

1/12

```
Out[ ]: Date (MM/DD/YYYY)      0
Flight Number                 0
Destination Airport            0
Scheduled departure time      0
Departure delay (Minutes)     0
Taxi-Out time (Minutes)       0
Source Airport                 0
dtype: int64

In [ ]: syrData
originData

Out[ ]:
      Date (MM/DD/YYYY)  Flight Number  Origin Airport  Scheduled Arrival Time  Scheduled Elapsed Time (Minutes)  Arrival Delay (Minutes)  Taxi-In time (Minutes)
0      2022-01-01         1282          IAD      2023-04-20 23:10:00              70              51              6
1      2022-01-02         1282          IAD      2023-04-20 23:10:00              70              17              8
2      2022-01-03         1282          IAD      2023-04-20 23:10:00              70              21              6
3      2022-01-04         1282          IAD      2023-04-20 23:44:00              69             135              4
4      2022-01-05         1282          IAD      2023-04-20 23:44:00              69             -14              5
...      ...              ...              ...              ...              ...              ...              ...
1033    2022-12-30         1998          ORD      2023-04-20 21:07:00             113             -11              5
1034    2022-12-30         2488          EWR      2023-04-20 23:14:00              75              -7              4
1035    2022-12-31          604          DEN      2023-04-20 14:58:00             193             -12              7
1036    2022-12-31         1998          ORD      2023-04-20 21:08:00             113             -24              4
1037    2022-12-31         2488          EWR      2023-04-20 23:14:00              75              92              6

1038 rows x 7 columns

Out[ ]:
      Date (MM/DD/YYYY)  Flight Number  Destination Airport  Scheduled departure time  Departure delay (Minutes)  Taxi-Out time (Minutes)  Source Airport
0      2022-02-11         1200          SYR      2023-04-20 09:50:00              -3              15          DEN
1      2022-02-12         1200          SYR      2023-04-20 09:50:00              19              18          DEN
2      2022-02-13         1200          SYR      2023-04-20 09:50:00              1              10          DEN
3      2022-02-14         1200          SYR      2023-04-20 09:50:00              9              21          DEN
4      2022-02-15         1200          SYR      2023-04-20 09:50:00              -3              10          DEN
...      ...              ...              ...              ...              ...              ...
1033    2022-10-26         2198          SYR      2023-04-20 22:20:00              -4              14          IAD
1034    2022-10-27         2198          SYR      2023-04-20 22:20:00              -6              17          IAD
1035    2022-10-28         2198          SYR      2023-04-20 22:20:00             -13              18          IAD
1036    2022-10-29         2198          SYR      2023-04-20 22:20:00             -10              15          IAD
1037    2022-12-26         1968          SYR      2023-04-20 22:00:00              0              0          IAD

1038 rows x 7 columns

In [ ]: # Merging arrival and departure data into flightData
flightData = pd.merge(syrData, originData, left_on = ['Date (MM/DD/YYYY)', 'Flight Number', 'Origin Airport'], right_on= ['Date (MM/DD/YYYY)', 'Flight Number', 'Source Airport'])
flightData.shape
flightData.head()

Out[ ]: (1038, 12)

Out[ ]:
      Date (MM/DD/YYYY)  Flight Number  Origin Airport  Scheduled Arrival Time  Scheduled Elapsed Time (Minutes)  Arrival Delay (Minutes)  Taxi-In time (Minutes)  Destination Airport  Scheduled departure time  Departure delay (Minutes)  Taxi-Out time (Minutes)  Source Airport
0      2022-01-01         1282          IAD      2023-04-20 23:10:00              70              51              6          SYR      2023-04-20 22:00:00              45              21          IAD
1      2022-01-02         1282          IAD      2023-04-20 23:10:00              70              17              8          SYR      2023-04-20 22:00:00              23              9          IAD
2      2022-01-03         1282          IAD      2023-04-20 23:10:00              70              21              6          SYR      2023-04-20 22:00:00              23              13          IAD
3      2022-01-04         1282          IAD      2023-04-20 23:44:00              69             135              4          SYR      2023-04-20 22:35:00             115              43          IAD
4      2022-01-05         1282          IAD      2023-04-20 23:44:00              69             -14              5          SYR      2023-04-20 22:35:00              -6              12          IAD

In [ ]: # Reading Weather data
syrWeather = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/Weather_2022_final/Weather_2022/syracuse_weather_2022.csv", parse_dates=['datetime', 'timestamp_utc', 'timestamp_local'])
chWeather = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/Weather_2022_final/Weather_2022/chicago_weather_2022.csv", parse_dates=['datetime', 'timestamp_utc', 'timestamp_local'])
wasWeather = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/Weather_2022_final/Weather_2022/washingtondc_weather_2022.csv", parse_dates=['datetime', 'timestamp_utc', 'timestamp_local'])
newWeather = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/Weather_2022_final/Weather_2022/newark_weather_2022.csv", parse_dates=['datetime', 'timestamp_utc', 'timestamp_local'])
denWeather = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/Weather_2022_final/Weather_2022/denver_weather_2022.csv", parse_dates=['datetime', 'timestamp_utc', 'timestamp_local'])

syrWeather.head()
chWeather.head()
wasWeather.head()
newWeather.head()
denWeather.head()

Out[ ]:
      datetime  temp  pres  dewpt  wind_spd  wind_dir  precip  snow  clouds  rh  solar_rad  uv  vis  pod  timestamp_utc  timestamp_local  description
0  2022-01-01 00:00:00  5.00  996.70  4.30  1.50  140  0.00  0.00  0  95  0  0.00  16  n  2022-01-01 00:00:00  2021-12-31 19:00:00  Clear Sky
1  2022-01-01 01:00:00  3.30  996.40  3.30  2.10  110  0.00  0.00  25  100  0  0.00  14  n  2022-01-01 01:00:00  2021-12-31 20:00:00  Fog
2  2022-01-01 02:00:00  2.20  996.00  1.60  1.50  90  0.00  0.00  25  96  0  0.00  0  n  2022-01-01 02:00:00  2021-12-31 21:00:00  Haze
3  2022-01-01 03:00:00  1.70  995.70  1.70  2.60  110  0.00  0.00  25  100  0  0.00  11  n  2022-01-01 03:00:00  2021-12-31 22:00:00  Fog
4  2022-01-01 04:00:00  1.70  995.00  1.70  2.60  80  0.00  0.00  25  100  0  0.00  13  n  2022-01-01 04:00:00  2021-12-31 23:00:00  Fog

Out[ ]:
      datetime  temp  pres  dewpt  wind_spd  wind_dir  precip  snow  clouds  rh  solar_rad  uv  vis  pod  timestamp_utc  timestamp_local  description
0  2022-01-01 00:00:00  3.30  982.96  3.30  3.60  340  0.00  0.00  100  100  0  0.00  NaN  n  2022-01-01 00:00:00  2021-12-31 18:00:00  Fog
1  2022-01-01 01:00:00  4.90  981.60  3.60  3.60  340  0.00  0.00  100  91  0  0.00  0.00  n  2022-01-01 01:00:00  2021-12-31 19:00:00  Haze
2  2022-01-01 02:00:00  4.90  981.60  3.60  3.10  330  0.00  0.00  100  91  0  0.00  0.00  n  2022-01-01 02:00:00  2021-12-31 20:00:00  Haze
3  2022-01-01 03:00:00  4.80  982.20  3.50  4.10  330  0.00  0.00  100  91  0  0.00  0.00  n  2022-01-01 03:00:00  2021-12-31 21:00:00  Haze
4  2022-01-01 04:00:00  4.50  982.90  3.30  5.10  340  0.25  0.00  100  92  0  0.00  0.00  n  2022-01-01 04:00:00  2021-12-31 22:00:00  Haze

Out[ ]:
      datetime  temp  pres  dewpt  wind_spd  wind_dir  precip  snow  clouds  rh  solar_rad  uv  vis  pod  timestamp_utc  timestamp_local  description
0  2022-01-01 00:00:00  12.40  1,009.82  10.10  1.50  160  0.50  0.00  100  86  0  0.00  NaN  n  2022-01-01 00:00:00  2021-12-31 19:00:00  Overcast clouds
1  2022-01-01 01:00:00  13.30  1,002.50  11.20  0.50  130  0.00  0.00  100  87  0  0.00  2.00  n  2022-01-01 01:00:00  2021-12-31 20:00:00  Haze
2  2022-01-01 02:00:00  13.10  1,002.20  11.00  2.60  160  3.00  0.00  100  87  0  0.00  2.00  n  2022-01-01 02:00:00  2021-12-31 21:00:00  Light rain
3  2022-01-01 03:00:00  13.30  1,002.20  12.20  1.50  220  0.50  0.00  100  93  0  0.00  10.00  n  2022-01-01 03:00:00  2021-12-31 22:00:00  Overcast clouds
4  2022-01-01 04:00:00  12.80  1,002.20  11.70  2.10  160  0.50  0.00  100  93  0  0.00  10.00  n  2022-01-01 04:00:00  2021-12-31 23:00:00  Overcast clouds

Out[ ]:
      datetime  temp  pres  dewpt  wind_spd  wind_dir  precip  snow  clouds  rh  solar_rad  uv  vis  pod  timestamp_utc  timestamp_local  description
0  2022-01-01 00:00:00  10.00  1,013.10  8.30  2.10  210  0.00  0.00  100  89  0  0.00  11  n  2022-01-01 00:00:00  2021-12-31 19:00:00  Overcast clouds
1  2022-01-01 01:00:00  10.60  1,013.10  8.20  1.50  160  0.00  0.00  100  85  0  0.00  10  n  2022-01-01 01:00:00  2021-12-31 20:00:00  Overcast clouds
2  2022-01-01 02:00:00  10.00  1,012.80  8.30  2.00  160  0.00  0.00  100  89  0  0.00  10  n  2022-01-01 02:00:00  2021-12-31 21:00:00  Overcast clouds
3  2022-01-01 03:00:00  10.00  1,012.50  8.30  2.00  160  0.00  0.00  100  89  0  0.00  10  n  2022-01-01 03:00:00  2021-12-31 22:00:00  Overcast clouds
4  2022-01-01 04:00:00  10.00  1,012.10  8.30  2.10  160  0.00  0.00  100  89  0  0.00  11  n  2022-01-01 04:00:00  2021-12-31 23:00:00  Overcast clouds
```

