In [25]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import statsmodels.api as sm
import numpy as np
```

In [4]:
```
dataset=pd.read_excel('Flyzy Flight Cancellation.xlsx')
```

In [5]:
```
#PRINTING THE FIRST FEW ROWS OF THE DATASET.
print(dataset.head())
```

```
   Flight ID    Airline  Flight_Distance Origin_Airport Destination_Airport  \
0    7319483  Airline D              475      Airport 3           Airport 2
1    4791965  Airline E              538      Airport 5           Airport 4
2    2991718  Airline C              565      Airport 1           Airport 2
3    4220106  Airline E              658      Airport 5           Airport 3
4    2263008  Airline E              566      Airport 2           Airport 2

   Scheduled_Departure_Time  Day_of_Week  Month Airplane_Type  Weather_Score  \
0                         4            6      1        Type C       0.225122
1                        12            1      6        Type B       0.060346
2                        17            3      9        Type C       0.093920
3                         1            1      8        Type B       0.656750
4                        19            7     12        Type E       0.505211

   Previous_Flight_Delay_Minutes  Airline_Rating  Passenger_Load  \
0                            5.0        2.151974        0.477202
1                           68.0        1.600779        0.159718
2                           18.0        4.406848        0.256803
3                           13.0        0.998757        0.504077
4                            4.0        3.806206        0.019638

   Flight_Cancelled
0                 0
1                 1
2                 0
3                 1
4                 0
```

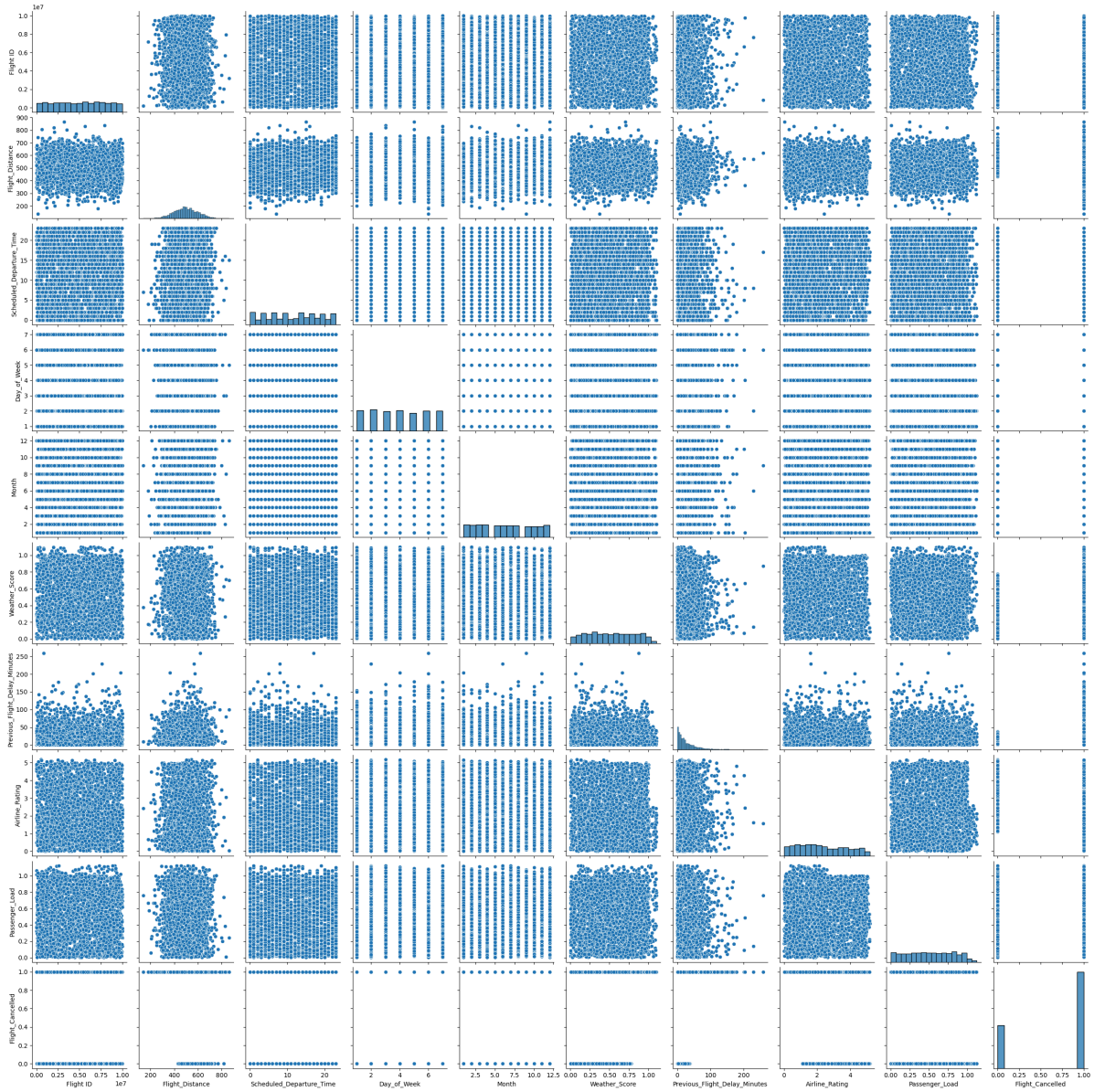
In [6]:
```
#SUMMARY STATISTICS
print(dataset.describe())
```

```
          Flight ID  Flight_Distance  Scheduled_Departure_Time  Day_of_Week  \
count  3.000000e+03      3000.000000               3000.000000  3000.000000
mean   4.997429e+06       498.909333                 11.435000     3.963000
std    2.868139e+06        98.892266                  6.899298     2.016346
min    3.681000e+03       138.000000                  0.000000     1.000000
25%    2.520313e+06       431.000000                  6.000000     2.000000
50%    5.073096e+06       497.000000                 12.000000     4.000000
75%    7.462026e+06       566.000000                 17.000000     6.000000
max    9.999011e+06       864.000000                 23.000000     7.000000

             Month  Weather_Score  Previous_Flight_Delay_Minutes  \
count  3000.000000    3000.000000                    3000.000000
mean      6.381000       0.524023                      26.793383
std       3.473979       0.290694                      27.874733
min       1.000000       0.000965                       0.000000
25%       3.000000       0.278011                       7.000000
50%       6.000000       0.522180                      18.000000
75%       9.000000       0.776323                      38.000000
max      12.000000       1.099246                     259.000000

       Airline_Rating  Passenger_Load  Flight_Cancelled
count     3000.000000     3000.000000       3000.000000
mean         2.317439        0.515885          0.690667
std          1.430386        0.295634          0.462296
min          0.000103        0.001039          0.000000
25%          1.092902        0.265793          0.000000
50%          2.126614        0.517175          1.000000
75%          3.525746        0.770370          1.000000
max          5.189038        1.123559          1.000000
```

In [10]:

```
sns.pairplot(dataset)
plt.show()
```

```
C:\Users\Deviare User\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWar
ning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



In [13]:
```
from sklearn.preprocessing import LabelEncoder
```

In [14]:
```
label_encoder=LabelEncoder()
```

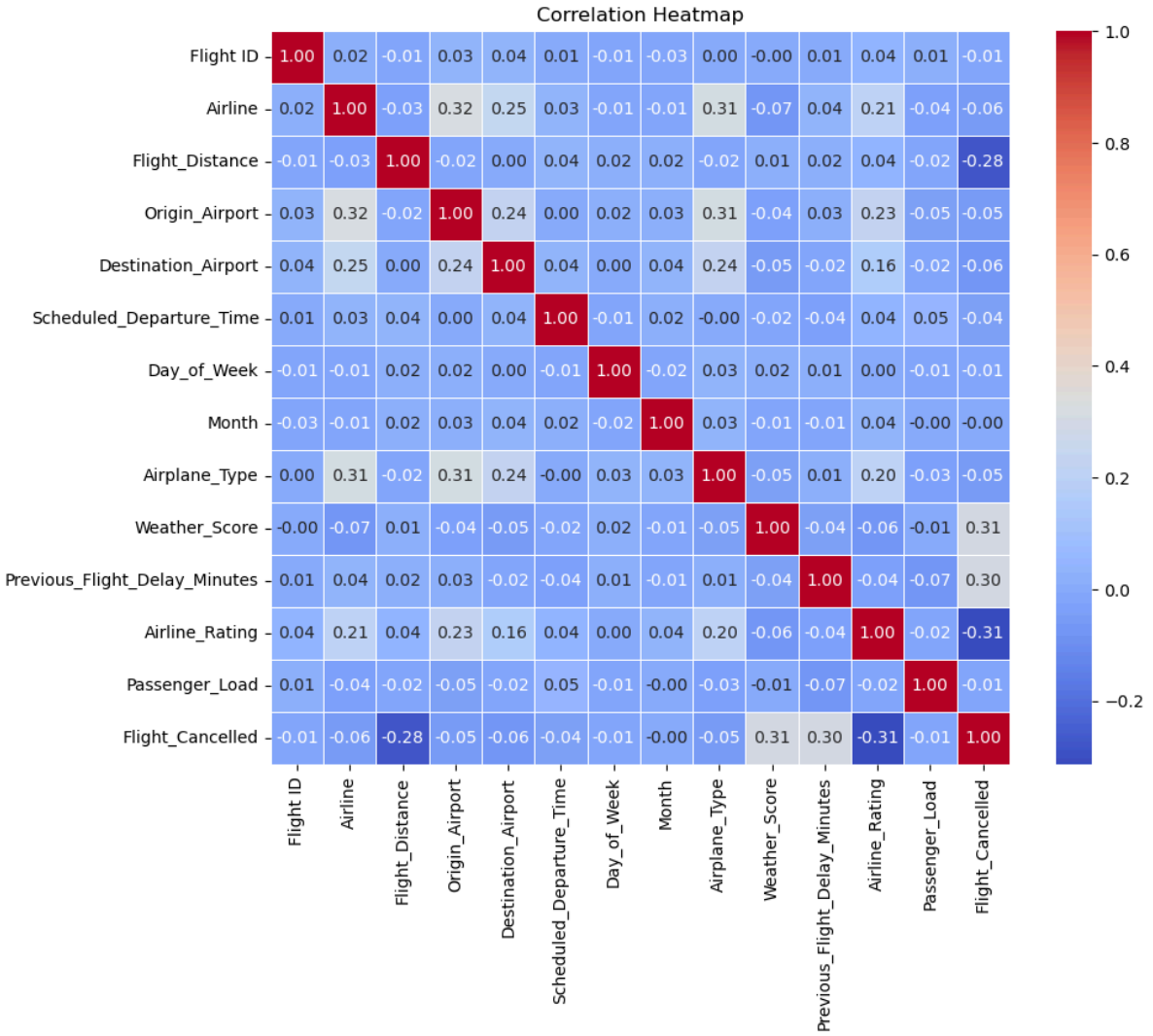
In [15]:
```
dataset['Airline']=label_encoder.fit_transform(dataset['Airline'])
dataset['Origin_Airport']=label_encoder.fit_transform(dataset['Origin_Airport'])
dataset['Destination_Airport']=label_encoder.fit_transform(dataset['Destination_Air
dataset['Airplane_Type']=label_encoder.fit_transform(dataset['Airplane_Type'])
```

In [16]:
```
dataset.dtypes
```

Out[16]:
```
Flight ID                        int64
Airline                          int32
Flight_Distance                  int64
Origin_Airport                   int32
Destination_Airport              int32
Scheduled_Departure_Time         int64
Day_of_Week                      int64
Month                            int64
Airplane_Type                    int32
Weather_Score                  float64
Previous_Flight_Delay_Minutes  float64
Airline_Rating                 float64
Passenger_Load                 float64
Flight_Cancelled                 int64
dtype: object
```

In [18]:
```
#CALCULATING THE CORRELATION MATRIX.
correlation_matrix=dataset.corr()
```

In [19]:
```
#CREATING A HEATMAP.
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=
plt.title('Correlation Heatmap')
plt.show()
```



In [20]:
```
x=dataset[['Flight ID', 'Airline','Flight_Distance', 'Origin_Airport', 'Destinatior
           'Month', 'Airplane_Type', 'Weather_Score', 'Previous_Flight_Delay_Minutes
y=dataset['Flight_Cancelled']
```

In [26]:
```
correlation=np.corrcoef(x['Flight ID'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.009100544102721824
```

In [27]:
```
correlation=np.corrcoef(x['Airline'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.057915220742543655
```

In [28]:
```
correlation=np.corrcoef(x['Flight_Distance'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.2774709629101504
```

In [29]:
```
correlation=np.corrcoef(x['Origin_Airport'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.0499254513188594
```

In [30]:
```
correlation=np.corrcoef(x['Destination_Airport'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.06475308532828992
```

In [31]:
```
correlation=np.corrcoef(x['Scheduled_Departure_Time'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.04373279921720964
```

In [32]:
```
correlation=np.corrcoef(x['Day_of_Week'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.008705376908752012
```

In [33]:
```
correlation=np.corrcoef(x['Month'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.0042421620100930955
```

In [34]:
```
correlation=np.corrcoef(x['Airplane_Type'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.04994233077062425
```

In [35]:
```
correlation=np.corrcoef(x['Weather_Score'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: 0.3057616250531161
```

In [36]:
```
correlation=np.corrcoef(x['Previous_Flight_Delay_Minutes'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: 0.30280464054787787
```

In [37]:
```
correlation=np.corrcoef(x['Airline_Rating'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.3140986375154463
```

In [38]:
```
correlation=np.corrcoef(x['Passenger_Load'], y)[0, 1]
print("Correlation coefficient:", correlation)
```

```
Correlation coefficient: -0.008319756091970044
```

In [39]:
```
from sklearn.linear_model import LinearRegression
```

In [40]:
```
model=LinearRegression()
model.fit(x, y)
print("Intercept:", model.intercept_)
print("Coefficients:", model.coef_)
```

```
Intercept: 1.1322946462844259
Coefficients: [-7.78946167e-11 -1.00440931e-03 -1.28197407e-03  3.22053796e-03
 -1.33395687e-03 -2.33343775e-04 -2.62244908e-03  2.19967335e-03
  3.11880877e-03  4.85104135e-01  5.16096743e-03 -8.89749801e-02
  7.14363056e-03]
```

In [41]:
```
#EVALUATION METRIC : R-SQUARED.
from sklearn.metrics import r2_score
```

In [43]:
```
y_pred=model.predict(x)
r2=r2_score(y, y_pred)
print("R-squared:", r2)
```

```
R-squared: 0.3495804329516544
```

In [44]:
```
#ADDING MORE INDEPENDENT VARIABLES TO ENSURE MY MODEL IS GOOD.
```

In [46]:
```
dataset_2=pd.read_excel("Flyzy Flight Cancellation.xlsx")
```

In [49]:
```
#DEFINING INDEPENDENT VARIABLES.

x=dataset.drop(columns=['Flight ID', 'Airline','Flight_Distance', 'Origin_Airport',
          'Month', 'Airplane_Type', 'Weather_Score', 'Previous_Flight_Delay_Minutes
y=dataset['Flight_Cancelled']
```

In [51]:
```
#SPLITTING THE DATA INTO TRAINING AND TESTING SETS.

x_train, x_test, y_train, y_test=train_test_split(x, y, test_size=0.2, random_state
```

In [52]:
```
#FEEDING THE MODEL.

model=LinearRegression()
model.fit(x_train, y_train)
```

Out[52]:
```
▾ LinearRegression
LinearRegression()
```

In [53]:
```
#PREDICT ON THE TEST SET.

y_pred=model.predict(x_test)
```

In [54]:
```
#EVALUATING THE MODEL'S PERFORMANCE.

r_squared=r2_score(y_test, y_pred)
print("R-squared:", r_squared)
```

```
R-squared: 1.0
```

In [ ]: