In [1]:
```
# Generic inputs for most ML tasks
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor

pd.options.display.float_format = '{:,.2f}'.format

# setup interactive notebook mode
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

from IPython.display import display, HTML
```

In [2]:
```
# fetch data

df = pd.read_excel('Detailed_Statistics_Arrivals_with_all_years.xlsx')
#Actual_output = pd.read_csv('project csv(Apr 12-15)_actuals.csv')

#Train_data.head()
```

In [3]:
```
df.isna().sum()
```

Out[3]:
```
Carrier Code                           1
Date (MM/DD/YYYY)                      2
Flight Number                          2
Tail Number                           59
Origin Airport                         3
Scheduled Arrival Time                 2
Actual Arrival Time                    2
Scheduled Elapsed Time (Minutes)       2
Actual Elapsed Time (Minutes)          2
Arrival Delay (Minutes)                2
Wheels-on Time                         2
Taxi-In time (Minutes)                 2
Delay Carrier (Minutes)                2
Delay Weather (Minutes)                2
Delay National Aviation System (Minutes)    2
Delay Security (Minutes)               2
Delay Late Aircraft Arrival (Minutes)       2
dummy                                  2
Temperature(celsius)                  12
Pressure(mb)                          12
Dew point(celsius)                    12
Wind Speed(m/s)                       12
Wind Direction(degrees)               12
Precipitation(mm)                     12
Snowfall(mm)                          12
Cloud Cover(%)                        12
Humidity(%)                           12
Solar radiaton ((W/m^2))              12
UV Index(0-11+)                       12
Visibility(km)                        12
Weather Description                   12
Day/Night                             12
dtype: int64
```

In [4]:
```
df.isna().sum()
```

Out[4]:
```
Carrier Code                            1
Date (MM/DD/YYYY)                       2
Flight Number                           2
Tail Number                            59
Origin Airport                          3
Scheduled Arrival Time                  2
Actual Arrival Time                     2
Scheduled Elapsed Time (Minutes)        2
Actual Elapsed Time (Minutes)           2
Arrival Delay (Minutes)                 2
Wheels-on Time                          2
Taxi-In time (Minutes)                  2
Delay Carrier (Minutes)                 2
Delay Weather (Minutes)                 2
Delay National Aviation System (Minutes)    2
Delay Security (Minutes)                2
Delay Late Aircraft Arrival (Minutes)       2
dummy                                   2
Temperature(celsius)                   12
Pressure(mb)                           12
Dew point(celsius)                     12
Wind Speed(m/s)                        12
Wind Direction(degrees)                12
Precipitation(mm)                      12
Snowfall(mm)                           12
Cloud Cover(%)                         12
Humidity(%)                            12
Solar radiaton ((W/m^2))               12
UV Index(0-11+)                        12
Visibility(km)                         12
Weather Description                    12
Day/Night                              12
dtype: int64
```

In [5]:
```
df.describe()
```

Out[5]:

	Flight Number	Scheduled Elapsed Time (Minutes)	Actual Elapsed Time (Minutes)	Arrival Delay (Minutes)	Taxi-In time (Minutes)	Delay Carrier (Minutes)	Delay Weather (Minutes)	Delay National Aviation System (Minutes)	Delay Security (Minutes)	Delay Late Aircraft Arrival (Minutes)	...	Dew point(celsius)	Wind Speed(m/s)	Di
count	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	1,984.00	...	1,974.00	1,974.00	
mean	1,353.42	119.47	110.57	9.85	5.54	4.82	1.53	1.70	0.00	7.75	...	5.30	4.28	
std	709.82	39.57	46.23	58.41	3.78	29.85	26.97	8.73	0.00	34.31	...	10.14	2.25	
min	212.00	69.00	0.00	-42.00	0.00	0.00	0.00	0.00	0.00	0.00	...	-22.20	0.00	
25%	607.00	108.00	91.00	-12.00	4.00	0.00	0.00	0.00	0.00	0.00	...	-2.80	2.60	
50%	1,282.00	111.00	103.00	-4.00	5.00	0.00	0.00	0.00	0.00	0.00	...	6.50	4.10	
75%	1,998.00	117.00	115.00	11.00	6.00	0.00	0.00	0.00	0.00	0.00	...	14.10	5.70	
max	2,645.00	202.00	252.00	986.00	70.00	800.00	985.00	240.00	0.00	565.00	...	24.90	14.40	

8 rows × 22 columns

In [6]:
```
df = df.dropna()
```

In [7]:
```
df.head()
```

Out[7]:

	Carrier Code	Date (MM/DD/YYYY)	Flight Number	Tail Number	Origin Airport	Scheduled Arrival Time	Actual Arrival Time	Scheduled Elapsed Time (Minutes)	Actual Elapsed Time (Minutes)	Arrival Delay (Minutes)	...	Wind Direction(degrees)	Precipitation(mi
0	UA	2022-01-01	1,282.00	N4901U	IAD	23:10:00	00:01:00	70.00	76.00	51.00	...	300.00	1.
1	UA	2023-01-01	604.00	N814UA	DEN	14:58:00	14:52:00	193.00	177.00	-6.00	...	250.00	0.
2	UA	2023-01-01	2,488.00	N38458	EWR	23:14:00	23:15:00	75.00	62.00	1.00	...	330.00	0.
3	UA	2023-01-01	2,645.00	N23721	ORD	23:57:00	23:47:00	107.00	100.00	-10.00	...	330.00	0.
4	UA	2022-01-02	1,282.00	N4901U	IAD	23:10:00	23:27:00	70.00	64.00	17.00	...	290.00	0.

5 rows × 32 columns

In [8]:
```
df.dtypes
```

Out[8]:
```
Carrier Code                                object
Date (MM/DD/YYYY)                   datetime64[ns]
Flight Number                              float64
Tail Number                                 object
Origin Airport                              object
Scheduled Arrival Time                      object
Actual Arrival Time                         object
Scheduled Elapsed Time (Minutes)           float64
Actual Elapsed Time (Minutes)              float64
Arrival Delay (Minutes)                    float64
Wheels-on Time                              object
Taxi-In time (Minutes)                     float64
Delay Carrier (Minutes)                    float64
Delay Weather (Minutes)                    float64
Delay National Aviation System (Minutes)   float64
Delay Security (Minutes)                   float64
Delay Late Aircraft Arrival (Minutes)      float64
dummy                                       object
Temperature(celsius)                       float64
Pressure(mb)                               float64
Dew point(celsius)                         float64
Wind Speed(m/s)                            float64
Wind Direction(degrees)                    float64
Precipitation(mm)                          float64
Snowfall(mm)                               float64
Cloud Cover(%)                             float64
Humidity(%)                                float64
Solar radiaton ((W/m^2))                   float64
UV Index(0-11+)                            float64
Visibility(km)                             float64
Weather Description                         object
Day/Night                                   object
dtype: object
```

In [9]:
```
subset_data=df.drop(columns=['Carrier Code','Flight Number','Tail Number','Wheels-on Time','Weather Description'])
```

In [10]:
```
#subset_data = df[['Date (MM/DD/YYYY)','Origin Airport','Flight Number','Scheduled Arrival Time','Arrival Delay (Minut
```

In [11]:
```
subset_data.dtypes
```

Out[11]:
```
Date (MM/DD/YYYY)                   datetime64[ns]
Origin Airport                              object
Scheduled Arrival Time                      object
Actual Arrival Time                         object
Scheduled Elapsed Time (Minutes)           float64
Actual Elapsed Time (Minutes)              float64
Arrival Delay (Minutes)                    float64
Taxi-In time (Minutes)                     float64
Delay Carrier (Minutes)                    float64
Delay Weather (Minutes)                    float64
Delay National Aviation System (Minutes)   float64
Delay Security (Minutes)                   float64
Delay Late Aircraft Arrival (Minutes)      float64
dummy                                       object
Temperature(celsius)                       float64
Pressure(mb)                               float64
Dew point(celsius)                         float64
Wind Speed(m/s)                            float64
Wind Direction(degrees)                    float64
Precipitation(mm)                          float64
Snowfall(mm)                               float64
Cloud Cover(%)                             float64
Humidity(%)                                float64
Solar radiaton ((W/m^2))                   float64
UV Index(0-11+)                            float64
Visibility(km)                             float64
Day/Night                                   object
dtype: object
```

In [12]:
```
# Convert the date column to datetime type

subset_data['Day'] = subset_data['Date (MM/DD/YYYY)'].dt.strftime('%A')
```

In [13]:
```
subset_data=subset_data.drop(columns=['Date (MM/DD/YYYY)'])
```

In [14]:
```
import pandas as pd

# Access the 'Scheduled Arrival Time' column in subset_data DataFrame
time_col = subset_data['Scheduled Arrival Time']

# Extract hours, minutes, and seconds components from datetime.time objects
hours = []
minutes = []


for time_obj in time_col:
    hours.append(time_obj.hour)
    minutes.append(time_obj.minute)


# Create new columns in the DataFrame for hours, minutes, and seconds
subset_data['Scheduled_Arrival_hours'] = hours
subset_data['Scheduled_Arrival_minutes'] = minutes
```

In [15]:
```
time_col = subset_data['Actual Arrival Time']

# Extract hours, minutes, and seconds components from datetime.time objects
hours = []
minutes = []


for time_obj in time_col:
    hours.append(time_obj.hour)
    minutes.append(time_obj.minute)


# Create new columns in the DataFrame for hours, minutes, and seconds
subset_data['Actual_Arrival_hours'] = hours
subset_data['Actual_Arrival_minutes'] = minutes
```

In [16]:
```
subset_data = subset_data.drop(columns = ['Scheduled Arrival Time','Actual Arrival Time','dummy'])
```

In [17]:
```
subset_data.dtypes
```

Out[17]:
```
Origin Airport                           object
Scheduled Elapsed Time (Minutes)        float64
Actual Elapsed Time (Minutes)           float64
Arrival Delay (Minutes)                 float64
Taxi-In time (Minutes)                  float64
Delay Carrier (Minutes)                 float64
Delay Weather (Minutes)                 float64
Delay National Aviation System (Minutes)    float64
Delay Security (Minutes)                float64
Delay Late Aircraft Arrival (Minutes)   float64
Temperature(celsius)                    float64
Pressure(mb)                            float64
Dew point(celsius)                      float64
Wind Speed(m/s)                         float64
Wind Direction(degrees)                 float64
Precipitation(mm)                       float64
Snowfall(mm)                            float64
Cloud Cover(%)                          float64
Humidity(%)                             float64
Solar radiaton ((W/m^2))                float64
UV Index(0-11+)                         float64
Visibility(km)                          float64
Day/Night                                object
Day                                      object
Scheduled_Arrival_hours                   int64
Scheduled_Arrival_minutes                 int64
Actual_Arrival_hours                      int64
Actual_Arrival_minutes                    int64
dtype: object
```

In [18]:
```
subset_data=pd.get_dummies(subset_data, drop_first=True)
```

In [19]:
```
subset_data.dtypes
```

Out[19]:
```
Scheduled Elapsed Time (Minutes)           float64
Actual Elapsed Time (Minutes)              float64
Arrival Delay (Minutes)                    float64
Taxi-In time (Minutes)                     float64
Delay Carrier (Minutes)                    float64
Delay Weather (Minutes)                    float64
Delay National Aviation System (Minutes)   float64
Delay Security (Minutes)                   float64
Delay Late Aircraft Arrival (Minutes)      float64
Temperature(celsius)                       float64
Pressure(mb)                               float64
Dew point(celsius)                         float64
Wind Speed(m/s)                            float64
Wind Direction(degrees)                    float64
Precipitation(mm)                          float64
Snowfall(mm)                               float64
Cloud Cover(%)                             float64
Humidity(%)                                float64
Solar radiaton ((W/m^2))                   float64
UV Index(0-11+)                            float64
Visibility(km)                             float64
Scheduled_Arrival_hours                      int64
Scheduled_Arrival_minutes                    int64
Actual_Arrival_hours                         int64
Actual_Arrival_minutes                       int64
Origin Airport_EWR                           uint8
Origin Airport_IAD                           uint8
Origin Airport_ORD                           uint8
Day/Night_n                                  uint8
Day_Monday                                   uint8
Day_Saturday                                 uint8
Day_Sunday                                   uint8
Day_Thursday                                 uint8
Day_Tuesday                                  uint8
Day_Wednesday                                uint8
dtype: object
```

In [20]:
```
subset_data['Day_Friday']=0
subset_data['Day_Friday'] = subset_data['Day_Friday'].astype('uint8')
```

In [21]:
```
#X_train = subset_data.drop(columns=['Scheduled_Arrival_hours','Scheduled_Arrival_minutes'])
X_train = subset_data
```

In [22]:
```
y_train = subset_data['Scheduled_Arrival_minutes']
```

In [23]:
```
Test_data = pd.read_csv('project csv(Apr 21-24).csv')


Test_data['Arrival Delay (Minutes)'] = 0
#Test_data['Scheduled_Arrival_minutes'] =0
Test_data
subset_Test_data = Test_data[['Date','Day','Origin Airport','Flight Number','Arrival Time']]
```

perature(celsius)	Pressure(mb)	Dew point(celsius)	...	Precipitation(mm)	Snowfall(mm)	Cloud Cover(%)	Humidity(%)	Solar radiaton ((W/m^2))	UV Index(0-11+)	Visibility(km)	Weather Description	Day/Nig
9.60	998.50	4.60	...	0.00	0.00	70.00	71.00	0.00	0.00	22.59	Broken clouds	
27.40	996.00	9.20	...	0.00	0.00	1.00	32.00	118.13	7.50	0.58	Clear Sky	
28.60	992.50	7.30	...	0.00	0.00	100.00	26.00	93.65	0.90	0.69	Overcast clouds	
											Overcast	

In [24]:
```
subset_Test_data.head()
```

Out[24]:

	Date	Day	Origin Airport	Flight Number	Arrival Time
0	4/21/2023	Friday	ORD	UA 3839	10:00 AM
1	4/21/2023	Friday	ORD	UA 3524	4:50 PM
2	4/21/2023	Friday	ORD	UA 538	9:34 PM
3	4/22/2023	Saturday	ORD	UA 3839	10:00 AM
4	4/22/2023	Saturday	ORD	UA 3524	4:50 PM

In [25]:
```
Test_data.isna().sum()
```

Out[25]:
```
Date                                          0
Day                                           0
Origin Airport                                0
Flight Number                                 0
Arrival Time                                  0
Status (Early, On-time, Late, Severly Late)    32
dummy                                         0
Temperature(celsius)                          16
Pressure(mb)                                  16
Dew point(celsius)                            16
Wind Speed(m/s)                               16
Wind Direction(degrees)                       16
Precipitation(mm)                             16
Snowfall(mm)                                  16
Cloud Cover(%)                                16
Humidity(%)                                   16
Solar radiaton ((W/m^2))                      16
UV Index(0-11+)                               16
Visibility(km)                                16
Weather Description                           16
Day/Night                                      0
Arrival Delay (Minutes)                        0
dtype: int64
```

In [26]:
```
Test_data.dtypes
```

Out[26]:
```
Date                                          object
Day                                           object
Origin Airport                                object
Flight Number                                 object
Arrival Time                                  object
Status (Early, On-time, Late, Severly Late)    float64
dummy                                         object
Temperature(celsius)                          float64
Pressure(mb)                                  float64
Dew point(celsius)                            float64
Wind Speed(m/s)                               float64
Wind Direction(degrees)                       float64
Precipitation(mm)                             float64
Snowfall(mm)                                  float64
Cloud Cover(%)                                float64
Humidity(%)                                   float64
Solar radiaton ((W/m^2))                      float64
UV Index(0-11+)                               float64
Visibility(km)                                float64
Weather Description                           object
Day/Night                                     object
Arrival Delay (Minutes)                        int64
dtype: object
```

In [27]:
```
Test_data = Test_data.drop(columns = ['Flight Number','Date','dummy','Weather Description'])
```

In [28]:
```
Test_data.dtypes
```

Out[28]:
```
Day                                             object
Origin Airport                                  object
Arrival Time                                    object
Status (Early, On-time, Late, Severly Late)    float64
Temperature(celsius)                           float64
Pressure(mb)                                   float64
Dew point(celsius)                             float64
Wind Speed(m/s)                                float64
Wind Direction(degrees)                        float64
Precipitation(mm)                              float64
Snowfall(mm)                                   float64
Cloud Cover(%)                                 float64
Humidity(%)                                    float64
Solar radiaton ((W/m^2))                       float64
UV Index(0-11+)                                float64
Visibility(km)                                 float64
Day/Night                                       object
Arrival Delay (Minutes)                          int64
dtype: object
```

In [29]:
```
Test_data['Scheduled Arrival Time'] = Test_data["Arrival Time"]
Test_data=Test_data.drop(columns=["Arrival Time"])
```

In [30]:
```
from datetime import datetime

# Access the 'Scheduled Arrival Time' column in subset_data DataFrame
time_str = Test_data['Scheduled Arrival Time']

# Use the .apply() method to apply datetime.strptime() to each element of the Series
time_col = time_str.apply(lambda x: datetime.strptime(x, "%I:%M %p").time())

# Extract hours, minutes, and seconds components from datetime.time objects
hours = []
minutes = []

for time_obj in time_col:
    hours.append(time_obj.hour)
    minutes.append(time_obj.minute)

# Create new columns in the DataFrame for hours, minutes, and seconds
Test_data['Scheduled_Arrival_hours'] = hours
Test_data['Scheduled_Arrival_minutes'] = minutes
```

In [31]:
```
Test_data = Test_data.drop(columns=['Scheduled Arrival Time'])
```

In [32]:
```
Test_data
```

Out[32]:

	Day	Origin Airport	Status (Early, On-time, Late, Severly Late)	Temperature(celsius)	Pressure(mb)	Dew point(celsius)	Wind Speed(m/s)	Wind Direction(degrees)	Precipitation(mm)	Snowfall(mm)
0	Friday	ORD	NaN	9.60	998.50	4.60	3.18	61.00	0.00	0.00
1	Friday	ORD	NaN	27.40	996.00	9.20	4.19	194.00	0.00	0.00
2	Friday	ORD	NaN	28.60	992.50	7.30	4.31	280.00	0.00	0.00
3	Saturday	ORD	NaN	11.90	992.50	9.30	2.42	87.00	0.00	0.00
4	Saturday	ORD	NaN	22.40	989.50	10.80	8.16	139.00	0.00	0.00
5	Saturday	ORD	NaN	21.60	987.50	7.00	7.53	137.00	0.00	0.00
6	Sunday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7	Sunday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8	Sunday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	Monday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	Monday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11	Monday	ORD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12	Friday	DEN	NaN	25.40	997.00	10.00	2.66	194.00	0.00	0.00
13	Saturday	DEN	NaN	21.70	990.50	10.80	7.98	147.00	0.00	0.00
14	Sunday	DEN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15	Monday	DEN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16	Friday	EWR	NaN	9.60	998.50	4.60	3.18	61.00	0.00	0.00
17	Friday	EWR	NaN	24.50	993.50	10.40	2.01	335.00	0.00	0.00
18	Saturday	EWR	NaN	11.90	992.50	9.30	2.42	87.00	0.00	0.00
19	Saturday	EWR	NaN	19.90	987.50	8.80	6.76	127.00	0.00	0.00
20	Sunday	EWR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
21	Sunday	EWR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
22	Monday	EWR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
23	Monday	EWR	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
24	Friday	IAD	NaN	18.80	997.50	9.80	1.34	90.00	0.00	0.00
25	Friday	IAD	NaN	29.80	994.00	10.30	5.00	176.00	0.00	0.00
26	Saturday	IAD	NaN	17.40	991.50	9.90	7.47	154.00	0.00	0.00
27	Saturday	IAD	NaN	23.10	988.00	7.60	9.72	141.00	0.00	0.00
28	Sunday	IAD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
29	Sunday	IAD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
30	Monday	IAD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
31	Monday	IAD	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

