

```
In [1]: 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 LogisticRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor
from sklearn.preprocessing import StandardScaler

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]: import plotly
plotly.offline.init_notebook_mode(connected=True)
from plotly.graph_objs import *
from plotly import tools
import plotly.graph_objects as go
import seaborn as sns
```

```
In [3]: from datetime import datetime
```

```
airlines_data = pd.read_csv('../datasets/project/Detailed_Statistics_Arrivals.csv', parse_dates=['Date'])
airlines_data.head()

syr_weather_data = pd.read_csv('../datasets/project/hs/syracuse-syr.csv', parse_dates=['timestamp_local'])
ord_weather_data = pd.read_csv('../datasets/project/hs/chicago-ord.csv', parse_dates=['timestamp_local'])
den_weather_data = pd.read_csv('../datasets/project/hs/denver-den.csv', parse_dates=['timestamp_local'])
ewr_weather_data = pd.read_csv('../datasets/project/hs/newark-ewr.csv', parse_dates=['timestamp_local'])
iad_weather_data = pd.read_csv('../datasets/project/hs/washington-iad.csv', parse_dates=['timestamp_local'])

syr_weather_data.columns
ord_weather_data.columns
den_weather_data.columns
ewr_weather_data.columns
iad_weather_data.columns
```

```
Out[3]:
```

	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)	Wheels- on Time	Taxi-In time (Minutes)	Delay Carrier (Minutes)	
0	UA	2022-01-01	1282	N4901U	IAD	23:10	0:01	70	76	51	23:55	6	23	
1	UA	2023-01-01	604	N814UA	DEN	14:58	14:52	193	177	-6	14:48	4	0	
2	UA	2023-01-01	2488	N38458	EWB	23:14	23:15	75	62	1	23:10	5	0	
3	UA	2023-01-01	2645	N23721	ORD	23:57	23:47	107	100	-10	23:41	6	0	
4	UA	2022-01-02	1282	N4901U	IAD	23:10	23:27	70	64	17	23:19	8	17	

```
Out[3]: Index(['app_temp', 'azimuth', 'clouds', 'datetime', 'dewpt', 'dhi', 'dni',
               'elev_angle', 'ghi', 'h_angle', 'pod', 'precip', 'pres',
               'revision_status', 'rh', 'slp', 'snow', 'solar_rad', 'temp',
               'timestamp_local', 'timestamp_utc', 'ts', 'uv', 'vis', 'weather',
               'wind_dir', 'wind_gust_spd', 'wind_spd'],
              dtype='object')
```

```
Out[3]: Index(['app_temp', 'azimuth', 'clouds', 'datetime', 'dewpt', 'dhi', 'dni',  
             'elev_angle', 'ghi', 'h_angle', 'pod', 'precip', 'pres',  
             'revision_status', 'rh', 'slp', 'snow', 'solar_rad', 'temp',  
             'timestamp_local', 'timestamp_utc', 'ts', 'uv', 'vis', 'weather',  
             'wind_dir', 'wind_gust_spd', 'wind_spd'],  
            dtype='object')
```

```
Out[3]: Index(['app_temp', 'azimuth', 'clouds', 'datetime', 'dewpt', 'dhi', 'dni',  
             'elev_angle', 'ghi', 'h_angle', 'pod', 'precip', 'pres',  
             'revision_status', 'rh', 'slp', 'snow', 'solar_rad', 'temp',  
             'timestamp_local', 'timestamp_utc', 'ts', 'uv', 'vis', 'weather',  
             'wind_dir', 'wind_gust_spd', 'wind_spd'],  
            dtype='object')
```

```
Out[3]: Index(['app_temp', 'azimuth', 'clouds', 'datetime', 'dewpt', 'dhi', 'dni',  
             'elev_angle', 'ghi', 'h_angle', 'pod', 'precip', 'pres',  
             'revision_status', 'rh', 'slp', 'snow', 'solar_rad', 'temp',  
             'timestamp_local', 'timestamp_utc', 'ts', 'uv', 'vis', 'weather',  
             'wind_dir', 'wind_gust_spd', 'wind_spd'],  
            dtype='object')
```

```
Out[3]: Index(['app_temp', 'azimuth', 'clouds', 'datetime', 'dewpt', 'dhi', 'dni',  
             'elev_angle', 'ghi', 'h_angle', 'pod', 'precip', 'pres',  
             'revision_status', 'rh', 'slp', 'snow', 'solar_rad', 'temp',  
             'timestamp_local', 'timestamp_utc', 'ts', 'uv', 'vis', 'weather',  
             'wind_dir', 'wind_gust_spd', 'wind_spd'],  
            dtype='object')
```

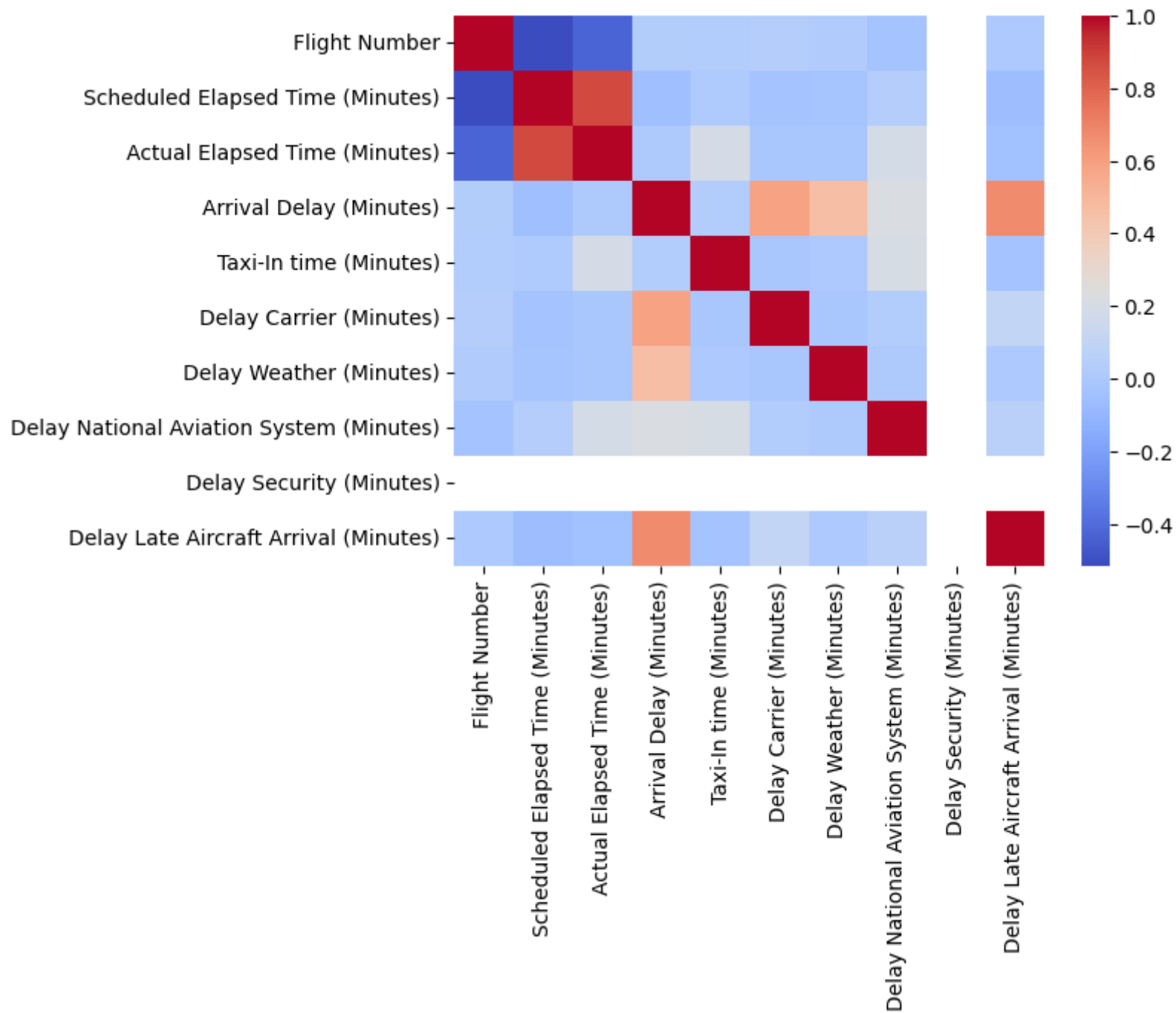
```
In [4]: airlines_data.dtypes
```

```
Out[4]: Carrier Code                object
Date (MM/DD/YYYY)                 datetime64[ns]
Flight Number                      int64
Tail Number                       object
Origin Airport                    object
Scheduled Arrival Time            object
Actual Arrival Time               object
Scheduled Elapsed Time (Minutes)  int64
Actual Elapsed Time (Minutes)     int64
Arrival Delay (Minutes)          int64
Wheels-on Time                   object
Taxi-In time (Minutes)           int64
Delay Carrier (Minutes)          int64
Delay Weather (Minutes)          int64
Delay National Aviation System (Minutes) int64
Delay Security (Minutes)         int64
Delay Late Aircraft Arrival (Minutes) int64
dtype: object
```

```
In [5]: corr_matrix = airlines_data.corr(numeric_only=True)
```

```
In [6]: sns.heatmap(corr_matrix, cmap='coolwarm')
```

```
Out[6]: <AxesSubplot: >
```




```
In [10]: syr_weather_data.describe()
```

```
Out[10]:
```

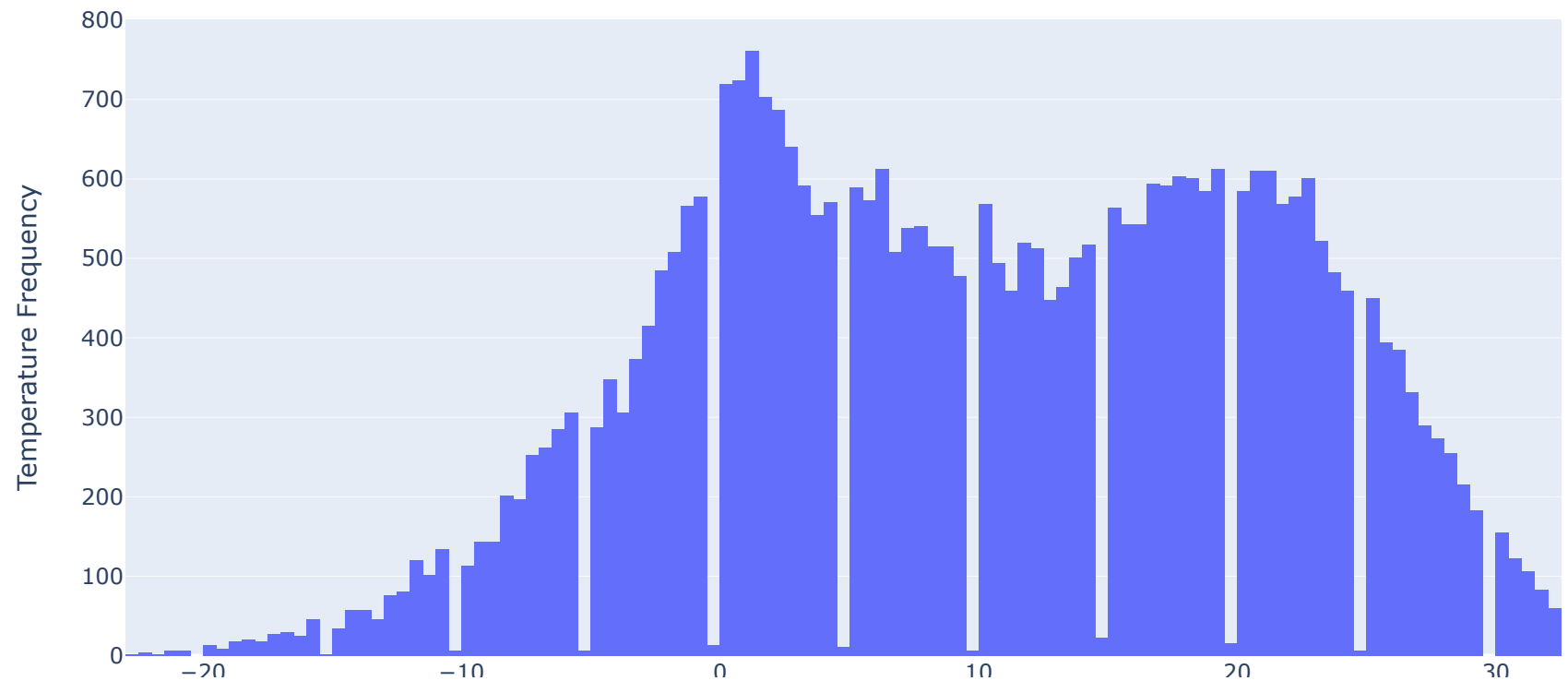
	temp	clouds	rh	dewpt	precip	snow	vis
count	35,808.00	35,808.00	35,808.00	35,808.00	35,808.00	35,808.00	35,804.00
mean	10.11	75.95	67.93	3.90	0.20	0.49	14.79
std	10.97	29.87	17.30	10.28	0.81	4.90	3.23
min	-22.80	0.00	10.00	-26.70	0.00	0.00	0.00
25%	1.32	50.00	56.00	-3.90	0.00	0.00	16.00
50%	10.00	87.00	70.00	3.80	0.00	0.00	16.00
75%	19.40	100.00	82.00	12.70	0.00	0.00	16.00
max	35.60	100.00	100.00	24.90	30.50	277.50	16.00

```
In [11]: syr_temp = syr_weather_data['temp']
```



```
In [12]: trace = go.Histogram(x=syr_temp)
layout = go.Layout(title='Histogram for Temperature Frequency in Syracuse',
                    xaxis=dict(title='Temperature'), yaxis=dict(title='Temperature Frequency'))
fig = go.Figure(data=[trace], layout=layout)
fig.show()
```

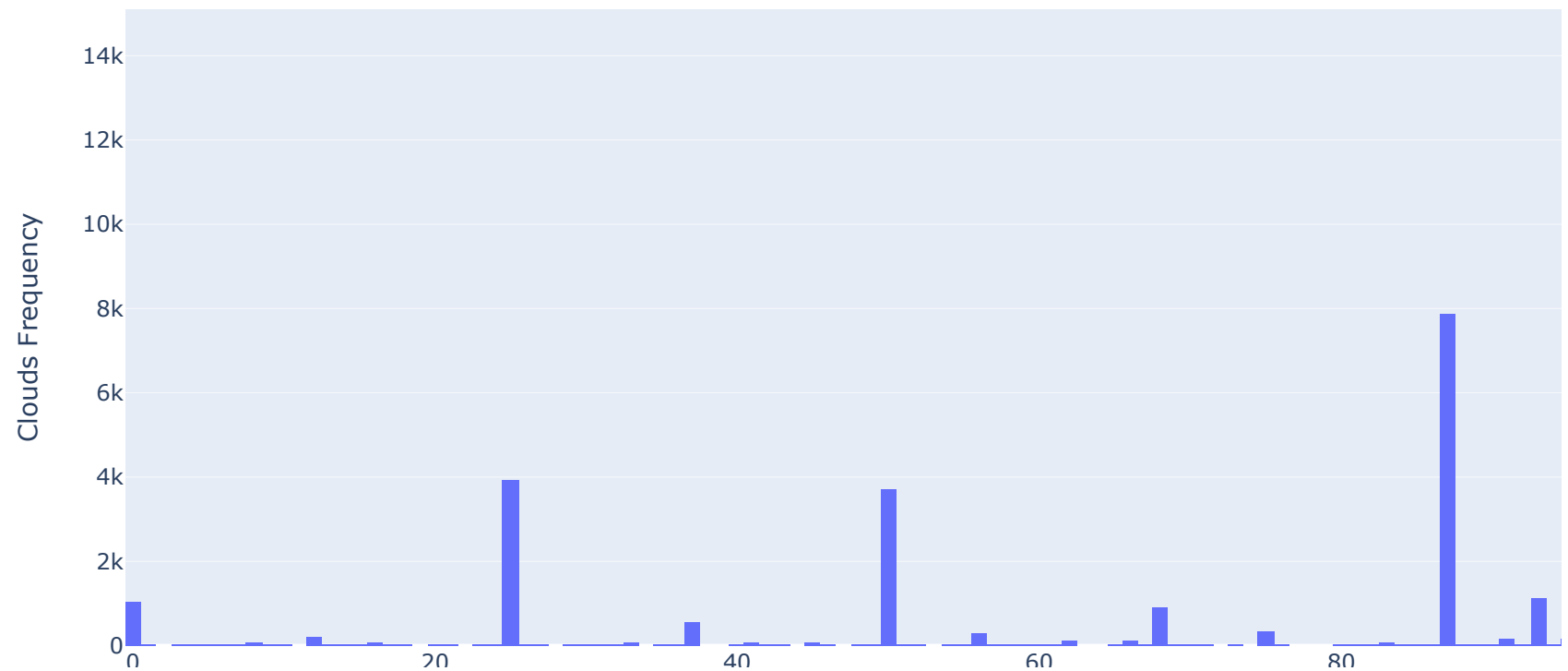
Histogram for Temperature Frequency in Syracuse



```
In [13]: syr_clouds = syr_weather_data['clouds']
```

```
In [14]: trace = go.Histogram(x=syr_clouds)
layout = go.Layout(title='Histogram for Clouds Frequency in Syracuse',
                    xaxis=dict(title='Clouds'), yaxis=dict(title='Clouds Frequency'))
fig = go.Figure(data=[trace], layout=layout)
fig.show()
```

Histogram for Clouds Frequency in Syracuse



```
In [15]: ewr_weather_data.describe()
```

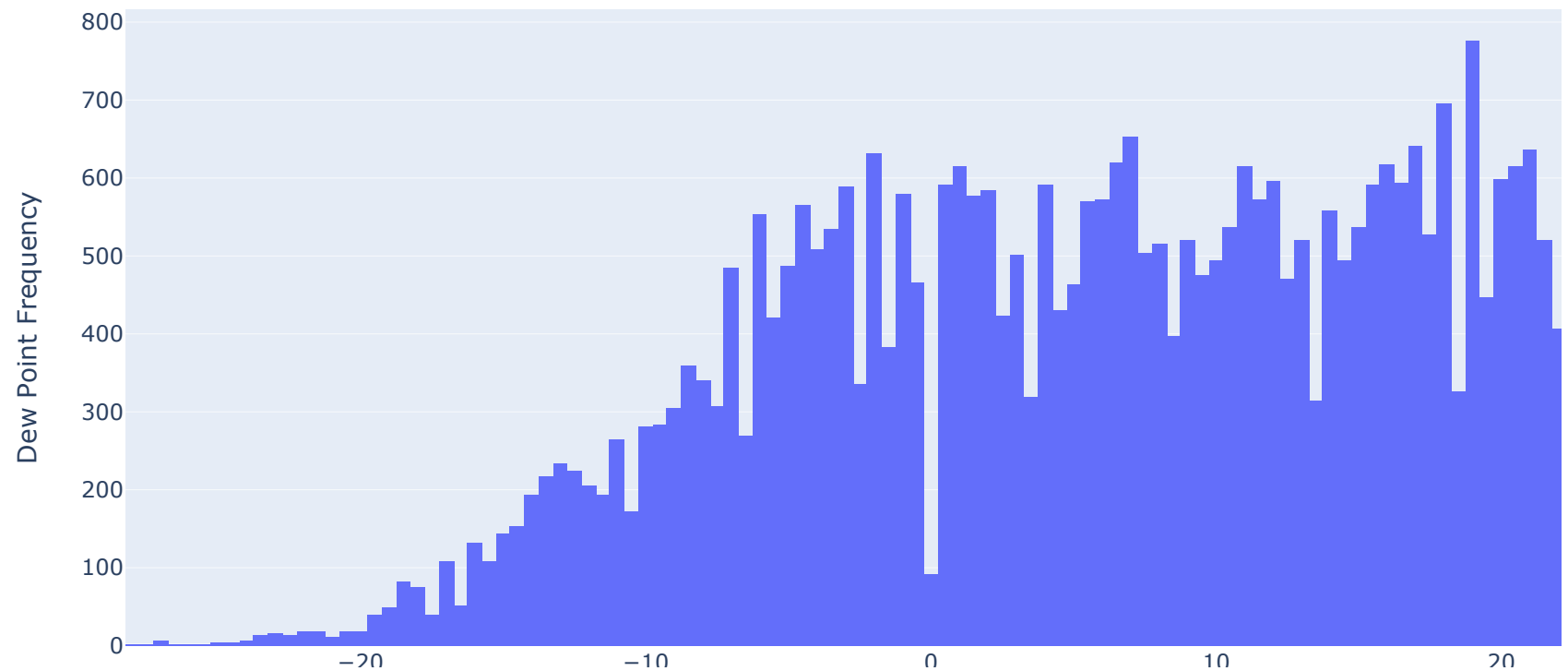
```
Out[15]:
```

	temp	clouds	rh	dewpt	precip	snow	vis
count	35,808.00	35,808.00	35,808.00	35,808.00	35,808.00	35,808.00	35,808.00
mean	13.77	69.88	61.02	5.56	0.20	0.15	14.97
std	10.12	31.69	20.17	10.65	0.93	2.55	3.10
min	-16.70	0.00	6.00	-28.00	0.00	0.00	0.00
25%	5.60	50.00	45.00	-2.80	0.00	0.00	16.00
50%	13.90	87.00	61.00	6.30	0.00	0.00	16.00
75%	22.20	100.00	77.00	14.80	0.00	0.00	16.00
max	38.90	100.00	100.00	24.90	28.50	250.00	16.00

```
In [16]: ewr_dewpt = ewr_weather_data['dewpt']
```

```
In [17]: trace = go.Histogram(x=ewr_dewpt)
layout = go.Layout(title='Histogram for Dew Point Frequency in New Jersey (EWR)',
                    xaxis=dict(title='Dew Point'), yaxis=dict(title='Dew Point Frequency'))
fig = go.Figure(data=[trace], layout=layout)
fig.show()
```

Histogram for Dew Point Frequency in New Jersey (EWR)



```

In [18]: airlines_factors = ['Date (MM/DD/YYYY)',
                             'Delay National Aviation System (Minutes)',
                             'Delay Weather (Minutes)',
                             'Delay Carrier (Minutes)',
                             'Delay Late Aircraft Arrival (Minutes)',
                             'Arrival Delay (Minutes)',
                             'Scheduled Arrival Time',
                             'Origin Airport']

airlines_data = airlines_data[airlines_factors]
airlines_data = airlines_data.rename(columns={'Date (MM/DD/YYYY)': 'Date',
                                             'Delay Carrier (Minutes)': 'Delay Carrier',
                                             'Delay Weather (Minutes)': 'Delay Weather',
                                             'Delay Late Aircraft Arrival (Minutes)': 'Delay Late Aircraft Arrival',
                                             'Arrival Delay (Minutes)': 'Arrival Delay',
                                             'Delay National Aviation System (Minutes)': 'Delay National Aviation System'})

airlines_data.columns

syr_weather_data = syr_weather_data.rename(columns={'timestamp_local': 'time'})
syr_weather_data.columns

ord_weather_data = ord_weather_data.rename(columns={'timestamp_local': 'time'})
ord_weather_data.columns

den_weather_data = den_weather_data.rename(columns={'timestamp_local': 'time'})
den_weather_data.columns

ewr_weather_data = ewr_weather_data.rename(columns={'timestamp_local': 'time'})
ewr_weather_data.columns

iad_weather_data = iad_weather_data.rename(columns={'timestamp_local': 'time'})
iad_weather_data.columns

```

```

Out[18]: Index(['Date', 'Delay National Aviation System', 'Delay Weather',
               'Delay Carrier', 'Delay Late Aircraft Arrival', 'Arrival Delay',
               'Scheduled Arrival Time', 'Origin Airport'],
              dtype='object')

```

```

Out[18]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

```

```

Out[18]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

```

```

Out[18]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

```

```
Out[18]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
```

```
Out[18]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
```

```
In [19]: airlines_data['time'] = pd.to_datetime(airlines_data['Date'].astype(str) + 'T' + airlines_data['Scheduled  
airlines_data = airlines_data.drop(columns=['Date', 'Scheduled Arrival Time'])  
airlines_data.head()
```

```
Out[19]:
```

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time
0	6	0	23	22	51	IAD	2022-01-01 23:00:00
1	0	0	0	0	-6	DEN	2023-01-01 15:00:00
2	0	0	0	0	1	EWR	2023-01-01 23:00:00
3	0	0	0	0	-10	ORD	2023-01-02 00:00:00
4	0	0	17	0	17	IAD	2022-01-02 23:00:00

```
In [20]: syr_weather_data.dtypes  
  
ord_weather_data.dtypes  
  
den_weather_data.dtypes  
  
ewr_weather_data.dtypes  
  
iad_weather_data.dtypes
```

```
Out[20]: time      datetime64[ns]  
temp          float64  
clouds         int64  
rh             int64  
dewpt          float64  
precip         float64  
snow           float64  
vis            float64  
dtype: object
```

```
Out[20]: time      datetime64[ns]  
temp          float64  
clouds         int64  
rh             int64  
dewpt          float64  
precip         float64  
snow           float64  
vis            float64  
dtype: object
```

```
Out[20]: time      datetime64[ns]  
temp          float64  
clouds         int64  
rh             int64  
dewpt          float64  
precip         float64  
snow           float64  
vis            float64  
dtype: object
```

```
Out[20]: time      datetime64[ns]
temp      float64
clouds    int64
rh         int64
dewpt     float64
precip    float64
snow      float64
vis        int64
dtype: object
```

```
Out[20]: time      datetime64[ns]
temp      float64
clouds    int64
rh         int64
dewpt     float64
precip    float64
snow      float64
vis        int64
dtype: object
```

```
In [21]: airlines_data.dtypes
```

```
Out[21]: Delay National Aviation System      int64
Delay Weather                               int64
Delay Carrier                               int64
Delay Late Aircraft Arrival                 int64
Arrival Delay                              int64
Origin Airport                             object
time                                         datetime64[ns]
dtype: object
```

```
In [22]: airlines_data_cp = airlines_data.copy()
syr_dataset = airlines_data_cp.merge(syr_weather_data, on='time', how='left')
```



```
In [23]: syr_dataset.columns
airlines_data.columns
```

```
Out[23]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
               'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
               'time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'],
              dtype='object')
```

```
Out[23]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
               'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
               'time'],
              dtype='object')
```

```
In [24]: corrmat = syr_dataset.corr()
corrmat['Arrival Delay'].sort_values()
```

/var/folders/hm/06_n0ww96r7d2cltmt4xyjr0000gn/T/ipykernel_4470/2392745747.py:1: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the value of numeric_only to silence this warning.

```
Out[24]: vis                -0.04
         snow                -0.00
         temp                 0.01
         clouds               0.03
         dewpt                0.03
         precip               0.04
         rh                   0.05
         Delay National Aviation System  0.22
         Delay Weather         0.47
         Delay Carrier         0.59
         Delay Late Aircraft Arrival  0.67
         Arrival Delay         1.00
         Name: Arrival Delay, dtype: float64
```

```

In [25]: ord_dataset = airlines_data[airlines_data['Origin Airport'] == 'ORD']
ord_dataset.columns

ord_weather_data = ord_weather_data.rename(columns={'timestamp_local': 'time'})
ord_weather_data.columns

ord_dataset = ord_dataset.merge(ord_weather_data, on='time', how='left')

ord_dataset.columns

ord_dataset.head()

corrmat = ord_dataset.corr()
corrmat['Arrival Delay'].sort_values()

```

```

Out[25]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
               'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
               'time'],
              dtype='object')

```

```

Out[25]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

```

```

Out[25]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
               'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
               'time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'],
              dtype='object')

```

```

Out[25]:

```

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time	temp	clouds	rh	dewpt	precip	snow	vis
0	0	0	0	0	-10	ORD	2023-01-02 00:00:00	5.20	92	89	3.50	0.00	0.00	10.00
1	0	0	0	0	-21	ORD	2023-01-02 21:00:00	2.90	96	94	2.00	0.08	0.00	10.00
2	0	0	0	0	-3	ORD	2023-01-03 21:00:00	3.00	92	96	2.40	0.23	0.00	10.00
3	0	0	0	0	14	ORD	2023-01-04 21:00:00	0.50	85	89	-1.10	0.09	1.07	10.00
4	0	0	0	0	8	ORD	2020-01-05 22:00:00	2.10	54	68	-3.20	0.00	0.00	NaN

```
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/3030693318.py:13: FutureWarning:
```

The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
Out[25]: vis                -0.15
temp                -0.03
snow                 0.00
dewpt                0.02
precip               0.07
clouds               0.09
rh                   0.12
Delay National Aviation System  0.22
Delay Weather        0.50
Delay Carrier        0.55
Delay Late Aircraft Arrival  0.69
Arrival Delay        1.00
Name: Arrival Delay, dtype: float64
```

```

In [26]: den_dataset = airlines_data[airlines_data['Origin Airport'] == 'DEN']
den_dataset.shape

den_weather_data = den_weather_data.rename(columns={'timestamp_local': 'time'})
den_weather_data.columns

den_dataset = den_dataset.merge(den_weather_data, on='time', how='left')

den_dataset.columns

den_dataset.head()

corrmat = den_dataset.corr()
corrmat['Arrival Delay'].sort_values()

```

Out[26]: (354, 7)

Out[26]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

Out[26]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
'time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'],
dtype='object')

Out[26]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time	temp	clouds	rh	dewpt	precip	snow	vis
0	0	0	0	0	-6	DEN	2023-01-01 15:00:00	2.00	94	64	-4.10	0.00	0.00	16.00
1	24	0	40	91	155	DEN	2023-01-02 15:00:00	-3.00	100	100	-3.00	0.00	0.00	3.00
2	0	0	0	0	-6	DEN	2023-01-03 15:00:00	-3.00	87	79	-6.10	0.00	0.00	16.00
3	0	0	0	0	-12	DEN	2023-01-04 15:00:00	2.00	25	31	-13.30	0.00	0.00	16.00
4	0	0	0	0	0	DEN	2023-01-05 15:00:00	0.00	87	50	-9.20	0.00	0.00	16.00

/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/4010358392.py:13: FutureWarning:

The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
Out[26]: vis                -0.19
temp                  -0.15
precip                -0.02
dewpt                 -0.01
snow                   0.01
clouds                 0.08
Delay Weather          0.19
rh                     0.21
Delay National Aviation System  0.52
Delay Late Aircraft Arrival  0.63
Delay Carrier           0.73
Arrival Delay           1.00
Name: Arrival Delay, dtype: float64
```

```

In [27]: ewr_dataset = airlines_data[airlines_data['Origin Airport'] == 'EWR']
ewr_dataset.shape

ewr_weather_data = ewr_weather_data.rename(columns={'timestamp_local': 'time'})
ewr_weather_data.columns

ewr_dataset = ewr_dataset.merge(ewr_weather_data, on='time', how='left')

ewr_dataset.columns

ewr_dataset.head()

corrmat = ewr_dataset.corr()
corrmat['Arrival Delay'].sort_values()

```

Out[27]: (64, 7)

Out[27]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

Out[27]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
'time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'],
dtype='object')

Out[27]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time	temp	clouds	rh	dewpt	precip	snow	vis
0	0	0	0	0	1	EWR	2023-01-01 23:00:00	10.00	100	58	2.10	0.00	0.00	16
1	0	0	52	0	52	EWR	2023-01-09 23:00:00	3.30	25	54	-5.10	0.00	0.00	16
2	0	0	0	0	-23	EWR	2023-01-10 23:00:00	4.40	100	54	-4.10	0.00	0.00	16
3	0	0	0	0	-4	EWR	2023-01-11 23:00:00	3.90	100	64	-2.30	0.00	0.00	16
4	88	0	0	58	146	EWR	2023-01-12 23:00:00	13.30	87	90	11.70	0.00	0.00	14

```
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/4148591676.py:13: FutureWarning:
```

The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
Out[27]: temp                -0.29
vis                  -0.25
dewpt               -0.15
clouds              -0.03
precip               0.05
rh                   0.21
Delay National Aviation System  0.30
Delay Carrier        0.69
Delay Late Aircraft Arrival    0.90
Arrival Delay        1.00
Delay Weather        NaN
snow                 NaN
Name: Arrival Delay, dtype: float64
```

```

In [28]: iad_dataset = airlines_data[airlines_data['Origin Airport'] == 'IAD']
iad_dataset.shape

iad_weather_data = iad_weather_data.rename(columns={'timestamp_local': 'time'})
iad_weather_data.columns

iad_dataset = iad_dataset.merge(iad_weather_data, on='time', how='left')

iad_dataset.columns

iad_dataset.head()

corrmat = iad_dataset.corr()
corrmat['Arrival Delay'].sort_values()

```

Out[28]: (331, 7)

Out[28]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')

Out[28]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
'Delay Late Aircraft Arrival', 'Arrival Delay', 'Origin Airport',
'time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'],
dtype='object')

Out[28]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time	temp	clouds	rh	dewpt	precip	snow	vis
0	6	0	23	22	51	IAD	2022-01-01 23:00:00	13.90	100	96	13.30	1.25	0.00	16
1	0	0	17	0	17	IAD	2022-01-02 23:00:00	6.70	100	65	0.60	0.75	0.00	16
2	1	0	0	26	27	IAD	2023-01-02 23:00:00	11.10	87	80	7.80	0.00	0.00	16
3	0	0	21	0	21	IAD	2022-01-03 23:00:00	-2.20	0	62	-8.50	0.00	0.00	16
4	20	0	115	0	135	IAD	2022-01-05 00:00:00	-2.80	16	82	-5.40	0.00	0.00	16

/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/238141727.py:13: FutureWarning:

The default value of `numeric_only` in `DataFrame.corr` is deprecated. In a future version, it will default to `False`. Select only valid columns or specify the value of `numeric_only` to silence this warning.

```
Out[28]: vis                -0.06
         snow               -0.01
         temp               0.13
         rh                 0.16
         dewpt              0.16
         Delay National Aviation System 0.16
         clouds             0.19
         precip             0.27
         Delay Weather      0.44
         Delay Late Aircraft Arrival 0.62
         Delay Carrier      0.63
         Arrival Delay      1.00
         Name: Arrival Delay, dtype: float64
```

```
In [29]: dataset = pd.concat([syr_dataset, ord_dataset, iad_dataset, den_dataset, ewr_dataset])
```

```
In [30]: from sklearn.utils import shuffle
         dataset = shuffle(dataset)
```

```
In [31]: dataset.head()
```

Out[31]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	Arrival Delay	Origin Airport	time	temp	clouds	rh	dewpt	precip	snow	vis
894	0	0	0	0	-8	ORD	2022-09-25 00:00:00	15.80	85	84	13.10	0.66	0.00	16.00
196	0	0	0	0	11	IAD	2022-08-08 00:00:00	26.70	0	76	22.10	0.00	0.00	16.00
1622	0	0	0	0	-18	ORD	2022-10-21 17:00:00	16.70	37	28	-1.90	0.00	0.00	16.00
939	0	0	0	0	6	ORD	2022-07-14 17:00:00	26.10	37	42	12.20	0.00	0.00	16.00
1197	0	0	0	0	14	ORD	2019-08-22 21:00:00	18.30	50	70	12.70	0.00	0.00	16.00

```
In [32]: dataset.dtypes
```

Out[32]: Delay National Aviation System int64
Delay Weather int64
Delay Carrier int64
Delay Late Aircraft Arrival int64
Arrival Delay int64
Origin Airport object
time datetime64[ns]
temp float64
clouds int64
rh int64
dewpt float64
precip float64
snow float64
vis float64
dtype: object

```
In [33]: dataset.isna().sum()
```

```
Out[33]: Delay National Aviation System      0
          Delay Weather                      0
          Delay Carrier                      0
          Delay Late Aircraft Arrival        0
          Arrival Delay                     0
          Origin Airport                     0
          time                              0
          temp                              0
          clouds                            0
          rh                               0
          dewpt                             0
          precip                            0
          snow                              0
          vis                               780
          dtype: int64
```

```
In [34]: dataset = dataset.dropna()
```

```
In [35]: dataset = dataset.drop(columns=['time'])
          dataset = pd.get_dummies(dataset, columns=['Origin Airport'], drop_first=True)
          dataset.shape
```

```
Out[35]: (3168, 15)
```

```
In [36]: dataset.columns
```

```
Out[36]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',
                'Delay Late Aircraft Arrival', 'Arrival Delay', 'temp', 'clouds', 'rh',
                'dewpt', 'precip', 'snow', 'vis', 'Origin Airport_EWR',
                'Origin Airport_IAD', 'Origin Airport_ORD'],
                dtype='object')
```

```
In [37]: X_train, X_test, y_train, y_test = train_test_split(dataset.drop(columns=['Arrival Delay']), dataset['Arrival Delay'],
test_size=0.2, random_state=42)

X_train
X_test
y_train
y_test
```

Out[37]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	temp	clouds	rh	dewpt	precip	snow	vis	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_ORD
791	0	0	0	0	25.10	88	58	16.30	0.00	0.00	16.00	0	0	1
17	34	0	0	14	-3.00	100	92	-4.10	0.25	5.00	2.00	0	0	0
265	0	0	0	0	-1.10	100	68	-6.20	0.00	0.00	8.00	0	0	0
613	0	0	0	0	10.60	100	96	10.00	0.00	0.00	8.00	0	1	0
794	0	0	0	0	14.40	25	64	7.70	0.00	0.00	16.00	0	1	0
...
600	0	0	0	0	13.90	25	21	-8.00	0.00	0.00	16.00	0	0	0
428	0	0	0	0	12.80	100	38	-1.10	0.00	0.00	16.00	0	0	1
1779	0	0	0	0	-0.60	50	31	-15.60	0.00	0.00	16.00	0	0	0
1415	0	0	0	0	18.30	100	75	13.80	0.00	0.00	16.00	0	0	1
66	0	0	0	0	8.00	87	36	-6.10	0.00	0.00	16.00	0	0	0

2534 rows × 14 columns

Out[37]:

	Delay National Aviation System	Delay Weather	Delay Carrier	Delay Late Aircraft Arrival	temp	clouds	rh	dewpt	precip	snow	vis	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_ORD
1272	0	0	0	0	20.00	37	40	6.00	0.00	0.00	16.00	0	0	1
990	0	0	0	0	4.70	73	61	-2.20	0.27	0.00	16.00	0	0	1
723	0	0	0	0	23.60	71	66	16.90	0.00	0.00	16.00	0	0	1
585	0	0	0	0	3.90	95	52	-5.00	0.00	0.00	16.00	0	0	0
1239	0	0	0	0	16.70	25	83	13.80	0.00	0.00	16.00	0	1	0
...
723	0	0	0	0	22.80	25	49	11.50	0.00	0.00	16.00	0	0	1
250	0	0	0	0	27.00	25	22	3.50	0.00	0.00	16.00	0	0	0
452	0	0	0	0	2.20	100	86	0.10	0.50	0.00	6.00	0	0	0
627	0	0	0	0	13.90	100	48	3.10	0.00	0.00	16.00	0	0	1
296	0	0	51	67	3.00	100	51	-6.10	0.00	0.00	13.00	0	0	0

634 rows × 14 columns

Out[37]:

```
791    -12
17      48
265    -18
613     -7
794    -10
      ..
600     -9
428    -27
1779    -2
1415    -5
66     -11
```

Name: Arrival Delay, Length: 2534, dtype: int64

```
Out[37]: 1272      12
          990     -11
          723     -2
          585     -6
          1239      4
          ...
          723    -15
          250    -12
          452      3
          627    -15
          296    118
          Name: Arrival Delay, Length: 634, dtype: int64
```

```
In [38]: sc = StandardScaler()
X_train_scaled = pd.DataFrame(sc.fit_transform(X_train), columns = X_train.columns, index = X_train.index)
```

```
In [39]: from sklearn.tree import DecisionTreeRegressor

clf = DecisionTreeRegressor(random_state=50)

clf = clf.fit(X_train_scaled, y_train)

clf.score(X_train_scaled, y_train)
```

```
Out[39]: 0.999424055086456
```

```
In [40]: from sklearn.tree import export_text  
r = export_text(clf)  
print(r)
```

```

--- feature_3 <= 2.53
|--- feature_1 <= 11.35
|   |--- feature_2 <= -0.16
|   |   |--- feature_3 <= -0.17
|   |   |   |--- feature_1 <= 0.50
|   |   |   |   |--- feature_0 <= 0.76
|   |   |   |   |   |--- feature_13 <= -0.05
|   |   |   |   |   |   |--- feature_4 <= 1.06
|   |   |   |   |   |   |   |--- feature_5 <= 0.60
|   |   |   |   |   |   |   |   |--- feature_7 <= 0.36
|   |   |   |   |   |   |   |   |   |--- feature_8 <= 0.53
|   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 20
|   |   |   |   |   |   |   |   |   |   |--- feature_8 > 0.53
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |--- feature_7 > 0.36
|   |   |   |   |   |   |   |   |   |   |   |--- feature_10 <= -0.93
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 3
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_10 > -0.93
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 21
|   |   |   |   |   |   |   |   |   |   |--- feature_5 > 0.60
|   |   |   |   |   |   |   |   |   |   |   |--- feature_6 <= -0.76
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_7 <= 0.37
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- feature_7 > 0.37
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |--- value: [7.00]
|   |   |   |   |   |   |   |   |   |   |--- feature_6 > -0.76
|   |   |   |   |   |   |   |   |   |   |   |--- feature_4 <= 0.45
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 20
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_4 > 0.45
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 15
|   |   |   |   |   |   |   |--- feature_4 > 1.06
|   |   |   |   |   |   |   |   |--- feature_7 <= -0.07
|   |   |   |   |   |   |   |   |   |--- feature_7 <= -0.16
|   |   |   |   |   |   |   |   |   |   |--- feature_7 <= -0.74
|   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 7
|   |   |   |   |   |   |   |   |   |   |   |--- feature_7 > -0.74
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 14
|   |   |   |   |   |   |   |   |   |   |--- feature_7 > -0.16
|   |   |   |   |   |   |   |   |   |   |   |--- feature_6 <= -1.93
|   |   |   |   |   |   |   |   |   |   |   |   |--- truncated branch of depth 4
|   |   |   |   |   |   |   |   |   |   |   |   |--- feature_6 > -1.93
|   |   |   |   |   |   |   |   |   |   |   |   |   |--- value: [-2.00]
|   |   |   |   |   |   |   |--- feature_7 > -0.07

```



```

|--- feature_7 <= 0.06
|   |--- feature_5 <= 0.01
|   |   |--- truncated branch of depth 2
|   |--- feature_5 > 0.01
|   |   |--- truncated branch of depth 4
|--- feature_7 > 0.06
|   |--- feature_7 <= 1.66
|   |   |--- truncated branch of depth 15
|   |--- feature_7 > 1.66
|   |   |--- truncated branch of depth 5
--- feature_13 > -0.05
|--- feature_6 <= 0.93
|   |--- feature_4 <= -1.70
|   |   |--- feature_6 <= 0.72
|   |   |   |--- feature_10 <= 0.04
|   |   |   |   |--- truncated branch of depth 5
|   |   |   |--- feature_10 > 0.04
|   |   |   |   |--- truncated branch of depth 9
|   |   |--- feature_6 > 0.72
|   |   |   |--- feature_7 <= -2.00
|   |   |   |   |--- truncated branch of depth 2
|   |   |   |--- feature_7 > -2.00
|   |   |   |   |--- truncated branch of depth 2
|   |--- feature_4 > -1.70
|   |   |--- feature_4 <= 0.50
|   |   |   |--- feature_7 <= 0.70
|   |   |   |   |--- truncated branch of depth 24
|   |   |   |--- feature_7 > 0.70
|   |   |   |   |--- truncated branch of depth 9
|   |--- feature_4 > 0.50
|   |   |--- feature_4 <= 1.79
|   |   |   |--- truncated branch of depth 26
|   |   |--- feature_4 > 1.79
|   |   |   |--- truncated branch of depth 2
--- feature_6 > 0.93
|   |--- feature_7 <= 1.44
|   |   |--- feature_9 <= 0.10
|   |   |   |--- feature_7 <= 0.71
|   |   |   |   |--- truncated branch of depth 16
|   |   |   |--- feature_7 > 0.71
|   |   |   |   |--- truncated branch of depth 13
|   |--- feature_9 > 0.10
|   |   |--- feature_8 <= -0.07

```

```

|--- truncated branch of depth 2
|--- feature_8 > -0.07
|--- truncated branch of depth 6
|--- feature_7 > 1.44
|--- feature_1 <= 0.21
|--- feature_4 <= 0.90
|--- truncated branch of depth 4
|--- feature_4 > 0.90
|--- truncated branch of depth 6
|--- feature_1 > 0.21
|--- value: [18.00]
--- feature_0 > 0.76
--- feature_0 <= 19.43
--- feature_0 <= 5.20
|--- feature_0 <= 2.85
|--- feature_0 <= 2.03
|--- feature_0 <= 1.77
|--- value: [15.00]
|--- feature_0 > 1.77
|--- truncated branch of depth 2
|--- feature_0 > 2.03
|--- feature_0 <= 2.41
|--- truncated branch of depth 3
|--- feature_0 > 2.41
|--- truncated branch of depth 2
|--- feature_0 > 2.85
|--- feature_0 <= 3.55
|--- feature_0 <= 3.17
|--- truncated branch of depth 2
|--- feature_0 > 3.17
|--- truncated branch of depth 3
|--- feature_0 > 3.55
|--- feature_7 <= 0.76
|--- truncated branch of depth 2
|--- feature_7 > 0.76
|--- value: [33.00]
--- feature_0 > 5.20
|--- feature_0 <= 7.30
|--- feature_6 <= 0.23
|--- value: [54.00]
|--- feature_6 > 0.23
|--- value: [52.00]
|--- feature_0 > 7.30

```

```

|--- feature_4 <= 0.32
|   |--- value: [64.00]
|   |--- feature_4 > 0.32
|       |--- value: [69.00]
|--- feature_0 > 19.43
|   |--- value: [240.00]
--- feature_1 > 0.50
    |--- feature_1 <= 2.07
        |--- feature_0 <= 4.50
            |--- feature_1 <= 0.91
                |--- feature_5 <= -0.16
                    |--- value: [19.00]
                |--- feature_5 > -0.16
                    |--- value: [30.00]
            |--- feature_1 > 0.91
                |--- feature_7 <= -0.54
                    |--- feature_1 <= 1.48
                        |--- value: [51.00]
                    |--- feature_1 > 1.48
                        |--- value: [59.00]
                |--- feature_7 > -0.54
                    |--- feature_0 <= 0.50
                        |--- feature_6 <= 1.10
                            |--- value: [41.00]
                        |--- feature_6 > 1.10
                            |--- truncated branch of depth 2
                    |--- feature_0 > 0.50
                        |--- value: [47.00]
            |--- feature_0 > 4.50
                |--- value: [85.00]
        |--- feature_1 > 2.07
            |--- feature_1 <= 2.61
                |--- feature_0 <= -0.07
                    |--- value: [82.00]
                |--- feature_0 > -0.07
                    |--- value: [87.00]
            |--- feature_1 > 2.61
                |--- feature_8 <= 0.22
                    |--- feature_12 <= 0.69
                        |--- value: [119.00]
                    |--- feature_12 > 0.69
                        |--- value: [115.00]
                |--- feature_8 > 0.22

```

```

|--- feature_9 <= 0.06
|   |--- value: [162.00]
|--- feature_9 > 0.06
|   |--- value: [147.00]
--- feature_3 > -0.17
    |--- feature_3 <= 1.20
        |--- feature_3 <= 0.88
            |--- feature_0 <= 2.41
                |--- feature_3 <= 0.40
                    |--- feature_3 <= 0.02
                        |--- feature_7 <= 0.73
                            |--- value: [24.00]
                        |--- feature_7 > 0.73
                            |--- value: [27.00]
                    |--- feature_3 > 0.02
                        |--- feature_7 <= 1.37
                            |--- feature_6 <= -0.83
                                |--- value: [17.00]
                            |--- feature_6 > -0.83
                                |--- truncated branch of depth 3
                        |--- feature_7 > 1.37
                            |--- value: [15.00]
                |--- feature_3 > 0.40
                    |--- feature_3 <= 0.65
                        |--- feature_0 <= 0.88
                            |--- feature_10 <= -0.12
                                |--- truncated branch of depth 2
                            |--- feature_10 > -0.12
                                |--- truncated branch of depth 3
                        |--- feature_0 > 0.88
                            |--- feature_0 <= 1.77
                                |--- truncated branch of depth 2
                            |--- feature_0 > 1.77
                                |--- value: [41.00]
                    |--- feature_3 > 0.65
                        |--- feature_4 <= 0.35
                            |--- feature_8 <= 0.07
                                |--- truncated branch of depth 2
                            |--- feature_8 > 0.07
                                |--- value: [31.00]
                        |--- feature_4 > 0.35
                            |--- feature_1 <= 0.17
                                |--- value: [38.00]

```

```

|--- feature_1 > 0.17
|--- value: [44.00]
--- feature_0 > 2.41
|--- feature_5 <= -0.88
|--- value: [33.00]
--- feature_5 > -0.88
|--- feature_4 <= -0.20
|--- feature_7 <= -1.52
|--- value: [56.00]
|--- feature_7 > -1.52
|--- value: [48.00]
|--- feature_4 > -0.20
|--- value: [66.00]
--- feature_3 > 0.88
--- feature_0 <= 0.44
|--- feature_3 <= 1.08
|--- feature_3 <= 0.97
|--- feature_4 <= 0.25
|--- value: [37.00]
|--- feature_4 > 0.25
|--- value: [38.00]
|--- feature_3 > 0.97
|--- feature_0 <= 0.06
|--- feature_4 <= -0.72
|--- truncated branch of depth 2
|--- feature_4 > -0.72
|--- value: [41.00]
|--- feature_0 > 0.06
|--- value: [43.00]
|--- feature_3 > 1.08
|--- feature_12 <= 0.69
|--- value: [43.00]
|--- feature_12 > 0.69
|--- value: [45.00]
--- feature_0 > 0.44
--- feature_0 <= 1.46
|--- feature_4 <= -0.16
|--- feature_3 <= 1.08
|--- value: [51.00]
|--- feature_3 > 1.08
|--- value: [50.00]
|--- feature_4 > -0.16
|--- value: [46.00]

```

```

|--- feature_0 > 1.46
|   |--- feature_0 <= 1.90
|   |   |--- value: [57.00]
|   |--- feature_0 > 1.90
|   |   |--- value: [55.00]
|--- feature_3 > 1.20
|   |--- feature_11 <= 2.35
|   |   |--- feature_3 <= 2.06
|   |   |   |--- feature_0 <= 3.42
|   |   |   |   |--- feature_1 <= 0.64
|   |   |   |   |   |--- feature_3 <= 1.72
|   |   |   |   |   |   |--- feature_0 <= 0.57
|   |   |   |   |   |   |   |--- truncated branch of depth 5
|   |   |   |   |   |   |   |--- feature_0 > 0.57
|   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |--- feature_3 > 1.72
|   |   |   |   |   |   |   |   |--- feature_3 <= 1.90
|   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |   |   |--- feature_3 > 1.90
|   |   |   |   |   |   |   |   |   |--- truncated branch of depth 2
|   |   |   |   |   |   |--- feature_1 > 0.64
|   |   |   |   |   |   |   |--- value: [89.00]
|   |   |   |--- feature_0 > 3.42
|   |   |   |   |--- value: [94.00]
|   |   |--- feature_3 > 2.06
|   |   |   |--- feature_6 <= 0.95
|   |   |   |   |--- feature_0 <= 0.31
|   |   |   |   |   |--- value: [86.00]
|   |   |   |   |--- feature_0 > 0.31
|   |   |   |   |   |--- value: [88.00]
|   |   |   |--- feature_6 > 0.95
|   |   |   |   |--- value: [76.00]
|   |--- feature_11 > 2.35
|   |   |--- value: [146.00]
|--- feature_2 > -0.16
|   |--- feature_2 <= 18.19
|   |   |--- feature_2 <= 2.39
|   |   |   |--- feature_3 <= 0.74
|   |   |   |   |--- feature_2 <= 1.04
|   |   |   |   |   |--- feature_0 <= 1.65
|   |   |   |   |   |   |--- feature_2 <= 0.62
|   |   |   |   |   |   |   |--- feature_3 <= 0.39
|   |   |   |   |   |   |   |   |--- feature_0 <= 1.01

```

```

|--- truncated branch of depth 12
|--- feature_0 > 1.01
|--- truncated branch of depth 5
|--- feature_3 > 0.39
|--- feature_2 <= 0.34
|--- truncated branch of depth 5
|--- feature_2 > 0.34
|--- value: [44.00]
|--- feature_2 > 0.62
|--- feature_3 <= 0.32
|--- feature_0 <= 0.44
|--- truncated branch of depth 7
|--- feature_0 > 0.44
|--- truncated branch of depth 3
|--- feature_3 > 0.32
|--- feature_3 <= 0.48
|--- value: [51.00]
|--- feature_3 > 0.48
|--- value: [50.00]
|--- feature_0 > 1.65
|--- feature_0 <= 3.49
|--- feature_2 <= 0.67
|--- feature_2 <= 0.09
|--- value: [44.00]
|--- feature_2 > 0.09
|--- truncated branch of depth 4
|--- feature_2 > 0.67
|--- feature_10 <= -0.93
|--- value: [58.00]
|--- feature_10 > -0.93
|--- truncated branch of depth 3
|--- feature_0 > 3.49
|--- feature_2 <= 0.74
|--- feature_0 <= 3.74
|--- value: [55.00]
|--- feature_0 > 3.74
|--- truncated branch of depth 3
|--- feature_2 > 0.74
|--- value: [66.00]
|--- feature_2 > 1.04
|--- feature_2 <= 1.70
|--- feature_0 <= 2.09
|--- feature_2 <= 1.29

```

```

|--- feature_3 <= -0.07
|   |--- truncated branch of depth 5
|   |--- feature_3 > -0.07
|       |--- truncated branch of depth 2
|   |--- feature_2 > 1.29
|       |--- feature_3 <= -0.18
|           |--- truncated branch of depth 5
|           |--- feature_3 > -0.18
|               |--- truncated branch of depth 3
|   |--- feature_0 > 2.09
|       |--- value: [67.00]
|--- feature_2 > 1.70
|   |--- feature_0 <= 0.06
|       |--- feature_2 <= 2.00
|           |--- feature_2 <= 1.84
|               |--- truncated branch of depth 3
|               |--- feature_2 > 1.84
|                   |--- truncated branch of depth 2
|       |--- feature_2 > 2.00
|           |--- feature_7 <= 0.31
|               |--- truncated branch of depth 2
|               |--- feature_7 > 0.31
|                   |--- truncated branch of depth 2
|   |--- feature_0 > 0.06
|       |--- feature_4 <= 0.07
|           |--- feature_2 <= 2.04
|               |--- value: [65.00]
|               |--- feature_2 > 2.04
|                   |--- truncated branch of depth 2
|       |--- feature_4 > 0.07
|           |--- value: [91.00]
|--- feature_3 > 0.74
|   |--- feature_3 <= 1.60
|       |--- feature_2 <= 0.30
|           |--- feature_3 <= 1.27
|               |--- feature_0 <= -0.13
|                   |--- feature_3 <= 0.94
|                       |--- truncated branch of depth 2
|                       |--- feature_3 > 0.94
|                           |--- truncated branch of depth 2
|           |--- feature_0 > -0.13
|               |--- feature_7 <= 0.09
|                   |--- value: [53.00]

```



```

|--- feature_7 > 0.09
|   |--- value: [54.00]
|--- feature_3 > 1.27
|   |--- feature_6 <= 0.86
|       |--- feature_6 <= 0.56
|           |--- truncated branch of depth 4
|               |--- feature_6 > 0.56
|                   |--- value: [56.00]
|               |--- feature_6 > 0.86
|                   |--- value: [73.00]
|--- feature_2 > 0.30
|   |--- feature_2 <= 1.27
|       |--- feature_7 <= 0.04
|           |--- feature_4 <= 0.16
|               |--- value: [70.00]
|           |--- feature_4 > 0.16
|               |--- value: [73.00]
|       |--- feature_7 > 0.04
|           |--- feature_0 <= 0.31
|               |--- value: [68.00]
|           |--- feature_0 > 0.31
|               |--- truncated branch of depth 2
|   |--- feature_2 > 1.27
|       |--- value: [85.00]
|--- feature_3 > 1.60
|   |--- feature_2 <= 0.73
|       |--- feature_3 <= 2.10
|           |--- feature_0 <= 1.33
|               |--- feature_3 <= 1.81
|                   |--- truncated branch of depth 3
|               |--- feature_3 > 1.81
|                   |--- truncated branch of depth 3
|           |--- feature_0 > 1.33
|               |--- value: [90.00]
|   |--- feature_3 > 2.10
|       |--- feature_7 <= 0.97
|           |--- feature_6 <= -0.99
|               |--- value: [88.00]
|           |--- feature_6 > -0.99
|               |--- truncated branch of depth 2
|       |--- feature_7 > 0.97
|           |--- feature_4 <= 0.89
|               |--- value: [91.00]

```

```

|--- feature_4 > 0.89
|--- value: [93.00]
|--- feature_2 > 0.73
|--- feature_7 <= 0.53
|--- value: [118.00]
|--- feature_7 > 0.53
|--- value: [103.00]
|--- feature_2 > 2.39
|--- feature_2 <= 6.34
|--- feature_2 <= 3.88
|--- feature_7 <= 0.63
|--- feature_2 <= 2.92
|--- feature_2 <= 2.66
|--- feature_6 <= -1.93
|--- value: [78.00]
|--- feature_6 > -1.93
|--- truncated branch of depth 2
|--- feature_2 > 2.66
|--- feature_8 <= 0.22
|--- value: [82.00]
|--- feature_8 > 0.22
|--- value: [83.00]
|--- feature_2 > 2.92
|--- feature_12 <= 0.69
|--- feature_11 <= 2.35
|--- value: [93.00]
|--- feature_11 > 2.35
|--- value: [92.00]
|--- feature_12 > 0.69
|--- value: [102.00]
|--- feature_7 > 0.63
|--- feature_7 <= 1.55
|--- feature_4 <= 0.19
|--- value: [83.00]
|--- feature_4 > 0.19
|--- feature_5 <= -0.79
|--- value: [93.00]
|--- feature_5 > -0.79
|--- truncated branch of depth 5
|--- feature_7 > 1.55
|--- value: [81.00]
|--- feature_2 > 3.88
|--- feature_6 <= -0.08

```

```

|--- value: [149.00]
|--- feature_6 > -0.08
|--- feature_13 <= -0.05
|--- value: [135.00]
|--- feature_13 > -0.05
|--- value: [131.00]
|--- feature_2 > 6.34
|--- feature_2 <= 8.00
|--- value: [219.00]
|--- feature_2 > 8.00
|--- value: [243.00]
|--- feature_2 > 18.19
|--- value: [794.00]
|--- feature_1 > 11.35
|--- feature_13 <= -0.05
|--- value: [594.00]
|--- feature_13 > -0.05
|--- value: [986.00]
|--- feature_3 > 2.53
|--- feature_2 <= 1.73
|--- feature_3 <= 6.55
|--- feature_3 <= 3.86
|--- feature_4 <= -0.47
|--- feature_5 <= 0.73
|--- feature_0 <= 1.01
|--- value: [129.00]
|--- feature_0 > 1.01
|--- value: [108.00]
|--- feature_5 > 0.73
|--- feature_0 <= 3.93
|--- feature_13 <= -0.05
|--- value: [155.00]
|--- feature_13 > -0.05
|--- value: [157.00]
|--- feature_0 > 3.93
|--- value: [147.00]
|--- feature_4 > -0.47
|--- feature_3 <= 3.40
|--- feature_0 <= 0.12
|--- feature_3 <= 2.90
|--- feature_4 <= 0.60
|--- feature_6 <= 0.74
|--- value: [109.00]

```

```

| | | | |--- feature_6 > 0.74
| | | | |--- value: [101.00]
| | | | |--- feature_4 > 0.60
| | | | |--- value: [93.00]
| | | | |--- feature_3 > 2.90
| | | | |--- feature_3 <= 3.19
| | | | |--- feature_7 <= 1.14
| | | | |--- value: [108.00]
| | | | |--- feature_7 > 1.14
| | | | |--- value: [105.00]
| | | | |--- feature_3 > 3.19
| | | | |--- value: [114.00]
| | | | |--- feature_0 > 0.12
| | | | |--- feature_4 <= 0.47
| | | | |--- value: [115.00]
| | | | |--- feature_4 > 0.47
| | | | |--- value: [125.00]
| | | | |--- feature_3 > 3.40
| | | | |--- feature_3 <= 3.70
| | | | |--- feature_13 <= -0.05
| | | | |--- value: [126.00]
| | | | |--- feature_13 > -0.05
| | | | |--- value: [124.00]
| | | | |--- feature_3 > 3.70
| | | | |--- value: [131.00]
| | | | |--- feature_3 > 3.86
| | | | |--- feature_3 <= 4.11
| | | | |--- feature_3 <= 3.94
| | | | |--- value: [185.00]
| | | | |--- feature_3 > 3.94
| | | | |--- feature_10 <= -3.83
| | | | |--- value: [161.00]
| | | | |--- feature_10 > -3.83
| | | | |--- feature_0 <= 1.39
| | | | |--- feature_0 <= 0.38
| | | | |--- feature_9 <= 1.50
| | | | |--- feature_7 <= 0.44
| | | | |--- value: [139.00]
| | | | |--- feature_7 > 0.44
| | | | |--- value: [138.00]
| | | | |--- feature_9 > 1.50
| | | | |--- value: [142.00]
| | | | |--- feature_0 > 0.38

```

```
| | | | | | |-- value: [144.00]
| | | | | | |--- feature_0 > 1.39
| | | | | | |--- value: [155.00]
|-- feature_3 > 4.11
|   |-- feature_7 <= -0.63
|       |-- feature_3 <= 4.34
|           |--- feature_4 <= -1.15
|               |--- value: [174.00]
|                   |--- feature_4 > -1.15
|                       |--- value: [154.00]
|               --- feature_3 > 4.34
|                   |--- feature_3 <= 4.85
|                       |--- feature_0 <= -0.00
|                           |--- value: [176.00]
|                               |--- feature_0 > -0.00
|                                   |--- value: [178.00]
|                   |--- feature_3 > 4.85
|                       |--- value: [170.00]
|               --- feature_7 > -0.63
|                   |-- feature_1 <= -0.04
|                       |--- feature_8 <= 1.14
|                           |--- feature_0 <= 1.46
|                               |--- feature_3 <= 4.89
|                                   |--- value: [184.00]
|                                       |--- feature_3 > 4.89
|                                           |--- value: [182.00]
|                           |--- feature_0 > 1.46
|                               |--- value: [191.00]
|                       |--- feature_8 > 1.14
|                           |--- value: [174.00]
|                   --- feature_1 > -0.04
|                       |--- value: [170.00]
|-- feature_3 > 6.55
|   |-- feature_6 <= 1.38
|       |-- feature_3 <= 9.15
|           |-- feature_5 <= -0.16
|               |--- value: [285.00]
|           --- feature_5 > -0.16
|               |-- feature_2 <= 0.02
|                   |--- value: [273.00]
|               --- feature_2 > 0.02
|                   |--- value: [276.00]
|       --- feature_3 > 9.15
```

```

|--- feature_6 <= -0.22
|   |--- value: [370.00]
|   |--- feature_6 > -0.22
|       |--- feature_5 <= 0.01
|           |--- value: [355.00]
|           |--- feature_5 > 0.01
|               |--- value: [353.00]
|       |--- feature_6 > 1.38
|           |--- value: [578.00]
|--- feature_2 > 1.73
|   |--- feature_2 <= 17.91
|       |--- feature_0 <= 0.69
|           |--- feature_7 <= 0.14
|               |--- feature_3 <= 5.69
|                   |--- value: [317.00]
|                   |--- feature_3 > 5.69
|                       |--- value: [380.00]
|               |--- feature_7 > 0.14
|                   |--- value: [233.00]
|       |--- feature_0 > 0.69
|           |--- value: [514.00]
|--- feature_2 > 17.91
|   |--- value: [912.00]

```

```
In [41]: X_test_scaled = pd.DataFrame(sc.transform(X_test), columns = X_test.columns, index = X_test.index)
```

```
In [42]: test_output = pd.DataFrame(clf.predict(X_test_scaled), index = X_test_scaled.index, columns = ['pred_Arrival'])
test_output.head()
```

Out[42]:

	pred_Arrival_Delay
1272	-19.00
990	-7.00
723	-15.00
585	3.00
1239	11.00

```
In [43]: test_output = test_output.merge(y_test, left_index = True, right_index = True)
test_output.head()
mae = abs(test_output['pred_Arrival_Delay'] - test_output['Arrival Delay']).mean()
mae.round(2)
```

Out[43]:

	pred_Arrival_Delay	Arrival Delay
0	3.00	-6
1	52.00	52
3	-2.00	-12
4	146.00	146
6	-15.00	1

Out[43]: 13.5

```
In [44]: (abs(test_output['pred_Arrival_Delay'] - test_output['Arrival Delay']).mean()/test_output['Arrival Delay'])
```

Out[44]: 1.5

```
In [45]: clf.score(X_test_scaled, y_test)
```

Out[45]: 0.9061376204302364


```
In [46]: import json
```

```
with open('../datasets/project/hs/forecast/syr-forecast.json', 'r') as f:
    syr_forecast_data = json.load(f)
    syr_forecast_df = pd.json_normalize(syr_forecast_data['data'])
    syr_forecast_df = syr_forecast_df[weather_factors]
    syr_forecast_df = syr_forecast_df.rename(columns={'timestamp_local': 'time'})
    syr_forecast_df['time'] = syr_forecast_df['time'].astype(np.datetime64)
    syr_forecast_df.columns

with open('../datasets/project/hs/forecast/den-forecast.json', 'r') as f:
    den_forecast_data = json.load(f)
    den_forecast_df = pd.json_normalize(den_forecast_data['data'])
    den_forecast_df = den_forecast_df[weather_factors]
    den_forecast_df = den_forecast_df.rename(columns={'timestamp_local': 'time'})
    den_forecast_df['time'] = syr_forecast_df['time'].astype(np.datetime64)
    den_forecast_df.columns

with open('../datasets/project/hs/forecast/ewr-forecast.json', 'r') as f:
    ewr_forecast_data = json.load(f)
    ewr_forecast_df = pd.json_normalize(ewr_forecast_data['data'])
    ewr_forecast_df = ewr_forecast_df[weather_factors]
    ewr_forecast_df = ewr_forecast_df.rename(columns={'timestamp_local': 'time'})
    ewr_forecast_df['time'] = syr_forecast_df['time'].astype(np.datetime64)
    ewr_forecast_df.columns

with open('../datasets/project/hs/forecast/iad-forecast.json', 'r') as f:
    iad_forecast_data = json.load(f)
    iad_forecast_df = pd.json_normalize(iad_forecast_data['data'])
    iad_forecast_df = iad_forecast_df[weather_factors]
    iad_forecast_df = iad_forecast_df.rename(columns={'timestamp_local': 'time'})
    iad_forecast_df['time'] = syr_forecast_df['time'].astype(np.datetime64)
    iad_forecast_df.columns

with open('../datasets/project/hs/forecast/ord-forecast.json', 'r') as f:
    ord_forecast_data = json.load(f)
    ord_forecast_df = pd.json_normalize(ord_forecast_data['data'])
    ord_forecast_df = ord_forecast_df[weather_factors]
    ord_forecast_df = ord_forecast_df.rename(columns={'timestamp_local': 'time'})
    ord_forecast_df['time'] = syr_forecast_df['time'].astype(np.datetime64)
    ord_forecast_df.columns
```

```
Out[46]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/438693687.py:16: FutureWarning:
Passing unit-less datetime64 dtype to .astype is deprecated and will raise in a future version. Pass
'datetime64[ns]' instead

Out[46]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/438693687.py:24: FutureWarning:
Passing unit-less datetime64 dtype to .astype is deprecated and will raise in a future version. Pass
'datetime64[ns]' instead

Out[46]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/438693687.py:32: FutureWarning:
Passing unit-less datetime64 dtype to .astype is deprecated and will raise in a future version. Pass
'datetime64[ns]' instead

Out[46]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
/var/folders/hm/06_n0ww96r7d2cltmt4xtyjr0000gn/T/ipykernel_4470/438693687.py:40: FutureWarning:
Passing unit-less datetime64 dtype to .astype is deprecated and will raise in a future version. Pass
'datetime64[ns]' instead

Out[46]: Index(['time', 'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis'], dtype='object')
```

```
In [47]: pred_data = pd.read_csv('../datasets/project/project csv(Apr 21-24).csv')
pred_data.head()
```

Out[47]:

	Date	Day	Origin Airport	Flight Number	Arrival Time	Status (Early, On-time, Late, Severly Late)
0	4/21/23	Friday	ORD	UA 3839	10:00 AM	Late
1	4/21/23	Friday	ORD	UA 3524	4:50 PM	Late
2	4/21/23	Friday	ORD	UA 538	9:34 PM	Late
3	4/22/23	Saturday	ORD	UA 3839	10:00 AM	Late
4	4/22/23	Saturday	ORD	UA 3524	4:50 PM	Late

```
In [48]: pred_data['Date'] = pred_data['Date'].astype(np.datetime64)
```

```
In [49]: pred_data.isna().sum()
```

```
Out[49]: Date          0
Day          0
Origin Airport  0
Flight Number  0
Arrival Time   0
Status (Early, On-time, Late, Severly Late)  0
dtype: int64
```

```
In [50]: pred_data['time'] = pd.to_datetime(pred_data['Date'].astype(str) + 'T' + pred_data['Arrival Time'].astype(str))
pred_data = pred_data.drop(columns=['Date', 'Arrival Time', 'Day', 'Flight Number'])
pred_data.head()
```

Out[50]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	time
0	ORD	Late	2023-04-21 10:00:00
1	ORD	Late	2023-04-21 17:00:00
2	ORD	Late	2023-04-21 22:00:00
3	ORD	Late	2023-04-22 10:00:00
4	ORD	Late	2023-04-22 17:00:00

```
In [51]: ord_pred = pred_data[pred_data['Origin Airport'] == 'ORD']
ord_pred.shape

ord_pred = ord_pred.merge(ord_forecast_df, on='time', how='left')
ord_pred.head()
```

Out[51]: (12, 3)

Out[51]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	time	temp	clouds	rh	dewpt	precip	snow	vis
0	ORD	Late	2023-04-21 10:00:00	10.20	99	55	1.60	0.00	0	32.29
1	ORD	Late	2023-04-21 17:00:00	16.00	29	30	-1.50	0.00	0	0.62
2	ORD	Late	2023-04-21 22:00:00	11.10	0	39	-2.30	0.00	0	0.49
3	ORD	Late	2023-04-22 10:00:00	4.30	100	53	-4.40	0.00	0	33.79
4	ORD	Late	2023-04-22 17:00:00	3.60	100	92	2.50	2.52	0	1.40

```
In [52]: den_pred = pred_data[pred_data['Origin Airport'] == 'DEN']
den_pred.shape

den_pred = den_pred.merge(den_forecast_df, on='time', how='left')
den_pred.head()
```

Out[52]: (4, 3)

Out[52]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	time	temp	clouds	rh	dewpt	precip	snow	vis
0	DEN	Late	2023-04-21 15:00:00	8.40	55	16	-15.90	0.00	0.00	0.87
1	DEN	Late	2023-04-22 15:00:00	0.80	100	70	-4.10	0.50	3.48	4.20
2	DEN	Late	2023-04-23 15:00:00	10.60	76	40	-2.40	0.00	0.00	24.13
3	DEN	Late	2023-04-24 15:00:00	13.00	83	43	0.70	0.00	0.00	24.13

```
In [53]: ewr_pred = pred_data[pred_data['Origin Airport'] == 'EWR']
ewr_pred.shape

ewr_pred = ewr_pred.merge(ewr_forecast_df, on='time', how='left')
ewr_pred.head()
```

Out[53]: (8, 3)

Out[53]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	time	temp	clouds	rh	dewpt	precip	snow	vis
0	EWR	Late	2023-04-21 11:00:00	16.60	5	64	9.80	0.00	0	25.39
1	EWR	Late	2023-04-22 00:00:00	13.20	100	100	13.20	0.04	0	0.10
2	EWR	Late	2023-04-22 11:00:00	17.00	62	86	14.60	0.00	0	15.20
3	EWR	Late	2023-04-22 23:00:00	11.30	99	99	11.10	0.00	0	0.30
4	EWR	Late	2023-04-23 11:00:00	16.60	63	58	8.30	0.00	0	24.13

```
In [54]: iad_pred = pred_data[pred_data['Origin Airport'] == 'IAD']
iad_pred.shape

iad_pred = iad_pred.merge(iad_forecast_df, on='time', how='left')
iad_pred.head()
```

Out[54]: (8, 3)

Out[54]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	time	temp	clouds	rh	dewpt	precip	snow	vis
0	IAD	Late	2023-04-21 14:00:00	31.30	0	20	5.70	0.00	0	0.80
1	IAD	Late	2023-04-21 19:00:00	29.30	12	20	4.00	0.00	0	0.80
2	IAD	Late	2023-04-22 14:00:00	25.90	100	36	9.70	0.00	0	0.53
3	IAD	Late	2023-04-22 19:00:00	13.60	100	90	12.00	0.92	0	14.60
4	IAD	Late	2023-04-23 14:00:00	16.70	62	41	3.40	0.00	0	24.13

```
In [55]: d_nas = dataset['Delay National Aviation System'].mean()
d_w = dataset['Delay Weather'].mean()
d_c = dataset['Delay Carrier'].mean()
d_laa = dataset['Delay Late Aircraft Arrival'].mean()
```

```
In [56]: pred_dataset = pd.concat([ord_pred, den_pred, ewr_pred, iad_pred])
```

```
In [57]: pred_dataset = pred_dataset.drop(columns=['time'])
pred_dataset.head()
```

Out[57]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	temp	clouds	rh	dewpt	precip	snow	vis
0	ORD	Late	10.20	99	55	1.60	0.00	0.00	32.29
1	ORD	Late	16.00	29	30	-1.50	0.00	0.00	0.62
2	ORD	Late	11.10	0	39	-2.30	0.00	0.00	0.49
3	ORD	Late	4.30	100	53	-4.40	0.00	0.00	33.79
4	ORD	Late	3.60	100	92	2.50	2.52	0.00	1.40

```
In [58]: pred_dataset.insert(2, 'Delay National Aviation System', d_nas)
pred_dataset.insert(2, 'Delay Weather', d_w)
pred_dataset.insert(2, 'Delay Carrier', d_c)
pred_dataset.insert(2, 'Delay Late Aircraft Arrival', d_laa)
pred_dataset.head()
```

Out[58]:

	Origin Airport	Status (Early, On-time, Late, Severly Late)	Delay Late Aircraft Arrival	Delay Carrier	Delay Weather	Delay National Aviation System	temp	clouds	rh	dewpt	precip	snow	vis
0	ORD	Late	6.94	4.97	1.75	1.65	10.20	99	55	1.60	0.00	0.00	32.29
1	ORD	Late	6.94	4.97	1.75	1.65	16.00	29	30	-1.50	0.00	0.00	0.62
2	ORD	Late	6.94	4.97	1.75	1.65	11.10	0	39	-2.30	0.00	0.00	0.49
3	ORD	Late	6.94	4.97	1.75	1.65	4.30	100	53	-4.40	0.00	0.00	33.79
4	ORD	Late	6.94	4.97	1.75	1.65	3.60	100	92	2.50	2.52	0.00	1.40

```
In [59]: pred_dataset = pd.get_dummies(pred_dataset, columns=['Origin Airport'], drop_first=True)
```

```
In [60]: pred_dataset.columns
```

```
Out[60]: Index(['Status (Early, On-time, Late, Severly Late)',  
               'Delay Late Aircraft Arrival', 'Delay Carrier', 'Delay Weather',  
               'Delay National Aviation System', 'temp', 'clouds', 'rh', 'dewpt',  
               'precip', 'snow', 'vis', 'Origin Airport_EWR', 'Origin Airport_IAD',  
               'Origin Airport_ORD'],  
              dtype='object')
```

```
In [61]: ordered_cols = ['Delay National Aviation System', 'Delay Weather', 'Delay Carrier', 'Delay Late Aircraft  
                        'temp', 'clouds', 'rh', 'dewpt', 'precip', 'snow', 'vis',  
                        'Origin Airport_EWR', 'Origin Airport_IAD', 'Origin Airport_ORD']  
pred_dataset = pred_dataset.reindex(columns=ordered_cols)
```

```
In [62]: pred_dataset.columns
```

```
Out[62]: Index(['Delay National Aviation System', 'Delay Weather', 'Delay Carrier',  
               'Delay Late Aircraft Arrival', 'temp', 'clouds', 'rh', 'dewpt',  
               'precip', 'snow', 'vis', 'Origin Airport_EWR', 'Origin Airport_IAD',  
               'Origin Airport_ORD'],  
              dtype='object')
```

```
In [63]: pred_output = pd.DataFrame(clf.predict(pred_dataset), index = pred_dataset.index, columns = ['Pred Arrival Delay'])
```

```
In [64]: bins = [-float('inf'), -10, 10, 30, float('inf')]  
labels = ['Early', 'On-time', 'Late', 'Severly Late']  
  
pred_output['Status (Early, On-time, Late, Severly Late)'] = pd.cut(pred_output['Pred Arrival Delay'], bins=bins, labels=labels)
```

```
In [65]: pred_output = pred_output.drop(columns=['Pred Arrival Delay'])
```

```
In [66]: pred_output.to_csv('project csv(Apr 21-24).csv', index=False)
```

In []:


```
import requests, datetime, calendar, json, os
```

```
api_key = "INSERT_YOUR_API_KEY"  
# New Jersey Airport Code  
airport = "ewr"  
# Update coordinates as per the city  
lat = 40.69  
lon = -74.17
```

```
def add_months(source_date, months) -> datetime.datetime:  
    month = source_date.month - 1 + months  
    year = source_date.year + month // 12  
    month = month % 12 + 1  
    day = min(source_date.day, calendar.monthrange(year, month)[1])  
    return datetime.datetime(year, month, day)
```

```
if not os.path.isdir(os.path.join('data', airport, 'hourly')):  
    os.makedirs(os.path.join('data', airport, 'hourly'))
```

```
start_date = datetime.datetime(2019, 1, 1)
```

```
while start_date.year != 2023 or start_date.month != 4:
```

```
    try:  
        end_date = add_months(start_date, 1) - datetime.timedelta(days=1)  
        start_date_str, end_date_str = start_date.strftime('%Y-%m-%d'), end_date.strftime('%Y-%  
%m-%d')  
        url = "https://api.weatherbit.io/v2.0/history/hourly?lat=" + str(lat) + "&lon=" + str(  
            lon) + "&start_date=" + start_date_str + "&end_date=" + end_date_str +  
            "&tz=local&key=" + api_key
```

```
        print("hitting end points with start_date = " + start_date_str + ", end_date = " +  
end_date_str)
```

```
        resp = requests.get(url)
```

```
        if resp.status_code != 200:
```

```
            print(resp.json().get('status_message', default="Unsuccessful response. Could have  
exceeded day's limit"))  
            break
```

```
        data = resp.json()
```

```
        with open('data/' + airport + '/hourly/' + start_date_str + '-to-' + end_date_str +  
' .json', 'w') as f:
```

```
            json.dump(data, f)
```

```
            f.close()
```

```
            print("data saved from " + start_date_str + " to " + end_date_str)
```

```
except requests.exceptions.RequestException as e:
```

```
    print(e)
```

```
except Exception as e:
```

```
    print(e)
```

```
start_date = end_date + datetime.timedelta(days=1)
```