.000000	1592.000000	1584.000000	5.0

2. Check data info

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 129880 entries, 0 to 129879
Data columns (total 24 columns):
 #   Column                                  Non-Null Count   Dtype  
---  ------                                  --------------   -----  
 0   ID                                      129880 non-null  int64  
 1   Gender                                  129880 non-null  object 
 2   Age                                     129880 non-null  int64  
 3   Customer Type                           129880 non-null  object 
 4   Type of Travel                          129880 non-null  object 
 5   Class                                   129880 non-null  object 
 6   Flight Distance                         129880 non-null  int64  
 7   Departure Delay                         129880 non-null  int64  
 8   Arrival Delay                           129487 non-null  float64
 9   Departure and Arrival Time Convenience  129880 non-null  int64  
 10  Ease of Online Booking                  129880 non-null  int64  
 11  Check-in Service                        129880 non-null  int64  
 12  Online Boarding                         129880 non-null  int64  
 13  Gate Location                           129880 non-null  int64  
 14  On-board Service                        129880 non-null  int64  
 15  Seat Comfort                            129880 non-null  int64  
 16  Leg Room Service                        129880 non-null  int64  
 17  Cleanliness                             129880 non-null  int64  
 18  Food and Drink                          129880 non-null  int64  
 19  In-flight Service                       129880 non-null  int64  
 20  In-flight Wifi Service                  129880 non-null  int64  
 21  In-flight Entertainment                 129880 non-null  int64  
 22  Baggage Handling                        129880 non-null  int64  
 23  Satisfaction                            129880 non-null  object 
dtypes: float64(1), int64(18), object(5)
memory usage: 23.8+ MB
```

3. Check Missing Value

In [5]:

```
null_value = (129880 - 129487 ) /129880
percentage = null_value * 100

print("missing value = {:.1f}%".format(percentage))
```

```
missing value = 0.3%
```

4. DATA PREPARATION

1. Handling Missing Value

Since there are very few missing values, the rows containing missing values will be dropped.

In [6]:

```
df.dropna(inplace=True)
```

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129487 entries, 0 to 129879
Data columns (total 24 columns):
 #   Column                                  Non-Null Count   Dtype  
---  ------                                  --------------   -----  
 0   ID                                      129487 non-null  int64  
 1   Gender                                  129487 non-null  object 
 2   Age                                     129487 non-null  int64  
 3   Customer Type                           129487 non-null  object 
 4   Type of Travel                          129487 non-null  object 
 5   Class                                   129487 non-null  object 
 6   Flight Distance                         129487 non-null  int64  
 7   Departure Delay                         129487 non-null  int64  
 8   Arrival Delay                           129487 non-null  float64
 9   Departure and Arrival Time Convenience  129487 non-null  int64  
 10  Ease of Online Booking                  129487 non-null  int64  
 11  Check-in Service                        129487 non-null  int64  
 12  Online Boarding                         129487 non-null  int64  
 13  Gate Location                           129487 non-null  int64  
 14  On-board Service                        129487 non-null  int64  
 15  Seat Comfort                            129487 non-null  int64  
 16  Leg Room Service                        129487 non-null  int64  
 17  Cleanliness                             129487 non-null  int64  
 18  Food and Drink                          129487 non-null  int64  
 19  In-flight Service                       129487 non-null  int64  
 20  In-flight Wifi Service                  129487 non-null  int64  
 21  In-flight Entertainment                 129487 non-null  int64  
 22  Baggage Handling                        129487 non-null  int64  
 23  Satisfaction                            129487 non-null  object 
dtypes: float64(1), int64(18), object(5)
memory usage: 24.7+ MB
```

