(https://skills.network/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=1000SkillsNetwork-Channel-SkillsNetworkCoursesIBMDS0321ENSkillsNetwork26802033-2022-01-01)

Sricharan Nallan Chakravarthula

Space X 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 (EDA)
- Prepare Data Feature Engineering (Pandas and Matplotlib)

Import Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
print("All libraries have been imported.")
```

```
All libraries have been imported.
```

Start Here

Exploratory Data Analysis

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

In [2]:

```python
# Load the dataset from the last section (dataset_part_2.csv)
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdo
df.head(4)
```

Out[2]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1

In [3]:

```python
df.shape
```

Out[3]: (90, 18)

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

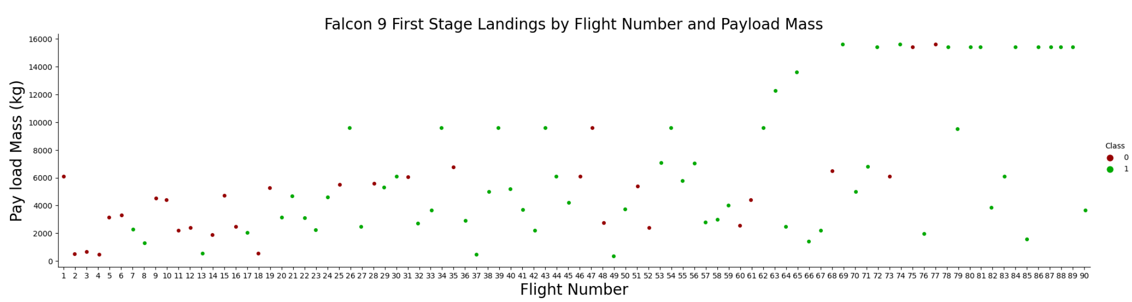
We can plot out the `FlightNumber` vs. `PayloadMass` and overlay the outcome of the launch. We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will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

In [4]:

```python
# Set colors for successful and failed launches.
# df['Class'] = 0  (FAILURE -  RED  - Falcon 9 first stage did not land su
# df['Class'] = 1  (SUCCESS - GREEN - Falcon 9 first stage landed Successf
colors_dict = {0:'#990000',1:'#00aa00'}
```

In [5]:

```python
# Plot FlightNumber vs PayloadMass
sns.catplot(data=df, x="FlightNumber", y="PayloadMass", hue="Class", aspec
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Pay load Mass (kg)", fontsize=20)
plt.title("Falcon 9 First Stage Landings by Flight Number and Payload Mass
plt.show()
```



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.

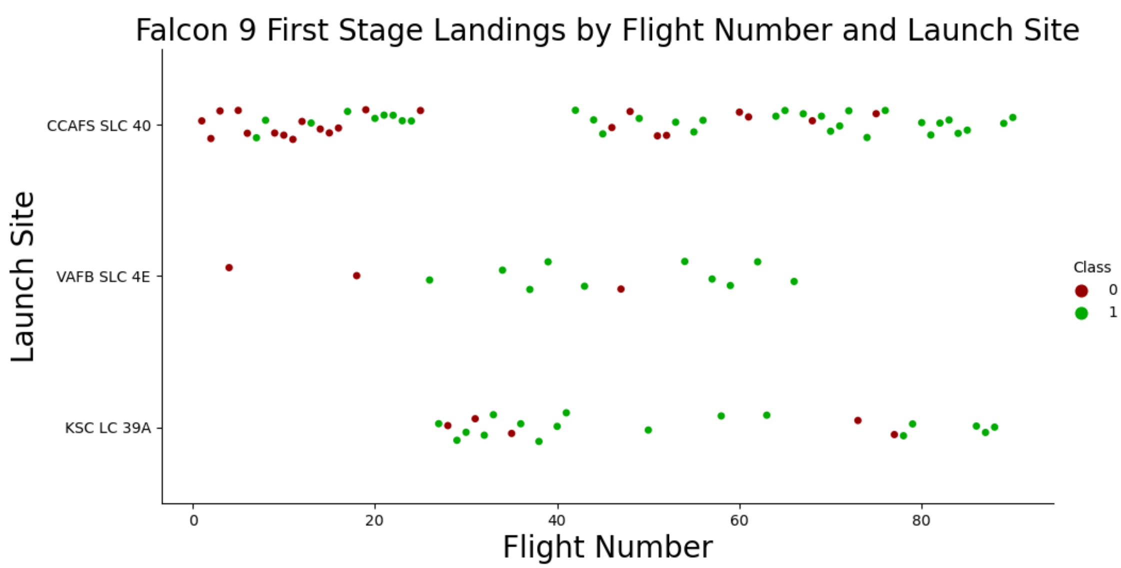
Task 1

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]:

```
# Task 1 ANSWER
# Plot a scatter point chart with x axis to be Flight Number and y axis to
sns.catplot(data=df, x="FlightNumber", y="LaunchSite", hue="Class", aspect
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.title("Falcon 9 First Stage Landings by Flight Number and Launch Site"
plt.show()
```



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

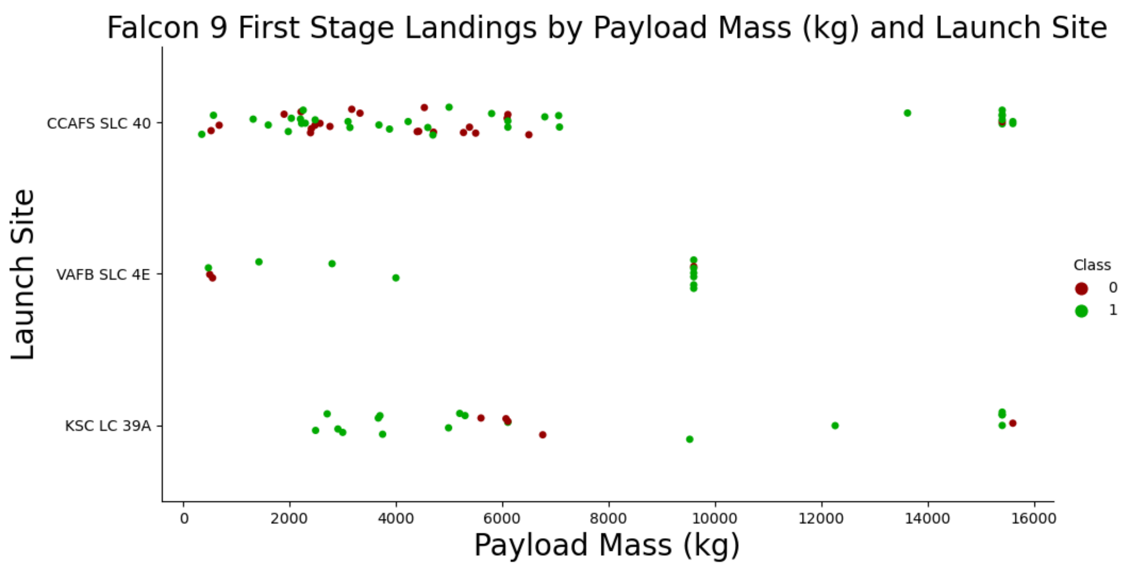
Task
2

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 [9]:

```
# Task 2 ANSWER
# Plot a scatter point chart with x axis to be Payload Mass (kg) and y axi:
sns.catplot(data=df, x="PayloadMass", y="LaunchSite", hue="Class", aspect=
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Launch Site", fontsize=20)
plt.title("Falcon 9 First Stage Landings by Payload Mass (kg) and Launch S:
plt.show()
```



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

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.

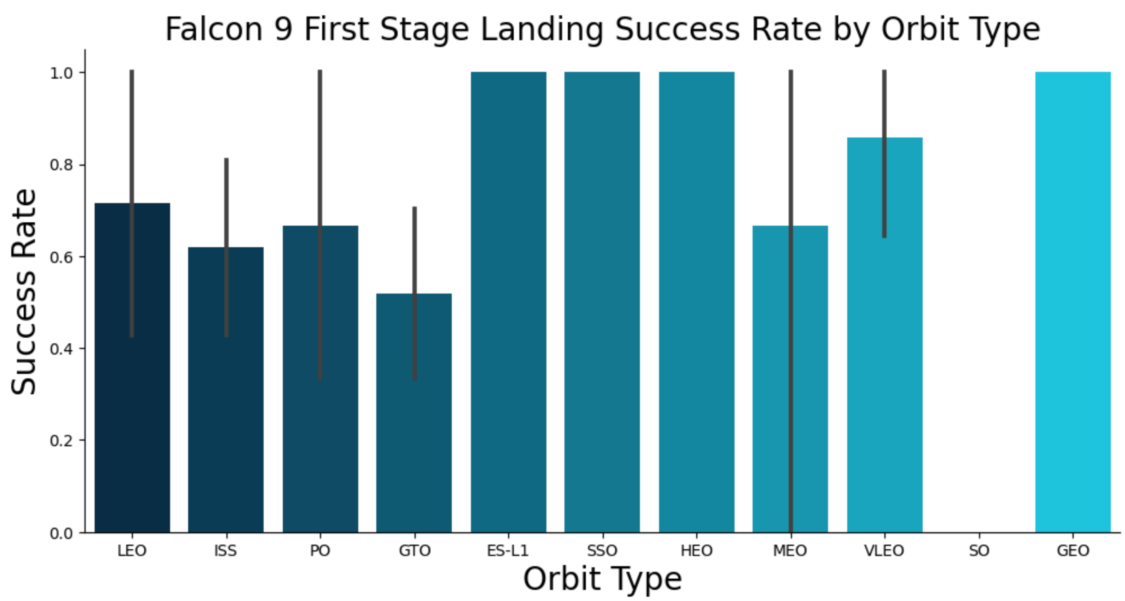
Let's create a `bar chart` for the sucess rate of each orbit

In [10]:
```
# Orbit Types
list(df['Orbit'].unique())
```

Out[10]:
```
['LEO', 'ISS', 'PO', 'GTO', 'ES-L1', 'SSO', 'HEO', 'MEO', 'VLEO', 'SO',
'GEO']
```

In [11]:
```
# Set colors for orbit types.
orbit_colors_dict = {'LEO':'#003355', 'ISS':'#004466', 'PO':'#005577', 'GT
```

In [12]:
```
# Task 3 ANSWER
# Plot a bar chart with x axis to be Orbit and y axis to be the launch site
sns.catplot(kind="bar", data=df, x="Orbit", y="Class", aspect=2, palette=o
plt.xlabel("Orbit Type", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.title("Falcon 9 First Stage Landing Success Rate by Orbit Type", fonts
plt.show()
```



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

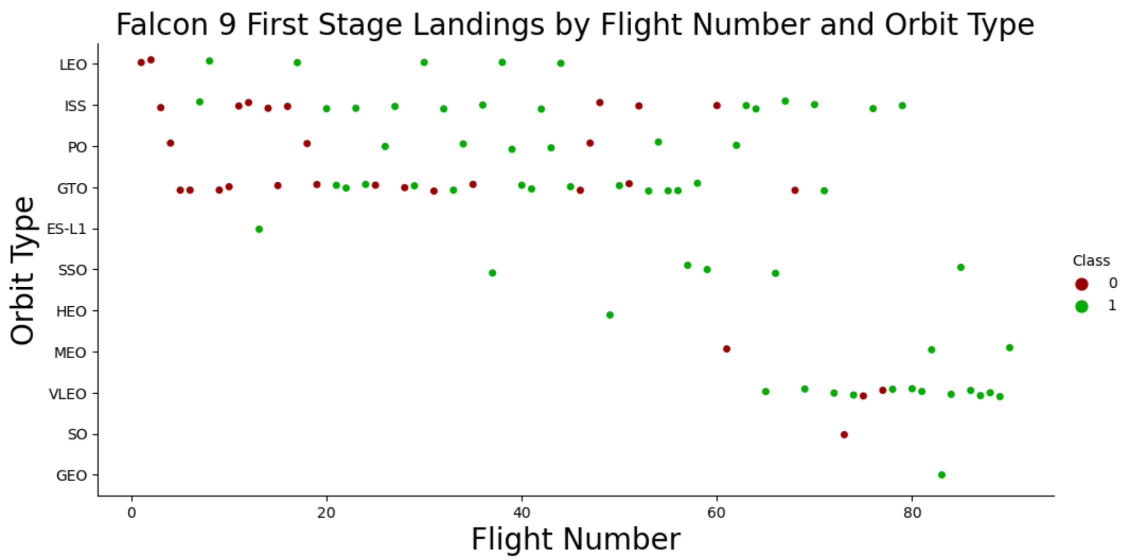
Task
4

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 [13]:

```
# Task 4 ANSWER
# Plot a scatter point chart with x axis to be FlightNumber and y axis to l
sns.catplot(data=df, x="FlightNumber", y="Orbit", hue="Class", aspect=2, p
plt.xlabel("Flight Number", fontsize=20)
plt.ylabel("Orbit Type", fontsize=20)
plt.title("Falcon 9 First Stage Landings by Flight Number and Orbit Type",
plt.show()
```



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.

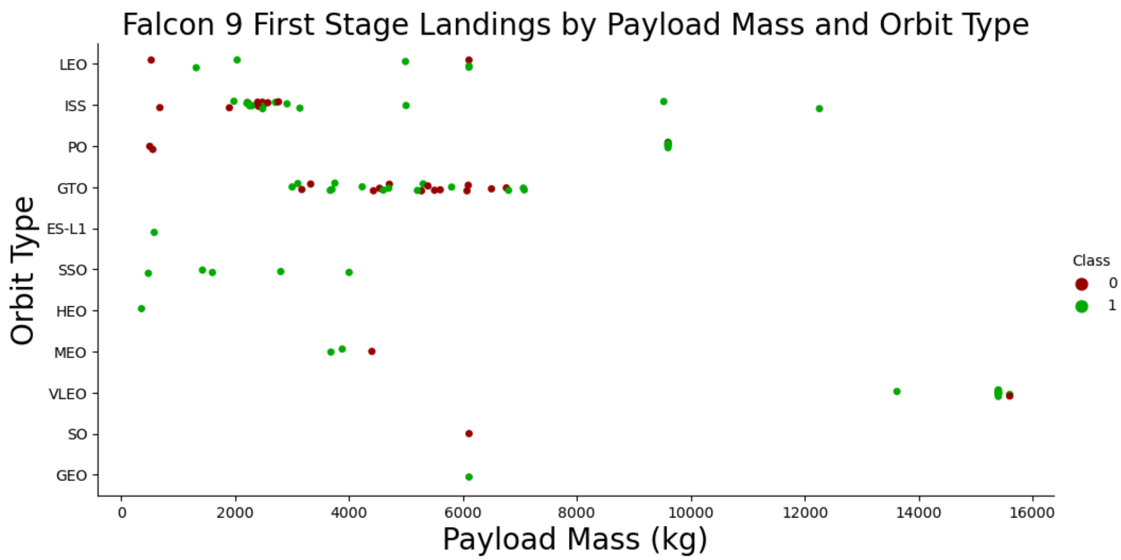
Task 5

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 [14]:

```
# Task 5 ANSWER
# Plot a scatter point chart with x axis to be Payload and y axis to be the
sns.catplot(data=df, x="PayloadMass", y="Orbit", hue="Class", aspect=2, pa
plt.xlabel("Payload Mass (kg)", fontsize=20)
plt.ylabel("Orbit Type", fontsize=20)
plt.title("Falcon 9 First Stage Landings by Payload Mass and Orbit Type", 
plt.show()
```



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

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 [15]:
```python
# 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
```

In [16]:
```python
df.head(3)
```

Out[16]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1

In [17]:
```python
df['year'] = Extract_year(df['Date'])
```

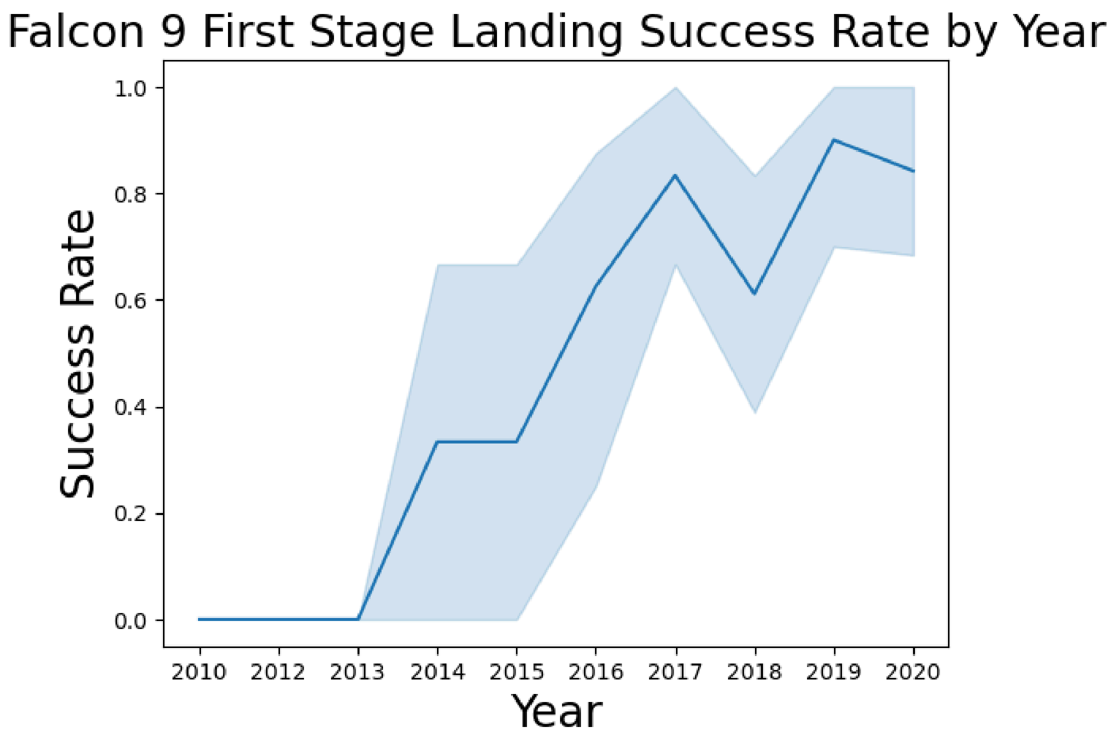
In [18]:
```python
df.head(3)
```

Out[18]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1

In [19]:

```
# Task 6 ANSWER
# Plot a line chart with x axis to be the extracted year and y axis to be 
sns.lineplot(data=df, x="year", y="Class")
plt.xlabel("Year", fontsize=20)
plt.ylabel("Success Rate", fontsize=20)
plt.title("Falcon 9 First Stage Landing Success Rate by Year", fontsize=20
plt.show()
```



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

Features Engineering

By now, you should 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 [20]:
```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flig
features.head()
```

Out[20]:

	FlightNumber	PayloadMass	Orbit	LaunchSite	Flights	GridFins	Reused	Legs	Landin
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	

Task 7

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 [21]:
```
# HINT: Use get_dummies() function on the categorical columns
features_one_hot = pd.get_dummies(features, columns=['Orbit', 'LaunchSite'
features_one_hot.head()
```

Out[21]:

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Orbi
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	

5 rows × 80 columns

Task
8

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 [22]:
```
# HINT: use astype function
features_one_hot.astype('float64')
```

Out[22]:

	FlightNumber	PayloadMass	Flights	GridFins	Reused	Legs	Block	ReusedCount	Ort
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

Export DataFrame to .CSV

Note
dataset_part_3.csv

In [23]:
```
# Export DataFrame as .csv
features_one_hot.to_csv('dataset_part_3.csv', index=False)
```

End Here