SpaceX Falcon 9 First Stage Landing Prediction

Assignment: Exploring and Preparing Data

Estimated time needed: **70** minutes

In this assignment, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is due to the fact that SpaceX can reuse the first stage.

In this lab, you will perform Exploratory Data Analysis and Feature Engineering.

Falcon 9 first stage will land successfully

Several examples of an unsuccessful landing are shown here:

Most unsuccessful landings are planned. Space X performs a controlled landing in the oceans.

Objectives

Perform exploratory Data Analysis and Feature Engineering using `Pandas` and `Matplotlib`

- Exploratory Data Analysis
- Preparing Data Feature Engineering

Import Libraries and Define Auxiliary Functions

We will import the following libraries the lab

In [1]:
```
# andas is a software library written for the Python programming language for d
import pandas as pd
#NumPy is a library for the Python programming language, adding support for lar
import numpy as np
# Matplotlib is a plotting library for python and pyplot gives us a MatLab like
import matplotlib.pyplot as plt
#Seaborn is a Python data visualization library based on matplotlib. It provide
import seaborn as sns
```

Exploratory Data Analysis

First, let's read the SpaceX dataset into a Pandas dataframe and print its summary

In [2]:
```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cl
df.tail(5)
```

Out[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
85	86	2020-09-03	Falcon 9	15400.0	VLEO	KSC LC 39A	True ASDS	2
86	87	2020-10-06	Falcon 9	15400.0	VLEO	KSC LC 39A	True ASDS	3
87	88	2020-10-18	Falcon 9	15400.0	VLEO	KSC LC 39A	True ASDS	6
88	89	2020-10-24	Falcon 9	15400.0	VLEO	CCAFS SLC 40	True ASDS	3
89	90	2020-11-05	Falcon 9	3681.0	MEO	CCAFS SLC 40	True ASDS	1

First, let's try to see how the `FlightNumber` (indicating the continuous launch attempts.) and `Payload` variables would affect the launch outcome.

We can plot out the `FlightNumber` vs. `PayloadMass` and overlay the outcome of the launch. We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

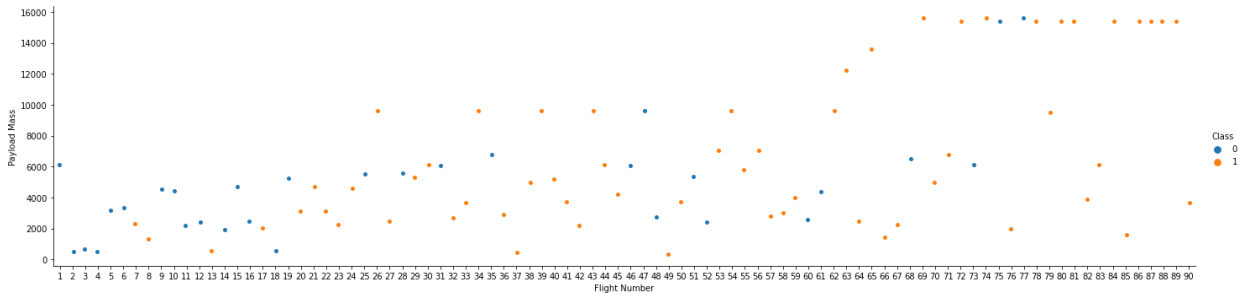
In [3]:
```
df[['FlightNumber' , 'PayloadMass','Outcome']]
```

Out[3]:

	FlightNumber	PayloadMass	Outcome
0	1	6104.959412	None None
1	2	525.000000	None None
2	3	677.000000	None None
3	4	500.000000	False Ocean
4	5	3170.000000	None None
...	...	...	...
85	86	15400.000000	True ASDS
86	87	15400.000000	True ASDS
87	88	15400.000000	True ASDS
88	89	15400.000000	True ASDS
89	90	3681.000000	True ASDS

90 rows × 3 columns

In [6]:
```
sns.catplot(x='FlightNumber', y='PayloadMass', hue='Class', aspect = 4,data=df)
plt.xlabel('Flight Number')
plt.ylabel('Payload Mass')
plt.show()
```



In [ ]:
```
```

We see that different launch sites have different success rates. `CCAFS LC-40`, has a success rate of 60 %, while `KSC LC-39A` and `VAFB SLC 4E` has a success rate of 77%.