NO MORE MISSING VALUE DETECTED

2. Duplicated Data

In [8]:

```
df.duplicated().sum()
```

Out[8]:

```
0
```

5. STATISTICAL SUMMARY

In [9]:

```
# select columns with categorical data and save column names
categoricals = list(df.select_dtypes(include=['object']).columns)

# select columns with numerical data and save column names
numericals = list(df.select_dtypes(include=['float', 'int']).columns)

categorical_count = len(df.select_dtypes(include=['object']).columns)
numerical_count = len(df.select_dtypes(include=['float', 'int']).columns)


# print column names
print('Categorical columns:', categorical_count,"->", categoricals)
print('Numerical columns:', numerical_count, "->",numericals)
```

```
Categorical columns: 5 -> ['Gender', 'Customer Type', 'Type of Travel', 'C
lass', 'Satisfaction']
Numerical columns: 19 -> ['ID', 'Age', 'Flight Distance', 'Departure Dela
y', 'Arrival Delay', 'Departure and Arrival Time Convenience', 'Ease of On
line 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 Ente
rtainment', 'Baggage Handling']
```

1. Numerical columns

In [10]:

```
df[numericals].describe().T
```

Out[10]:

	count	mean	std	min	25%	50%	75%	ma
ID	129487.0	64958.335169	37489.781165	1.0	32494.5	64972.0	97415.5	129880.
Age	129487.0	39.428761	15.117597	7.0	27.0	40.0	51.0	85.
Flight Distance	129487.0	1190.210662	997.560954	31.0	414.0	844.0	1744.0	4983.
Departure Delay	129487.0	14.643385	37.932867	0.0	0.0	0.0	12.0	1592.
Arrival Delay	129487.0	15.091129	38.465650	0.0	0.0	0.0	13.0	1584.
Departure and Arrival Time Convenience	129487.0	3.057349	1.526787	0.0	2.0	3.0	4.0	5.
Ease of Online Booking	129487.0	2.756786	1.401662	0.0	2.0	3.0	4.0	5.
Check-in Service	129487.0	3.306239	1.266146	0.0	3.0	3.0	4.0	5.
Online Boarding	129487.0	3.252720	1.350651	0.0	2.0	3.0	4.0	5.
Gate Location	129487.0	2.976909	1.278506	0.0	2.0	3.0	4.0	5.
On-board Service	129487.0	3.383204	1.287032	0.0	2.0	4.0	4.0	5.
Seat Comfort	129487.0	3.441589	1.319168	0.0	2.0	4.0	5.0	5.
Leg Room Service	129487.0	3.351078	1.316132	0.0	2.0	4.0	4.0	5.
Cleanliness	129487.0	3.286222	1.313624	0.0	2.0	3.0	4.0	5.
Food and Drink	129487.0	3.204685	1.329905	0.0	2.0	3.0	4.0	5.
In-flight Service	129487.0	3.642373	1.176614	0.0	3.0	4.0	5.0	5.
In-flight Wifi Service	129487.0	2.728544	1.329235	0.0	2.0	3.0	4.0	5.
In-flight Entertainment	129487.0	3.358067	1.334149	0.0	2.0	4.0	4.0	5.
Baggage Handling	129487.0	3.631886	1.180082	1.0	3.0	4.0	5.0	5.

In [11]:

```
filtered_columns = [col for col in numericals if df[col].mean() > df[col].median()]

# print filtered columns
print('Numerical columns with mean greater than median:', filtered_columns)
```

```
Numerical columns with mean greater than median: ['Flight Distance', 'Depa
rture Delay', 'Arrival Delay', 'Departure and Arrival Time Convenience',
'Check-in Service', 'Online Boarding', 'Cleanliness', 'Food and Drink']
```

OBSERVATION:

Min-Max gap per column:

- `ID` is a key value so we can ignore
- `Age` has normal gap
- `Flight Distance` , `Departure Delay` , `Arrival Delay` the gap is too big, not normal.
- For the remaining columns, since they have only 1-5 unique values, they can be ignored when looking at their minimum and maximum values.

columns with skewed distribution because mean > median :

`Flight Distance` , `Departure Delay` , `Arrival Delay` , `Departure and Arrival Time Convenience` , `Check-in Service` , `Online Boarding` , `Cleanliness` , `Food and Drink`

2. Categorical Columns

In [12]:

```
df[categoricals].describe()
```

Out[12]:

	Gender	Customer Type	Type of Travel	Class	Satisfaction
count	129487	129487	129487	129487	129487
unique	2	2	2	3	2
top	Female	Returning	Business	Business	Neutral or Dissatisfied
freq	65703	105773	89445	61990	73225