In [33]:
```
Test_data=pd.get_dummies(Test_data, drop_first=True)
Test_data.dtypes
```

Out[33]:
```
Status (Early, On-time, Late, Severly Late)    float64
Temperature(celsius)                           float64
Pressure(mb)                                   float64
Dew point(celsius)                             float64
Wind Speed(m/s)                                float64
Wind Direction(degrees)                        float64
Precipitation(mm)                              float64
Snowfall(mm)                                   float64
Cloud Cover(%)                                 float64
Humidity(%)                                    float64
Solar radiaton ((W/m^2))                       float64
UV Index(0-11+)                                float64
Visibility(km)                                 float64
Arrival Delay (Minutes)                          int64
Scheduled_Arrival_hours                          int64
Scheduled_Arrival_minutes                        int64
Day_Monday                                       uint8
Day_Saturday                                     uint8
Day_Sunday                                       uint8
Origin Airport_EWR                               uint8
Origin Airport_IAD                               uint8
Origin Airport_ORD                               uint8
Day/Night_n                                      uint8
dtype: object
```

In [34]:
```
Test_data['Day_Monday'] =0
Test_data['Day_Tuesday'] =0
Test_data['Day_Friday'] =0
Test_data['Day_Sunday'] =0

Test_data['Day_Monday'] =Test_data['Day_Monday'].astype('uint8')
Test_data['Day_Tuesday'] = Test_data['Day_Tuesday'].astype('uint8')
Test_data['Day_Friday'] = Test_data['Day_Friday'].astype('uint8')
Test_data['Day_Sunday'] = Test_data['Day_Sunday'].astype('uint8')
```

In [35]:
```
Test_data.dtypes
```

Out[35]:
```
Status (Early, On-time, Late, Severly Late)    float64
Temperature(celsius)                           float64
Pressure(mb)                                   float64
Dew point(celsius)                             float64
Wind Speed(m/s)                                float64
Wind Direction(degrees)                        float64
Precipitation(mm)                              float64
Snowfall(mm)                                   float64
Cloud Cover(%)                                 float64
Humidity(%)                                    float64
Solar radiaton ((W/m^2))                       float64
UV Index(0-11+)                                float64
Visibility(km)                                 float64
Arrival Delay (Minutes)                          int64
Scheduled_Arrival_hours                          int64
Scheduled_Arrival_minutes                        int64
Day_Monday                                       uint8
Day_Saturday                                     uint8
Day_Sunday                                       uint8
Origin Airport_EWR                               uint8
Origin Airport_IAD                               uint8
Origin Airport_ORD                               uint8
Day/Night_n                                      uint8
Day_Tuesday                                      uint8
Day_Friday                                       uint8
dtype: object
```

In [36]:
```
X_test = Test_data
```

In [37]:
```
len(X_train.columns)
len(X_test.columns)
```

Out[37]:
```
36
```

Out[37]:
```
25
```

In [38]:
```
X_train.columns
X_test.columns
```

Out[38]:
```
Index(['Scheduled Elapsed Time (Minutes)', 'Actual Elapsed Time (Minutes)',
       'Arrival Delay (Minutes)', 'Taxi-In time (Minutes)',
       'Delay Carrier (Minutes)', 'Delay Weather (Minutes)',
       'Delay National Aviation System (Minutes)', 'Delay Security (Minutes)',
       'Delay Late Aircraft Arrival (Minutes)', 'Temperature(celsius)',
       'Pressure(mb)', 'Dew point(celsius)', 'Wind Speed(m/s)',
       'Wind Direction(degrees)', 'Precipitation(mm)', 'Snowfall(mm)',
       'Cloud Cover(%)', 'Humidity(%)', 'Solar radiaton ((W/m^2))',
       'UV Index(0-11+)', 'Visibility(km)', 'Scheduled_Arrival_hours',
       'Scheduled_Arrival_minutes', 'Actual_Arrival_hours',
       'Actual_Arrival_minutes', 'Origin Airport_EWR', 'Origin Airport_IAD',
       'Origin Airport_ORD', 'Day/Night_n', 'Day_Monday', 'Day_Saturday',
       'Day_Sunday', 'Day_Thursday', 'Day_Tuesday', 'Day_Wednesday',
       'Day_Friday'],
      dtype='object')
```

Out[38]:
```
Index(['Status (Early, On-time, Late, Severly Late)', 'Temperature(celsius)',
       'Pressure(mb)', 'Dew point(celsius)', 'Wind Speed(m/s)',
       'Wind Direction(degrees)', 'Precipitation(mm)', 'Snowfall(mm)',
       'Cloud Cover(%)', 'Humidity(%)', 'Solar radiaton ((W/m^2))',
       'UV Index(0-11+)', 'Visibility(km)', 'Arrival Delay (Minutes)',
       'Scheduled_Arrival_hours', 'Scheduled_Arrival_minutes', 'Day_Monday',
       'Day_Saturday', 'Day_Sunday', 'Origin Airport_EWR',
       'Origin Airport_IAD', 'Origin Airport_ORD', 'Day/Night_n',
       'Day_Tuesday', 'Day_Friday'],
      dtype='object')
```

In [39]:
```
Test_data.dtypes
```

Out[39]:
```
Status (Early, On-time, Late, Severly Late)    float64
Temperature(celsius)                           float64
Pressure(mb)                                   float64
Dew point(celsius)                             float64
Wind Speed(m/s)                                float64
Wind Direction(degrees)                        float64
Precipitation(mm)                              float64
Snowfall(mm)                                   float64
Cloud Cover(%)                                 float64
Humidity(%)                                    float64
Solar radiaton ((W/m^2))                       float64
UV Index(0-11+)                                float64
Visibility(km)                                 float64
Arrival Delay (Minutes)                          int64
Scheduled_Arrival_hours                          int64
Scheduled_Arrival_minutes                        int64
Day_Monday                                       uint8
Day_Saturday                                     uint8
Day_Sunday                                       uint8
Origin Airport_EWR                               uint8
Origin Airport_IAD                               uint8
Origin Airport_ORD                               uint8
Day/Night_n                                      uint8
Day_Tuesday                                      uint8
Day_Friday                                       uint8
dtype: object
```

In [40]:
```
X_train.columns
X_test.columns
```

Out[40]:
```
Index(['Scheduled Elapsed Time (Minutes)', 'Actual Elapsed Time (Minutes)',
       'Arrival Delay (Minutes)', 'Taxi-In time (Minutes)',
       'Delay Carrier (Minutes)', 'Delay Weather (Minutes)',
       'Delay National Aviation System (Minutes)', 'Delay Security (Minutes)',
       'Delay Late Aircraft Arrival (Minutes)', 'Temperature(celsius)',
       'Pressure(mb)', 'Dew point(celsius)', 'Wind Speed(m/s)',
       'Wind Direction(degrees)', 'Precipitation(mm)', 'Snowfall(mm)',
       'Cloud Cover(%)', 'Humidity(%)', 'Solar radiaton ((W/m^2))',
       'UV Index(0-11+)', 'Visibility(km)', 'Scheduled_Arrival_hours',
       'Scheduled_Arrival_minutes', 'Actual_Arrival_hours',
       'Actual_Arrival_minutes', 'Origin Airport_EWR', 'Origin Airport_IAD',
       'Origin Airport_ORD', 'Day/Night_n', 'Day_Monday', 'Day_Saturday',
       'Day_Sunday', 'Day_Thursday', 'Day_Tuesday', 'Day_Wednesday',
       'Day_Friday'],
      dtype='object')
```

Out[40]:
```
Index(['Status (Early, On-time, Late, Severly Late)', 'Temperature(celsius)',
       'Pressure(mb)', 'Dew point(celsius)', 'Wind Speed(m/s)',
       'Wind Direction(degrees)', 'Precipitation(mm)', 'Snowfall(mm)',
       'Cloud Cover(%)', 'Humidity(%)', 'Solar radiaton ((W/m^2))',
       'UV Index(0-11+)', 'Visibility(km)', 'Arrival Delay (Minutes)',
       'Scheduled_Arrival_hours', 'Scheduled_Arrival_minutes', 'Day_Monday',
       'Day_Saturday', 'Day_Sunday', 'Origin Airport_EWR',
       'Origin Airport_IAD', 'Origin Airport_ORD', 'Day/Night_n',
       'Day_Tuesday', 'Day_Friday'],
      dtype='object')
```

In [41]:
```
X_train= X_train[[ 'Temperature(celsius)',
       'Pressure(mb)', 'Dew point(celsius)', 'Wind Speed(m/s)',
       'Wind Direction(degrees)', 'Precipitation(mm)', 'Snowfall(mm)',
       'Cloud Cover(%)', 'Humidity(%)', 'Solar radiaton ((W/m^2))',
       'UV Index(0-11+)', 'Visibility(km)', 'Arrival Delay (Minutes)',
       'Scheduled_Arrival_hours', 'Scheduled_Arrival_minutes', 'Day_Monday',
       'Day_Saturday', 'Day_Sunday', 'Origin Airport_EWR',
       'Origin Airport_IAD', 'Origin Airport_ORD', 'Day/Night_n',
       'Day_Tuesday', 'Day_Friday']]
```

