In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
plt.style.use('ggplot')
```

In [2]:
```
dataset=pd.read_excel('Flyzy Flight Cancellation.xlsx')
```

In [7]:
```
dataset.shape
```

Out[7]:
```
(3000, 14)
```

In [9]:
```
dataset.head()
```

Out[9]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Airport	Scheduled_Departure_Time
0	7319483	Airline D	475	Airport 3	Airport 2	4
1	4791965	Airline E	538	Airport 5	Airport 4	12
2	2991718	Airline C	565	Airport 1	Airport 2	17
3	4220106	Airline E	658	Airport 5	Airport 3	1
4	2263008	Airline E	566	Airport 2	Airport 2	19

In [10]:
```
type(dataset)
```

Out[10]:
```
pandas.core.frame.DataFrame
```

In [12]:
```
dataset.columns
```

Out[12]:
```
Index(['Flight ID', 'Airline', 'Flight_Distance', 'Origin_Airport',
       'Destination_Airport', 'Scheduled_Departure_Time', 'Day_of_Week',
       'Month', 'Airplane_Type', 'Weather_Score',
       'Previous_Flight_Delay_Minutes', 'Airline_Rating', 'Passenger_Load',
       'Flight_Cancelled'],
      dtype='object')
```

In [14]:
```
dataset.isnull().sum()
```

Out[14]:
```
Flight ID                          0
Airline                            0
Flight_Distance                    0
Origin_Airport                     0
Destination_Airport                0
Scheduled_Departure_Time           0
Day_of_Week                        0
Month                              0
