CIND 820 Big Data Analytics Project

Name: Nelly Grillo

Student number: 501144764

Supervisor: Ceni Babaoglu, Ph.D

Big Data Analytics Project

# Data Description

Gender: Gender of the passengers (Female, Male)

Customer Type: The customer type (Loyal customer, disloyal customer)

Age: The actual age of the passengers

Type of Travel: Purpose of the flight of the passengers (Personal travel, Business travel)

Class: Travel class in the plane of the passengers (Business, Eco, Eco Plus)

Flight distance: The flight distance of this journey

Inflight wifi service: Satisfaction level of the inflight wifi service (0:Not applicable; 1-5)

Departure/Arrival time convenient: Satisfaction level of Departure/Arrival time convenient

Ease of Online booking: Satisfaction level of online booking

Gate location: Satisfaction level of Gate location

Food and drink: Satisfaction level of Food and drink

Online boarding: Satisfaction level of Online boarding

Seat comfort: Satisfaction level of Seat comfort

Inflight entertainment: Satisfaction level of Inflight entertainment

On-board service: Satisfaction level of On-board service

Leg room service: Satisfaction level of Leg room service

Baggage handling: Satisfaction level of Baggage handling

Check-in service: Satisfaction level of Check-in service

Inflight service: Satisfaction level of Inflight service

Cleanliness: Satisfaction level of Cleanliness

Departure Delay in Minutes: Minutes delayed when departure

Arrival Delay in Minutes: Minutes delayed when Arrival

Satisfaction: Airline Satisfaction level (Satisfied, neutral or dissatisfied)

# Library Imports

In [96]:
```
import pandas as pd
```

```
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

In [97]:
```
# Loading the dataset
df_train = pd.read_csv('train.csv', index_col='id')
df_test = pd.read_csv('test.csv')
```

In [98]:
```
df_train.columns
```

Out[98]:
```
Index(['Unnamed: 0', 'Gender', 'Customer Type', 'Age', 'Type of Travel',
       'Class', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes', 'Arrival Delay in Minutes',
       'satisfaction'],
      dtype='object')
```

In [99]:
```
df_train.shape
```

Out[99]:
```
(103904, 24)
```

- 'Unnamed:0' will be deleted

In [100…
```
# Drop "Unnamed"
df_train = df_train.drop('Unnamed: 0', axis=1)
df_train = df_train.sort_values('id', ascending= True)

df_test = df_test.drop('Unnamed: 0', axis=1)
df_test = df_test.sort_values('id', ascending= True)
```

In [101…
```
# Looking at first few instances
df_train.head()
```

Out[101]:

| id | Gender | Customer Type | Age | Type of Travel | Class | Flight Distance | Inflight wifi service | Departure/Arrival time convenient | Ease of Online booking | Gate location | ... | Inflight entertainment | On-board service |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | Male | disloyal Customer | 48 | Business travel | Business | 821 | 3 | 3 | 3 | 3 | ... | 5 | 3 |
| **2** | Female | Loyal Customer | 35 | Business travel | Business | 821 | 2 | 2 | 2 | 2 | ... | 5 | 5 |
| **3** | Male | Loyal Customer | 41 | Business travel | Business | 853 | 4 | 4 | 4 | 4 | ... | 3 | 3 |
| **4** | Male | Loyal Customer | 50 | Business travel | Business | 1905 | 2 | 2 | 2 | 2 | ... | 5 | 5 |
| **5** | Female | Loyal Customer | 49 | Business travel | Business | 3470 | 3 | 3 | 3 | 3 | ... | 3 | 3 |

5 rows × 23 columns

In [102…
```
print("The data shape is: {}".format(df_train.shape))
```