In [42]:
```
X_test= X_test.drop(columns=['Status (Early, On-time, Late, Severly Late)'])
```

In [43]:
```
X_train.columns
X_test.columns
```

Out[43]:
```
Index(['Temperature(celsius)', 'Pressure(mb)', 'Dew point(celsius)',
       'Wind Speed(m/s)', 'Wind Direction(degrees)', 'Precipitation(mm)',
       'Snowfall(mm)', 'Cloud Cover(%)', 'Humidity(%)',
       'Solar radiaton ((W/m^2))', 'UV Index(0-11+)', 'Visibility(km)',
       'Arrival Delay (Minutes)', 'Scheduled_Arrival_hours',
       'Scheduled_Arrival_minutes', 'Day_Monday', 'Day_Saturday', 'Day_Sunday',
       'Origin Airport_EWR', 'Origin Airport_IAD', 'Origin Airport_ORD',
       'Day/Night_n', 'Day_Tuesday', 'Day_Friday'],
      dtype='object')
```

Out[43]:
```
Index(['Temperature(celsius)', 'Pressure(mb)', 'Dew point(celsius)',
       'Wind Speed(m/s)', 'Wind Direction(degrees)', 'Precipitation(mm)',
       'Snowfall(mm)', 'Cloud Cover(%)', 'Humidity(%)',
       'Solar radiaton ((W/m^2))', 'UV Index(0-11+)', 'Visibility(km)',
       'Arrival Delay (Minutes)', 'Scheduled_Arrival_hours',
       'Scheduled_Arrival_minutes', 'Day_Monday', 'Day_Saturday', 'Day_Sunday',
       'Origin Airport_EWR', 'Origin Airport_IAD', 'Origin Airport_ORD',
       'Day/Night_n', 'Day_Tuesday', 'Day_Friday'],
      dtype='object')
```

In [44]:
```
len(X_train.columns)
len(X_test.columns)
```

Out[44]:
```
24
```

In [45]:
```
y_train = X_train['Arrival Delay (Minutes)']
X_train = X_train.drop(columns = ['Arrival Delay (Minutes)'])
```

In [46]:
```
y_test = X_test['Arrival Delay (Minutes)']
X_test  = X_test.drop(columns = ['Arrival Delay (Minutes)'])
```

In [47]:
```
X_train
X_test
y_train
y_test
```

Out[47]:

	Temperature(celsius)	Pressure(mb)	Dew point(celsius)	Wind Speed(m/s)	Wind Direction(degrees)	Precipitation(mm)	Snowfall(mm)	Cloud Cover(%)	Humidity(%
0	6.70	991.30	6.00	3.10	300.00	1.00	0.00	100.00	95.0
1	4.40	999.70	0.60	3.60	250.00	0.00	0.00	100.00	76.0
2	3.90	1,000.00	1.60	2.10	330.00	0.00	0.00	100.00	85.0
3	3.90	1,000.00	1.60	2.10	330.00	0.00	0.00	100.00	85.0
4	-4.40	1,002.40	-5.60	4.10	290.00	0.00	0.00	100.00	91.0
...	...	...	...	...	...	...	...	...	.
1977	15.60	1,002.30	6.10	3.60	200.00	0.00	0.00	68.00	53.0
1978	13.90	1,002.40	4.80	4.10	200.00	0.00	0.00	50.00	54.0
1981	7.80	998.33	3.80	2.60	110.00	0.00	0.00	54.00	76.0

In [48]:
```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = pd.DataFrame(sc.fit_transform(X_train), columns = X_train.columns, index = X_train.index)
X_test = pd.DataFrame(sc.transform(X_test), columns = X_test.columns, index = X_test.index)
```

In [49]:
```
X_test.fillna(0,inplace=True)
```

In [50]:
```
X_train
X_test
y_train
y_test
```

Out[50]:

	Temperature(celsius)	Pressure(mb)	Dew point(celsius)	Wind Speed(m/s)	Wind Direction(degrees)	Precipitation(mm)	Snowfall(mm)	Cloud Cover(%)	Humidity(%
0	-0.62	-1.31	0.06	-0.52	0.92	1.08	-0.09	0.84	1.8
1	-0.83	-0.16	-0.48	-0.30	0.38	-0.29	-0.09	0.84	0.8
2	-0.88	-0.12	-0.38	-0.97	1.24	-0.29	-0.09	0.84	1.3
3	-0.88	-0.12	-0.38	-0.97	1.24	-0.29	-0.09	0.84	1.3
4	-1.63	0.21	-1.10	-0.07	0.81	-0.29	-0.09	0.84	1.6
...	...	...	...	...	...	...	...	...	.
1977	0.20	0.19	0.07	-0.30	-0.15	-0.29	-0.09	-0.35	-0.4
1978	0.04	0.21	-0.06	-0.07	-0.15	-0.29	-0.09	-1.02	-0.4
1981	-0.52	-0.35	-0.16	-0.75	-1.12	-0.29	-0.09	-0.87	0.8

