Introduction:

This report represents the third assignment completed during my Data Science Internship. The aim of this task was to apply advanced data cleaning and preprocessing techniques to a large Airline Passenger Satisfaction dataset. The objective was to identify data inconsistencies and transform the dataset into an analysis-ready format using Python.

LIBRARIES IMPORT

[1]

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

DATASET LOAD KARO

```
df = pd.read_csv("/content/train.csv")
df.head()
```

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling	Checl serv
0	0	70172	Male	Loyal Customer	13	Personal Travel	Eco Plus	460	3	4	...	5	4	3	4	
1	1	5047	Male	disloyal Customer	25	Business travel	Business	235	3	2	...	1	1	5	3	
2	2	110028	Female	Loyal Customer	26	Business travel	Business	1142	2	2	...	5	4	3	4	
3	3	24026	Female	Loyal Customer	25	Business travel	Business	562	2	5	...	2	2	5	3	

Dataset Description:

The Airline Passenger Satisfaction dataset contains survey responses from airline customers. It includes features such as gender, age, travel class, flight distance, and delay information. Due to its size and complexity, this dataset was an excellent choice for practicing professional data cleaning techniques

INITIAL DIAGNOSTICS

```
df.shape
df.info()
df.describe(include='all')
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Unnamed: 0                         103904 non-null  int64
 1   id                                 103904 non-null  int64
 2   Gender                             103904 non-null  object
 3   Customer Type                      103904 non-null  object
 4   Age                                103904 non-null  int64
 5   Type of Travel                     103904 non-null  object
 6   Class                              103904 non-null  object
 7   Flight Distance                    103904 non-null  int64
 8   Inflight wifi service              103904 non-null  int64
 9   Departure/Arrival time convenient  103904 non-null  int64
```

```
 22  Departure Delay in Minutes         103904 non-null  int64
 23  Arrival Delay in Minutes           103594 non-null  float64
 24  satisfaction                       103904 non-null  object
dtypes: float64(1), int64(19), object(5)
memory usage: 19.8+ MB
```

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Infli entertainm
count	103904.000000	103904.000000	103904	103904	103904.000000	103904	103904	103904.000000	103904.000000	103904.000000	...	103904.00C
unique	NaN	NaN	2	2	NaN	2	3	NaN	NaN	NaN	...	I
top	NaN	NaN	Female	Loyal Customer	NaN	Business travel	Business	NaN	NaN	NaN	...	I
freq	NaN	NaN	52727	84923	NaN	71655	49665	NaN	NaN	NaN	...	I
mean	51951.500000	64924.210502	NaN	NaN	39.379706	NaN	NaN	1189.448375	2.729683	3.060296	...	3.358
std	29994.645522	37463.812252	NaN	NaN	15.114964	NaN	NaN	997.147281	1.327829	1.525075	...	1.332
min	0.000000	1.000000	NaN	NaN	7.000000	NaN	NaN	31.000000	0.000000	0.000000	...	0.00C
25%	25975.750000	32533.750000	NaN	NaN	27.000000	NaN	NaN	414.000000	2.000000	2.000000	...	2.00C

Missing Value Analysis:

Missing values were identified in several columns including delay-related features. To better understand their distribution, tabular summaries and heatmap visualization were used.

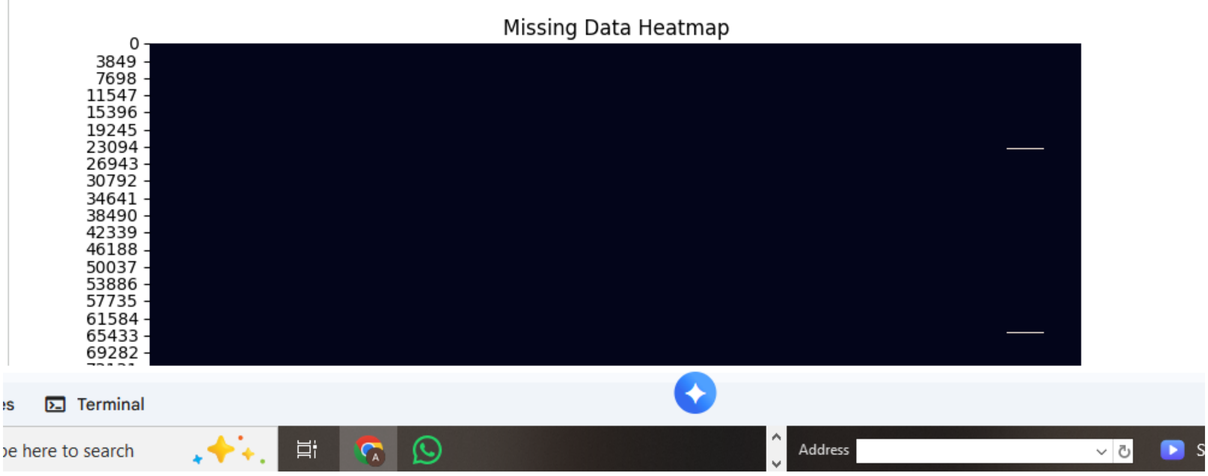
MISSING VALUES ANALYSIS

```
df.isnull().sum()
```

	0
Unnamed: 0	0
id	0
Gender	0
Customer Type	0
Age	0
Type of Travel	0
Class	0
Flight Distance	0
Inflight wifi service	0

HeatMap

```
plt.figure(figsize=(10,5))
sns.heatmap(df.isnull(), cbar=False)
plt.title("Missing Data Heatmap")
plt.show()
```



Missing Value Imputation:

Two imputation techniques were applied. Statistical imputation using mean and mode was used for suitable columns. Additionally, KNN Imputation was implemented as an advanced method. After applying these techniques, the dataset was verified and all missing values were successfully handled.

IMPUTATION TECHNIQUES

Method 1 – Mean / Mode

```
df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].mean(), inplace=True)
df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
```

```
/tmp/ipython-input-3222318373.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behave

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst


  df['Arrival Delay in Minutes'].fillna(df['Arrival Delay in Minutes'].mean(), inplace=True)
/tmp/ipython-input-3222318373.py:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an in
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behave

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inst


  df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
```

Method 2 – KNN Imputation

[7]

```
from sklearn.impute import KNNImputer

imputer = KNNImputer(n_neighbors=5)
df[['Departure Delay in Minutes']] = imputer.fit_transform(df[['Departure Delay in Minutes']])
```

Data Type Correction:

Necessary data type corrections were applied to ensure all numeric features were properly formatted. The date-related columns were parsed into datetime format for better usability.

DATA TYPE CORRECTION

```
df['Flight Distance'] = pd.to_numeric(df['Flight Distance'], errors='coerce')
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103904 entries, 0 to 103903
Data columns (total 25 columns):
 #   Column                             Non-Null Count   Dtype
---  ------                             --------------   -----
 0   Unnamed: 0                         103904 non-null  int64
 1   id                                 103904 non-null  int64
 2   Gender                             103904 non-null  object
 3   Customer Type                      103904 non-null  object
 4   Age                                103904 non-null  int64
 5   Type of Travel                     103904 non-null  object
 6   Class                              103904 non-null  object
 7   Flight Distance                    103904 non-null  int64
 8   Inflight wifi service              103904 non-null  int64
 9   Departure/Arrival time convenient  103904 non-null  int64
 10  Ease of Online booking             103904 non-null  int64
 11  Gate location                      103904 non-null  int64
```

Outlier Detection & Treatment:

Outliers in the Flight Distance column were detected using the Interquartile Range (IQR) method. Extreme values were removed in a documented and reproducible way to enhance dataset reliability.

OUTLIER DETECTION (IQR)&OUTLIER REMOVAL

[9]

```
Q1 = df['Flight Distance'].quantile(0.25)
Q3 = df['Flight Distance'].quantile(0.75)

IQR = Q3 - Q1
lower = Q1 - 1.5 * IQR
upper = Q3 + 1.5 * IQR

df = df[(df['Flight Distance'] >= lower) & (df['Flight Distance'] <= upper)]
df.shape
```

```
(101613, 25)
```

Final Clean Dataset:

FINAL DATA CHECK

[10]

```
df.head()
df.tail()
```

	Unnamed: 0	id	Gender	Customer Type	Age	Type of Travel	Class	Flight Distance	Inflight wifi service	Departure/Arrival time convenient	...	Inflight entertainment	On-board service	Leg room service	Baggage handling
103899	103899	94171	Female	disloyal Customer	23	Business travel	Eco	192	2	1	...	2	3	1	4
103900	103900	73097	Male	Loyal Customer	49	Business travel	Business	2347	4	4	...	5	5	5	5
103901	103901	68825	Male	disloyal Customer	30	Business travel	Business	1995	1	1	...	4	3	2	4
103902	103902	54173	Female	disloyal Customer	22	Business travel	Eco	1000	1	1	...	1	4	5	1

Saved CSV:

Files

..
sample_data
cleaned_airline_satisfaction.csv
train.csv

CLEANED CSV SAVE

[12]

```
df.to_csv("cleaned_airline_satisfaction.csv", index=False)
```