Out[] :

	datetime	temp	pres	dewpt	wind_spd	wind_dir	precip	snow	clouds	rh	solar_rad	uv	vis	pod	timestamp_utc	timestamp_local	description
0	2022-01-01 00:00:00	-3.00	809.70	-3.00	4.60	60	0.50	10.00	100	100	0	0.00	1	n	2022-01-01 00:00:00	2021-12-31 17:00:00	Light snow
1	2022-01-01 01:00:00	-3.00	814.90	-4.10	5.10	40	0.50	10.00	100	92	0	0.00	1	n	2022-01-01 01:00:00	2021-12-31 18:00:00	Light snow
2	2022-01-01 02:00:00	-7.00	814.90	-8.10	5.70	350	1.00	25.00	100	92	0	0.00	1	n	2022-01-01 02:00:00	2021-12-31 19:00:00	Light snow
3	2022-01-01 03:00:00	-8.00	814.90	-9.10	4.10	360	1.00	30.00	100	92	0	0.00	2	n	2022-01-01 03:00:00	2021-12-31 20:00:00	Light snow
4	2022-01-01 04:00:00	-9.00	814.90	-10.10	2.10	10	1.00	30.00	100	92	0	0.00	2	n	2022-01-01 04:00:00	2021-12-31 21:00:00	Light snow

In [] :

```
# checking for NaN
syWeather.isna().sum()
chiWeather.isna().sum()
wasWeather.isna().sum()
newWeather.isna().sum()
denWeather.isna().sum()

chiWeather.dropna(inplace=True)
wasWeather.dropna(inplace=True)
```

Out[] :

datetime	0
temp	0
pres	0
dewpt	0
wind_spd	0
wind_dir	0
precip	0
snow	0
clouds	0
rh	0
solar_rad	0
uv	0
vis	0
pod	0
timestamp_utc	0
timestamp_local	0
description	0
dtype:	int64

Out[] :

datetime	0
temp	0
pres	0
dewpt	0
wind_spd	0
wind_dir	0
precip	0
snow	0
clouds	0
rh	0
solar_rad	0
uv	0
vis	1
pod	0
timestamp_utc	0
timestamp_local	0
description	0
dtype:	int64

Out[] :

datetime	0
temp	0
pres	0
dewpt	0
wind_spd	0
wind_dir	0
precip	0
snow	0
clouds	0
rh	0
solar_rad	0
uv	0
vis	1
pod	0
timestamp_utc	0
timestamp_local	0
description	0
dtype:	int64

Out[] :

datetime	0
temp	0
pres	0
dewpt	0
wind_spd	0
wind_dir	0
precip	0
snow	0
clouds	0
rh	0
solar_rad	0
uv	0
vis	0
pod	0
timestamp_utc	0
timestamp_local	0
description	0
dtype:	int64

Out[] :

datetime	0
temp	0
pres	0
dewpt	0
wind_spd	0
wind_dir	0
precip	0
snow	0
clouds	0
rh	0
solar_rad	0
uv	0
vis	0
pod	0
timestamp_utc	0
timestamp_local	0
description	0
dtype:	int64

In [] :

```
# Cleaning the weather data

denWeather['Origin Airport'] = 'DEN'
chiWeather['Origin Airport'] = 'ORD'
wasWeather['Origin Airport'] = 'IAD'
newWeather['Origin Airport'] = 'EWR'
syWeather['Destination Airport'] = 'SVR'

syWeather['date'] = syWeather['timestamp_local'].dt.date
syWeather['time'] = syWeather['timestamp_local'].dt.time

chiWeather['date'] = chiWeather['timestamp_local'].dt.date
chiWeather['time'] = chiWeather['timestamp_local'].dt.time

wasWeather['date'] = wasWeather['timestamp_local'].dt.date
wasWeather['time'] = wasWeather['timestamp_local'].dt.time

newWeather['date'] = newWeather['timestamp_local'].dt.date
newWeather['time'] = newWeather['timestamp_local'].dt.time

denWeather['date'] = denWeather['timestamp_local'].dt.date
denWeather['time'] = denWeather['timestamp_local'].dt.time

syWeather.drop(columns=['datetime', 'timestamp_utc', 'uv', 'solar_rad', 'clouds', 'pod', 'timestamp_local'],inplace = True)
chiWeather.drop(columns=['datetime', 'timestamp_utc', 'uv', 'solar_rad', 'clouds', 'pod', 'timestamp_local'],inplace = True)
wasWeather.drop(columns=['datetime', 'timestamp_utc', 'uv', 'solar_rad', 'clouds', 'pod', 'timestamp_local'],inplace = True)
newWeather.drop(columns=['datetime', 'timestamp_utc', 'uv', 'solar_rad', 'clouds', 'pod', 'timestamp_local'],inplace = True)
```

```
denWeather.drop(columns=['datetime', 'timestamp_utc', 'uv', 'solar_rad', 'clouds', 'pod', 'timestamp_local'],inplace = True)

syrWeather.head()
chiWeather.head()
wasWeather.head()
newWeather.head()
denWeather.head()

Out[ ] :
temp    pres    dewpt    wind_spd    wind_dir    precip    snow    rh    vis    description    Destination Airport    date    time
0    5.00    996.70    4.30    1.50    140    0.00    0.00    95    16    Clear Sky    SYR    2021-12-31    19:00:00
1    3.30    996.40    3.30    2.10    110    0.00    0.00    100    14    Fog    SYR    2021-12-31    20:00:00
2    2.20    996.00    1.60    1.50    90    0.00    0.00    96    0    Haze    SYR    2021-12-31    21:00:00
3    1.70    995.70    1.70    2.60    110    0.00    0.00    100    11    Fog    SYR    2021-12-31    22:00:00
4    1.70    995.00    1.70    2.60    80    0.00    0.00    100    13    Fog    SYR    2021-12-31    23:00:00

Out[ ] :
temp    pres    dewpt    wind_spd    wind_dir    precip    snow    rh    vis    description    Origin Airport    date    time
1    4.90    981.60    3.60    3.60    340    0.00    0.00    91    0.00    Haze    ORD    2021-12-31    19:00:00
2    4.90    981.60    3.60    3.10    330    0.00    0.00    91    0.00    Haze    ORD    2021-12-31    20:00:00
3    4.80    982.20    3.50    4.10    330    0.00    0.00    91    0.00    Haze    ORD    2021-12-31    21:00:00
4    4.50    982.90    3.30    5.10    340    0.25    0.00    92    0.00    Haze    ORD    2021-12-31    22:00:00
5    3.70    983.10    2.70    5.10    350    0.00    0.00    93    0.00    Haze    ORD    2021-12-31    23:00:00

Out[ ] :
temp    pres    dewpt    wind_spd    wind_dir    precip    snow    rh    vis    description    Origin Airport    date    time
1    13.30    1002.50    11.20    0.50    130    0.00    0.00    87    2.00    Haze    IAD    2021-12-31    20:00:00
2    13.10    1002.20    11.00    2.60    160    3.00    0.00    87    2.00    Light rain    IAD    2021-12-31    21:00:00
3    13.30    1002.20    12.20    1.50    220    0.50    0.00    93    10.00    Overcast clouds    IAD    2021-12-31    22:00:00
4    12.80    1002.20    11.70    2.10    160    0.50    0.00    93    10.00    Overcast clouds    IAD    2021-12-31    23:00:00
5    12.20    1001.50    10.10    1.00    150    0.00    0.00    87    10.00    Overcast clouds    IAD    2022-01-01    00:00:00

Out[ ] :
temp    pres    dewpt    wind_spd    wind_dir    precip    snow    rh    vis    description    Origin Airport    date    time
0    10.00    1013.10    8.30    2.10    210    0.00    0.00    89    11    Overcast clouds    EWR    2021-12-31    19:00:00
1    10.60    1013.10    8.20    1.50    160    0.00    0.00    85    10    Overcast clouds    EWR    2021-12-31    20:00:00
2    10.00    1012.80    8.30    2.00    160    0.00    0.00    89    10    Overcast clouds    EWR    2021-12-31    21:00:00
3    10.00    1012.50    8.30    2.00    160    0.00    0.00    89    10    Overcast clouds    EWR    2021-12-31    22:00:00
4    10.00    1012.10    8.30    2.10    160    0.00    0.00    89    11    Overcast clouds    EWR    2021-12-31    23:00:00

Out[ ] :
temp    pres    dewpt    wind_spd    wind_dir    precip    snow    rh    vis    description    Origin Airport    date    time
0    -3.00    809.70    -3.00    4.60    60    0.50    10.00    100    1    Light snow    DEN    2021-12-31    17:00:00
1    -3.00    814.90    -4.10    5.10    40    0.50    10.00    92    1    Light snow    DEN    2021-12-31    18:00:00
2    -7.00    814.90    -8.10    5.70    350    1.00    25.00    92    1    Light snow    DEN    2021-12-31    19:00:00
3    -8.00    814.90    -9.10    4.10    360    1.00    30.00    92    2    Light snow    DEN    2021-12-31    20:00:00
4    -9.00    814.90    -10.10    2.10    10    1.00    30.00    92    2    Light snow    DEN    2021-12-31    21:00:00

In [ ] : # Merging the weather data with flight data

flightData['Scheduled Arrival Time'] = flightData['Scheduled Arrival Time'].dt.round('H')
flightData['Scheduled departure time'] = flightData['Scheduled departure time'].dt.round('H')
flightData['Scheduled Arrival Time'] = flightData['Scheduled Arrival Time'].dt.time
flightData['Scheduled departure time'] = flightData['Scheduled departure time'].dt.time
flightData.drop(columns = ['Source Airport'], inplace = True)
flightData
flightData.dtypes

Out[ ] :
Date (MM/DD/YYYY)    Flight Number    Origin Airport    Scheduled Arrival Time    Scheduled Elapsed Time (Minutes)    Arrival Delay (Minutes)    Taxi-In time (Minutes)    Destination Airport    Scheduled departure time    Departure delay (Minutes)    Taxi-Out time (Minutes)
0    2022-01-01    1282    IAD    23:00:00    70    51    6    SYR    22:00:00    45    21
1    2022-01-02    1282    IAD    23:00:00    70    17    8    SYR    22:00:00    23    9
2    2022-01-03    1282    IAD    23:00:00    70    21    6    SYR    22:00:00    23    13
3    2022-01-04    1282    IAD    00:00:00    69    135    4    SYR    23:00:00    115    43
4    2022-01-05    1282    IAD    00:00:00    69    -14    5    SYR    23:00:00    -6    12
...    ...    ...    ...    ...    ...    ...    ...    ...    ...
1033    2022-12-30    1998    ORD    21:00:00    113    -11    5    SYR    18:00:00    -3    22
1034    2022-12-30    2488    EWR    23:00:00    75    -7    4    SYR    22:00:00    18    14
1035    2022-12-31    604    DEN    15:00:00    193    -12    7    SYR    10:00:00    3    12
1036    2022-12-31    1998    ORD    21:00:00    113    -24    4    SYR    18:00:00    -9    16
1037    2022-12-31    2488    EWR    23:00:00    75    92    6    SYR    22:00:00    94    30

1038 rows x 11 columns

Out[ ] :
Date (MM/DD/YYYY)    datetime64[ns]
Flight Number    int64
Origin Airport    object
Scheduled Arrival Time    object
Scheduled Elapsed Time (Minutes)    int64
Arrival Delay (Minutes)    int64
Taxi-In time (Minutes)    int64
Destination Airport    object
Scheduled departure time    object
Departure delay (Minutes)    int64
Taxi-Out time (Minutes)    int64
dtype: object

In [ ] :
sourceColumnNames = {'temp': 'temp_s', 'pres': 'pres_s', 'dewpt': 'dewpt_s', 'wind_spd': 'wind_spd_s', 'wind_dir': 'wind_dir_s',
    , 'precip': 'precip_s', 'snow': 'snow_s', 'rh': 'rh_s', 'vis': 'vis_s', 'description': 'description_s'
    , 'date': 'date_s', 'time': 'time_s', 'Origin Airport': 'Origin Airport'}
destinationColumnNames = {'temp': 'temp_d', 'pres': 'pres_d', 'dewpt': 'dewpt_d', 'wind_spd': 'wind_spd_d', 'wind_dir': 'wind_dir_d'
    , 'precip': 'precip_d', 'snow': 'snow_d', 'rh': 'rh_d', 'vis': 'vis_d', 'description': 'description_d'
    , 'date': 'date_d', 'time': 'time_d', 'Destination Airport': 'Destination Airport'}

syrWeather = syrWeather.rename(columns=destinationColumnNames)
denWeather = denWeather.rename(columns=sourceColumnNames)
wasWeather = wasWeather.rename(columns=sourceColumnNames)
chiWeather = chiWeather.rename(columns=sourceColumnNames)
newWeather = newWeather.rename(columns=sourceColumnNames)

syrWeather['date_d'] = pd.to_datetime(syrWeather['date_d'])
denWeather['date_s'] = pd.to_datetime(denWeather['date_s'])
wasWeather['date_s'] = pd.to_datetime(wasWeather['date_s'])
chiWeather['date_s'] = pd.to_datetime(chiWeather['date_s'])
newWeather['date_s'] = pd.to_datetime(newWeather['date_s'])

syrWeather.head()
chiWeather.head()
wasWeather.head()
newWeather.head()
denWeather.head()

originWeatherData = pd.concat([denWeather, wasWeather, chiWeather, newWeather], ignore_index=True)
originWeatherData.shape
originWeatherData
```

Out[]:

	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	vis_d	description_d	Destination Airport	date_d	time_d
0	5.00	996.70	4.30	1.50	140	0.00	0.00	95	16	Clear Sky	SYR	2021-12-31	19:00:00
1	3.30	996.40	3.30	2.10	110	0.00	0.00	100	14	Fog	SYR	2021-12-31	20:00:00
2	2.20	996.00	1.60	1.50	90	0.00	0.00	96	0	Haze	SYR	2021-12-31	21:00:00
3	1.70	995.70	1.70	2.60	110	0.00	0.00	100	11	Fog	SYR	2021-12-31	22:00:00
4	1.70	995.00	1.70	2.60	80	0.00	0.00	100	13	Fog	SYR	2021-12-31	23:00:00

Out[]:

	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	Origin Airport	date_s	time_s
1	4.90	981.60	3.60	3.60	340	0.00	0.00	91	0.00	Haze	ORD	2021-12-31	19:00:00
2	4.90	981.60	3.60	3.10	330	0.00	0.00	91	0.00	Haze	ORD	2021-12-31	20:00:00
3	4.80	982.20	3.50	4.10	330	0.00	0.00	91	0.00	Haze	ORD	2021-12-31	21:00:00
4	4.50	982.90	3.30	5.10	340	0.25	0.00	92	0.00	Haze	ORD	2021-12-31	22:00:00
5	3.70	983.10	2.70	5.10	350	0.00	0.00	93	0.00	Haze	ORD	2021-12-31	23:00:00

Out[]:

	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	Origin Airport	date_s	time_s
1	13.30	1,002.50	11.20	0.50	130	0.00	0.00	87	2.00	Haze	IAD	2021-12-31	20:00:00
2	13.10	1,002.20	11.00	2.60	160	3.00	0.00	87	2.00	Light rain	IAD	2021-12-31	21:00:00
3	13.30	1,002.20	12.20	1.50	220	0.50	0.00	93	10.00	Overcast clouds	IAD	2021-12-31	22:00:00
4	12.80	1,002.20	11.70	2.10	160	0.50	0.00	93	10.00	Overcast clouds	IAD	2021-12-31	23:00:00
5	12.20	1,001.50	10.10	1.00	150	0.00	0.00	87	10.00	Overcast clouds	IAD	2022-01-01	00:00:00

Out[]:

	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	Origin Airport	date_s	time_s
0	10.00	1,013.10	8.30	2.10	210	0.00	0.00	89	11	Overcast clouds	EWR	2021-12-31	19:00:00
1	10.60	1,013.10	8.20	1.50	160	0.00	0.00	85	10	Overcast clouds	EWR	2021-12-31	20:00:00
2	10.00	1,012.80	8.30	2.00	160	0.00	0.00	89	10	Overcast clouds	EWR	2021-12-31	21:00:00
3	10.00	1,012.50	8.30	2.00	160	0.00	0.00	89	10	Overcast clouds	EWR	2021-12-31	22:00:00
4	10.00	1,012.10	8.30	2.10	160	0.00	0.00	89	11	Overcast clouds	EWR	2021-12-31	23:00:00

Out[]:

	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	Origin Airport	date_s	time_s
0	-3.00	809.70	-3.00	4.60	60	0.50	10.00	100	1	Light snow	DEN	2021-12-31	17:00:00
1	-3.00	814.90	-4.10	5.10	40	0.50	10.00	92	1	Light snow	DEN	2021-12-31	18:00:00
2	-7.00	814.90	-8.10	5.70	350	1.00	25.00	92	1	Light snow	DEN	2021-12-31	19:00:00
3	-8.00	814.90	-9.10	4.10	360	1.00	30.00	92	2	Light snow	DEN	2021-12-31	20:00:00
4	-9.00	814.90	-10.10	2.10	10	1.00	30.00	92	2	Light snow	DEN	2021-12-31	21:00:00

Out[]: (34942, 13)

Out[]:

	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	Origin Airport	date_s	time_s
0	-3.00	809.70	-3.00	4.60	60	0.50	10.00	100	1.00	Light snow	DEN	2021-12-31	17:00:00
1	-3.00	814.90	-4.10	5.10	40	0.50	10.00	92	1.00	Light snow	DEN	2021-12-31	18:00:00
2	-7.00	814.90	-8.10	5.70	350	1.00	25.00	92	1.00	Light snow	DEN	2021-12-31	19:00:00
3	-8.00	814.90	-9.10	4.10	360	1.00	30.00	92	2.00	Light snow	DEN	2021-12-31	20:00:00
4	-9.00	814.90	-10.10	2.10	10	1.00	30.00	92	2.00	Light snow	DEN	2021-12-31	21:00:00
...
34937	16.10	1,021.90	1.40	3.60	170	0.00	0.00	37	16.00	Broken clouds	EWR	2022-12-30	14:00:00
34938	16.70	1,021.60	2.00	2.60	200	0.00	0.00	37	16.00	Broken clouds	EWR	2022-12-30	15:00:00
34939	16.10	1,021.30	2.20	4.10	190	0.00	0.00	39	16.00	Overcast clouds	EWR	2022-12-30	16:00:00
34940	14.40	1,021.60	2.00	2.60	170	0.00	0.00	43	16.00	Overcast clouds	EWR	2022-12-30	17:00:00
34941	12.20	1,021.90	2.60	2.10	160	0.00	0.00	52	16.00	Overcast clouds	EWR	2022-12-30	18:00:00

34942 rows × 13 columns

In []:

```
finalData = pd.merge(flightData, syrWeather, right_on=['Destination Airport', 'date_d', 'time_d'], left_on=['Destination Airport', 'Date (MM/DD/YYYY)', 'Scheduled Arrival Time'])
finalData = pd.merge(finalData, originWeatherData, right_on=['Origin Airport', 'date_s', 'time_s'], left_on=['Origin Airport', 'Date (MM/DD/YYYY)', 'Scheduled departure time'])
finalData
```

Out[]:

	Date (MM/DD/YYYY)	Flight Number	Origin Airport	Scheduled Arrival Time	Scheduled Elapsed Time (Minutes)	Arrival Delay (Minutes)	Taxi-In time (Minutes)	Destination Airport	Scheduled departure time	Departure delay (Minutes)	...	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	date_s	time_s
0	2022-01-01	1282	IAD	23:00:00	70	51	6	SYR	22:00:00	45	...	14.30	2.10	350	3.00	0.00	88	10.00	Light rain	2022-01-01	22:00:00
1	2022-01-02	1282	IAD	23:00:00	70	17	8	SYR	22:00:00	23	...	1.70	2.10	350	0.25	0.00	57	10.00	Overcast clouds	2022-01-02	22:00:00
2	2022-01-03	1282	IAD	23:00:00	70	21	6	SYR	22:00:00	23	...	-7.70	2.80	65	0.00	0.00	63	10.00	Scattered clouds	2022-01-03	22:00:00
3	2022-01-04	1282	IAD	00:00:00	69	135	4	SYR	23:00:00	115	...	-4.20	1.00	140	0.00	0.00	80	10.00	Broken clouds	2022-01-04	23:00:00
4	2022-01-05	1282	IAD	00:00:00	69	-14	5	SYR	23:00:00	-6	...	1.90	2.10	180	0.00	0.00	80	10.00	Overcast clouds	2022-01-05	23:00:00
...
1028	2022-12-28	2488	EWR	23:00:00	75	5	7	SYR	22:00:00	13	...	-4.60	3.60	230	0.00	0.00	52	16.00	Overcast clouds	2022-12-28	22:00:00
1029	2022-12-29	604	DEN	15:00:00	193	76	5	SYR	10:00:00	72	...	-7.10	3.10	320	0.00	0.00	63	16.00	Broken clouds	2022-12-29	10:00:00
1030	2022-12-29	1998	ORD	21:00:00	113	21	6	SYR	18:00:00	17	...	9.50	7.80	190	0.00	0.00	78	16.00	Overcast clouds	2022-12-29	18:00:00
1031	2022-12-29	2488	EWR	23:00:00	75	-23	6	SYR	22:00:00	-3	...	-2.20	2.10	210	0.00	0.00	62	16.00	Scattered clouds	2022-12-29	22:00:00
1032	2022-12-30	604	DEN	15:00:00	193	17	5	SYR	10:00:00	9	...	-11.00	3.60	160	0.00	0.00	54	16.00	Scattered clouds	2022-12-30	10:00:00

1033 rows × 35 columns

In []:

```
# Cleaning the final data
finalData.columns
finalData.drop(columns = ['Date (MM/DD/YYYY)', 'Destination Airport', 'date_d', 'time_d', 'date_s', 'time_s'], inplace = True)
finalData
```

Out[]:

```
Index(['Date (MM/DD/YYYY)', 'Flight Number', 'Origin Airport',  
      'Scheduled Arrival Time', 'Scheduled Elapsed Time (Minutes)',  
      'Arrival Delay (Minutes)', 'Taxi-In time (Minutes)',  
      'Destination Airport', 'Scheduled departure time',  
      'Departure delay (Minutes)', 'Taxi-Out time (Minutes)', 'temp_d',  
      'pres_d', 'dewpt_d', 'wind_spd_d', 'wind_dir_d', 'precip_d', 'snow_d',  
      'rh_d', 'vis_d', 'description_d', 'date_d', 'time_d', 'temp_s',  
      'pres_s', 'dewpt_s', 'wind_spd_s', 'wind_dir_s', 'precip_s', 'snow_s',  
      'rh_s', 'vis_s', 'description_s', 'date_s', 'time_s'],  
      dtype='object')
```

Out[]:

	Flight Number	Origin Airport	Scheduled Arrival Time	Scheduled Elapsed Time (Minutes)	Arrival Delay (Minutes)	Taxi-In time (Minutes)	Scheduled departure time	Departure delay (Minutes)	Taxi-Out time (Minutes)	temp_d	temp_s	pres_s	dewpt_s	wind_spd_s	wind_dir_s	precip_s	snow_s	rh_s	vis_s	description_s	
0	1282	IAD	23:00:00	70	51	6	22:00:00	45	21	3.30	...	16.30	991.70	14.30	2.10	350	3.00	0.00	88	10.00	Light rain
1	1282	IAD	23:00:00	70	17	8	22:00:00	23	9	-7.20	...	9.80	1,003.90	1.70	2.10	350	0.25	0.00	57	10.00	Overcast clouds
2	1282	IAD	23:00:00	70	21	6	22:00:00	23	13	-11.10	...	-1.60	1,016.10	-7.70	2.80	65	0.00	0.00	63	10.00	Scattered clouds
3	1282	IAD	00:00:00	69	135	4	23:00:00	115	43	-11.40	...	-1.20	1,016.40	-4.20	1.00	140	0.00	0.00	80	10.00	Broken clouds
4	1282	IAD	00:00:00	69	-14	5	23:00:00	-6	12	-1.70	...	5.10	998.10	1.90	2.10	180	0.00	0.00	80	10.00	Overcast clouds
...
1028	2488	EWB	23:00:00	75	5	7	22:00:00	13	21	2.20	...	4.40	1,025.30	-4.60	3.60	230	0.00	0.00	52	16.00	Overcast clouds
1029	604	DEN	15:00:00	193	76	5	10:00:00	72	17	10.60	...	-1.00	818.00	-7.10	3.10	320	0.00	0.00	63	16.00	Broken clouds
1030	1998	ORD	21:00:00	113	21	6	18:00:00	17	30	10.00	...	13.30	988.60	9.50	7.80	190	0.00	0.00	78	16.00	Overcast clouds
1031	2488	EWB	23:00:00	75	-23	6	22:00:00	-3	15	10.00	...	4.40	1,025.00	-2.20	2.10	210	0.00	0.00	62	16.00	Scattered clouds
1032	604	DEN	15:00:00	193	17	5	10:00:00	9	26	17.20	...	-3.00	820.00	-11.00	3.60	160	0.00	0.00	54	16.00	Scattered clouds

1033 rows x 29 columns

In []:

```
set(finalData['description_s'])
set(finalData['description_d'])

finalData['description_s'] = finalData['description_s'].replace({'Broken clouds': 'Cloudy', 'Few clouds': 'Cloudy', 'Fog': 'Cloudy', 'Haze': 'Cloudy', 'Overcast clouds': 'Cloudy',
'Scattered clouds': 'Cloudy', 'Heavy rain': 'Rainy', 'Light rain': 'Rainy', 'Moderate rain': 'Rainy',
'Thunderstorm with heavy rain': 'Rainy', 'Light snow': 'Snow', 'Mix snow/rain': 'Snow',
'Snow': 'Snow', 'Clear Sky': 'Clear'})

finalData['description_d'] = finalData['description_d'].replace({'Broken clouds': 'Cloudy', 'Few clouds': 'Cloudy', 'Fog': 'Cloudy', 'Haze': 'Cloudy', 'Overcast clouds': 'Cloudy',
'Scattered clouds': 'Cloudy', 'Heavy rain': 'Rainy', 'Light rain': 'Rainy', 'Moderate rain': 'Rainy',
'Thunderstorm with heavy rain': 'Rainy', 'Light snow': 'Snow', 'Mix snow/rain': 'Snow',
'Snow': 'Snow', 'Flurries': 'Snow', 'Clear Sky': 'Clear', 'Moderate rain': 'Rainy', 'Heavy snow': 'Snow'})

set(finalData['description_s'])
set(finalData['description_d'])
```

Out[]:

{'Broken clouds',
'Clear Sky',
'Few clouds',
'Fog',
'Haze',
'Heavy rain',
'Light rain',
'Light snow',
'Mix snow/rain',
'Moderate rain',
'Overcast clouds',
'Scattered clouds',
'Snow',
'Thunderstorm with heavy rain'}

Out[]:

{'Broken clouds',
'Clear Sky',
'Few clouds',
'Flurries',
'Fog',
'Haze',
'Heavy rain',
'Heavy snow',
'Light rain',
'Light snow',
'Mix snow/rain',
'Moderate rain',
'Overcast clouds',
'Scattered clouds',
'Snow',
'Thunderstorm with heavy rain'}

Out[]:

{'Clear', 'Cloudy', 'Rainy', 'Snow'}

Out[]:

{'Clear', 'Cloudy', 'Rainy', 'Snow'}

In []:

```
from sklearn.preprocessing import OneHotEncoder

def getOhe(df, col):
    ohe = OneHotEncoder(drop='first', handle_unknown='error', sparse=False, dtype='int')
    ohe.fit(df[[col]])
    temp_df = pd.DataFrame(data=ohe.transform(df[[col]]), columns=ohe.get_feature_names_out())
    # If you have a newer version, replace with columns=ohe.get_feature_names_out()
    df.drop(columns=[col], axis=1, inplace=True)
    df = pd.concat([df.reset_index(drop=True), temp_df], axis=1)
    return df
```

In []:

```
subsetData = getOhe(finalData, 'description_d')
subsetData = getOhe(subsetData, 'description_s')
subsetData = getOhe(subsetData, 'Origin Airport')
subsetData.columns
subsetData.drop(columns = ['Taxi-In time (Minutes)', 'Departure delay (Minutes)', 'Taxi-Out time (Minutes)', 'Scheduled Arrival Time', 'Scheduled departure time'], inplace= True)
# subsetData['Scheduled Arrival Time'] = subsetData['Scheduled Arrival Time'].apply(lambda x: int(x.strftime('%H%M%S')))
# subsetData['Scheduled departure time'] = subsetData['Scheduled departure time'].apply(lambda x: int(x.strftime('%H%M%S')))

subsetData.dtypes
```

Out[]:

Index(['Flight Number', 'Scheduled Arrival Time',
Scheduled Elapsed Time (Minutes)', 'Arrival Delay (Minutes)',
Taxi-In time (Minutes)', 'Scheduled departure time',
Departure delay (Minutes)', 'Taxi-Out time (Minutes)', 'temp_d',
'pres_d', 'dewpt_d', 'wind_spd_d', 'wind_dir_d', 'precip_d', 'snow_d',
'rh_d', 'vis_d', 'temp_s', 'pres_s', 'dewpt_s', 'wind_spd_s',
'wind_dir_s', 'precip_s', 'snow_s', 'rh_s', 'vis_s',
'description_d_Cloudy', 'description_d_Rainy', 'description_d_Snow',
'description_s_Cloudy', 'description_s_Rainy', 'description_s_Snow',
'Origin Airport_EWB', 'Origin Airport_IAD', 'Origin Airport_ORD'],
dtype='object')

```
Out[ ]: Flight Number      int64
Scheduled Elapsed Time (Minutes)  int64
Arrival Delay (Minutes)          int64
temp_d                          float64
pres_d                          float64
dewpt_d                         float64
wind_spd_d                      float64
wind_dir_d                     int64
precip_d                        float64
snow_d                         float64
rh_d                           int64
vis_d                          int64
temp_s                          float64
pres_s                          float64
dewpt_s                        float64
wind_spd_s                     float64
wind_dir_s                     int64
precip_s                       float64
snow_s                        float64
rh_s                           int64
vis_s                          float64
description_d_Cloudy            int32
description_d_Rainy             int32
description_d_Snow              int32
description_s_Cloudy            int32
description_s_Rainy             int32
description_s_Snow              int32
Origin Airport_EWR              int32
Origin Airport_IAD             int32
Origin Airport_ORD             int32
dtype: object

In [ ]: # Scaling the data
xTrain, xTest, yTrain, yTest = train_test_split(subsetData.drop(columns = ['Arrival Delay (Minutes)']), subsetData['Arrival Delay (Minutes)'], test_size=0.20, random_state=40)
xTrain
xTest
yTrain
yTest

Out[ ]:
      Scheduled Elapsed Time (Minutes)  temp_d  pres_d  dewpt_d  wind_spd_d  wind_dir_d  precip_d  snow_d  rh_d  ...  vis_s  description_d_Cloudy  description_d_Rainy  description_d_Snow  description_s_Cloudy  description_s_Rainy  description_s_Snow  Origin Airport_EWR  Origin Airport_IAD  Origin Airport_ORD
502      604      199      27.20      991.00      19.80      8.20      200      0.50      0.00      64  ...  16.00      1      0      0      1      0      0      0      0      0
957      1998      112      2.80      1,007.10      -5.10      6.20      280      0.00      0.00      56  ...  16.00      1      0      0      1      0      0      0      0      0
262      2027      70      16.70      999.40      7.60      5.70      180      0.00      0.00      55  ...  10.00      1      0      0      1      0      0      0      0      1
341      2051      111      26.70      998.00      15.40      4.10      310      0.00      0.00      50  ...  10.00      1      0      0      1      0      0      0      0      0
215      2027      70      3.90      992.30      -1.90      6.70      280      0.00      0.00      66  ...  10.00      1      0      0      0      0      0      0      0      1
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...  ...  ...      ...      ...      ...      ...      ...      ...      ...
626      2615      73      20.60      999.40      14.90      2.40      230      0.00      0.00      70  ...  16.00      1      0      0      1      0      0      0      0      1
1016     2488      75      -10.00      988.30      -18.40      10.30      220      0.00      0.00      50  ...  16.00      1      0      0      1      0      0      0      1      0
165      1874      69      5.00      990.70      2.70      3.60      50      1.00      0.00      85  ...  2.00      0      1      0      1      0      0      0      0      1
7      1282      69      -9.70      1,011.05      -12.30      5.70      295      0.00      0.00      81  ...  10.00      1      0      0      0      0      0      0      0      1
219      1041      198      16.10      1,003.40      -3.90      6.70      280      0.00      0.00      25  ...  16.00      1      0      0      1      0      0      0      0      0

826 rows x 29 columns

Out[ ]:
      Scheduled Elapsed Time (Minutes)  temp_d  pres_d  dewpt_d  wind_spd_d  wind_dir_d  precip_d  snow_d  rh_d  ...  vis_s  description_d_Cloudy  description_d_Rainy  description_d_Snow  description_s_Cloudy  description_s_Rainy  description_s_Snow  Origin Airport_EWR  Origin Airport_IAD  Origin Airport_ORD
466      2615      73      19.40      1,000.40      12.70      0.80      275      0.00      0.00      65  ...  16.00      1      0      0      1      0      0      0      0      1
643      604      198      28.30      1,000.40      14.90      5.10      310      0.00      0.00      44  ...  16.00      1      0      0      1      0      0      0      0      0
937      604      194      6.70      995.70      1.60      7.20      280      0.00      0.00      70  ...  16.00      1      0      0      0      0      0      0      0      0
498      604      199      30.60      999.40      18.00      2.10      30      0.00      0.00      47  ...  16.00      1      0      0      1      0      0      0      0      0
285      1041      198      12.20      999.00      10.90      2.60      350      0.25      0.00      92  ...  5.00      1      0      0      1      0      0      0      0      0
...      ...      ...      ...      ...      ...      ...      ...      ...      ...      ...  ...  ...      ...      ...      ...      ...      ...      ...
239      1041      198      3.30      989.70      -0.70      7.20      240      0.00      0.00      75  ...  16.00      1      0      0      1      0      0      0      0      0
1      1282      70      -7.20      1,005.40      -11.20      3.10      340      0.00      0.00      73  ...  10.00      1      0      0      1      0      0      0      0      1
15      1282      69      -18.60      1,016.10      -22.50      1.30      130      0.00      0.00      71  ...  2.00      1      0      0      0      1      0      0      0      1
694      604      200      22.20      1,001.70      17.80      3.60      40      0.00      0.00      76  ...  16.00      1      0      0      1      0      0      0      0      0
980      1998      114      0.00      1,003.40      -1.10      2.40      235      0.00      0.00      92  ...  16.00      1      0      0      1      0      0      0      0      0

207 rows x 29 columns

Out[ ]:
502      8
957      -5
262      77
341      56
215      3
...
626      2
1016     380
165      6
7      -6
219      -6
Name: Arrival Delay (Minutes), Length: 826, dtype: int64

Out[ ]:
466      -9
643      28
937      4
498      -19
285      -6
...
239      -6
1      17
15      0
694      -8
980      -18
Name: Arrival Delay (Minutes), Length: 207, dtype: int64

In [ ]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
xTrain = pd.DataFrame(sc.fit_transform(xTrain), columns = xTrain.columns, index = xTrain.index)
xTest = pd.DataFrame(sc.transform(xTest), columns = xTest.columns, index = xTest.index)
xTrain
yTrain
yTest
```

Out[]:

Flight Number	Scheduled Elapsed Time (Minutes)	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	vis_s	description_d_Cloudy	description_d_Rainy	description_d_Snow	description_s_Cloudy	description_s_Rainy	description_s_Snow	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_OR		
502	-1.07	1.41	1.30	-1.36	1.47	1.75	-0.14	0.34	-0.11	0.10	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
957	0.86	-0.32	-0.96	0.86	-1.07	0.90	0.78	-0.27	-0.11	-0.35	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	1.1
262	0.90	-1.15	0.33	-0.20	0.23	0.69	-0.38	-0.27	-0.11	-0.40	...	-1.14	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	1.72	-0.1
341	0.94	-0.34	1.25	-0.39	1.02	0.00	1.13	-0.27	-0.11	-0.68	...	-1.14	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	1.1
215	0.90	-1.15	-0.86	-1.18	-0.74	1.11	0.78	-0.27	-0.11	0.21	...	-1.14	0.42	-0.29	-0.16	-1.74	-0.26	-0.14	-0.18	1.72	-0.1
...
626	1.72	-1.09	0.69	-0.20	0.97	-0.72	0.20	-0.27	-0.11	0.43	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	1.72	-0.1
1016	1.54	-1.05	-2.14	-1.73	-2.43	2.65	0.09	-0.27	-0.11	-0.68	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	5.55	-0.58	-0.1
165	0.69	-1.17	-0.76	-1.40	-0.27	-0.21	-1.88	0.95	-0.11	1.27	...	-3.41	-2.39	3.48	-0.16	0.57	-0.26	-0.14	-0.18	1.72	-0.1
7	-0.13	-1.17	-2.12	1.41	-1.80	0.69	0.96	-0.27	-0.11	1.05	...	-1.14	0.42	-0.29	-0.16	-1.74	-0.26	-0.14	-0.18	1.72	-0.1
219	-0.46	1.39	0.27	0.35	-0.95	1.11	0.78	-0.27	-0.11	-2.08	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1

826 rows × 29 columns

Out[]:

Flight Number	Scheduled Elapsed Time (Minutes)	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	vis_s	description_d_Cloudy	description_d_Rainy	description_d_Snow	description_s_Cloudy	description_s_Rainy	description_s_Snow	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_OR		
466	1.72	-1.09	0.58	-0.06	0.75	-1.40	0.72	-0.27	-0.11	0.16	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	1.72	-0.1
643	-1.07	1.39	1.40	-0.06	0.97	0.43	1.13	-0.27	-0.11	-1.02	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
937	-1.07	1.31	-0.60	-0.71	-0.39	1.33	0.78	-0.27	-0.11	0.43	...	0.56	0.42	-0.29	-0.16	-1.74	-0.26	-0.14	-0.18	-0.58	-0.1
498	-1.07	1.41	1.62	-0.20	1.29	-0.85	-2.11	-0.27	-0.11	-0.85	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
285	-0.46	1.39	-0.09	-0.25	0.56	-0.64	1.59	0.04	-0.11	1.66	...	-2.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
...
239	-0.46	1.39	-0.91	-1.54	-0.62	1.33	0.32	-0.27	-0.11	0.71	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
1	-0.13	-1.15	-1.89	0.63	-1.69	-0.42	1.48	-0.27	-0.11	0.60	...	-1.14	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	1.72	-0.1
15	-0.13	-1.17	-2.94	2.11	-2.85	-1.19	-0.95	-0.27	-0.11	0.49	...	-3.41	0.42	-0.29	-0.16	-1.74	3.90	-0.14	-0.18	-0.58	-0.1
694	-1.07	1.43	0.84	0.12	1.27	-0.21	-2.00	-0.27	-0.11	0.77	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	-0.1
980	0.86	-0.28	-1.22	0.35	-0.66	-0.72	0.26	-0.27	-0.11	1.66	...	0.56	0.42	-0.29	-0.16	0.57	-0.26	-0.14	-0.18	-0.58	1.2

207 rows × 29 columns

Out[]:

502 8
957 -5
262 77
341 56
215 3
...
626 2
1016 380
165 6
7 -6
219 -6
Name: Arrival Delay (Minutes), Length: 826, dtype: int64

Out[]:

466 -9
643 28
937 4
498 -19
285 -6
...
239 -6
1 17
15 0
694 -8
980 -18
Name: Arrival Delay (Minutes), Length: 207, dtype: int64

In []:

```
# Linear Regression
modelLR = LinearRegression(fit_intercept = True)
modelLR.fit(xTrain, yTrain)
modelLR.score(xTrain, yTrain)
modelLR.coef_
modelLR.intercept_
```

Out[]: LinearRegression()

Out[]: 0.12660284486689242

Out[]: array([5.58287231, -49.66701073, -20.50783963, -4.15590261, 20.30866595, -2.09116916, 2.47561637, 1.05241551, 2.20822367, -11.47300879, -1.06995903, 9.09713295, -23.36618311, -5.49526383, 2.76959123, -0.4609661, -3.1647191, 3.06971284, 11.58043457, -4.01767378, -5.16863082, -5.57568456, -4.2864745, 1.95973776, 4.29638362, 4.45816428, -8.08710334, -35.31068112, -28.28439781])

Out[]: 7.5447941888619185

In []:

testOutputLR = pd.DataFrame(modelLR.predict(xTest), index = xTest.index, columns = ['predArrivalDelay'])
testOutputLR.head()

Out[]:

	predArrivalDelay
466	29.69
643	6.56
937	-7.83
498	0.73
285	43.39

In []:

testOutputLR = testOutputLR.merge(yTest, left_index = True, right_index = True)
testOutputLR.head()
mean_absolute_error = abs(testOutputLR['predArrivalDelay'] - testOutputLR['Arrival Delay (Minutes)']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)

Out[]:

	predArrivalDelay	Arrival Delay (Minutes)
466	29.69	-9
643	6.56	28
937	-7.83	4
498	0.73	-19
285	43.39	-6

Mean absolute error is
31.016880186271027

In []:

abs(testOutputLR['predArrivalDelay'] - testOutputLR['Arrival Delay (Minutes)']).mean()/testOutputLR['Arrival Delay (Minutes)'].mean()

Out[]:

2.1337634425251255

In []:

```
from sklearn.ensemble import RandomForestRegressor  
  
gb = RandomForestRegressor(random_state=50, min_samples_leaf = 3, max_features = "sqrt")
```



```
gb = gb.fit(xTrain, yTrain)

In [ ]: gb.feature_importances_

Out[ ]: array([0.05156407, 0.04626348, 0.05118878, 0.05242685, 0.06649194,
0.04867106, 0.03894583, 0.00748094, 0.00211279, 0.05116327,
0.00259291, 0.05267695, 0.05725776, 0.09339538, 0.06075506,
0.04775362, 0.06705325, 0.01991984, 0.09003049, 0.03616949,
0.00384873, 0.00134781, 0.00012764, 0.0045355 , 0.00677958,
0.02162802, 0.00170434, 0.00768361, 0.00843182])

In [ ]: testOutputGB = pd.DataFrame(gb.predict(xTest), index = xTest.index, columns = ['predArrivalDelay'])
testOutputGB.head()

Out[ ]:      predArrivalDelay
466          25.95
643          11.64
937          -1.92
498           1.48
285          15.02

In [ ]: testOutputGB = testOutputGB.merge(yTest, left_index = True, right_index = True)
testOutputGB.head()
mean_absolute_error = abs(testOutputGB['predArrivalDelay'] - testOutputGB['Arrival Delay (Minutes)']).mean()
print('Mean absolute error is ')
print(mean_absolute_error)
abs(testOutputGB['predArrivalDelay'] - testOutputGB['Arrival Delay (Minutes)']).mean()/testOutputGB['Arrival Delay (Minutes)'].mean()

Out[ ]:      predArrivalDelay  Arrival Delay (Minutes)
466          25.95          -9
643          11.64          28
937          -1.92           4
498           1.48         -19
285          15.02          -6

Mean absolute error is
29.828280736108514

Out[ ]: 2.051995384637575

In [ ]: # Reading Test Data
finalTesttData = pd.read_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/Data sets/project_test_data.csv")
finalTesttData.head()

Out[ ]:   Origin Airport  Flight Number  Scheduled Arrival Time  Scheduled departure time  Scheduled Elapsed Time  temp_d  pres_d  dewpt_d  wind_spd_d  wind_dir_d  ...  temp_s  pres_s  dewpt_s  wind_spd_s  wind_dir_s  precip_s  snow_s  rh_s  vis_s  description_s
0      ORD          3839          10:00:00          07:00:00          14:00:00          110    47.30    982.50    42.10    6.20    254  ...    42.70    993.00    39.40    8.00    248    0.00    0    88    9.40    Cloudy
1      ORD          3524          17:00:00          14:00:00          14:00:00          115    58.20    981.00    37.00    10.40    260  ...    56.50    991.00    38.10    15.00    224    0.00    0    50    24.20    Cloudy
2      ORD           538          22:00:00          19:00:00          19:00:00          114    52.50    978.50    37.70    4.60    163  ...    59.90    988.50    30.80    7.80    232    0.00    0    33    0.40    Cloudy
3      ORD          3839          10:00:00          07:00:00          14:00:00          110    43.70    976.00    38.00    13.90    292  ...    38.90    989.00    35.40    7.10    266    0.00    0    87    9.60    Cloudy
4      ORD          3524          17:00:00          14:00:00          14:00:00          115    46.70    976.50    31.00    15.70    298  ...    46.90    987.50    28.30    15.30    281    0.00    0    48    15.00    Cloudy

5 rows x 25 columns

In [ ]: finalTesttData['description_d'] = finalTesttData['description_d'].replace({'Light shower rain' : 'Rainy'})
finalTesttData

Out[ ]:   Origin Airport  Flight Number  Scheduled Arrival Time  Scheduled departure time  Scheduled Elapsed Time  temp_d  pres_d  dewpt_d  wind_spd_d  wind_dir_d  ...  temp_s  pres_s  dewpt_s  wind_spd_s  wind_dir_s  precip_s  snow_s  rh_s  vis_s  description_s
0      ORD          3839          10:00:00          07:00:00          14:00:00          110    47.30    982.50    42.10    6.20    254  ...    42.70    993.00    39.40    8.00    248    0.00    0    88    9.40    Cloudy
1      ORD          3524          17:00:00          14:00:00          14:00:00          115    58.20    981.00    37.00    10.40    260  ...    56.50    991.00    38.10    15.00    224    0.00    0    50    24.20    Cloudy
2      ORD           538          22:00:00          19:00:00          19:00:00          114    52.50    978.50    37.70    4.60    163  ...    59.90    988.50    30.80    7.80    232    0.00    0    33    0.40    Cloudy
3      ORD          3839          10:00:00          07:00:00          14:00:00          110    43.70    976.00    38.00    13.90    292  ...    38.90    989.00    35.40    7.10    266    0.00    0    87    9.60    Cloudy
4      ORD          3524          17:00:00          14:00:00          14:00:00          115    46.70    976.50    31.00    15.70    298  ...    46.90    987.50    28.30    15.30    281    0.00    0    48    15.00    Cloudy
5      ORD           538          22:00:00          19:00:00          19:00:00          114    52.50    978.50    37.70    4.60    163  ...    59.90    988.50    30.80    7.80    232    0.00    0    33    0.40    Cloudy
6      ORD          3839          10:00:00          07:00:00          14:00:00          110    43.70    976.00    38.00    13.90    292  ...    38.90    989.00    35.40    7.10    266    0.00    0    87    9.60    Cloudy
7      ORD          3524          17:00:00          14:00:00          14:00:00          115    46.70    976.50    31.00    15.70    298  ...    46.90    987.50    28.30    15.30    281    0.00    0    48    15.00    Cloudy
8      ORD           538          22:00:00          19:00:00          19:00:00          114    52.50    978.50    37.70    4.60    163  ...    59.90    988.50    30.80    7.80    232    0.00    0    33    0.40    Cloudy
9      ORD          3839          10:00:00          07:00:00          14:00:00          110    43.70    976.00    38.00    13.90    292  ...    38.90    989.00    35.40    7.10    266    0.00    0    87    9.60    Cloudy
10     ORD          3524          17:00:00          14:00:00          14:00:00          115    47.40    990.50    25.50    15.30    312  ...    53.70    999.00    18.80    9.10    298    0.00    0    25    15.00    Cloudy
11     ORD           538          22:00:00          19:00:00          19:00:00          114    36.00    991.50    25.10    5.60    359  ...    48.80    998.50    24.40    7.00    73    0.00    0    38    15.00    Cloudy
12     DEN           604          15:00:00          10:00:00          10:00:00          201    56.50    981.50    37.60    9.00    252  ...    70.20    988.00    46.00    10.20    225    0.00    0    42    0.30    Clear
13     DEN           604          15:00:00          10:00:00          10:00:00          201    47.50    976.00    38.60    13.20    304  ...    66.70    976.00    58.90    13.80    191    0.00    0    76    12.00    Cloudy
14     DEN           604          15:00:00          10:00:00          10:00:00          201    41.30    986.50    26.90    14.70    296  ...    54.60    987.50    31.40    9.50    316    0.00    0    41    15.00    Clear
15     DEN           604          15:00:00          10:00:00          10:00:00          201    48.80    990.50    26.80    15.30    301  ...    51.10    992.00    30.50    8.90    2    0.00    0    45    15.00    Cloudy
16     EWR          4189          11:00:00          09:00:00          09:00:00          91    48.50    982.50    41.30    5.40    259  ...    54.60    1,007.50    50.80    11.10    28    0.00    0    87    9.90    Cloudy
17     EWR          1412          00:00:00          22:00:00          09:00:00          72    60.30    980.50    57.00    12.00    238  ...    57.20    1,004.50    53.40    4.40    196    0.00    0    87    9.30    Cloudy
18     EWR          4189          11:00:00          09:00:00          09:00:00          91    44.00    977.00    36.20    12.80    280  ...    50.60    1,008.50    36.80    13.60    340    0.00    0    59    20.10    Clear
19     EWR          1412          23:00:00          22:00:00          22:00:00          72    38.30    980.00    32.00    11.40    286  ...    51.40    1,009.50    40.50    9.40    300    0.01    0    66    15.00    Cloudy
20     EWR          4189          11:00:00          09:00:00          09:00:00          91    37.60    986.00    30.40    12.40    287  ...    47.00    1,009.00    32.60    5.20    15    0.00    0    57    15.00    Cloudy
21     EWR          1412          00:00:00          22:00:00          22:00:00          72    37.00    980.50    31.80    12.60    286  ...    46.50    1,007.00    35.80    1.60    149    0.00    0    66    15.00    Cloudy
22     EWR          4189          11:00:00          09:00:00          09:00:00          91    45.60    990.50    26.60    11.30    297  ...    49.30    1,009.50    38.80    8.40    86    0.00    0    67    15.00    Cloudy
23     EWR          1412          00:00:00          22:00:00          22:00:00          72    32.70    988.50    24.40    3.20    270  ...    50.50    1,005.50    42.20    8.90    67    0.00    0    73    15.00    Cloudy
24     IAD          4490          14:00:00          13:00:00          13:00:00          74    54.00    982.00    39.10    8.90    242  ...    86.00    1,012.50    48.20    3.70    93    0.00    0    27    0.40    Clear
25     IAD          4165          19:00:00          18:00:00          18:00:00          83    57.50    979.00    38.00    8.50    280  ...    85.30    1,009.00    42.10    9.90    109    0.00    0    22    0.50    Cloudy
26     IAD          3805          14:00:00          13:00:00          13:00:00          75    45.40    976.50    33.20    14.70    258  ...    78.20    1,003.50    50.50    17.90    164    0.00    0    38    0.30    Cloudy
27     IAD          4165          19:00:00          18:00:00          18:00:00          83    43.50    978.00    30.20    15.10    298  ...    63.40    999.50    62.30    4.80    318    0.29    0    96    1.50    Rainy
28     IAD          4490          14:00:00          13:00:00          13:00:00          74    40.10    986.50    27.80    14.60    289  ...    63.90    1,005.00    39.70    10.40    310    0.00    0    41    15.00    Cloudy
29     IAD          4165          19:00:00          18:00:00          18:00:00          83    41.80    987.50    26.90    11.60    313  ...    64.20    1,005.50    36.00    11.50    323    0.00    0    35    15.00    Cloudy
30     IAD          4490          14:00:00          13:00:00          13:00:00          74    48.70    990.50    26.70    15.00    296  ...    61.50    1,009.00    31.40    13.80    317    0.00    0    32    15.00    Cloudy
31     IAD          4165          19:00:00          18:00:00          18:00:00          83    45.30    990.50    28.80    10.50    336  ...    61.20    1,009.00    28.70    11.50    326    0.00    0    29    15.00    Cloudy

32 rows x 25 columns

In [ ]: testSubsetData = getOhe(finalTesttData, 'description_d')
testSubsetData = getOhe(testSubsetData, 'description_s')
testSubsetData = getOhe(testSubsetData, 'Origin Airport')
testSubsetData.drop(columns=['Scheduled Arrival Time', 'Scheduled departure time'],inplace=True)
testSubsetData
```

Out []:

	Flight Number	Scheduled Elapsed Time	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	...	rh_s	vis_s	description_d_Cloudy	description_d_Rainy	description_d_Snow	description_s_Cloudy	description_s_Rainy	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_ORD
0	3839	110	47.30	982.50	42.10	6.20	254	0.00	0	82	...	88	9.40	1	0	0	1	0	0	0	1
1	3524	115	58.20	981.00	37.00	10.40	260	0.00	0	45	...	50	24.20	1	0	0	1	0	0	0	1
2	538	114	52.50	978.50	37.70	4.60	163	0.00	0	57	...	33	0.40	1	0	0	1	0	0	0	1
3	3839	110	43.70	976.00	38.00	13.90	292	0.00	0	80	...	87	9.60	1	0	0	1	0	0	0	1
4	3524	115	46.70	976.50	31.00	15.70	298	0.00	0	54	...	48	15.00	1	0	0	1	0	0	0	1
5	538	114	38.50	979.50	30.90	8.90	278	0.00	0	74	...	61	15.00	1	0	0	1	0	0	0	1
6	3839	110	36.40	985.00	30.50	12.30	280	0.00	0	79	...	73	15.00	0	0	1	1	0	0	0	1
7	3524	115	42.40	987.00	25.60	13.40	308	0.00	0	51	...	35	15.00	0	0	1	1	0	0	0	1
8	538	114	34.00	989.00	25.60	3.40	319	0.00	0	71	...	32	15.00	1	0	0	1	0	0	0	1
9	3839	110	43.30	990.50	28.30	9.70	300	0.00	0	55	...	52	15.00	1	0	0	0	0	0	0	1
10	3524	115	47.40	990.50	25.50	15.30	312	0.00	0	42	...	25	15.00	1	0	0	1	0	0	0	1
11	538	114	36.00	991.50	25.10	5.60	359	0.00	0	64	...	38	15.00	0	0	0	1	0	0	0	1
12	604	201	56.50	981.50	37.60	9.00	252	0.00	0	49	...	42	0.30	1	0	0	0	0	0	0	0
13	604	201	47.50	976.00	38.60	13.20	304	0.10	0	71	...	76	12.00	0	1	0	1	0	0	0	0
14	604	201	41.30	986.50	26.90	14.70	296	0.00	0	56	...	41	15.00	0	0	1	0	0	0	0	0
15	604	201	48.80	990.50	26.80	15.30	301	0.00	0	42	...	45	15.00	1	0	0	1	0	0	0	0
16	4189	91	48.50	982.50	41.30	5.40	259	0.00	0	76	...	87	9.90	1	0	0	1	0	1	0	0
17	1412	72	60.30	980.50	57.00	12.00	238	0.06	0	89	...	87	9.30	0	1	0	1	0	1	0	0
18	4189	91	44.00	977.00	36.20	12.80	280	0.00	0	74	...	59	20.10	1	0	0	0	0	1	0	0
19	1412	72	38.30	980.00	32.00	11.40	286	0.00	0	78	...	66	15.00	1	0	0	0	1	1	0	0
20	4189	91	37.60	986.00	30.40	12.40	287	0.01	0	75	...	57	15.00	0	0	1	1	0	1	0	0
21	1412	72	37.00	980.50	31.80	12.60	286	0.00	0	81	...	66	15.00	0	0	1	1	0	1	0	0
22	4189	91	45.60	990.50	26.60	11.30	297	0.00	0	47	...	67	15.00	1	0	0	1	0	1	0	0
23	1412	72	32.70	988.50	24.40	3.20	270	0.00	0	71	...	73	15.00	1	0	0	1	0	1	0	0
24	4490	74	54.00	982.00	39.10	8.90	242	0.00	0	57	...	27	0.40	1	0	0	0	0	0	1	0
25	4165	83	57.50	979.00	38.00	8.50	280	0.00	0	48	...	22	0.50	1	0	0	1	0	0	1	0
26	3805	75	45.40	976.50	33.20	14.70	258	0.00	0	62	...	38	0.30	1	0	0	1	0	0	1	0
27	4165	83	43.50	978.00	30.20	15.10	298	0.00	0	59	...	96	1.50	1	0	0	0	1	0	1	0
28	4490	74	40.10	986.50	27.80	14.60	289	0.00	0	61	...	41	15.00	0	0	1	1	0	0	1	0
29	4165	83	41.80	987.50	26.90	11.60	313	0.00	0	55	...	35	15.00	1	0	0	1	0	0	1	0
30	4490	74	48.70	990.50	26.70	15.00	296	0.00	0	42	...	32	15.00	1	0	0	1	0	0	1	0
31	4165	83	45.30	990.50	28.80	10.50	336	0.00	0	52	...	29	15.00	0	0	0	1	0	0	1	0

32 rows × 28 columns

In []:

```
# Final test data cleaning
testSubsetData['description_s_Snow'] = 0
testSubsetData
```

Out []:

	Flight Number	Scheduled Elapsed Time	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	...	rh_s	vis_s	description_d_Cloudy	description_d_Rainy	description_d_Snow	description_s_Cloudy	description_s_Rainy	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_ORD	description_s_Snow
0	3839	110	47.30	982.50	42.10	6.20	254	0.00	0	82	...	88	9.40	1	0	0	1	0	0	0	1	0
1	3524	115	58.20	981.00	37.00	10.40	260	0.00	0	45	...	50	24.20	1	0	0	1	0	0	0	1	0
2	538	114	52.50	978.50	37.70	4.60	163	0.00	0	57	...	33	0.40	1	0	0	1	0	0	0	1	0
3	3839	110	43.70	976.00	38.00	13.90	292	0.00	0	80	...	87	9.60	1	0	0	1	0	0	0	1	0
4	3524	115	46.70	976.50	31.00	15.70	298	0.00	0	54	...	48	15.00	1	0	0	1	0	0	0	1	0
5	538	114	38.50	979.50	30.90	8.90	278	0.00	0	74	...	61	15.00	1	0	0	1	0	0	0	1	0
6	3839	110	36.40	985.00	30.50	12.30	280	0.00	0	79	...	73	15.00	0	0	1	1	0	0	0	1	0
7	3524	115	42.40	987.00	25.60	13.40	308	0.00	0	51	...	35	15.00	0	0	1	1	0	0	0	1	0
8	538	114	34.00	989.00	25.60	3.40	319	0.00	0	71	...	32	15.00	1	0	0	1	0	0	0	1	0
9	3839	110	43.30	990.50	28.30	9.70	300	0.00	0	55	...	52	15.00	1	0	0	0	0	0	0	1	0
10	3524	115	47.40	990.50	25.50	15.30	312	0.00	0	42	...	25	15.00	1	0	0	1	0	0	0	1	0
11	538	114	36.00	991.50	25.10	5.60	359	0.00	0	64	...	38	15.00	0	0	0	1	0	0	0	1	0
12	604	201	56.50	981.50	37.60	9.00	252	0.00	0	49	...	42	0.30	1	0	0	0	0	0	0	0	0
13	604	201	47.50	976.00	38.60	13.20	304	0.10	0	71	...	76	12.00	0	1	0	1	0	0	0	0	0
14	604	201	41.30	986.50	26.90	14.70	296	0.00	0	56	...	41	15.00	0	0	1	0	0	0	0	0	0
15	604	201	48.80	990.50	26.80	15.30	301	0.00	0	42	...	45	15.00	1	0	0	1	0	0	0	0	0
16	4189	91	48.50	982.50	41.30	5.40	259	0.00	0	76	...	87	9.90	1	0	0	1	0	1	0	0	0
17	1412	72	60.30	980.50	57.00	12.00	238	0.06	0	89	...	87	9.30	0	1	0	1	0	1	0	0	0
18	4189	91	44.00	977.00	36.20	12.80	280	0.00	0	74	...	59	20.10	1	0	0	0	0	1	0	0	0
19	1412	72	38.30	980.00	32.00	11.40	286	0.00	0	78	...	66	15.00	1	0	0	1	0	1	0	0	0
20	4189	91	37.60	986.00	30.40	12.40	287	0.01	0	75	...	57	15.00	0	0	1	1	0	1	0	0	0
21	1412	72	37.00	980.50	31.80	12.60	286	0.00	0	81	...	66	15.00	0	0	1	1	0	1	0	0	0
22	4189	91	45.60	990.50	26.60	11.30	297	0.00	0	47	...	67	15.00	1	0	0	1	0	1	0	0	0
23	1412	72	32.70	988.50	24.40	3.20	270	0.00	0	71	...	73	15.00	1	0	0	1	0	1	0	0	0
24	4490	74	54.00	982.00	39.10	8.90	242	0.00	0	57	...	27	0.40	1	0	0	0	0	0	1	0	0
25	4165	83	57.50	979.00	38.00	8.50	280	0.00	0	48	...	22	0.50	1	0	0	1	0	0	1	0	0
26	3805	75	45.40	976.50	33.20	14.70	258	0.00	0	62	...	38	0.30	1	0	0	1	0	0	1	0	0
27	4165	83	43.50	978.00	30.20	15.10	298	0.00	0	59	...	96	1.50	1	0	0	0	1	0	1	0	0
28	4490	74	40.10	986.50	27.80	14.60	289	0.00	0	61	...	41	15.00	0	0	1	1	0	0	1	0	0
29	4165	83	41.80	987.50	26.90	11.60	313	0.00	0	55	...	35	15.00	1	0	0	1	0	0	1	0	0
30	4490	74	48.70	990.50	26.70	15.00	296	0.00	0	42	...	32	15.00	1	0	0	1	0	0	1	0	0
31	4165	83	45.30	990.50	28.80	10.50	336	0.00	0	52	...	29	15.00	0	0	0	1	0	0	1	0	0

32 rows × 29 columns

In []:

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
testData = pd.DataFrame(sc.fit_transform(testSubsetData), columns = testSubsetData.columns,
                        index = testSubsetData.index)
testData
```

Out[]:

	Flight Number	Scheduled Elapsed Time	temp_d	pres_d	dewpt_d	wind_spd_d	wind_dir_d	precip_d	snow_d	rh_d	...	vis_s	description_d_Cloudy	description_d_Rainy	description_d_Snow	description_s_Cloudy	description_s_Rainy	Origin Airport_EWR	Origin Airport_IAD	Origin Airport_ORD	description_s_Snow
0	0.65	0.06	0.32	-0.23	1.41	-1.31	-0.89	-0.27	0.00	1.45	...	-0.42	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
1	0.45	0.19	1.87	-0.52	0.67	-0.16	-0.72	-0.27	0.00	-1.30	...	2.00	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
2	-1.46	0.17	1.06	-1.00	0.77	-1.75	-3.60	-0.27	0.00	-0.41	...	-1.89	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
3	0.65	0.06	-0.19	-1.49	0.81	0.80	0.23	-0.27	0.00	1.30	...	-0.39	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
4	0.45	0.19	0.24	-1.39	-0.21	1.29	0.41	-0.27	0.00	-0.63	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
5	-1.46	0.17	-0.93	-0.81	-0.22	-0.57	-0.18	-0.27	0.00	0.86	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
6	0.65	0.06	-1.22	0.25	-0.28	0.36	-0.12	-0.17	0.00	1.23	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
7	0.45	0.19	-0.37	0.64	-1.00	0.66	0.71	-0.17	0.00	-0.85	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
8	-1.46	0.17	-1.56	1.03	-1.00	-2.08	1.04	-0.27	0.00	0.63	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
9	0.65	0.06	-0.24	1.32	-0.60	-0.35	0.47	-0.27	0.00	-0.55	...	0.50	0.67	-0.26	-0.48	-2.08	-0.18	-0.58	-0.58	1.29	0.00
10	0.45	0.19	0.34	1.32	-1.01	1.18	0.83	-0.27	0.00	-1.52	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
11	-1.46	0.17	-1.28	1.51	-1.07	-1.48	2.23	-0.27	0.00	0.11	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	1.29	0.00
12	-1.42	2.40	1.63	-0.42	0.76	-0.54	-0.95	-0.27	0.00	-1.00	...	-1.91	0.67	-0.26	-0.48	-2.08	-0.18	-0.58	-0.58	-0.77	0.00
13	-1.42	2.40	0.35	-1.49	0.90	0.61	0.59	4.67	0.00	0.63	...	0.01	-1.48	3.87	-0.48	0.48	-0.18	-0.58	-0.58	-0.77	0.00
14	-1.42	2.40	-0.53	0.54	-0.81	1.02	0.35	-0.17	0.00	-0.48	...	0.50	-1.48	-0.26	-0.48	-2.08	-0.18	-0.58	-0.58	-0.77	0.00
15	-1.42	2.40	0.54	1.32	-0.82	1.18	0.50	-0.27	0.00	-1.52	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	-0.58	-0.77	0.00
16	0.87	-0.43	0.49	-0.23	1.30	-1.53	-0.75	-0.27	0.00	1.00	...	-0.34	0.67	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
17	-0.90	-0.91	2.17	-0.62	3.59	0.28	-1.37	2.86	0.00	1.97	...	-0.44	-1.48	3.87	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
18	0.87	-0.43	-0.15	-1.29	0.55	0.50	-0.12	-0.27	0.00	0.86	...	1.33	0.67	-0.26	-0.48	-2.08	-0.18	1.73	-0.58	-0.77	0.00
19	-0.90	-0.91	-0.95	-0.71	-0.06	0.11	0.06	-0.27	0.00	1.15	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
20	0.87	-0.43	-1.05	0.45	-0.30	0.39	0.09	-0.02	0.00	0.93	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
21	-0.90	-0.91	-1.14	-0.62	-0.09	0.44	0.06	-0.27	0.00	1.38	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
22	0.87	-0.43	0.08	1.32	-0.85	0.09	0.38	-0.27	0.00	-1.15	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
23	-0.90	-0.91	-1.75	0.93	-1.17	-2.13	-0.42	-0.27	0.00	0.63	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	1.73	-0.58	-0.77	0.00
24	1.06	-0.86	1.27	-0.33	0.98	-0.57	-1.25	-0.27	0.00	-0.41	...	-1.89	0.67	-0.26	-0.48	-2.08	-0.18	-0.58	1.73	-0.77	0.00
25	0.86	-0.63	1.77	-0.91	0.81	-0.68	-0.12	-0.27	0.00	-1.07	...	-1.88	0.67	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00
26	0.63	-0.84	0.05	-1.39	0.11	1.02	-0.78	-0.27	0.00	-0.03	...	-1.91	0.67	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00
27	0.86	-0.63	-0.22	-1.10	-0.33	1.13	0.41	-0.27	0.00	-0.26	...	-1.71	0.67	-0.26	-0.48	-2.08	5.57	-0.58	1.73	-0.77	0.00
28	1.06	-0.86	-0.70	0.54	-0.68	0.99	0.15	-0.27	0.00	-0.11	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00
29	0.86	-0.63	-0.46	0.74	-0.81	0.17	0.86	-0.27	0.00	-0.55	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00
30	1.06	-0.86	0.52	1.32	-0.84	1.10	0.35	-0.27	0.00	-1.52	...	0.50	0.67	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00
31	0.86	-0.63	0.04	1.32	-0.53	-0.13	1.54	-0.27	0.00	-0.78	...	0.50	-1.48	-0.26	-0.48	0.48	-0.18	-0.58	1.73	-0.77	0.00

32 rows × 29 columns

In []:

testData.columns

Out[]:

Index(['Flight Number', 'Scheduled Elapsed Time', 'temp_d', 'pres_d', 'dewpt_d', 'wind_spd_d', 'wind_dir_d', 'precip_d', 'snow_d', 'rh_d', 'vis_d', 'temp_s', 'pres_s', 'dewpt_s', 'wind_spd_s', 'wind_dir_s', 'precip_s', 'snow_s', 'rh_s', 'vis_s', 'description_d_Cloudy', 'description_d_Rainy', 'description_d_Snow', 'description_s_Cloudy', 'description_s_Rainy', 'Origin Airport_EWR', 'Origin Airport_IAD', 'Origin Airport_ORD', 'description_s_Snow'], dtype=object)

In []:

test_output_a = pd.DataFrame(modelLR.predict(testData), index = testData.index, columns = ['pred_Arrival Delay'])
When extending to multiple features remove .array.reshape(-1, 1)
test_output_a.head()

c:\Users\nirmi\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The feature names should match those that were passed during fit. Starting version 1.2, an error will be raised.
Feature names unseen at fit time:
- Scheduled Elapsed Time
Feature names seen at fit time, yet now missing:
- Scheduled Elapsed Time (Minutes)
warnings.warn(message, FutureWarning)

Out[]:

	pred_Arrival Delay
0	-2.29
1	-38.23
2	-26.12
3	9.27
4	-12.55

In []:

from sklearn.tree import DecisionTreeRegressor
from sklearn.model_selection import cross_val_score

modelDT = DecisionTreeRegressor(max_depth=5, min_samples_split=6, min_samples_leaf=2, random_state=10)
modelDT.fit(xTrain, yTrain)
modelDT.score(xTrain, yTrain)

testOutputDT = pd.DataFrame(modelDT.predict(xTest), index=xTest.index, columns=['predArrivalDelay'])
testOutputDT = testOutputDT.merge(yTest, left_index=True, right_index=True)
mean_absolute_error = abs(testOutputDT['predArrivalDelay'] - testOutputDT['Arrival Delay (Minutes)']).mean()
print('Decision Tree Regression - Mean absolute error is ')
print(mean_absolute_error)

Out[]:

DecisionTreeRegressor(max_depth=5, min_samples_leaf=2, min_samples_split=6, random_state=10)

Out[]:

0.44291832914796336
Decision Tree Regression - Mean absolute error is
28.41618348194153

In []:

test_output_d = pd.DataFrame(modelDT.predict(testData), index = testData.index, columns = ['pred_Arrival Delay'])
When extending to multiple features remove .array.reshape(-1, 1)
test_output_d.head()

c:\Users\nirmi\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The feature names should match those that were passed during fit. Starting version 1.2, an error will be raised.
Feature names unseen at fit time:
- Scheduled Elapsed Time
Feature names seen at fit time, yet now missing:
- Scheduled Elapsed Time (Minutes)
warnings.warn(message, FutureWarning)

Out[]:

	pred_Arrival Delay
0	5.88
1	2.70
2	5.23
3	5.88
4	2.70

In []:

def flight_status(delay):
 if delay < -10:
 return "Early"
 elif delay >= -10 and delay < 10:
 return "On Time"
 else:
 return "Late"

file:///C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/projectFinal.html

11/12

```
        return "On-time"
    elif delay > 10 and delay <= 30:
        return "Late"
    else:
        return "Severely late"

test_output_d['Flight Status'] = test_output_d['pred_Arrival Delay'].apply(lambda x: flight_status(x))

test_output_d
```

Out[]:

	pred_Arrival Delay	Flight Status
0	5.88	On-time
1	2.70	On-time
2	5.23	On-time
3	5.88	On-time
4	2.70	On-time
5	2.70	On-time
6	2.70	On-time
7	2.70	On-time
8	2.70	On-time
9	2.70	On-time
10	2.70	On-time
11	2.70	On-time
12	29.05	Late
13	12.97	Late
14	2.70	On-time
15	2.70	On-time
16	52.40	Severely late
17	5.88	On-time
18	2.70	On-time
19	2.70	On-time
20	2.70	On-time
21	2.70	On-time
22	2.70	On-time
23	2.70	On-time
24	29.05	Late
25	5.23	On-time
26	29.05	Late
27	5.88	On-time
28	2.70	On-time
29	2.70	On-time
30	2.70	On-time
31	2.70	On-time

```
In [ ]: test_output_d.to_csv("C:/Users/nirmi/Desktop/SU_SEM3/IntroToML/Project/final_predictions_final_final_final.csv", index=False)
```