IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Maryia Makhnach>
<3/5/2022>

Outline

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

Executive Summary

- In my report I used the following methodoligies for data collection: API and Web Scrapping with BeautifulSoup python library;
- For data cleaning: Data Wrangling
- Exploratory Data Analysis: SQL, Data Visualization with matplotlib, seaborn libraries
- Interactive Visual Analytics with Folium
- Machine Learning Prediction using 4 methods: KNN, Logistic regression, SVM and Decision Tree

Results: Very deep EDA, app Dashboard, interactive maps. Three out four ML prediction methods have similar accuracy around 0.833%.

Introduction

- Project background and context

Space X 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 because Space X can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. In this lab, you will create a machine learning pipeline to predict if the first stage will land given the data from the preceding labs.

- Problems you want to find answers

–Factors and interactions amongst various features that determine the success rate of a landing.

– How can we predict the most accurate results for successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

Data was collected using SpaceX API and BeatifulSoup web scraping from Wikipedia.

- Perform data wrangling

One-hot encoding was applied to categorical features.

- Perform exploratory data analysis (EDA) using pandas, matplotlib, seaborn visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

KNN, Logistic regression, SVM and Decision Tree

Data Collection

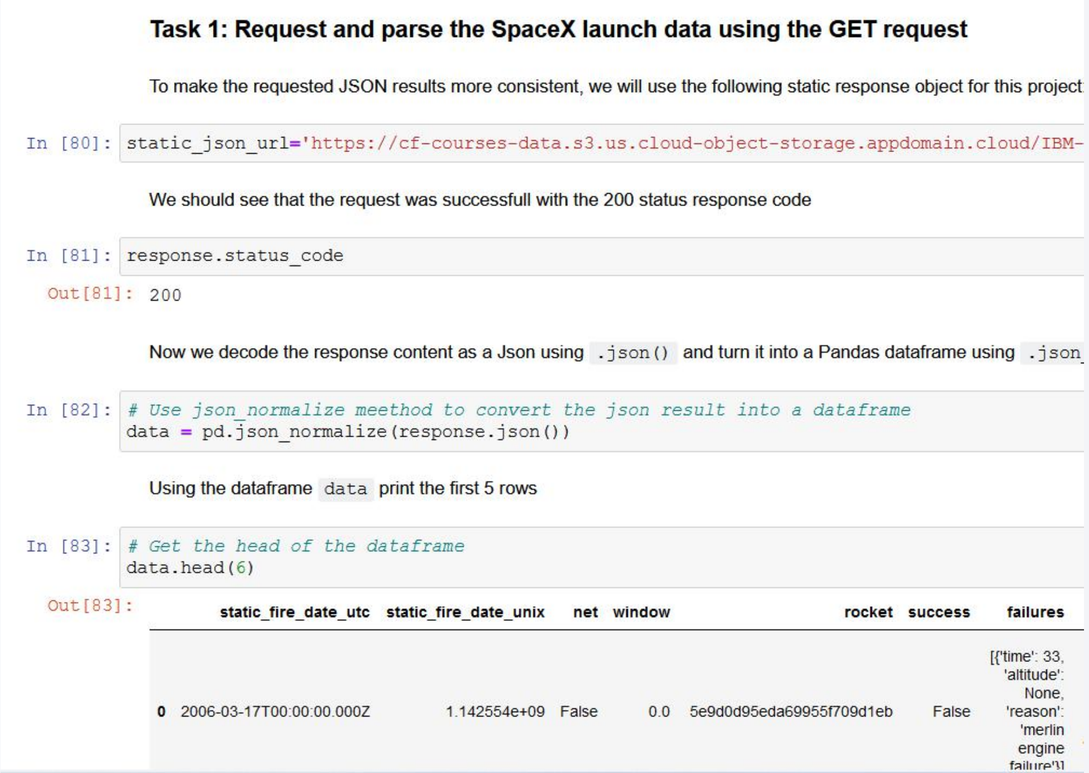
- I used Requests library that allows to make HTTP requests which I later used to get data from an API
- Then I requested json results using json_normalize meethod to convert the json result into a dataframe.
- Took a sunset of dataframe to keep only interesting features
- Then I construct a dataset using getBoosterVersion function.
- Cleaned the data, checked for missing values and fill in where necessary.
- Performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
- HTML table was converted to dataframe using pandas.

Data Collection – SpaceX API

I used Requests library that allows to make HTTP requests which I later used to get data from an API

Data wrangling, cleaning, fulfill the missing values.

(https://github.com/maryiamk/capstoneProject/blob/master/Data%20Collection%20API.ipynb)



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project

```
In [80]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-
```

We should see that the request was successfull with the 200 status response code

```
In [81]: response.status_code
Out[81]: 200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_`

```
In [82]: # Use json_normalize meethod to convert the json result into a dataframe
         data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
In [83]: # Get the head of the dataframe
         data.head(6)
Out[83]:
```

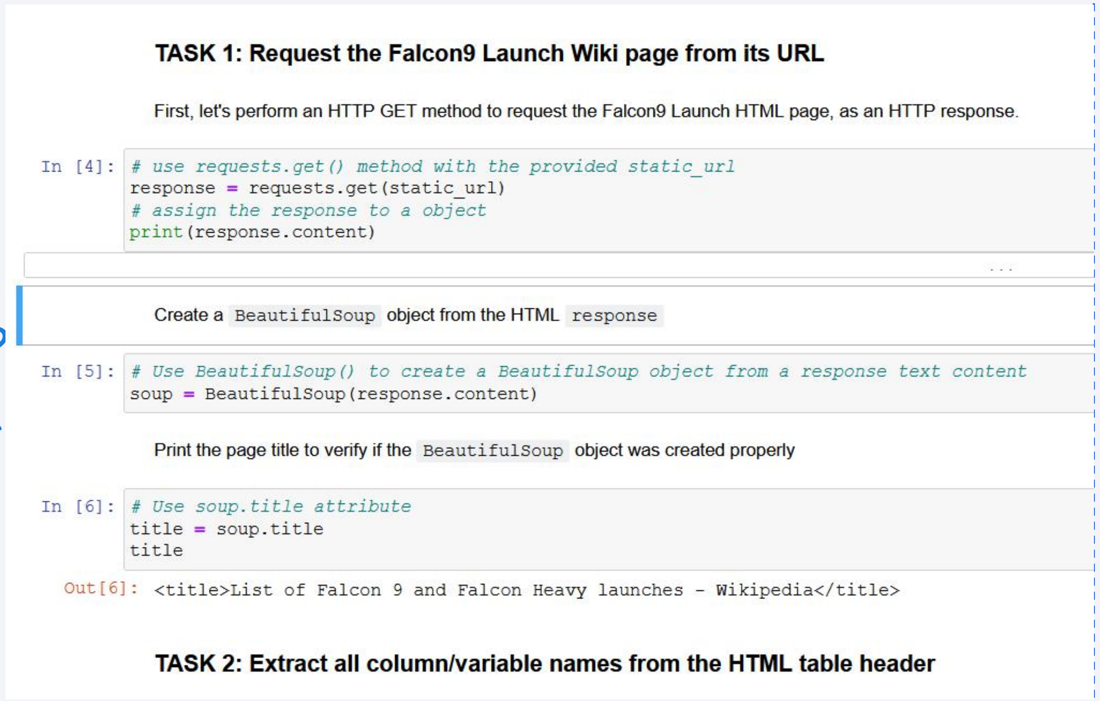
	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]

Data Collection – Scraping

- Request the Falcon9 Launch Wiki page from its URL using BeautifulSoup, Extract all column/variable names, Create a data frame by parsing the launch HTML tables

https://github.com/maryiamk/capstoneProject/blob/master/Data%20Collection%20with%20Web%20Scrapping.ipynb

P
h



TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
In [4]: # use requests.get() method with the provided static_url
        response = requests.get(static_url)
        # assign the response to a object
        print(response.content)
```

```
...
```

Create a `BeautifulSoup` object from the HTML `response`

```
In [5]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(response.content)
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [6]: # Use soup.title attribute
        title = soup.title
        title
```

```
Out[6]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Data Wrangling

- Calculated the number of launches on each site – value_counts()
- Calculated the number and occurrence of each orbit – value_counts()
- Created a landing outcome label "Class" from Outcome column

https://github.com/maryiamk/capstoneProject/blob/master/Data%20Wrangling.ipynb

EDA with Data Visualization

- Scatter plot was used to visualize the relationship between Flight Number and Launch Site (Orbit type as well), and between Payload and Launch Site (Orbit type as well)
- Horizontal Bar chart was used to visualize the relationship between success rate of each orbit type
- Line plot helped me visualize the launch success yearly trend

https://github.com/maryiamk/capstoneProject/blob/master/EDA%20with%20Visualization.ipynb

EDA with SQL

Here are some quesries I performed:

- %sql select … from …
- select … count(…) as count from .. group by …
- select * from … where … like ucase(…) limit(…)
- select min/max/avg() …
- select … from … where … = (select max(…) from …)

https://github.com/maryiamk/capstoneProject/blob/master/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- First I put all location of the launching sites.
- I have added such map objects as markers, circles, lines, etc.
- Circles – to add a highlighted circle area with a text label on a specific coordinate;
- Marker Cluster for successful and unsuccesssful launch
- Lines to calculate the distance to the sea shore/highway and other objects

https://github.com/maryiamk/capstoneProject/blob/master/Data%20Visualization%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboard consisted of the pie chart showing the total launches from a certain site
- Scatter graph showed the relationship between success launch outcome and payload mass for the different booster version.

https://github.com/maryiamk/capstoneProject/blob/master/Dashboard%20with%20Plotly.ipynb

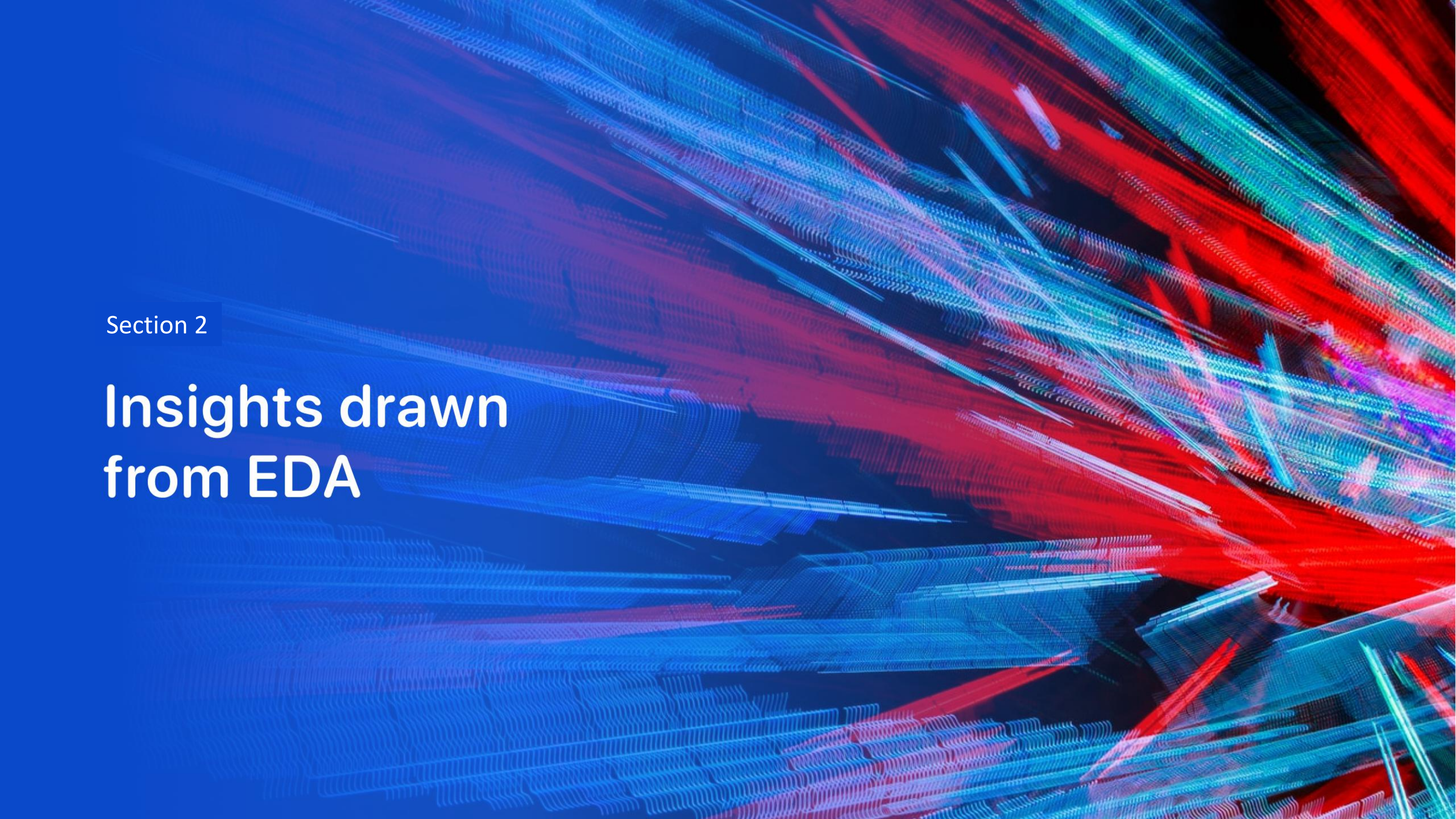
Predictive Analysis (Classification)

- I transformed the data, split it into training and testing sets.
- I built four machine learning models: KNN, Decision Tree, SVM and Logistic Regression, and used different hyperparameters with GridSearchCV to determine the attributes with the best accuracy.
- Models accuracy using score method was used to determine the best performing model.
- Three out of four classification models showed the same result - 0.83% of accuracy in predicting results on testing set.

https://github.com/maryiamk/capstoneProject/blob/master/Machine%20Learning%20Prediction.ipynb

Results

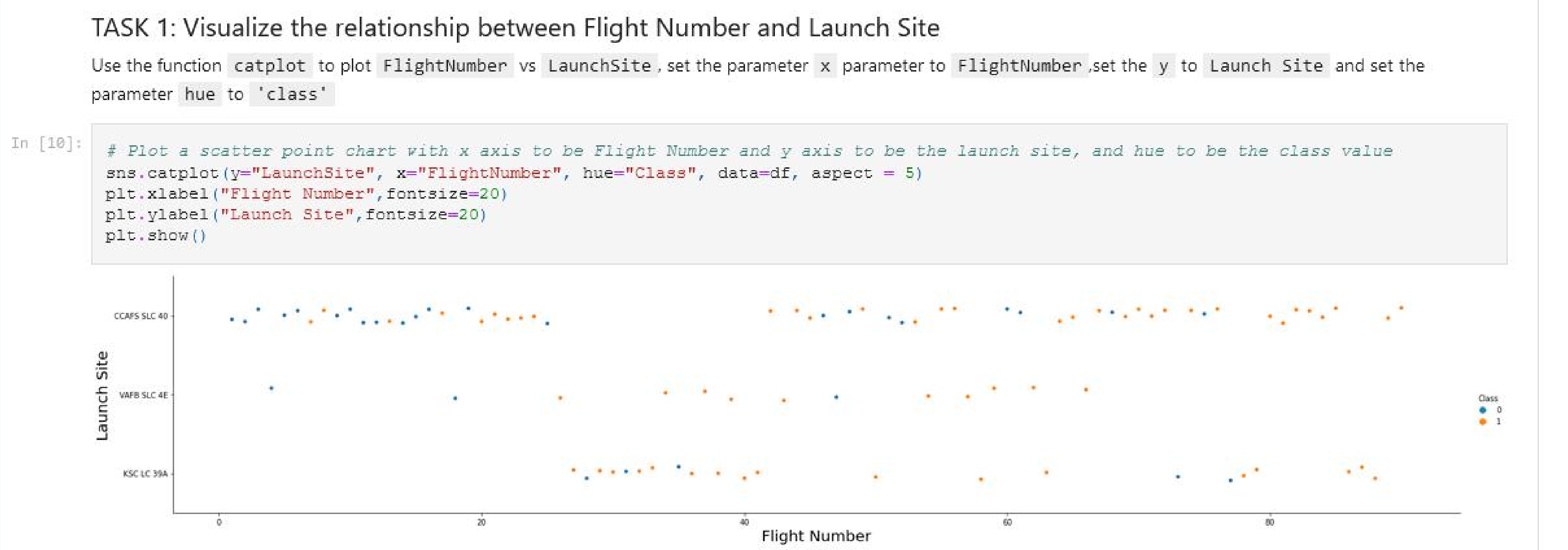
- Three out of four classification models showed the same result – 0.83% of accuracy in predicting results on testing set. Only Decision Tree model showed 0.78% of accuracy.



Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



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

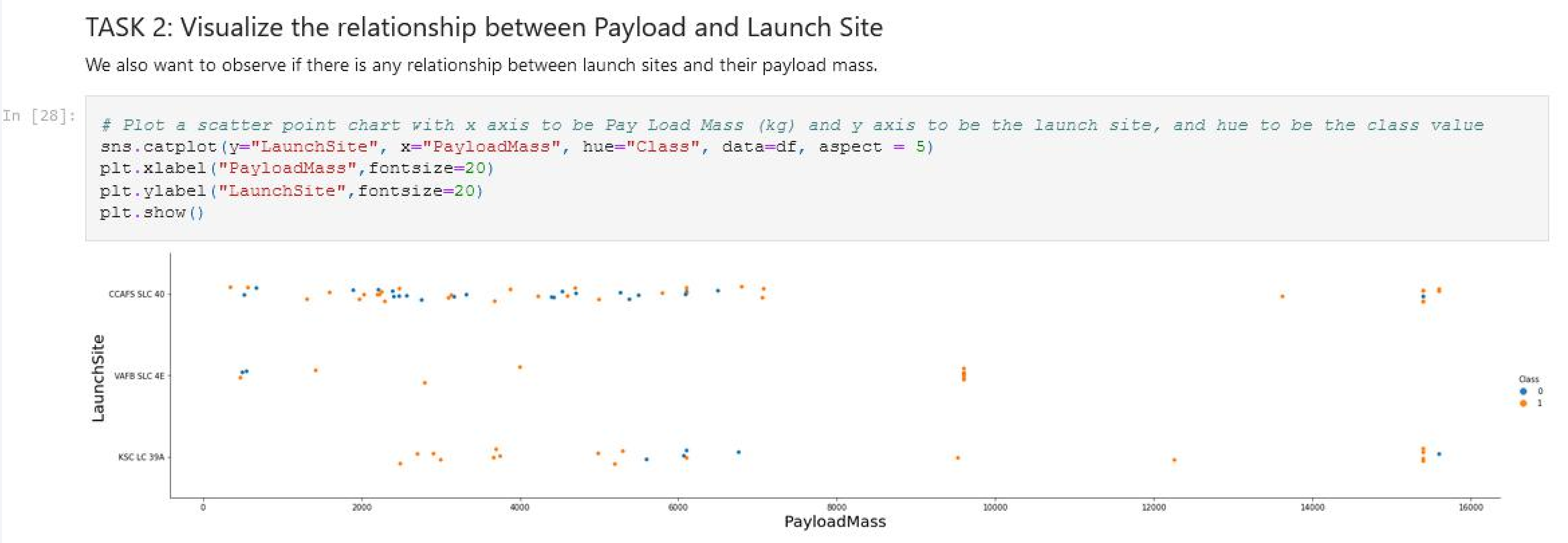
Use the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`,set the `y` to `Launch Site` and set the parameter `hue` to `'class'`

In [10]:

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

The larger the flight number at a launch site, the greater the success rate.

Payload vs. Launch Site



TASK 2: Visualize the relationship between Payload and Launch Site

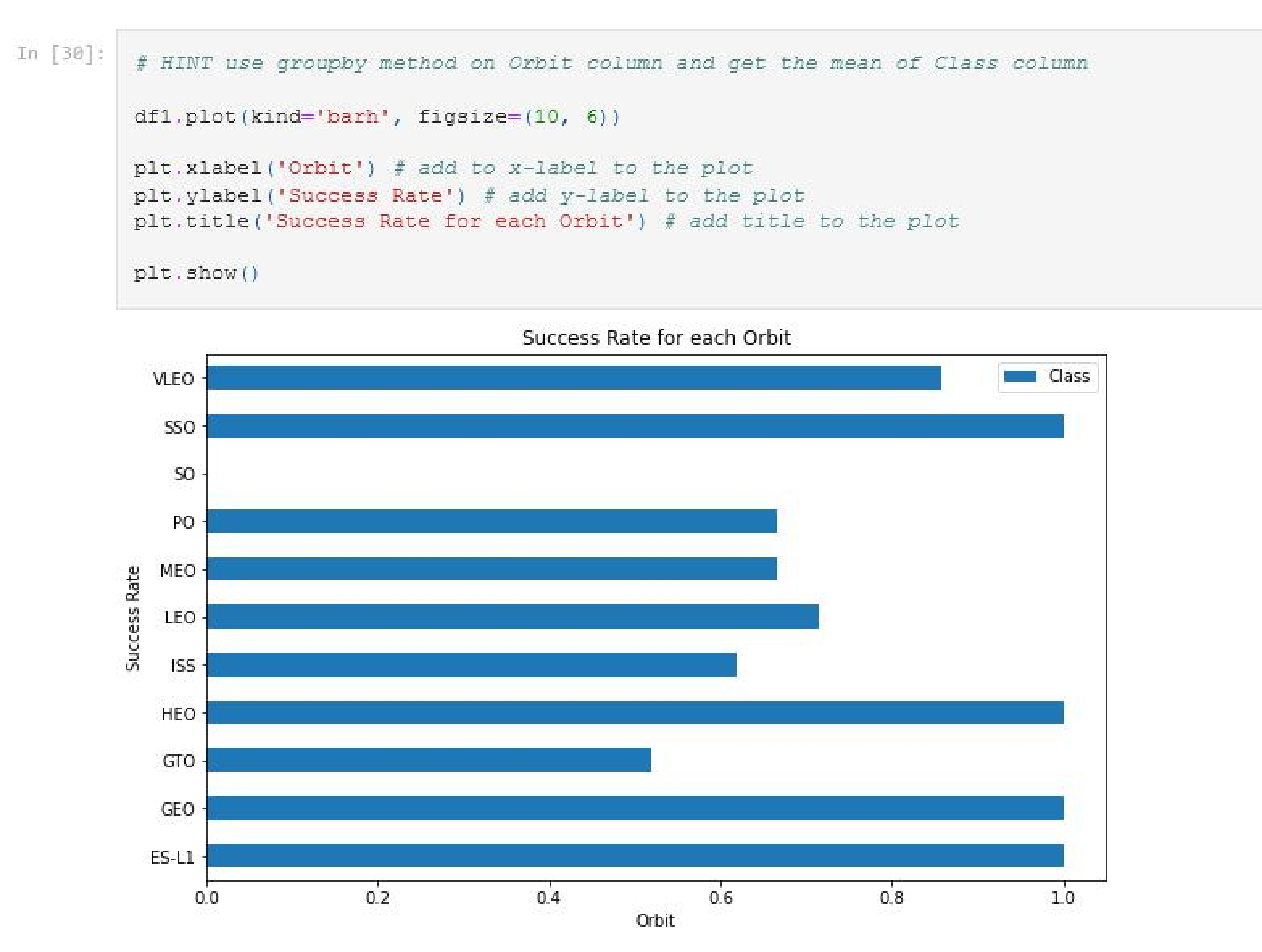
We also want to observe if there is any relationship between launch sites and their payload mass.

In [28]:

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```

We can see no rockets launched for heavypayload mass(greater than 10000)

Success Rate vs. Orbit Type

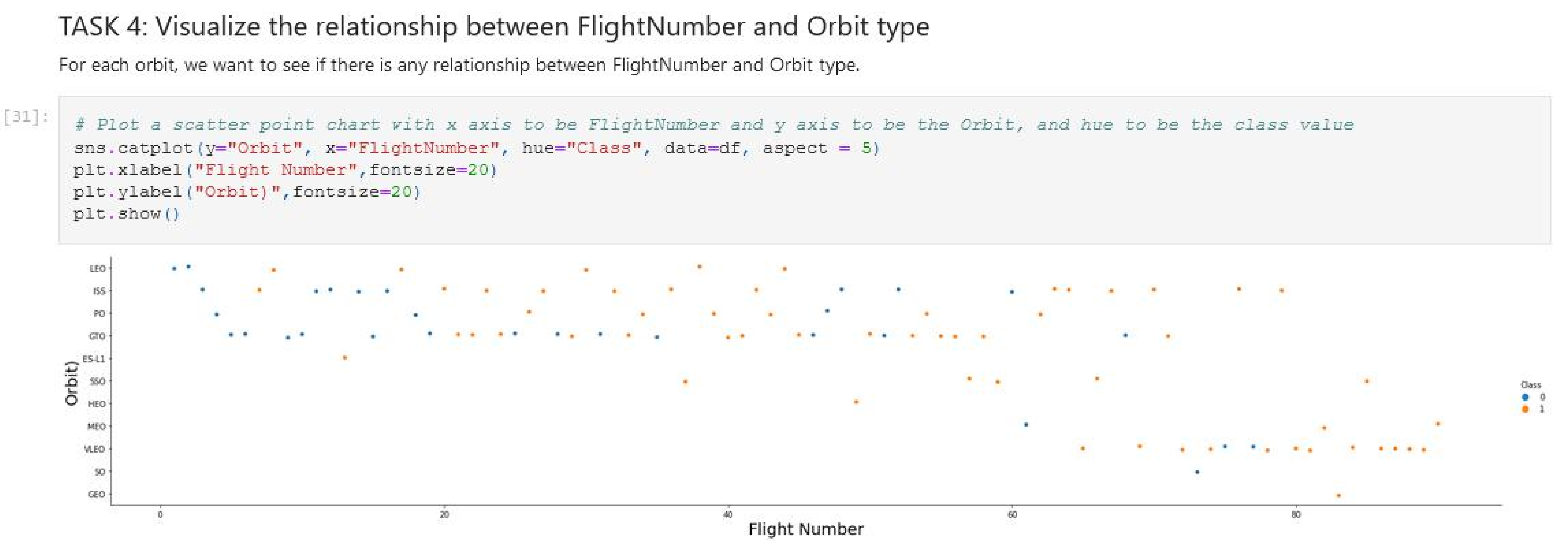


In [30]:

```
# HINT use groupby method on Orbit column and get the mean of Class column

df1.plot(kind='barh', figsize=(10, 6))

plt.xlabel('Orbit') # add to x-label to the plot
plt.ylabel('Success Rate') # add y-label to the plot
plt.title('Success Rate for each Orbit') # add title to the plot

plt.show()
```

SSO, HEO, GEO and ES-L1 have the highest success rate among all orbits.

Flight Number vs. Orbit Type



TASK 4: Visualize the relationship between FlightNumber and Orbit type

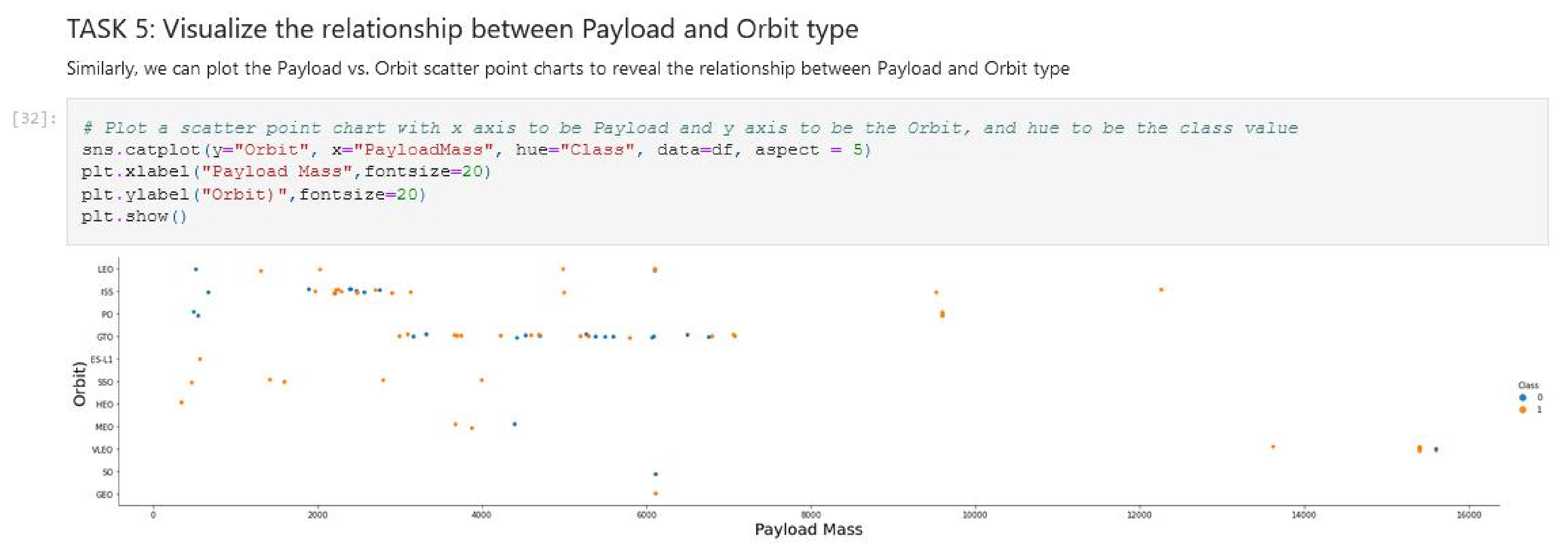
For each orbit, we want to see if there is any relationship between FlightNumber and Orbit type.

[31]:

```
# 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.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit)",fontsize=20)
plt.show()
```

In the LEO orbit the Success appears related to the number of flights

Payload vs. Orbit Type



TASK 5: Visualize the relationship between Payload and Orbit type

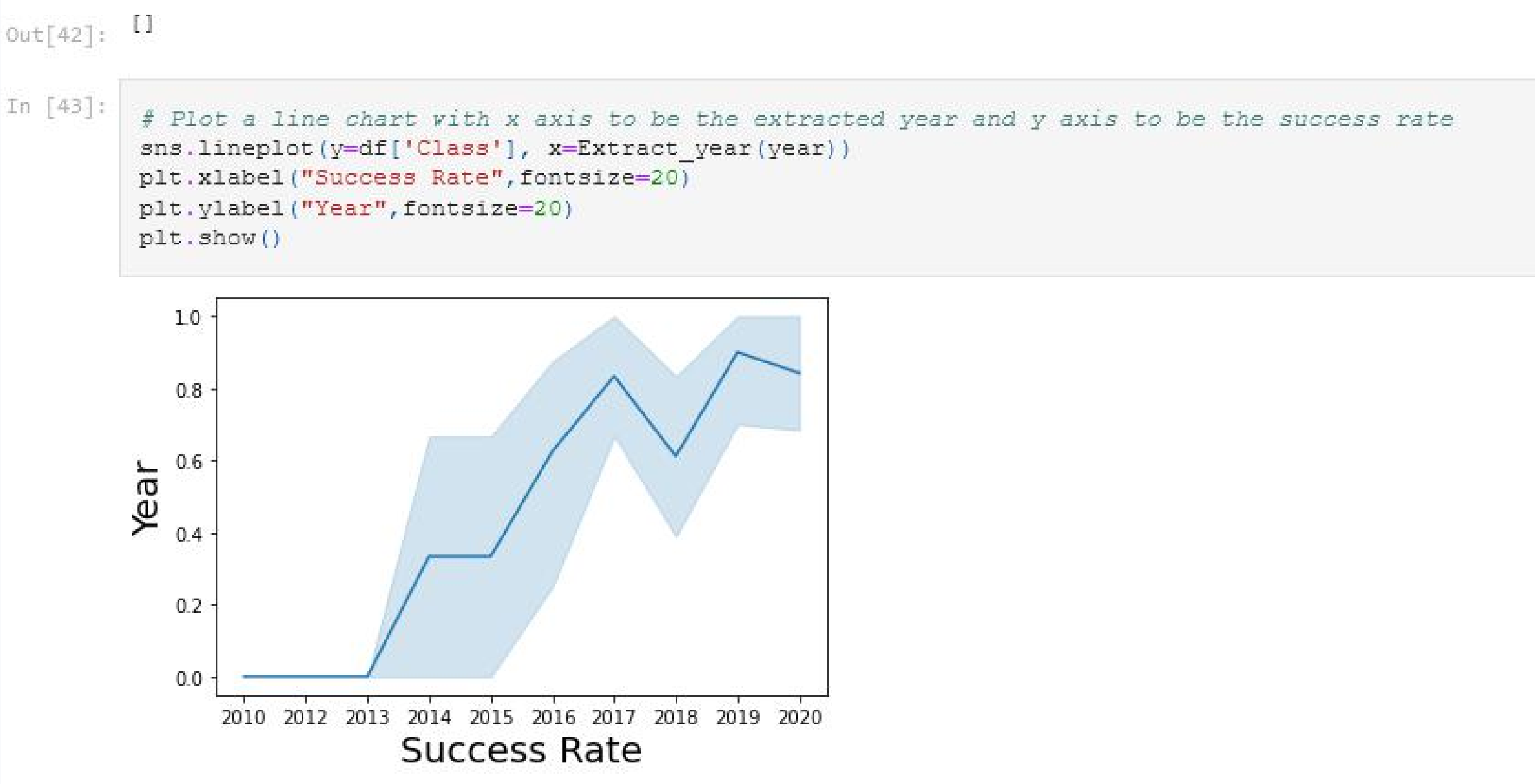
Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

[32]:

```
# 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 = 5)
plt.xlabel("Payload Mass",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.

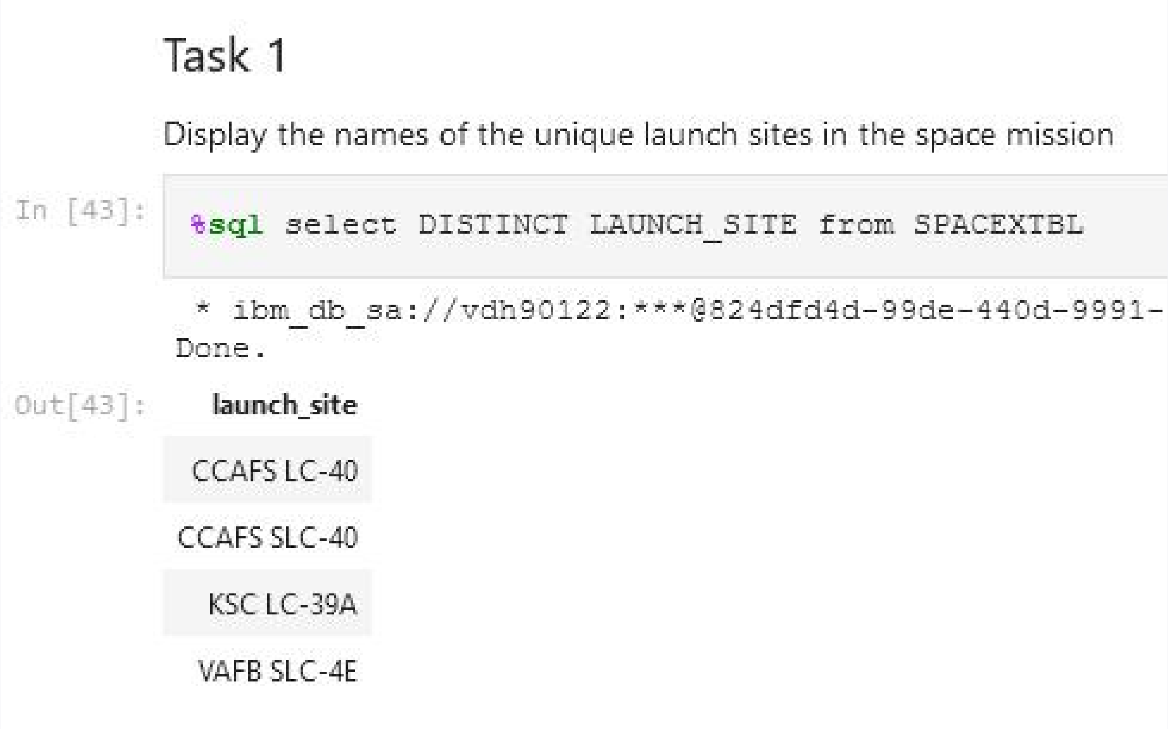
Launch Success Yearly Trend



```
Out[42]: []

In [43]:
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
sns.lineplot(y=df['Class'], x=Extract_year(year))
plt.xlabel("Success Rate",fontsize=20)
plt.ylabel("Year",fontsize=20)
plt.show()
```

The sucess rate since 2013 kept increasing till 2020.

All Launch Site Names



Task 1

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

```
In [43]: %sql select DISTINCT LAUNCH_SITE from SPACEXTBL
```

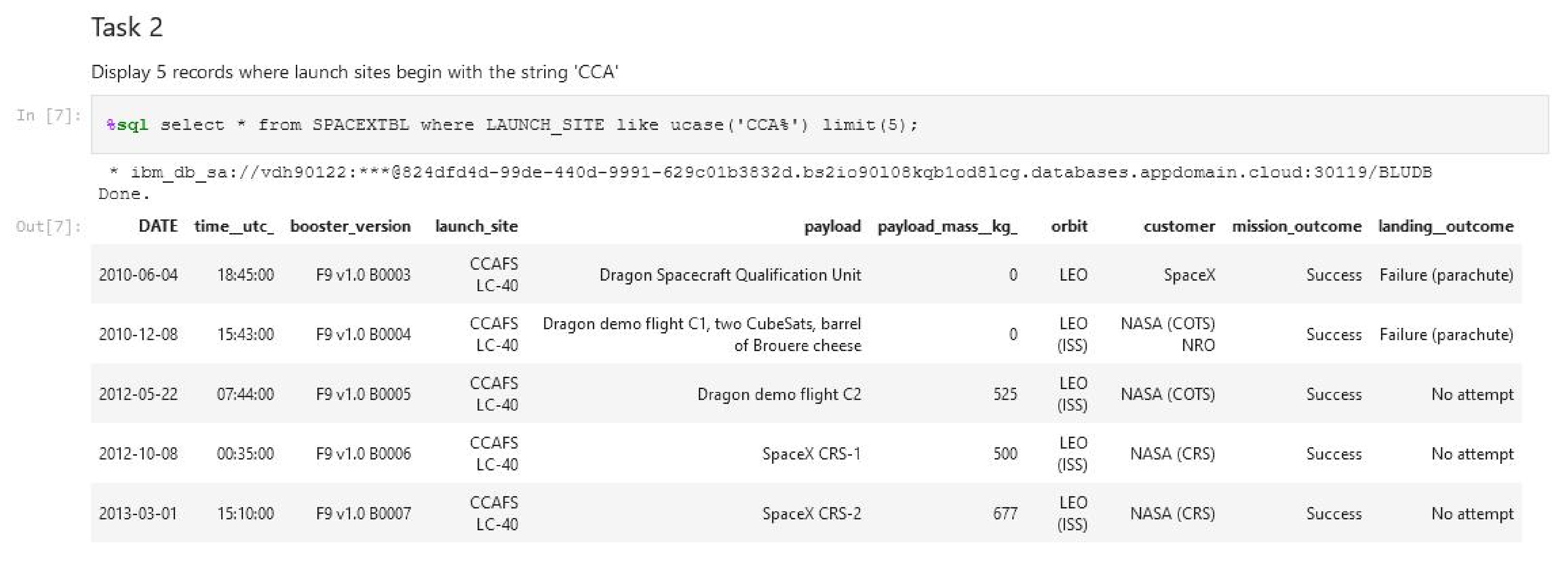
```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-
Done.
```

Out[43]:

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

There are four unique launch sites.

Launch Site Names Begin with 'CCA'



Task 2

Display 5 records where launch sites begin with the string 'CCA'

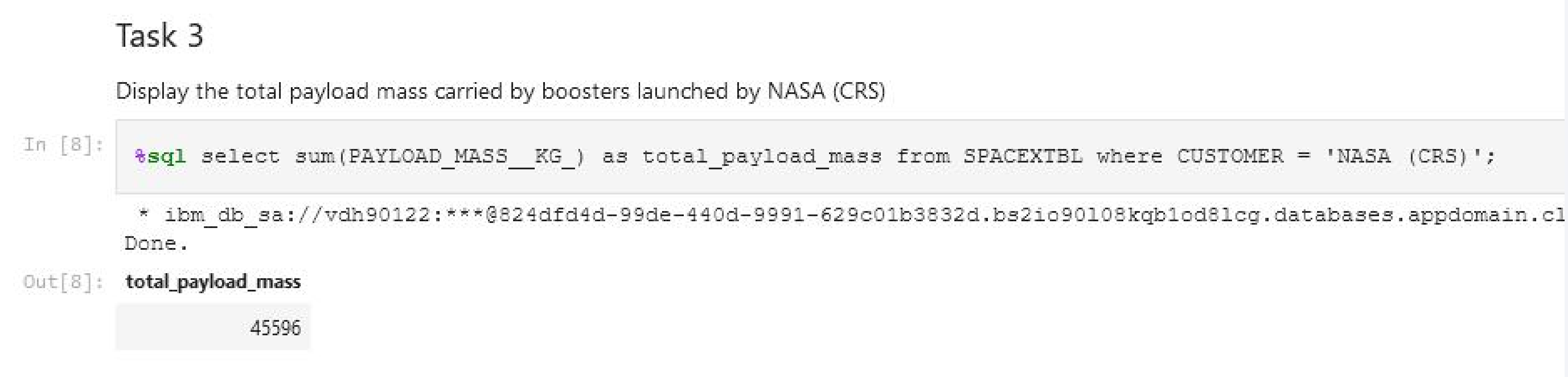
```
In [7]: %sql select * from SPACEXTBL where LAUNCH_SITE like ucase('CCA%') limit(5);
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[7]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass



Task 3

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

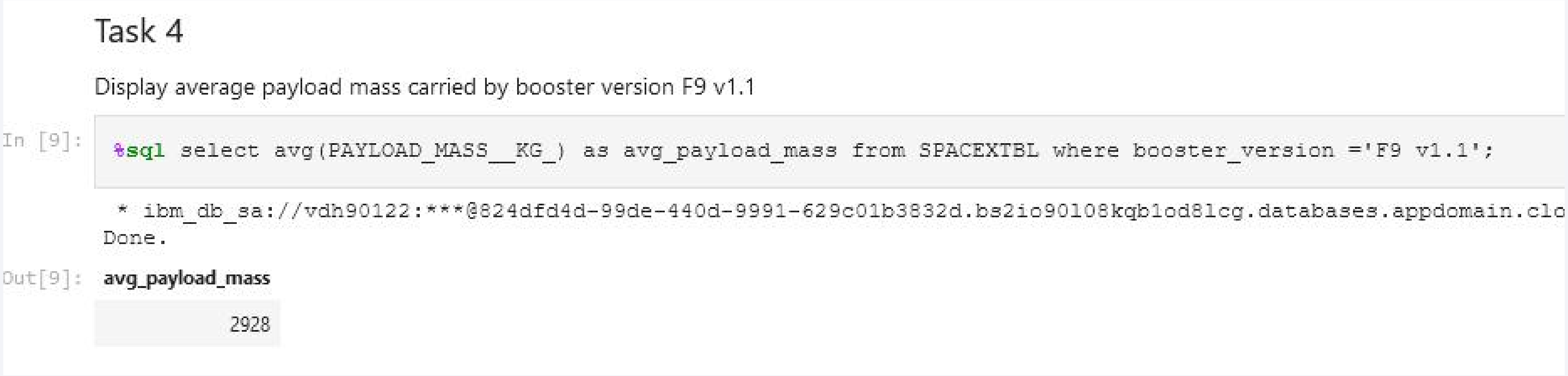
```
In [8]:  %sql select sum(PAYLOAD_MASS__KG_) as total_payload_mass from SPACEXTBL where CUSTOMER = 'NASA (CRS)';
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cl
Done.
```

Out[8]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1



Task 4

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

In [9]:

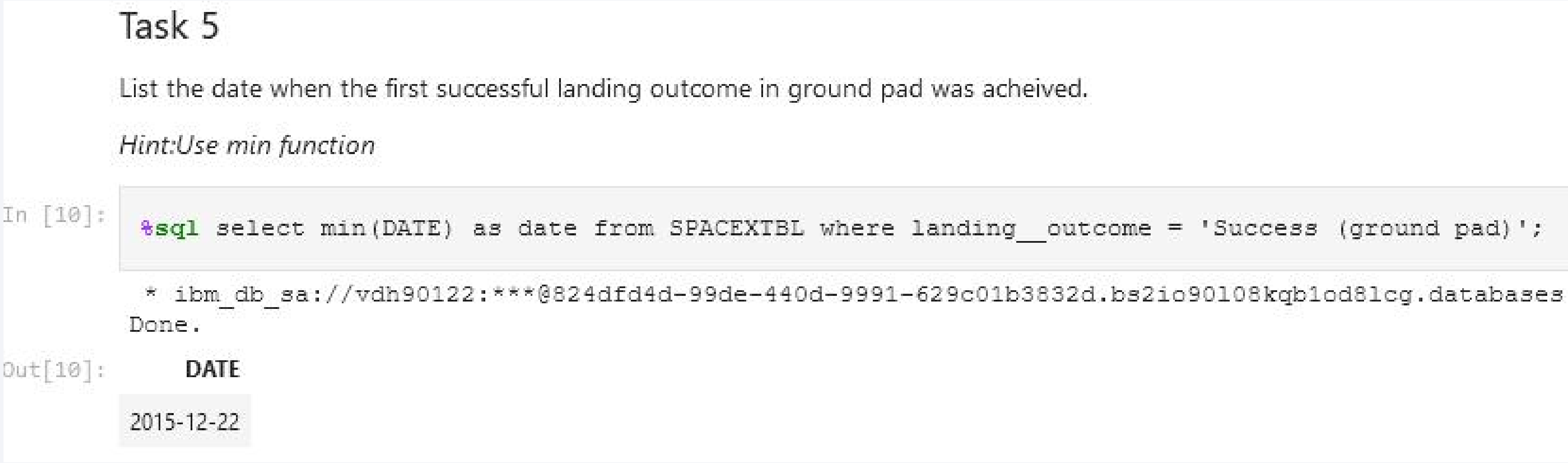
```
%sql select avg(PAYLOAD_MASS__KG_) as avg_payload_mass from SPACEXTBL where booster_version ='F9 v1.1';
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.clo
Done.
```

Out[9]:

avg_payload_mass
2928

First Successful Ground Landing Date



Task 5

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

Hint:Use min function

In [10]:

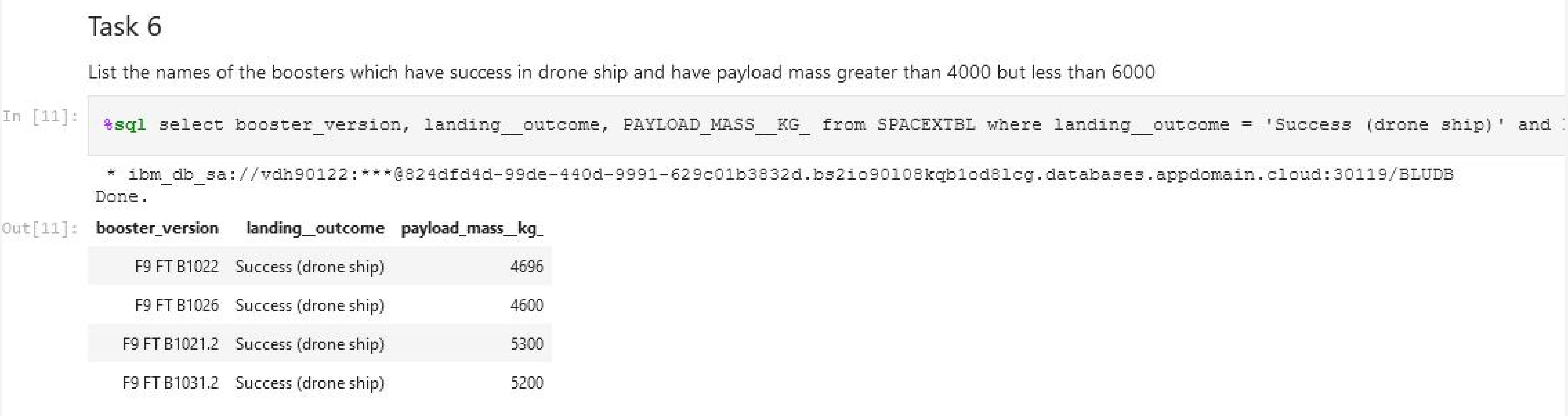
```
%sql select min(DATE) as date from SPACEXTBL where landing__outcome = 'Success (ground pad)';
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90108kqb1od8lcg.databases
Done.
```

Out[10]:

DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

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

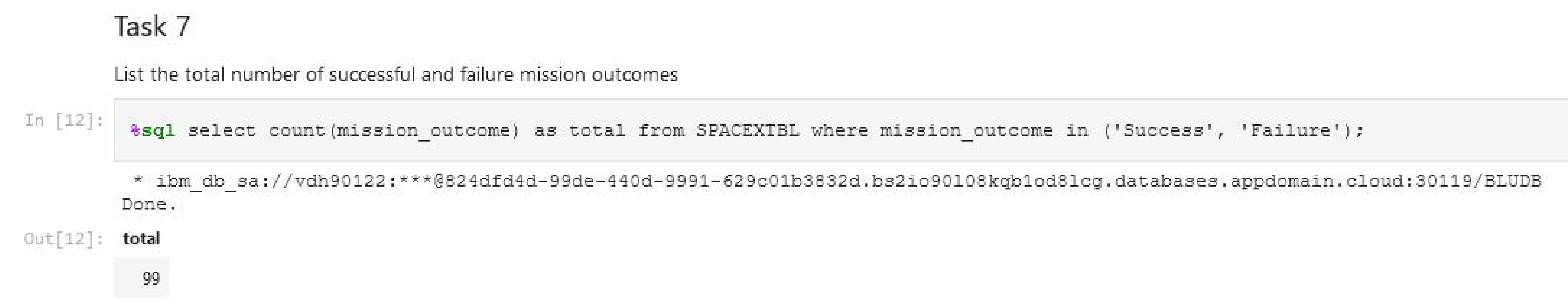
```
In [11]: %sql select booster_version, landing__outcome, PAYLOAD_MASS__KG_ from SPACEXTBL where landing__outcome = 'Success (drone ship)' and 
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[11]:

booster_version	landing__outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

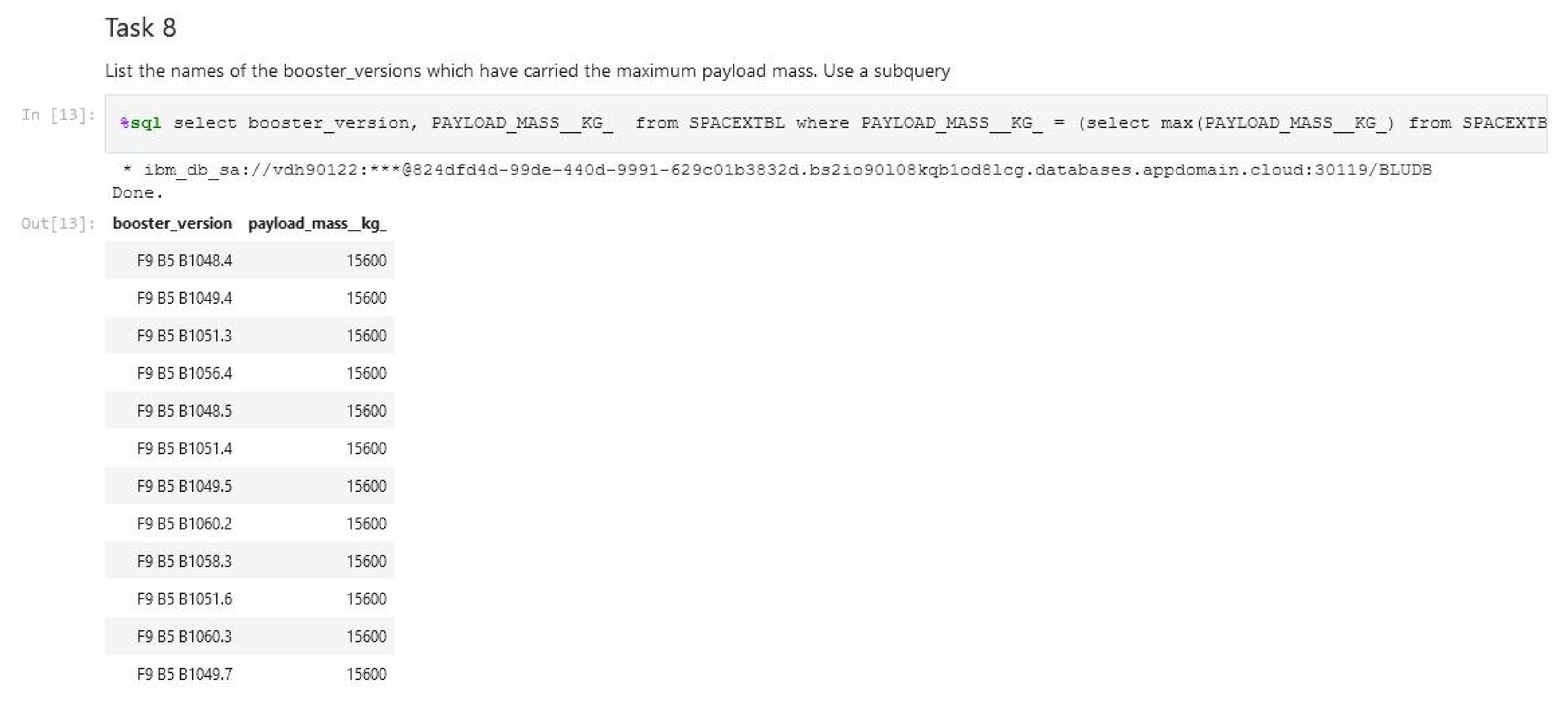
```
In [12]: %sql select count(mission_outcome) as total from SPACEXTBL where mission_outcome in ('Success', 'Failure');
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[12]:

total
99

Boosters Carried Maximum Payload



Task 8

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

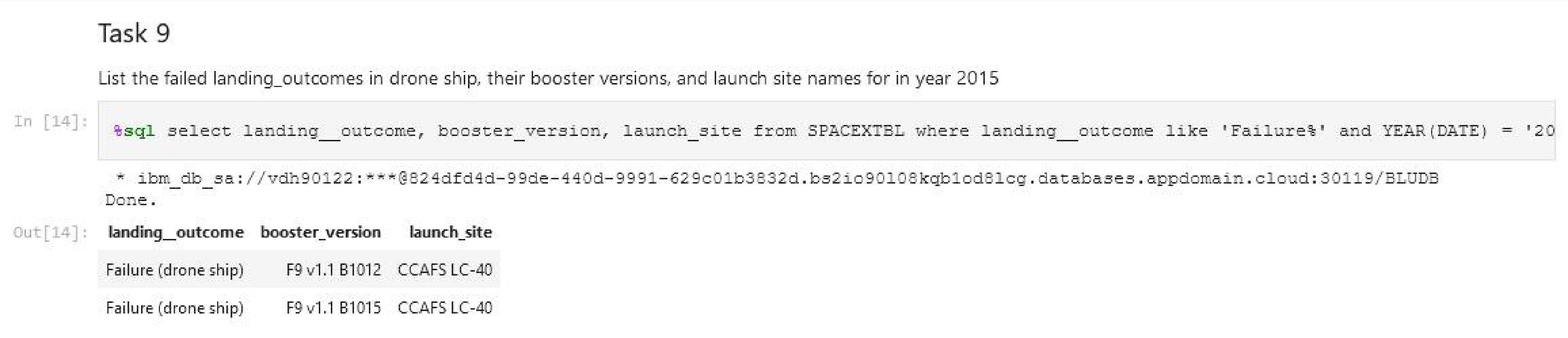
In [13]:
```
%sql select booster_version, PAYLOAD_MASS__KG_  from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTB
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[13]:

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records



Task 9

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

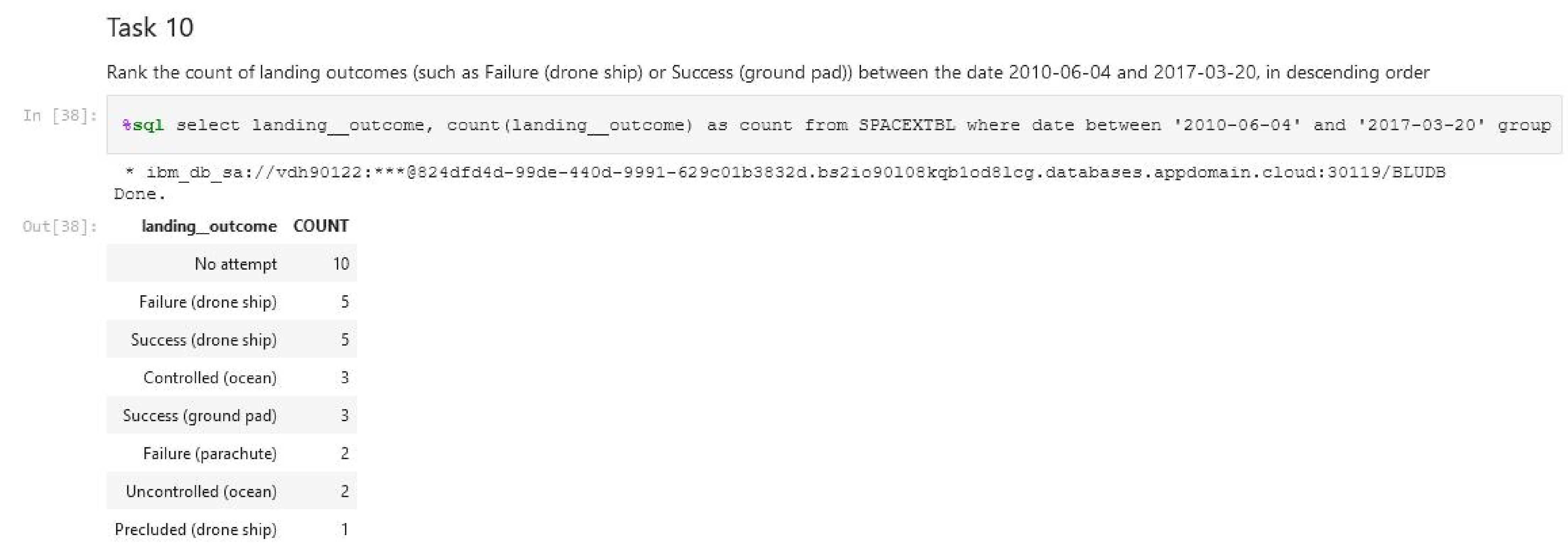
```
In [14]: %sql select landing__outcome, booster_version, launch_site from SPACEXTBL where landing__outcome like 'Failure%' and YEAR(DATE) = '20
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90108kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

Out[14]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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



Task 10

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

```
In [38]:  %sql select landing__outcome, count(landing__outcome) as count from SPACEXTBL where date between '2010-06-04' and '2017-03-20' group
```

```
 * ibm_db_sa://vdh90122:***@824dfd4d-99de-440d-9991-629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30119/BLUDB
Done.
```

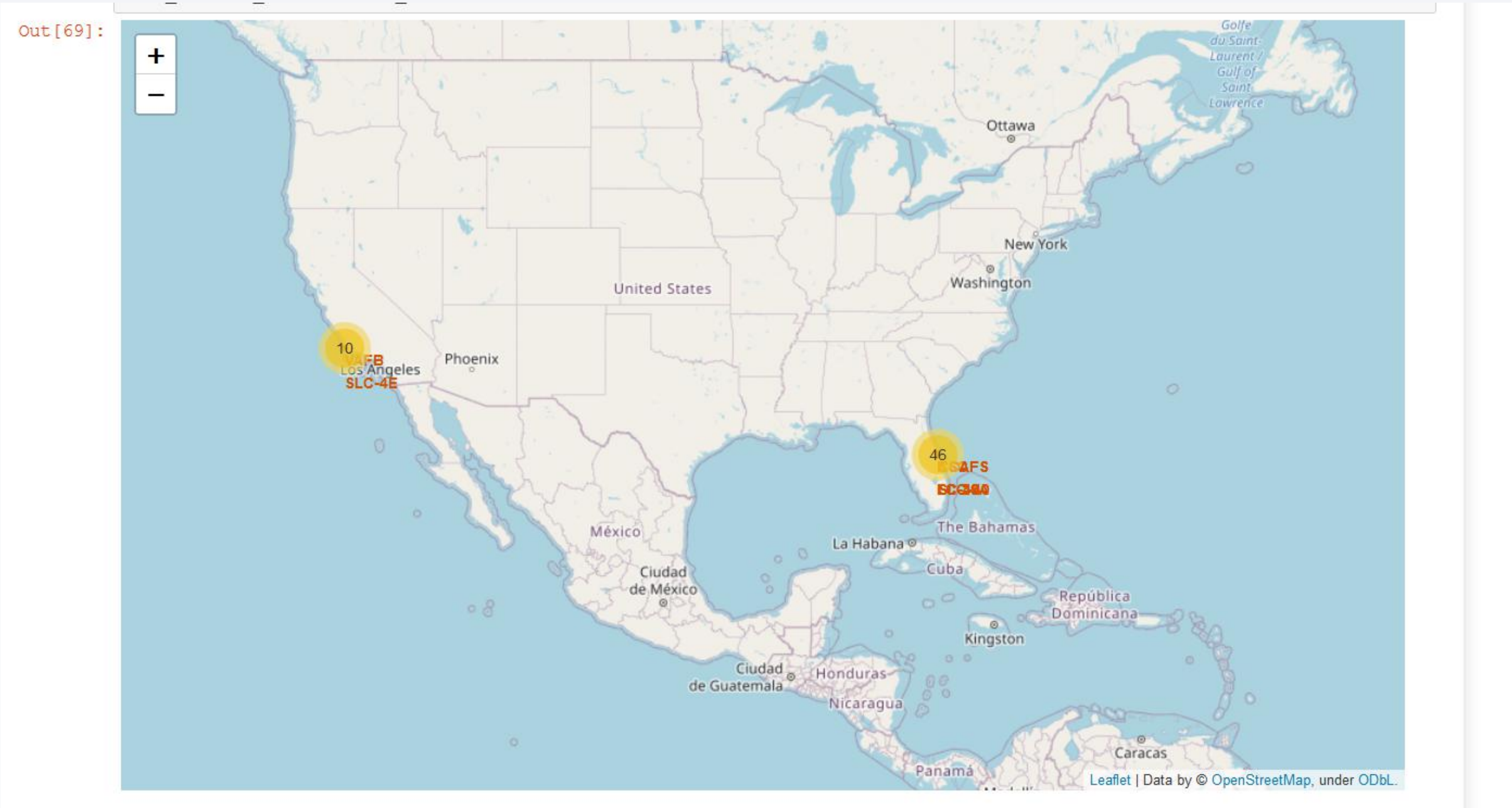
Out[38]:

landing__outcome	COUNT
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

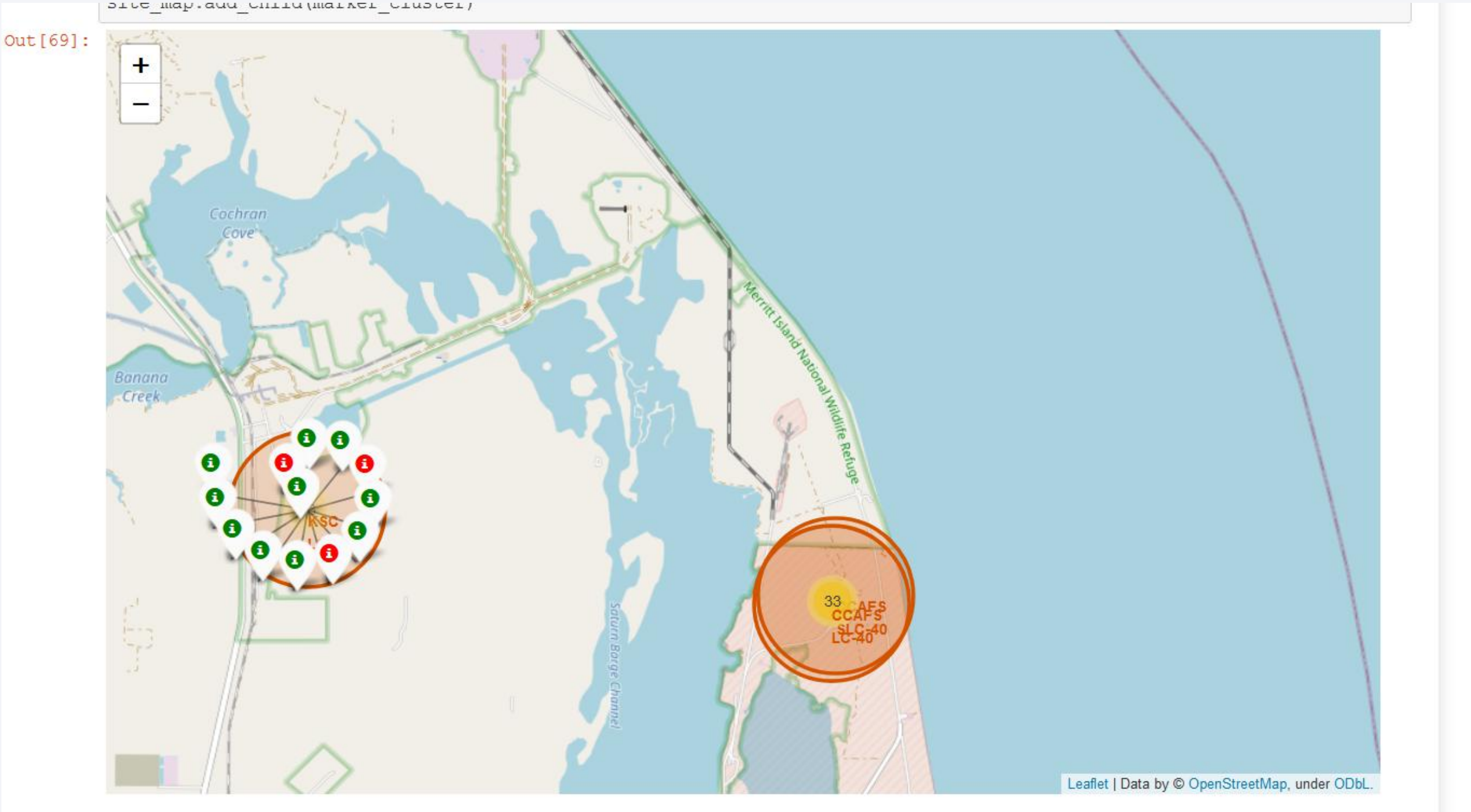
Launch sites - global map - North America



Out[69]:

Florida and California are two states where there are launch sites of Falcon 9.

Successful launch class markers



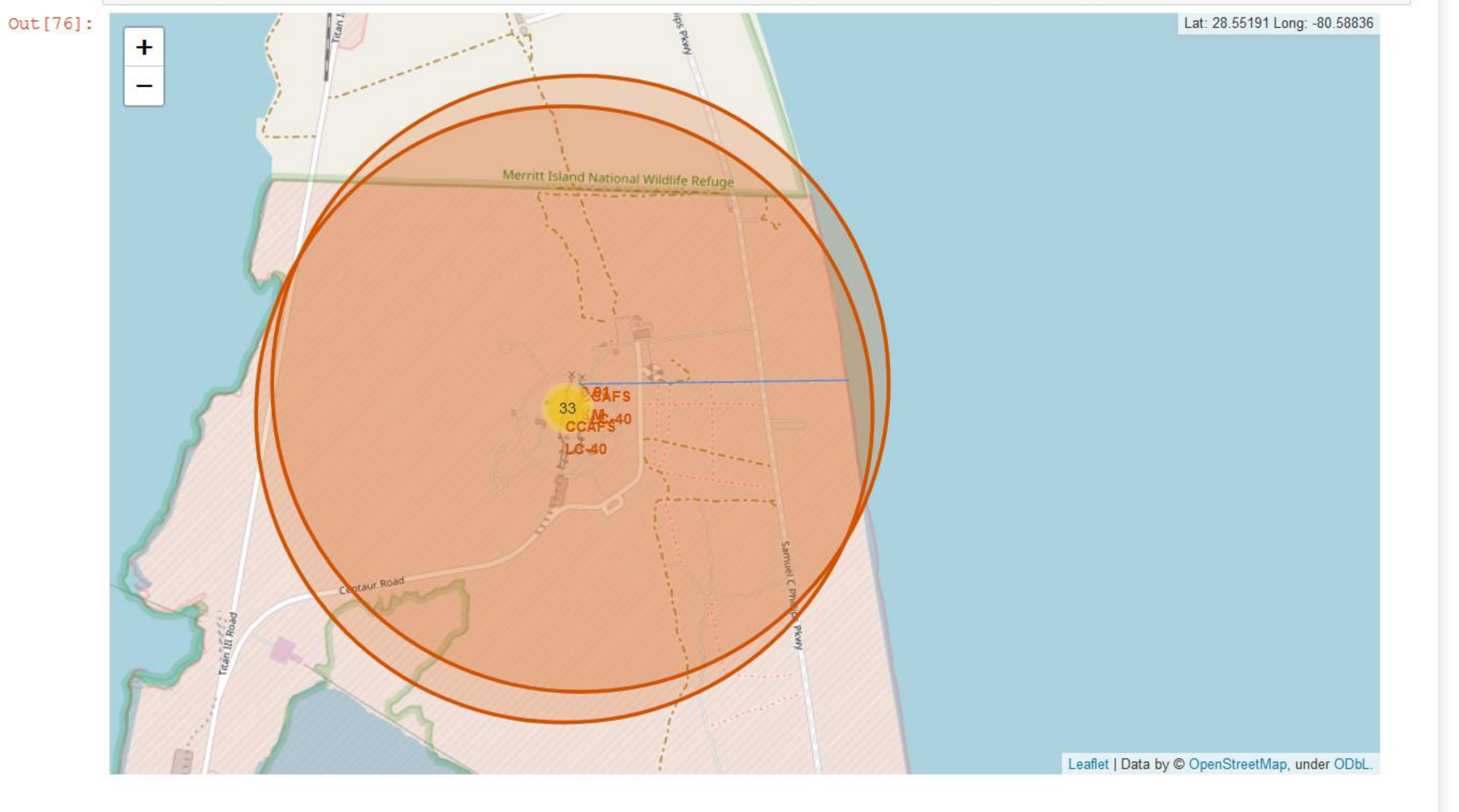
```
site_map.add_child(marker_cluster)
```

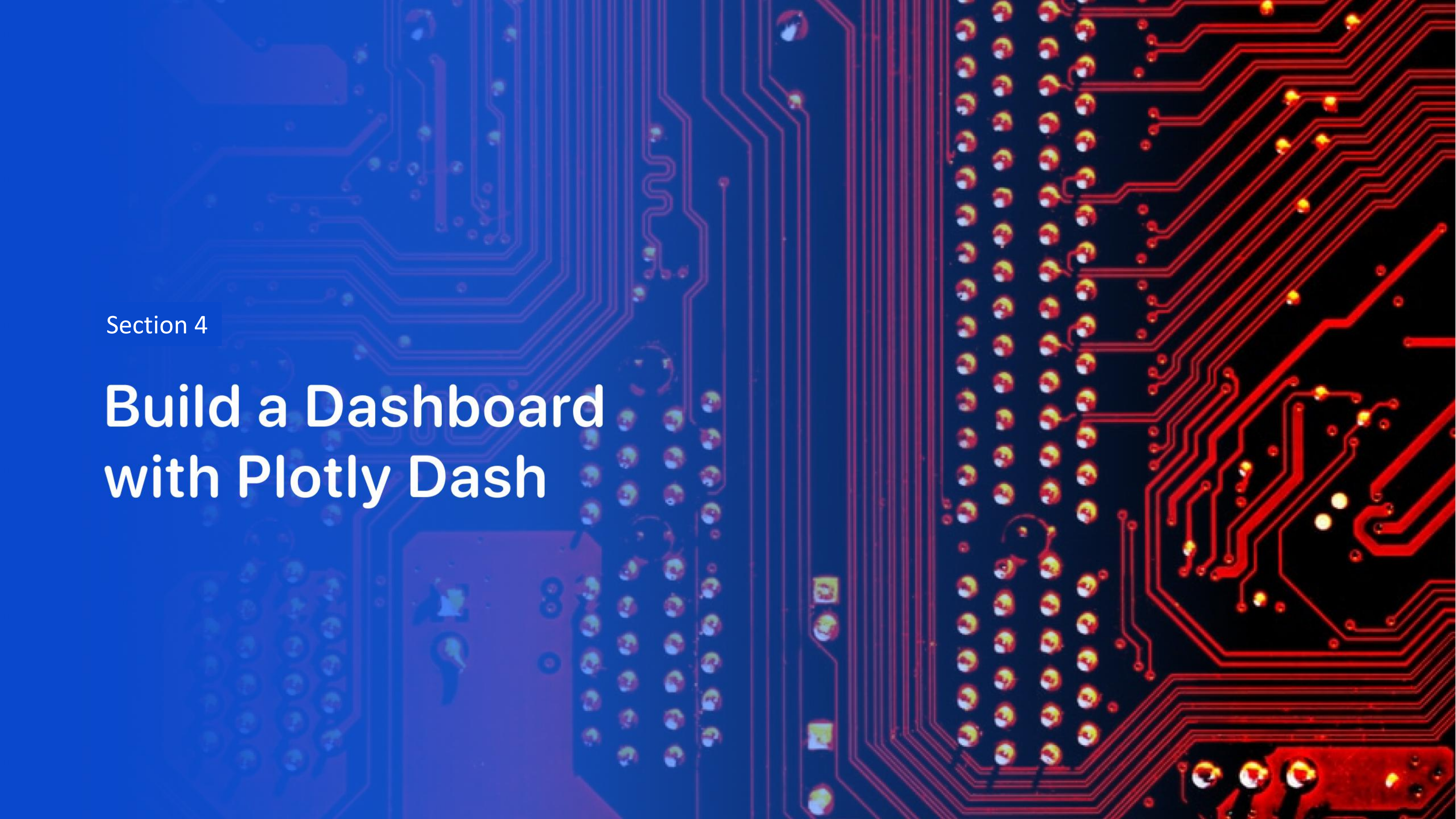
```
Out[69]:
```

+
−
Cochran Cove
Banana Creek
Merritt Island National Wildlife Refuge
Saturn Barge Channel
KSC
33
CCAFS
CCAFS SLC-40
LC-40
Leaflet | Data by © OpenStreetMap, under ODbL.

- Two closely related launch sites in Florida idicated by marker clusters which shows successful class of the launch. Green - success, red - failure.

It help us to understand if the site is in the close proximity to coastline

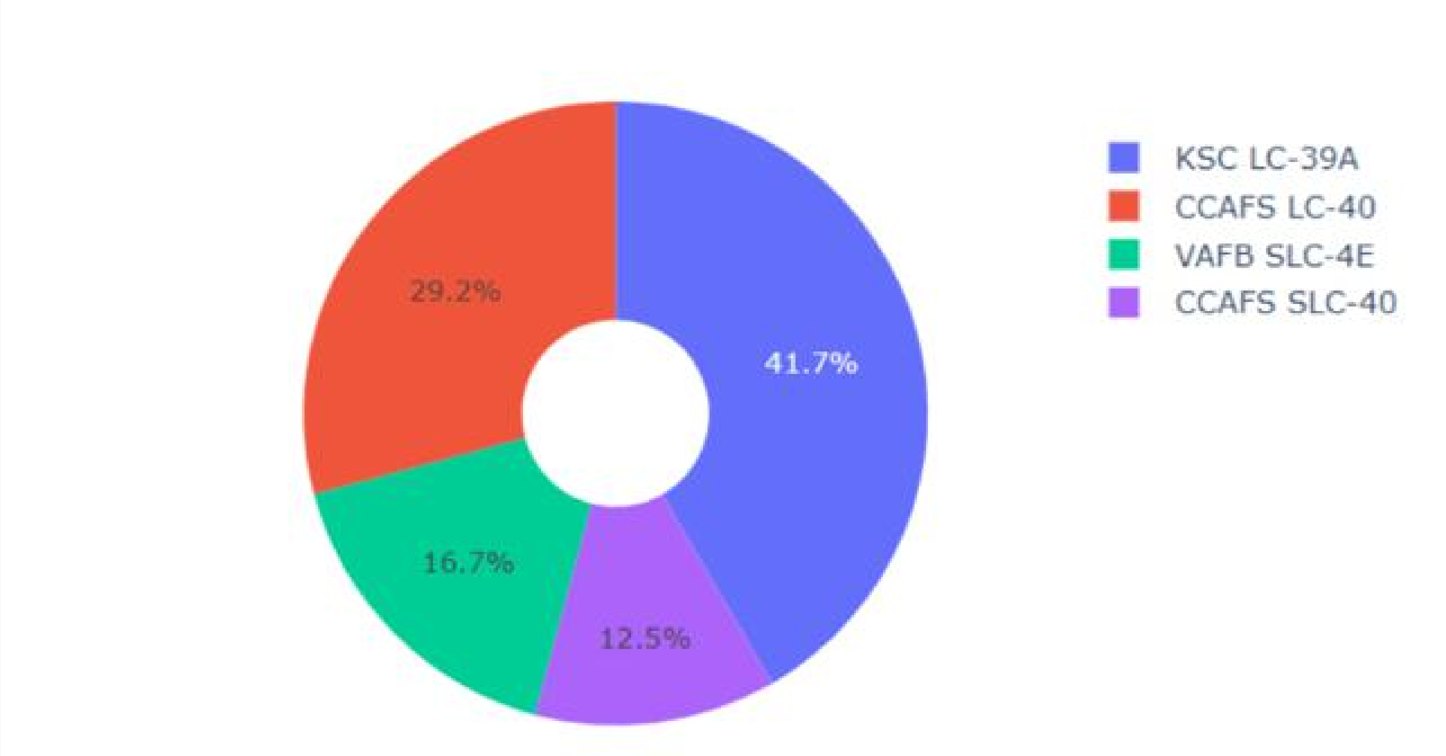

Out[76]:
+
−
Lat: 28.55191 Long: -80.58836
Merritt Island National Wildlife Refuge
33
CCAFS LC-40
CCAFS LC-40
Centaur Road
Titan III Road
Samuel C Phillips Pkwy
Leaflet | Data by © OpenStreetMap, under ODbL.



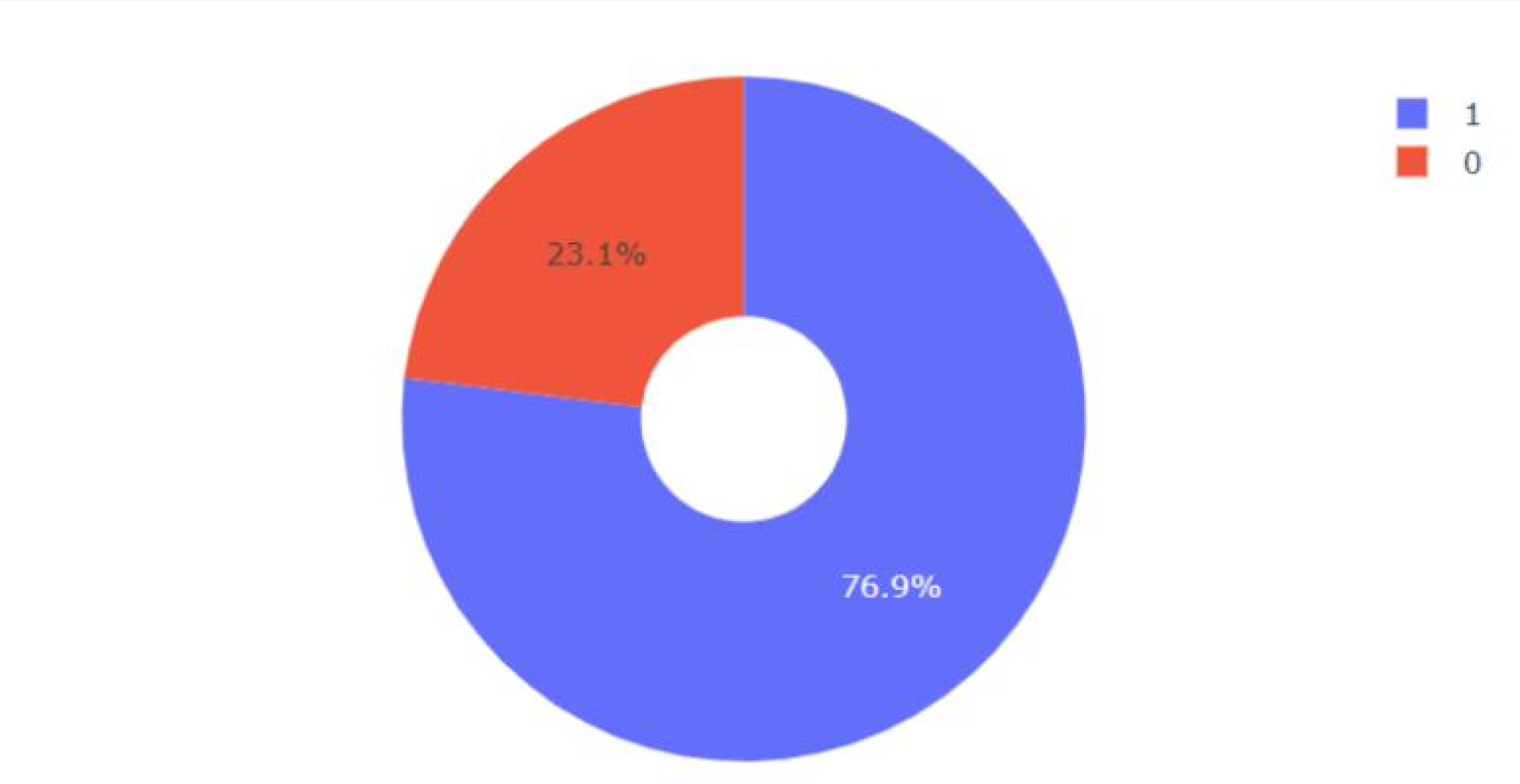
Section 4

Build a Dashboard with Plotly Dash

Pie chart of total success launches by the launch site

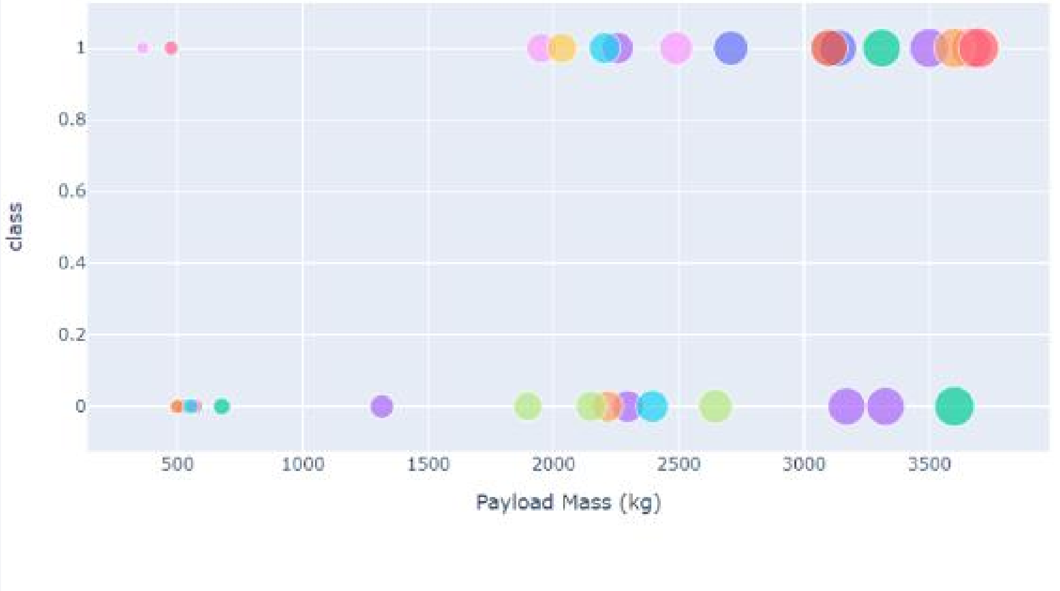


Success ratio of launches on the KSC LC 39A station

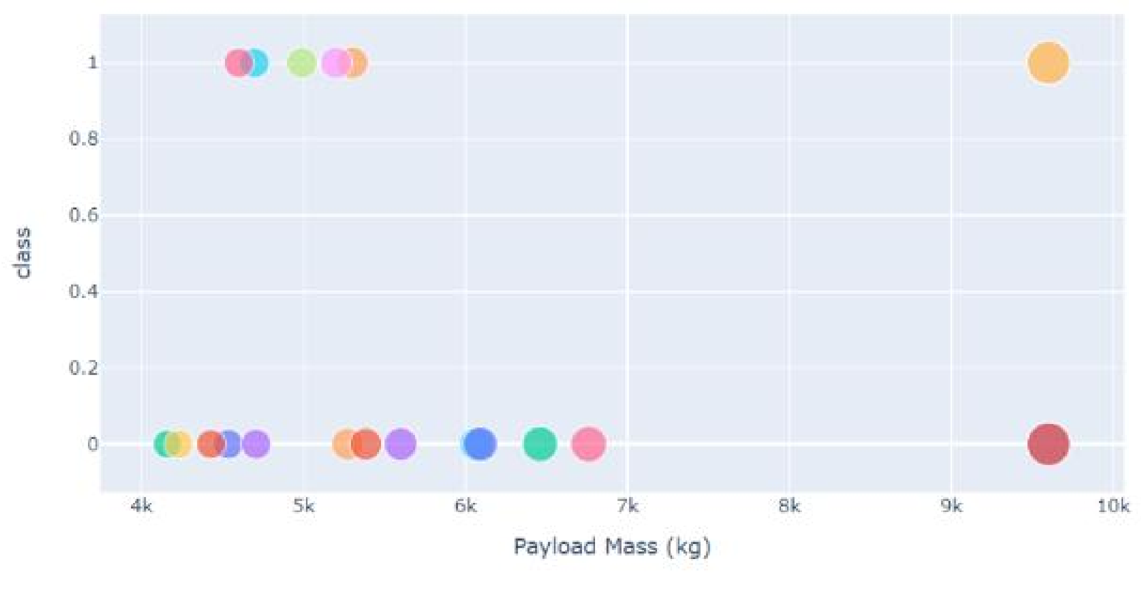


Payload Mass VS. Success lauching

Payload mass 0 - 4000 kg



Payload mass 4000 - 10000 kg

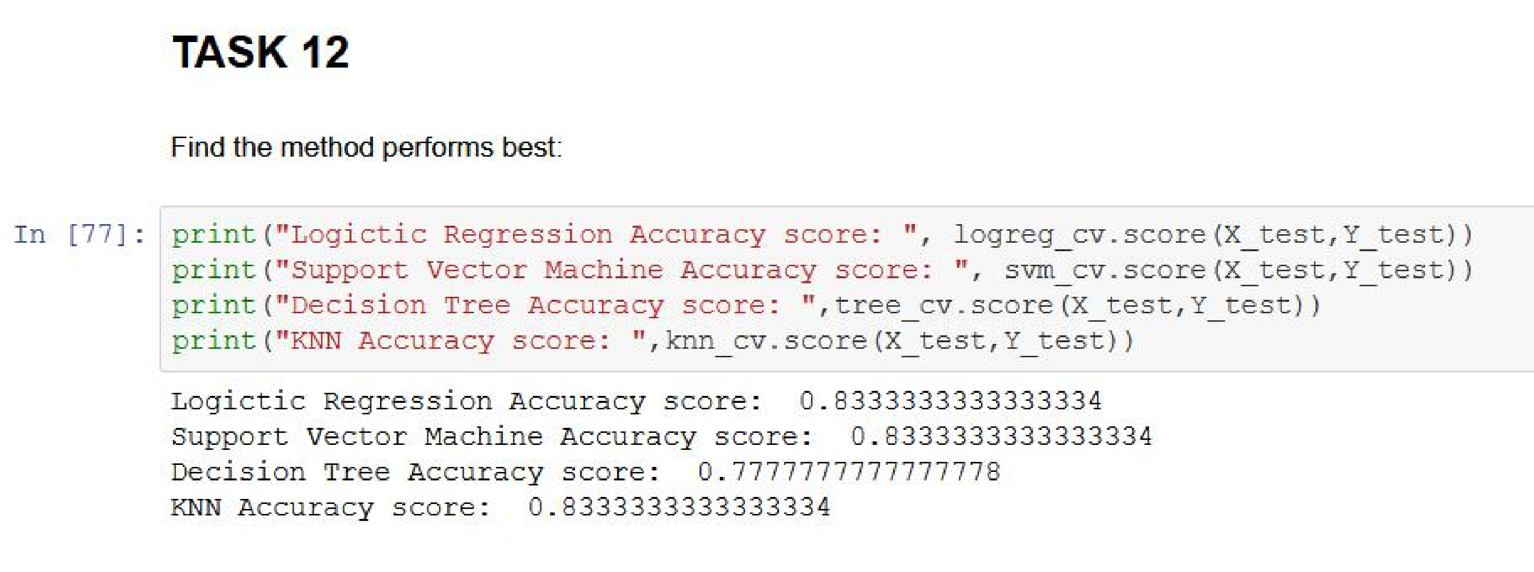


The success launching is higher if the payload mass is lower.

Section 5

Predictive Analysis (Classification)

Classification Accuracy



TASK 12

Find the method performs best:

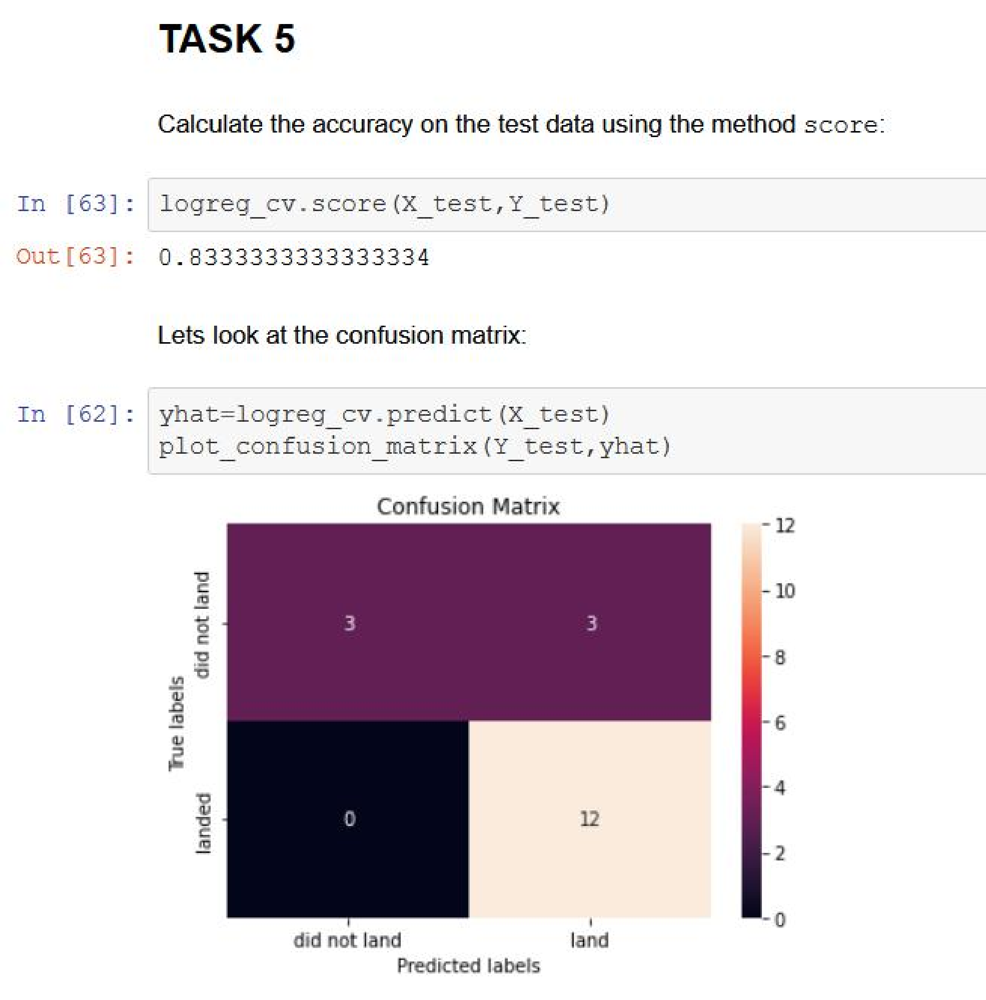
In [77]:

```
print("Logictic Regression Accuracy score: ", logreg_cv.score(X_test,Y_test))
print("Support Vector Machine Accuracy score: ", svm_cv.score(X_test,Y_test))
print("Decision Tree Accuracy score: ",tree_cv.score(X_test,Y_test))
print("KNN Accuracy score: ",knn_cv.score(X_test,Y_test))
```

```
Logictic Regression Accuracy score:  0.8333333333333334
Support Vector Machine Accuracy score:  0.8333333333333334
Decision Tree Accuracy score:  0.7777777777777778
KNN Accuracy score:  0.8333333333333334
```

- Machine Learning Prediction - 4 methods: KNN, Logistic regression, SVM and Decision Tree.
- Three out of four classification models showed the same result - 0.83% of accuracy in predicting results on testing set. Only Decision Tree model showed 0.78% of accuracy.

Confusion Matrix



TASK 5

Calculate the accuracy on the test data using the method score:

```
In [63]: logreg_cv.score(X_test,Y_test)
Out[63]: 0.8333333333333334
```

Lets look at the confusion matrix:

```
In [62]: yhat=logreg_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```

Since three models showed the same accuracy level, their confusion matrixes seemed to be the same as well.

On the screenshot you can see the confusion matrix for logistic regression model.

Conclusions

- The larger the number of flights at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 and continued to 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- We can use any of these three models: KNN, SVM and LG, - for machine learning task of prediction the success rate.

Thank you!