IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Austin Lee

2024-07-09

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

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. The goal of the project is to create machine learning pipeline to predict the first stage will land successfully.

- Problems you want to find answers
  - Features (Engine, Total Mass, Load Mass, and etc) for the first successful stage
  - The interaction between features
  - Operating conditions of launching sites, Ocean vs land

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data is collected from Space X API and Web Scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Using the data obtained, we train multiple machine learning algorithms and evaluated them using same sets of train and text datasets.
    - logistic regression, Support Vector Machine, decision tree, k-nearest neighbors.

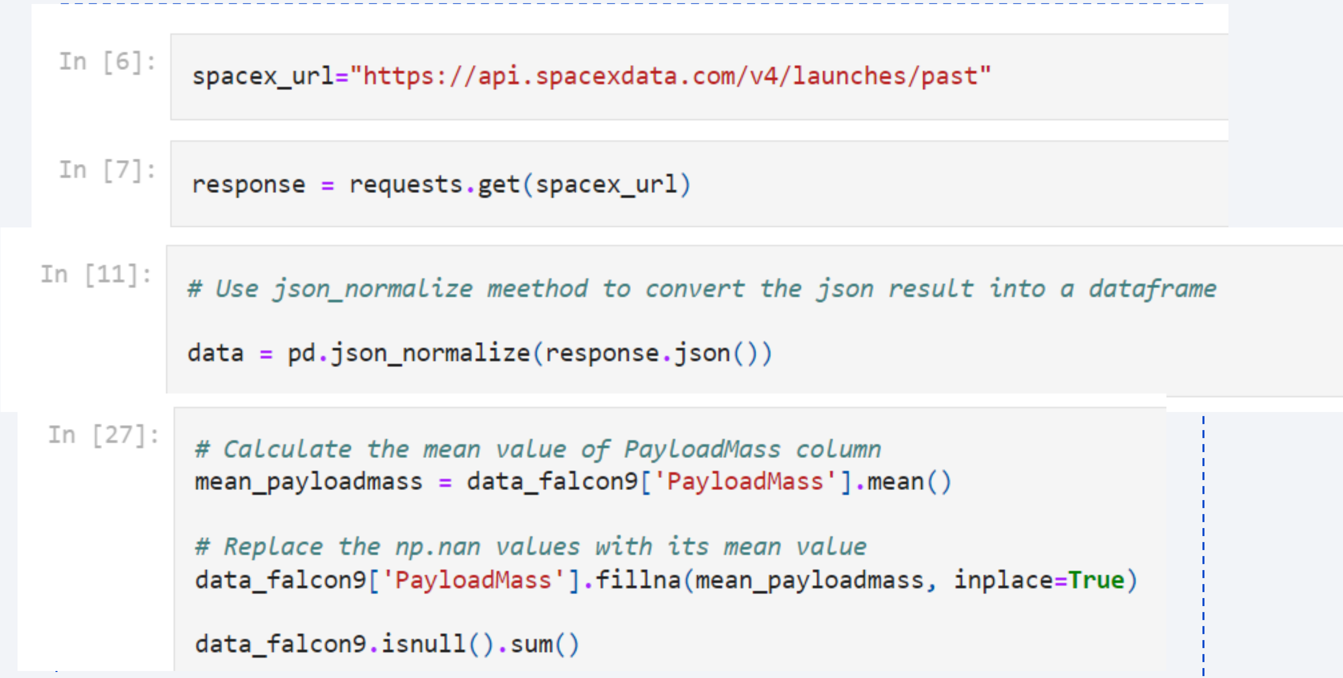
Data Collection

- SpaceX API.
  - The api url is invoked using request.get.
  - The json data from the api request is nomalized using .json_normalized in Pandas dataframe.
    - pd.json_normalize(response.json())
  - Data wrangling or cleaning to fill out the missing data suing .fillna method in pandas dataframe.
- Web scraping from Wikipedia for Falcon 9 launch records
  - The html data is obtained using BeautifulSoup.

Data Collection – SpaceX API

- The json format data is obtained from api url. The data is nomalized, cleaned, and formatted.

- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb



```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

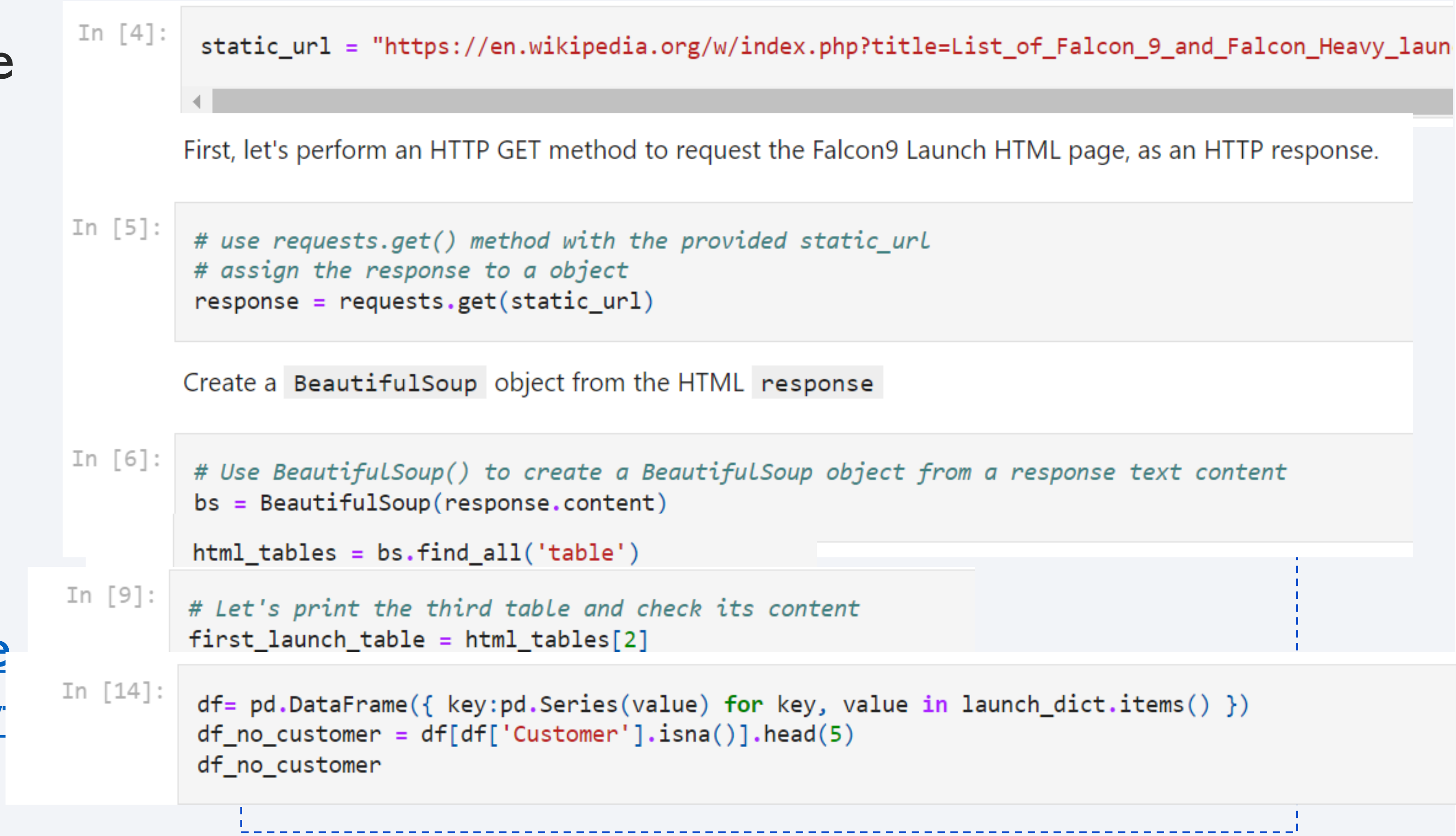
```
# Use json_normalize meethod to convert the json result into a dataframe

data = pd.json_normalize(response.json())
```

```
# Calculate the mean value of PayloadMass column
mean_payloadmass = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].fillna(mean_payloadmass, inplace=True)

data_falcon9.isnull().sum()
```

Data Collection - Scraping

- The html page containing the data table is scrapped and the html table is extracted using Beautifulsoap. The extracted data is converted to pandas dataframe.

- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/jupyter-labs-webscraping.ipynb



In [4]:

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_laun
```

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

In [5]:

```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

In [6]:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
bs = BeautifulSoup(response.content)
```

```
html_tables = bs.find_all('table')
```

In [9]:

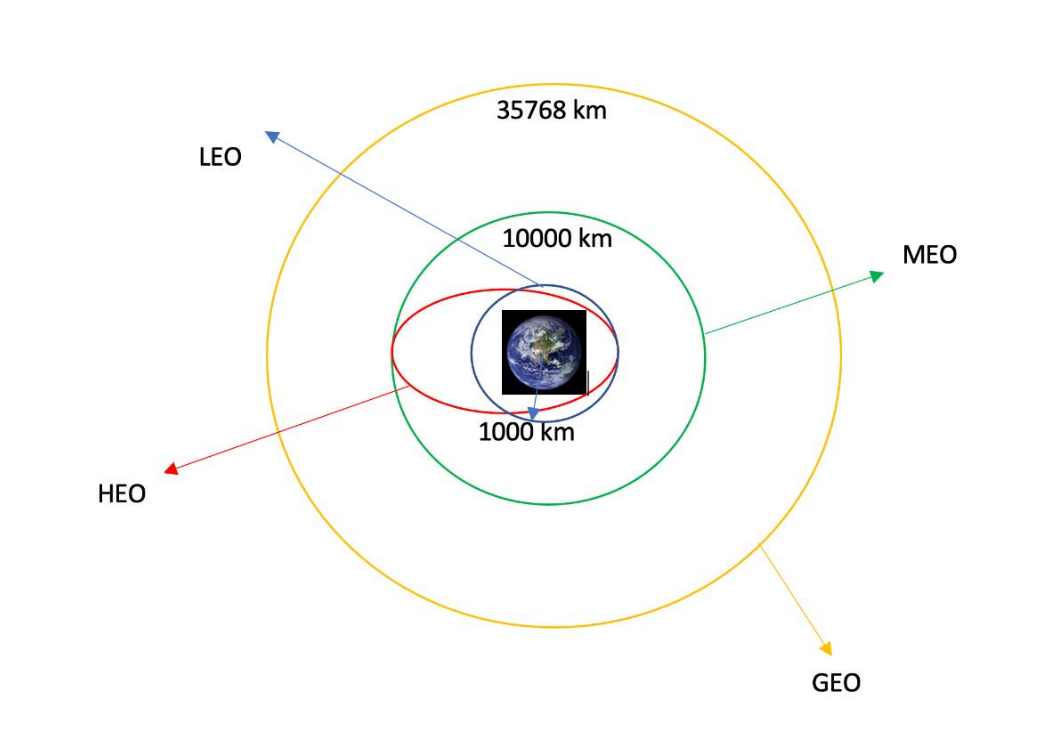
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
```

In [14]:

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df_no_customer = df[df['Customer'].isna()].head(5)
df_no_customer
```

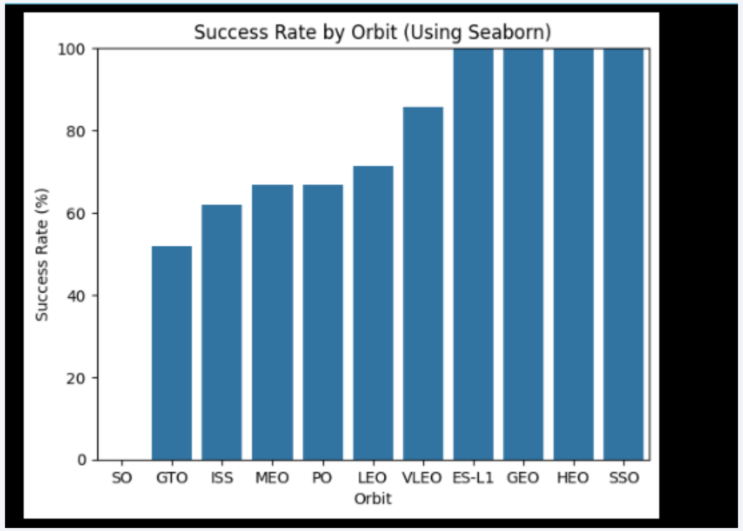
Data Wrangling

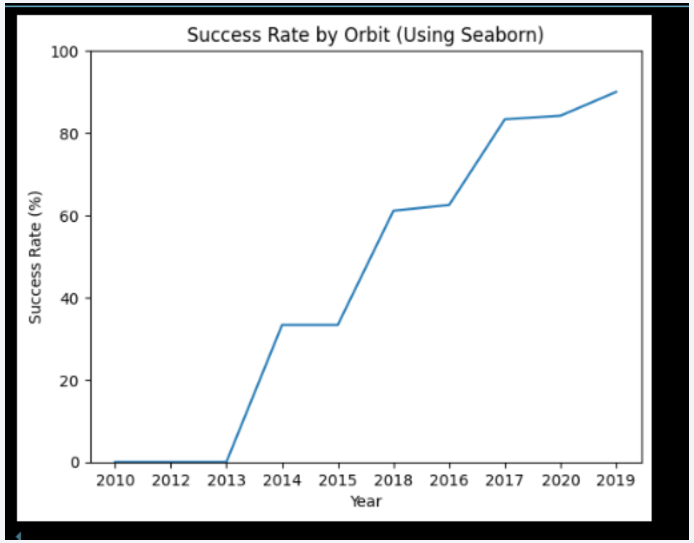
- Data Wrangling is to find some patterns in the data and determine what would be the label for training supervised models.
- Calculate
  - The number of launches on each site
  - The number and occurrence of each orbit.
  - The number and occurrence of mission outco,e of the orbits
    - Convert the landing outcome from Outcome column
- The calculation result is saved in csv file.
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- We explored the data
  - Scatter charts for the relationship between flight number and launch Site, payload and launch site, Flight number and orbit type, and Payload and Orbit type.
  - Bar chart for success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
  - Line chart for launch success yearly trend
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/jupyter-labs-eda-dataviz.ipynb





EDA with SQL

- 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 succesful landing outcome in ground pad was acheived.
- 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
- List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
- 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.
- 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.
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Mark all launch sites with Circle and Marker on the map
- Mark the success (green) /failed (red) launches for each site on the map
- Calculate the distances between a launch site to its proximities and mark down a point on the closest railroad, coastline, City, Highway
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Add a dropdown list to enable Launch Site selection
- Add a pie chart to show the total successful launches count for all sites
- Add a callback function to display success rate for the selected launch site
- Add a slider to select payload range
- Add a scatter chart to show the correlation between payload and launch success
- Add a callback function to show the correlation between payload and launch success on the scatter chart based on the launch site dropdown list and a payload slider inputs
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Read data to pandas dataframe and numpy, standardize and transform the data, split the data into training and testing data.
- Create Logistic Regression, Support Vector Machine, Tree Classifier, KNN Models
  - Each model is trained using same training data
  - The model is evaluated by accuracy using the same test data data
- Best model is DecisionTree with a score of 0.8767857142857143.
- https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

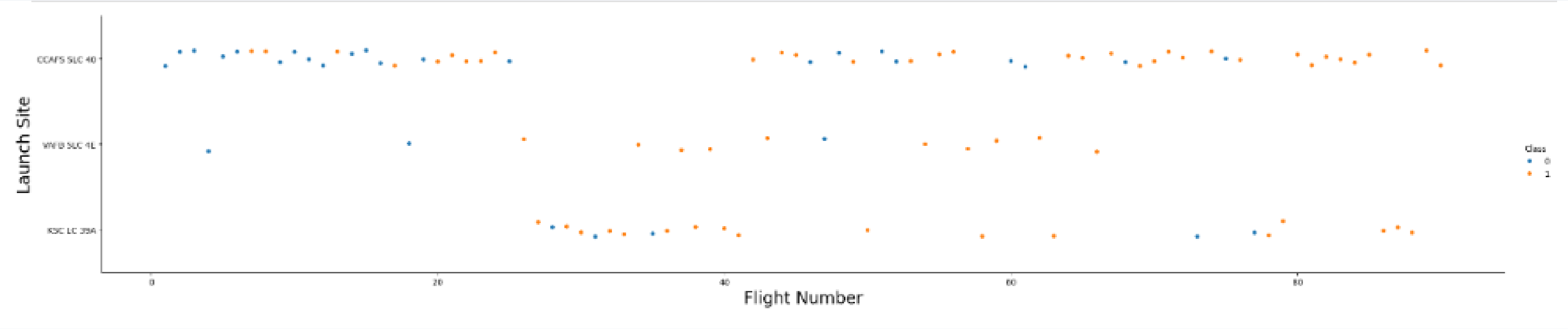
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

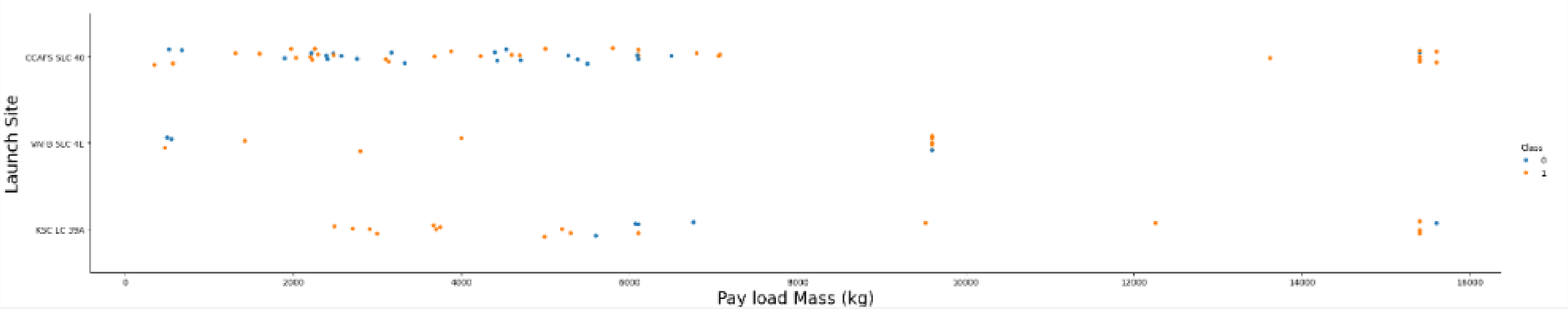
Flight Number vs. Launch Site

- At each launch site, higher flight number returns higher success late (Class = 1).



Payload vs. Launch Site

- Except VAFB-SLC, heavypayload mass(greater than 10000) return higher success rate.



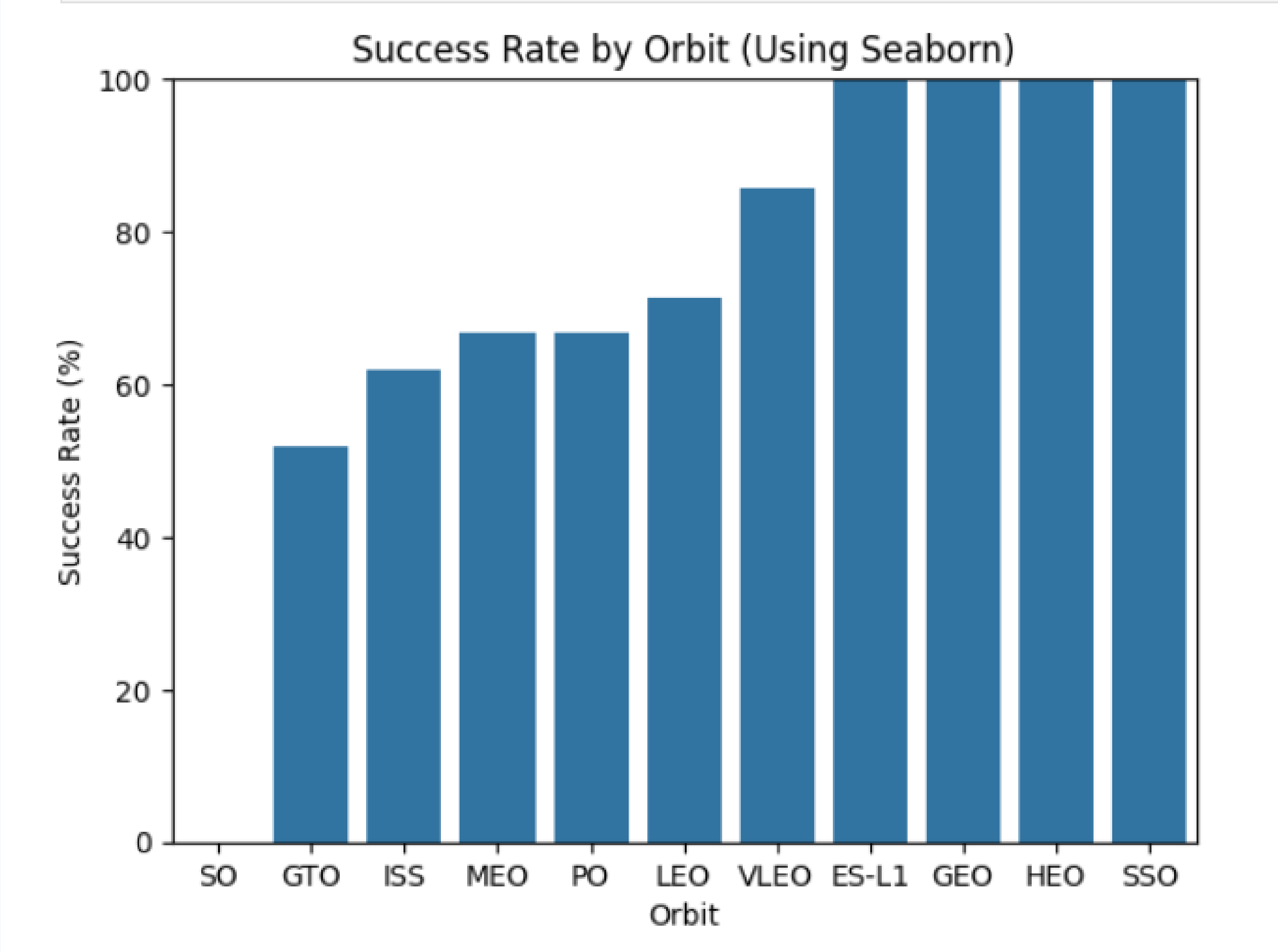
Success Rate vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations

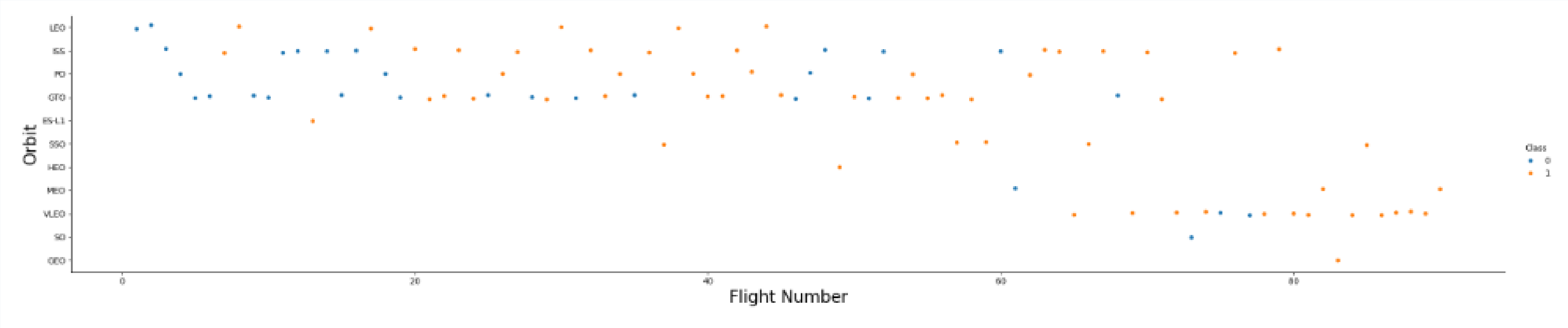
Flight Number vs. Orbit Type

- ES-L1, GEO, HEO, SSO have 100% success rate.



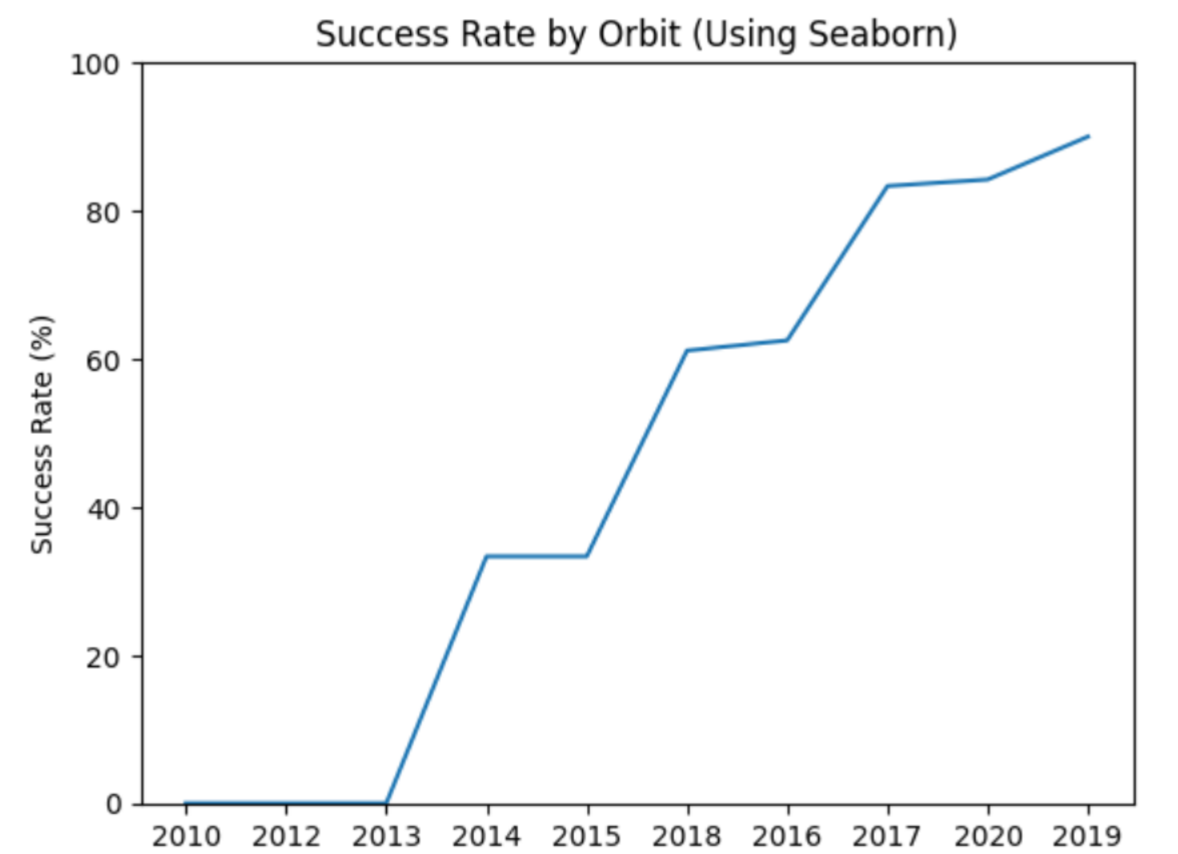
Payload vs. Orbit Type

- I cannot find co-relationship between Flight Number and Orbit Type. The success and failure are scatter evenly. 100% success rate orbits (ES-L1, GEO, HEO, SSO) have few data to find any relationship.



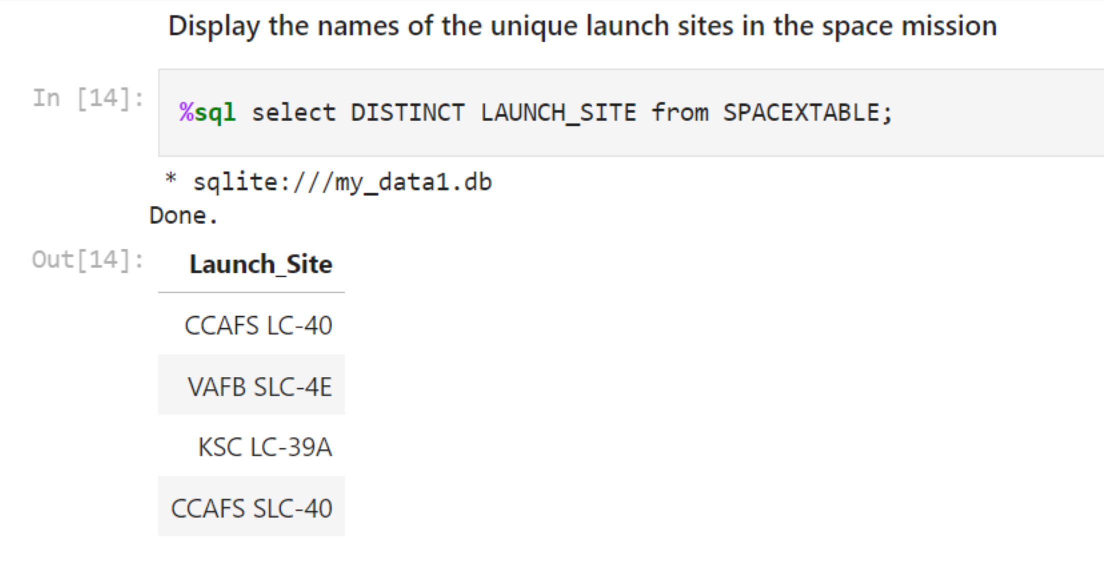
Launch Success Yearly Trend

- Later year yields higher success rate.



All Launch Site Names

- Use sql magic to **SELECT DISTINCT** to return unique launch sites.



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

```
In [14]:  %sql select DISTINCT LAUNCH_SITE from SPACEXTABLE;
```

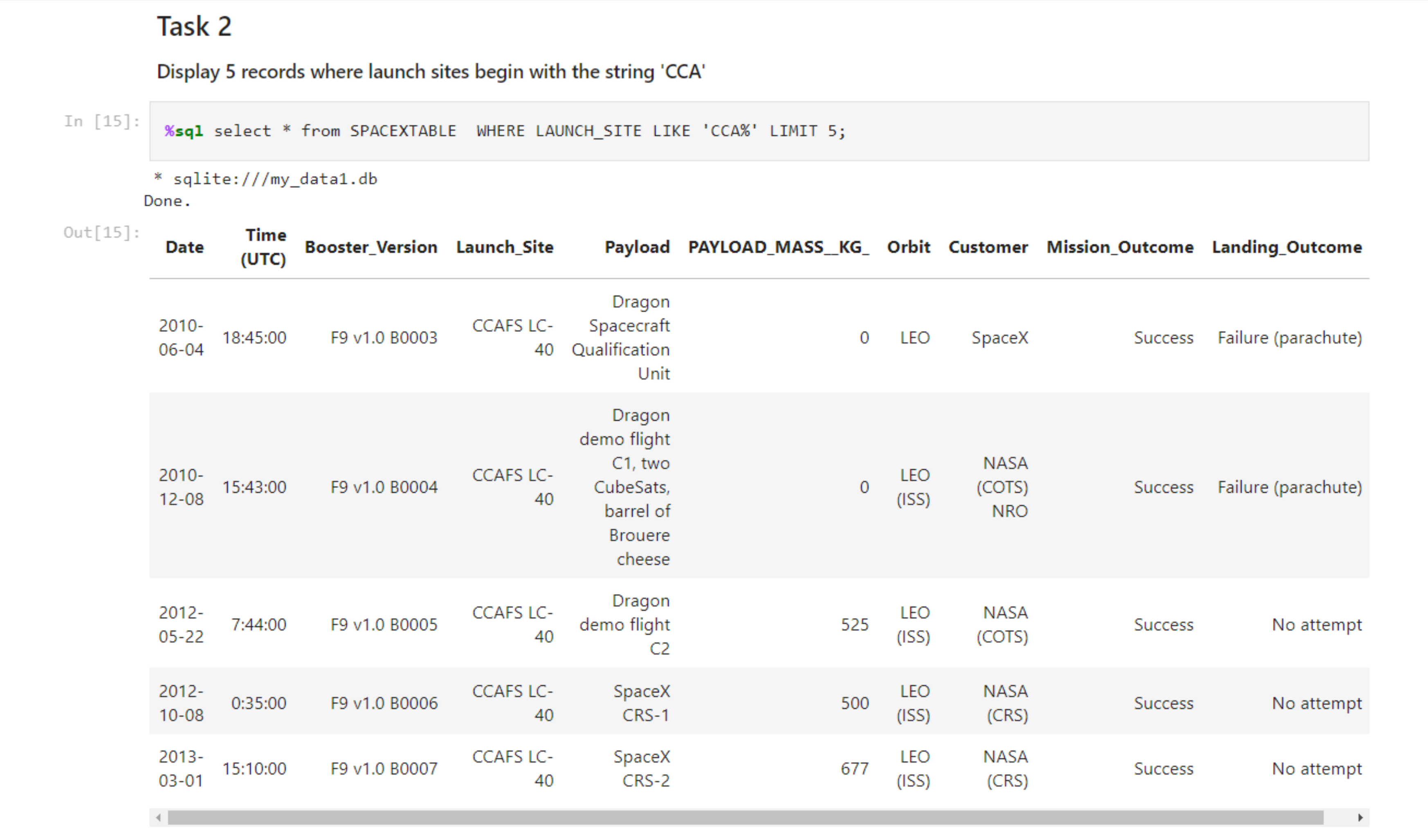
```
 * sqlite:///my_data1.db
Done.
```

Out[14]:

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

Launch Site Names Begin with 'CCA'

- Use sql magic to SELECT lanch sites which the name starts with ‘CCA’
- Use LIMIT 5 to display first 5 records.



Task 2

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

```
In [15]: %sql select * from SPACEXTABLE  WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

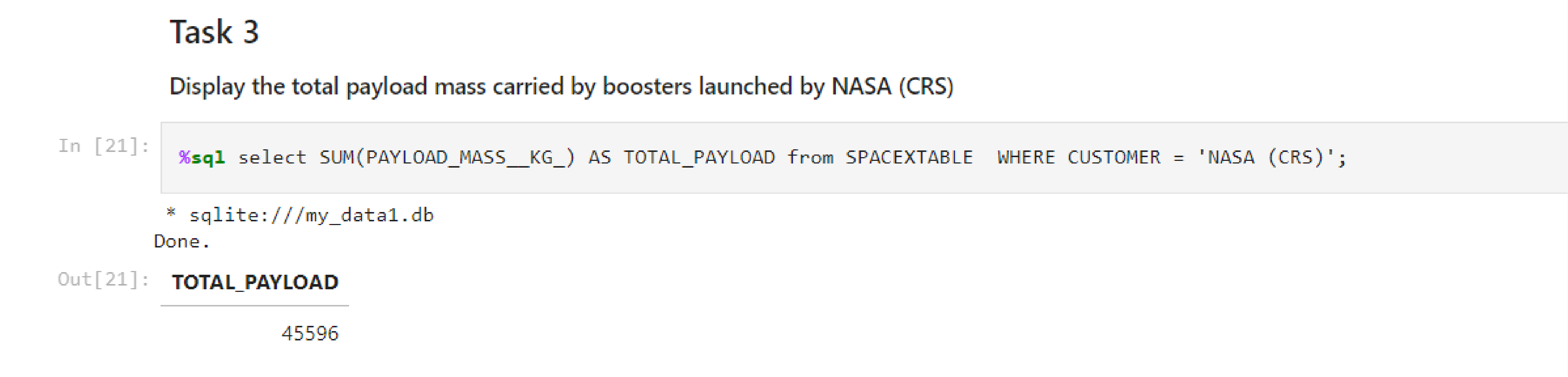
```
 * sqlite:///my_data1.db
Done.
```

Out[15]:

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

- Use SUM in SQL magic



Task 3

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

```
In [21]: %sql select SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD from SPACEXTABLE  WHERE CUSTOMER = 'NASA (CRS)';
```

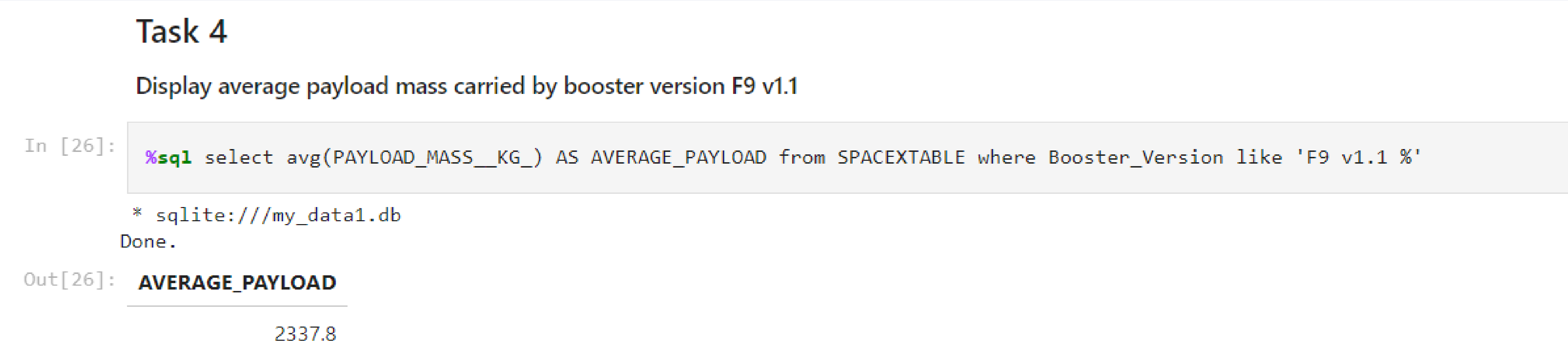
```
 * sqlite:///my_data1.db
Done.
```

Out[21]:

TOTAL_PAYLOAD
45596

Average Payload Mass by F9 v1.1

- Use AVG
- Use AS for remaining the value to AVERAGE_PAYLOAD



Task 4

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

```
In [26]: %sql select avg(PAYLOAD_MASS__KG_) AS AVERAGE_PAYLOAD from SPACEXTABLE where Booster_Version like 'F9 v1.1 %'
```

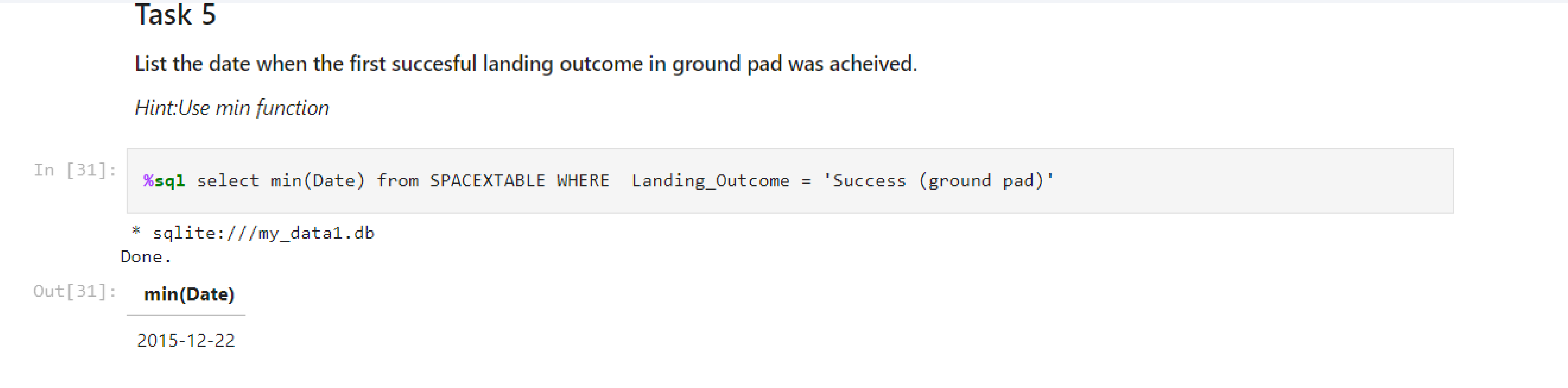
```
 * sqlite:///my_data1.db
Done.
```

Out[26]:

AVERAGE_PAYLOAD
2337.8

First Successful Ground Landing Date

- Use MIN to find minimum date with success launch on ground pad.



Task 5

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

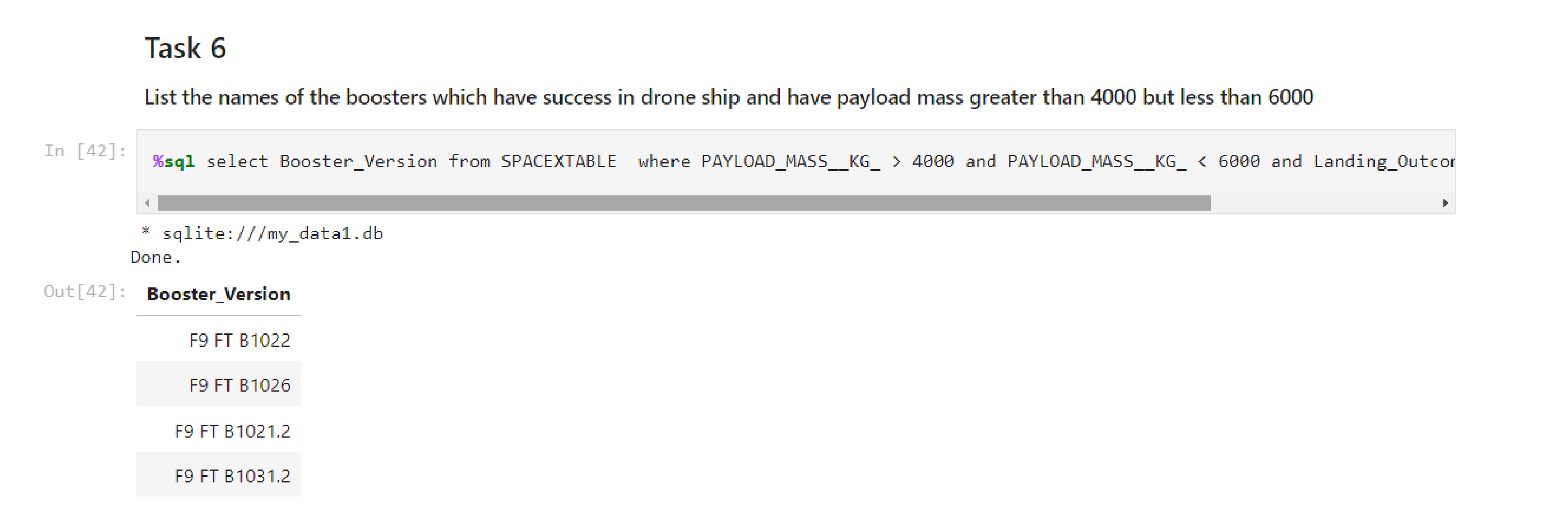
Hint:Use min function

```
In [31]: %sql select min(Date) from SPACEXTABLE WHERE  Landing_Outcome = 'Success (ground pad)'
```

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

```
Out[31]: min(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 [42]:

```
%sql select Booster_Version from SPACEXTABLE  where PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000 and Landing_Outcom
```

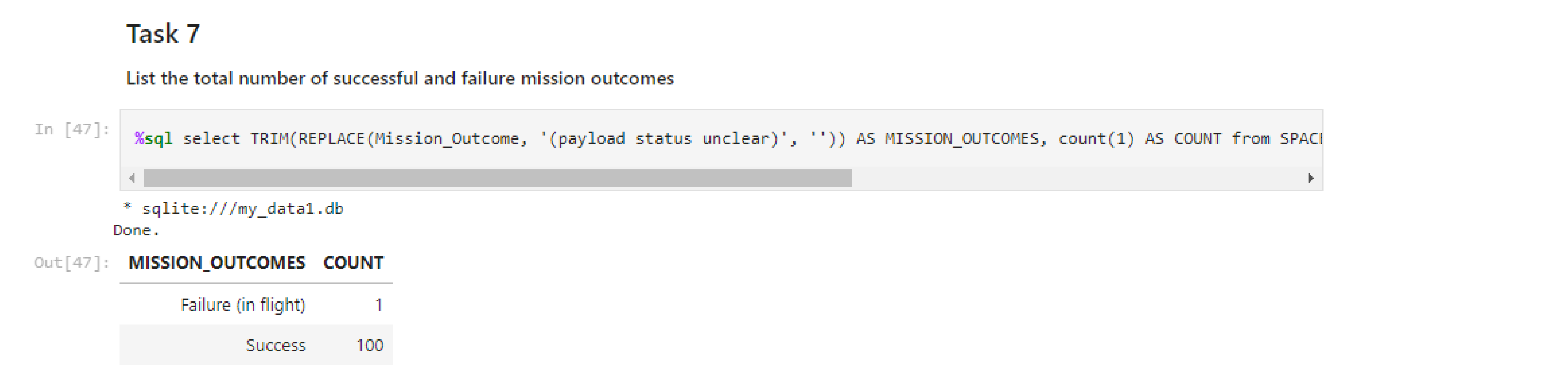
```
 * sqlite:///my_data1.db
Done.
```

Out[42]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- Use REPLACE to remove (payload status unclear)
- Use TRIM to remove extra whitespaces
- Use GROUP BY to group by MISSION_OUTCOMES
  - GROUP BY TRIM(REPLACE(Mission_Outcome, '(payload status unclear)', ''))



Task 7

List the total number of successful and failure mission outcomes

```
In [47]:  %sql select TRIM(REPLACE(Mission_Outcome, '(payload status unclear)', '')) AS MISSION_OUTCOMES, count(1) AS COUNT from SPACE
```

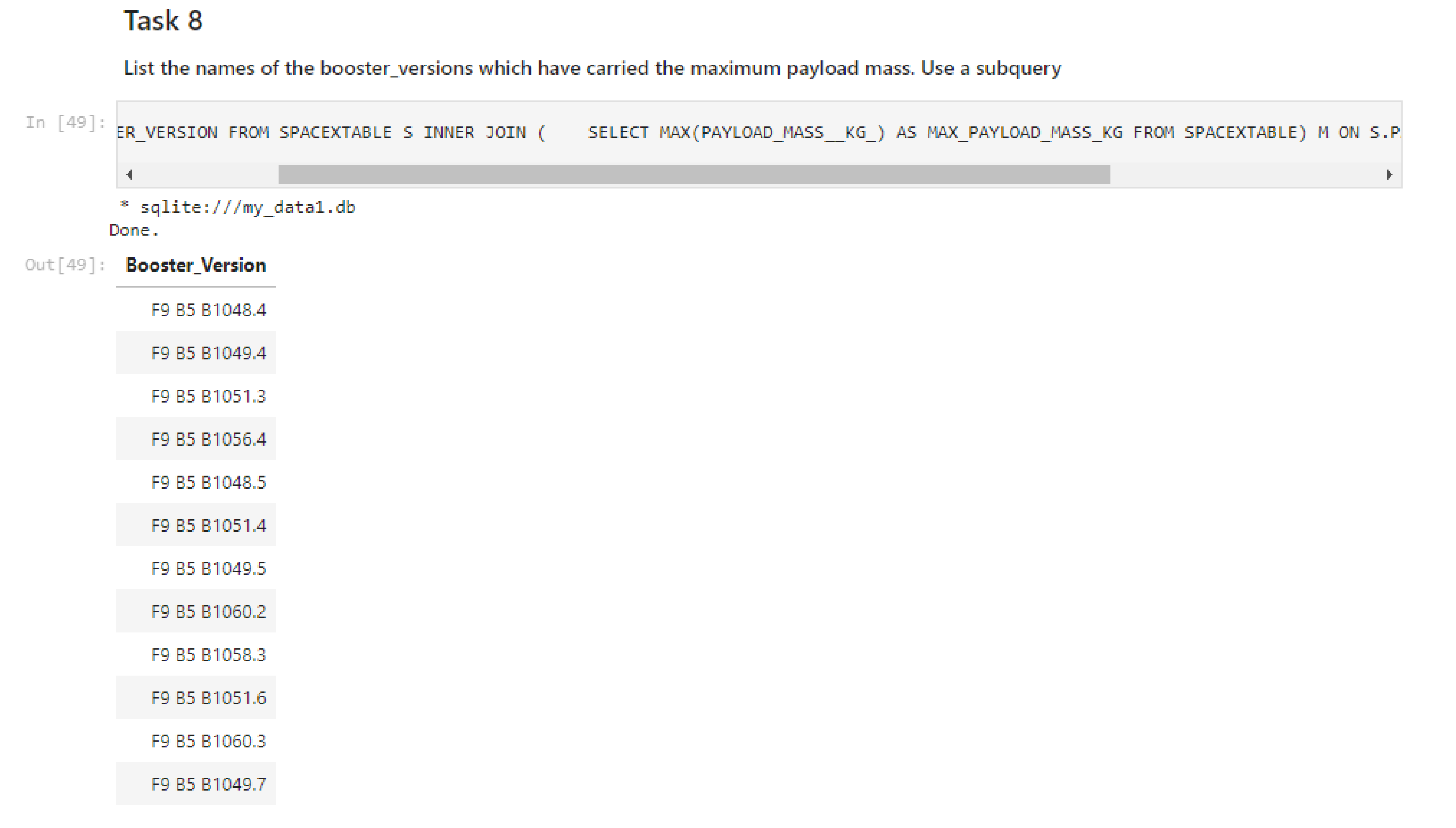
```
 * sqlite:///my_data1.db
Done.
```

Out[47]:

MISSION_OUTCOMES	COUNT
Failure (in flight)	1
Success	100

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

In [49]:

```
ER_VERSION FROM SPACEXTABLE S INNER JOIN (    SELECT MAX(PAYLOAD_MASS__KG_) AS MAX_PAYLOAD_MASS_KG FROM SPACEXTABLE) M ON S.P
```

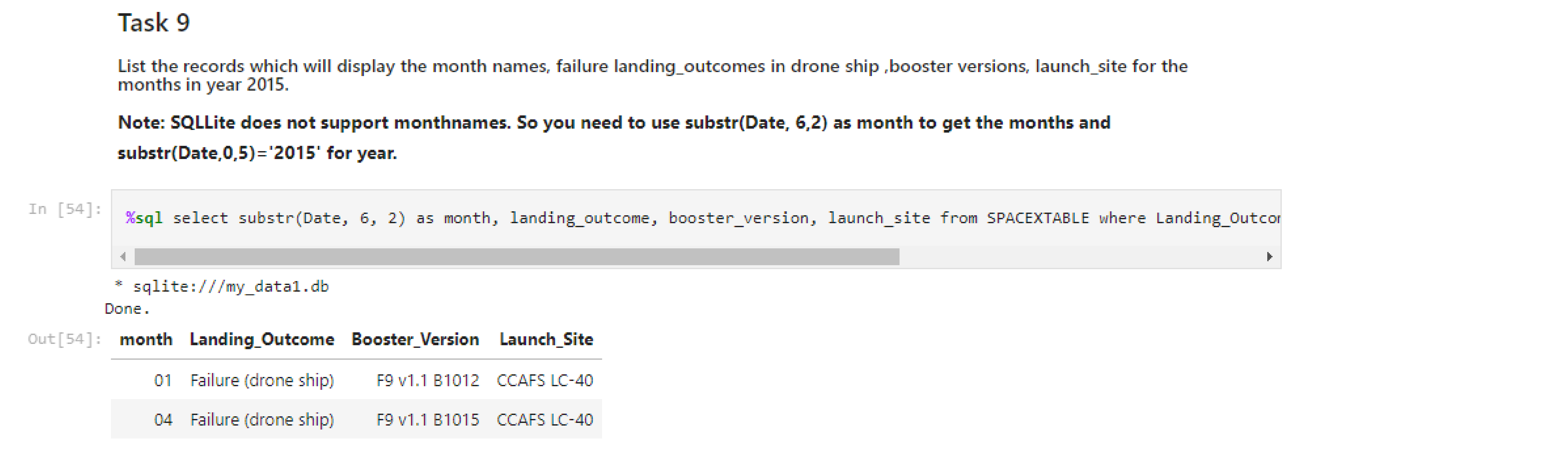
```
 * sqlite:///my_data1.db
Done.
```

Out[49]:

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

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 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 [54]: %sql select substr(Date, 6, 2) as month, landing_outcome, booster_version, launch_site from SPACEXTABLE where Landing_Outcor
```

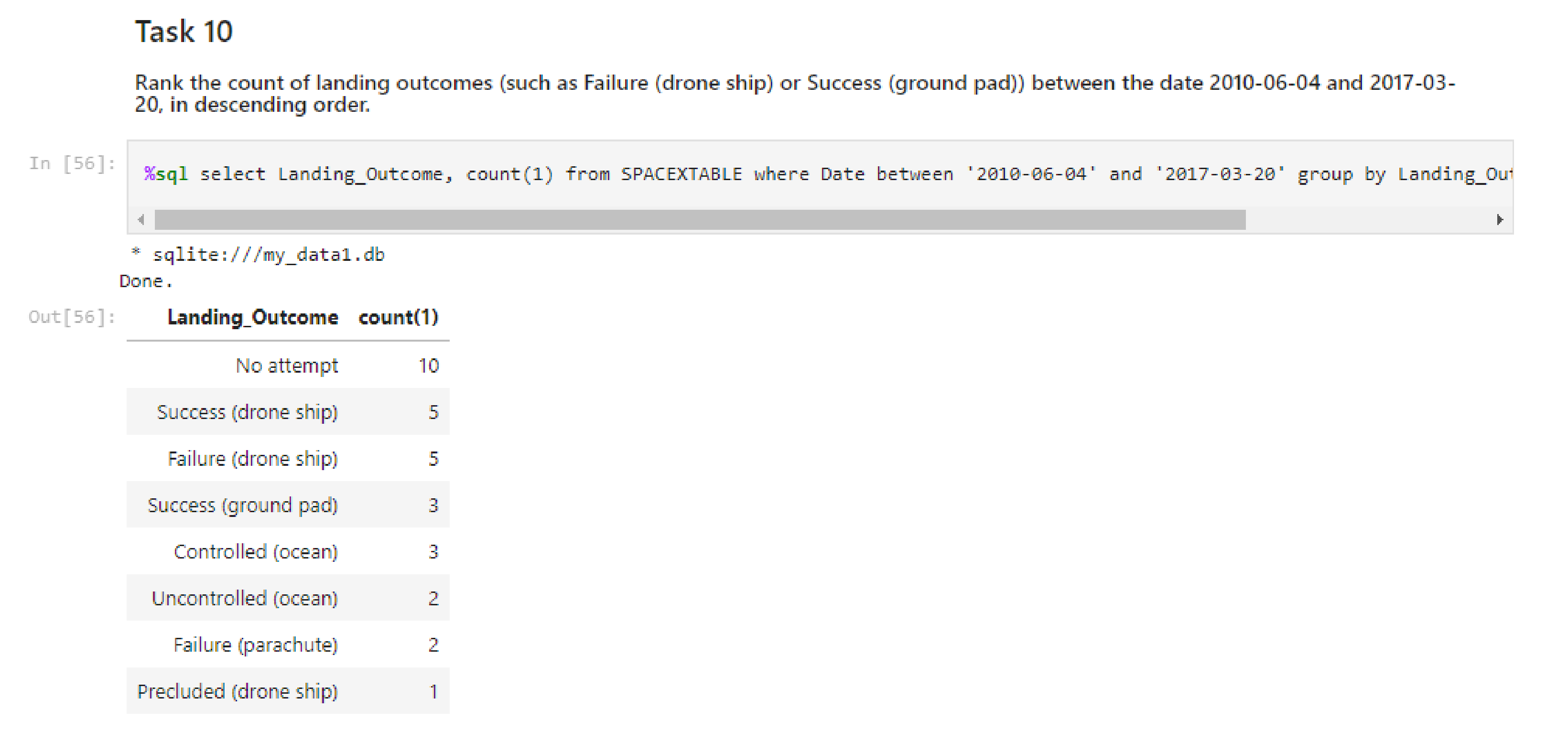
```
 * sqlite:///my_data1.db
Done.
```

Out[54]:

month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	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 2010-06-04 and 2017-03-20, in descending order.

In [56]:

```
%sql select Landing_Outcome, count(1) from SPACEXTABLE where Date between '2010-06-04' and '2017-03-20' group by Landing_Out
```

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

Out[56]:

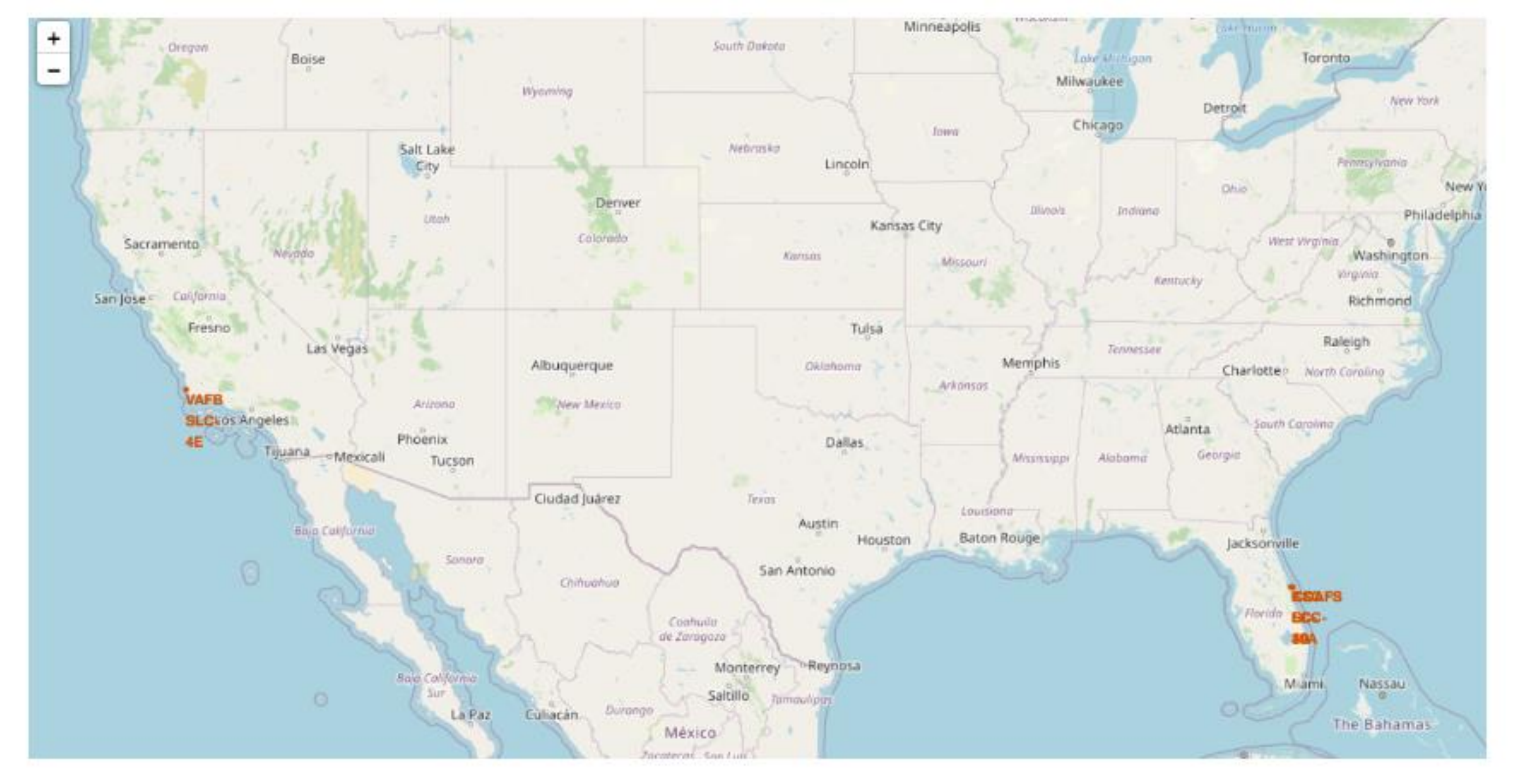
Landing_Outcome	count(1)
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

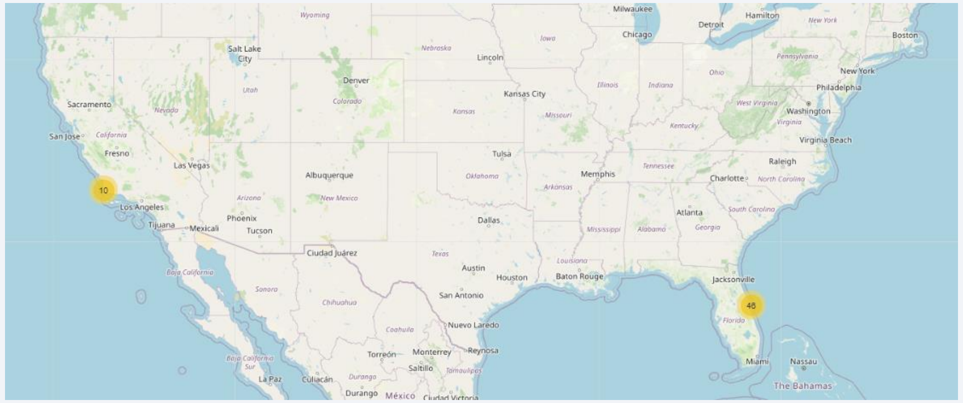
All launch sites

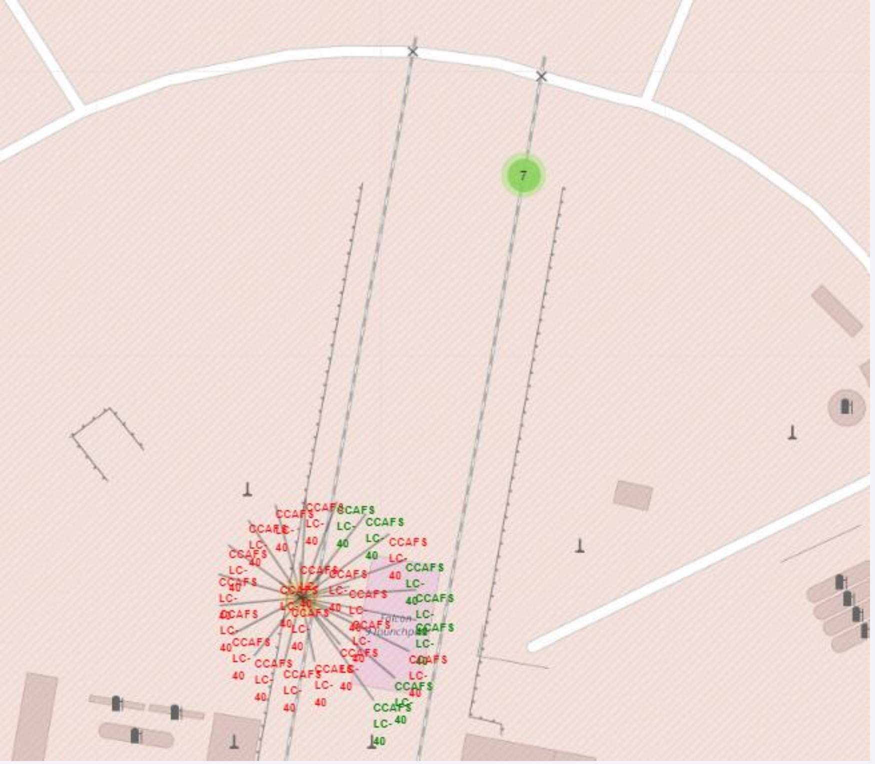
- All launch sites are located on US east and west coasts and near Equator line.



Mark the success/failed launches for each site on the map

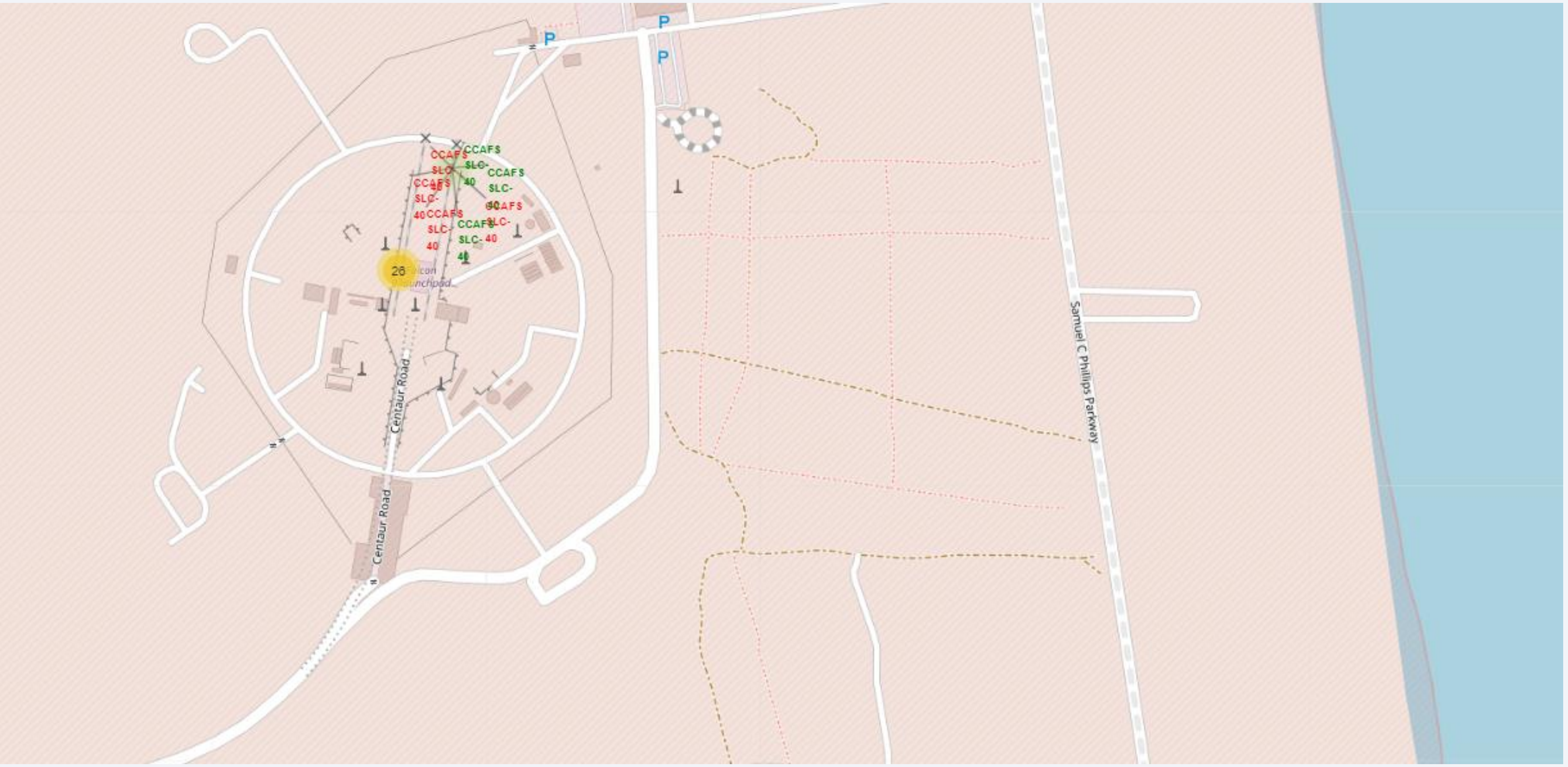
- GREEN marks successful launch
- RED marks failed launch





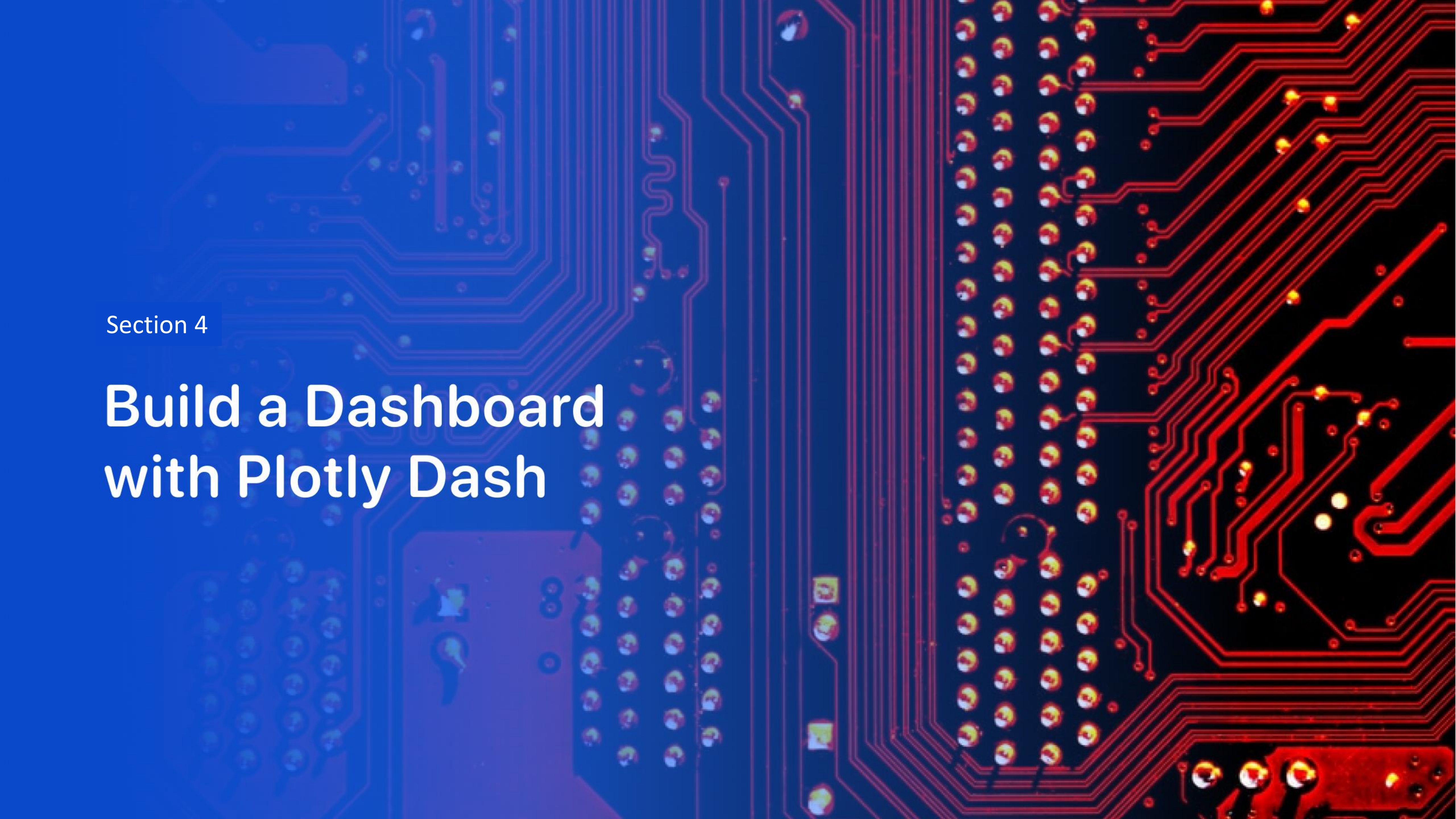
Calculate the distances between a launch site to its proximities

- No railway: No
- Near Highway: No
- Near coastline: Yes
- Near City: 78KM



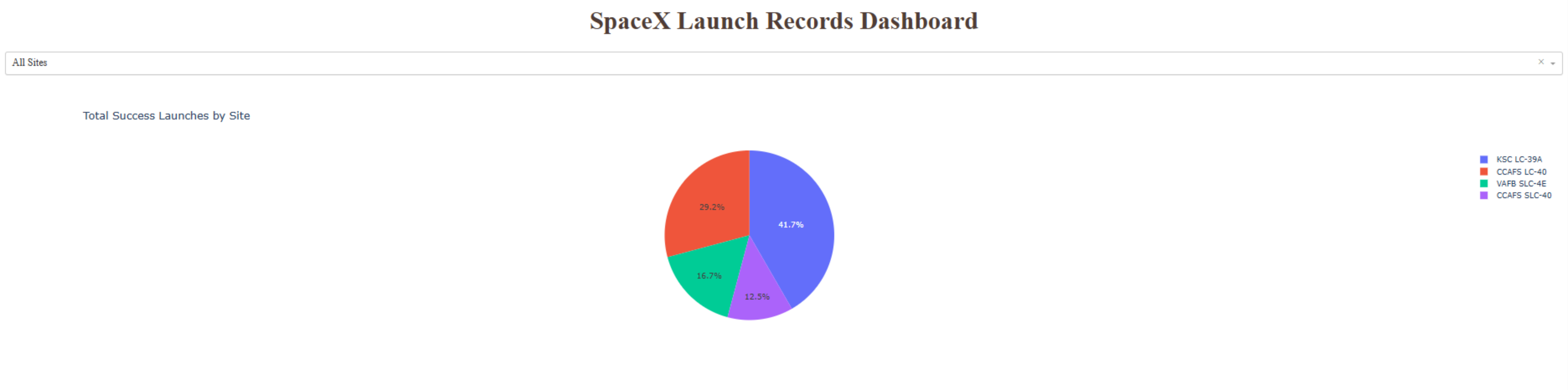
Section 4

Build a Dashboard with Plotly Dash



Total Success Launches by Site

- KSC LC-39A has the most successful launches
- CCAFS SLF-40 has the least successful launches



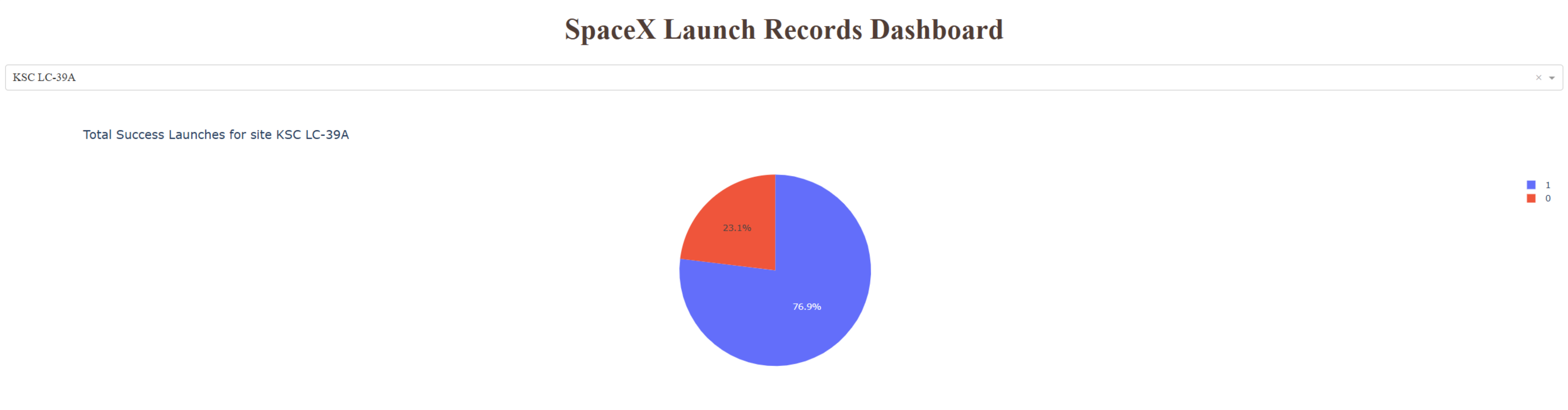
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by Site

Total Success Launches for site KSC LC-39A

- Explain the important elements and findings on the screenshot
- KSC LC-39A has the highest success rate (76.9%) compared to other sites.



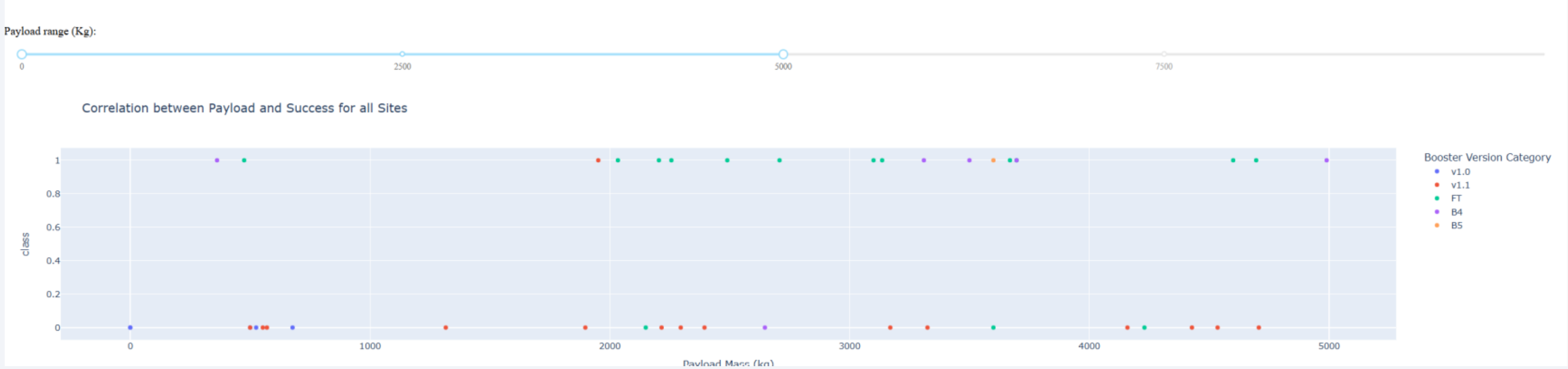
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for site KSC LC-39A

Correlation between Payload and Success for all Sites

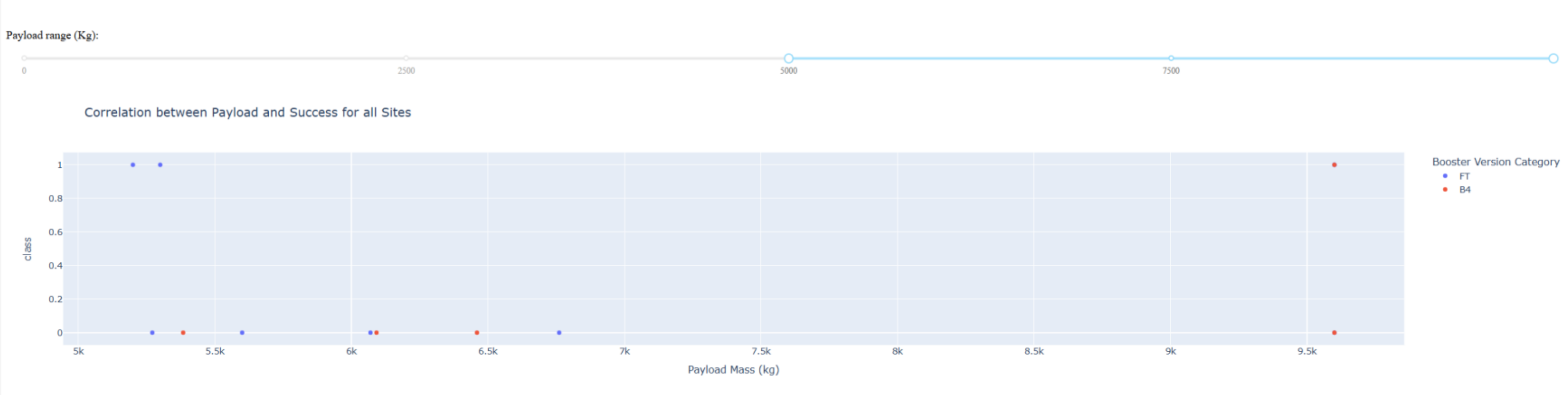
0 ~ 5K KG Payload Mass



Payload range (Kg):

Lower Payload Mass yields higher success rate.

5K ~ 10K KG Payload Mass



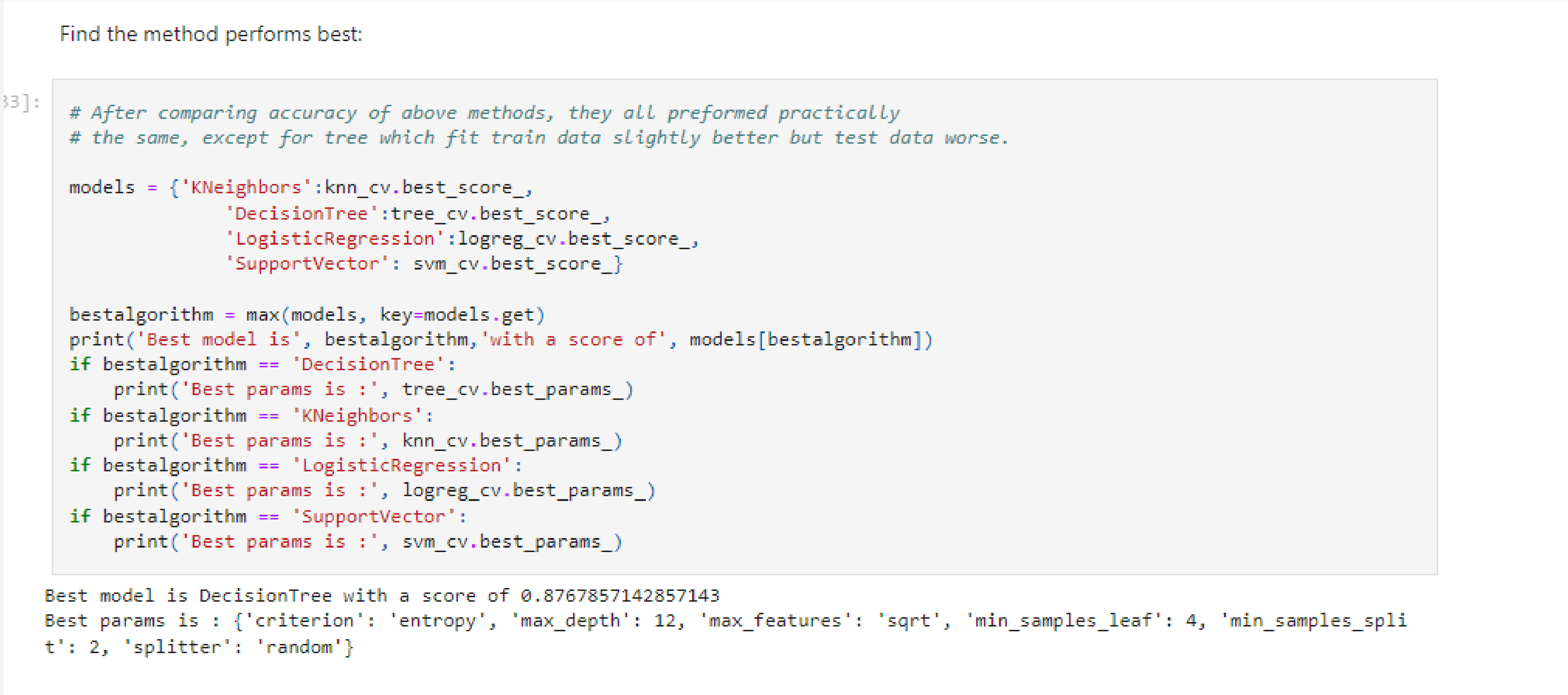
Payload range (Kg):

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Decision Tree classifier has the highest classification accuracy.



Find the method performs best:

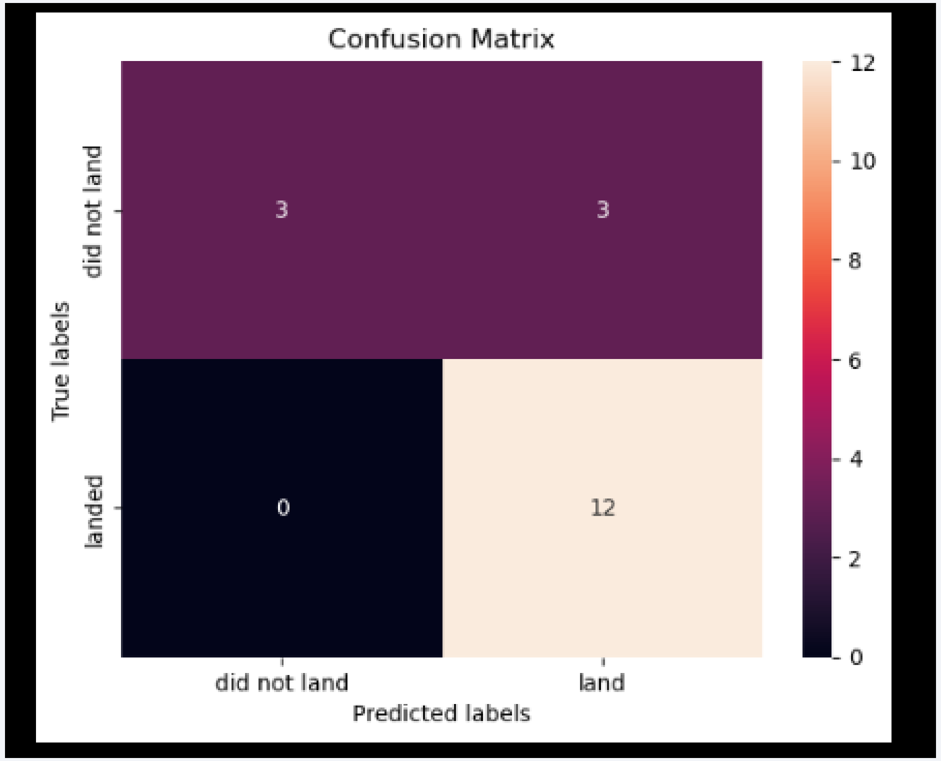
33]:

```
# After comparing accuracy of above methods, they all preformed practically
# the same, except for tree which fit train data slightly better but test data worse.

models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8767857142857143
Best params is : {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_spli
t': 2, 'splitter': 'random'}
```

Confusion Matrix

- Confusion Matrix of decision tree classifier
- The matrix are almost identical with other classifiers.



Conclusions

- Success rate increase over years.
- Highest success launch site has the highest success rate.
- Decision Tree has the highest classification accuracy.

Appendix

- GITHUB repository
  - https://github.com/austinleemv/IBM_DS_AppliedDataScienceCapstone/tree/main

Thank you!