IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Roberto Varillas
09/06/2025

Outline

Executive Summary

Summary of methodologies

-SpaceX Data Collection using SpaceX API

-SpaceX Data Collection with Web Scraping

-SpaceX Data Wrangling

-SpaceX Exploratory Data Analysis using SQL

-Space-X EDA DataViz Using Python Pandas and Matplotlib

-Space-X Launch Sites Analysis with Folium-Interactive Visual Analytics and PlotyDash

-SpaceX Machine Learning Landing Prediction

•Summary of all results

-EDA results

-Interactive Visual Analytics and Dashboards

-Predictive Analysis(Classification)

Introduction

- Project background and context

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 because SpaceX 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 SpaceX for a rocket launch.

- Problems you want to find answers

In this capstone, we will predict if the Falcon 9 first stage will land successfully using data from Falcon 9 rocket launches advertised on its website.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Description of how SpaceX Falcon9 data was collected.

Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.

Finally to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a Jsonresult which was then converted into a Pandas data frame.

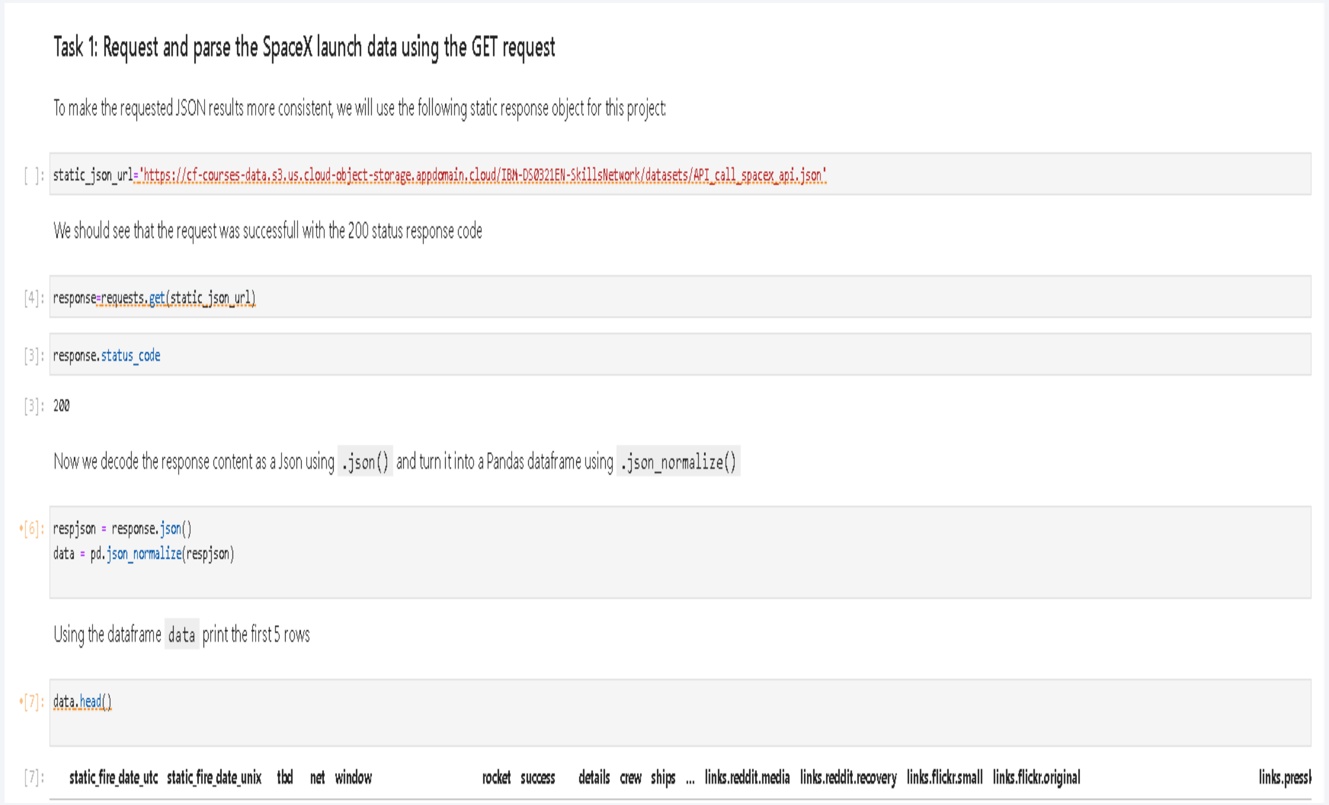
Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launchesof the launch records are stored in a HTML. Using BeautifulSoupand request Libraries, I extract the Falcon 9 launch HTML tablerecordsfrom the Wikipedia page, Parsed the table and converted it into a Pandas data frame

Data Collection – SpaceX API

- Data collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API then requested and parsed the SpaceX launch data using the GET request and decoded the response content as a Json result which was then converted into a Pandas data frame

•Here is the GitHub URL of the completed SpaceX API calls notebook

https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/5337b6bbff27618e2f57379e1e194abbb849d619/(1)jupyter-labs-spacex-data-collection-api.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:

```
[ ]: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

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

```
[4]: response=requests.get(static_json_url)
```

```
[3]: response.status_code
```

```
[3]: 200
```

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

```
[6]: respjson = response.json()
     data = pd.json_normalize(respjson)
```

Using the dataframe `data` print the first 5 rows

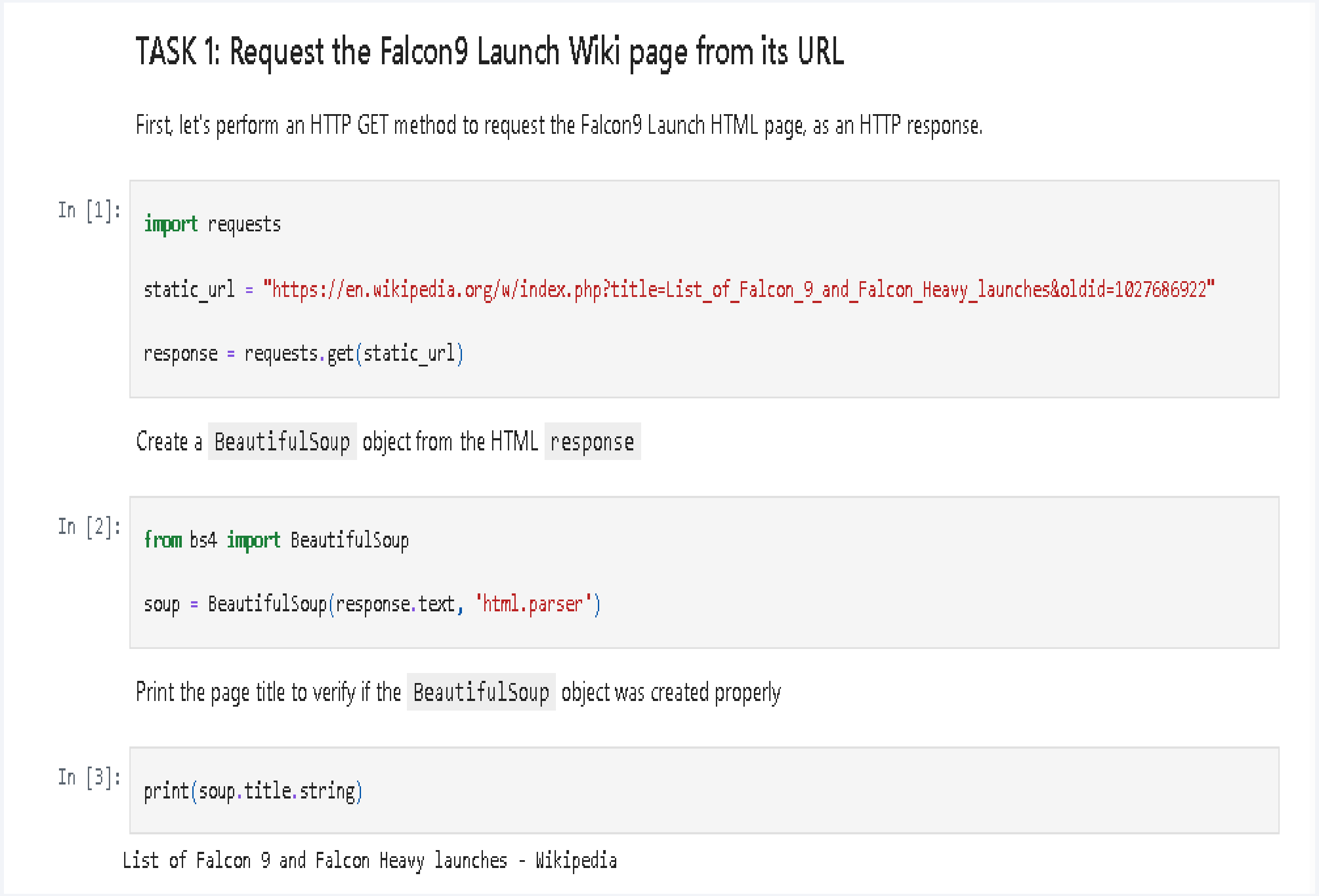
```
[7]: data.head()
```

```
[7]:  static_fire_date_utc  static_fire_date_unix  tbd  net  window          rocket  success   details  crew  ships  ...  links.reddit.media  links.reddit.recovery  links.flickr.small  links.flickr.original          links.pressk
```

Data Collection - Scraping

- Performed web scraping to collect Falcon 9 historical launch records from a Wikipedia using BeautifulSoupand request, to extract the Falcon 9 launch records from HTML table of the Wikipedia page, then created a data frame by parsing the launch HTML.
- •Here is the GitHub URL of the completed web scraping notebook.

https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/edd9a23e8545b13cb2a2b8933941223528069065/jupyter-labs-webscraping.ipynb



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 [1]:
```
import requests

static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

response = requests.get(static_url)
```

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

In [2]:
```
from bs4 import BeautifulSoup