Airplane_Type                      0
Weather_Score                      0
Previous_Flight_Delay_Minutes      0
Airline_Rating                     0
Passenger_Load                     0
Flight_Cancelled                   0
dtype: int64
```

In [15]:
```
from sklearn.preprocessing import LabelEncoder
```

In [16]:
```
label_encoder=LabelEncoder()
```

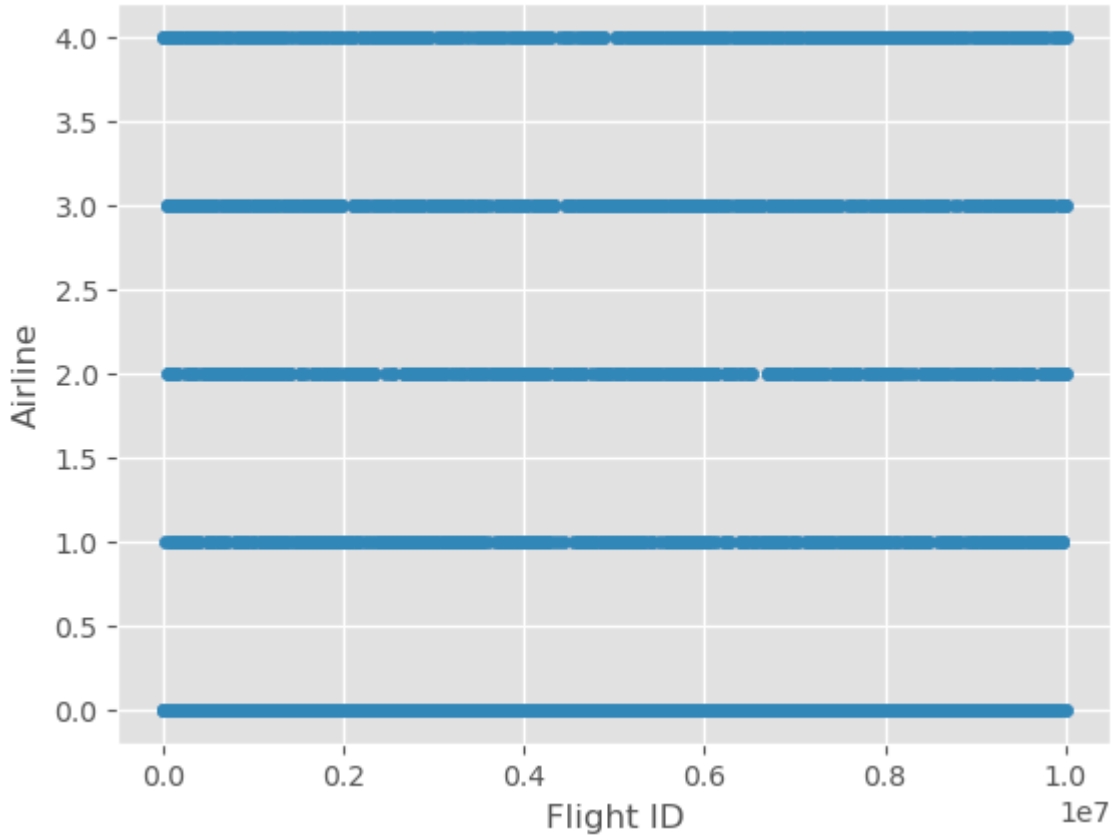
In [17]:
```
dataset['Airline']=label_encoder.fit_transform(dataset['Airline'])
dataset['Origin_Airport']=label_encoder.fit_transform(dataset['Origin_Airport'])
dataset['Destination_Airport']=label_encoder.fit_transform(dataset['Destination_Air
dataset['Airplane_Type']=label_encoder.fit_transform(dataset['Airplane_Type'])
```

In [18]:
```
dataset.dtypes
```

Out[18]:
```
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
```

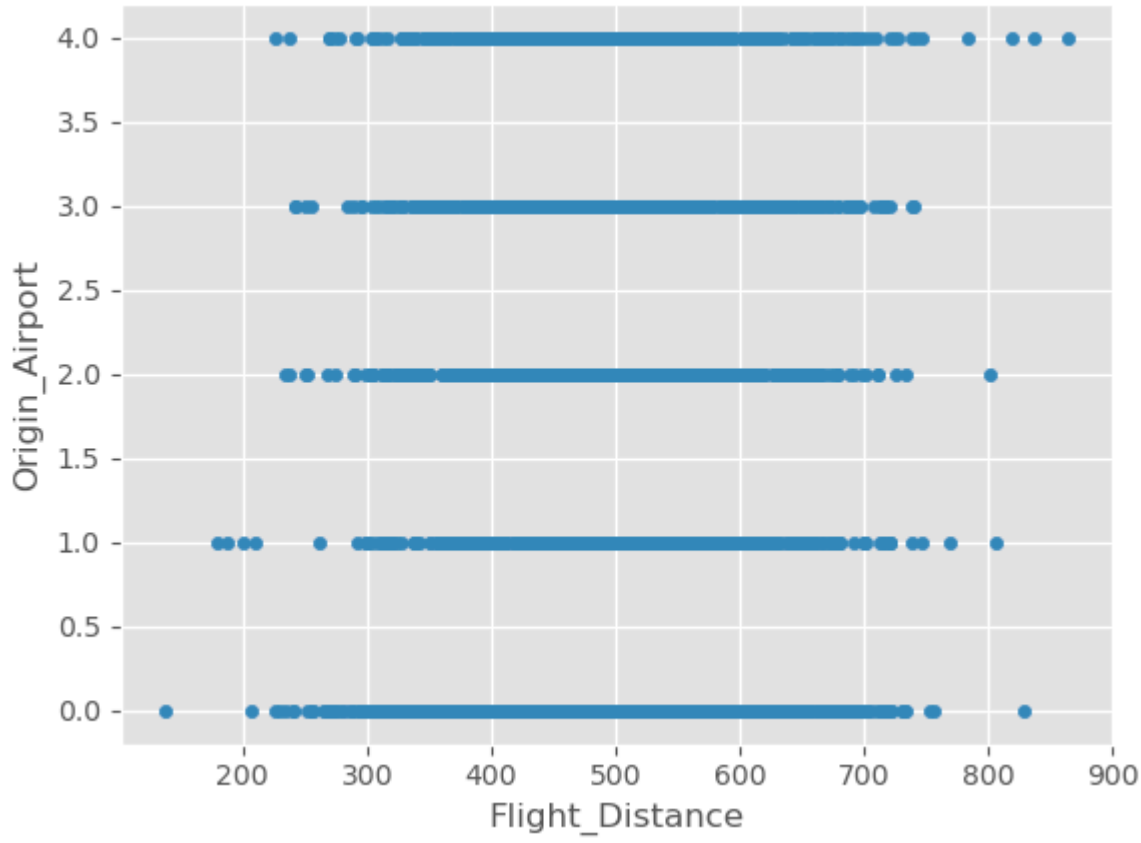
In [19]:
```
dataset.plot(kind='scatter',
             x='Flight ID',
             y='Airline')
```

Out[19]:
```
<Axes: xlabel='Flight ID', ylabel='Airline'>
```



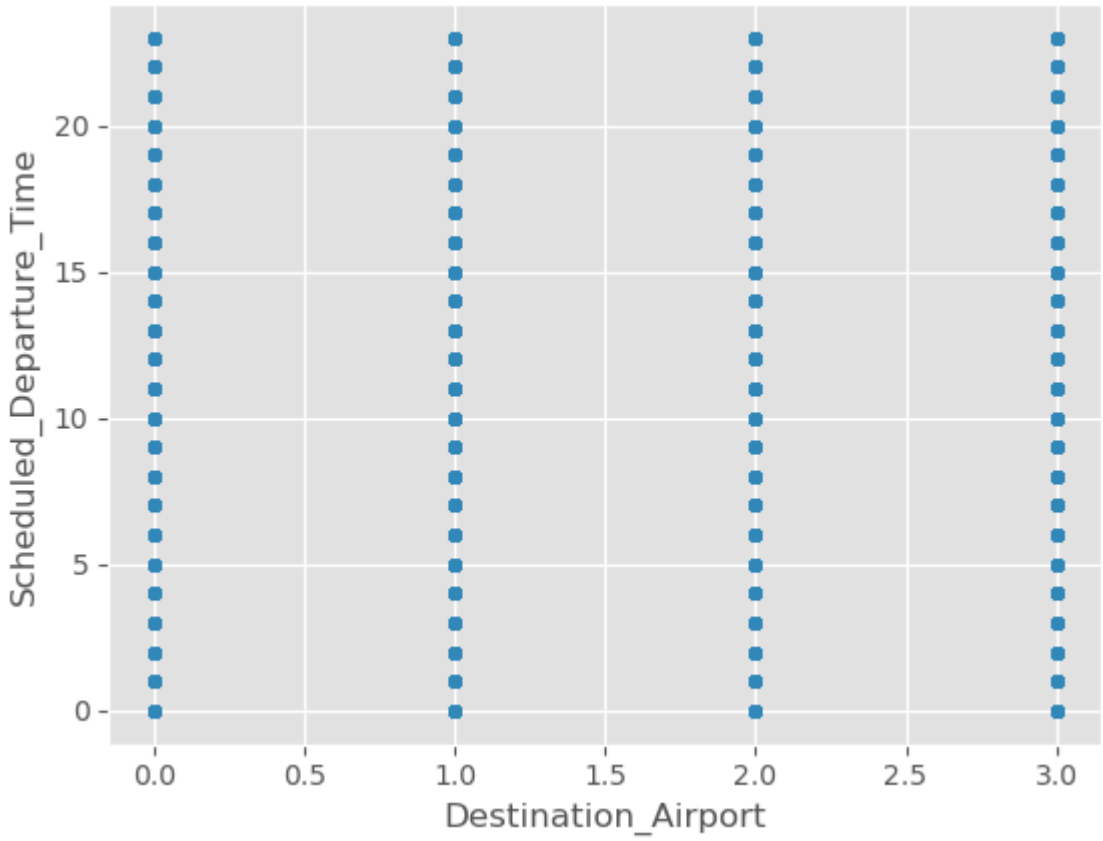
In [20]:
```
dataset.plot(kind='scatter',
             x='Flight_Distance',
             y='Origin_Airport')
```

Out[20]:
```
<Axes: xlabel='Flight_Distance', ylabel='Origin_Airport'>
```



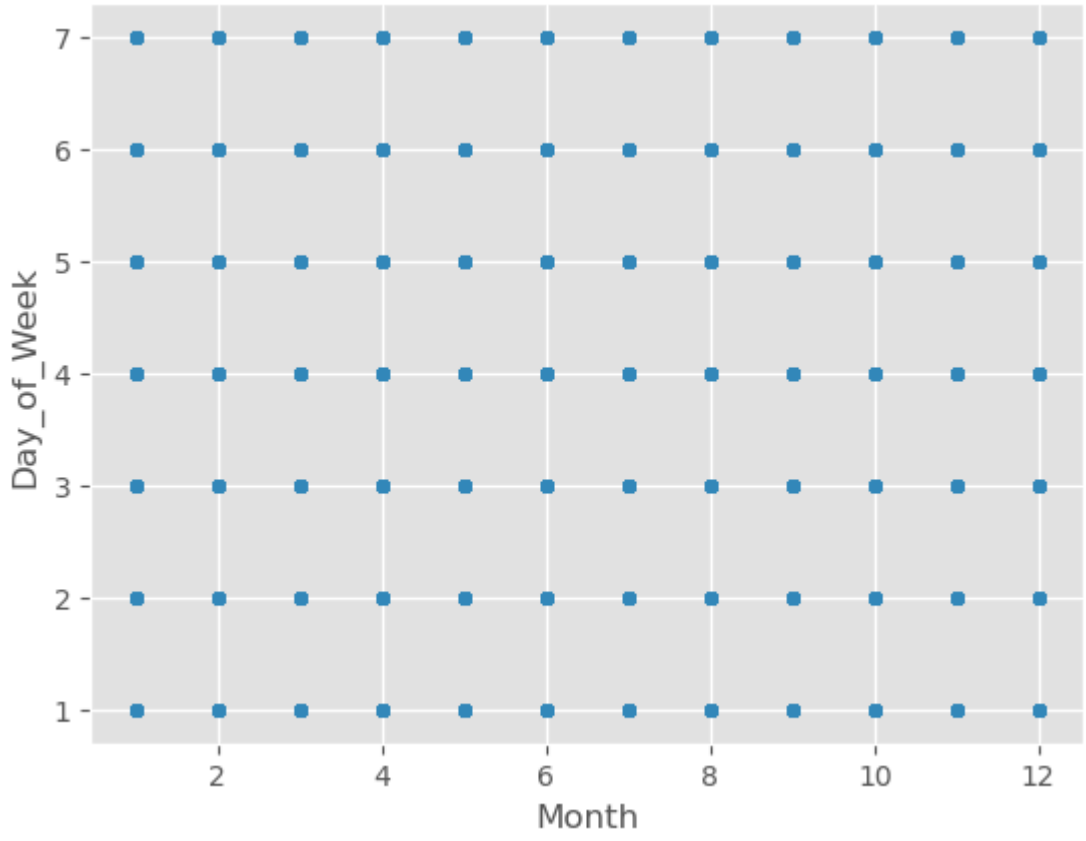
In [23]:
```
dataset.plot(kind='scatter',
             x='Destination_Airport',
             y='Scheduled_Departure_Time')
```

Out[23]:
```
<Axes: xlabel='Destination_Airport', ylabel='Scheduled_Departure_Time'>
```



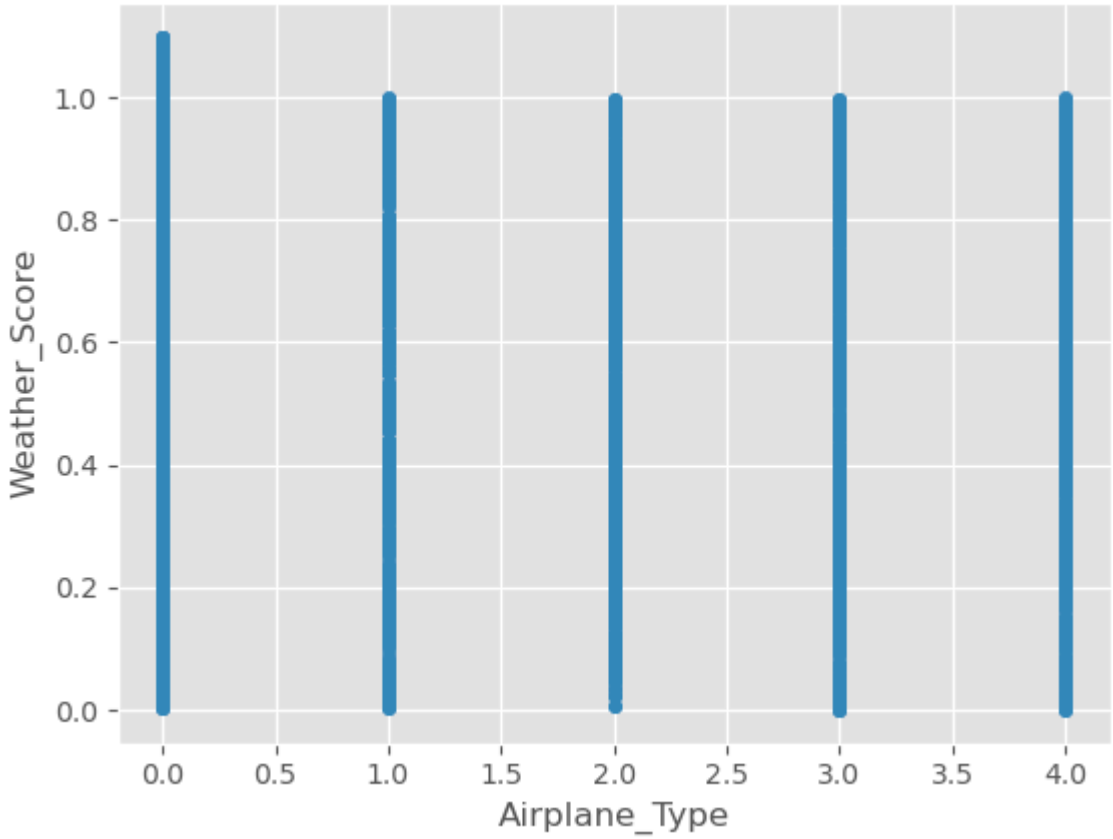
In [24]:
```
dataset.plot(kind='scatter',
             x='Month',
             y='Day_of_Week')
```

Out[24]:
```
<Axes: xlabel='Month', ylabel='Day_of_Week'>
```



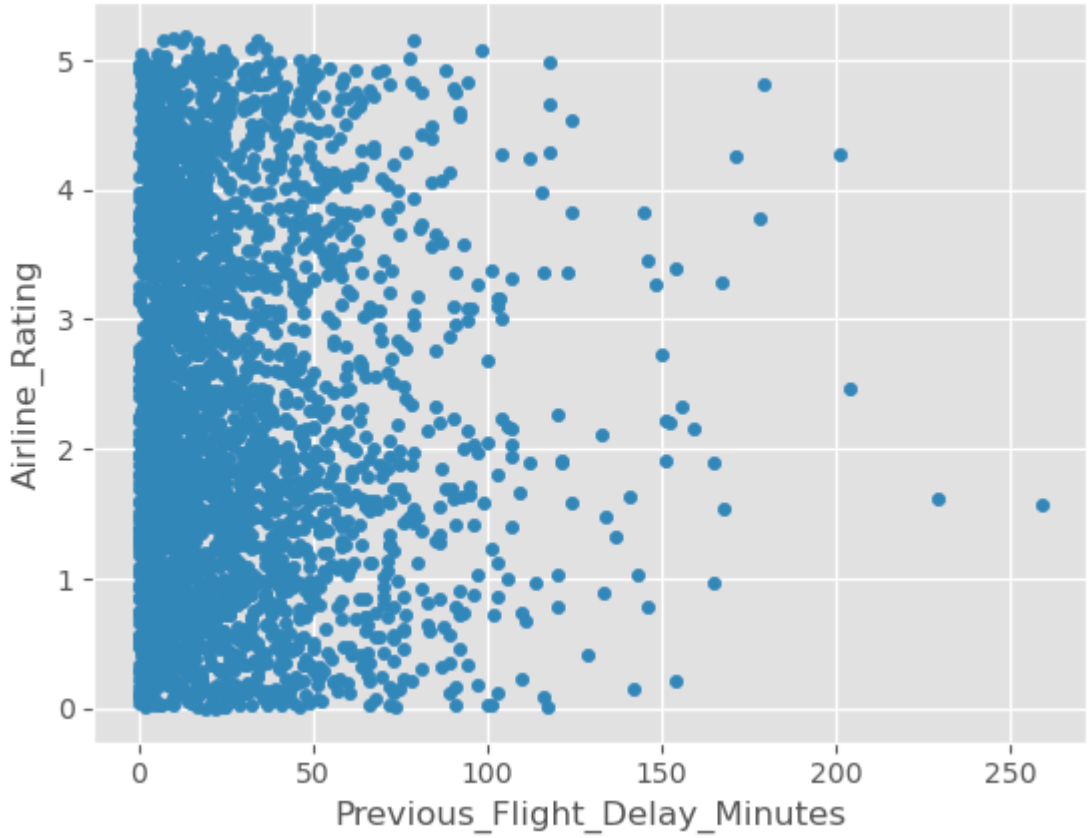
In [25]:
```
dataset.plot(kind='scatter',
             x='Airplane_Type',
             y='Weather_Score')
```

Out[25]:
```
<Axes: xlabel='Airplane_Type', ylabel='Weather_Score'>
```



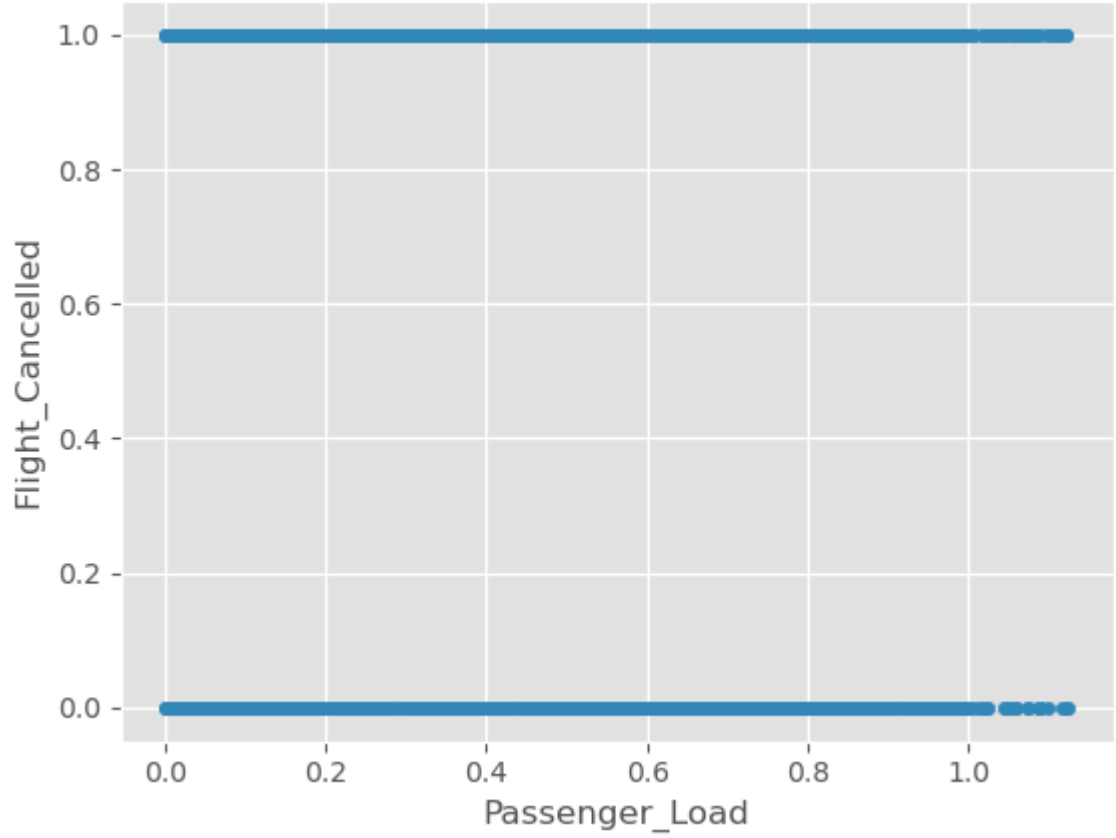
In [26]:
```
dataset.plot(kind='scatter',
             x='Previous_Flight_Delay_Minutes',
             y='Airline_Rating')
```

Out[26]:
```
<Axes: xlabel='Previous_Flight_Delay_Minutes', ylabel='Airline_Rating'>
```



In [27]:
```
dataset.plot(kind='scatter',
             x='Passenger_Load',
             y='Flight_Cancelled')
```

Out[27]:
```
<Axes: xlabel='Passenger_Load', ylabel='Flight_Cancelled'>
```



In [28]:
```
dataset[['Flight ID', 'Airline', 'Flight_Distance', 'Origin_Airport',
       'Destination_Airport', 'Scheduled_Departure_Time', 'Day_of_Week',
```

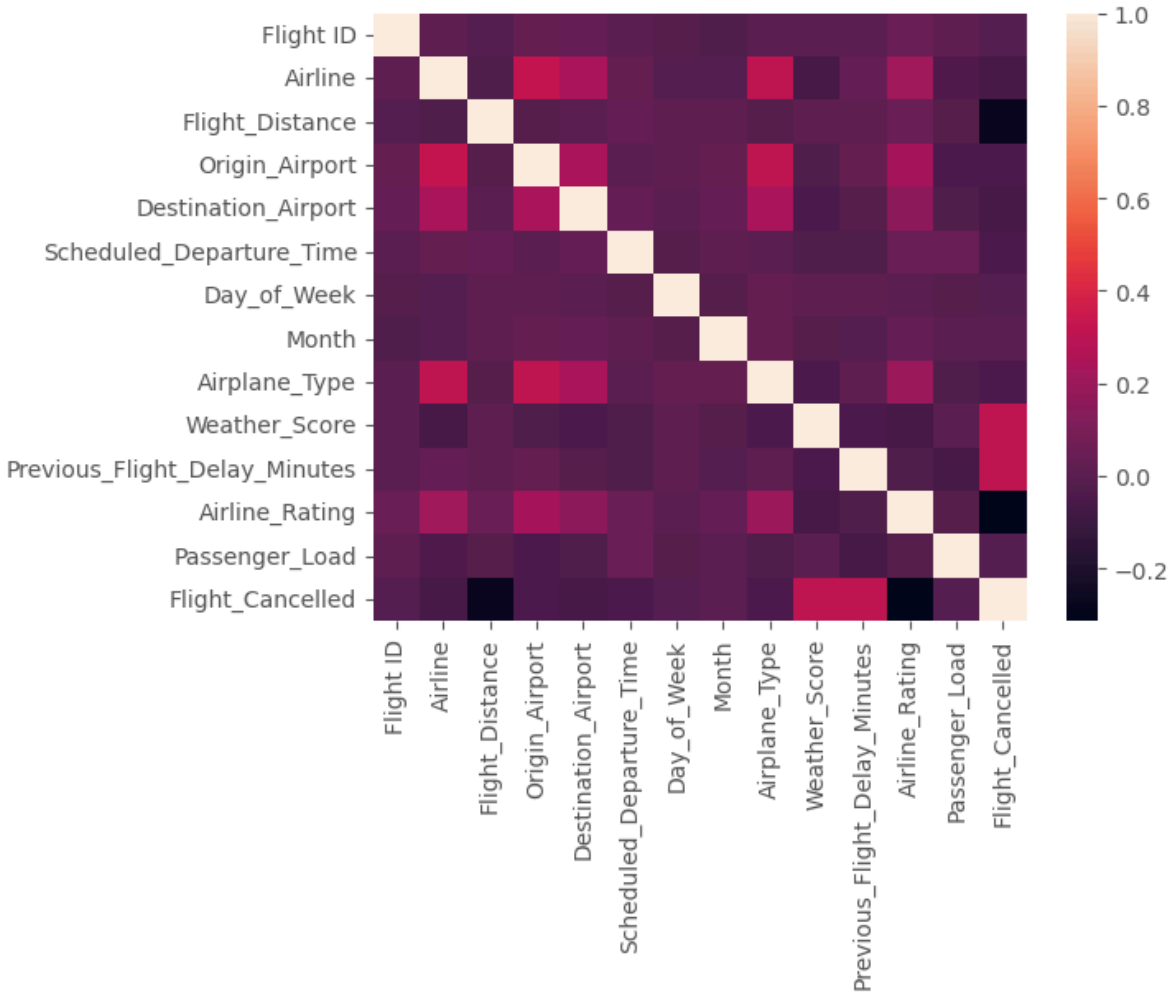
```
        'Month', 'Airplane_Type', 'Weather_Score',
        'Previous_Flight_Delay_Minutes', 'Airline_Rating', 'Passenger_Load',
        'Flight_Cancelled']].dropna().corr()
```

Out[28]:

	Flight ID	Airline	Flight_Distance	Origin_Airport	Destination_Air
Flight ID	1.000000	0.019593	-0.007541	0.031668	0.03
Airline	0.019593	1.000000	-0.027939	0.319073	0.24
Flight_Distance	-0.007541	-0.027939	1.000000	-0.020991	0.00
Origin_Airport	0.031668	0.319073	-0.020991	1.000000	0.23
Destination_Airport	0.036838	0.245179	0.000182	0.237130	1.00
Scheduled_Departure_Time	0.006207	0.031445	0.039727	0.001923	0.03
Day_of_Week	-0.012384	-0.007652	0.024455	0.021073	0.00
Month	-0.025743	-0.008972	0.019573	0.025791	0.03
Airplane_Type	0.000904	0.307505	-0.015445	0.311408	0.23
Weather_Score	-0.002007	-0.066312	0.010139	-0.036558	-0.05
Previous_Flight_Delay_Minutes	0.006172	0.036093	0.018413	0.025523	-0.01
Airline_Rating	0.043170	0.212891	0.042128	0.233298	0.16
Passenger_Load	0.009312	-0.037331	-0.018627	-0.045406	-0.02
Flight_Cancelled	-0.009101	-0.057915	-0.277471	-0.049925	-0.06

In [31]:
```
sns.heatmap(dataset_corr)
```

Out[31]:
```
<Axes: >
```

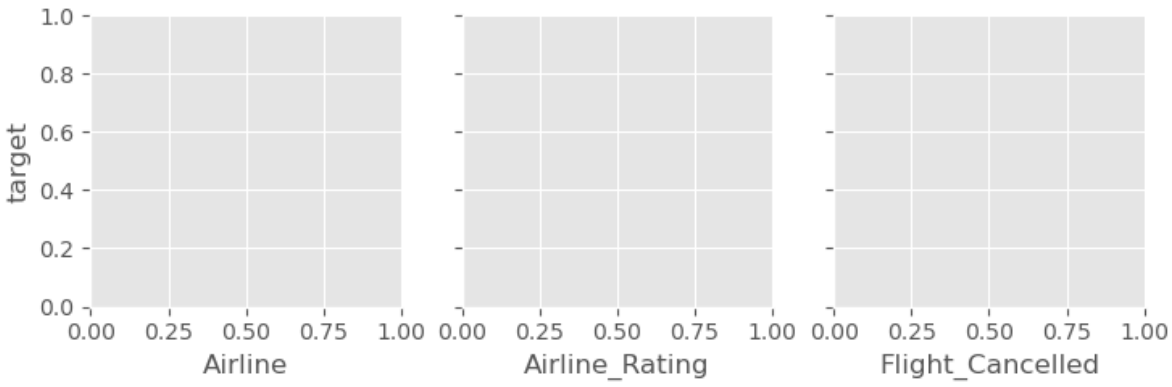


In [33]:
```
# Relationship between features and target variable: Investigating how different fe
```

In [40]:
```
sns.pairplot(dataset, x_vars=['Airline', 'Airline_Rating', 'Flight_Cancelled'], y_
```

```
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)
```

Out[40]:
```
<seaborn.axisgrid.PairGrid at 0x2bc46de2dd0>
```



In [ ]: