IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

R Johnson Manuel
23-04-2023

# Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

# Executive Summary

- Summary of methodologies
- Summary of all results

# Introduction

- Project background and context

There is no way to predict rockets success rate.

Problems you want to find answers

Analyze the data collected by previous launches and predict the falcon 9 launches will be successful in the future.

Section 1

# Methodology

# Methodology

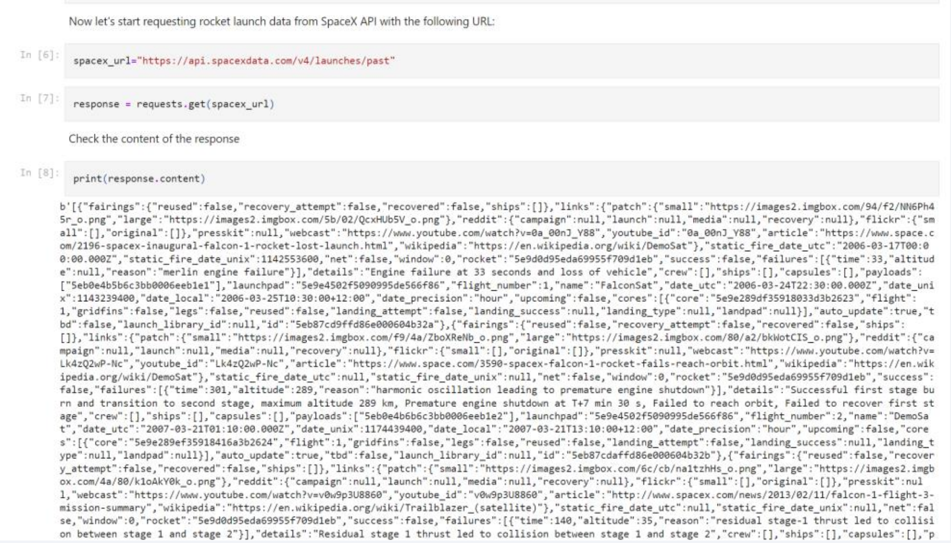
## Executive Summary

- Data collection methodology:
  - We use Data collection API and Web Scraping to collect data from Spacex and websites.
- Perform data wrangling
  - The process of cleaning the data for machine learning by removing the null values and other wrong values and converting categorical values to numeric.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We use scikit-learn to train and test our models.

# Data Collection

- Describe how data sets were collected.
  - By using web scraping method from various websites as well as using data provided by SpaceX
- You need to present your data collection process use key phrases and flowcharts
  - We use the Requests API in Python to Fetch the data from the websites and we use the Beautiful Soup package to process the webpage and extract the data from the content.
  - The Data collected has also been done by website having data related to the falcon 9s Ie: SpaceX has this data available on their research website. This is available on the IBM asset exchange as well.

# Data Collection – SpaceX API



Now let's start requesting rocket launch data from SpaceX API with the following URL:

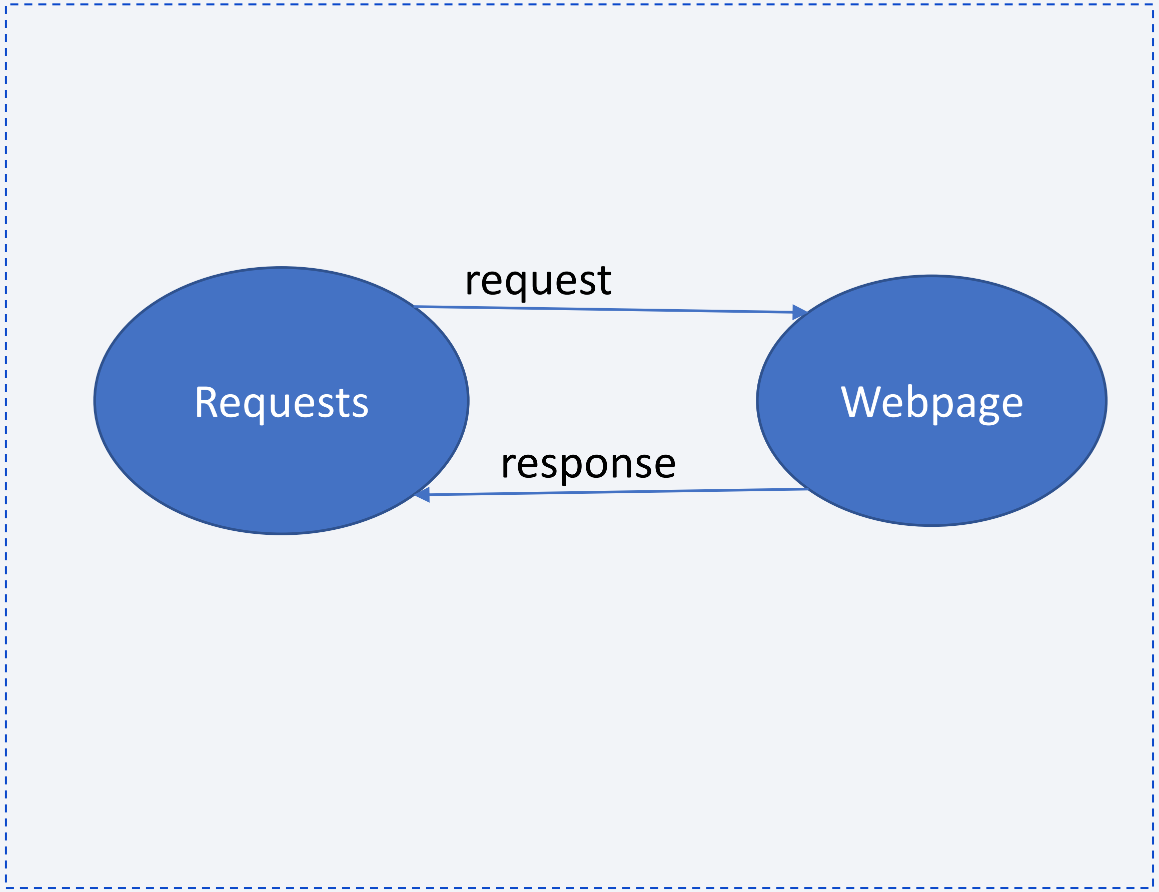
```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
In [7]: response = requests.get(spacex_url)
```

Check the content of the response

```
In [8]: print(response.content)
```

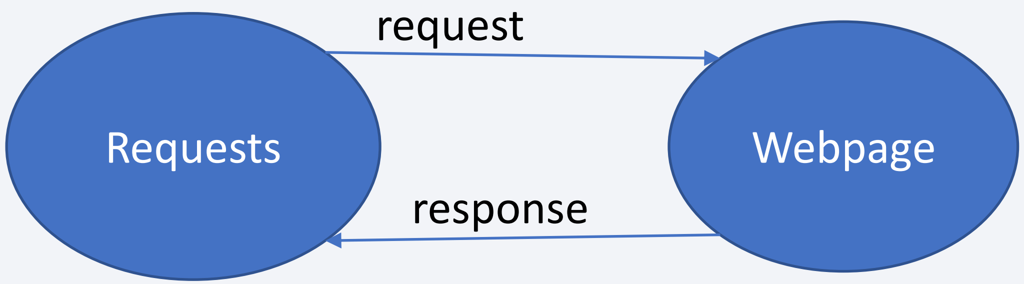
```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph4
5r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"sm
all":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.c
om/2196-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:0
0:00.000Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":33,"altitud
e":null,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew":[],"ships":[],"capsules":[],"payloads":
["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_uni
x":1143239400,"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e9e289df35918033d3b2623","flight":
1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"t
bd":false,"launch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":
[]},"links":{"patch":{"small":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgbox.com/80/a2/bkWotCIS_o.png"},"reddit":{"ca
mpaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=
Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wik
ipedia.org/wiki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":
false,"failures":[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}],"details":"Successful first stage bu
rn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first st
age","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4b6b6c3bb0006eeb1e2"],"launchpad":"5e9e4502f5090995de566f86","flight_number":2,"name":"DemoSa
t","date_utc":"2007-03-21T01:10:00.000Z","date_unix":1174439400,"date_local":"2007-03-21T13:10:00+12:00","date_precision":"hour","upcoming":false,"core
s":[{"core":"5e9e289ef35918416a3b2624","flight":1,"gridfins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_t
ype":null,"landpad":null}],"auto_update":true,"tbd":false,"launch_library_id":null,"id":"5eb87cdaffd86e000604b32b"},{"fairings":{"reused":false,"recover
y_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/6c/cb/na1tzhHs_o.png","large":"https://images2.imgb
ox.com/4a/80/k1oAkY0k_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":nul
l,"webcast":"https://www.youtube.com/watch?v=v0w9p3U8860","youtube_id":"v0w9p3U8860","article":"http://www.spacex.com/news/2013/02/11/falcon-1-flight-3-
mission-summary","wikipedia":"https://en.wikipedia.org/wiki/Trailblazer_(satellite)"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":fal
se,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"time":140,"altitude":35,"reason":"residual stage-1 thrust led to collisi
on between stage 1 and stage 2"}],"details":"Residual stage 1 thrust led to collision between stage 1 and stage 2","crew":[],"ships":[],"capsules":[],"p
```



Notebook GIT link:
https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Data%20Collection%20API%20LAB%201%20-%20IBM%20-%20Week%201.ipynb

# Data Collection - Scraping

- The webscraping is process of getting the data from webpages and converting them into pandas dataframes for wrangling and data processing
- Notebook URL:
- [https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Web%20Scraping%20LAB1%20-%20Week%201%20-%20IBM.ipynb](https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Web%20Scraping%20LAB1%20-%20Week%201%20-%20IBM.ipynb)



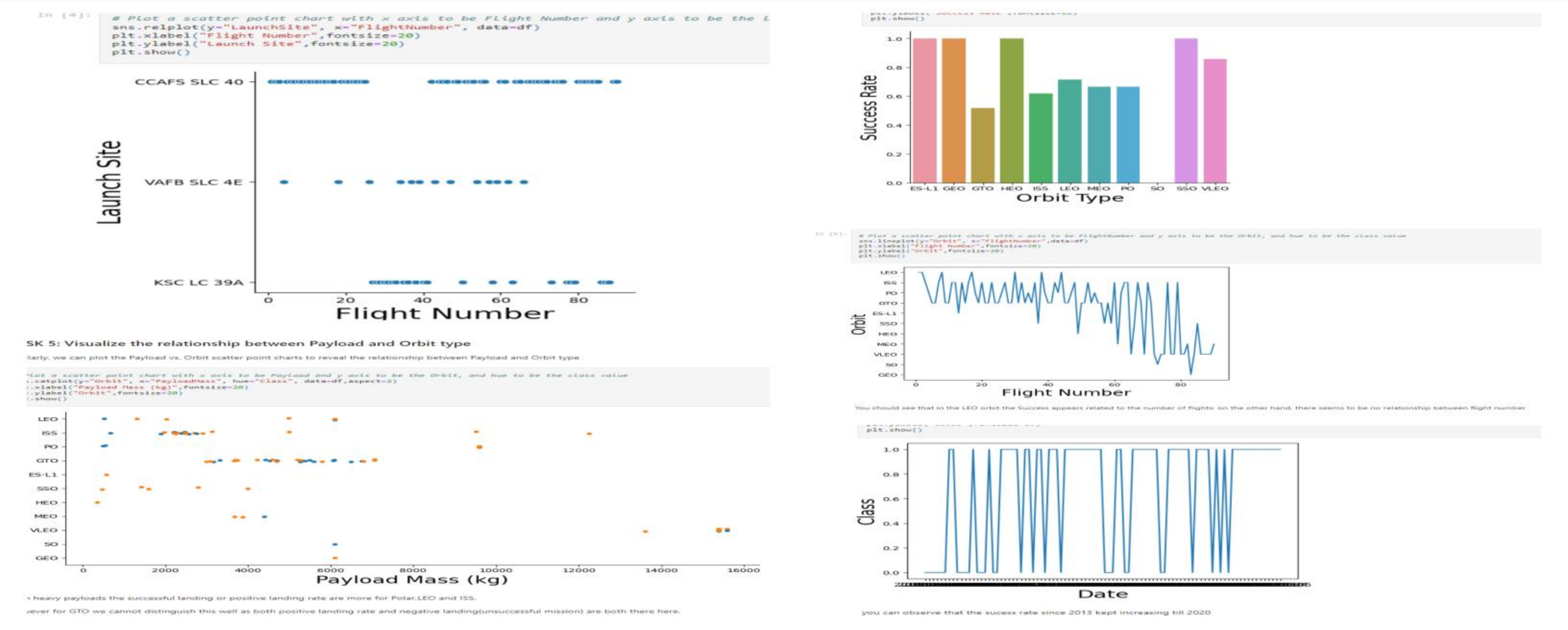
# Data Wrangling

- The data was cleaned by removing Null values .
- The data wrangling process removed empty values and converted Categorical values to weighted values
- The process one hot coding was used and data was replaced with columns to better represent them
- Wrong values were also removed and Replaced with either mean or average values in some cases.
- Notebook URL :

https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Data%20Wrangling%20-%20LAB%202%20%20-%20Week1%20-%20IBM.ipynb

# EDA with Data Visualization

- Charts plotted :



```
In [4]: # Plot a scatter point chart with x axis to be Flight Number and y axis to be the L
sns.relplot(y="LaunchSite", x="FlightNumber", data=df)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

SK 5: Visualize the relationship between Payload and Orbit type

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

```
lot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df,aspect=2)
.xlabel("Payload Mass (kg)",fontsize=20)
.ylabel("Orbit",fontsize=20)
.show()
```

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

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

```
plt.show()
```

```
In [8]: # Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.lineplot(y="Orbit", x="FlightNumber",data=df)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
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

```
plt.show()
```

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

- Github Link : https://github.com/JohnsonManuel/Capstone-IBM/blob/master/EDA%20WITH%20Visialization-%20IBM%20-%20WEEK%202.ipynb

# EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- The Sql Queries were done on the various falcon 9 Launches and its success and failures from different periods of time were calculated and were displayed as tables.
- The Data gave a clear overview of what to expect in terms of wrangling and data analysis standpoints.
- GitHub Notebook URL: https://github.com/JohnsonManuel/Capstone-IBM/blob/master/EDA%20with%20SQL%20-%20IBM%20-%20Week%202.ipynb

# Build an Interactive Map with Folium

- The folium the map was drawn with NASA Johnson Space Center at Houston, Texas as its center . Some of the launch sites in the data are marked down.The success and failures of each launch sites are also marked.
- The distance between the launch sites are also marked down. To extract more information on the data set.
- Github Notebook URL: https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Interactive%20Visual%20Analytics%20with%20Folium%20Lab%20-%20Week%203.ipynb

# Predictive Analysis (Classification)

- The data was loaded and the train/test split was done . The model was created on the training set and the GridSearchCV object was used.
- A Confusion matrix was formed and the best parameters for a grid search was also created and the model was trained again.
- Finally the model gave a prediction accuracy of 83- 84 percent.

- GitHub Notebook URL: https://github.com/JohnsonManuel/Capstone-IBM/blob/master/Machine%20Learning%20Prediction%20lab.ipynb

# Results

- The data was analyzed in various ways and the values are understood .
- The Data was plotted in various graphs and then used to represent the values in various ways.
- Finally the data was trained and then the model gave a prediction accuracy of 84%

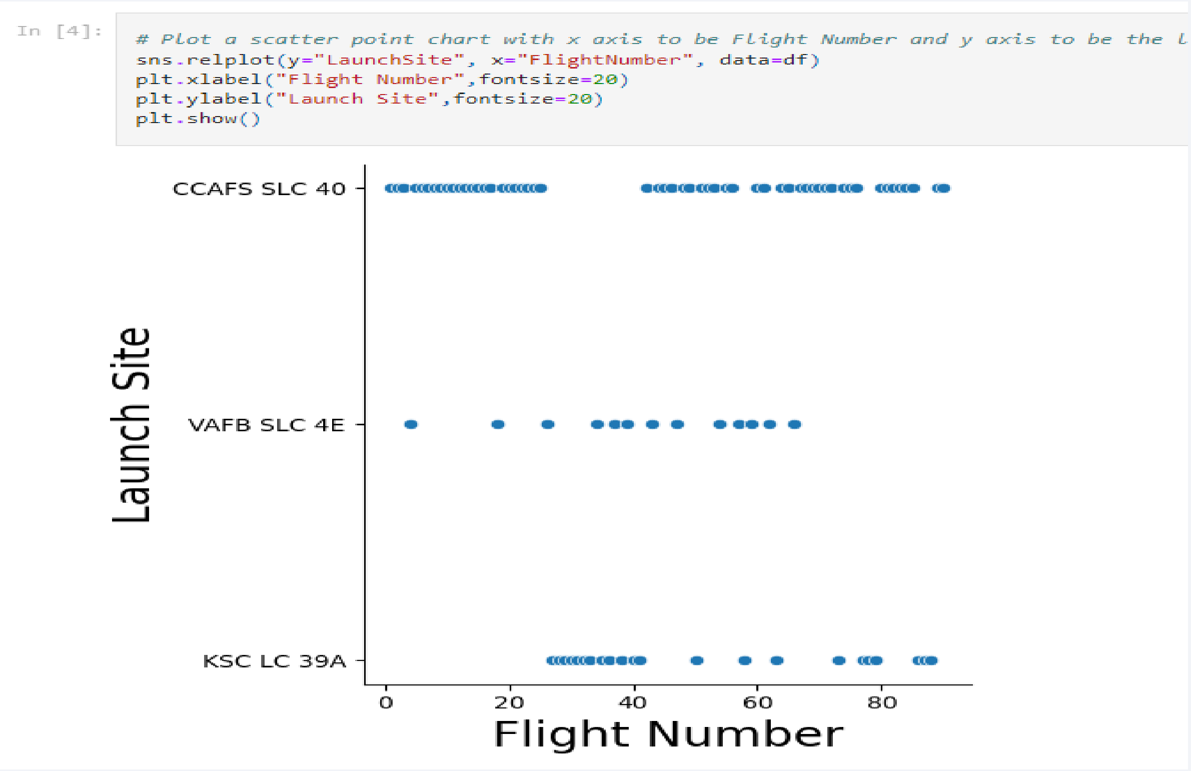
Section 2

# Insights drawn from EDA

# Flight Number vs. Launch Site

- Show a scatter plot of Flight Number vs. Launch Site

- Show the screenshot of the scatter plot with explanations



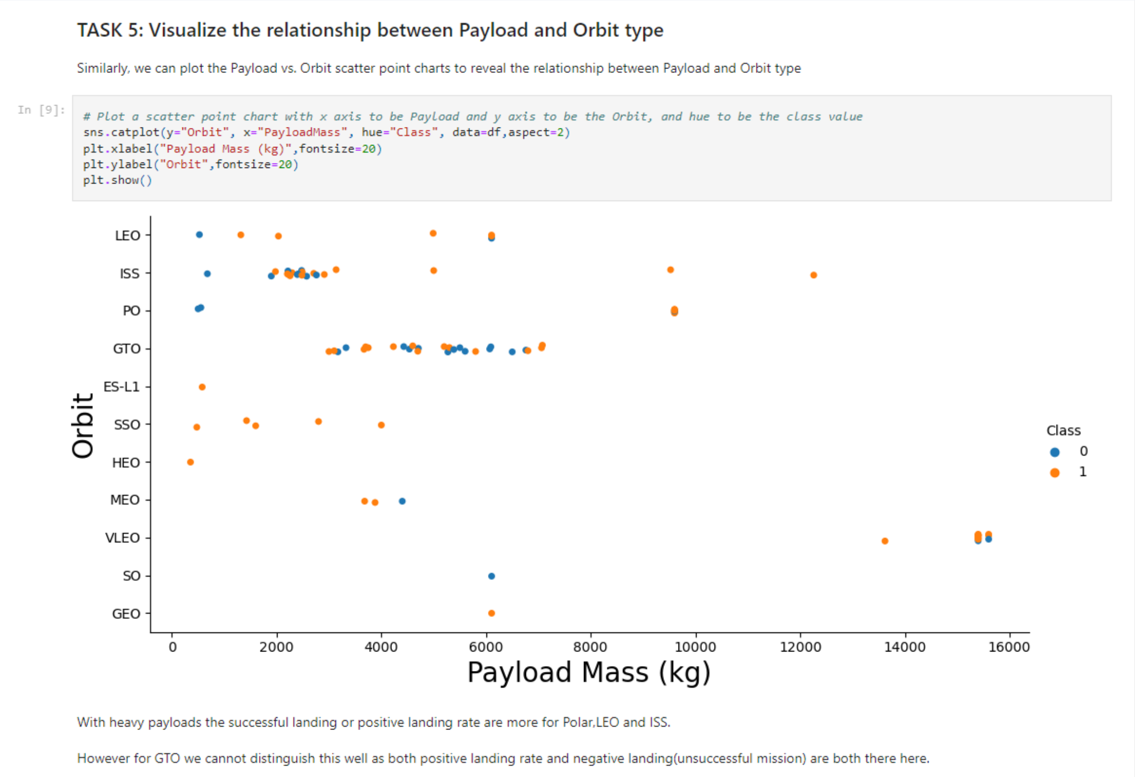
In [4]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the L
sns.relplot(y="LaunchSite", x="FlightNumber", data=df)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

# Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations



### 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 [9]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df,aspect=2)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

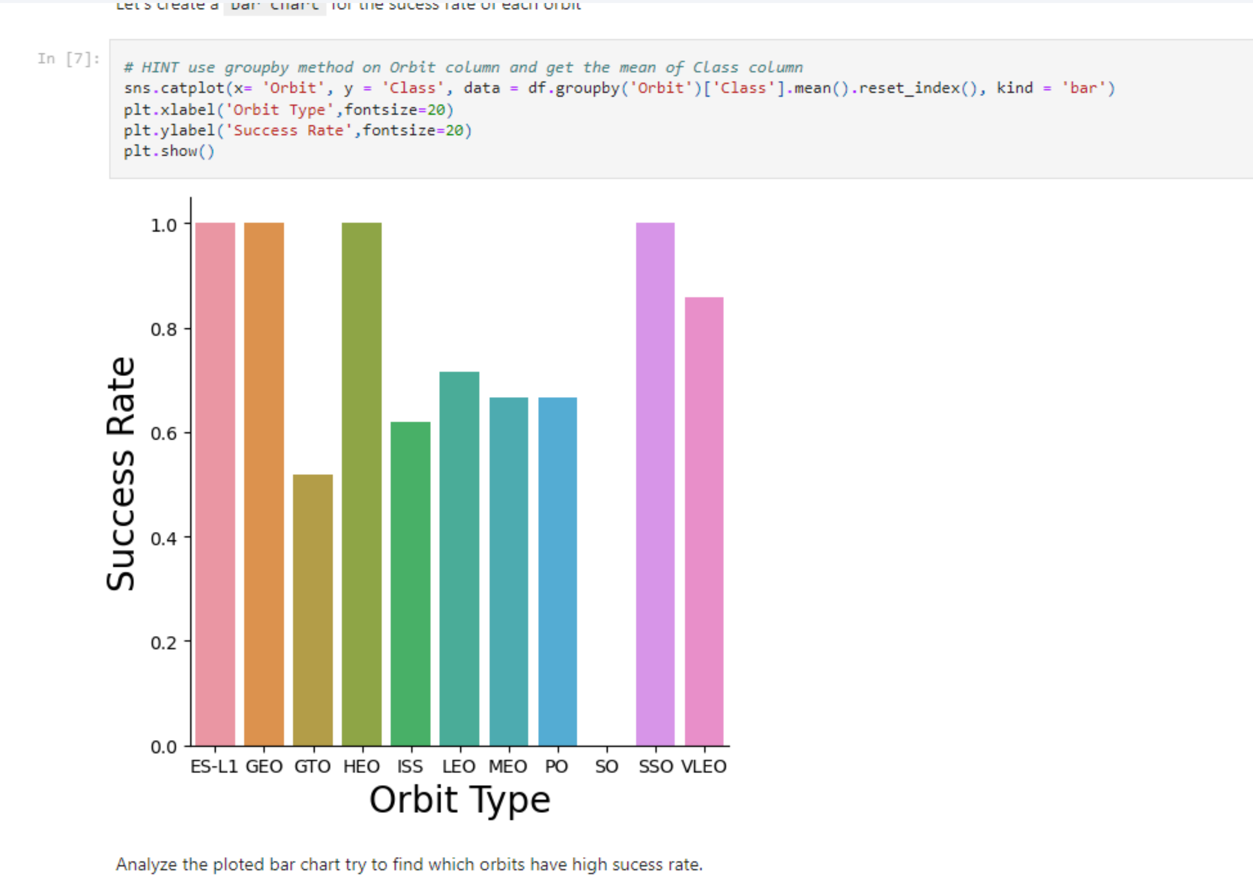
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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

# Success Rate vs. Orbit Type

- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations



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

In [7]:

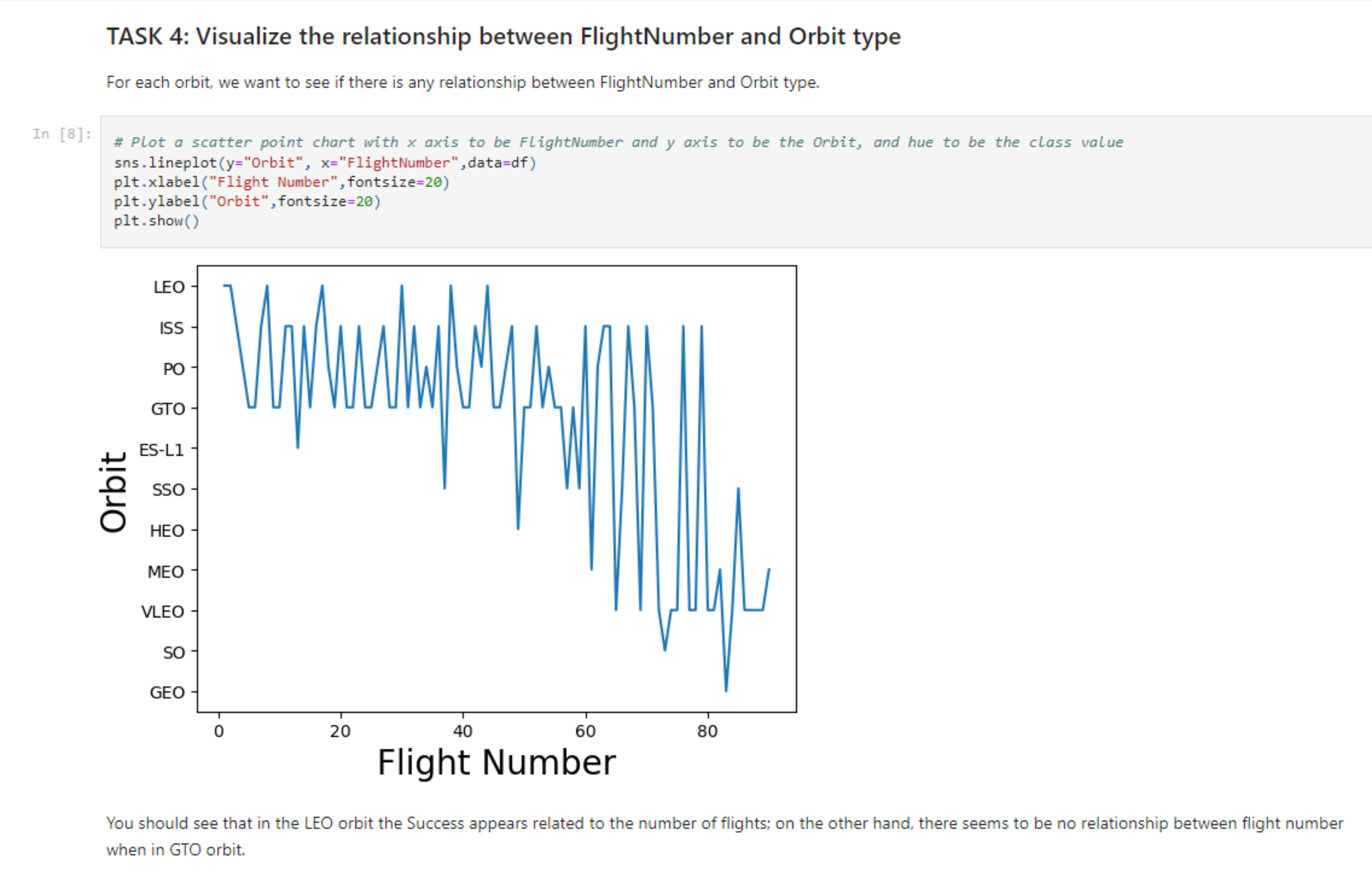
```
# HINT use groupby method on Orbit column and get the mean of Class column
sns.catplot(x= 'Orbit', y = 'Class', data = df.groupby('Orbit')['Class'].mean().reset_index(), kind = 'bar')
plt.xlabel('Orbit Type',fontsize=20)
plt.ylabel('Success Rate',fontsize=20)
plt.show()
```

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

# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

- Show the screenshot of the scatter plot with explanations



### 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 [8]:

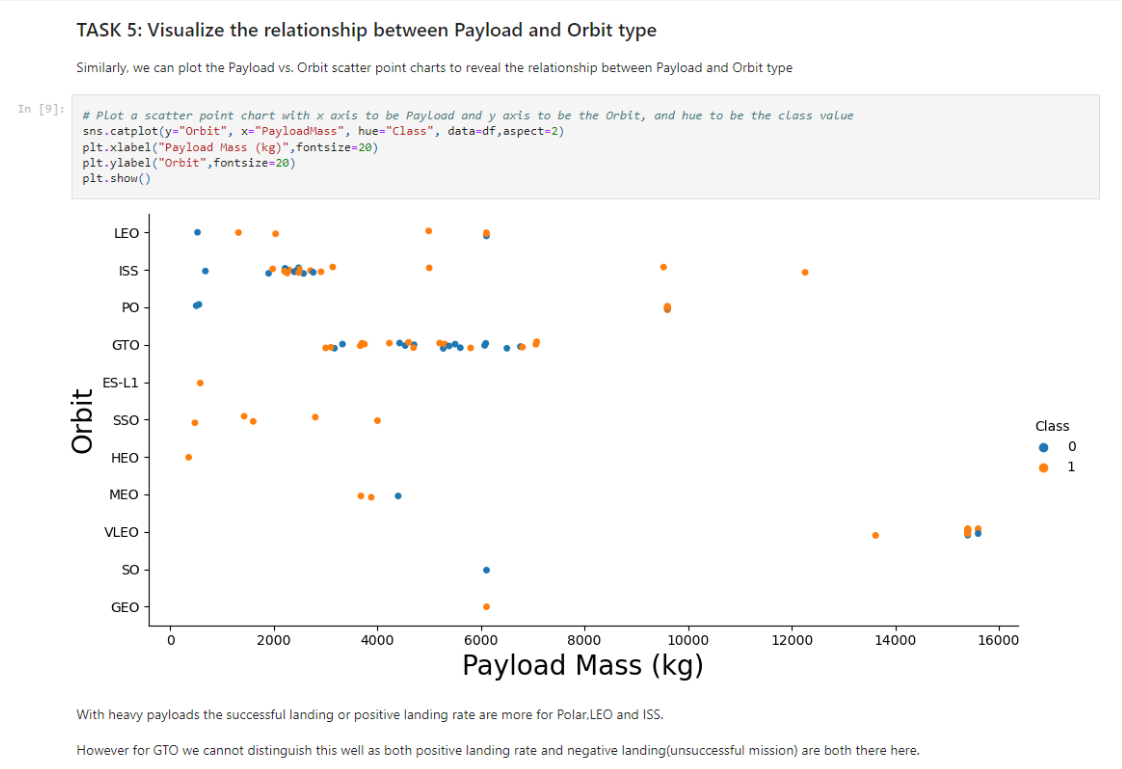
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.lineplot(y="Orbit", x="FlightNumber",data=df)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

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.

# Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type

- Show the screenshot of the scatter plot with explanations



### 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 [9]:

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df,aspect=2)
plt.xlabel("Payload Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

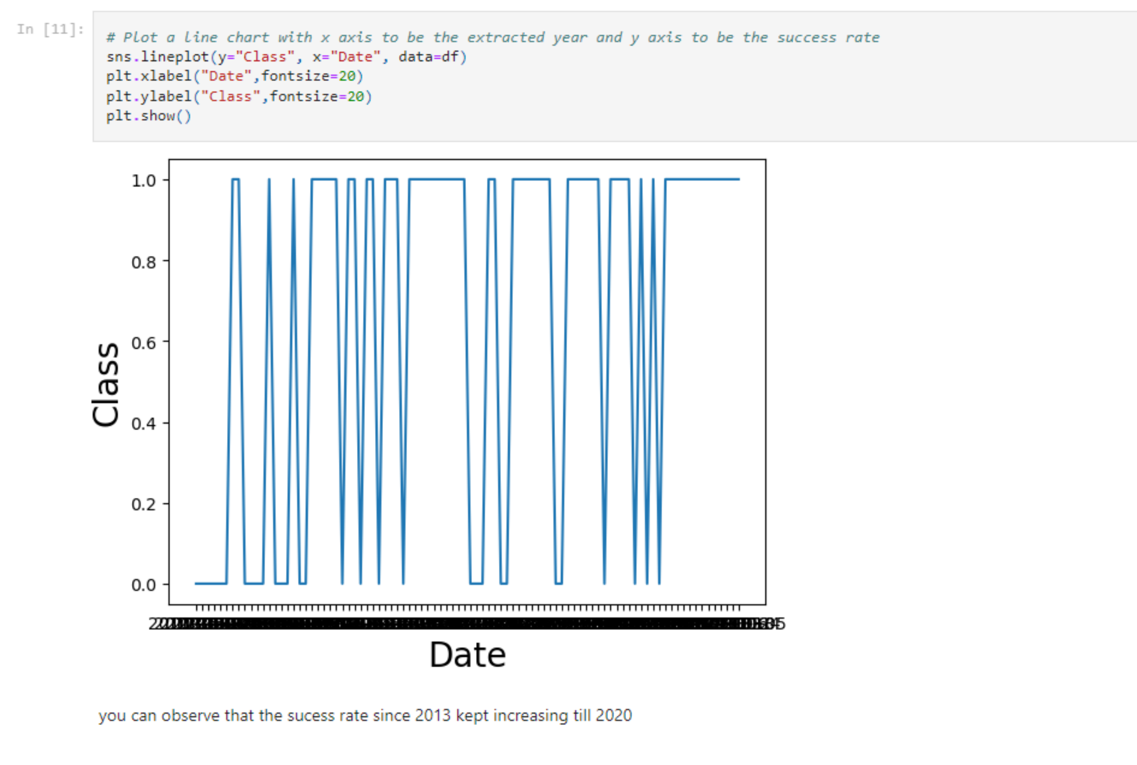
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

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

# Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations



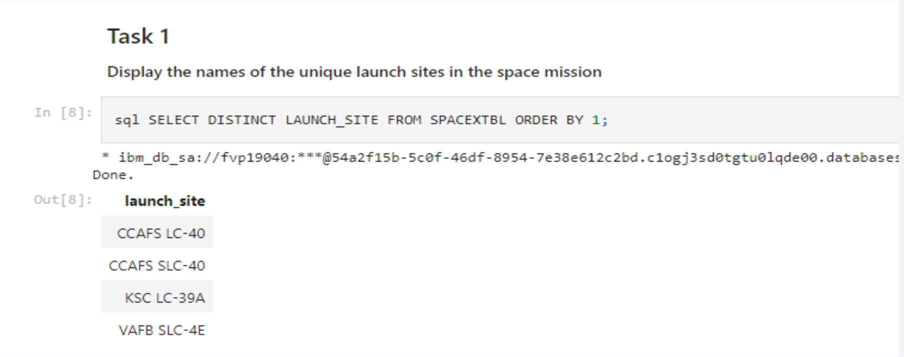
In [11]:

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(y="Class", x="Date", data=df)
plt.xlabel("Date",fontsize=20)
plt.ylabel("Class",fontsize=20)
plt.show()
```

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

# All Launch Site Names

- Find the names of the unique launch sites



Task 1

Display the names of the unique launch sites in the space mission

In [8]:
```
sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases
Done.
```

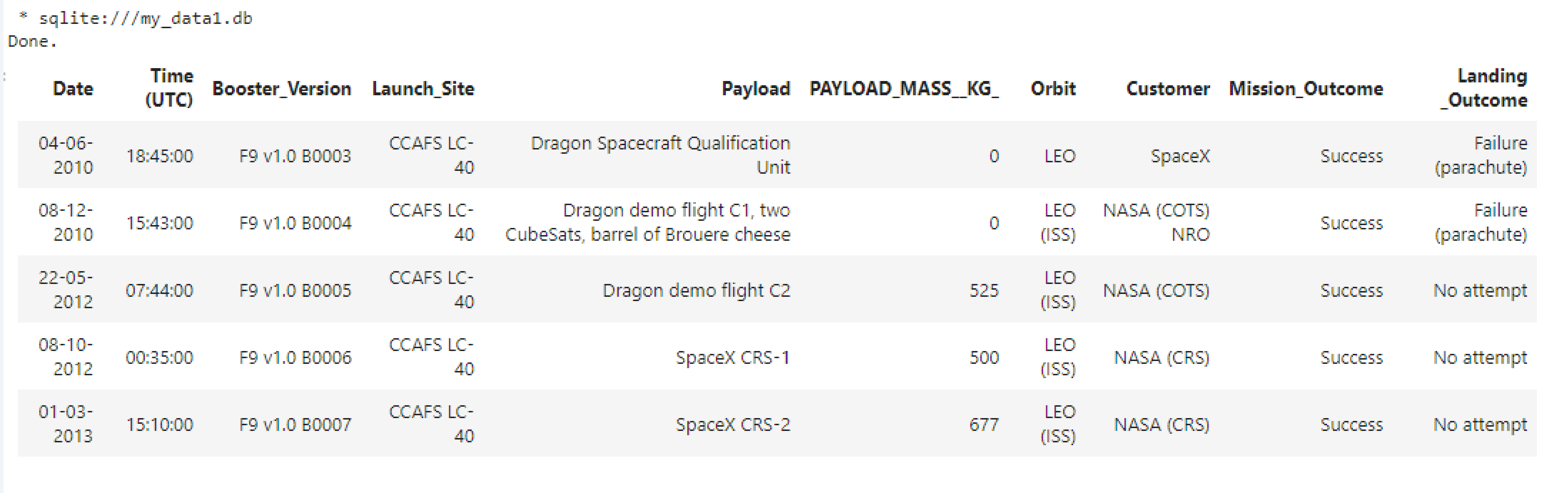
Out[8]:

| launch_site |
|---|
| CCAFS LC-40 |
| CCAFS SLC-40 |
| KSC LC-39A |
| VAFB SLC-4E |

- Launch sites are clearly mentioned above

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

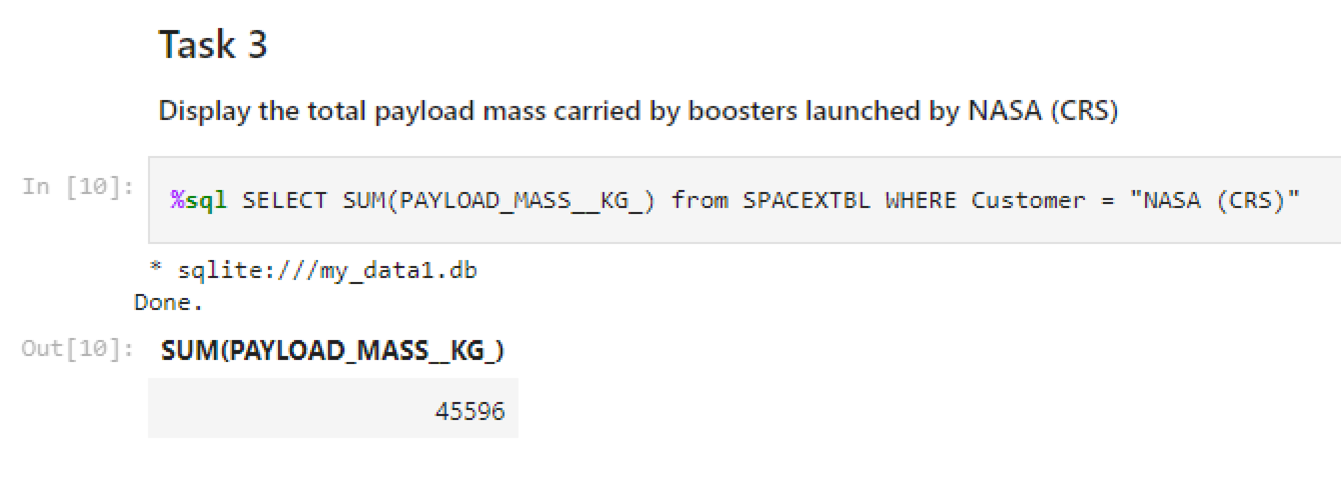


```
 * sqlite:///my_data1.db
Done.
```

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ | Orbit | Customer | Mission_Outcome | Landing _Outcome |
|---|---|---|---|---|---|---|---|---|---|
| 04-06-2010 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 | LEO | SpaceX | Success | Failure (parachute) |
| 08-12-2010 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 | LEO (ISS) | NASA (COTS) NRO | Success | Failure (parachute) |
| 22-05-2012 | 07:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 | LEO (ISS) | NASA (COTS) | Success | No attempt |
| 08-10-2012 | 00:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 | LEO (ISS) | NASA (CRS) | Success | No attempt |
| 01-03-2013 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 | LEO (ISS) | NASA (CRS) | Success | No attempt |

# Total Payload Mass

- Calculate the total payload carried by boosters from NASA



### Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

```
In [10]: %sql SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Customer = "NASA (CRS)"
```

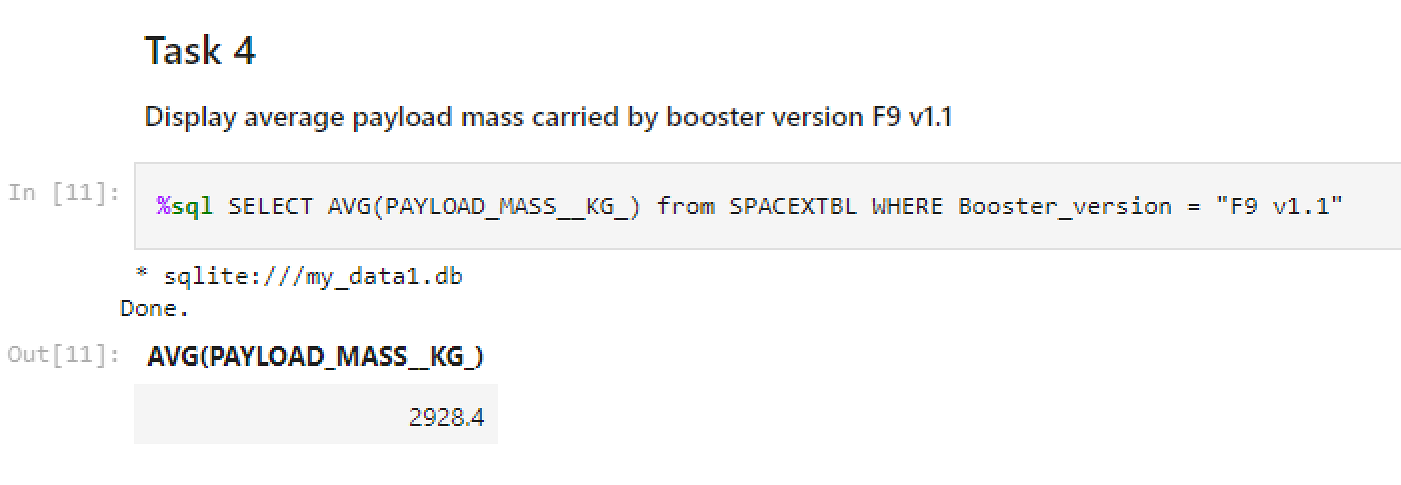
```
 * sqlite:///my_data1.db
Done.
```

Out[10]:

| SUM(PAYLOAD_MASS__KG_) |
|---|
| 45596 |

# Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1



### Task 4

Display average payload mass carried by booster version F9 v1.1

```
In [11]: %sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Booster_version = "F9 v1.1"
```

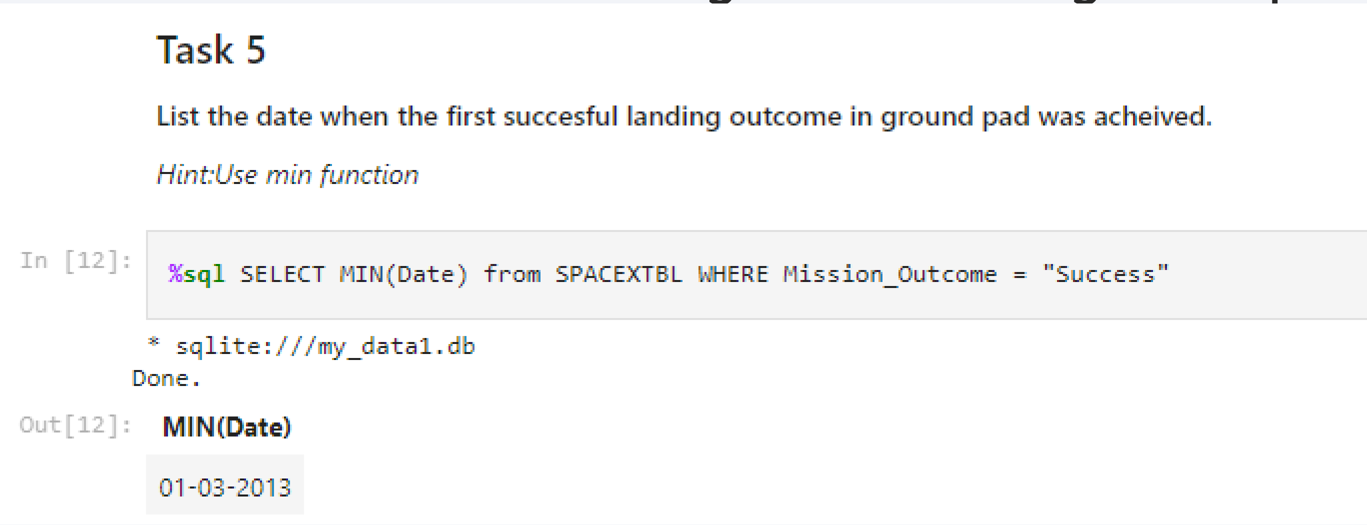
```
 * sqlite:///my_data1.db
Done.
```

Out[11]:

| AVG(PAYLOAD_MASS__KG_) |
|---|
| 2928.4 |

# First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



## Task 5

List the date when the first succesful landing outcome in ground pad was acheived.

*Hint:Use min function*

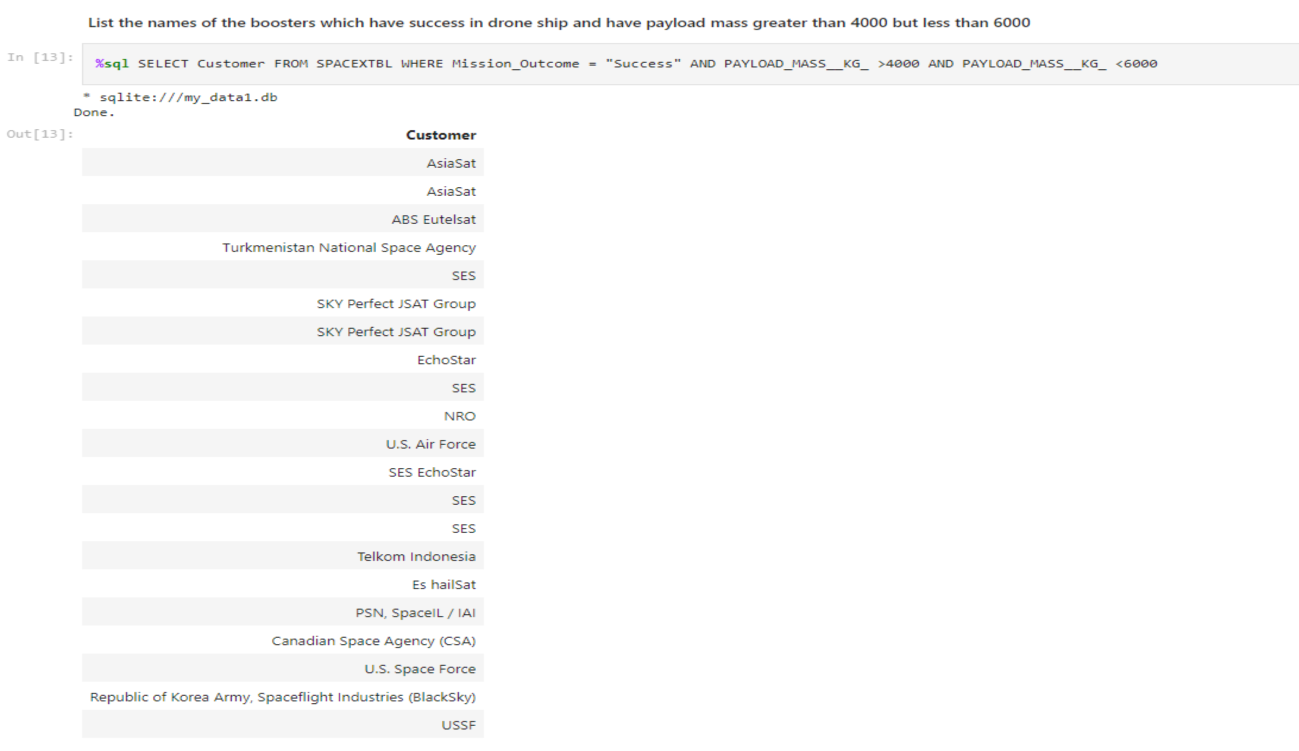
```
In [12]: %sql SELECT MIN(Date) from SPACEXTBL WHERE Mission_Outcome = "Success"

 * sqlite:///my_data1.db
Done.
```

Out[12]:

| MIN(Date) |
|---|
| 01-03-2013 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
In [13]: %sql SELECT Customer FROM SPACEXTBL WHERE Mission_Outcome = "Success" AND PAYLOAD_MASS__KG_ >4000 AND PAYLOAD_MASS__KG_ <6000
```

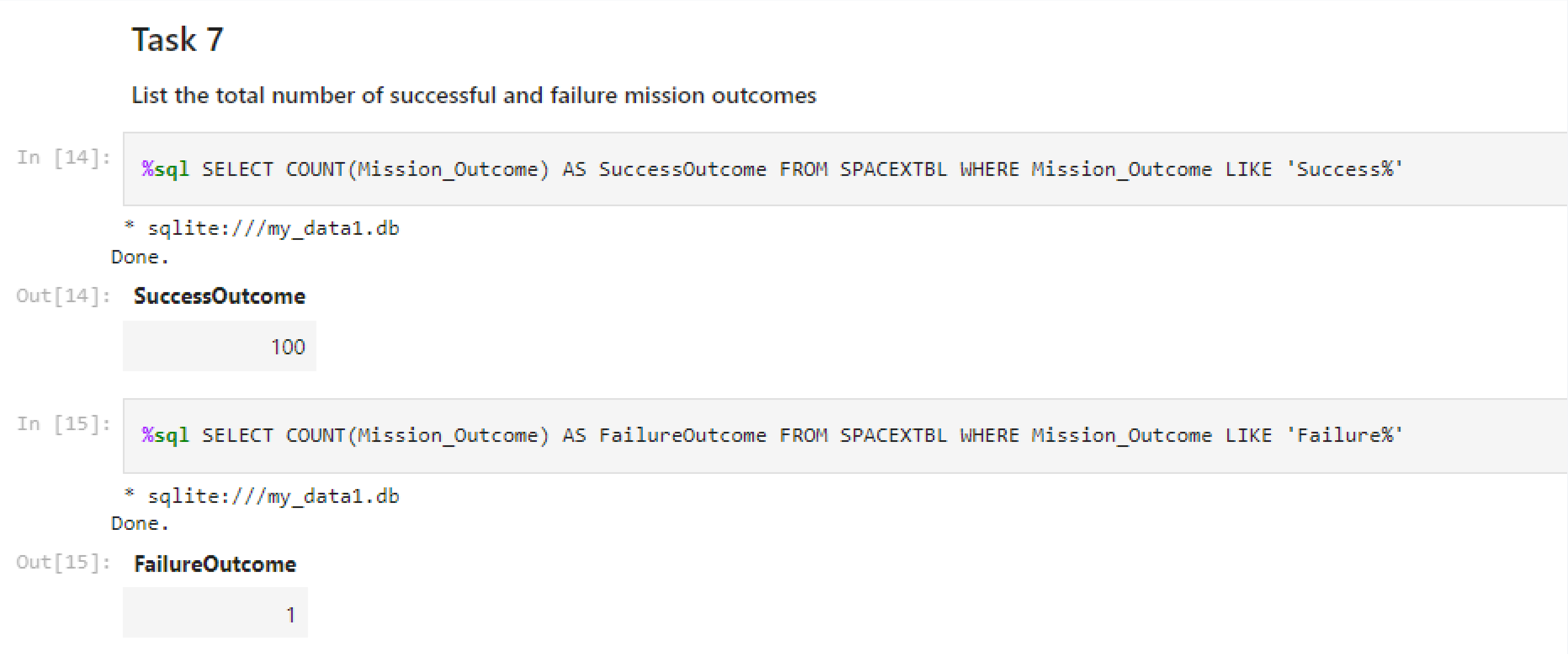
```
 * sqlite:///my_data1.db
Done.
```

Out[13]:

| Customer |
|---:|
| AsiaSat |
| AsiaSat |
| ABS Eutelsat |
| Turkmenistan National Space Agency |
| SES |
| SKY Perfect JSAT Group |
| SKY Perfect JSAT Group |
| EchoStar |
| SES |
| NRO |
| U.S. Air Force |
| SES EchoStar |
| SES |
| SES |
| Telkom Indonesia |
| Es hailSat |
| PSN, SpaceIL / IAI |
| Canadian Space Agency (CSA) |
| U.S. Space Force |
| Republic of Korea Army, Spaceflight Industries (BlackSky) |
| USSF |

# Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



## Task 7

List the total number of successful and failure mission outcomes

```
In [14]: %sql SELECT COUNT(Mission_Outcome) AS SuccessOutcome FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Success%'
```

```
 * sqlite:///my_data1.db
Done.
```

Out[14]:

| SuccessOutcome |
|---|
| 100 |

```
In [15]: %sql SELECT COUNT(Mission_Outcome) AS FailureOutcome FROM SPACEXTBL WHERE Mission_Outcome LIKE 'Failure%'
```

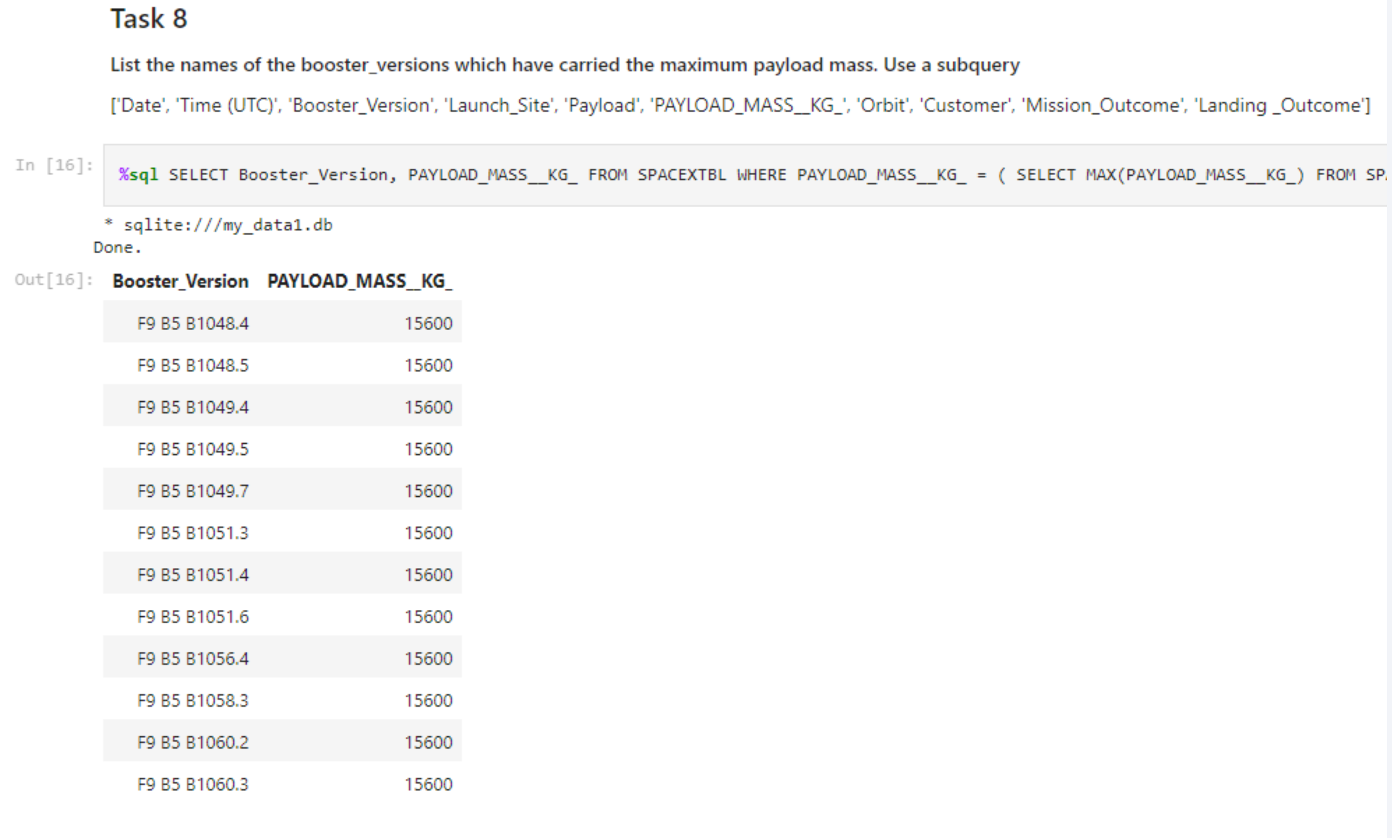
```
 * sqlite:///my_data1.db
Done.
```

Out[15]:

| FailureOutcome |
|---|
| 1 |

# Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



### Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

['Date', 'Time (UTC)', 'Booster_Version', 'Launch_Site', 'Payload', 'PAYLOAD_MASS__KG_', 'Orbit', 'Customer', 'Mission_Outcome', 'Landing _Outcome']

```
In [16]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = ( SELECT MAX(PAYLOAD_MASS__KG_) FROM SP
```

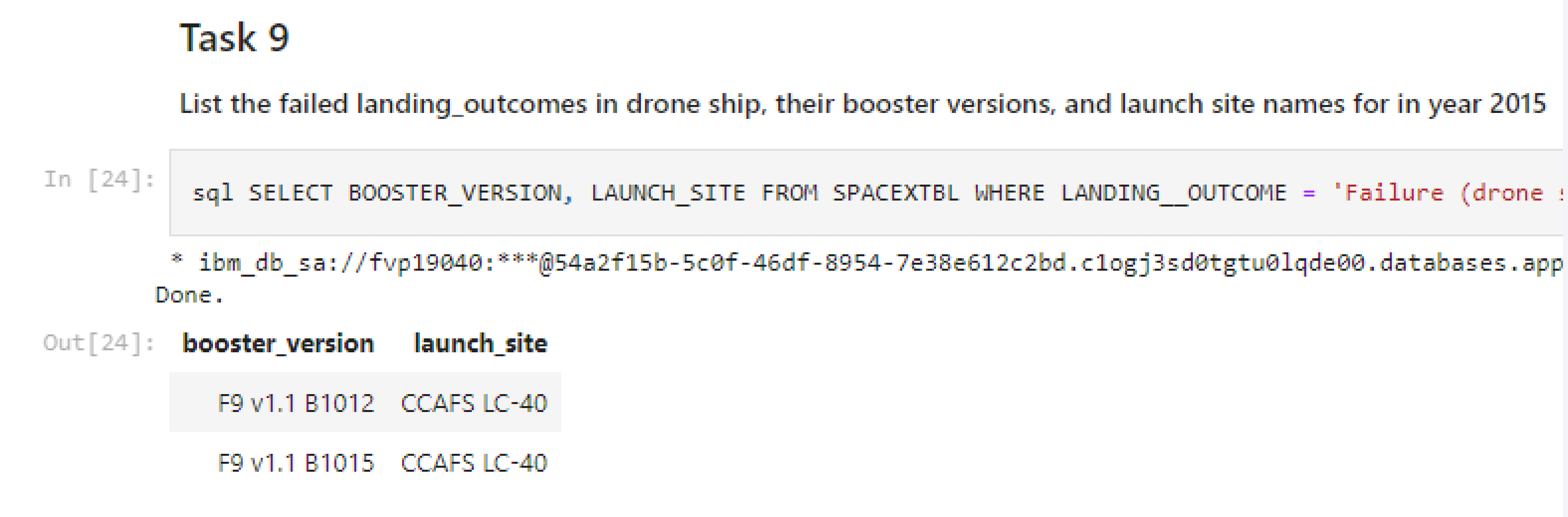
```
 * sqlite:///my_data1.db
Done.
```

Out[16]:

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1049.7 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1060.3 | 15600 |

# 2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015



### Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
In [24]:  sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone s
```

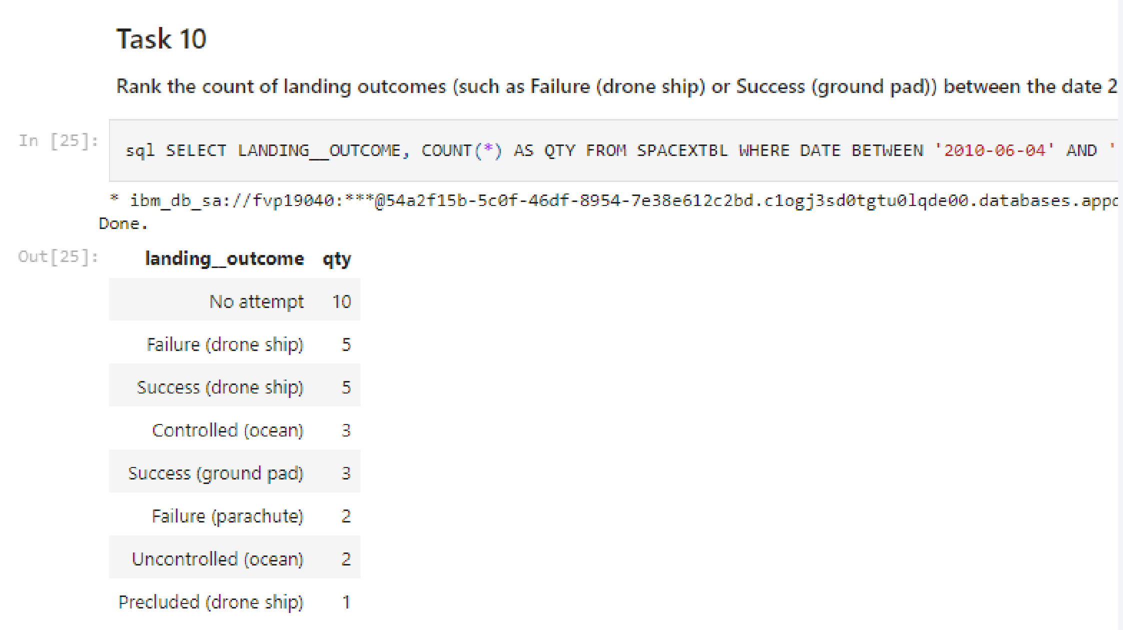
```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.app
Done.
```

Out[24]:

| booster_version | launch_site |
|---|---|
| F9 v1.1 B1012 | CCAFS LC-40 |
| F9 v1.1 B1015 | CCAFS LC-40 |

# Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order



### Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2

In [25]:

```
sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '
```

```
 * ibm_db_sa://fvp19040:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appc
Done.
```

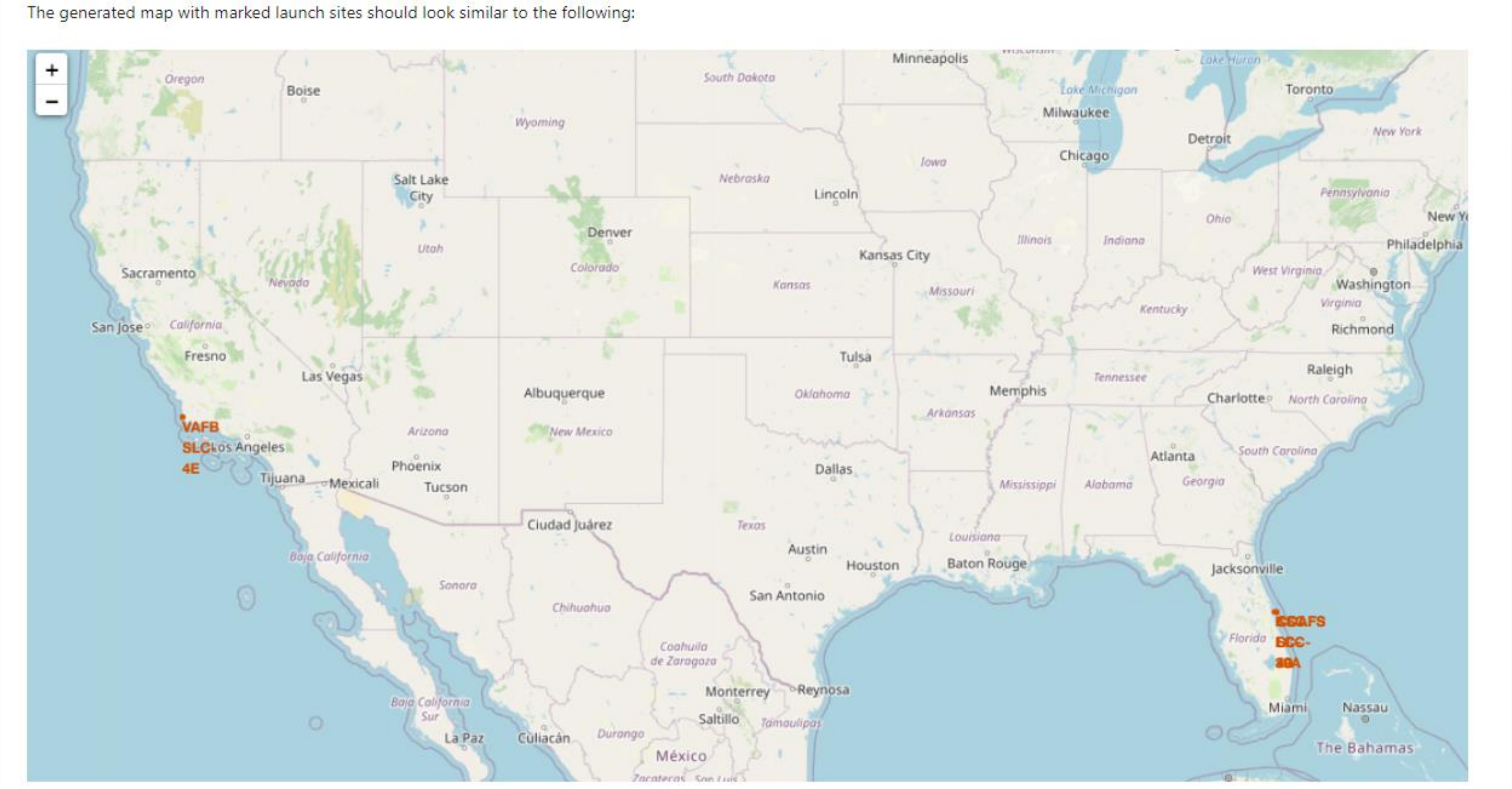
Out[25]:

| landing__outcome | qty |
|---|---|
| No attempt | 10 |
| Failure (drone ship) | 5 |
| Success (drone ship) | 5 |
| Controlled (ocean) | 3 |
| Success (ground pad) | 3 |
| Failure (parachute) | 2 |
| Uncontrolled (ocean) | 2 |
| Precluded (drone ship) | 1 |

Section 3

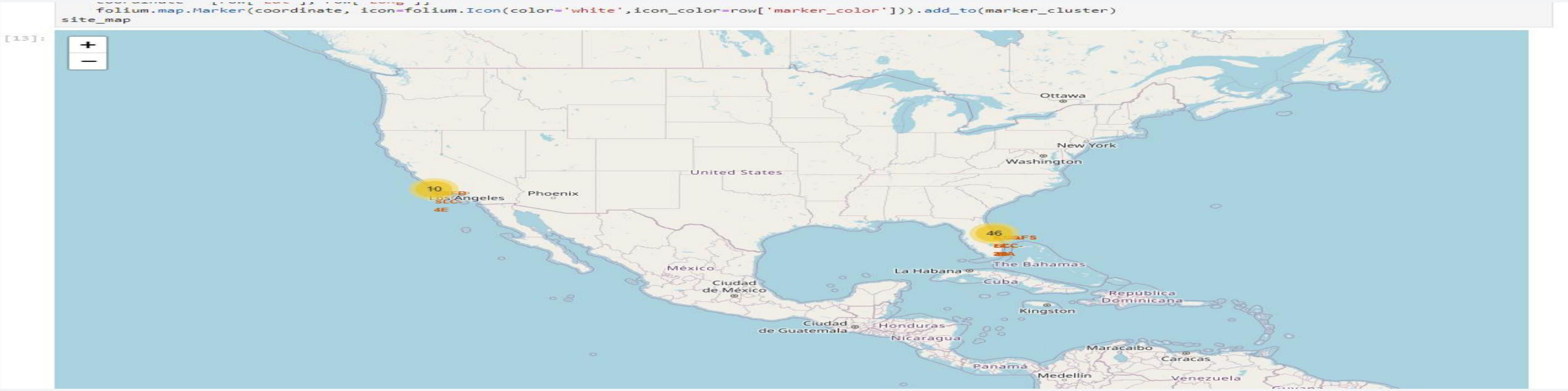
# Launch Sites Proximities Analysis

# All launch sites on a map

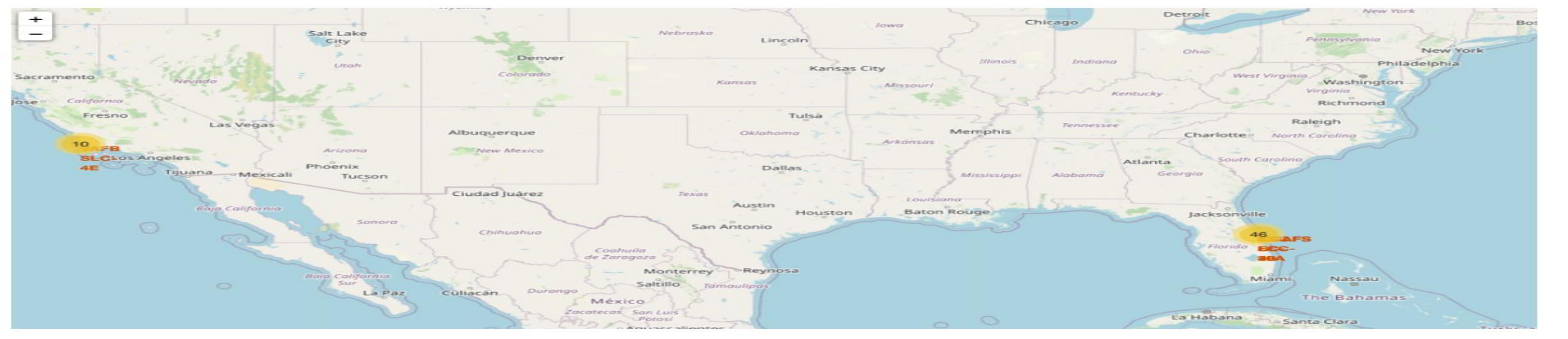


The generated map with marked launch sites should look similar to the following:

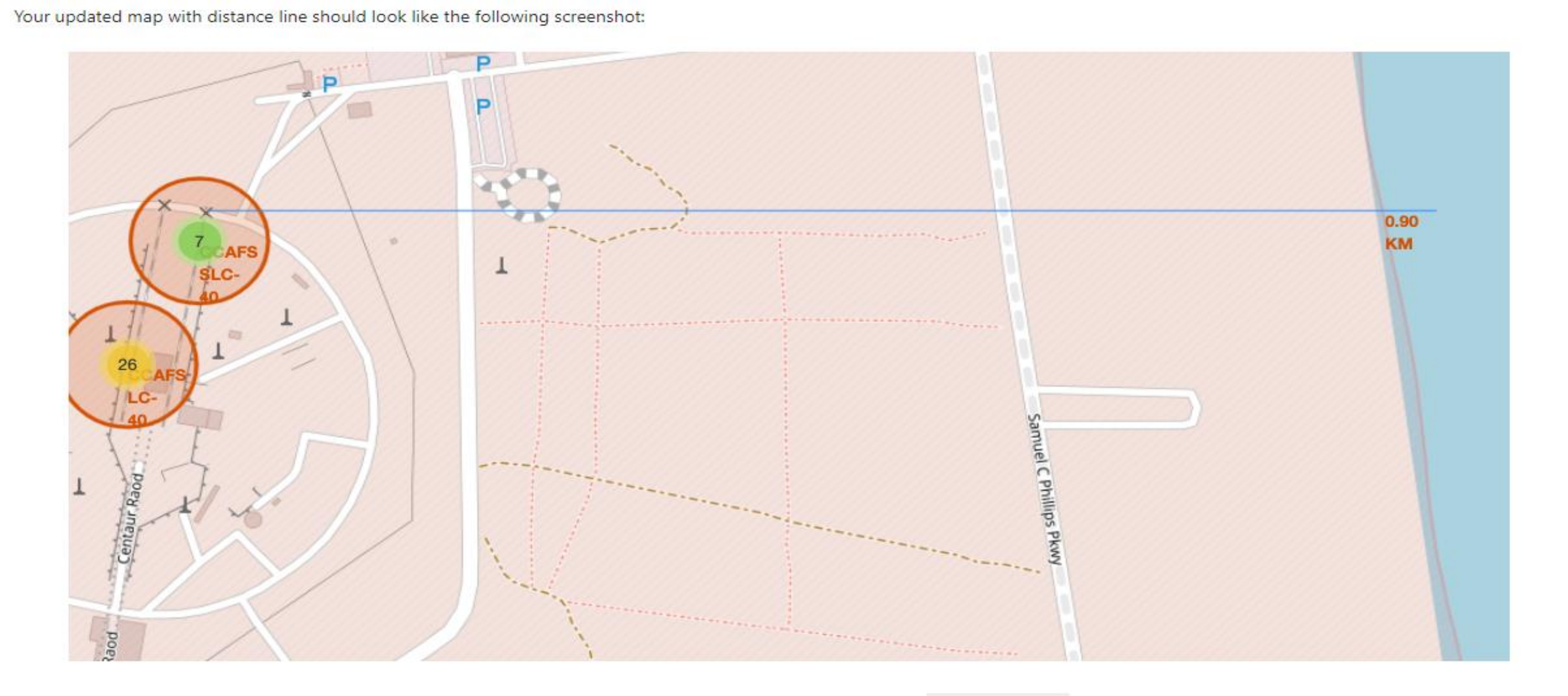
# Locations of success and failures in landings



```
    folium.map.Marker(coordinate, icon=folium.Icon(color='white',icon_color=row['marker_color'])).add_to(marker_cluster)
site_map
```



# Distance between the sites

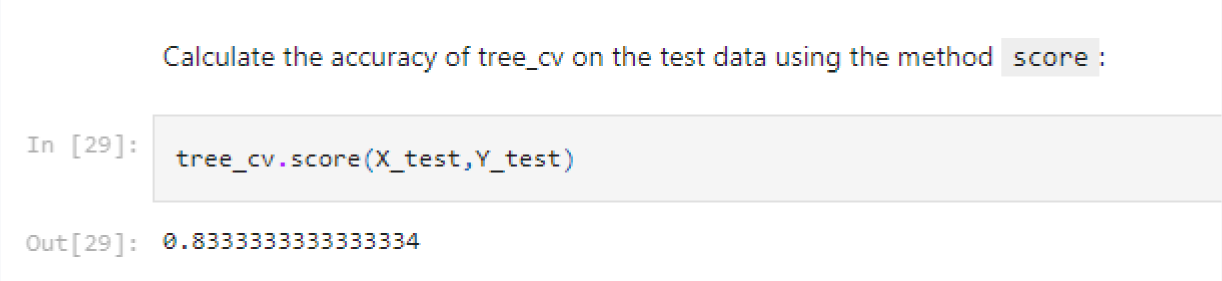


Your updated map with distance line should look like the following screenshot:

Section 5

# Predictive Analysis (Classification)

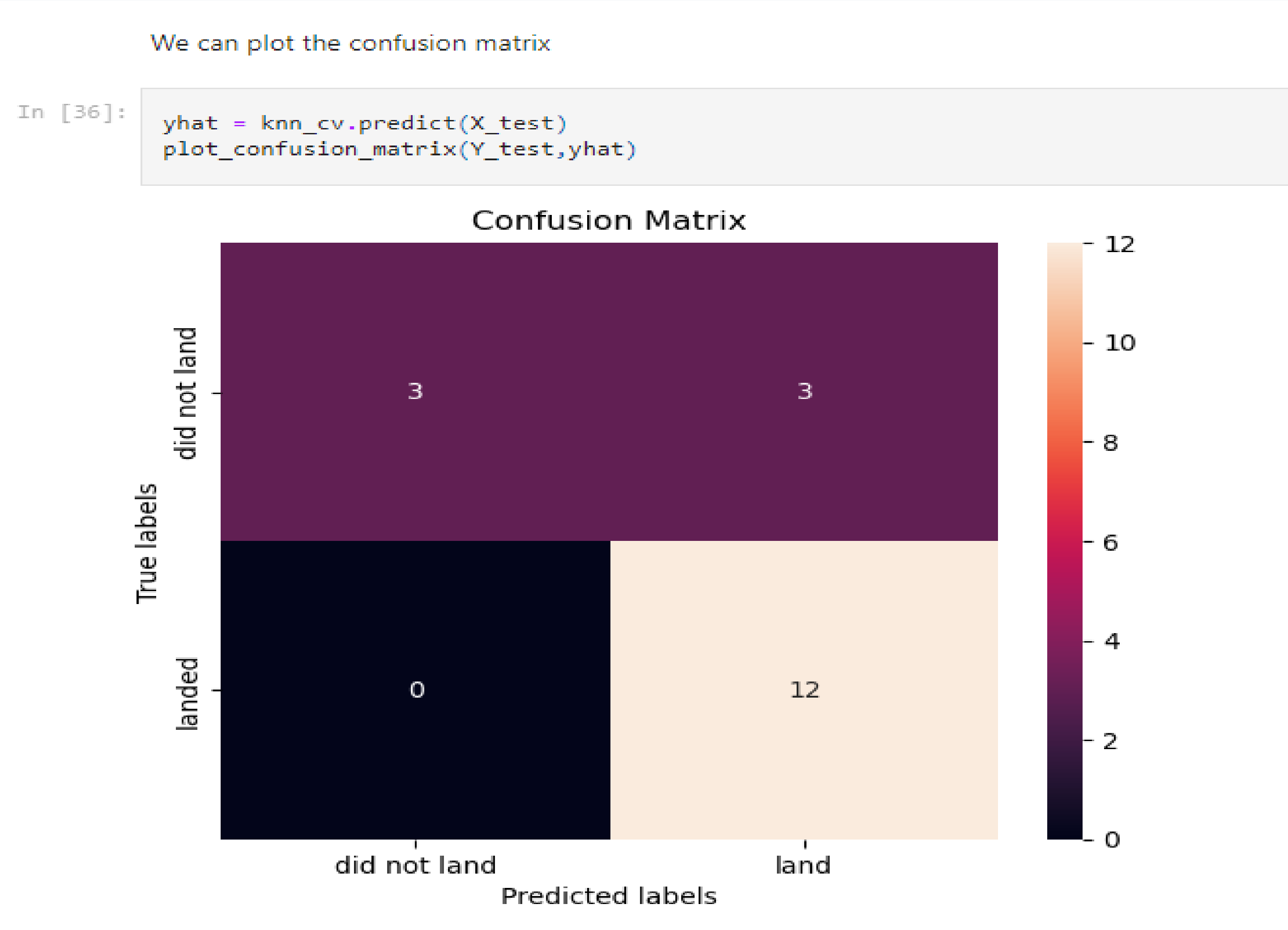
# Classification Accuracy



Calculate the accuracy of tree_cv on the test data using the method `score` :

```
In [29]: tree_cv.score(X_test,Y_test)

Out[29]: 0.8333333333333334
```

# Confusion Matrix



We can plot the confusion matrix

In [36]:

```
yhat = knn_cv.predict(X_test)
plot_confusion_matrix(Y_test,yhat)
```

## Thank you!