```
The data shape is: (103904, 23)
```

In [103…
```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 103904 entries, 1 to 129880
Data columns (total 23 columns):
 #   Column                            Non-Null Count   Dtype  
---  ------                            --------------   -----  
 0   Gender                            103904 non-null  object 
 1   Customer Type                     103904 non-null  object 
 2   Age                               103904 non-null  int64  
 3   Type of Travel                    103904 non-null  object 
 4   Class                             103904 non-null  object 
 5   Flight Distance                   103904 non-null  int64  
 6   Inflight wifi service             103904 non-null  int64  
 7   Departure/Arrival time convenient  103904 non-null  int64  
 8   Ease of Online booking            103904 non-null  int64  
 9   Gate location                     103904 non-null  int64  
 10  Food and drink                    103904 non-null  int64  
 11  Online boarding                   103904 non-null  int64  
 12  Seat comfort                      103904 non-null  int64  
 13  Inflight entertainment            103904 non-null  int64  
 14  On-board service                  103904 non-null  int64  
 15  Leg room service                  103904 non-null  int64  
 16  Baggage handling                  103904 non-null  int64  
 17  Checkin service                   103904 non-null  int64  
 18  Inflight service                  103904 non-null  int64  
 19  Cleanliness                       103904 non-null  int64  
 20  Departure Delay in Minutes        103904 non-null  int64  
 21  Arrival Delay in Minutes          103594 non-null  float64
 22  satisfaction                      103904 non-null  object 
dtypes: float64(1), int64(17), object(5)
memory usage: 19.0+ MB
```

In [104…
```
df_train.nunique()[:10].sort_values(ascending=False)
```

Out[104]:
```
Flight Distance                     3802
Age                                   75
Inflight wifi service                  6
Departure/Arrival time convenient      6
Ease of Online booking                 6
Gate location                          6
Class                                  3
Gender                                 2
Customer Type                          2
Type of Travel                         2
dtype: int64
```

# Data Cleaning

1. NaN Values

In [105…
```
df_train.isnull().sum()
```

Out[105]:
```
Gender                                0
Customer Type                         0
Age                                   0
Type of Travel                        0
Class                                 0
Flight Distance                       0
Inflight wifi service                 0
Departure/Arrival time convenient     0
Ease of Online booking                0
Gate location                         0
Food and drink                        0
Online boarding                       0
Seat comfort                          0
Inflight entertainment                0
On-board service                      0
Leg room service                      0
Baggage handling                      0
Checkin service                       0
Inflight service                      0
Cleanliness                           0
Departure Delay in Minutes            0
Arrival Delay in Minutes            310
satisfaction                          0
dtype: int64
```

There are 310 missing values in the Arrival Delay in Minutes row. To avoid skewing the data, the NaN values will be dropped.

In [106…
```
# Dropping NaN rows
df_train = df_train.dropna().copy()