soup = BeautifulSoup(response.text, 'html.parser')
```

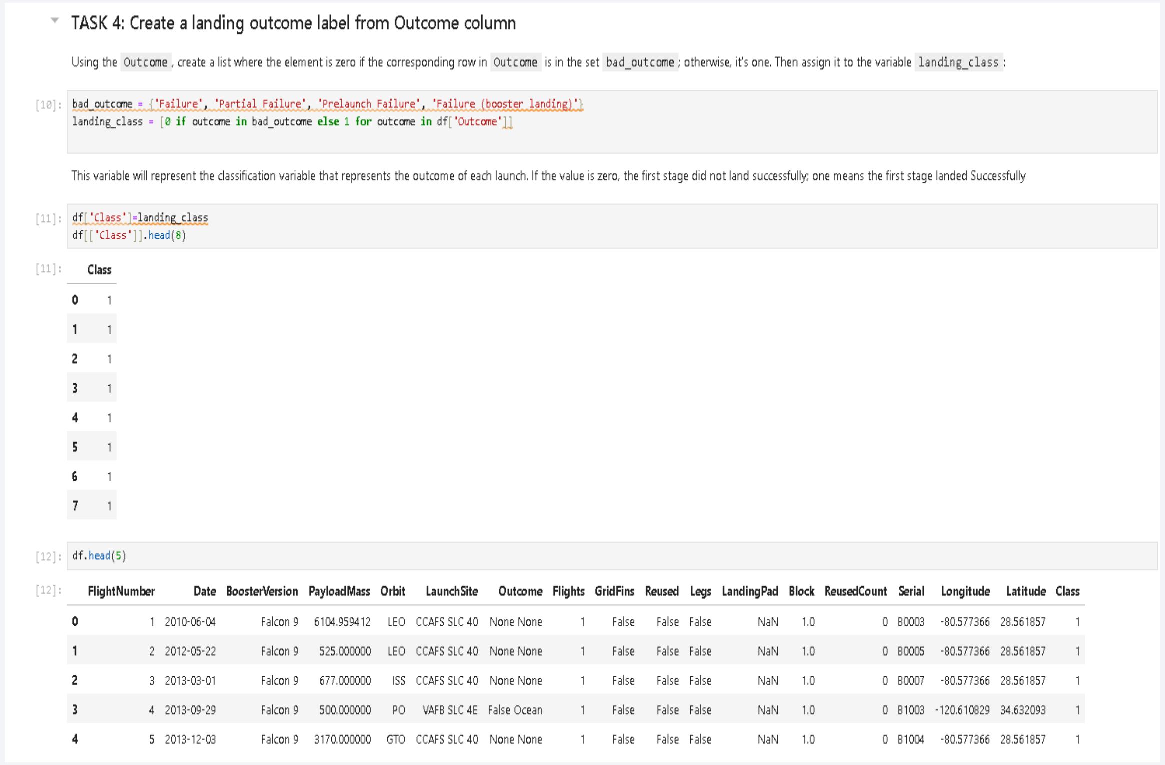
Print the page title to verify if the `BeautifulSoup` object was created properly

In [3]:
```
print(soup.title.string)
```

```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

Data Wrangling

- After obtaining and creating a Pandas DF from the collected data, data was filtered using the BoosterVersioncolumn to only keep the Falcon 9 launches, then dealt with the missing data values in the LandingPadand PayloadMasscolumns. For the PayloadMass, missing data values were replaced using mean value of column.
- •Also performed some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models
- •Here is the GitHub URL of the completed data wrangling related notebooks.
- https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/c14368714ee1b8b35fff0d57099aff8807080e9e/labs-jupyter-spacex-Data%20wrangling.ipynb



TASK 4: Create a landing outcome label from Outcome column

Using the `Outcome`, create a list where the element is zero if the corresponding row in `Outcome` is in the set `bad_outcome`; otherwise, it's one. Then assign it to the variable `landing_class`:

```
[10]: bad_outcome = {'Failure', 'Partial Failure', 'Prelaunch Failure', 'Failure (booster landing)'}
      landing_class = [0 if outcome in bad_outcome else 1 for outcome in df['Outcome']]
```

This variable will represent the classification variable that represents the outcome of each launch. If the value is zero, the first stage did not land successfully; one means the first stage landed Successfully

```
[11]: df['Class']=landing_class
      df[['Class']].head(8)
```

[11]:

	Class
0	1
1	1
2	1
3	1
4	1
5	1
6	1
7	1

```
[12]: df.head(5)
```

[12]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	1
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	1
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	1
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	1
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	1

EDA with Data Visualization

Performed data Analysis and Feature Engineering using Pandas and Matplotlib.i.e.

•Exploratory Data Analysis

•Preparing Data Feature Engineering

•Used scatter plots to Visualize the relationship between Flight Number and Launch Site, Payload and Launch Site, FlightNumberand Orbit type, Payload and Orbit type.

•Used Bar chart to Visualize the relationship between success rate of each orbit type

•Line plot to Visualize the launch success yearly trend.

•Here is the GitHub URL of your completed EDA with data visualization notebook.

https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/a9efb63266c13d71864b94385450ddf1fddbbac5/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

The following SQL queries were performed for EDA

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

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

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

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

•List the date when the first successful landing outcome in ground pad was achieved

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

•List the total number of successful and failure mission outcomes

•Here is the GitHub URL of your completed EDA with SQL notebook.

https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/a9efb63266c13d71864b94385450ddf1fddbbac5/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Created folium map to marked all the launch sites, and created map objects such as markers, circles, lines to mark the success or failure of launches for each launch site.

•Created a launch set outcomes (failure=0 or success=1).

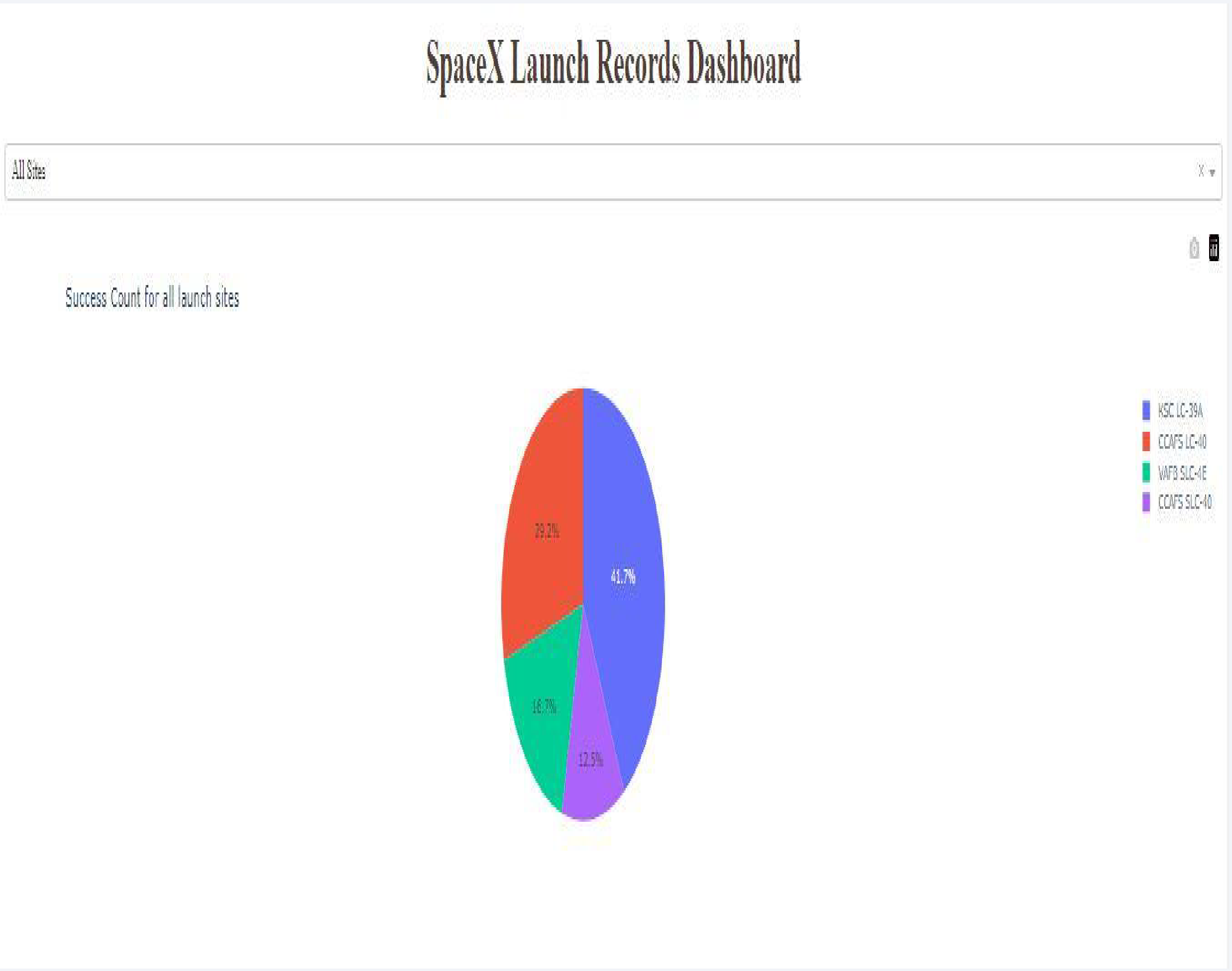
•Here is the GitHub URL of the completed interactive map with Folium map, as an external reference and peer-review purpose

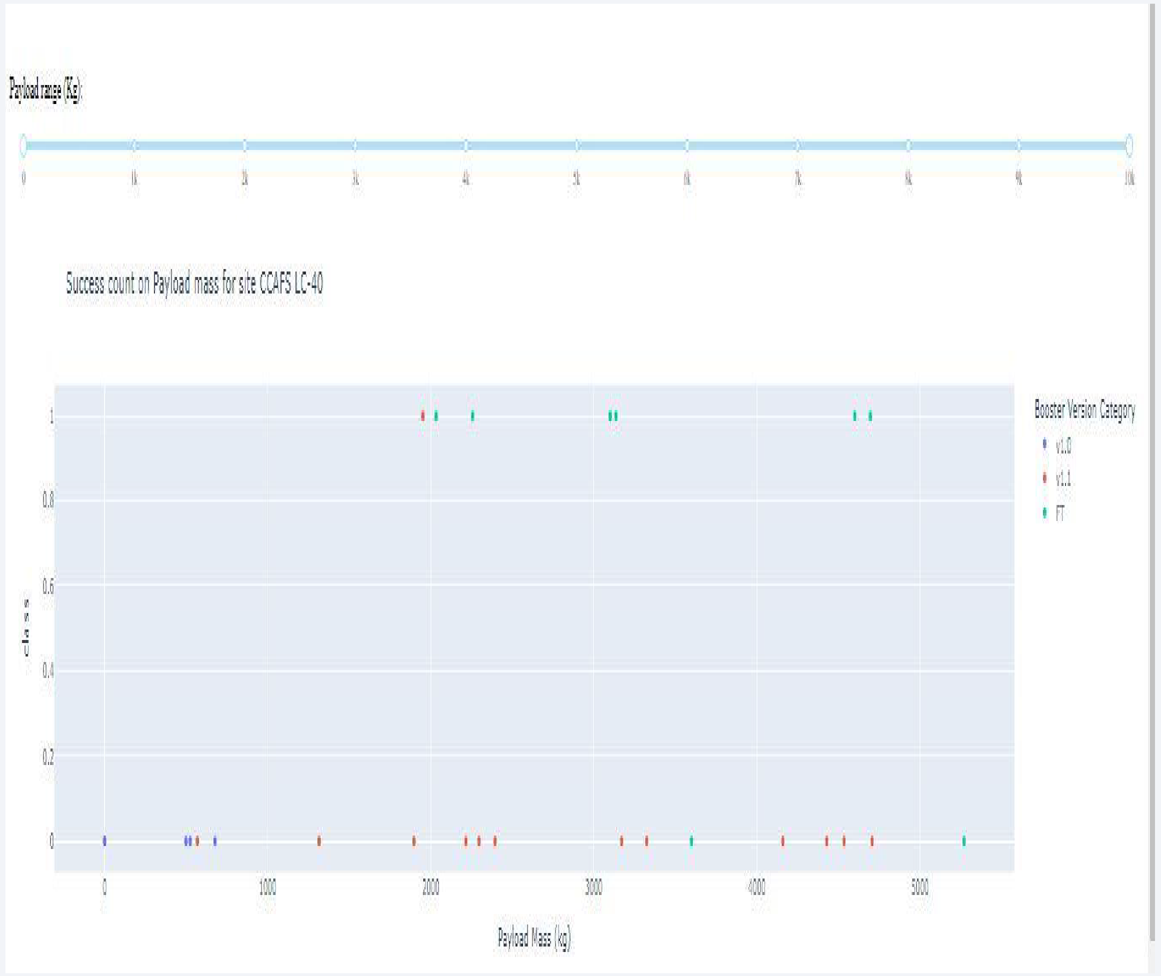
https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/6145774aec4c4405058a5ee28ca4e3d346002b46/lab_jupyter_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard application with Plotlydash by:
- •Adding a Launch Site Drop-down Input Component
- •Adding a callback function to render success-pie-chart based on selected site dropdown
- •Adding a Range Slider to Select Payload
- •Addenga callback function to render the success-payload-scatter-chart scatter plot
- •Here is the GitHub URL of your completed PlotlyDash lab
- https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/ca90c8c92f5bad05f77d13e140e3a49cb5a6217b/test%20peluche.py

Space x Dash App





Predictive Analysis (Classification)

- Summary of Classification Model Development

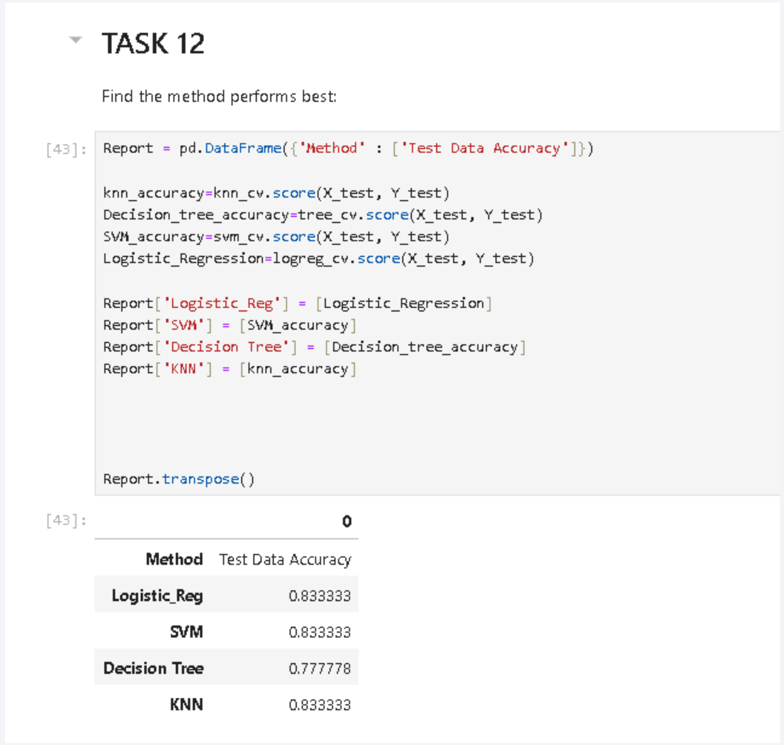
-Data Preparation:
-Loaded dataset as Pandas DataFrame.
-Extracted target variable Y from the Class columna.
-Standardized features using StandardScaler.
-Split data into training (80%) and testing (20%) sets.

- Model Selection & Tuning:

-Evaluated SVM, Decision Trees, KNN, and Logistic Regression.
-Used GridSearchCV with 10-fold CV to find optimal hyperparameters.
-Selected best parameters via best_params_ and validation accuracy via best_score_.

- Evaluation:

-Calculated test accuracy for each model.
-Plotted confusion matrices.
-Compared models—selected the one with highest test accuracy.



TASK 12

Find the method performs best:

```
[43]: Report = pd.DataFrame({'Method' : ['Test Data Accuracy']})

      knn_accuracy=knn_cv.score(X_test, Y_test)
      Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
      SVM_accuracy=svm_cv.score(X_test, Y_test)
      Logistic_Regression=logreg_cv.score(X_test, Y_test)

      Report['Logistic_Reg'] = [Logistic_Regression]
      Report['SVM'] = [SVM_accuracy]
      Report['Decision Tree'] = [Decision_tree_accuracy]
      Report['KNN'] = [knn_accuracy]



      Report.transpose()
```

[43]:

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

GitHub URL of the completed predictive analysis lab: https://github.com/rxbertx01/SpaceX-Falcon-9-1st-stage/blob/200ef2fe1de4763fe241cace0b62f2570a82a6ee/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

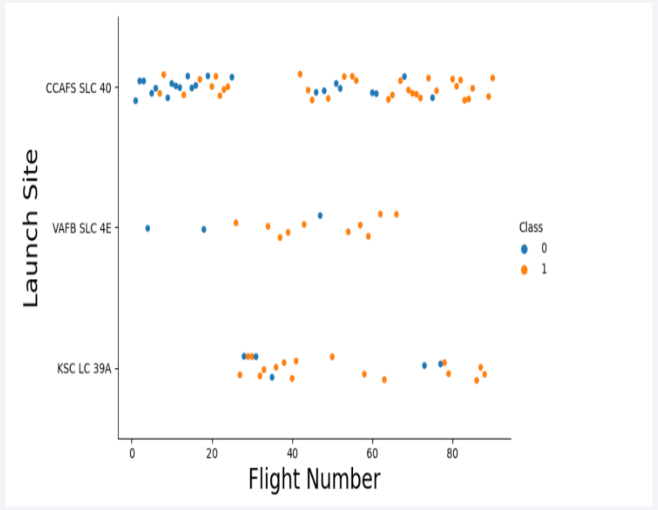
Section 2

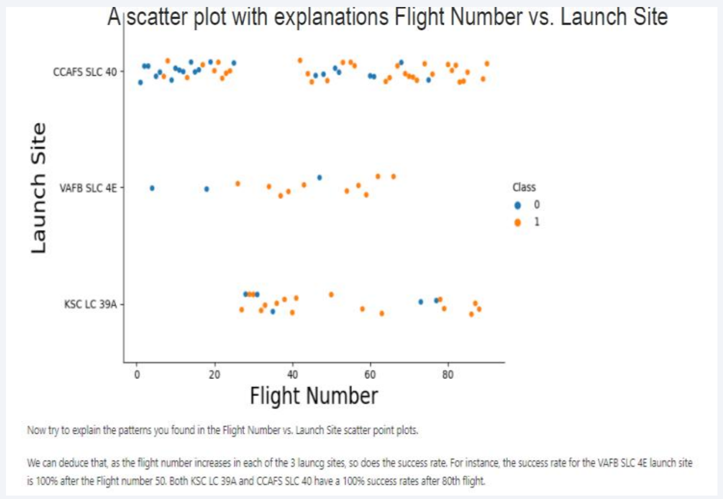
Insights drawn from EDA

Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations





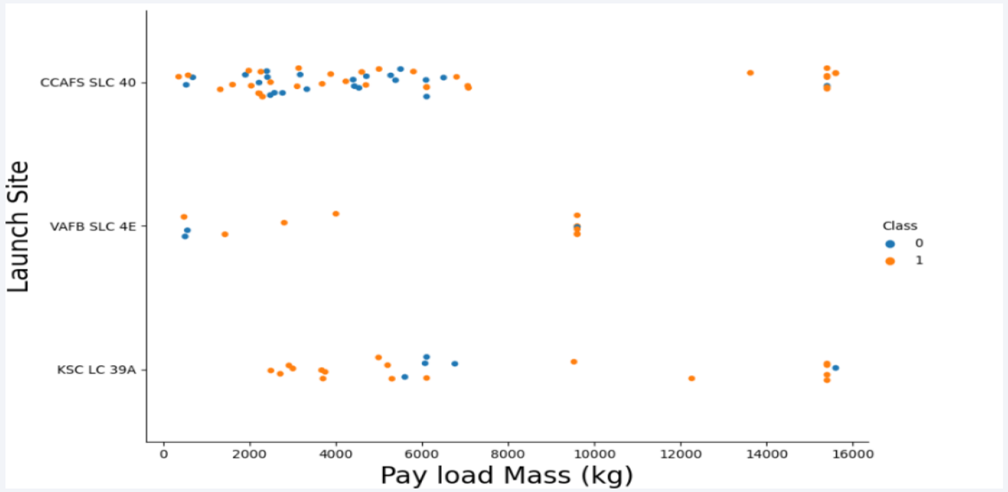
Now try to explain the patterns you found in the Flight Number vs. Launch Site scatter point plots.

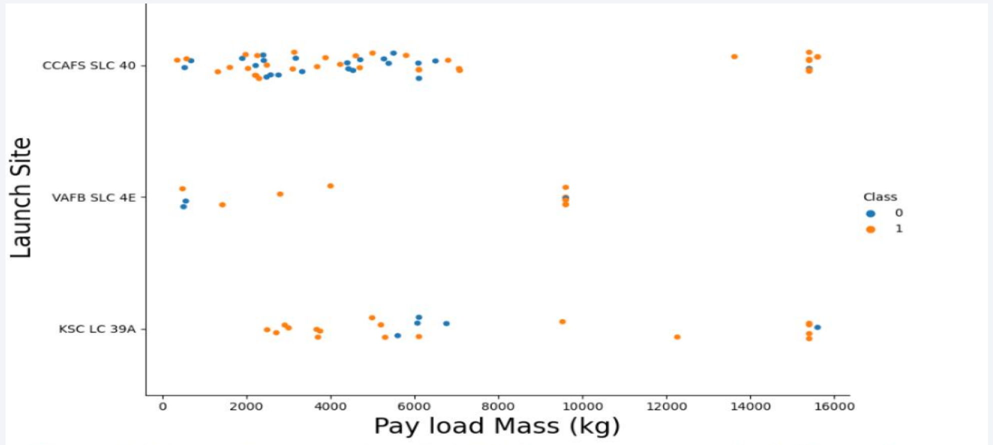
We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations

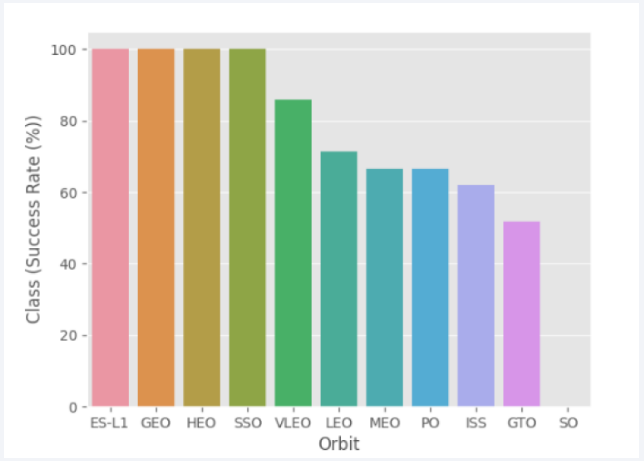




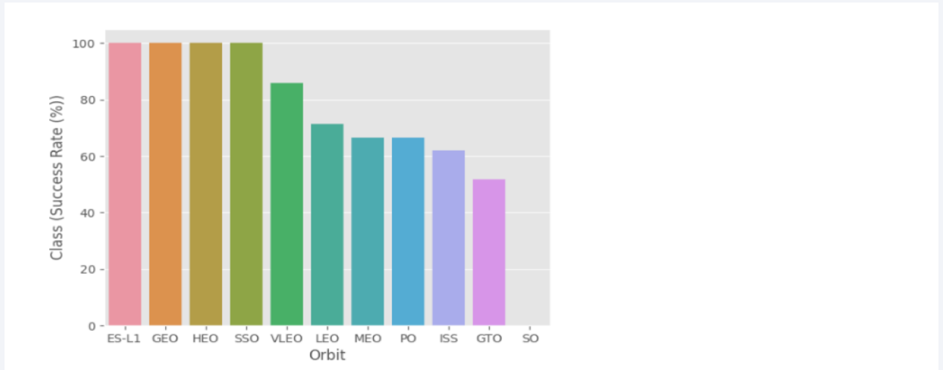
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

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



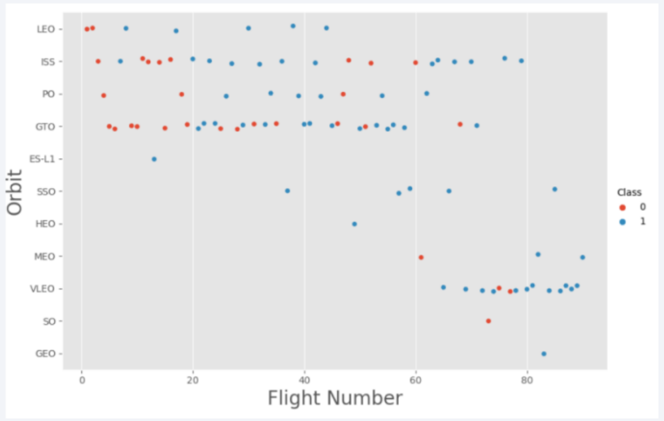
Analyze the ploted bar chart try to find which orbits have high sucess rate.

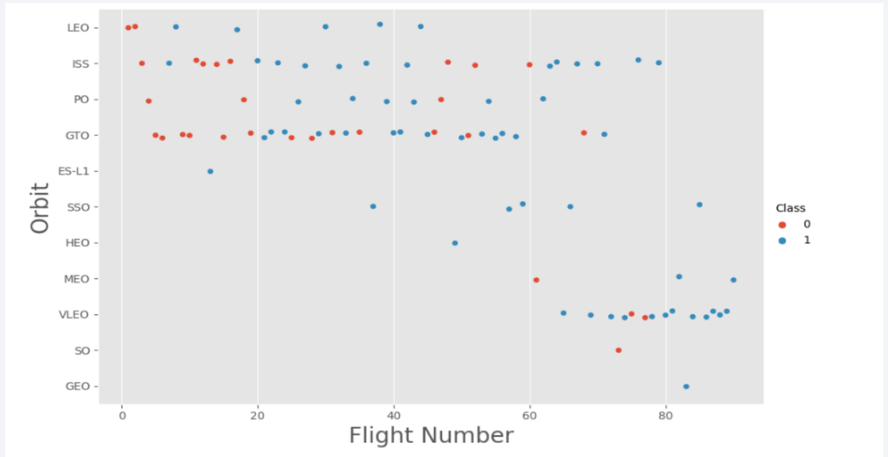
Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

Flight Number vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations

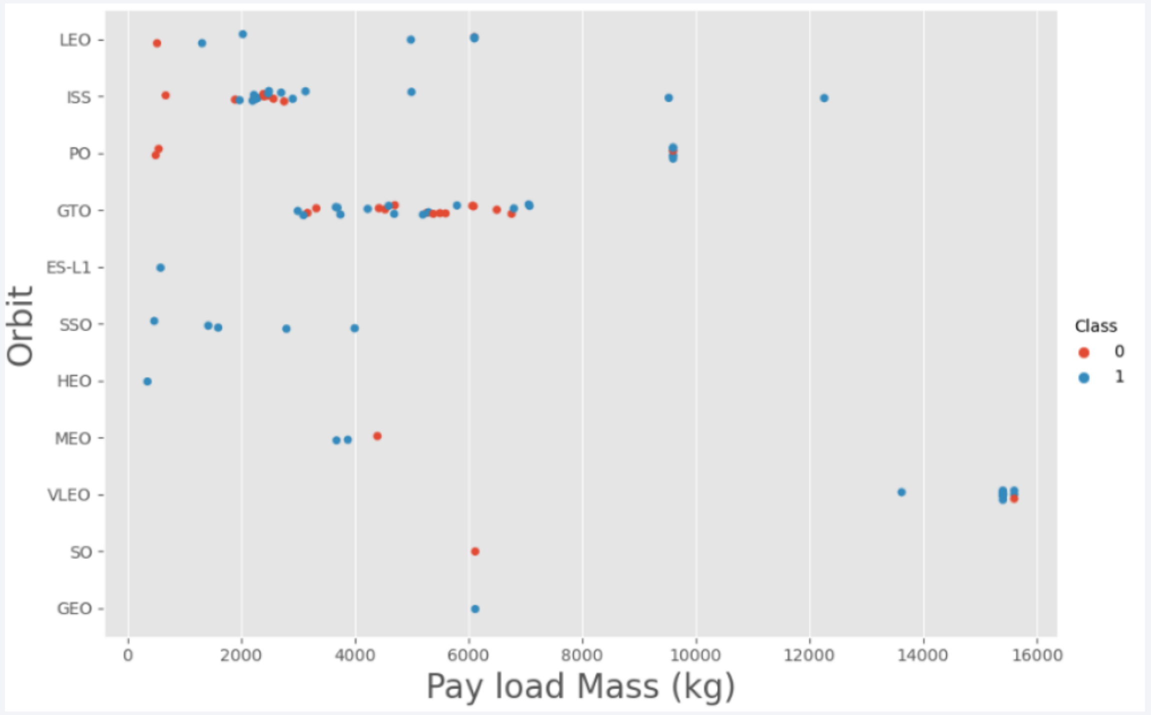




You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

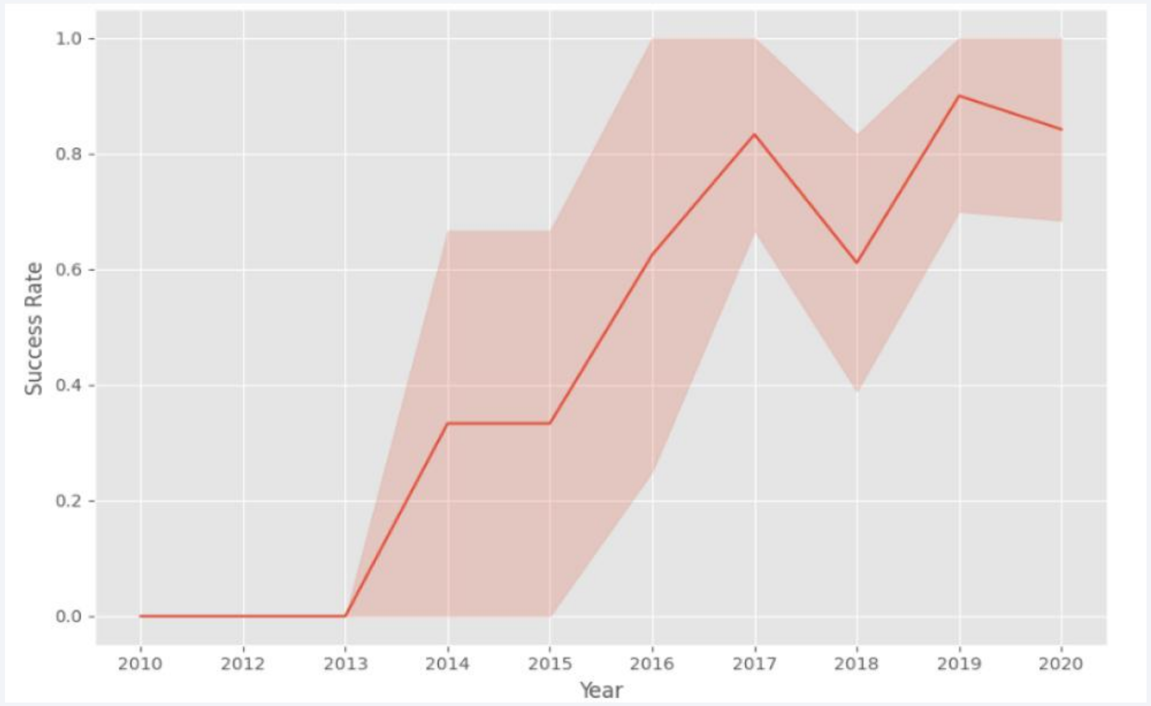
Payload vs. Orbit Type

- 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) both have near equal chances



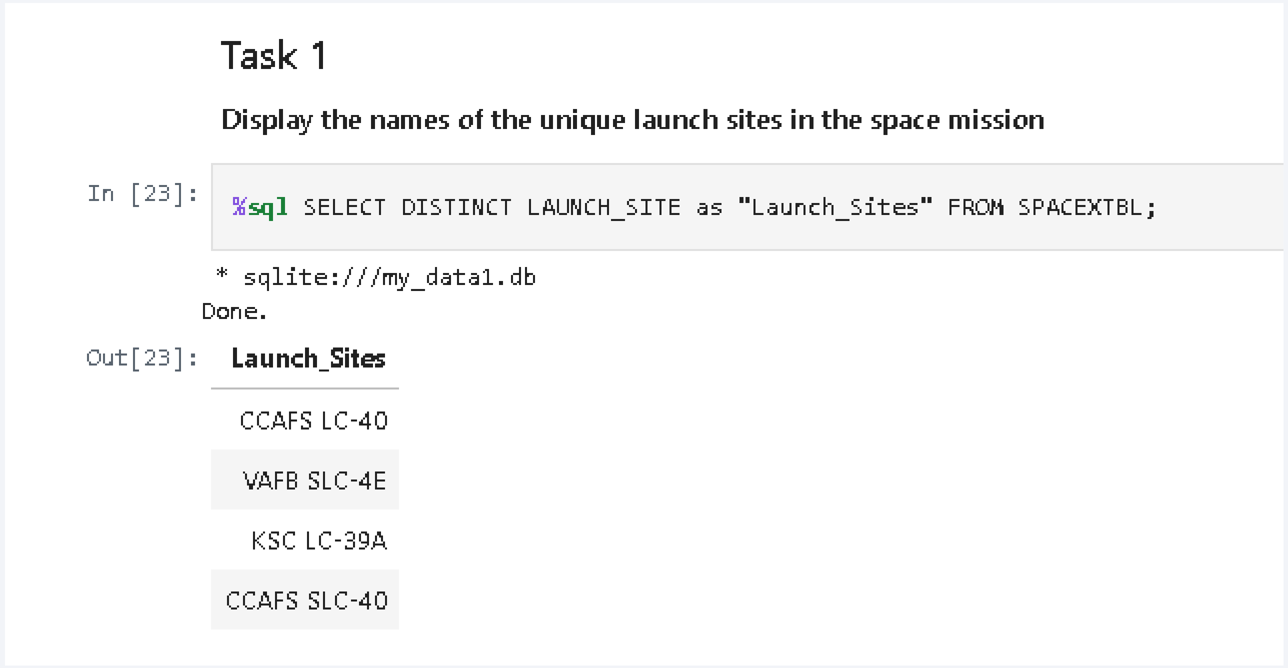
Launch Success Yearly Trend

- Since 2013, the success rate kept going up till 2020



All Launch Site Names

- Find the names of the unique launch sites
- Used 'SELECT DISTINCT' statement to return only the unique launch sites from the 'LAUNCH_SITE' column of the SPACEXTBL table



Task 1

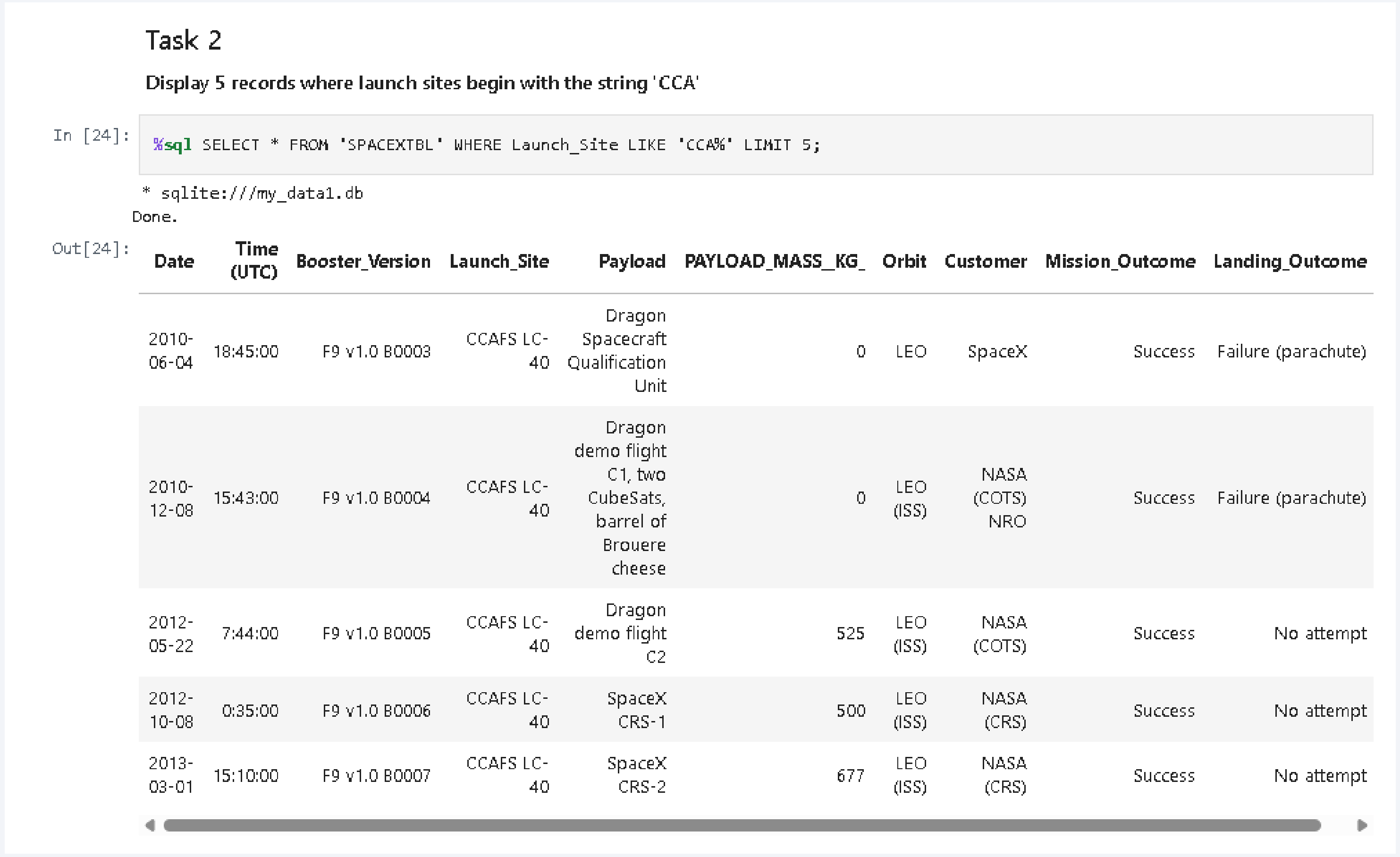
Display the names of the unique launch sites in the space mission

```
In [23]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;

 * sqlite:///my_data1.db
Done.
Out[23]:
```

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

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here



Task 2

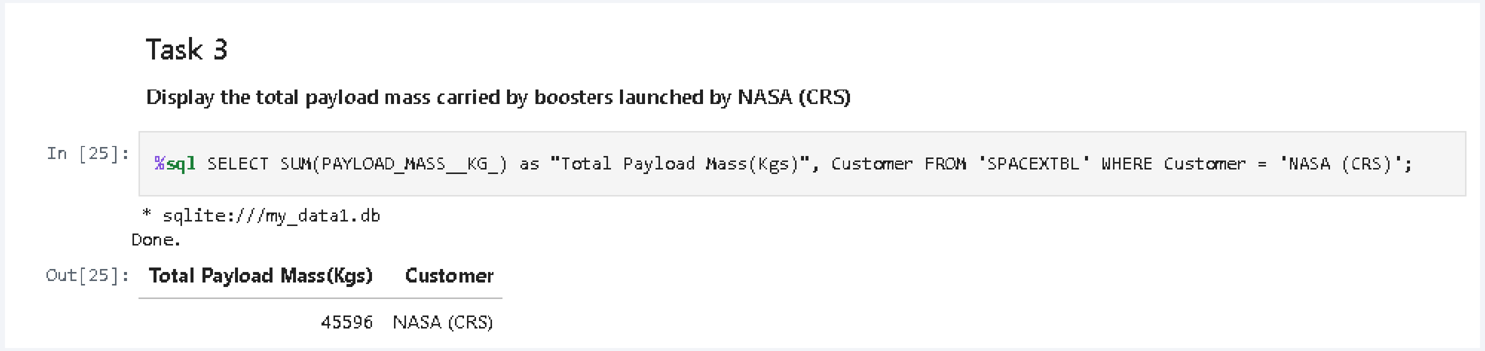
Display 5 records where launch sites begin with the string 'CCA'

```
In [24]: %sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

 * sqlite:///my_data1.db
Done.
Out[24]:
```

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here



Task 3

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

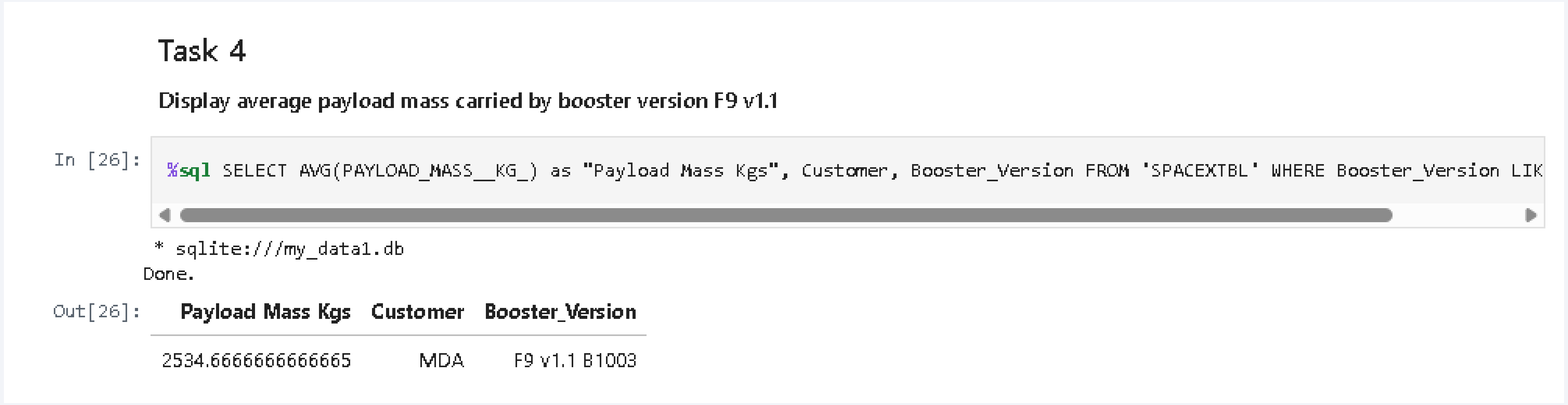
```
In [25]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';

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

Out[25]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here



Task 4

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

In [26]:

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIK
```

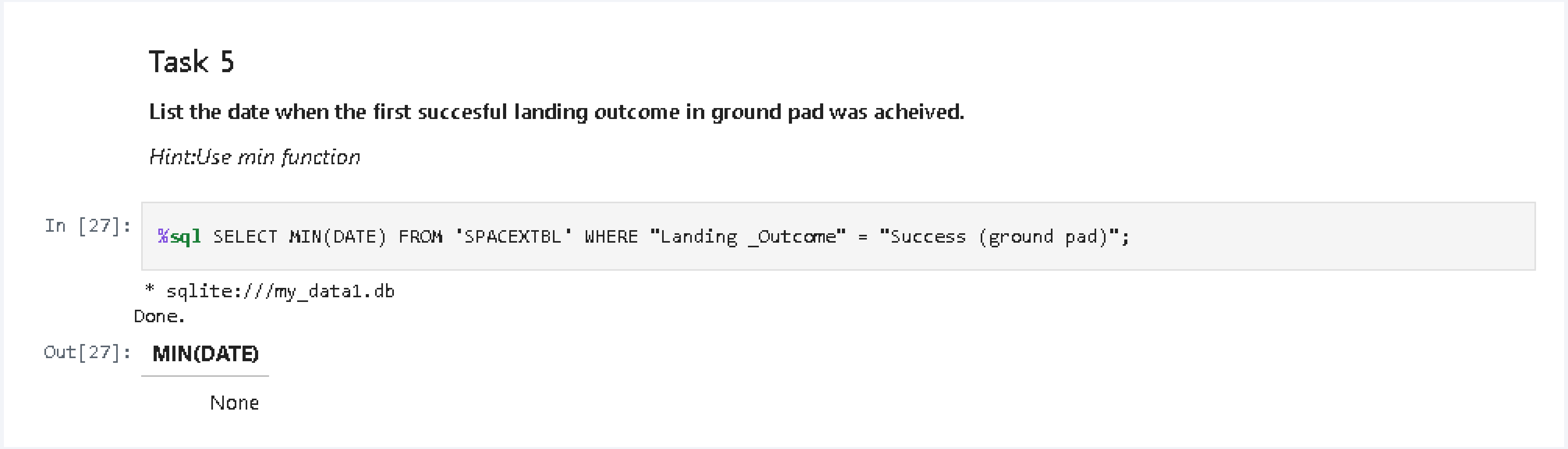
```
 * sqlite:///my_data1.db
Done.
```

Out[26]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here



Task 5

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

Hint:Use min function

```
In [27]: %sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

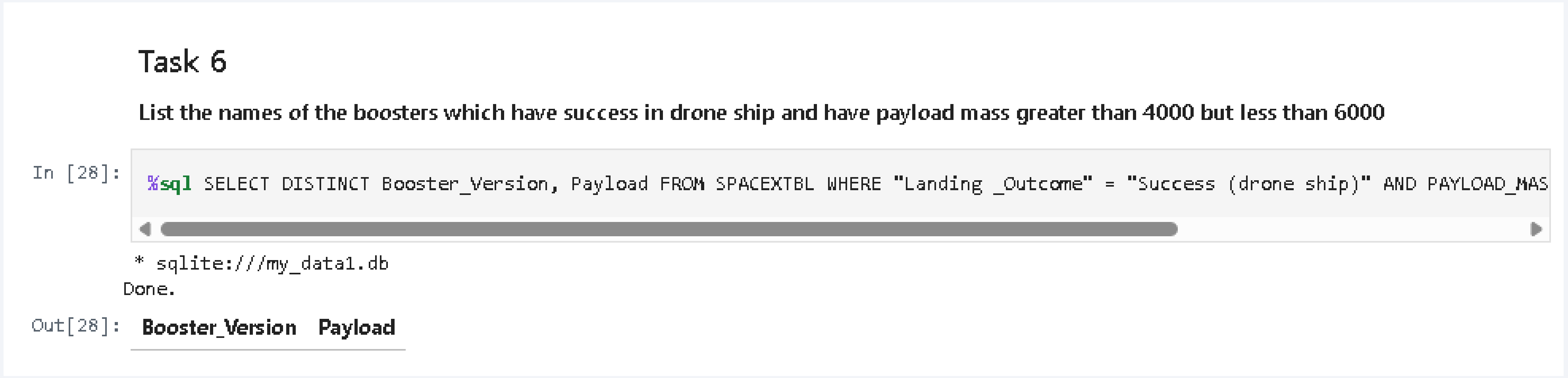
```
 * sqlite:///my_data1.db
Done.
```

```
Out[27]: MIN(DATE)
              None
```

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

- Present your query result with a short explanation here



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 [28]: %sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MAS
```

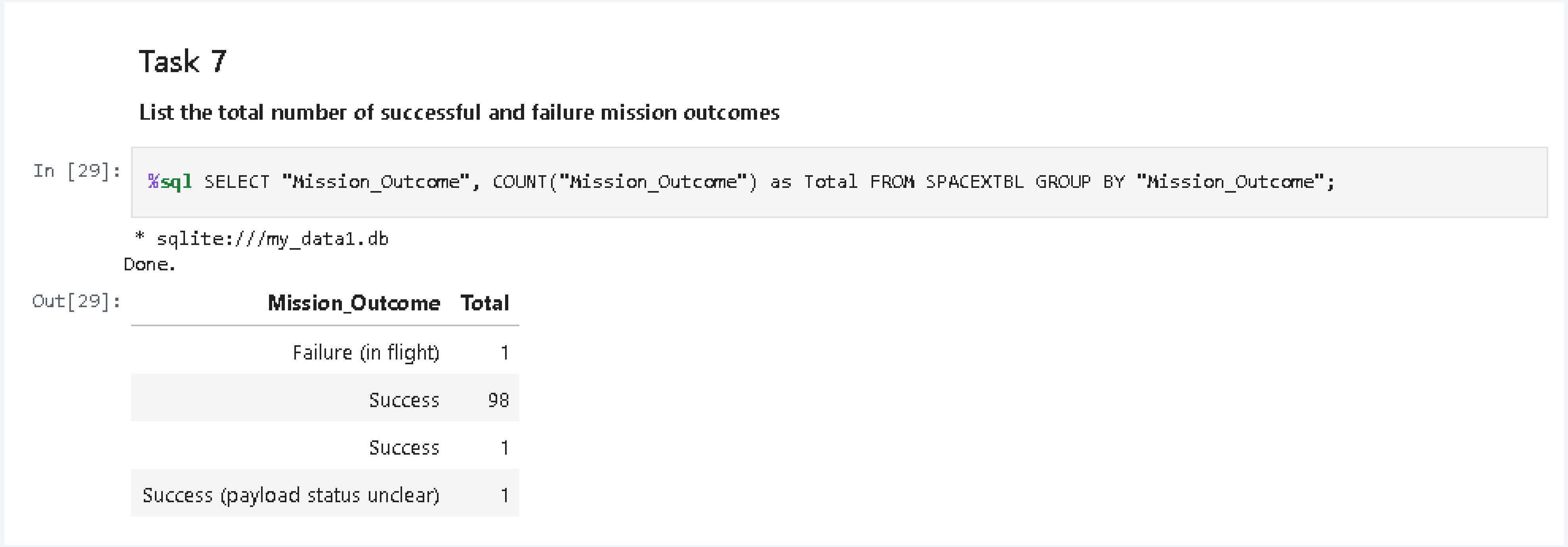
```
 * sqlite:///my_data1.db
Done.
```

Out[28]:

Booster_Version	Payload

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here



Task 7

List the total number of successful and failure mission outcomes

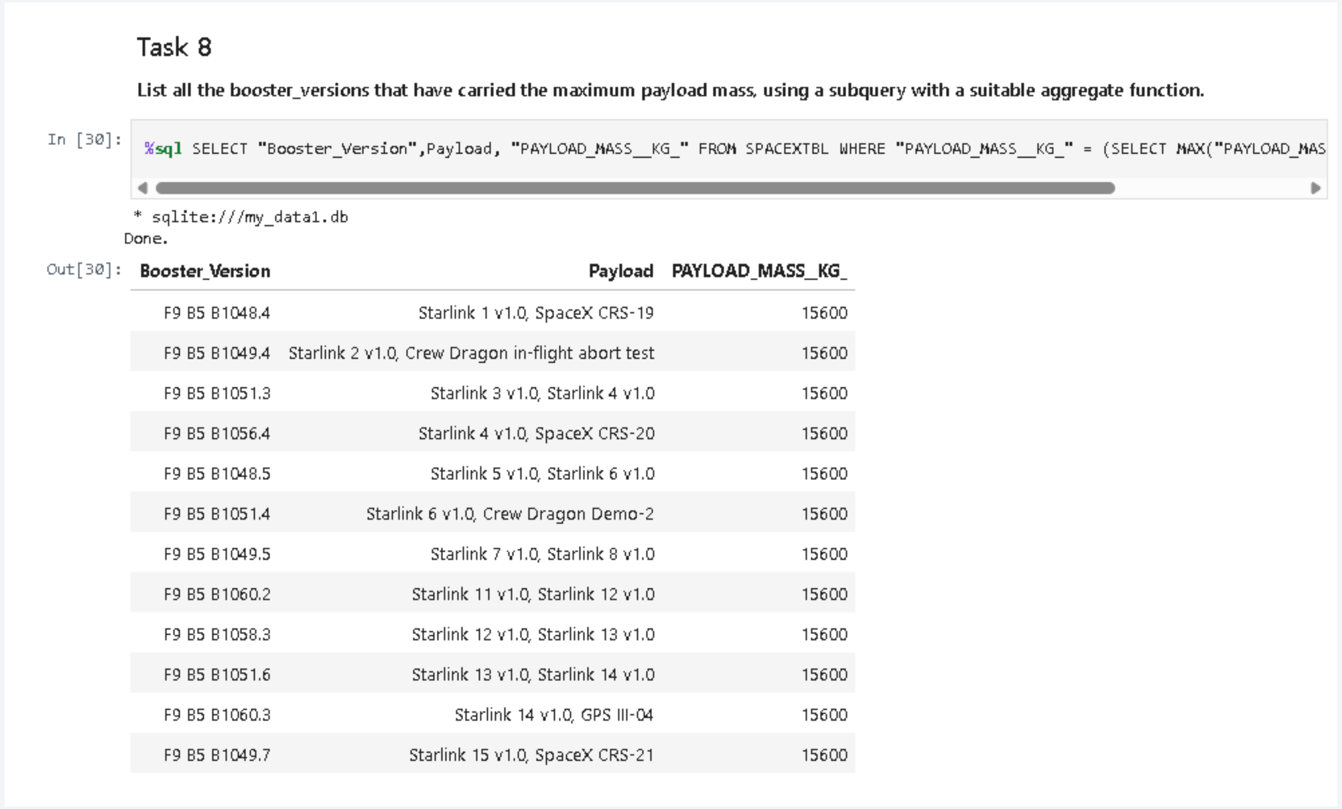
```
In [29]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";

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

Out[29]:

Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here



Task 8

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
In [30]: %sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MAS
```

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

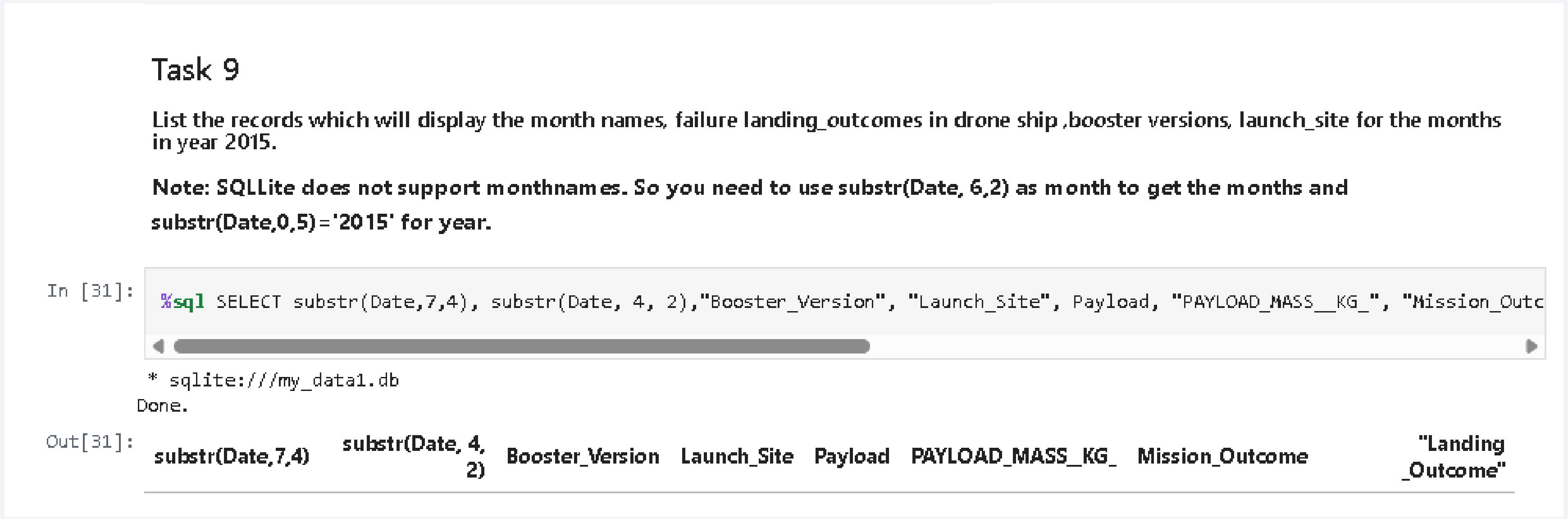
Out[30]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

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

- Present your query result with a short explanation here



Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
In [31]: %sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outc
```

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

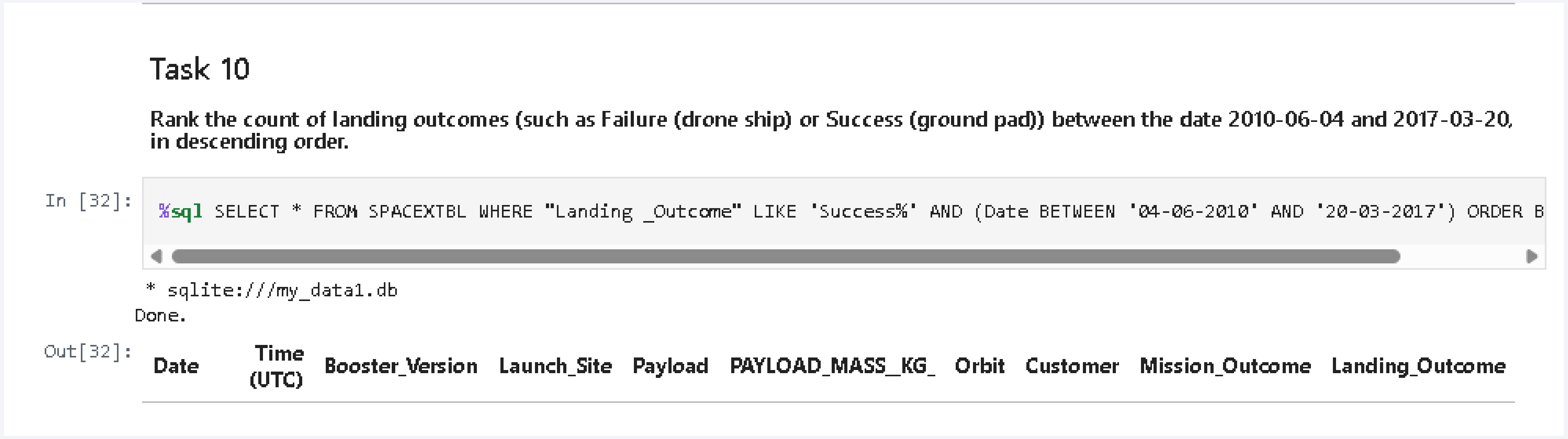
Out[31]:

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Mission_Outcome	"Landing _Outcome"

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

- Present your query result with a short explanation here



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 [32]: %sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER B
```

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

Out[32]:

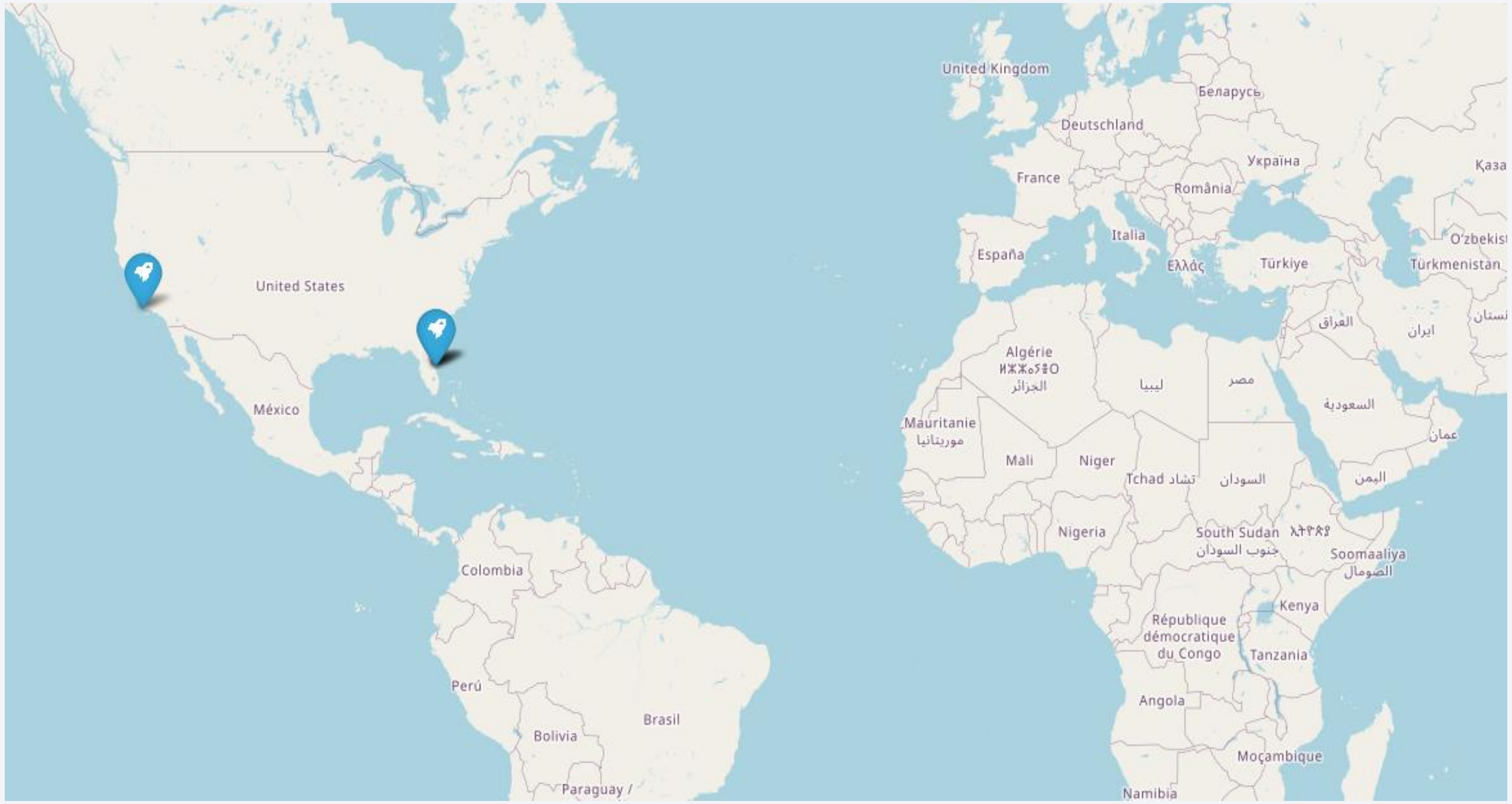
Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome

Section 3

Launch Sites Proximities Analysis

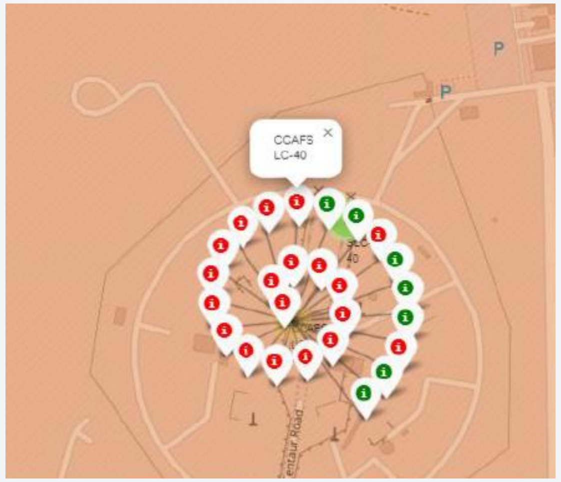
Markers of all launch sites on global map

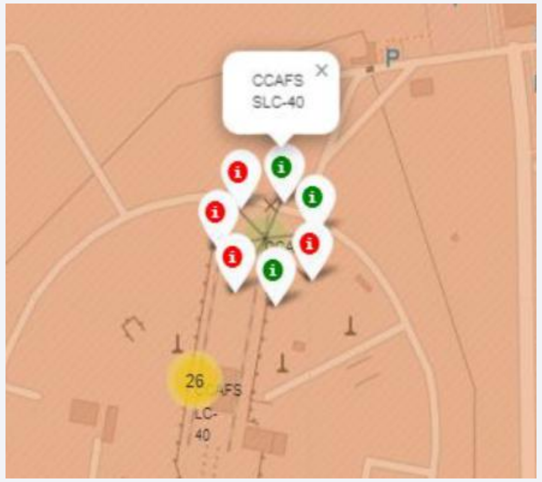
All launch sites are in proximity to the Equator, (located southwards of the US map). Also all the laumch sites are in very close proximity to the coast.

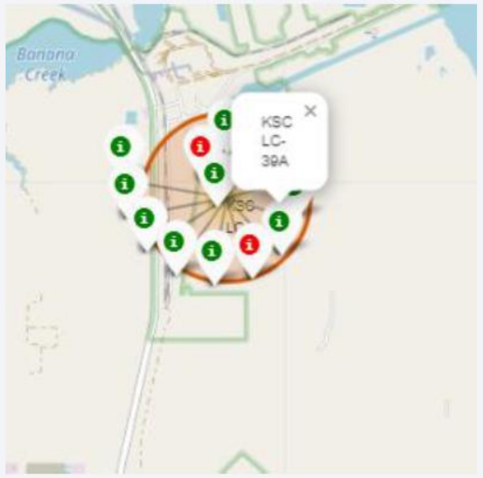


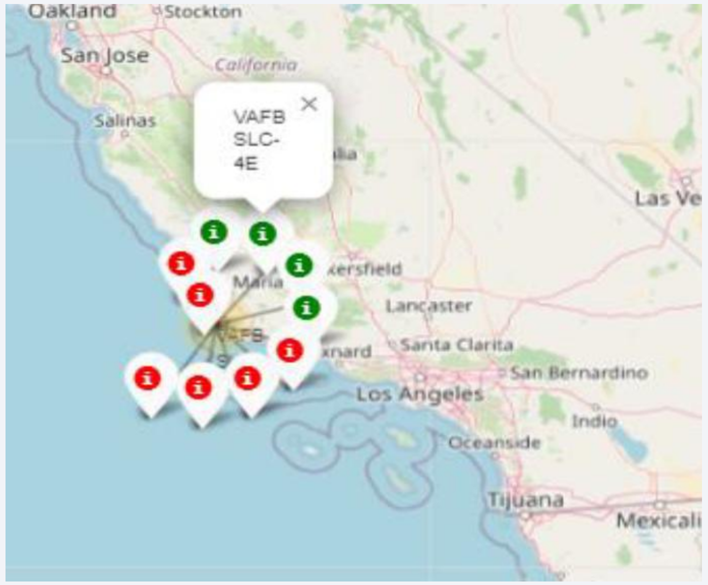
Launch outcomes for each site on the map With Color Markers

- In the Eastern coast (Florida) Launch site KSC LC-39A has relatively high success rates compared to CCAFS SLC-40 & CCAFS LC-40.
- In the West Coast (Californai) Launch site VAFB SLC-4E has relatively lower success rates 4/10 compared to KSC LC-39A launch site in the Eastern Coast of Florida.



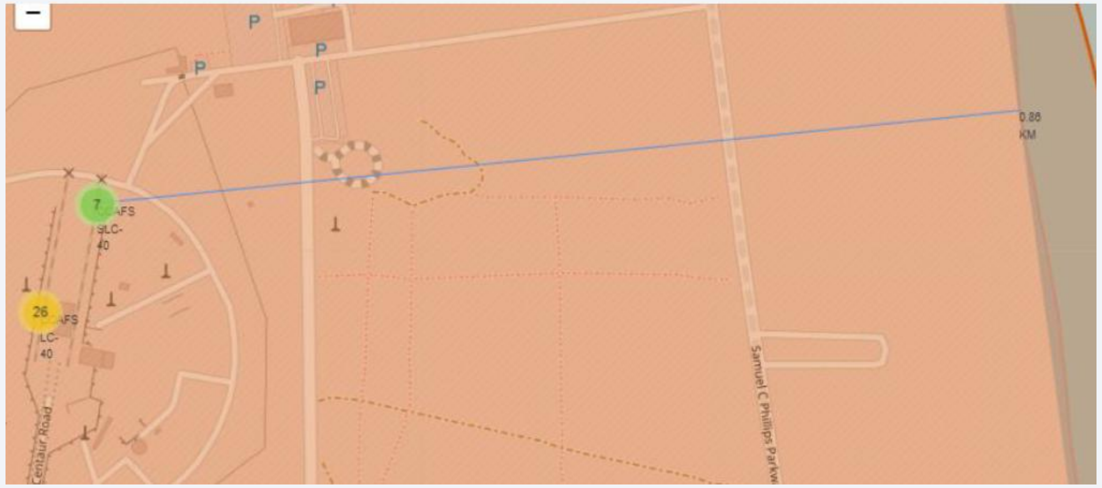




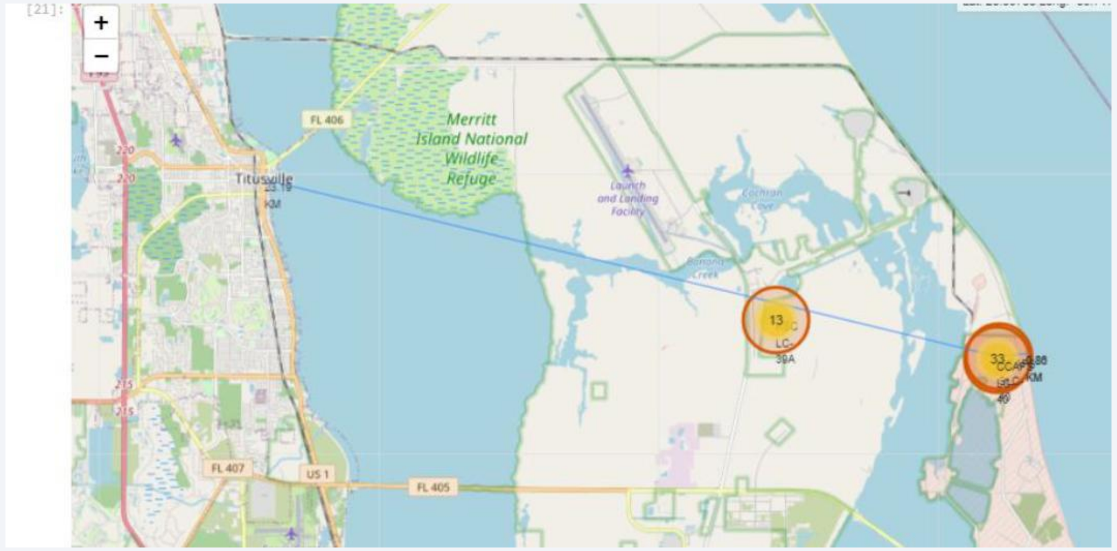


Distances between a launch site to its proximities

- Launch siteCCAFS SLC-40proximity to coastline is 0.86km



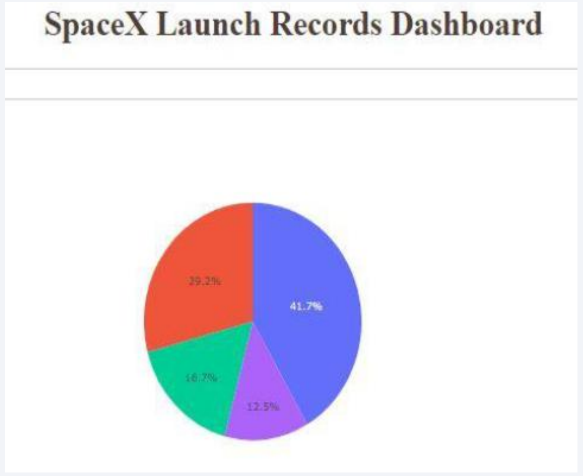
- Launch siteCCAFS SLC-40closest to highway (Washington Avenue) is 23.19km

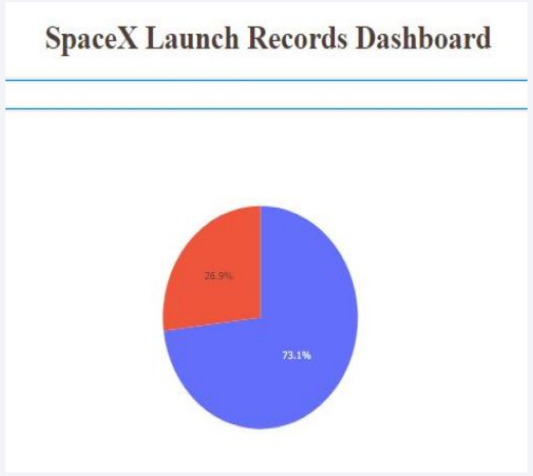

Titusville
23.19 KM
Merritt Island National Wildlife Refuge
Launch and Landing Facility
FL 406
FL 405
FL 407
US 1
13
LC-39A
33

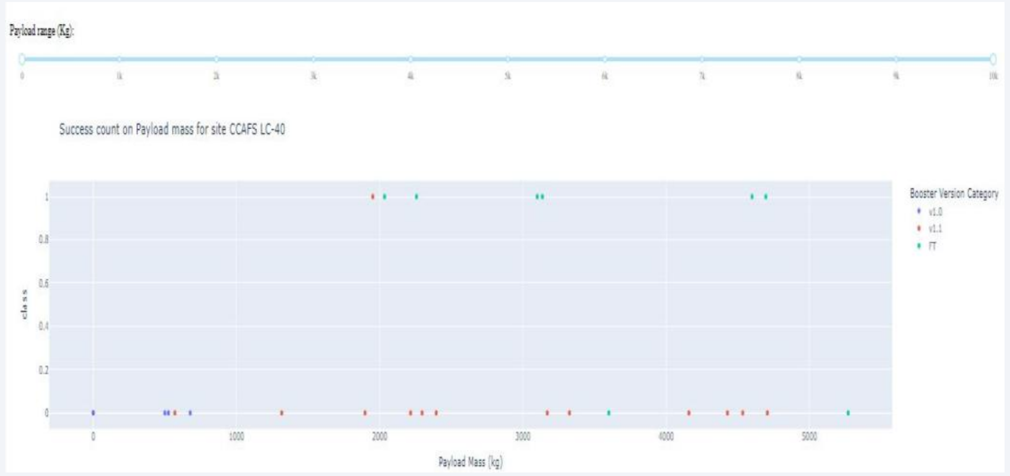
Section 4

Build a Dashboard with Plotly Dash

Pie-Chart for launch success count for all sites







- Launch site KSC LC-39A has the highest launch success rate at 42% followed by CCAFS LC-40 at 29%, VAFB SLC-4E at 17% and lastly launch site CCAFS SLC-40 with a success rate of 13%

Launch site CCAFS LC-40 had the 2ndhighest success ratio of 73% success against 27% failed launches

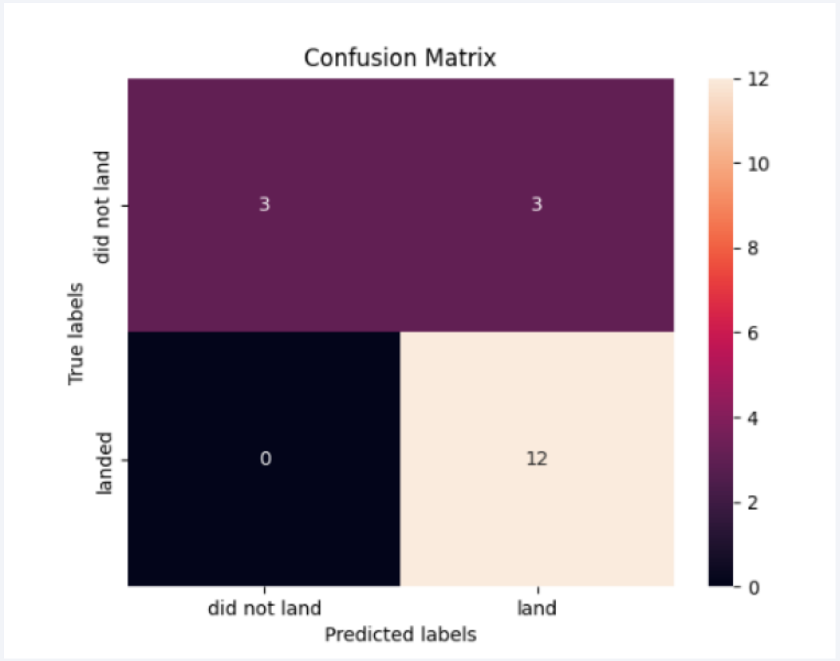
For Launch site CCAFS LC-40 the booster version FT has the largest success ratefrom a payload mass of >2000kg

Section 5

Predictive Analysis (Classification)

Confusion Matrix

- All the 4 classification model had the same confusion matrixes and were able equally distinguish between the different classes. The major problem is false positives for all the models.



Conclusions

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%.

•We can deduce that, as the flight number increases in each of the 3 launcg sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight

•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).

•Orbits ES-L1, GEO, HEO & SSO have the highest success rates at 100%, with SO orbit having the lowest success rate at ~50%. Orbit SO has 0% success rate.

•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

Thank you!