In [38]:
```
# Generic inputs for most ML tasks
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
# This is new
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.ensemble import RandomForestRegressor

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

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

from IPython.display import display, HTML
```

Read and pre-process data

Fetching departure data

In [39]:
```
# fetch data

main_data = pd.read_csv('flight_data\mco_syr_sw_dep_20_21_22_23.csv')
main_data.head()
```

Out[39]:

	Carrier Code	Date (MM/DD/YYYY)	Flight Number	Tail Number	Destination Airport	Scheduled departure time	Actual departure time	Scheduled elapsed time (Minutes)	Actual elapsed time (Minutes)	Departure delay (Minutes)	Wheels-off time	Taxi-Ou tim (Minutes
0	F9	1/1/2020	104.00	N365FR	SJU	23:05	23:01	170.00	163.00	-4.00	23:16	15.0
1	F9	1/1/2020	105.00	N309FR	ORD	7:05	6:57	189.00	181.00	-8.00	7:10	13.0
2	F9	1/1/2020	106.00	N701FR	SJU	13:49	13:44	170.00	172.00	-5.00	14:02	18.0
3	F9	1/1/2020	183.00	N238FR	OMA	18:25	19:01	208.00	199.00	36.00	19:14	13.0
4	F9	1/1/2020	679.00	N709FR	DEN	6:00	5:56	259.00	280.00	-4.00	6:22	26.0

In [40]:
```
main_data.dtypes
```

Out[40]:
```
Carrier Code                              object
Date (MM/DD/YYYY)                         object
Flight Number                            float64
Tail Number                               object
Destination Airport                       object
Scheduled departure time                  object
Actual departure time                     object
Scheduled elapsed time (Minutes)         float64
Actual elapsed time (Minutes)            float64
Departure delay (Minutes)                float64
Wheels-off time                           object
Taxi-Out time (Minutes)                  float64
Delay Carrier (Minutes)                  float64
Delay Weather (Minutes)                  float64
Delay National Aviation System (Minutes) float64
Delay Security (Minutes)                 float64
Delay Late Aircraft Arrival (Minutes)    float64
dtype: object
```

In [41]:
```
len(main_data)
```

Out[41]:
```
65004
```

In [42]:
```
main_data.rename(columns = {'Destination Airport' : 'Destination_Airport' ,
                            'Carrier Code' : 'Carrier_Code',
                            'Flight Number' : 'Flight_Number',
                            'Tail Number' : 'Tail_Number',
                            'Date (MM/DD/YYYY)':'Date',
                            'Delay Carrier (Minutes)' : 'dep_Delay_Carrier',
                             'Delay Weather (Minutes)' : 'dep_Delay_Weather',
                'Delay National Aviation System (Minutes)' : 'dep_Delay_National_Aviation_System',
                'Delay Security (Minutes)' : 'dep_Delay_Security',
                'Delay Late Aircraft Arrival (Minutes)' : 'dep_Delay_Late_Aircraft_Arrival'},inplace = Tr
```

In [43]:
```
main_data.dtypes
```

Out[43]:
```
Carrier_Code                          object
Date                                  object
Flight_Number                        float64
Tail_Number                           object
Destination_Airport                   object
Scheduled departure time              object
Actual departure time                 object
Scheduled elapsed time (Minutes)     float64
Actual elapsed time (Minutes)        float64
Departure delay (Minutes)            float64
Wheels-off time                       object
Taxi-Out time (Minutes)              float64
dep_Delay_Carrier                    float64
dep_Delay_Weather                    float64
dep_Delay_National_Aviation_System   float64
dep_Delay_Security                   float64
dep_Delay_Late_Aircraft_Arrival      float64
dtype: object
```

In [44]:
```
main_data = main_data[main_data['Destination_Airport'] == 'SYR']
len(main_data)
```

Out[44]:
```
880
```

In [45]:
```
main_data['dep_hour'] = main_data['Scheduled departure time'].str.split(":").str[0].astype('int64')
main_data['Date'] = pd.to_datetime( main_data['Date'],format ="%m/%d/%Y")
main_data['dep_day'] = main_data['Date'].dt.day_of_week.astype('int64')
main_data['dep_year'] = main_data['Date'].dt.year.astype('int64')
main_data.dtypes
```

Out[45]:
```
Carrier_Code                            object
Date                              datetime64[ns]
Flight_Number                            float64
Tail_Number                               object
Destination_Airport                       object
Scheduled departure time                  object
Actual departure time                     object
Scheduled elapsed time (Minutes)         float64
Actual elapsed time (Minutes)            float64
Departure delay (Minutes)                float64
Wheels-off time                           object
Taxi-Out time (Minutes)                  float64
dep_Delay_Carrier                        float64
dep_Delay_Weather                        float64
dep_Delay_National_Aviation_System       float64
dep_Delay_Security                       float64
dep_Delay_Late_Aircraft_Arrival          float64
dep_hour                                   int64
dep_day                                    int64
dep_year                                   int64
dtype: object
```

In [47]:
```
main_data['dep_hour'].value_counts()
```

Out[47]:
```
dep_hour
8     153
17    148
7     123
9      80
16     67
15     62
12     56
11     54
14     53
6      37
13     19
10     19
18      6
5       3
Name: count, dtype: int64
```

In [48]:
```
main_data['dep_year'].value_counts()
```

Out[48]:
```
dep_year
2023    281
2022    242
2021    213
2020    144
Name: count, dtype: int64
```

In [49]:
```
main_data['dep_order'] = 'latter'
```

In [50]:
```
main_data.head()
```

Out[50]:

	Carrier_Code	Date	Flight_Number	Tail_Number	Destination_Airport	Scheduled departure time	Actual departure time	Scheduled elapsed time (Minutes)	Actual elapsed time (Minutes)	Departure delay (Minutes)
35	F9	2020-01-01	1,214.00	N311FR	SYR	11:55	11:47	164.00	156.00	-8.00
64	F9	2021-01-01	1,214.00	N220FR	SYR	14:14	14:36	170.00	148.00	22.00
102	F9	2022-01-01	1,214.00	N233FR	SYR	17:01	20:00	166.00	158.00	179.00
243	F9	2021-01-02	1,214.00	N334FR	SYR	14:14	14:14	170.00	141.00	0.00
348	F9	2023-01-02	1,216.00	N360FR	SYR	7:55	9:52	169.00	168.00	117.00

5 rows × 21 columns

Fetching arrival data

In [51]:
```
# fetch data

arr_data = pd.read_csv('flight_data\mco_syr_sw_arr_20_21_22_23.csv')
arr_data.head()
```

Out[51]:

	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)	D Ca (Min
0	F9	1/1/2020	958.00	N227FR	RSW	12:44	12:15	179.00	162.00	-29.00	12:11	4.00	
1	F9	1/1/2020	1,214.00	N311FR	MCO	14:39	14:23	164.00	156.00	-16.00	14:17	6.00	
2	F9	1/1/2021	1,214.00	N220FR	MCO	17:04	17:04	170.00	148.00	0.00	16:59	5.00	
3	F9	1/1/2021	2,542.00	N232FR	TPA	12:03	11:39	163.00	157.00	-24.00	11:33	6.00	
4	F9	1/1/2022	958.00	NaN	RSW	9:59	0:00	179.00	0.00	0.00	0:00	0.00	

In [52]:
```
arr_data.rename(columns = {'Origin Airport' : 'Origin_Airport' ,
                           'Carrier Code' : 'Carrier_Code',
                           'Flight Number' : 'Flight_Number',
                           'Tail Number' : 'Tail_Number',
                           'Date (MM/DD/YYYY)':'Date',
                           'Delay Carrier (Minutes)' : 'arr_Delay_Carrier',
                            'Delay Weather (Minutes)' : 'arr_Delay_Weather',
                'Delay National Aviation System (Minutes)' : 'arr_Delay_National_Aviation_System',
                'Delay Security (Minutes)' : 'arr_Delay_Security',
                'Delay Late Aircraft Arrival (Minutes)' : 'arr_Delay_Late_Aircraft_Arrival'},inplace = Tr
```

In [53]:
```
arr_data.dtypes
```

Out[53]:
```
Carrier_Code                          object
Date                                  object
Flight_Number                        float64
Tail_Number                           object
Origin_Airport                        object
Scheduled Arrival Time                object
Actual Arrival Time                   object
Scheduled Elapsed Time (Minutes)     float64
Actual Elapsed Time (Minutes)        float64
Arrival Delay (Minutes)              float64
Wheels-on Time                        object
Taxi-In time (Minutes)               float64
arr_Delay_Carrier                    float64
arr_Delay_Weather                    float64
arr_Delay_National_Aviation_System   float64
arr_Delay_Security                   float64
arr_Delay_Late_Aircraft_Arrival      float64
dtype: object
```

In [54]:
```
len(arr_data)
```

Out[54]:
```
1649
```

In [55]:
```
arr_data = arr_data[arr_data['Origin_Airport'] == 'MCO']
len(arr_data)
```

Out[55]:
```
880
```

In [56]:
```
arr_data['arr_hour'] = arr_data['Scheduled Arrival Time'].str.split(":").str[0].astype('int64')

arr_data['Date'] = pd.to_datetime( arr_data['Date'],format ="%m/%d/%Y")
arr_data['arr_day'] = arr_data['Date'].dt.day_of_week.astype('int64')
arr_data['arr_year'] = arr_data['Date'].dt.year.astype('int64')
arr_data.drop(columns = ['Scheduled Elapsed Time (Minutes)', 'Actual Elapsed Time (Minutes)'],inplace = Tr

arr_data.dtypes
```

Out[56]:
```
Carrier_Code                          object
Date                          datetime64[ns]
Flight_Number                        float64
Tail_Number                           object
Origin_Airport                        object
Scheduled Arrival Time                object
Actual Arrival Time                   object
Arrival Delay (Minutes)              float64
Wheels-on Time                        object
Taxi-In time (Minutes)               float64
arr_Delay_Carrier                    float64
arr_Delay_Weather                    float64
arr_Delay_National_Aviation_System   float64
arr_Delay_Security                   float64
arr_Delay_Late_Aircraft_Arrival      float64
arr_hour                               int64
arr_day                                int64
arr_year                               int64
dtype: object
```

In [57]:
```
arr_data['arr_hour'].value_counts()
```

Out[57]:
```
arr_hour
19    178
11    152
10     98
9      73
17     62
14     58
12     57
18     56
15     44
20     35
8      24
16     23
13     19
21      1
Name: count, dtype: int64
```

In [58]:
```
arr_data['arr_year'].value_counts()
```

Out[58]:
```
arr_year
2023    281
2022    242
2021    213
2020    144
Name: count, dtype: int64
```

Merging departure and arrival data

In [59]:
```
merged_df = pd.merge(main_data, arr_data, on=['Carrier_Code','Date', 'Flight_Number', 'Tail_Number'], how=
merged_df.head()
```

Out[59]:

	Carrier_Code	Date	Flight_Number	Tail_Number	Destination_Airport	Scheduled departure time	Actual departure time	Scheduled elapsed time (Minutes)	Actual elapsed time (Minutes)	Departure delay (Minutes)	...
0	F9	2020-01-01	1,214.00	N311FR	SYR	11:55	11:47	164.00	156.00	-8.00	...
1	F9	2021-01-01	1,214.00	N220FR	SYR	14:14	14:36	170.00	148.00	22.00	...
2	F9	2022-01-01	1,214.00	N233FR	SYR	17:01	20:00	166.00	158.00	179.00	...
3	F9	2021-01-02	1,214.00	N334FR	SYR	14:14	14:14	170.00	141.00	0.00	...
4	F9	2023-01-02	1,216.00	N360FR	SYR	7:55	9:52	169.00	168.00	117.00	...

5 rows × 35 columns

In [60]:
```
merged_df.dtypes
```

Out[60]:
```
Carrier_Code                          object
Date                          datetime64[ns]
Flight_Number                        float64
Tail_Number                           object
Destination_Airport                   object
Scheduled departure time              object
Actual departure time                 object
Scheduled elapsed time (Minutes)     float64
Actual elapsed time (Minutes)        float64
Departure delay (Minutes)            float64
Wheels-off time                       object
Taxi-Out time (Minutes)              float64
dep_Delay_Carrier                    float64
dep_Delay_Weather                    float64
dep_Delay_National_Aviation_System   float64
dep_Delay_Security                   float64
dep_Delay_Late_Aircraft_Arrival      float64
dep_hour                               int64
dep_day                                int64
dep_year                               int64
dep_order                             object
Origin_Airport                        object
Scheduled Arrival Time                object
Actual Arrival Time                   object
Arrival Delay (Minutes)              float64
Wheels-on Time                        object
Taxi-In time (Minutes)               float64
arr_Delay_Carrier                    float64
arr_Delay_Weather                    float64
arr_Delay_National_Aviation_System   float64
arr_Delay_Security                   float64
arr_Delay_Late_Aircraft_Arrival      float64
arr_hour                               int64
arr_day                                int64
arr_year                               int64
dtype: object
```

In [61]:
```
len(merged_df)
```

Out[61]: 880

In [62]:
```
merged_df.to_csv('flight_data\mco_syr_sw_20_21_22_23.csv')
```

In [63]:
```
new_df = pd.read_csv('flight_data\mco_syr_sw_20_21_22_23.csv')
len(new_df)
new_df.head()
```

Out[63]: 880

Out[63]:

	Unnamed: 0	Carrier_Code	Date	Flight_Number	Tail_Number	Destination_Airport	Scheduled departure time	Actual departure time	Scheduled elapsed time (Minutes)	Actual elapsed time (Minutes)	..
0	0	F9	2020-01-01	1,214.00	N311FR	SYR	11:55	11:47	164.00	156.00	..
1	1	F9	2021-01-01	1,214.00	N220FR	SYR	14:14	14:36	170.00	148.00	..
2	2	F9	2022-01-01	1,214.00	N233FR	SYR	17:01	20:00	166.00	158.00	..
3	3	F9	2021-01-02	1,214.00	N334FR	SYR	14:14	14:14	170.00	141.00	..
4	4	F9	2023-01-02	1,216.00	N360FR	SYR	7:55	9:52	169.00	168.00	..

5 rows × 36 columns

combining multiple years of data

In [64]:

```
# # Read the CSV files
# df_20_21 = pd.read_csv('flight_data\ord_syr_ua_20_21.csv')
# df_22_23 = pd.read_csv('flight_data\ord_syr_ua_22_23.csv')

# # Concatenate the dataframes
# combined_df = pd.concat([df_20_21, df_22_23])

# # Reset the index
# combined_df.reset_index(drop=True, inplace=True)

# # Save the combined dataframe to a new CSV file
# combined_df.to_csv('flight_data\ord_syr_ua_20_21_22_23.csv', index=False)
```