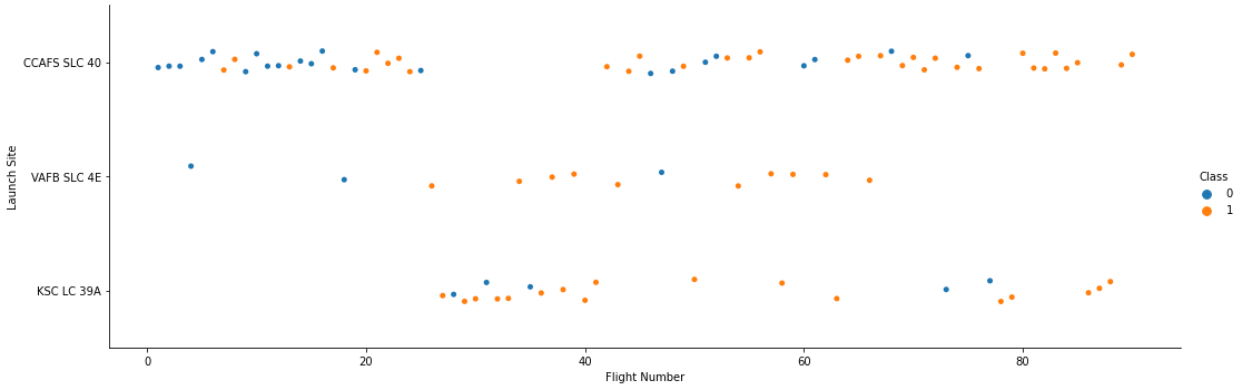
Next, let's drill down to each site visualize its detailed launch records.

TASK 1: Visualize the relationship between Flight Number and Launch Site

Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

In [8]:
```
sns.catplot(x='FlightNumber', y='LaunchSite', hue='Class', data=df, aspect=3)
```

```
plt.xlabel('Flight Number')
plt.ylabel('Launch Site')
plt.show()
```



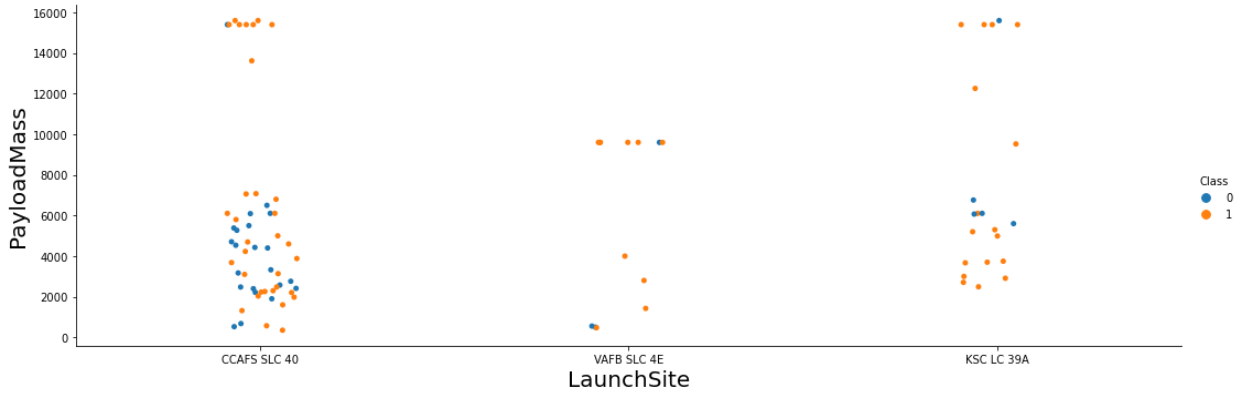
Flight Number X , Launch Site Y We can see from the data they SpaceX primarily flies Falon9 out of CCAFS. According to the scatterplot for each SITE the results are as follows: CCAFS:22failures/33successes = 67% = Success rate VAFB: 3 failures/10 successes = 70% Success Rate KSC: 5 failures /17 successes = 66% Success Rate We can see that throughout they have a steady success rate hovering around an average of 68% percent from 2010-2020. Meaning there is no strong coorelation that a launch site has impact on success rate. the ratio remains steady. This data does show that more flights have left CCAFS more than the other two centers as we knew before. We can see that historically, as the flight number increases which is also coorelated to the years the number of failures start to reduce. This means that the success rate could be trending higher with recent innovations and trial and error testing.

Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

In [81]:
```
sns.catplot(x='LaunchSite' , y='PayloadMass', data=df, hue='Class', aspect= 3)
plt.xlabel('LaunchSite', fontsize =20)
plt.ylabel('PayloadMass', fontsize=20)
plt.show()
```



In [82]:
```
df['LaunchSite'].value_counts()
```

Out[82]:
```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Launch Site X PayloadMass Y The above data shows the coorelation between the three LaunchSites SpaceX uses along with the respective Payload Mass. This data extends until 2020 so though outdated we can view what happened at this particular time. As listed in our previous recordings, we have launch site frequency of launches per site along with the ratio of successful landings and failed attempts indicated blue(failed) & orange(success) From this data we can visually see VAFB the payloads didnt exceed 100000. Does this mean there was a restriction in place at that time that was agreed upon between the two organizations? Holistically speaking, the SpaceX Falcon 9 is flying majority of it's payloads at 100000 KG and under. Though knowing Elon and his willingness to push boundaries, continous innovation will leave testing to exist above the comfortability zone. For instance on 8-12-22 The VAFB site (which for 10 years have flown payloads into Orbit at 10000 and under), have launched 46 starlink satelittes which accumulate to a total KG of 11,960 kg.

Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

TASK 3: Visualize the relationship between success rate of each orbit type

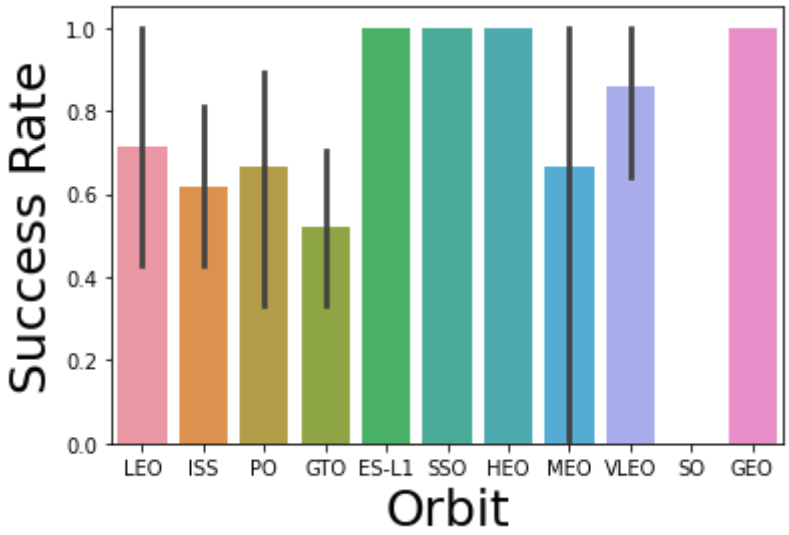
Next, we want to visually check if there are any relationship between success rate and orbit type.

Let's create a `bar chart` for the sucess rate of each orbit

In [83]:
```
df['Orbit'].value_counts()
```

Out[83]:
```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

In [84]:
```
sns.barplot(y="Class", x="Orbit", data=df)
plt.xlabel("Orbit", fontsize=25)
plt.ylabel("Success Rate", fontsize=25)
plt.show()
```



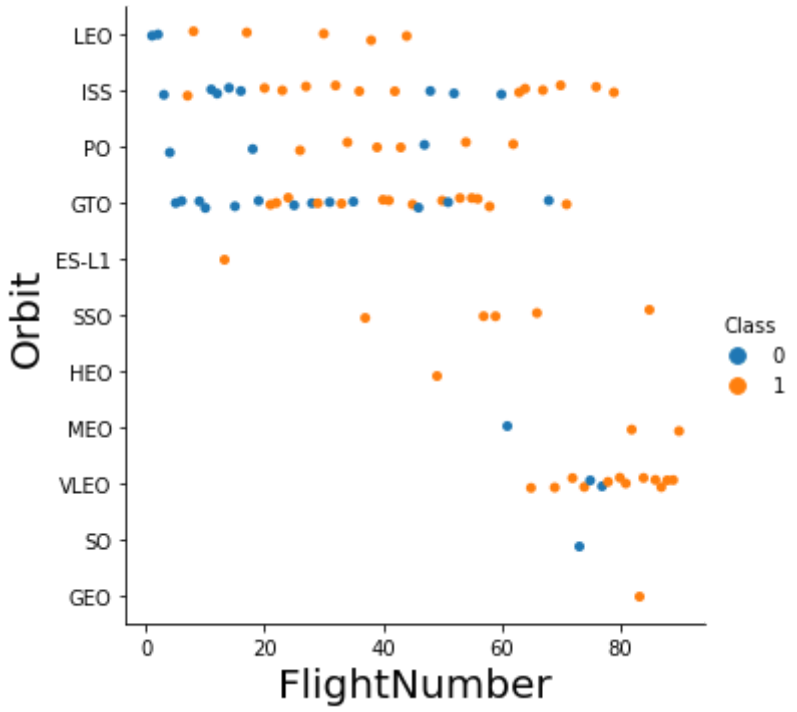
Looking at the Barplot and the above data we can see that SpaceX has flown into GTO (51% success rate) orbit more than any of the others. With that being said. The success rate is lower in comparison to the other data points due to a larger sample size. There is not enough evidence to conclude that the landings indicating 100% are truly highly successful based on a low sample size. We need more data here to give us a more accurate prediction of the data as they continue to fly in these orbital systems. Further questioning arises as to why the Space X team choose to fly into GTO more than the others. We can look to the internet for a further discovery. ISS orbital system is listed second for travel frequency. Third, VLEO.

Analyze the ploted bar chart try to find which orbits have high sucess rate.

TASK 4: Visualize the relationship between FlightNumber and Orbit type

For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

In [17]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be th
sns.catplot(x = 'FlightNumber', y = 'Orbit', hue='Class',  data=df )
plt.xlabel('FlightNumber', fontsize=20)
plt.ylabel('Orbit',fontsize=20)
plt.show()
```



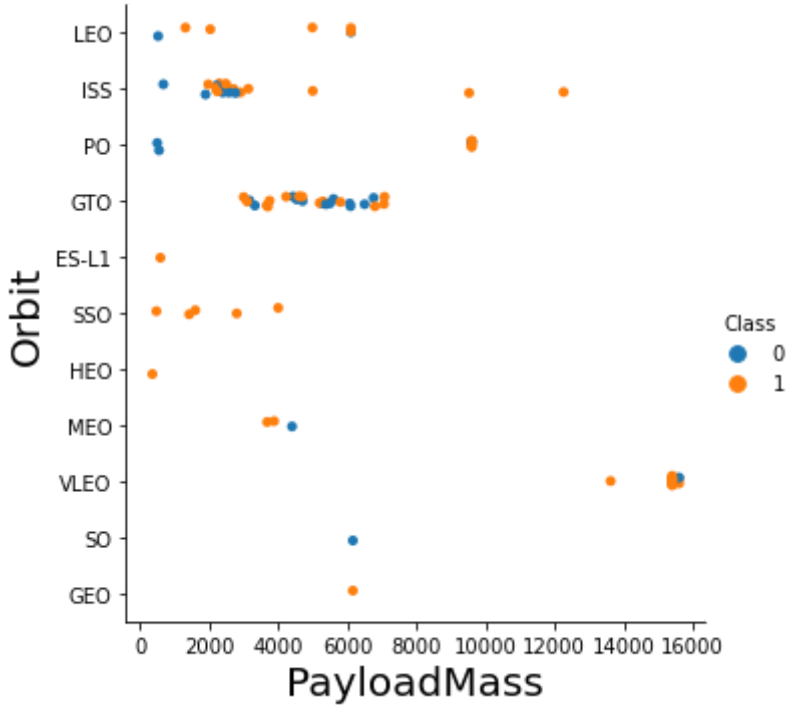
We can look at as the flight number increase the SpaceX team has experimented more within VLEO(Very Low Earth Orbit). We can see that majority of their success has come from flying within the VLEO and LEO system. This chart is to really determine where the success and failures are coming from in relation to the orbit.

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

TASK 5: Visualize the relationship between Payload and Orbit type

Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

In [18]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orb
sns.catplot(x = 'PayloadMass',y='Orbit', hue='Class', data=df)
plt.xlabel('PayloadMass',fontsize=20)
plt.ylabel('Orbit',fontsize=20)
plt.show()
```



We can look at the size of the payload and determine the where the success and failure is mostly in relation to what orbital system the rocket is flying in. LEO has payloads under 8000kg and is likely to continue its success with more data accumulated. If we look at what has happened above 10000kg payload more flights have been successful in their launch and landings. Above 8,000 similar results. It's safe to say that Space X Falcon 9 is well- equipped to travel through multiple orbits with higher payloads.

With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

TASK 6: Visualize the launch success yearly trend

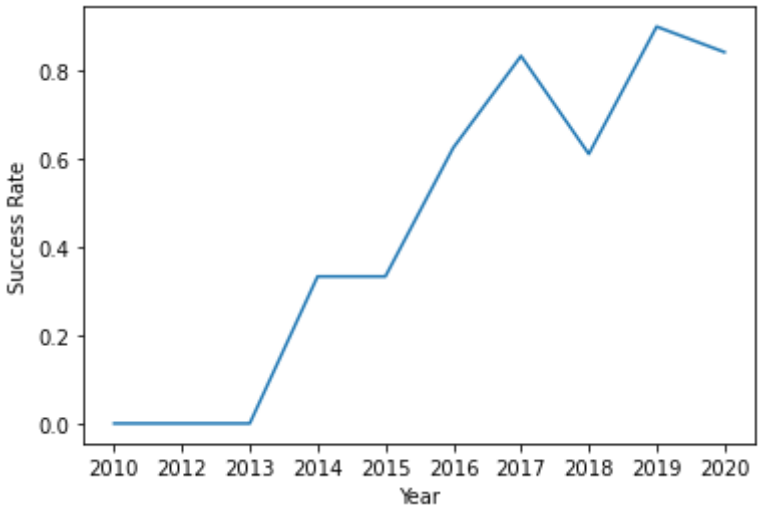
You can plot a line chart with x axis to be `Year` and y axis to be average success rate, to get the average launch success trend.

The function will help you get the year from the date:

In [39]:
```
# A function to Extract years from the date
year=[]
def Extract_year(date):
    for i in df["Date"]:
        year.append(i.split("-")[0])
    return year
Extract_year(1)
df["Year"]=year
average_by_year = df.groupby(by="Year").mean()
average_by_year.reset_index(inplace=True)
```

In [40]:
```
plt.plot(average_by_year["Year"],average_by_year["Class"])
plt.xlabel("Year")
```

```
plt.ylabel("Success Rate")
plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

Features Engineering

By now, you should have obtain some preliminary insights about how each important variable would affect the success rate, we will select the features that will be used in success prediction in the future module.

In [43]:
```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights',
features
```

Out[43]:

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	
0	1	6104.959412	LEO	CCAFS SLC 40	1	False	False	False	
1	2	525.000000	LEO	CCAFS SLC 40	1	False	False	False	
2	3	677.000000	ISS	CCAFS SLC 40	1	False	False	False	
3	4	500.000000	PO	VAFB SLC 4E	1	False	False	False	
4	5	3170.000000	GTO	CCAFS SLC 40	1	False	False	False	
...	...	...	...	...	...	...	...	...	
85	86	15400.000000	VLEO	KSC LC 39A	2	True	True	True	5e9e303
86	87	15400.000000	VLEO	KSC LC 39A	3	True	True	True	5e9e303
87	88	15400.000000	VLEO	KSC LC 39A	6	True	True	True	5e9e303
88	89	15400.000000	VLEO	CCAFS SLC 40	3	True	True	True	5e9e303
89	90	3681.000000	MEO	CCAFS SLC 40	1	True	False	True	5e9e303

90 rows × 12 columns

TASK 7: Create dummy variables to categorical columns

Use the function `get_dummies` and `features` dataframe to apply OneHotEncoder to the column `Orbits`, `LaunchSite`, `LandingPad`, and `Serial`. Assign the value to the variable `features_one_hot`, display the results using the method head. Your result dataframe must include all features including the encoded ones.

In [44]:
```
features_one_hot = pd.get_dummies(features , columns=['Serial', 'LandingPad', '
features_one_hot
```

Out[44]:

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Serial
0	1	6104.959412	1	False	False	False	1.0	0	
1	2	525.000000	1	False	False	False	1.0	0	
2	3	677.000000	1	False	False	False	1.0	0	
3	4	500.000000	1	False	False	False	1.0	0	
4	5	3170.000000	1	False	False	False	1.0	0	
...	...	...	...	...	...	...	...	...	
85	86	15400.000000	2	True	True	True	5.0	2	
86	87	15400.000000	3	True	True	True	5.0	2	
87	88	15400.000000	6	True	True	True	5.0	5	
88	89	15400.000000	3	True	True	True	5.0	2	
89	90	3681.000000	1	True	False	True	5.0	0	

90 rows × 80 columns

TASK 8: Cast all numeric columns to `float64`

Now that our `features_one_hot` dataframe only contains numbers cast the entire dataframe to variable type `float64`

In [45]:
```
features_one_hot = features_one_hot.astype(float)
features_one_hot
```

Out[45]:

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Serial
0	1.0	6104.959412	1.0	0.0	0.0	0.0	1.0	0.0	
1	2.0	525.000000	1.0	0.0	0.0	0.0	1.0	0.0	
2	3.0	677.000000	1.0	0.0	0.0	0.0	1.0	0.0	
3	4.0	500.000000	1.0	0.0	0.0	0.0	1.0	0.0	
4	5.0	3170.000000	1.0	0.0	0.0	0.0	1.0	0.0	
...	...	...	...	...	...	...	...	...	
85	86.0	15400.000000	2.0	1.0	1.0	1.0	5.0	2.0	
86	87.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	
87	88.0	15400.000000	6.0	1.0	1.0	1.0	5.0	5.0	
88	89.0	15400.000000	3.0	1.0	1.0	1.0	5.0	2.0	
89	90.0	3681.000000	1.0	1.0	0.0	1.0	5.0	0.0	

90 rows × 80 columns

We can now export it to a **CSV** for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

```
features_one_hot.to_csv('dataset_part\_3.csv', index=False)
```

Authors

Joseph Santarcangelo has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Nayef Abou Tayoun is a Data Scientist at IBM and pursuing a Master of Management in Artificial intelligence degree at Queen's University.

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2021-10-12	1.1	Lakshmi Holla	Modified markdown
2020-09-20	1.0	Joseph	Modified Multiple Areas
2020-11-10	1.1	Nayef	updating the input data