OBSERVATION:

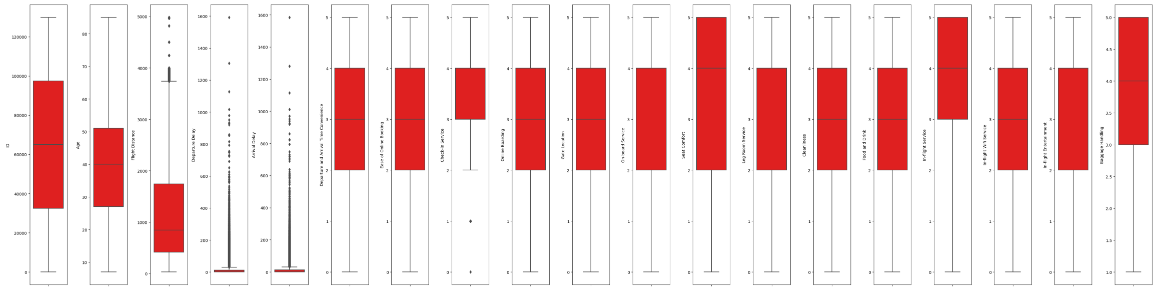
1. There are more female customers than male customers and more returning customers than new customers. The majority of the travel records are for business travel and business class, while the majority of the customers were neutral or dissatisfied with their travel experience.
2. The frequency percentage of Neutral or Dissatisfied passenger is 56% so this dataset is **imbalanced**

6. Outlier Detection

In [13]:

```
# adjust the figure size for better readability
plt.figure(figsize=(40,10))

# plotting
features = numericals
for i in range(0, len(features)):
    plt.subplot(1, len(features), i+1)
    sns.boxplot(y=df[features[i]], color='red')
    plt.tight_layout()
```



OBSERVATIONS:

Columns `Flight Distance` , `Departure Delay` , `Arrival Delay` , and `Check-in Service` has outliers

7. Encoding Categorical Columns

In [14]:

```
for col in categoricals:
    print(f"Unique values of {col}: {df[col].unique()}")
```

```
Unique values of Gender: ['Male' 'Female']
Unique values of Customer Type: ['First-time' 'Returning']
Unique values of Type of Travel: ['Business' 'Personal']
Unique values of Class: ['Business' 'Economy' 'Economy Plus']
Unique values of Satisfaction: ['Neutral or Dissatisfied' 'Satisfied']
```

1. One-hot encoding

In [15]:

```
df_encoded = pd.get_dummies(df, columns=['Gender', 'Customer Type', 'Type of Travel', 'Cl
```

In [16]:

```
df_encoded['Satisfaction'].unique()
```

Out[16]:

```
array(['Neutral or Dissatisfied', 'Satisfied'], dtype=object)
```

2. Label encoding on Target Column

In [17]:

```
# df_encoded['Satisfaction'] = (df_encoded['Satisfaction'] != 'Satisfied').astype(int)
df_encoded['Satisfaction'] = df_encoded['Satisfaction'].replace({"Neutral or Dissatisfiec
```

In [18]:

```
# Reorder column
df_encoded = df_encoded[['ID', 'Age', 'Flight Distance', 'Departure Delay', 'Arrival Dela
       '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',
       'Gender_Female', 'Gender_Male', 'Customer Type_First-time',
       'Customer Type_Returning', 'Type of Travel_Business',
       'Type of Travel_Personal', 'Class_Business', 'Class_Economy',
       'Class_Economy Plus','Satisfaction']]
```

In [19]:

```
df_encoded.head(3)
```

Out[19]:

	ID	Age	Flight Distance	Departure Delay	Arrival Delay	Departure and Arrival Time Convenience	Ease of Online Booking	Check-in Service	Online Boarding	Gate Location
0	1	48	821	2	5.0	3	3	4	3	3
1	2	35	821	26	39.0	2	2	3	5	2
2	3	41	853	0	0.0	4	4	4	5	4

3 rows × 29 columns

8. Export encoded data for EDA

In [20]:

```
df_encoded.to_csv("airline_passenger_satisfaction_EDA.csv",index=False)
```

In [ ]: