IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

FABIO SANTOS

March 26, 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection API
  - Data Collection with Web Scraping
  - Exploratory Data Analysis (EDA) with SQL
  - Exploratory Data Analysis (EDA) with data visualization
  - Interactive Visual Analytics with Folium
  - Build an Interactive Dashboard with Ploty Dash
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result

Introduction

- Project background and context
  - Falcon 9 is a two-stage rocket designed and manufactured by SpaceX, an American aerospace manufacturer, headquartered in California, for the reliable and safe transport of satellites and the Dragon spacecraft into orbit.
  - Rockets from the Falcon 9 family have been launched 148 times, with 146 full mission successes, one partial failure and one total loss of spacecraft.
- Problems you want to find answers
  - What variables influences the rocket will land successfully?
  - What conditions to ensure the best successful landing rate?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Get data from SpaceX API
  - Web Scraping page “List of Falcon 9 and Falcon Heavy launches” from Wikipedia
- Perform data wrangling
  - Filter and dealing with missing Values
  - Prepare the Data to a binary classification

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Identify the best prediction accuracy from Hyperparameter for SVM, Classification Trees and Logistic Regression models

Data Collection

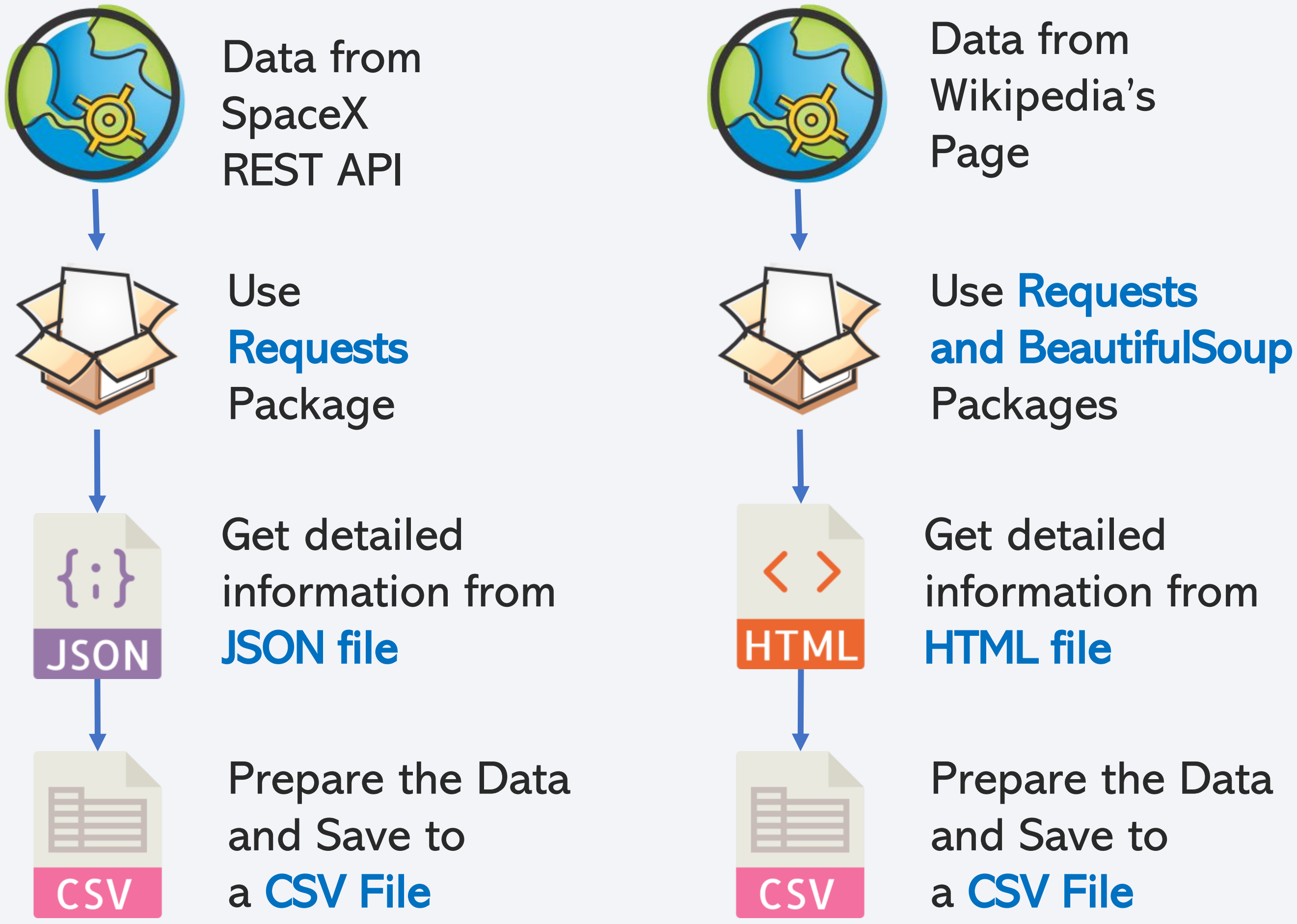
How data sets were collected.

Get columns from **SpaceX REST API** Data:

FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude

Get columns from **Wikipedia's Page** Data:

Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time



Data from SpaceX REST API

Use **Requests** Package

Get detailed information from **JSON file**

Prepare the Data and Save to a **CSV File**

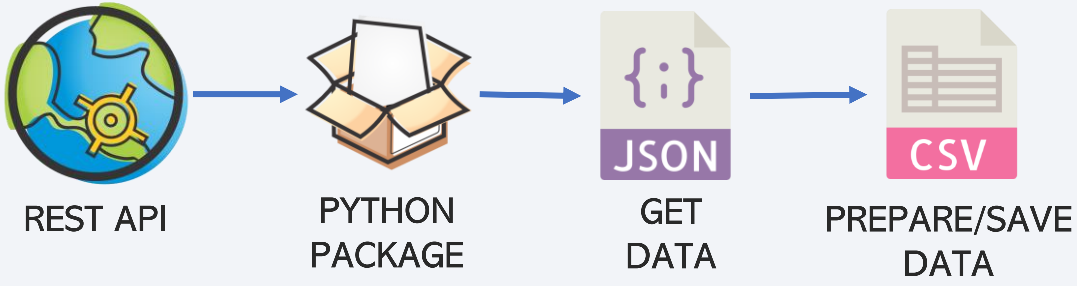
Data from Wikipedia's Page

Use **Requests and BeautifulSoup** Packages

Get detailed information from **HTML file**

Prepare the Data and Save to a **CSV File**

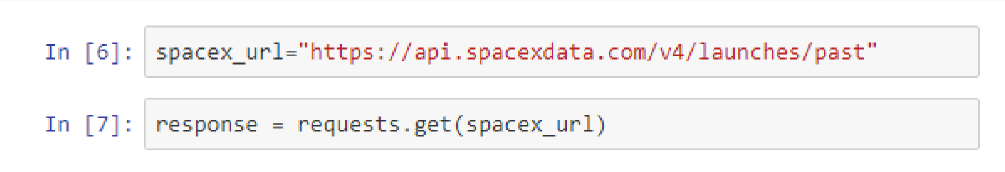
Data Collection – SpaceX API



REST API → PYTHON PACKAGE → GET DATA → PREPARE/SAVE DATA

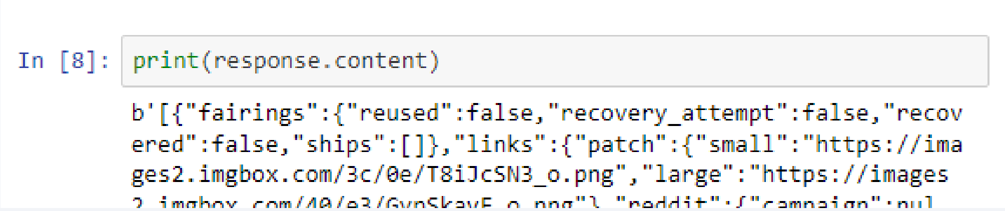
GitHub URL: https://github.com/fabioms-br/ibm-datascience/blob/main/01%20Data%20Collection%20API.ipynb

1. Request rocket launch data from SpaceX API with the following URL and GET request:



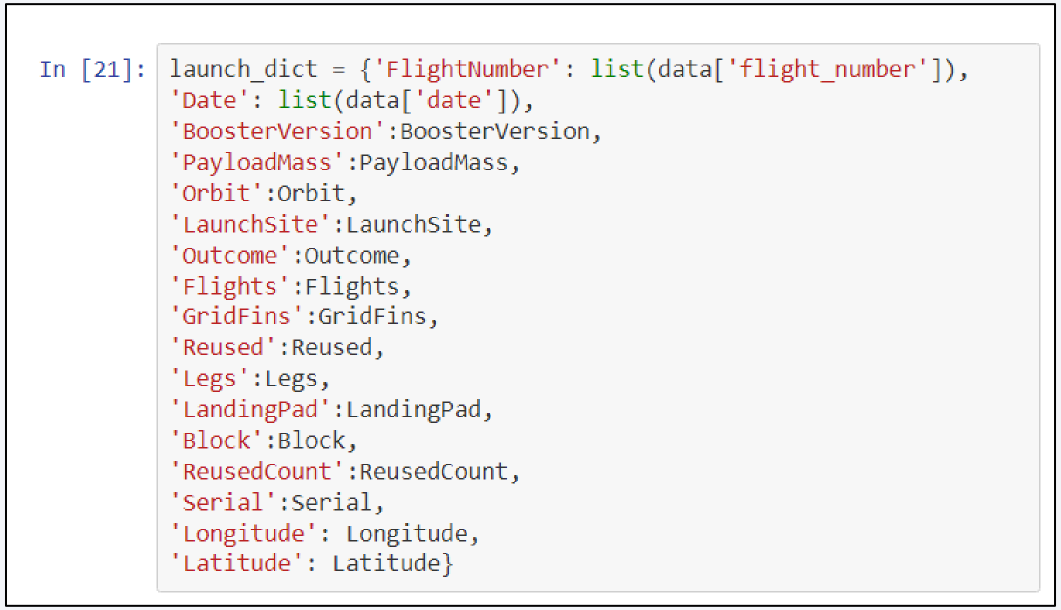
```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

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

2. Check the content of the response



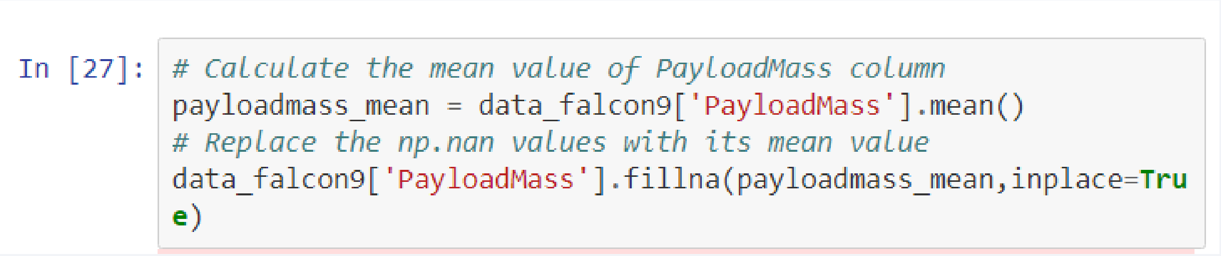
```
In [8]: print(response.content)

        b'[{"fairings":{"reused":false,"recovery_attempt":false,"recov
        ered":false,"ships":[]},"links":{"patch":{"small":"https://ima
        ges2.imgbox.com/3c/0e/T8iJcSN3_o.png","large":"https://images
        2.imgbox.com/40/e3/GvnSkayF_o.png"},"reddit":{"campaign":nul
```

3. Combine the columns into a dictionary.



```
In [21]: launch_dict = {'FlightNumber': list(data['flight_number']),
         'Date': list(data['date']),
         'BoosterVersion':BoosterVersion,
         'PayloadMass':PayloadMass,
         'Orbit':Orbit,
         'LaunchSite':LaunchSite,
         'Outcome':Outcome,
         'Flights':Flights,
         'GridFins':GridFins,
         'Reused':Reused,
         'Legs':Legs,
         'LandingPad':LandingPad,
         'Block':Block,
         'ReusedCount':ReusedCount,
         'Serial':Serial,
         'Longitude': Longitude,
         'Latitude': Latitude}
```

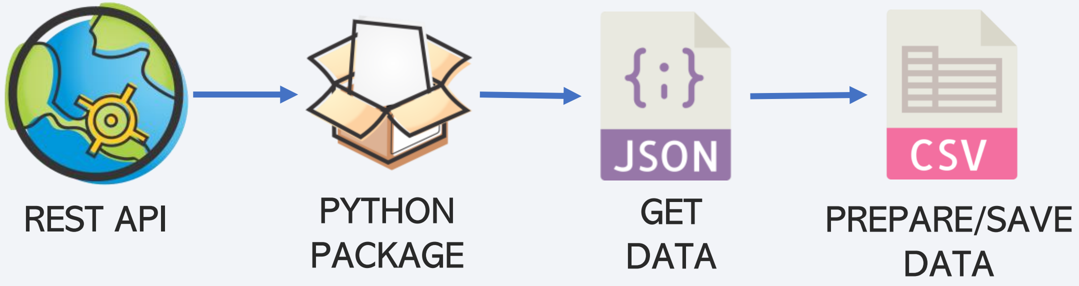
4. Dealing with Missing Values



```
In [27]: # Calculate the mean value of PayloadMass column
         payloadmass_mean = data_falcon9['PayloadMass'].mean()
         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'].fillna(payloadmass_mean,inplace=Tru
         e)
```

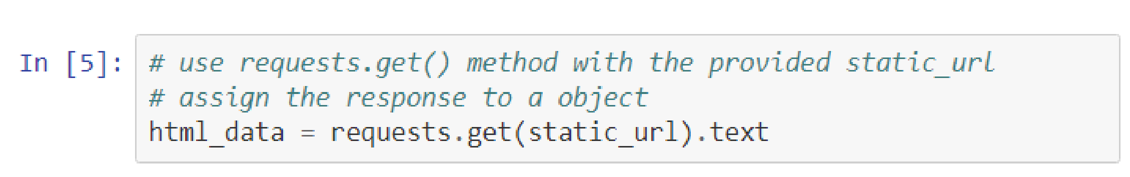
let's.

Data Collection - Scraping



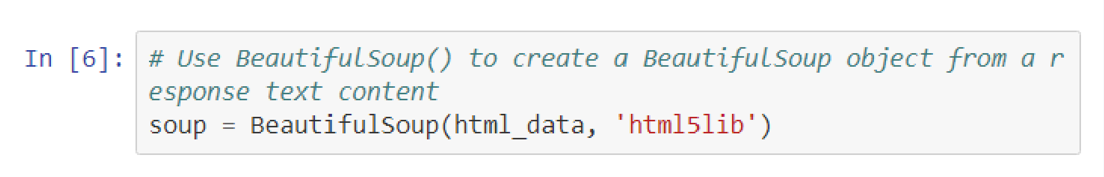
GitHub URL: https://github.com/fabioms-br/ibm-datascience/blob/main/02%20Data%20Collection%20with%20Web%20Scraping.ipynb

1. Perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response



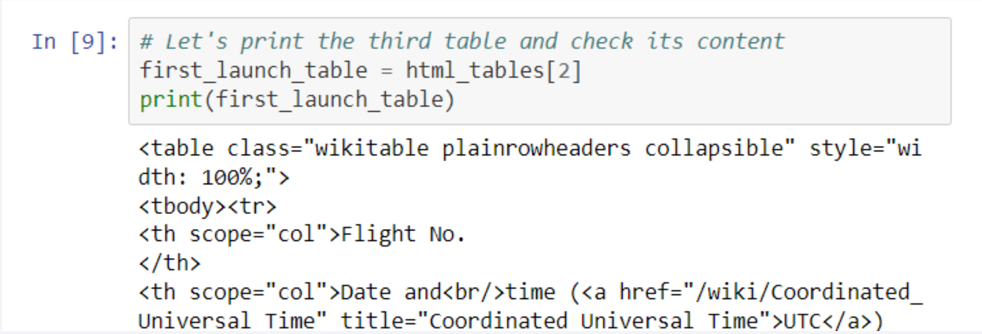
```
In [5]: # use requests.get() method with the provided static_url
        # assign the response to a object
        html_data = requests.get(static_url).text
```

2. Create a BeautifulSoup object from the HTML response



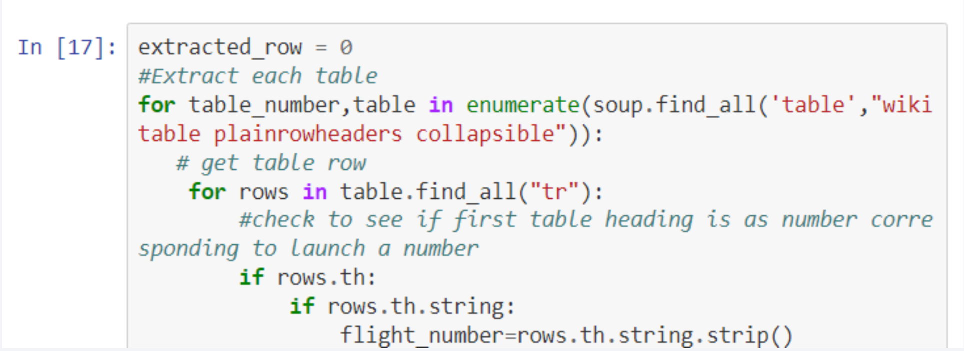
```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a r
        esponse text content
        soup = BeautifulSoup(html_data, 'html5lib')
```

3. Extract all column/variable names from the HTML table header



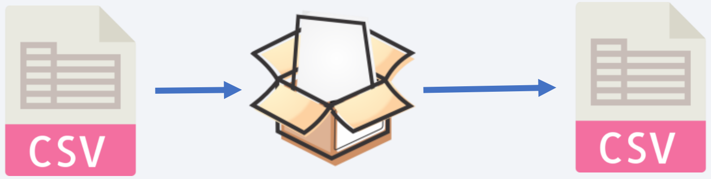
```
In [9]: # Let's print the third table and check its content
        first_launch_table = html_tables[2]
        print(first_launch_table)

        <table class="wikitable plainrowheaders collapsible" style="wi
        dth: 100%;">
        <tbody><tr>
        <th scope="col">Flight No.
        </th>
        <th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_
        Universal_Time" title="Coordinated Universal Time">UTC</a>)
```

4. Create a data frame by parsing the launch HTML tables



```
In [17]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(soup.find_all('table',"wiki
         table plainrowheaders collapsible")):
            # get table row
             for rows in table.find_all("tr"):
                 #check to see if first table heading is as number corre
         sponding to launch a number
                 if rows.th:
                     if rows.th.string:
                         flight_number=rows.th.string.strip()
```

Data Wrangling



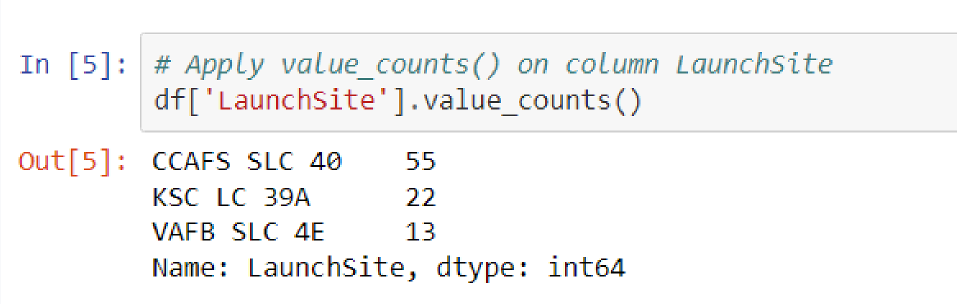
CSV DATASET

PYTHON PACKAGE

PREPARE/SAVE DATA

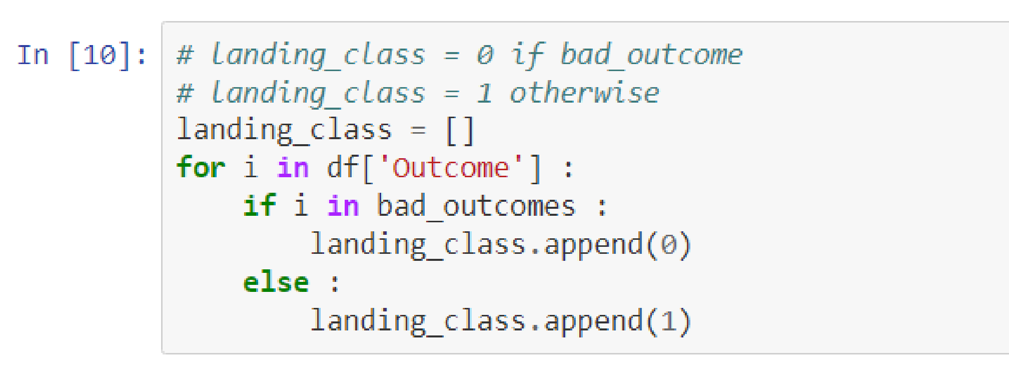
GitHub URL: https://github.com/fabioms-br/ibm-datascience/blob/main/03%20EDA.ipynb

1. Determine the number of launches on each site:



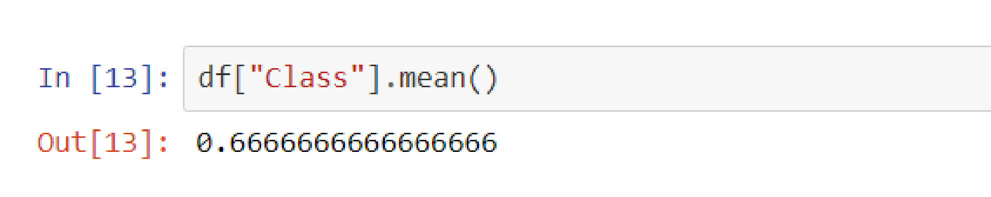
```
In [5]:  # Apply value_counts() on column LaunchSite
         df['LaunchSite'].value_counts()

Out[5]:  CCAFS SLC 40    55
         KSC LC 39A      22
         VAFB SLC 4E     13
         Name: LaunchSite, dtype: int64
```

2. Create a list from where the element is zero if the corresponding row in Outcome is in the set bad_outcome:



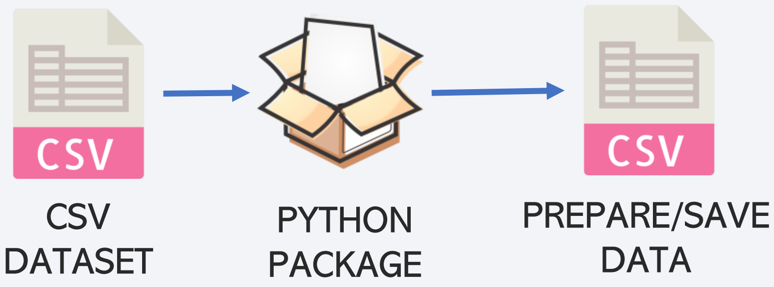
```
In [10]:  # landing_class = 0 if bad_outcome
          # landing_class = 1 otherwise
          landing_class = []
          for i in df['Outcome'] :
              if i in bad_outcomes :
                  landing_class.append(0)
              else :
                  landing_class.append(1)
```

3. We can use the following line of code to determine the success rate:



```
In [13]:  df["Class"].mean()

Out[13]:  0.6666666666666666
```

EDA with Data Visualization



CSV DATASET → PYTHON PACKAGE → PREPARE/SAVE DATA

GitHub URL: https://github.com/fabioms-br/ibm-datascience/blob/main/05%20EDA%20with%20Data%20Visualization.ipynb

Scatter chart:

- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Flight Number vs. Orbit Type
- Payload vs. Orbit Type

A scatter plot shows relationship between two variables

Bar chart:

- Orbit Type vs. Success Rate

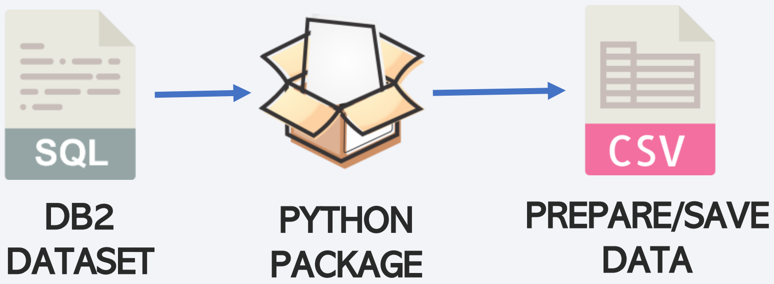
Easy to compare datasets with multiple groups where indicate the relationship between the two axes

Line chart:

- Year vs. Success Rate

Shows data variables and trends with the possibility of predict the results.

EDA with SQL



DB2 DATASET

PYTHON PACKAGE

PREPARE/SAVE DATA

GitHub URL: https://github.com/fabioms-br/ibm-datascience/blob/main/04%20EDA%20with%20SQL.ipynb

Answer following questions using executing SQL queries:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CCA'
- Displaying the total payload mass carried by boosters launched by NASA (CRS)
- Displaying average payload mass carried by booster version F9 v1.1
- Listing the date when the first successful landing outcome in ground pad was achieved
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- Listing the total number of successful and failure mission outcomes
- Listing the names of the booster_versions which have carried the maximum payload mass
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Ranking 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

Build an Interactive Map with Folium

Objects created and added to a folium map:

Map Object	Purpose
Markers	To show all launch sites on a map
Markers	To show the launch outcome for each site on the map
Mouse Position	To calculate the distances between a launch site to its proximities
Lines	To show the distances between a launch site to its proximities

GitHub URL:
https://github.com/fabioms-br/ibm-datascience/blob/main/06%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

Pie chart

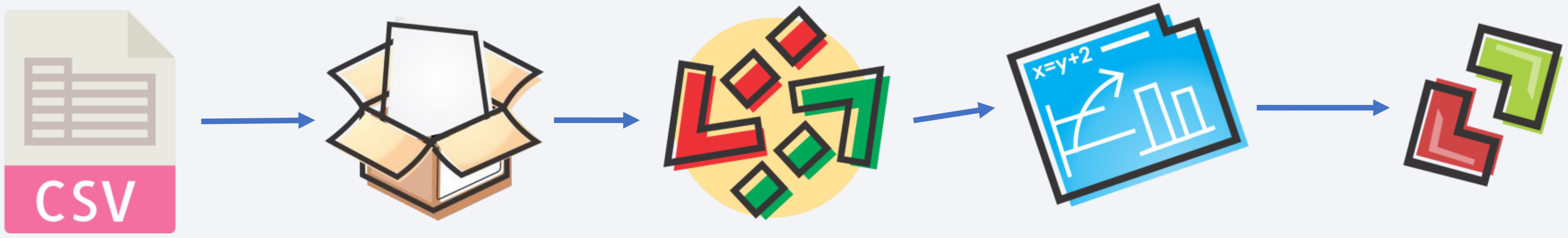
- To show total success launches by sites indicate the distribution between them
- An individual site can be selected to compare the success rate

Scatter chart

- To Show the relationship between the variables Outcomes and Payload mass (Kg) by different boosters
- Use of sliders to filter Payload mass between 0 and 10000 kg

GitHub URL:
https://github.com/fabioms-br/ibm-datascience/blob/main/07%20Build%20an%20Interactive%20Dashboard%20with%20Ploty%20Dash.py

Predictive Analysis (Classification)

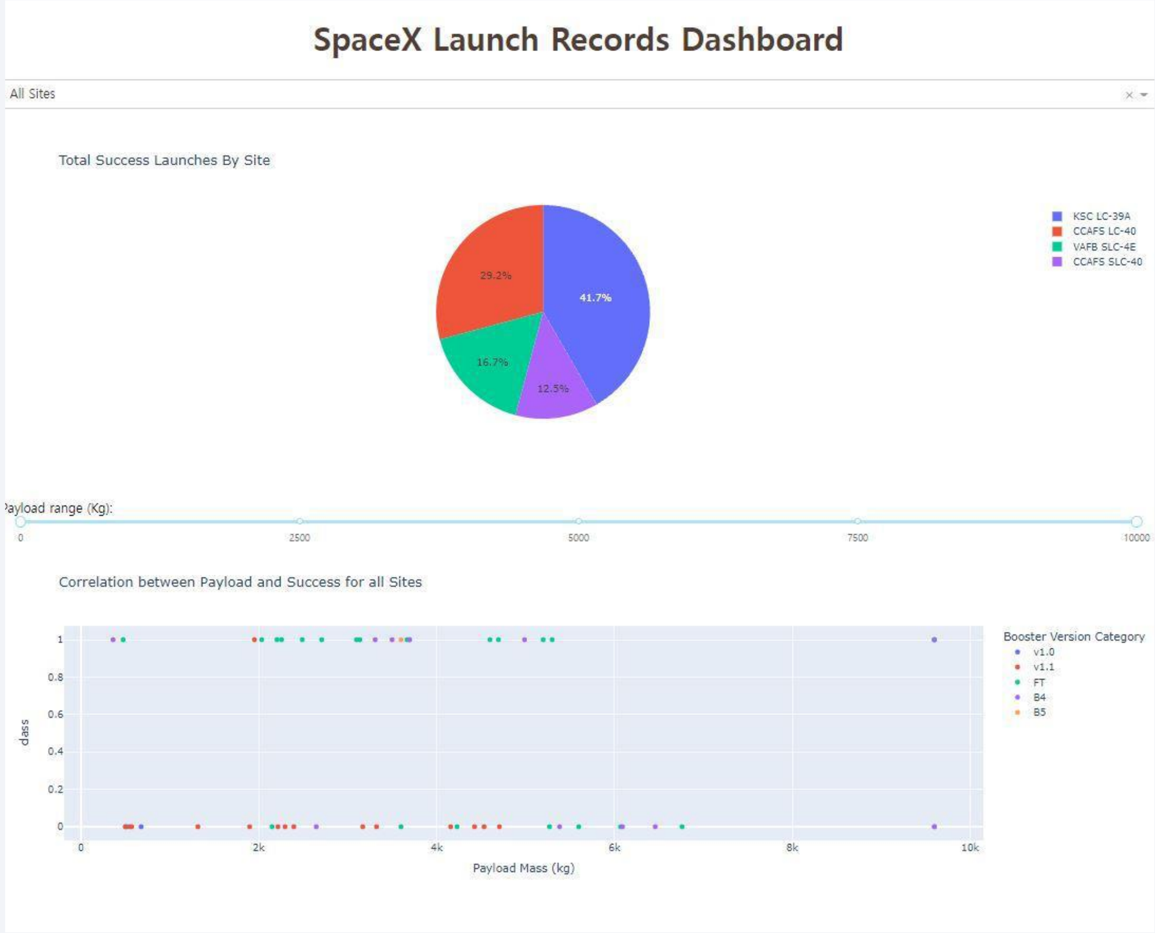


CSV DATASET → PYTHON PACKAGE → SPLIT DATA → EVALUATE MODEL → FIND BEST MODEL

- Get Data using Python packages
- Split Data into training and test
- Evaluate Support Vector Machine Model;
- Evaluate Classification Trees and Logistic Regression
- Find the best model accuracy

GitHub URL:
https://github.com/fabioms-br/ibm-datascience/blob/main/08%20Machine%20Learning%20Prediction.ipynb

Results



SpaceX Launch Records Dashboard

All Sites

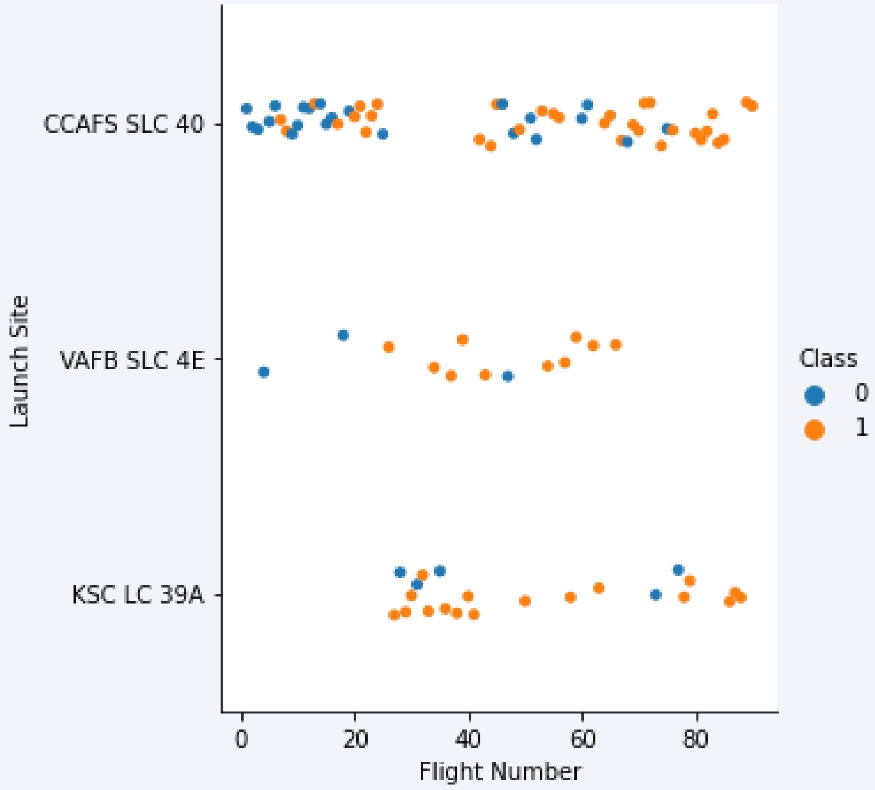
ayload range (Kg):

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

Section 2

Insights drawn from EDA

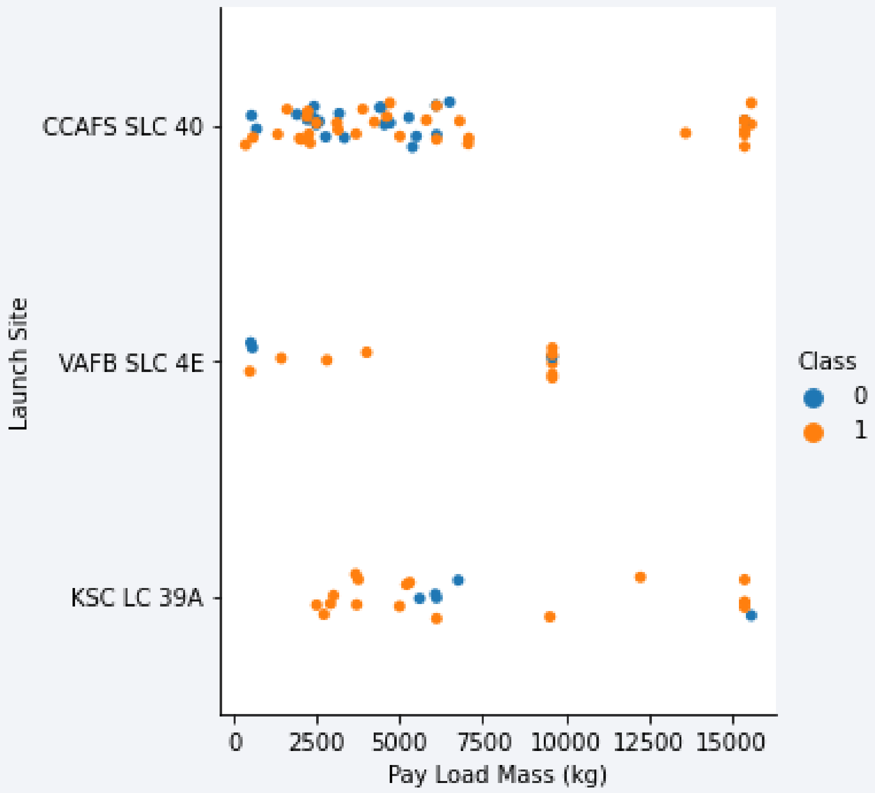
Flight Number vs. Launch Site



- Class 0 (blue) represents a not successful launch, and Class 1 (orange) represents a successful launch

- Success rate increases as flight number increases at all launching sites

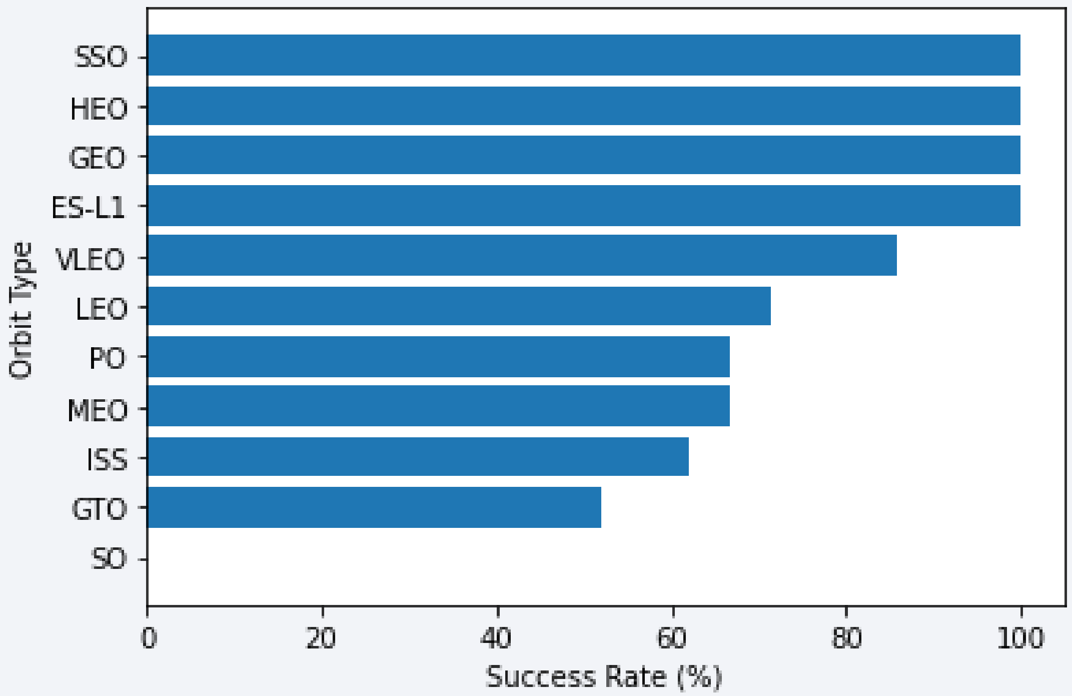
Payload vs. Launch Site



- Class 0 (blue) represents a not successful launch, and Class 1 (orange) represents a successful launch

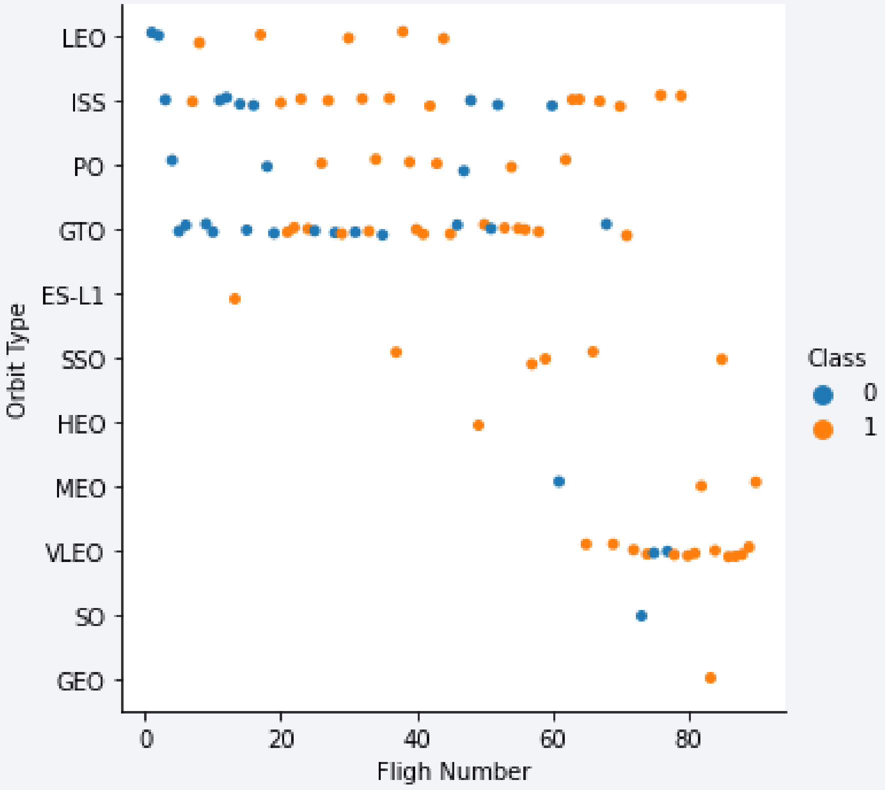
- The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket.

Success Rate vs. Orbit Type



- Chart represents the success rate of each orbit type;
- The highest value the better;
- Orbit types SSO, HEO, GEO, and ES-L1 have 100% success rates.

Flight Number vs. Orbit Type

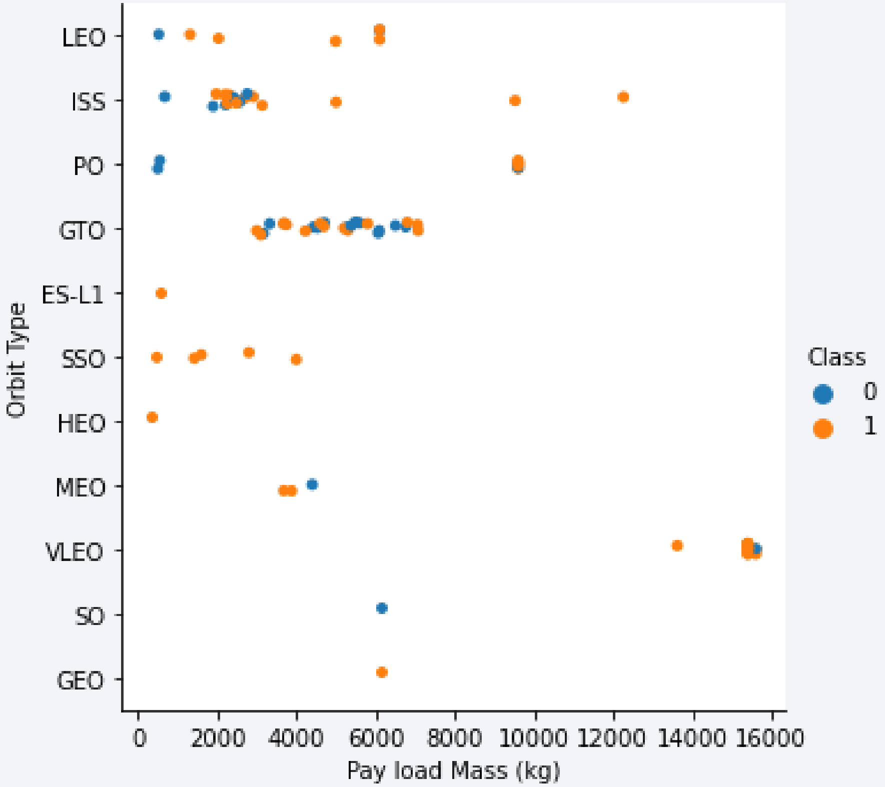


- Class 0 (blue) represents a not successful launch, and Class 1 (orange) represents a successful launch

- 5 launches from SSO flight were successful

- GTO success rate appears no relationship between flight numbers

Payload vs. Orbit Type

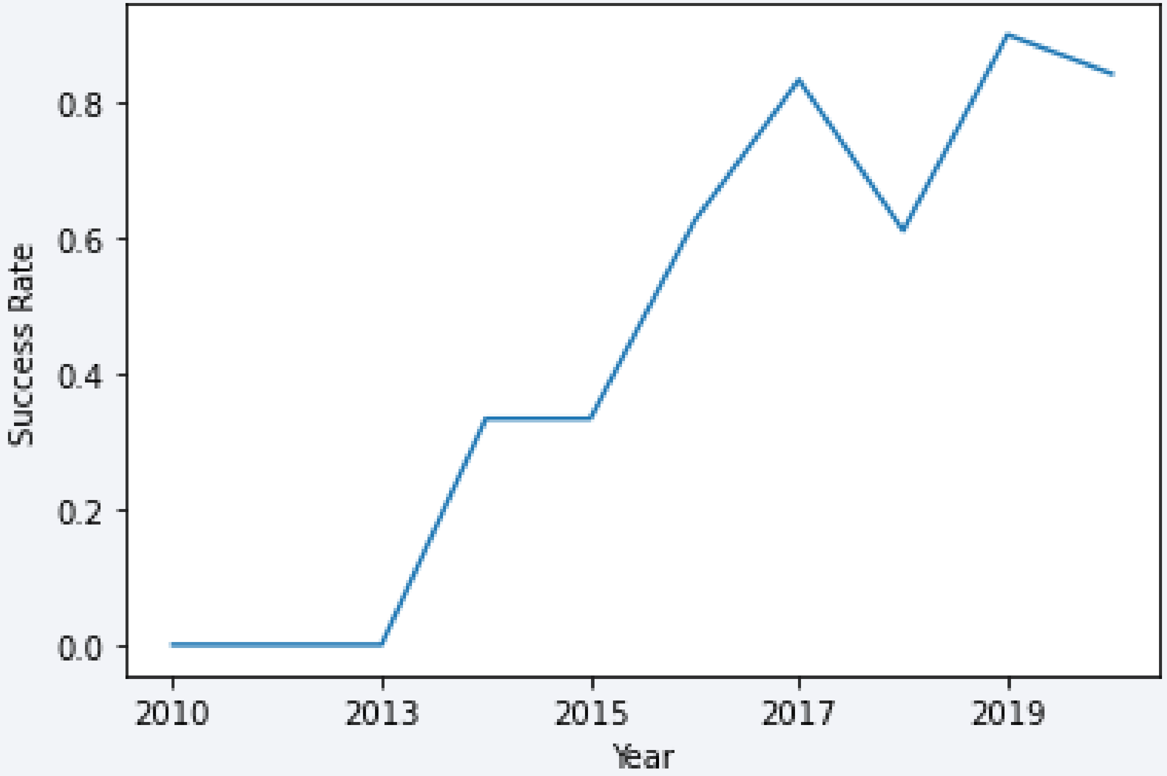


- Class 0 (blue) represents a not successful launch, and Class 1 (orange) represents a successful launch

- SSO 5 launches were successful with low payload launches

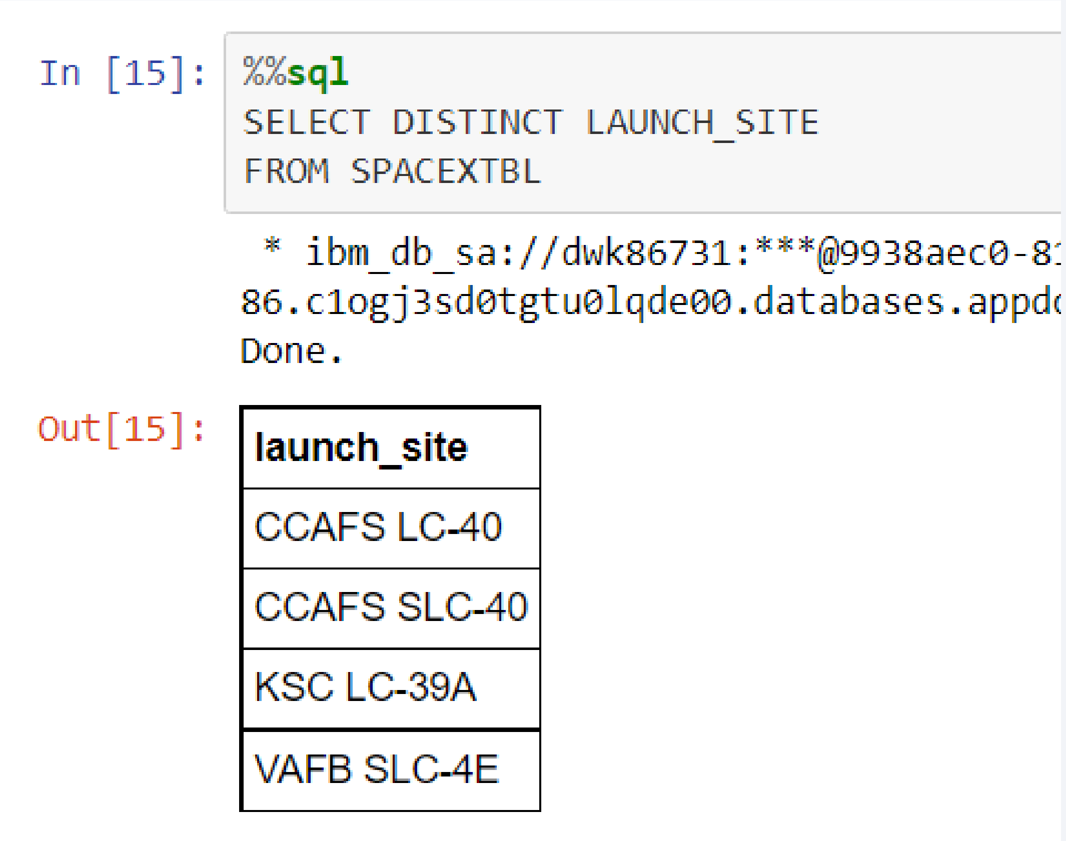
- It’s difficult to understand the GTO success rate between those variables

Launch Success Yearly Trend



- Chart represents yearly average success rate;
- The highest value the better and has been increasing over the years
- There were a rate decreased in 2018

All Launch Site Names



```
In [15]: %%sql
         SELECT DISTINCT LAUNCH_SITE
         FROM SPACEXTBL

          * ibm_db_sa://dwk86731:***@9938aec0-8
         86.c1ogj3sd0tgtu0lqde00.databases.appd
         Done.
```

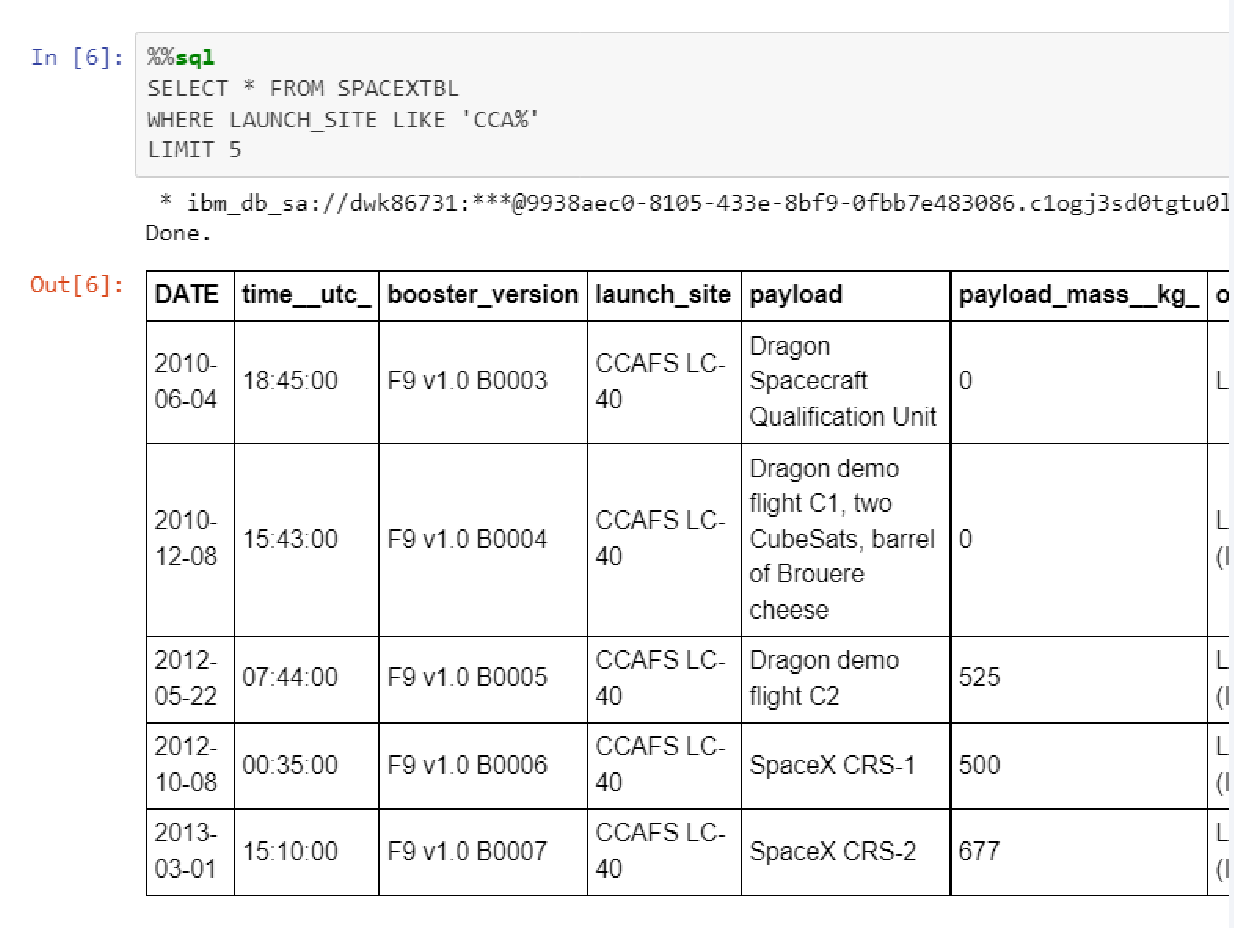
Out[15]:

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

SQL Query Explanation:

Select data from SPACEXTBL table showing only Unique values (DISTINCT) from column ‘launch_site’

Launch Site Names Begin with 'CCA'



```
In [6]: %%sql
        SELECT * FROM SPACEXTBL
        WHERE LAUNCH_SITE LIKE 'CCA%'
        LIMIT 5

         * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1ogj3sd0tgtu0l
        Done.
```

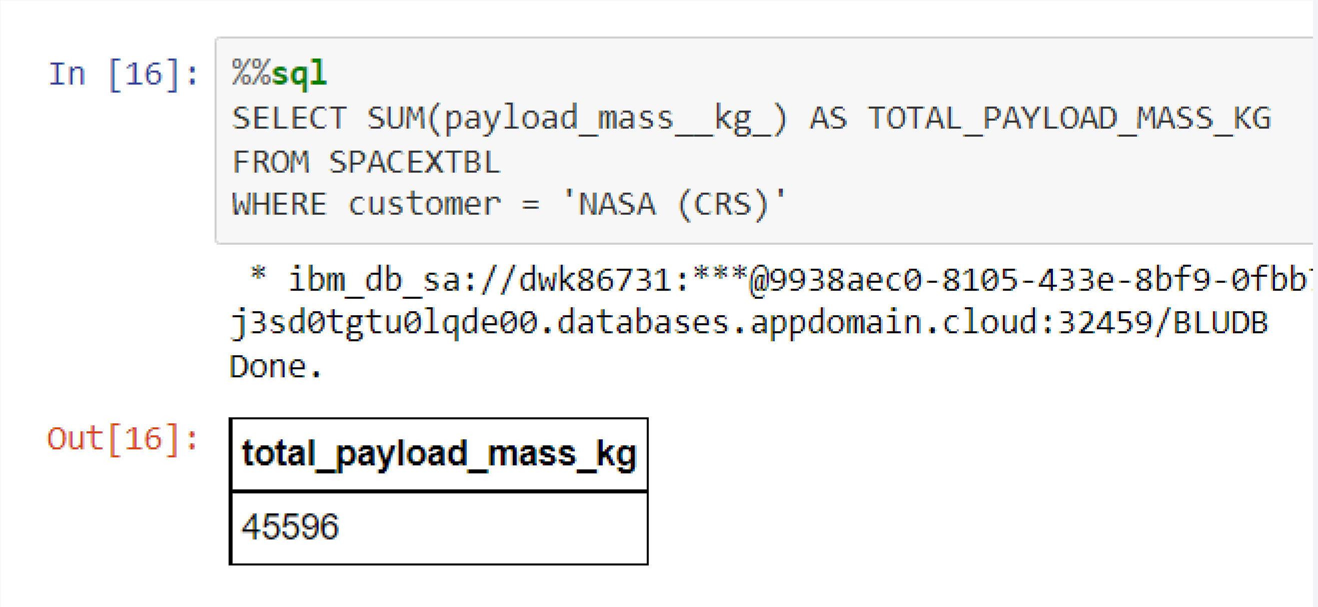
Out[6]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	o
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	L
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	L (I
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	L (I
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	L (I
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	L (I

SQL Query Explanation:

Select data from SPACEXTBL table showing only 5 records where launch sites begin (LIKE) with ‘CCA’ from column ‘launch_site’

Total Payload Mass



```
In [16]: %%sql
         SELECT SUM(payload_mass__kg_) AS TOTAL_PAYLOAD_MASS_KG
         FROM SPACEXTBL
         WHERE customer = 'NASA (CRS)'

          * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0fbb
         j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUDB
         Done.
```

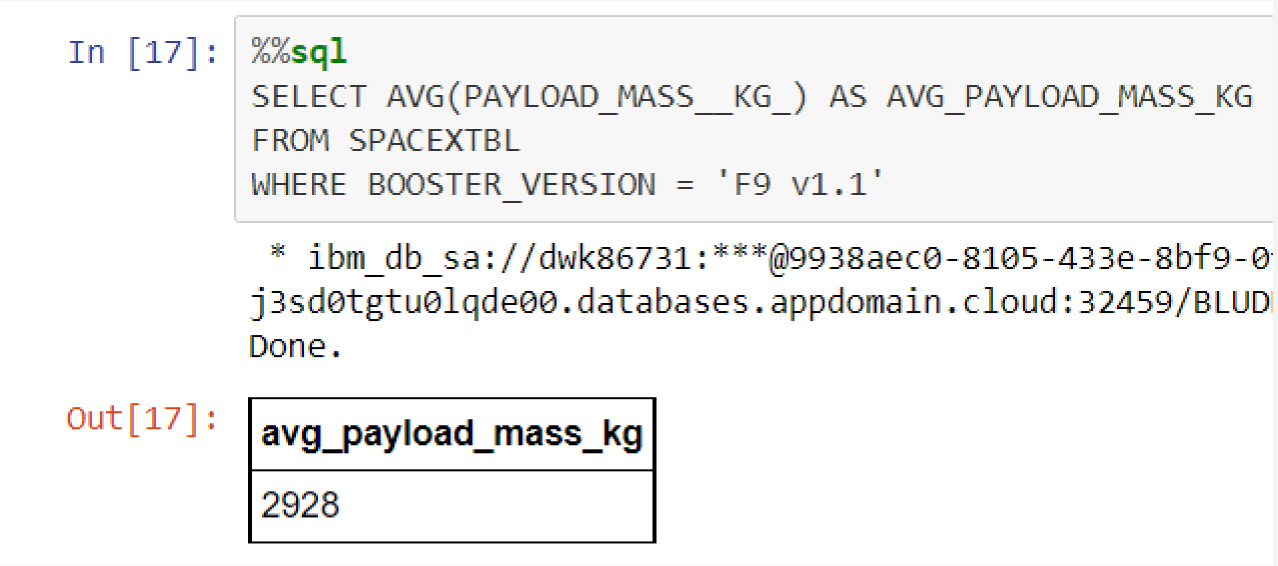
Out[16]:

total_payload_mass_kg
45596

SQL Query Explanation:

Select data from SPACEXTBL table calculating total payload carried (SUM) showing only records where customer is equal ‘NASA (CRS)’ from column ‘customer’

Average Payload Mass by F9 v1.1



```
In [17]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD_MASS_KG
         FROM SPACEXTBL
         WHERE BOOSTER_VERSION = 'F9 v1.1'
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUD
Done.
```

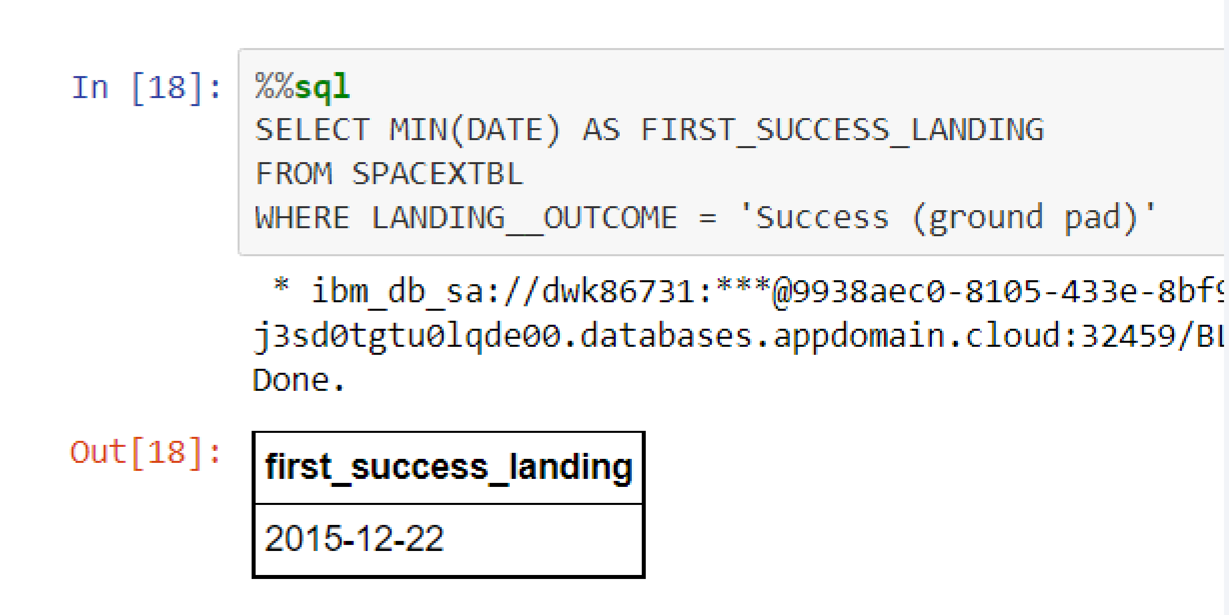
Out[17]:

avg_payload_mass_kg
2928

SQL Query Explanation:

Select data from SPACEXTBL table calculating average payload carried (AVG) showing only records where Booster Version is equal ‘F9 v1.1’ from column ‘BOOSTER_VERSION’

First Successful Ground Landing Date



```
In [18]: %%sql
         SELECT MIN(DATE) AS FIRST_SUCCESS_LANDING
         FROM SPACEXTBL
         WHERE LANDING__OUTCOME = 'Success (ground pad)'

          * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf
         j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/B
         Done.
```

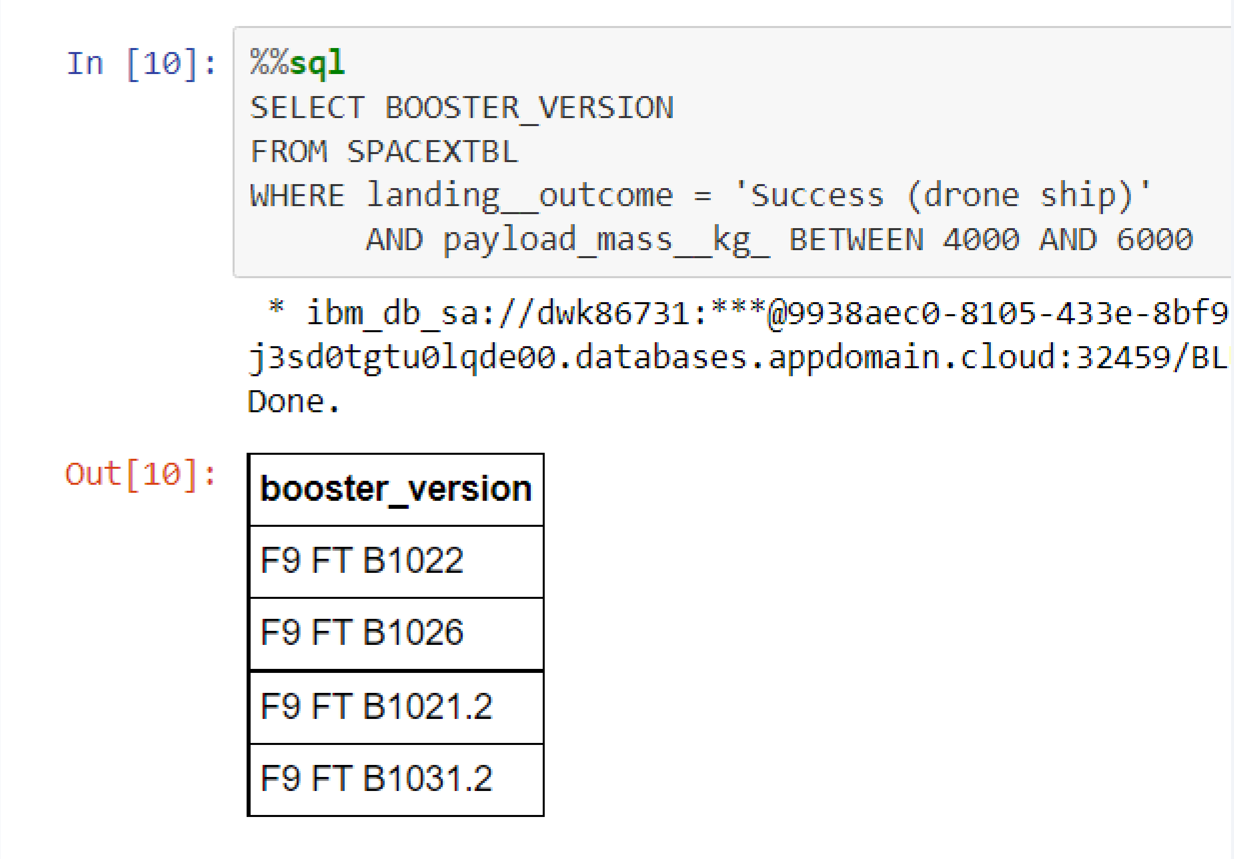
Out[18]:

first_success_landing
2015-12-22

SQL Query Explanation:

Select data from SPACEXTBL table showing the first record (MIN) where landing outcome is equal to ‘Success (ground pad)’ from column ‘LANDING__OUTCOME’

Successful Drone Ship Landing with Payload between 4000 and 6000



```
In [10]: %%sql
         SELECT BOOSTER_VERSION
         FROM SPACEXTBL
         WHERE landing__outcome = 'Success (drone ship)'
               AND payload_mass__kg_ BETWEEN 4000 AND 6000
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BL
Done.
```

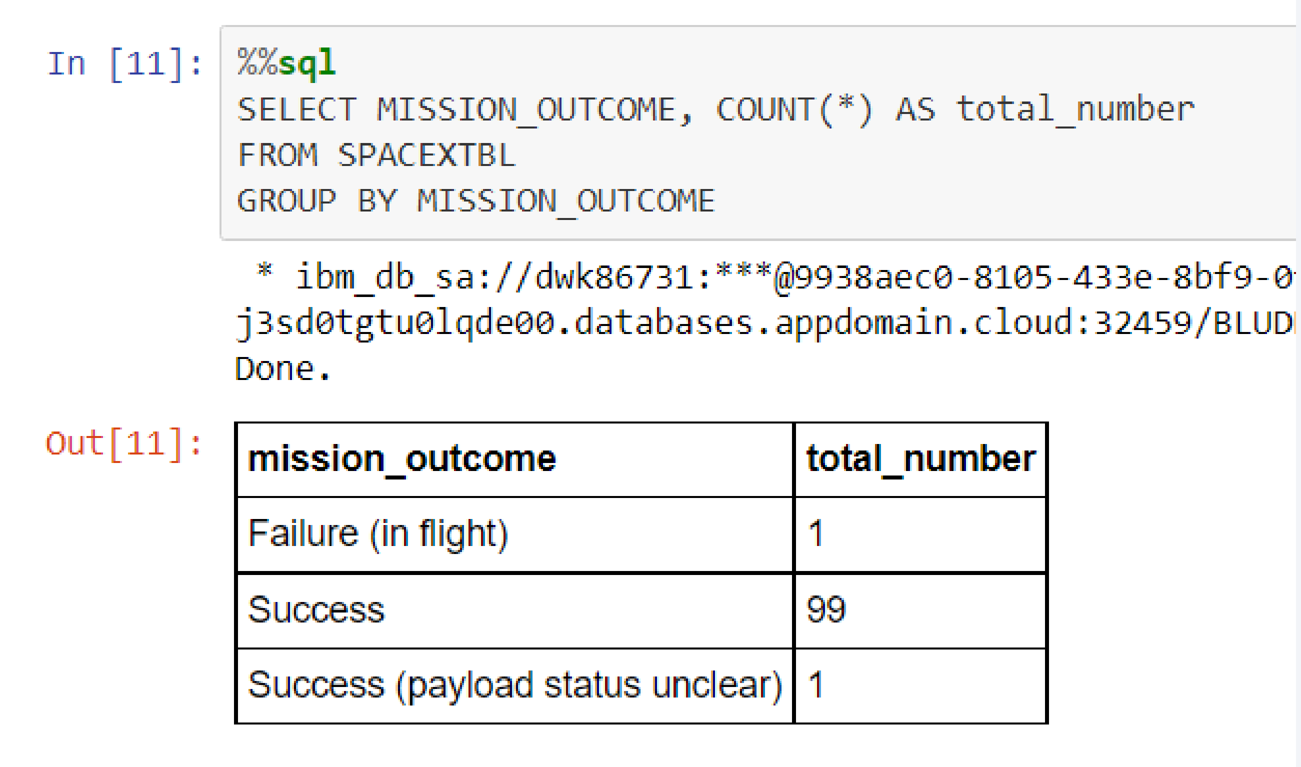
Out[10]:

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

SQL Query Explanation:

Select data from SPACEXTBL table showing the only records where landing outcome is equal to ‘Success (ground pad)’ from column ‘LANDING__OUTCOME’ and payload mass has the value between 4000 and 6000 from column ‘PAYLOAD_MASS__KG_’

Total Number of Successful and Failure Mission Outcomes



In [11]:

```
%%sql
SELECT MISSION_OUTCOME, COUNT(*) AS total_number
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUD
Done.
```

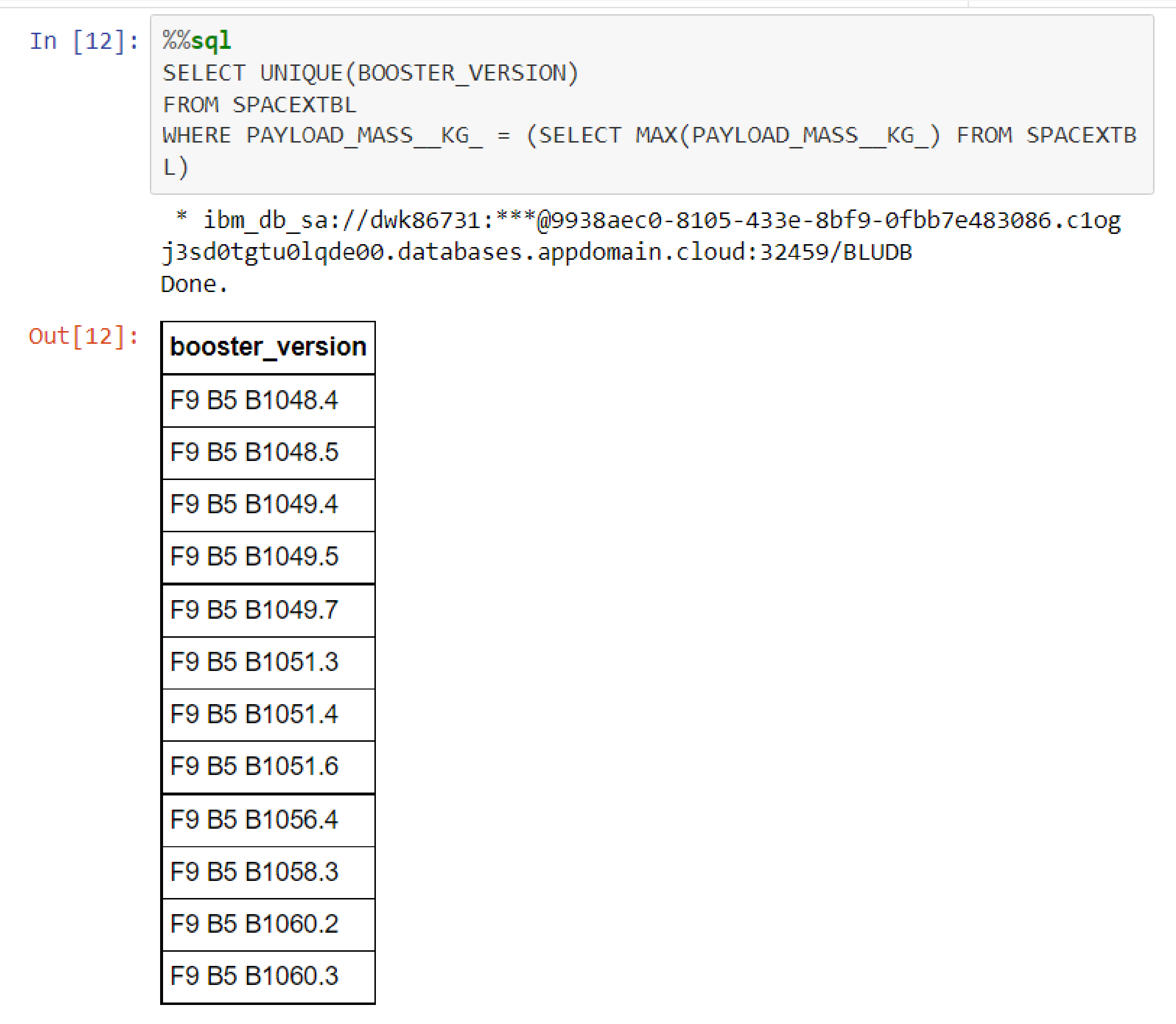
Out[11]:

mission_outcome	total_number
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

SQL Query Explanation:

Select data from SPACEXTBL table showing calculating the total number of successful and failure mission outcomes grouping values from column 'MISSION_OUTCOME'

Boosters Carried Maximum Payload



```
In [12]: %%sql
         SELECT UNIQUE(BOOSTER_VERSION)
         FROM SPACEXTBL
         WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTB
         L)
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0fbb7e483086.c1og
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

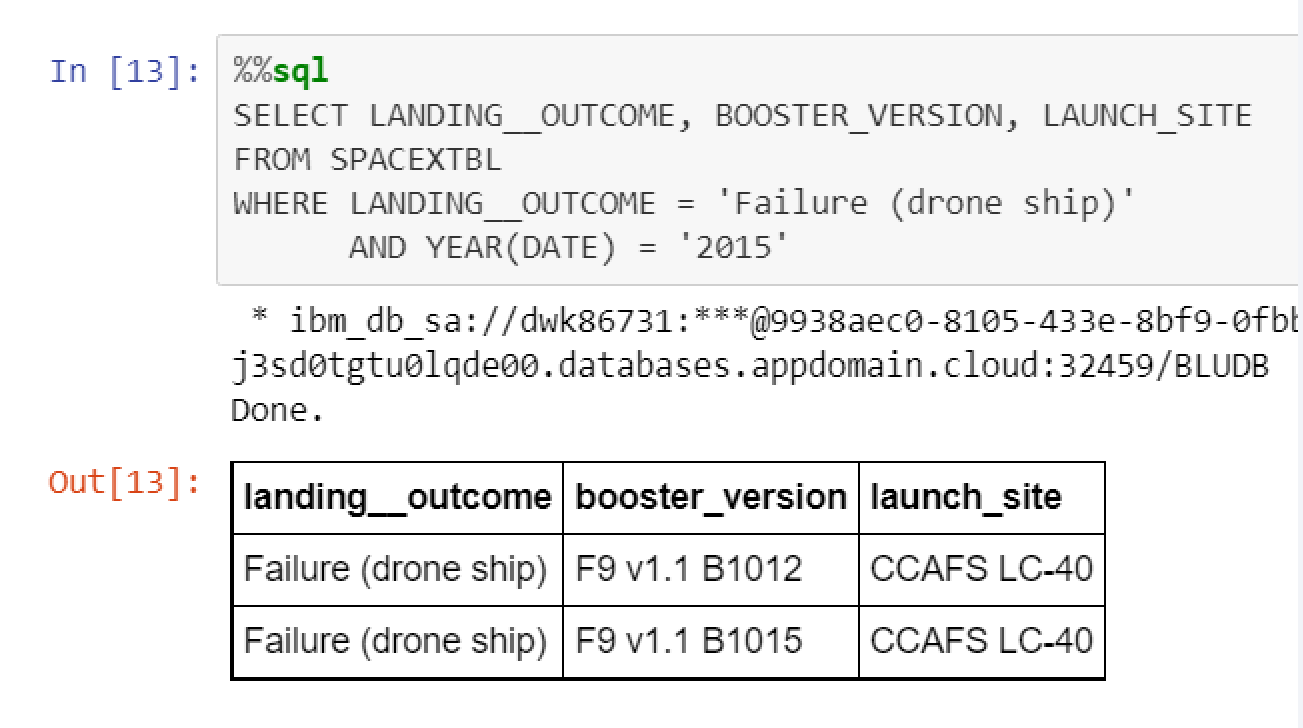
Out[12]:

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

SQL Query Explanation:

Select data from SPACEXTBL table showing only records where payload mass has the maximum payload mass (MAX into SUBQUERY) from column ‘PAYLOAD_MASS__KG_’

2015 Launch Records



In [13]:

```
%%sql
SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)'
      AND YEAR(DATE) = '2015'
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0fbb
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

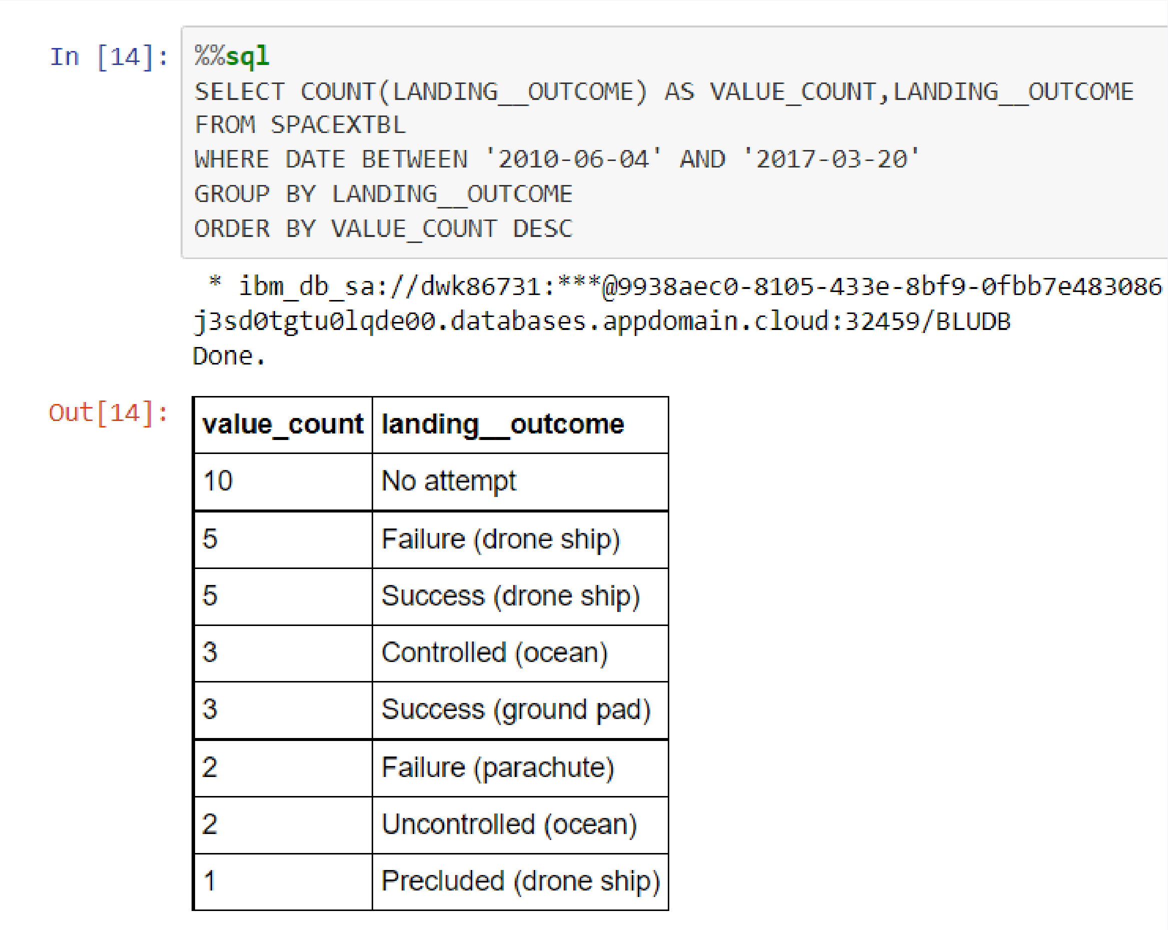
Out[13]:

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

SQL Query Explanation:

Select data from SPACEXTBL table showing only records where landing outcome is equal to 'Failure (drone ship)' from column 'LANDING__OUTCOME' and year is equal '2015' from column 'DATE' converted to get only year value (YEAR)

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



```
In [14]: %%sql
         SELECT COUNT(LANDING__OUTCOME) AS VALUE_COUNT,LANDING__OUTCOME
         FROM SPACEXTBL
         WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
         GROUP BY LANDING__OUTCOME
         ORDER BY VALUE_COUNT DESC
```

```
 * ibm_db_sa://dwk86731:***@9938aec0-8105-433e-8bf9-0fbb7e483086
j3sd0tgtu0lqde00.databases.appdomain.cloud:32459/BLUDB
Done.
```

Out[14]:

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

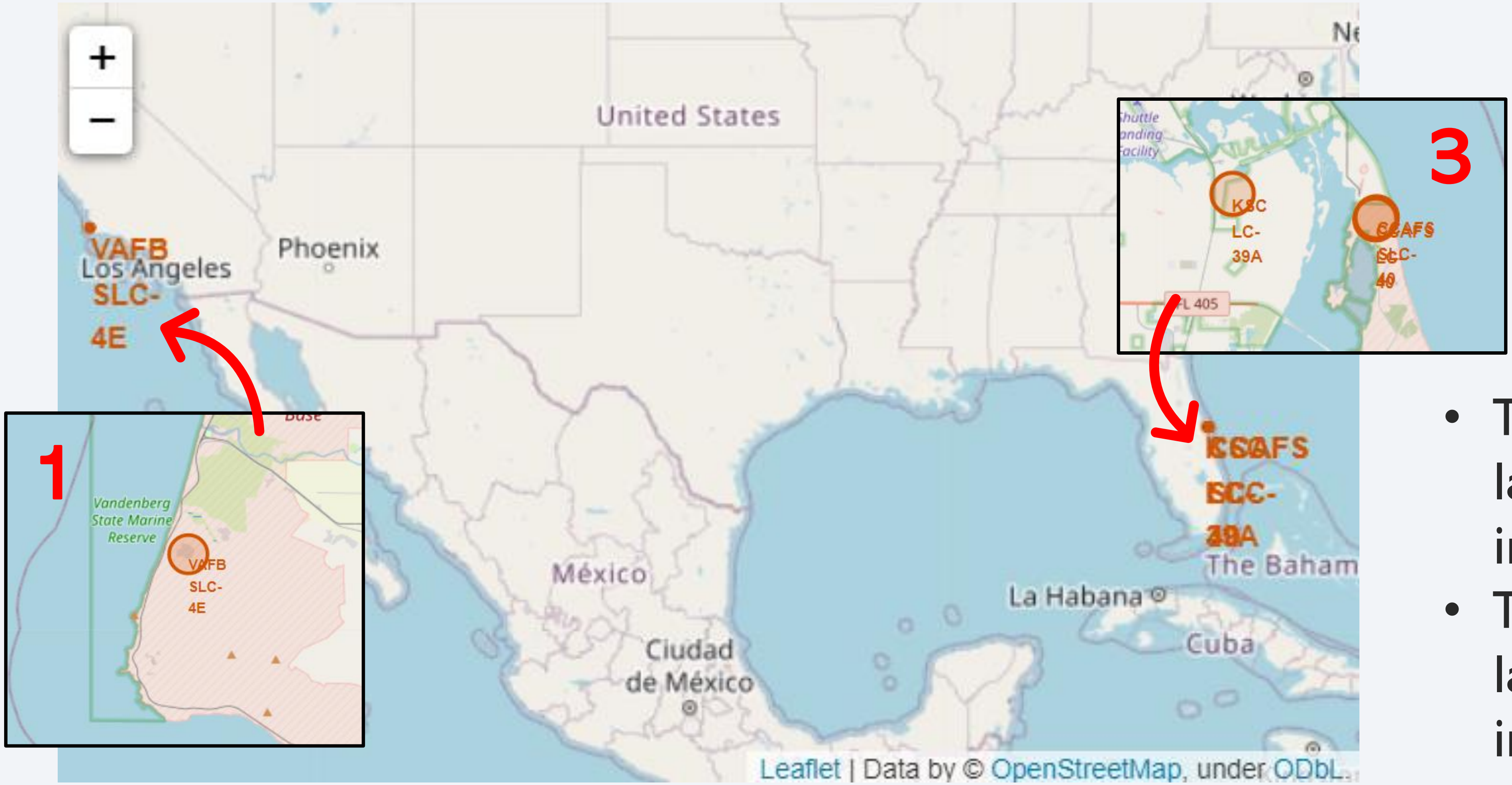
SQL Query Explanation:

Select data from SPACEXTBL table raking the number (COUNT) of landing outcomes showing only records where dates are between '2010-06-04' and '2017-03-20' grouping by values from column 'LADING__OUTCOME' and descending ordering by values from column 'VALUE_COUNT'

Section 3

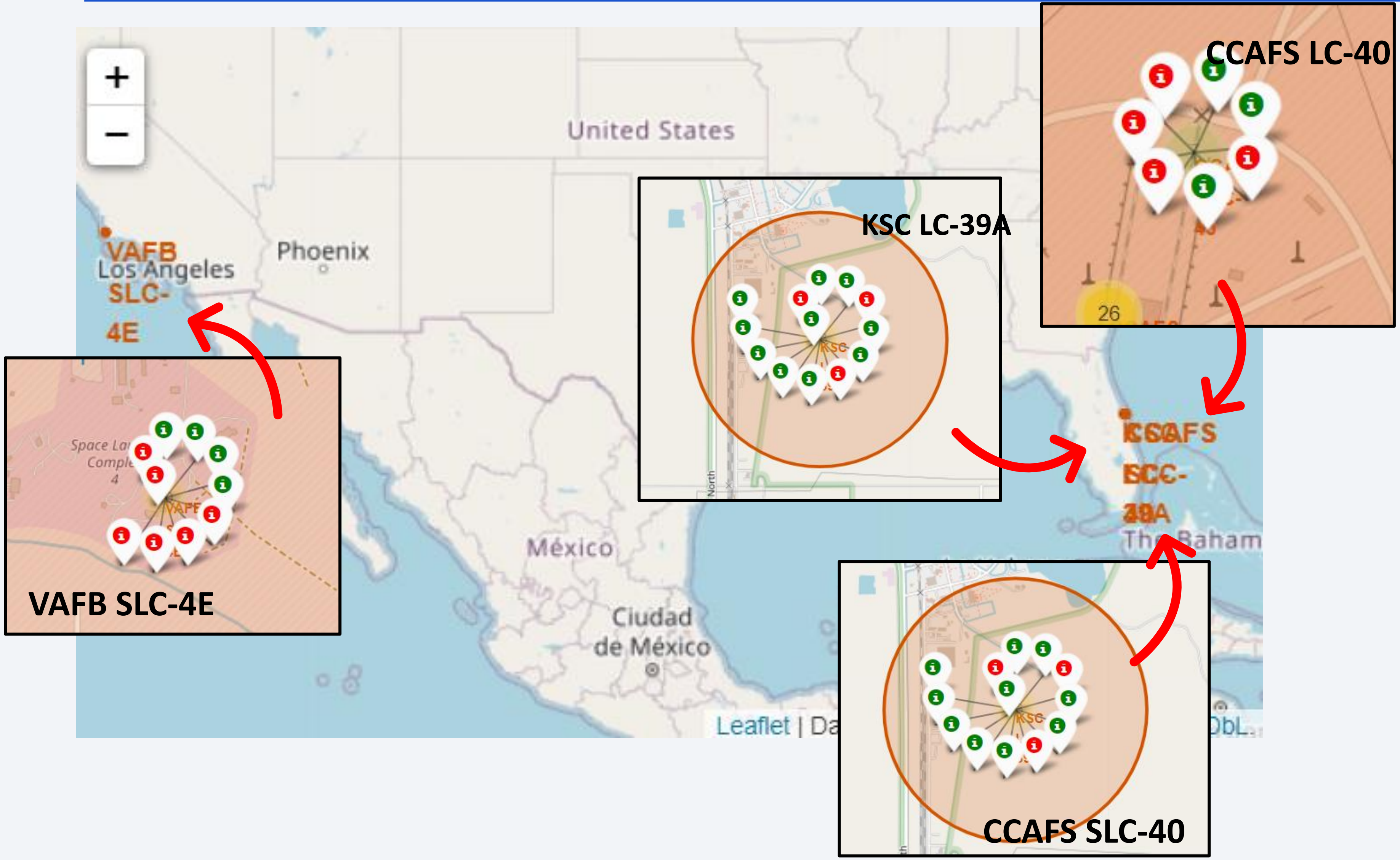
Launch Sites Proximities Analysis

All launch sites on a map



- There are 3 launching sites in Florida
- There is just 1 launching site in California

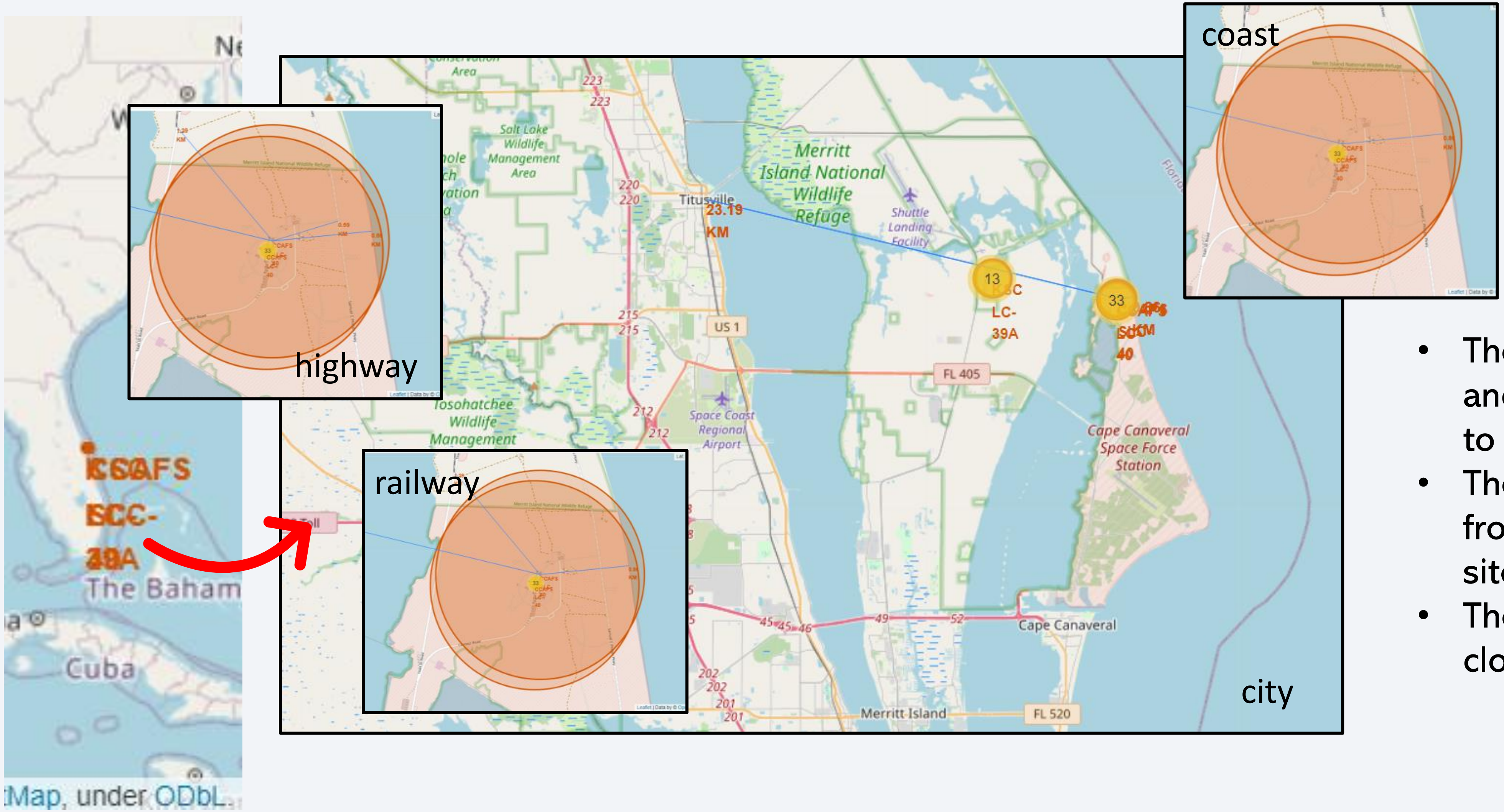
Success/Failed launches for each site on the map



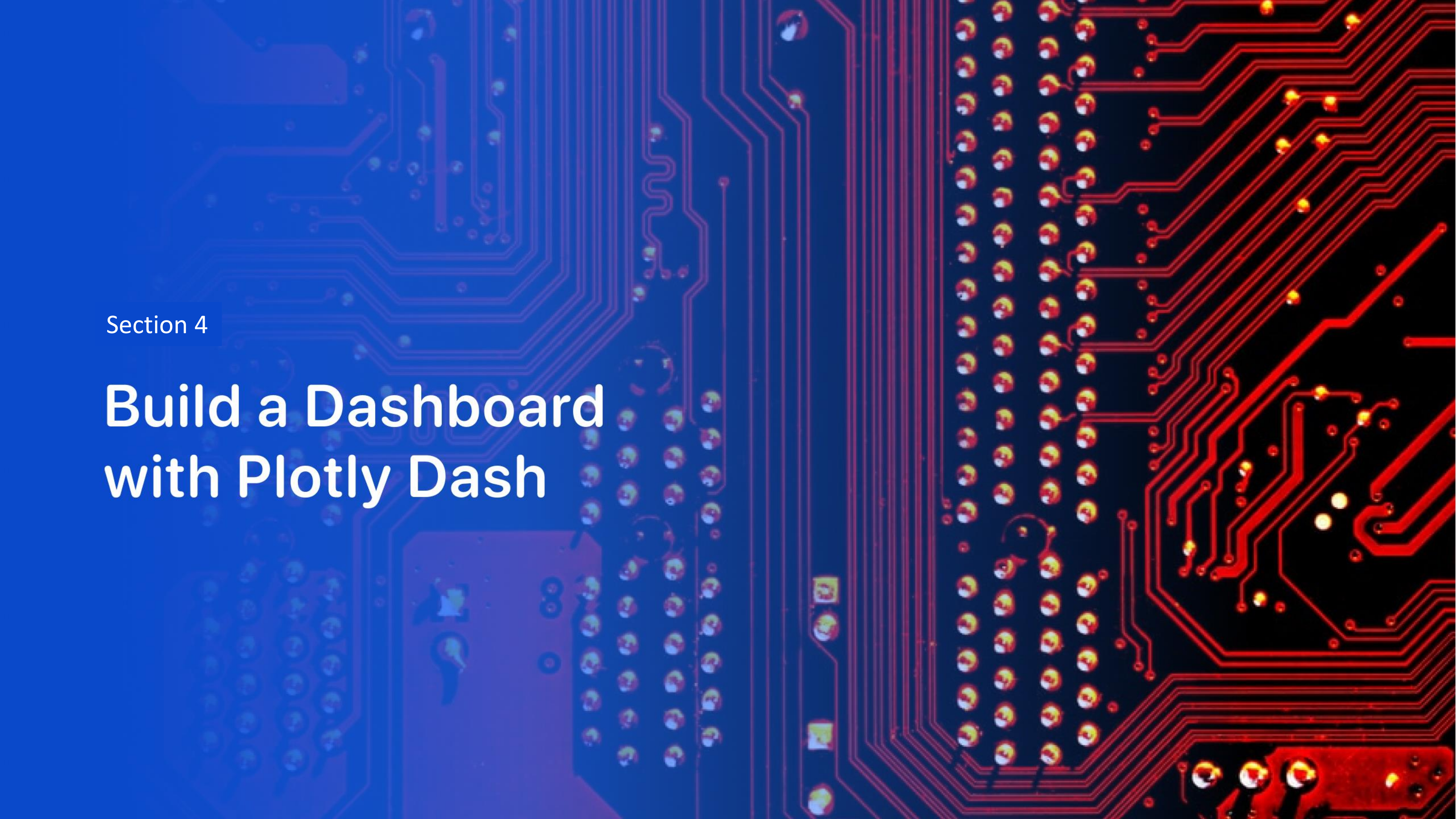
There are more numbers of launches in site CCASF SLC-40

The highest success rate is in KSC LC-39A

Proximities of Launch Sites


coast
highway
railway
city
23.19 KM
13
33
KSC LC-39A
CCAFS SLC-40
Titusville
Merritt Island National Wildlife Refuge
Shuttle Landing Facility
Space Coast Regional Airport
Cape Canaveral Space Force Station
Cape Canaveral
Merritt Island
Tosohatchee Wildlife Management
Salt Lake Wildlife Management Area
US 1
FL 405
FL 520
Cuba
The Baham
Map, under ODbL.

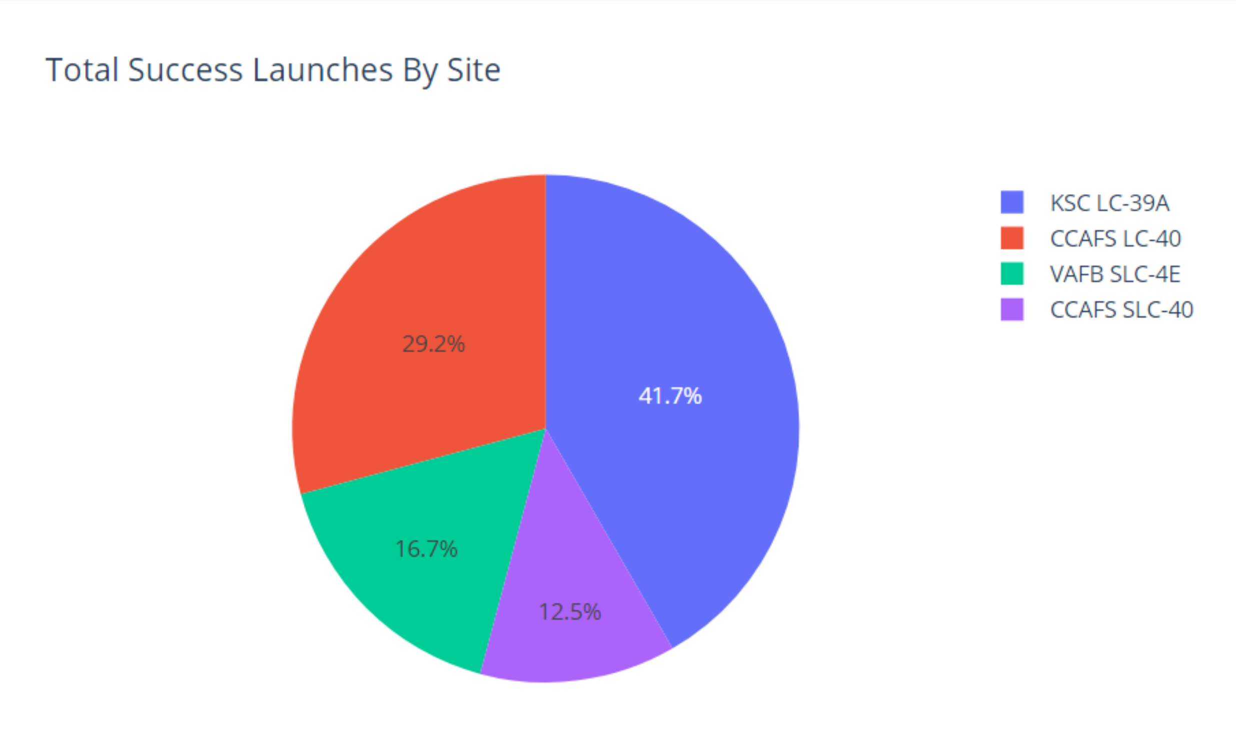
- There is railways and highways close to launch site;
- The cities are far from the launch site;
- The coastline are close to launch site.



Section 4

Build a Dashboard with Plotly Dash

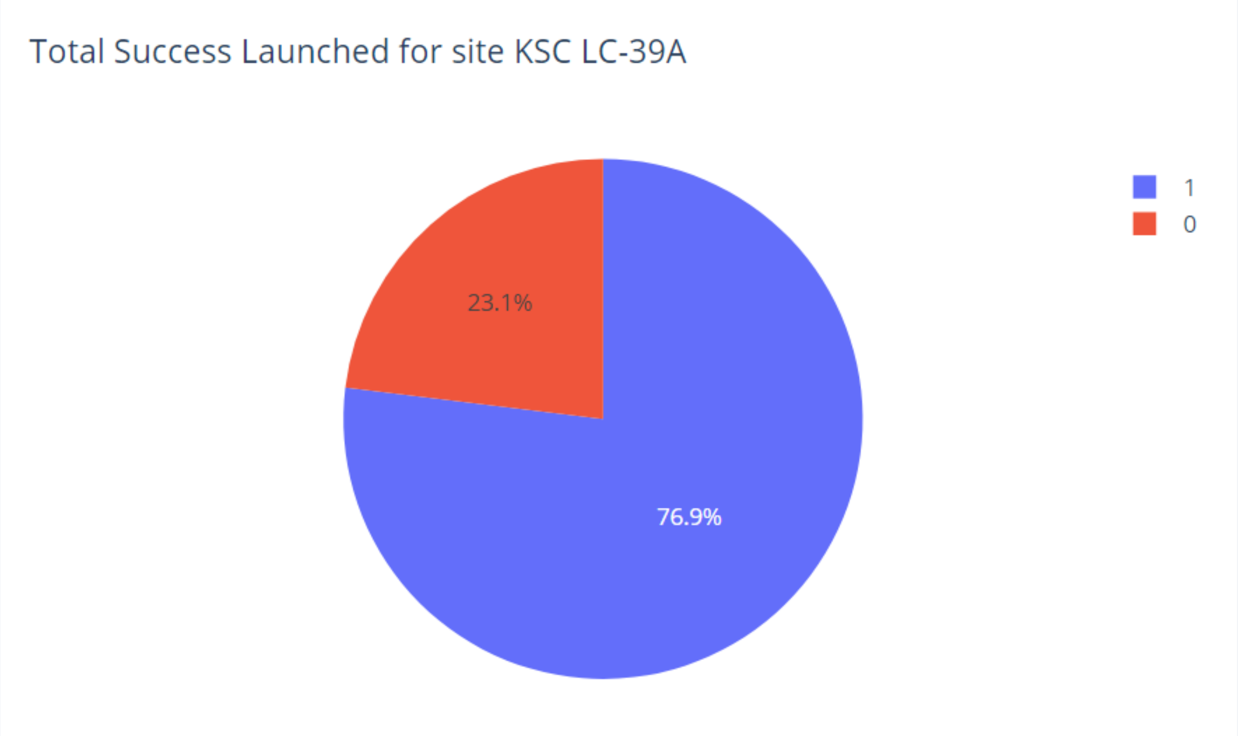
Total Success Launches by Sites



There are more records of success launch in KSLC-39 site.

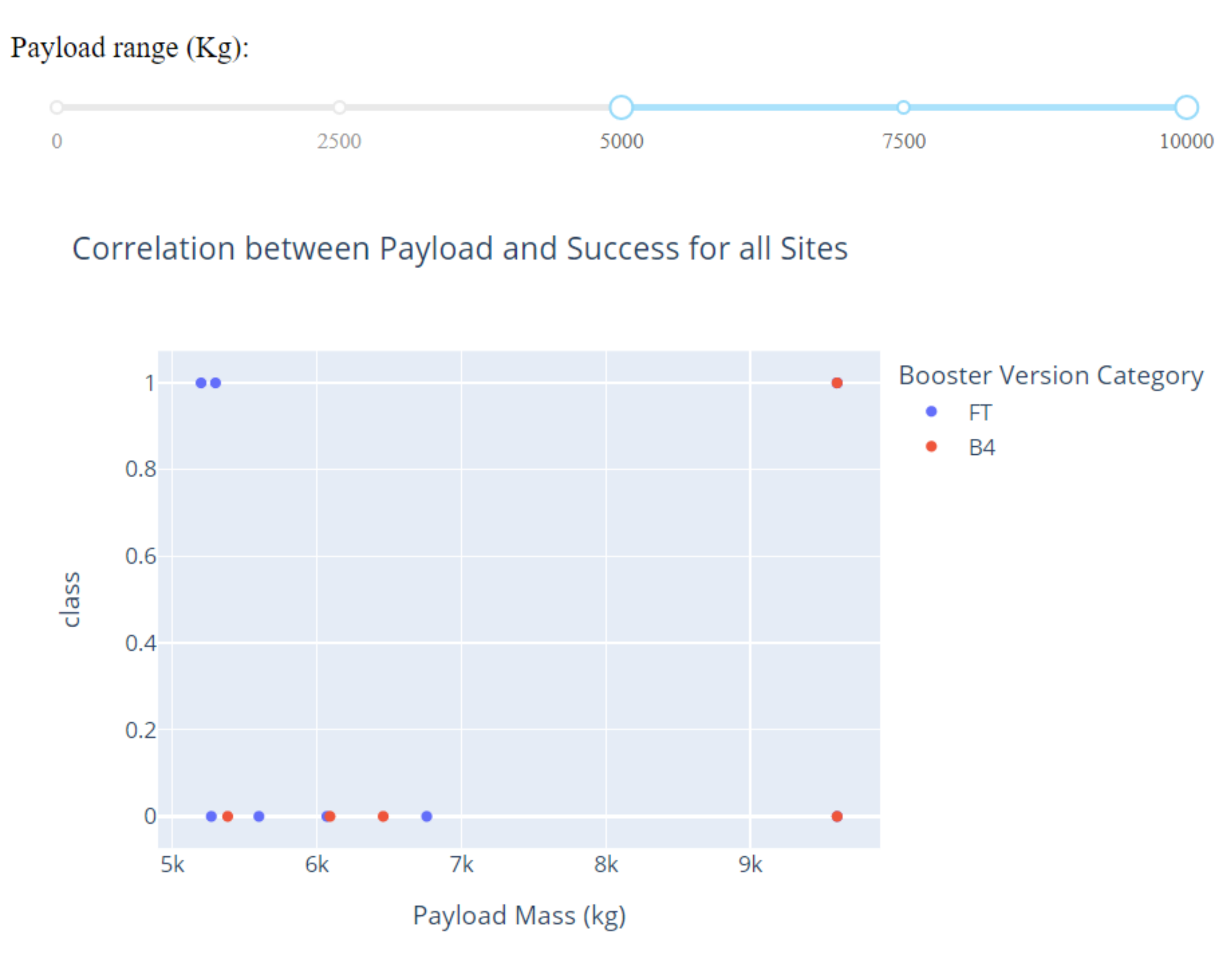
Site launch with fewer success rate is VAFB SLSC-4E

Success Launches at Site: KSC LC-39A



The records of success launch in this site are 76.9% compared by 23.1% of failure rate

Correlation between Payload and Success for all Sites

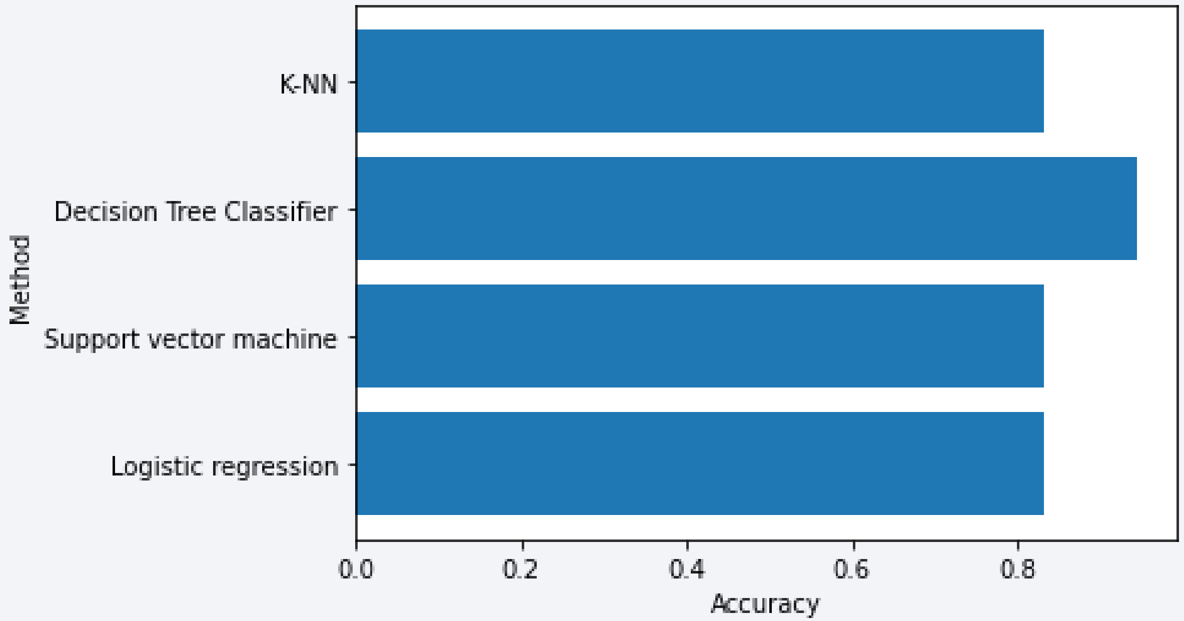


Within a Payload range of 5000 to 10000 kg, Booster Version B4 has the largest failure rate.

Section 5

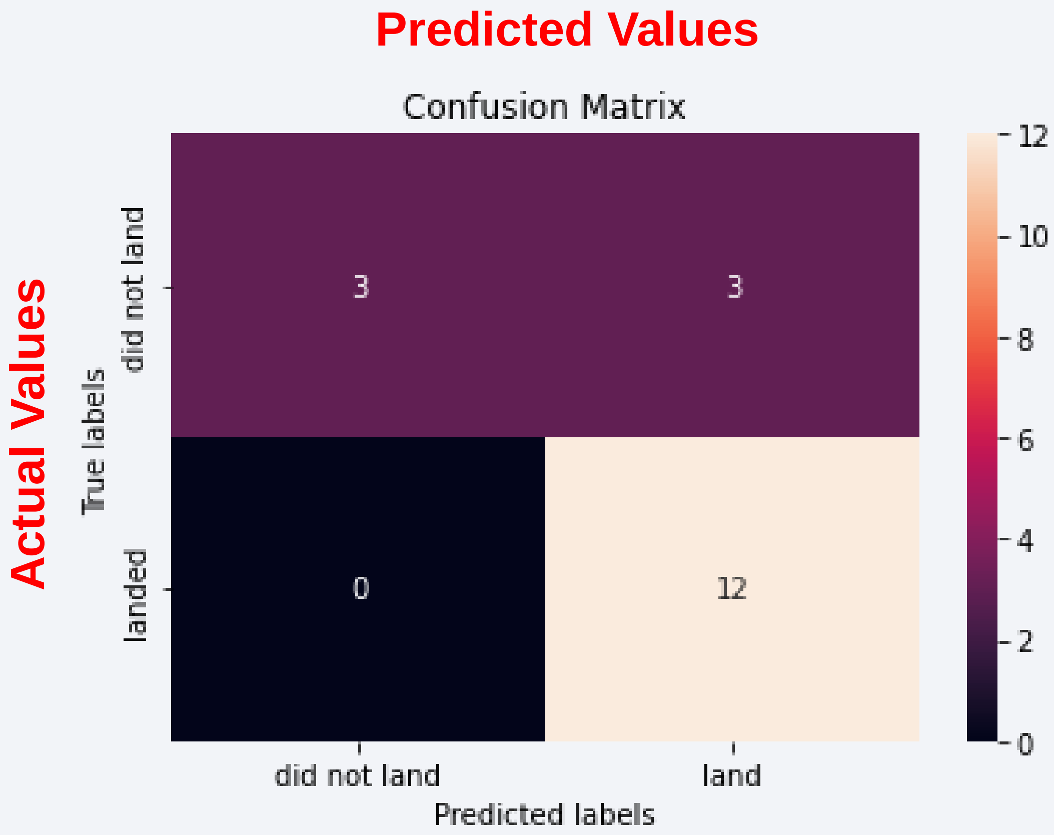
Predictive Analysis (Classification)

Classification Accuracy



Accuracy of 94,44% from Decision Tree Classifier is higher than the other models around value of 83.33%.

Confusion Matrix



- Success landing prediction well on all kind of models.

- False positive were found because 3 predictions that said successful landings when the actual value was failure

Conclusions

- Success rate increased when the number of flights increased.;
- Launch site is close to railways, highways, and coastline, but far from cities;
- Success landing rare perform better with low weighted payloads than the heavier payloads
- Prediction of successful launch with Machine learning models has above of 83% accuracy.
- The highest model accuracy is Decision Tree model

Appendix

- List of Falcon 9 and Falcon Heavy launches

https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&direction=next&oldid=1027686922#References

- FALCON 9

https://web.archive.org/web/20140805175724/http://www.spacex.com/falcon9

- SpaceX

https://en.wikipedia.org/wiki/SpaceX

Thank you!