In [51]:
```
from sklearn.ensemble import RandomForestRegressor

model2 = RandomForestRegressor(random_state=2)
model2.fit(X_train, y_train)

# The following gives the R-square score
model2.score(X_train, y_train)
```

Out[51]: RandomForestRegressor(random_state=2)

Out[51]: 0.8302205377964764

In [52]:
```
test_output2 = pd.DataFrame(model2.predict(X_test), index = X_test.index, columns = ['pred_arrival_delay'])
# When extending to multiple features remove .array.reshape(-1, 1)
test_output2.head()
```

Out[52]:

	pred_arrival_delay
0	58.40
1	7.56
2	78.54
3	57.65
4	15.62

In [53]:
```
def categorize_flight_delays(delay_minutes):
    if delay_minutes <= -30:
        return 'Severely Late'
    elif delay_minutes <= -10:
        return 'Late'
    elif delay_minutes <= 10:
        return 'On-time'
    else:
        return 'Early'

# Use the 'apply()' method to apply the function to each row in the DataFrame and create a new column 'Status (Early,
test_output2['Status (Early, On-time, Late, Severly Late)'] = test_output2['pred_arrival_delay'].apply(categorize_flig
```

In [54]:
```
subset_Test_data = subset_Test_data.merge(test_output2,left_index =True, right_index = True)
subset_Test_data
```

Out[54]:

	Date	Day	Origin Airport	Flight Number	Arrival Time	pred_arrival_delay	Status (Early, On-time, Late, Severly Late)
0	4/21/2023	Friday	ORD	UA 3839	10:00 AM	58.40	Early
1	4/21/2023	Friday	ORD	UA 3524	4:50 PM	7.56	On-time
2	4/21/2023	Friday	ORD	UA 538	9:34 PM	78.54	Early
3	4/22/2023	Saturday	ORD	UA 3839	10:00 AM	57.65	Early
4	4/22/2023	Saturday	ORD	UA 3524	4:50 PM	15.62	Early
5	4/22/2023	Saturday	ORD	UA 538	9:34 PM	26.04	Early
6	4/23/2023	Sunday	ORD	UA 3839	10:00 AM	56.26	Early
7	4/23/2023	Sunday	ORD	UA 3524	4:55 PM	-3.04	On-time
8	4/23/2023	Sunday	ORD	UA 538	9:34 PM	0.88	On-time
9	4/24/2023	Monday	ORD	UA 3839	10:00 AM	56.26	Early
10	4/24/2023	Monday	ORD	UA 3524	4:50 PM	-1.97	On-time
11	4/24/2023	Monday	ORD	UA 538	9:34 PM	0.88	On-time
12	4/21/2023	Friday	DEN	UA 604	3:12 PM	13.85	Early
13	4/22/2023	Saturday	DEN	UA 604	3:12 PM	11.32	Early
14	4/23/2023	Sunday	DEN	UA 604	3:12 PM	7.48	On-time
15	4/24/2023	Monday	DEN	UA 604	3:12 PM	7.48	On-time
16	4/21/2023	Friday	EWR	UA 4189	10:46 AM	1.46	On-time
17	4/21/2023	Friday	EWR	UA 1412	11:42 PM	142.30	Early
18	4/22/2023	Saturday	EWR	UA 4189	10:46 AM	4.04	On-time
19	4/22/2023	Saturday	EWR	UA 1412	11:17 PM	16.95	Early
20	4/23/2023	Sunday	EWR	UA 4189	10:46 AM	3.95	On-time
21	4/23/2023	Sunday	EWR	UA 1412	11:42 PM	7.55	On-time
22	4/24/2023	Monday	EWR	UA 4189	10:46 AM	3.95	On-time
23	4/24/2023	Monday	EWR	UA 1412	11:42 PM	7.55	On-time
24	4/21/2023	Friday	IAD	UA 4490	1:57 PM	7.02	On-time
25	4/21/2023	Friday	IAD	UA 4165	6:59 PM	37.88	Early
26	4/22/2023	Saturday	IAD	UA 3805	1:58 PM	31.92	Early
27	4/22/2023	Saturday	IAD	UA 4165	6:59 PM	71.23	Early
28	4/23/2023	Sunday	IAD	UA 4490	1:57 PM	3.65	On-time
29	4/23/2023	Sunday	IAD	UA 4165	6:59 PM	1.30	On-time
30	4/24/2023	Monday	IAD	UA 4490	1:57 PM	3.65	On-time
31	4/24/2023	Monday	IAD	UA 4165	6:59 PM	1.30	On-time