print("The data shape is: {}".format(df_train.shape))
```

```
The data shape is: (103594, 23)
```

1. Duplicate Values

In [107…
```
df_train.duplicated().any()
```

Out[107]:
```
False
```

# Descriptive Statistics

In [108…
```
df_train.describe().T
```

Out[108]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **Age** | 103594.0 | 39.380466 | 15.113125 | 7.0 | 27.0 | 40.0 | 51.0 | 85.0 |
| **Flight Distance** | 103594.0 | 1189.325202 | 997.297235 | 31.0 | 414.0 | 842.0 | 1743.0 | 4983.0 |
| **Inflight wifi service** | 103594.0 | 2.729753 | 1.327866 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Departure/Arrival time convenient** | 103594.0 | 3.060081 | 1.525233 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Ease of Online booking** | 103594.0 | 2.756984 | 1.398934 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Gate location** | 103594.0 | 2.977026 | 1.277723 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Food and drink** | 103594.0 | 3.202126 | 1.329401 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Online boarding** | 103594.0 | 3.250497 | 1.349433 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Seat comfort** | 103594.0 | 3.439765 | 1.318896 | 0.0 | 2.0 | 4.0 | 5.0 | 5.0 |
| **Inflight entertainment** | 103594.0 | 3.358341 | 1.333030 | 0.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| **On-board service** | 103594.0 | 3.382609 | 1.288284 | 0.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| **Leg room service** | 103594.0 | 3.351401 | 1.315409 | 0.0 | 2.0 | 4.0 | 4.0 | 5.0 |
| **Baggage handling** | 103594.0 | 3.631687 | 1.181051 | 1.0 | 3.0 | 4.0 | 5.0 | 5.0 |
| **Checkin service** | 103594.0 | 3.304323 | 1.265396 | 0.0 | 3.0 | 3.0 | 4.0 | 5.0 |
| **Inflight service** | 103594.0 | 3.640761 | 1.175603 | 0.0 | 3.0 | 4.0 | 5.0 | 5.0 |
| **Cleanliness** | 103594.0 | 3.286397 | 1.312194 | 0.0 | 2.0 | 3.0 | 4.0 | 5.0 |
| **Departure Delay in Minutes** | 103594.0 | 14.747939 | 38.116737 | 0.0 | 0.0 | 0.0 | 12.0 | 1592.0 |
| **Arrival Delay in Minutes** | 103594.0 | 15.178678 | 38.698682 | 0.0 | 0.0 | 0.0 | 13.0 | 1584.0 |

1. Outliers

In [109…

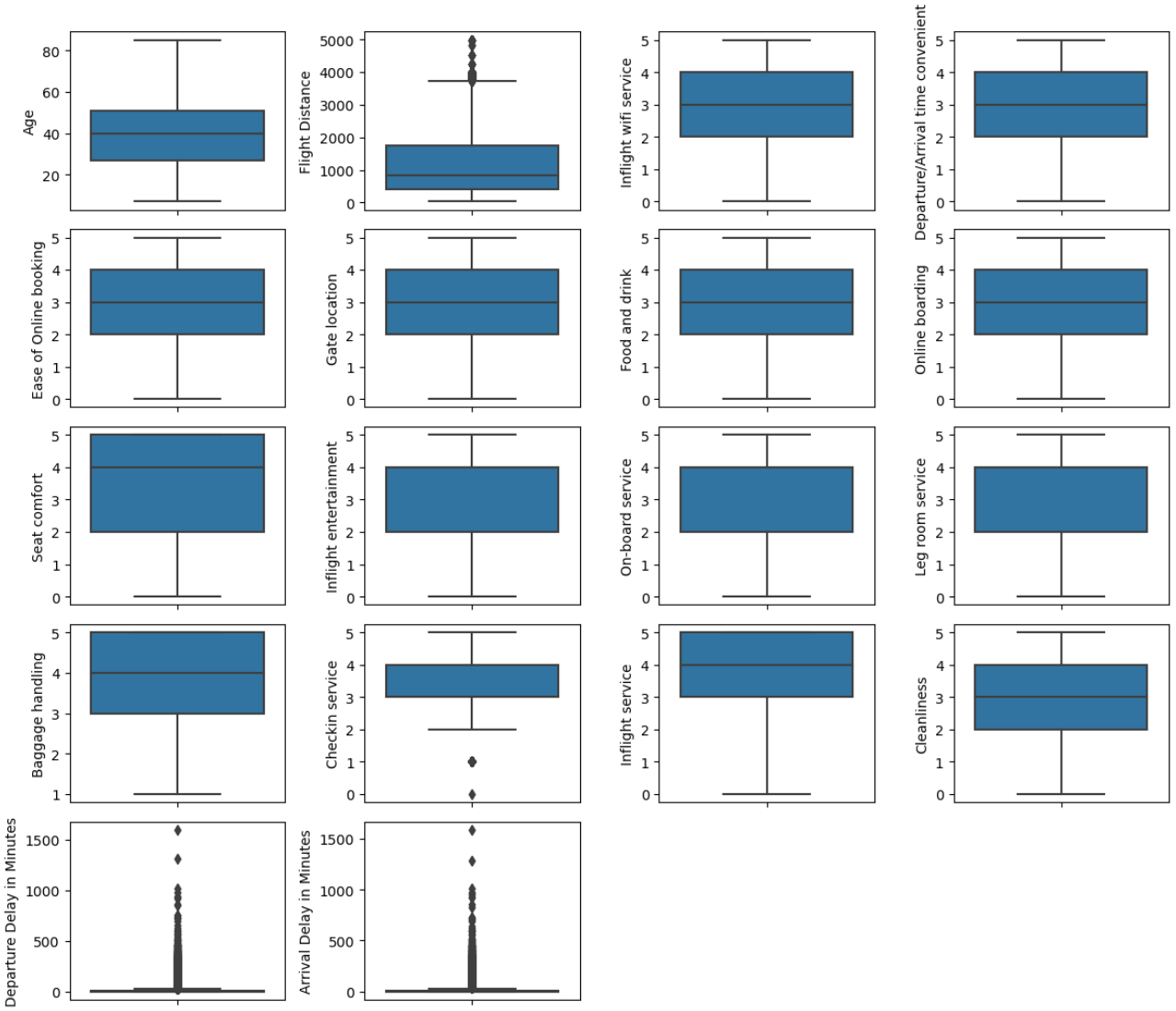
```
# Looking for outliers
numer_features = df_train.select_dtypes(exclude=['object'])
numer_features.columns
```

Out[109]:

```
Index(['Age', 'Flight Distance', 'Inflight wifi service',
       'Departure/Arrival time convenient', 'Ease of Online booking',
       'Gate location', 'Food and drink', 'Online boarding', 'Seat comfort',
       'Inflight entertainment', 'On-board service', 'Leg room service',
       'Baggage handling', 'Checkin service', 'Inflight service',
       'Cleanliness', 'Departure Delay in Minutes',
       'Arrival Delay in Minutes'],
      dtype='object')
```

In [110…

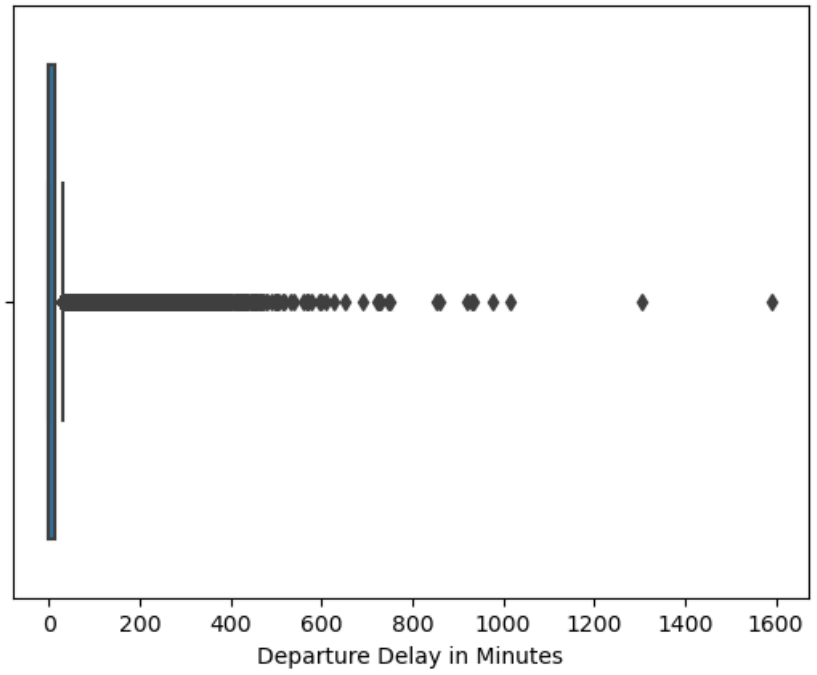
```
fig = plt.figure(figsize=(12,18))
for i in range(len(numer_features.columns)):
    fig.add_subplot(9,4,i+1)
    sns.boxplot(y=numer_features.iloc[:,i])

plt.tight_layout()
plt.show()
```




Here and from the descriptives statistics table, we can see that there are large values for the Departure Delay in Minutes and Arrival Delay in Minutes: 1592 and 1584 respectively.

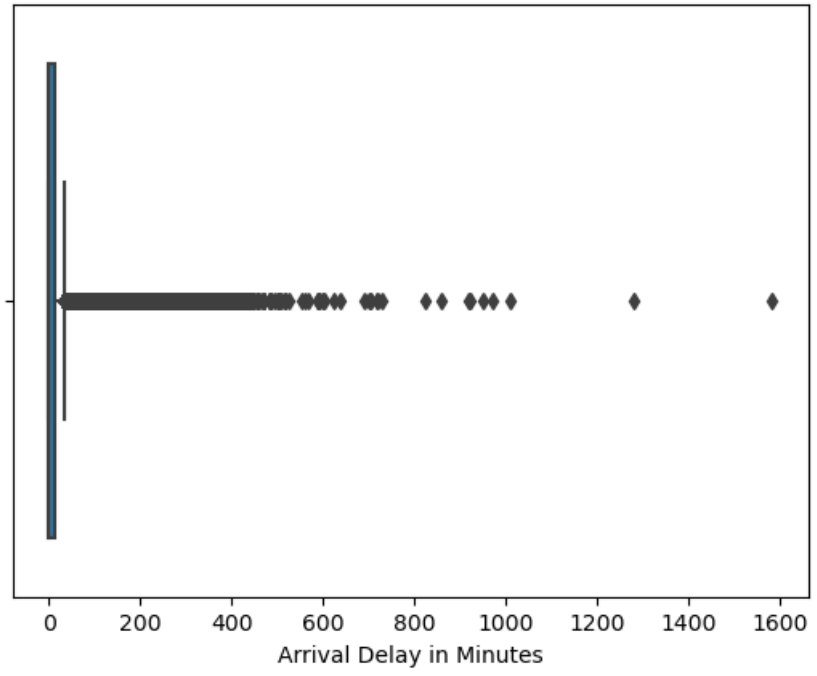
In [111…
```
sns.boxplot(x=df_train['Departure Delay in Minutes'])
```

Out[111]:
```
<AxesSubplot:xlabel='Departure Delay in Minutes'>
```




In [112…
```
sns.boxplot(x=df_train['Arrival Delay in Minutes'])
```

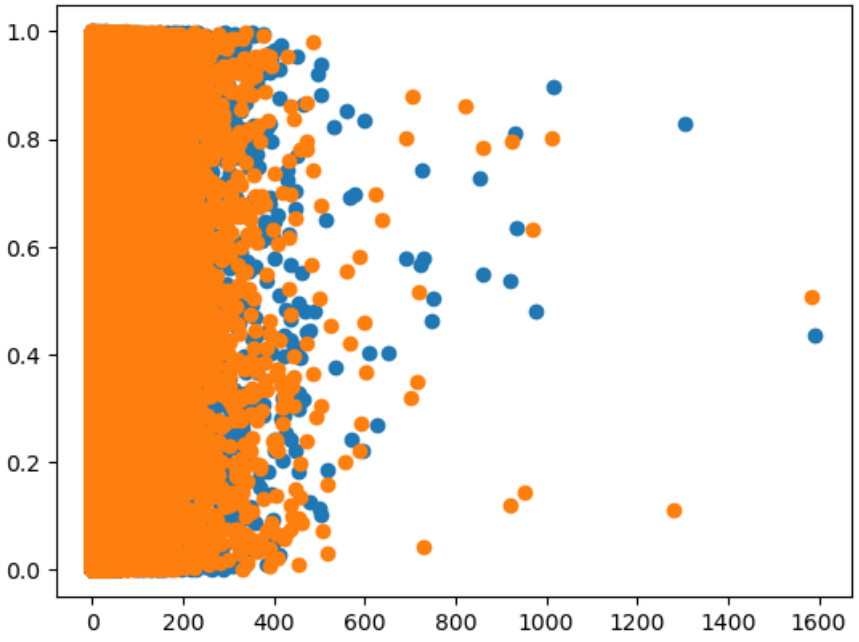
Out[112]:
```
<AxesSubplot:xlabel='Arrival Delay in Minutes'>
```




The boxplots above shows that the two values (1592 and 1584) are significantly larger than the rest of values. Both will be removed.

In [113…
```
plt.scatter(df_train['Departure Delay in Minutes'], np.random.rand(df_train.shape[0]))
plt.scatter(df_train['Arrival Delay in Minutes'], np.random.rand(df_train.shape[0]))
```

Out[113]:
```
<matplotlib.collections.PathCollection at 0x1d8ecfb8ac0>
```




In [114…
```
df_train.loc[df_train['Departure Delay in Minutes'] > 1200]
df_train.loc[df_train['Arrival Delay in Minutes'] > 1200]
```

Out[114]:

| id | Gender | Customer Type | Age | Type of Travel | Class | Flight Distance | Inflight wifi service | Departure/Arrival time convenient | Ease of Online booking | Gate location | ... | Inflight entertainment | bo ser |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **69661** | Male | Loyal Customer | 32 | Business travel | Business | 2916 | 2 | 2 | 2 | 2 | ... | 3 | |
| **73471** | Female | Loyal Customer | 47 | Personal Travel | Eco | 1120 | 2 | 2 | 2 | 3 | ... | 2 | |

2 rows × 23 columns

In [115…
```
print("The data shape is: {}".format(df_train.shape))
```

```
The data shape is: (103594, 23)
```

In [116…
```
outliers = df_train[df_train['Arrival Delay in Minutes'] > 1250].index
df_train.drop(outliers, inplace=True)
print("The data shape is: {}".format(df_train.shape))
```

```
The data shape is: (103592, 23)
```

There are 23 columns of data and some of them are categorical.

In [117…
```
# Categorical data
```

In [118…
```
categ_columns = df_train.select_dtypes(include = ['object'])
unique_values = categ_columns.nunique(dropna = False)
print(unique_values)
```

```
Gender            2
Customer Type     2
Type of Travel    2
Class             3
satisfaction      2
dtype: int64
```

There are 5 categorical columns: Gender, Customer Type, Type of Travel and Satisfaction contains 2 possible values, and Class contains 3 possible values.

In [ ]:
```
```