IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Madhawa Hulangamuwa
23/05/2025

Outline

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

Executive Summary

Methodology

Data - SpaceX REST API & Web scraping techniques

Data wrangling to create data success/fail outcome variable

Exploratory Data Analysis (EDA) using visualization and SQL

Interactive visual analytics using Folium and Plotly Dash

Prediction techniques - logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN).

Results

I. Launch success has improved over time

II. KSC LC-39A has the highest success rate among landing sites

III. Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate

IV. All models performed similarly on the test set. The decision tree model slightly outperformed

Introduction

Background

SpaceX, a leading company in the space industry, is dedicated to making space travel accessible and affordable for everyone. They have achieved remarkable milestones such as successfully delivering spacecraft to the International Space Station, launching a satellite constellation that provides global internet access, and conducting manned missions to space. One key factor contributing to SpaceX's cost efficiency is their innovative approach of reusing the first stage of their Falcon 9 rockets, significantly reducing launch expenses to approximately $62 million per launch. In contrast, other providers, lacking the capability to reuse the first stage, face costs upwards of $165 million per launch.

Objective

1. Predict whether the first stage will be reusable using public and machine learning techniques.
2. Identify the best predictive machine learning models.

Section 1

Methodology

Methodology

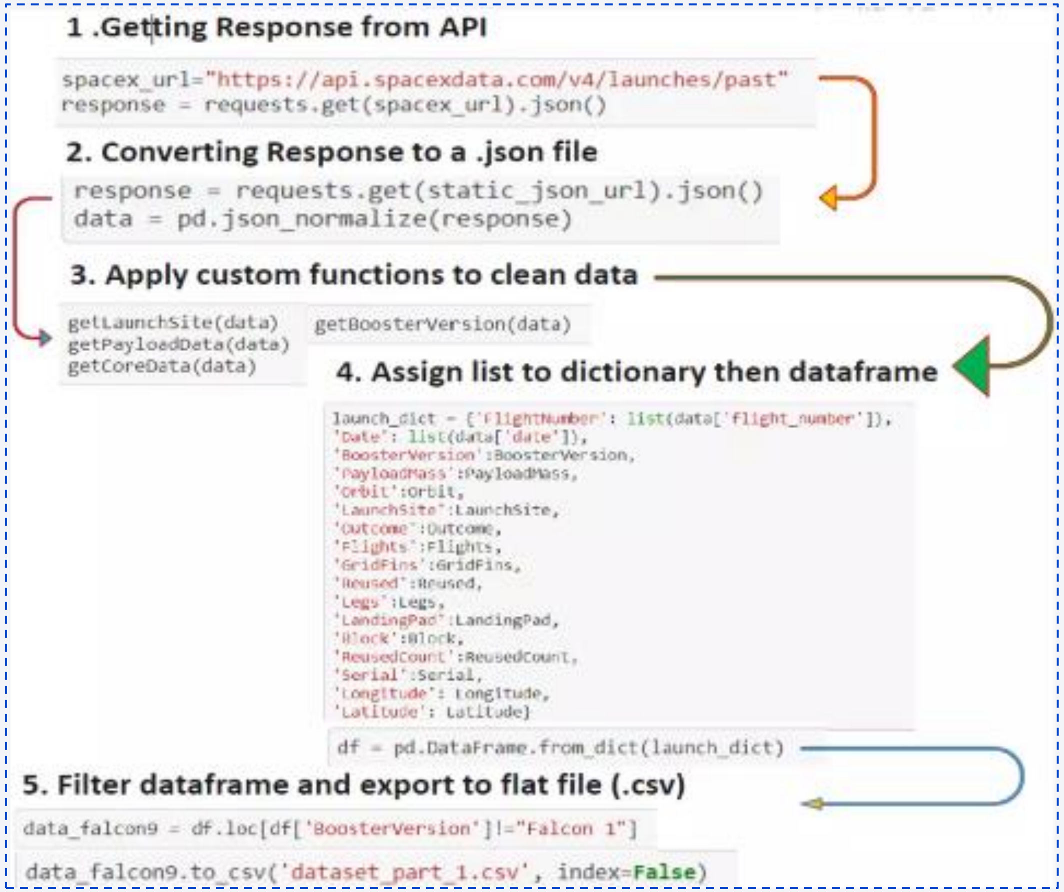
- Data collected using SpaceX REST API and web scraping techniques.
- Data wrangling – handling missing values, one hot encoding applied to transfer categorical variables.
- EDA carried out through SQL and Python.
- Data visualized using Folium and Plotly Dash.
- Logistic regression, support vector machine (SVM), decision tree, and K-nearest neighbor (KNN) were considered in modeling.
- Models were evaluated using accuracy score, F1 score, and Jaccard score.

Data Collection

- Request rocket launch data from SpaceX API.
- Decode the API response using .json() and convert it to a dataframe using .json_normalize().
- Retrieve information about the launches from SpaceX API using custom functions.
- Create a dictionary from the data.
- Generate a dataframe from the dictionary.
- Filter the dataframe to include only Falcon 9 launches.
- Replace missing values of Payload Mass with the calculated mean.
- Export the data to a CSV file.

Data Collection – SpaceX API

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/01_jupyter-labs-spacex-data-collection-api.ipynb



1 .Getting Response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url).json()
```

2. Converting Response to a .json file

```
response = requests.get(static_json_url).json()
data = pd.json_normalize(response)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

```
getBoosterVersion(data)
```

4. Assign list to dictionary then dataframe

```
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}
```

```
df = pd.DataFrame.from_dict(launch_dict)
```

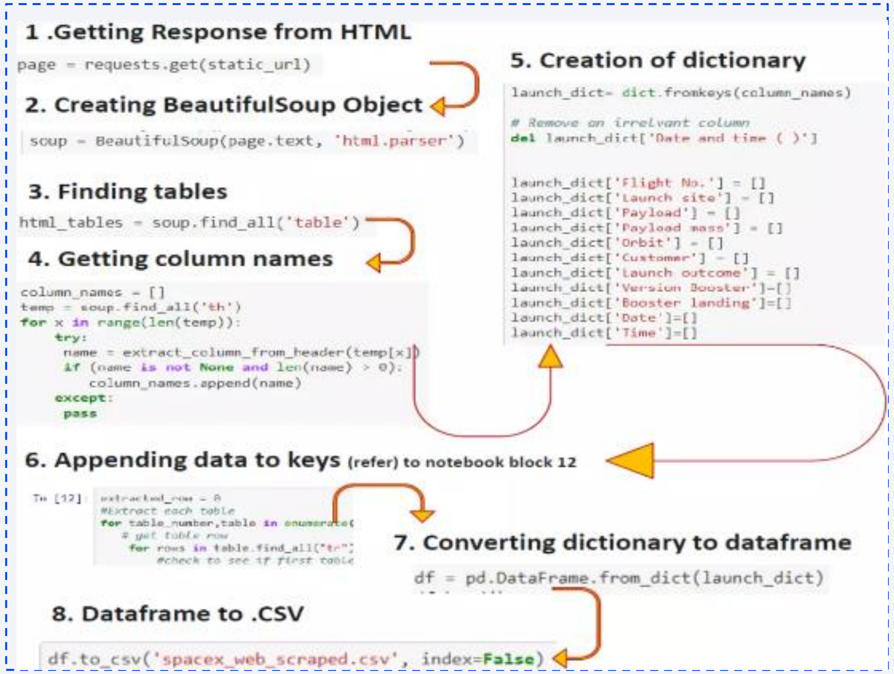
5. Filter dataframe and export to flat file (.csv)

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

- Retrieve Falcon 9 launch data from Wikipedia.
- Instantiate a BeautifulSoup object using the HTML response.
- Extract column names from the HTML table header.
- Gather data by parsing HTML tables.
- Generate a dictionary from the collected data.
- Create a dataframe from the dictionary and export data to a CSV.



1 .Getting Response from HTML

```
page = requests.get(static_url)
```

2. Creating BeautifulSoup Object

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

3. Finding tables

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

4. Getting column names

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']


launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Appending data to keys (refer) to notebook block 12

```
In [12]: extracted_row = 0
         #Extract each table
         for table_number,table in enumerate(
             # get table row
              for rows in table.find_all("tr")
                   #check to see if first table
```

7. Converting dictionary to dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Dataframe to .CSV

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/02_jupyter-labs-webscraping.ipynb

Data Wrangling

- Check null values
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Create a landing outcome label from outcome column
- Handle null values

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/03_labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Charts

Flight Number vs. Payload

Flight Number vs. Launch Site

Payload Mass (kg) vs. Launch Site

Payload Mass (kg) vs. Orbit type

- Scatter plots were drawn to visualize the relationships between continuous variables.
- Bar graphs are used to compare and display categorical data.
- Catplots were used to visualize the distribution of categorical variables.

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/05_jupyter-labs-eda-dataviz.ipynb

EDA with SQL

SQL queries you performed

Displayed names of unique launch site

Displayed 5 records where launch site begins with ‘CCA’

Displayed total payload mass carried by boosters launched by NASA (CRS)

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

List date of first successful landing on ground pad

List names of boosters which had success landing on drone ship and have payload mass greater than 4,000 but less than 6,000

List total number of successful and failed missions

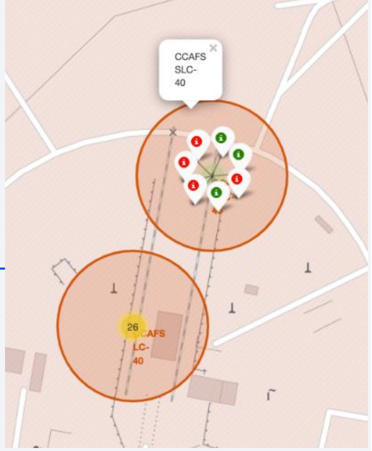
List names of booster versions which have carried the max payload

List failed landing outcomes on drone ship, their booster version and launch site for the months in the year 2015

Ranked count of landing outcomes between 2010-06-04 and 2017-03-20 (desc)

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/04_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Interactive Map with Folium



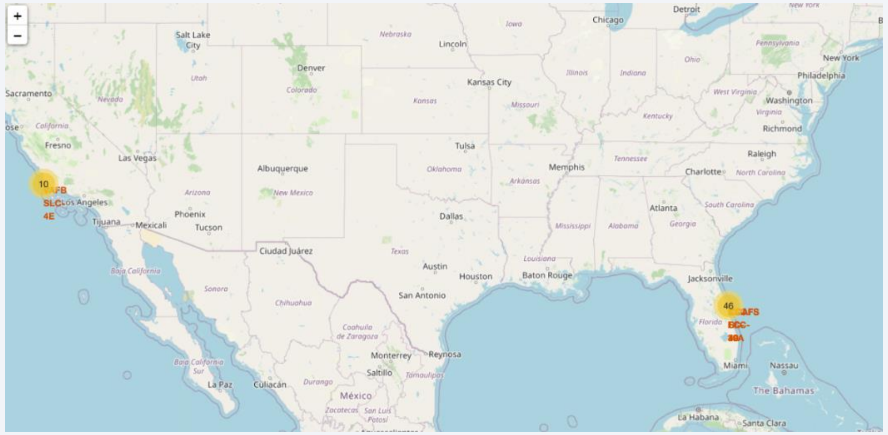
Circles & Markers

- **Blue** circle at NASA Johnson Space Center's coordinate with a popup label (name, latitude, longitudes)
- **Red** circles at all launch sites coordinates with a popup label (name, latitude, longitudes)
- Colored markers of successful (green) and unsuccessful (red) launches

Lines

- Colored lines to show distance between launch site CCAFS SLC- 40 and its proximity to the nearest coastline, railway, highway, and city

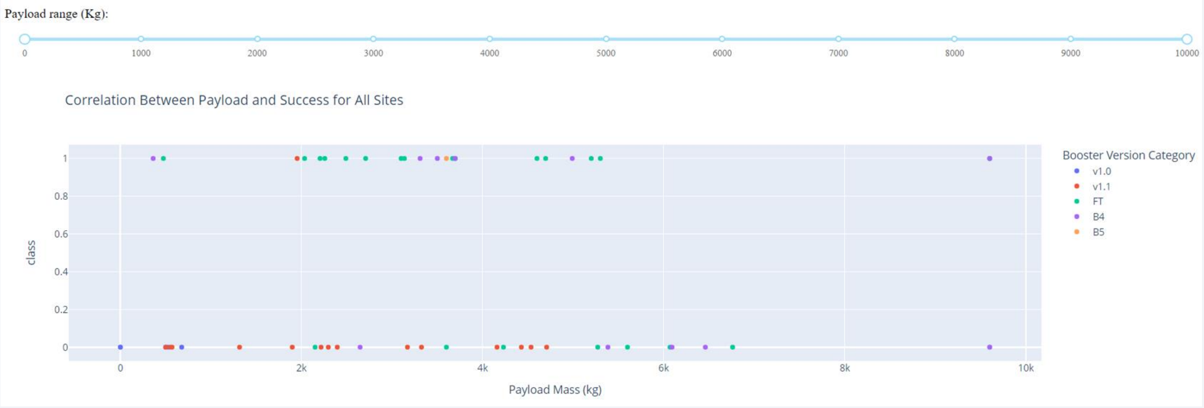
GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/06_lab_jupyter_launch_site_location_Folium.ipynb



Dashboard with Plotly Dash

- Dropdown list with launch sites - allow users to select all launch or a certain launch site
- Pie chart showing successful launches - allow user to see successful and unsuccessful launches as a percent of the total
- Slider of payload mass range - to select payload mass range
- Scatter chart showing payload mass vs. success rate by booster version - to see the correlation between payload and launch success

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/07_spacex_dash_app.py



Predictive Analysis (Classification)

- Create NumPy array from the Class column
- Standardize the data with StandardScaler. Fit and transform the data.
- Split the data using train_test_split
- Create a GridSearchCV object with cv=10 for parameter optimization
- Apply GridSearchCV on different algorithms: logistic regression (LogisticRegression()), support vector machine (SVC()), decision tree (DecisionTreeClassifier()), K-Nearest Neighbor (KNeighborsClassifier())
- Calculate accuracy on the test data using .score() for all models
- Assess the confusion matrix for all models
- Identify the best model using Jaccard_Score, F1_Score and Accuracy

GitHub URL - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone/blob/main/08_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

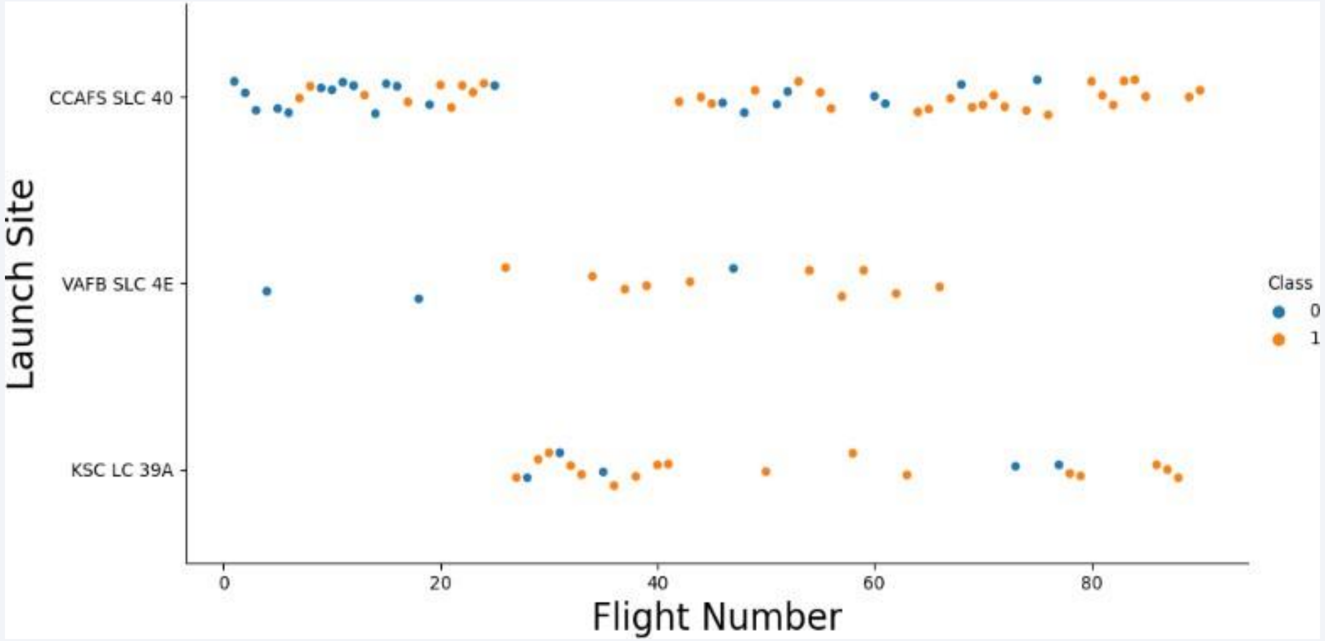
Results

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

Section 2

Insights drawn from EDA

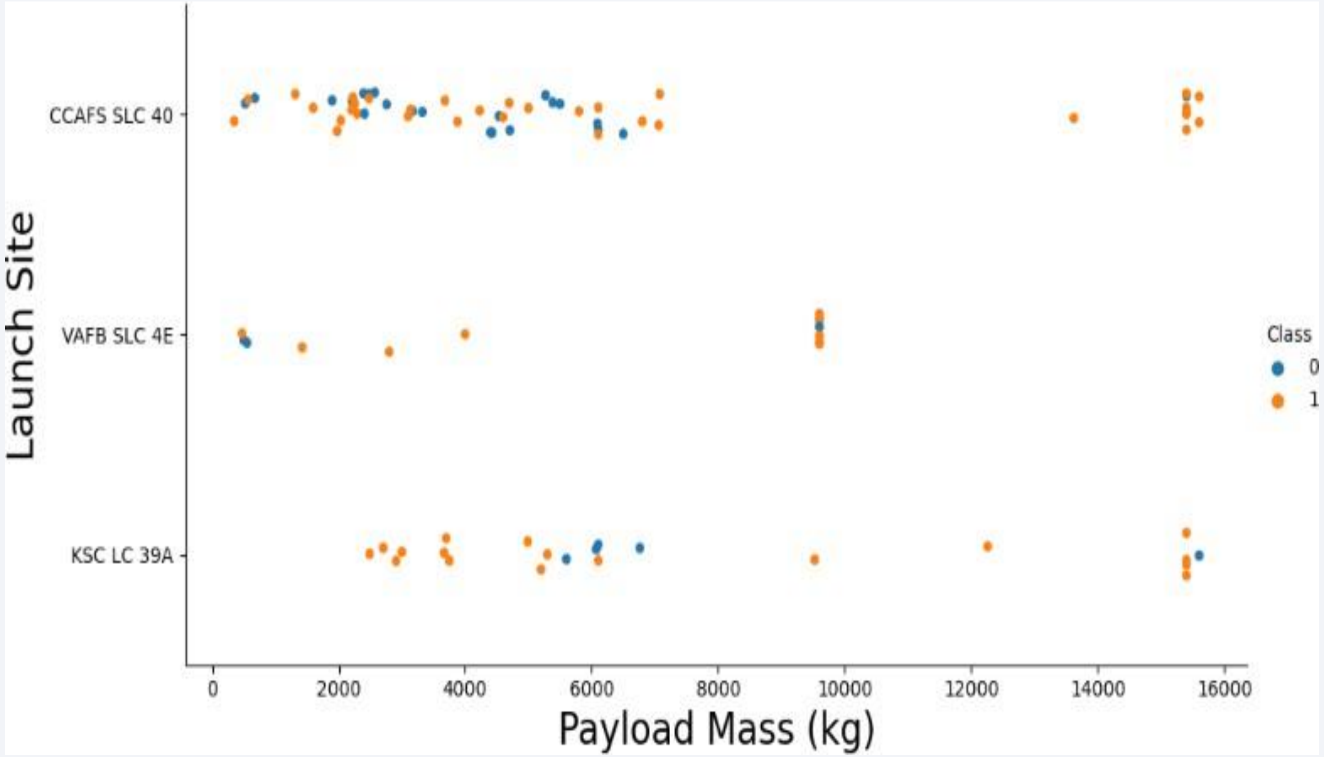
Flight Number vs. Launch Site



- Launches from CCAFS SLC 40 are significantly higher than other launch sites
- VAFB SLC 4E and KSC LC 39A have higher success rates

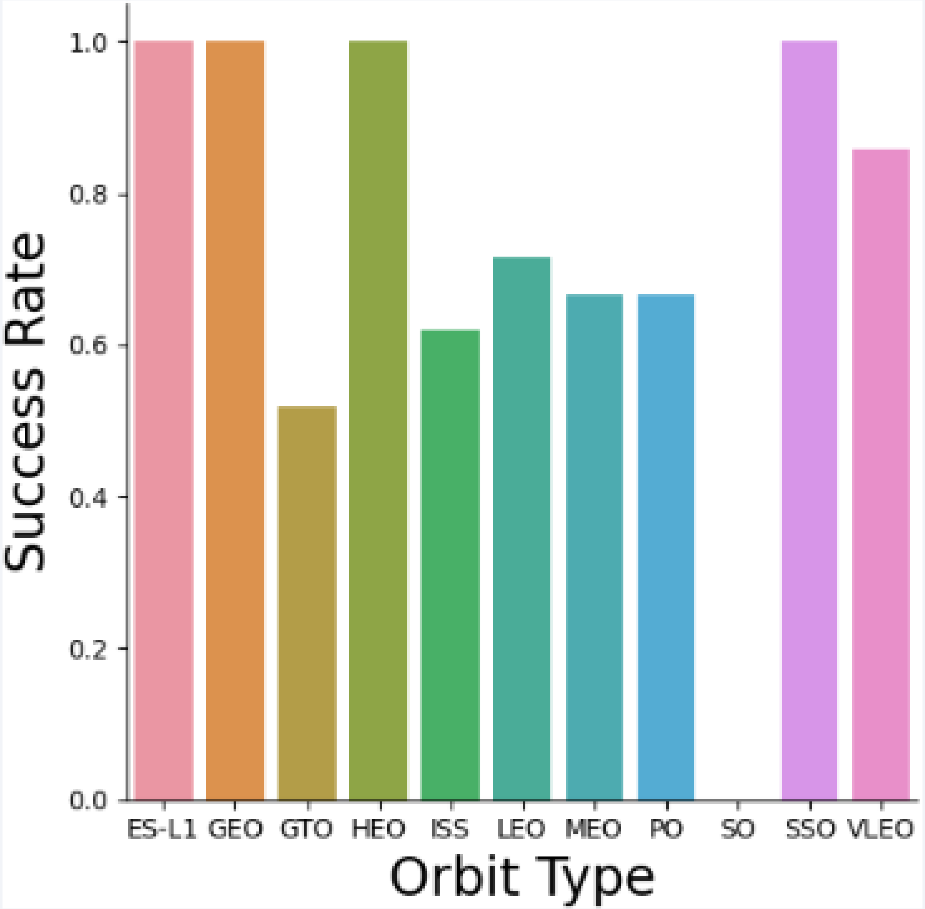
Payload vs. Launch Site

- Most launches with a payload greater than 7,000 kg were successful.
- VAFB SKC 4E has not launched anything greater than ~10,000 kg.
- Lower-mass payloads were launched from CCAFS SLC 40.

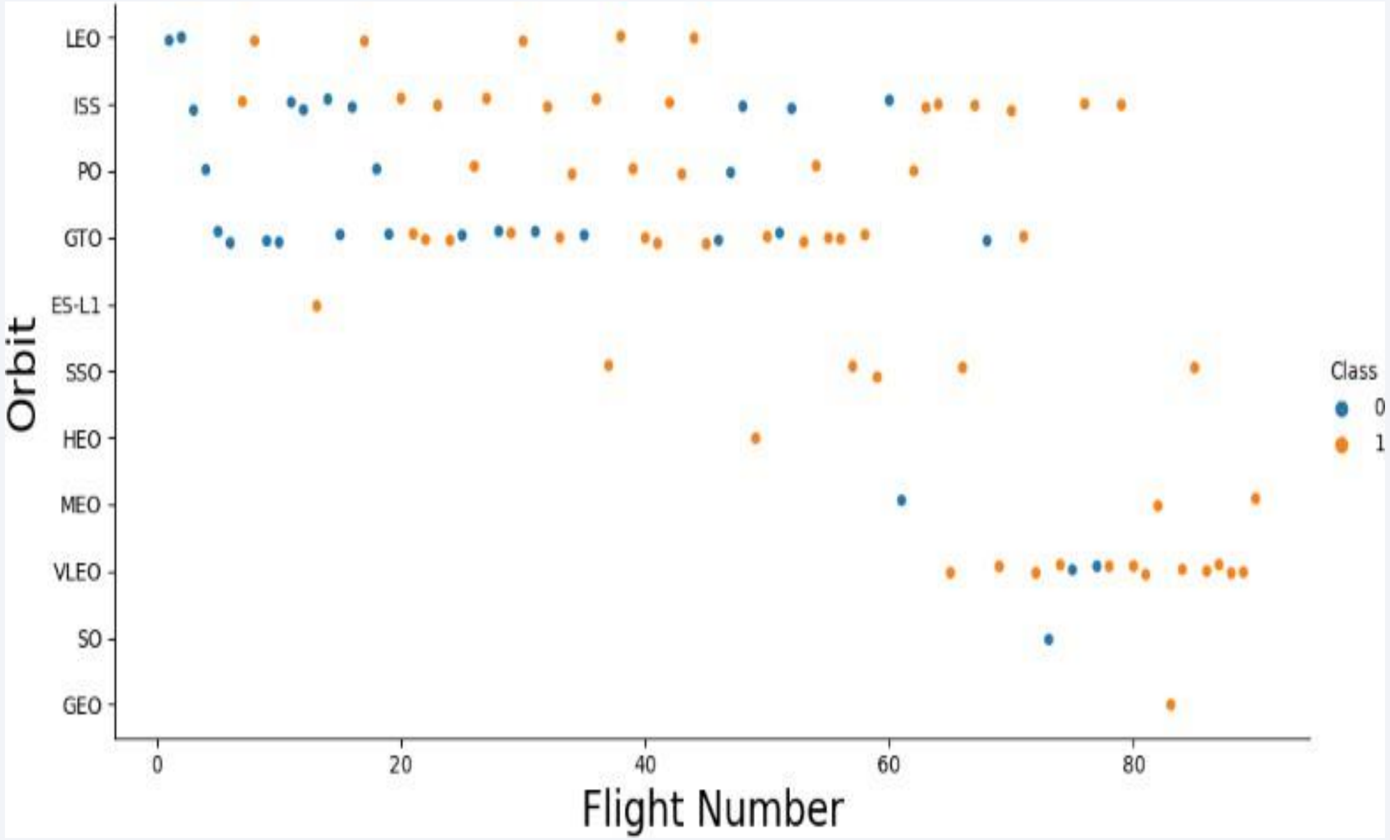


Success Rate vs. Orbit Type

- ES-L1, GEO, HEO and SSO had the highest success rates.



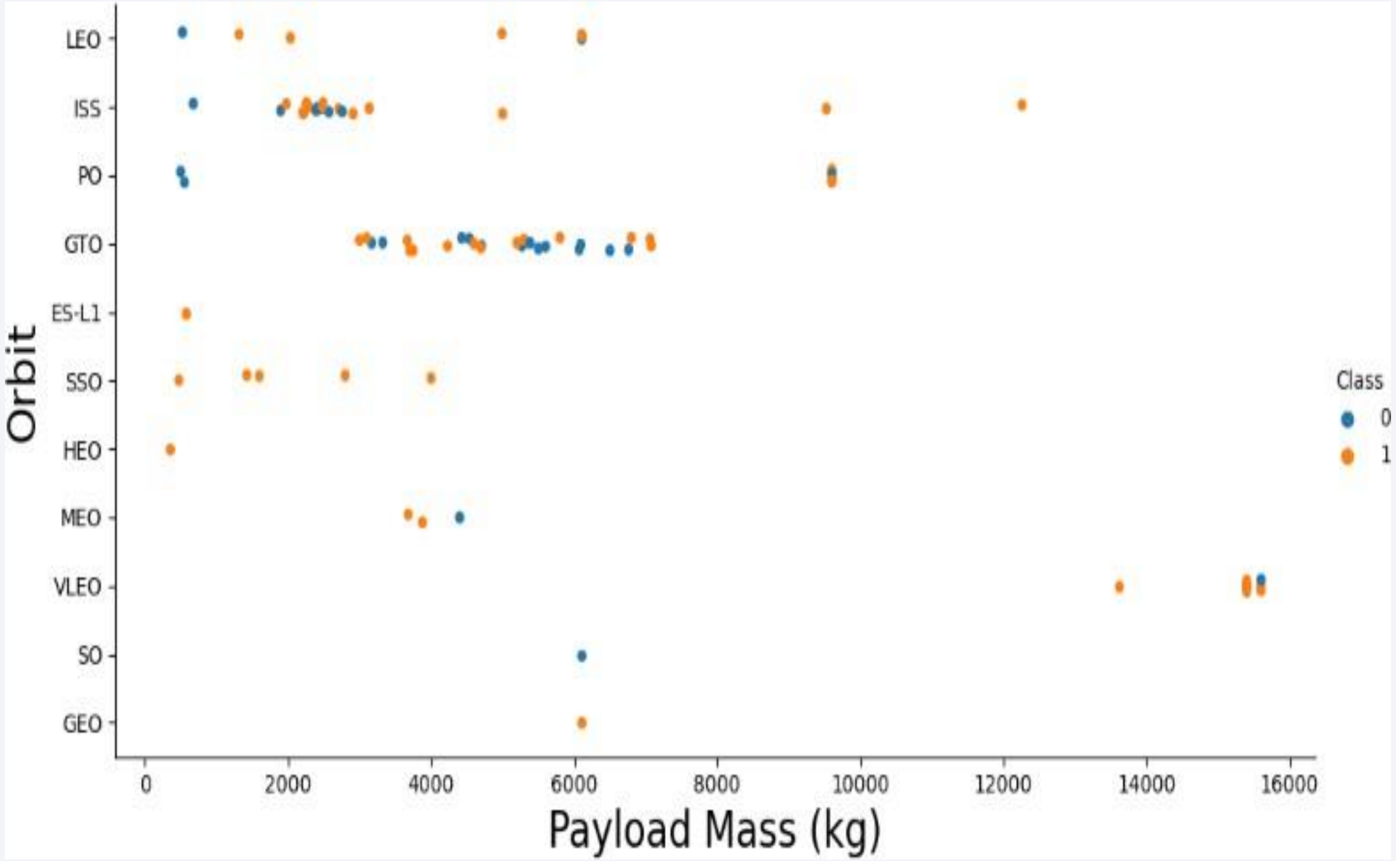
Flight Number vs. Orbit Type



- In recent years, there has been a trend observed in the use of VLEO orbit

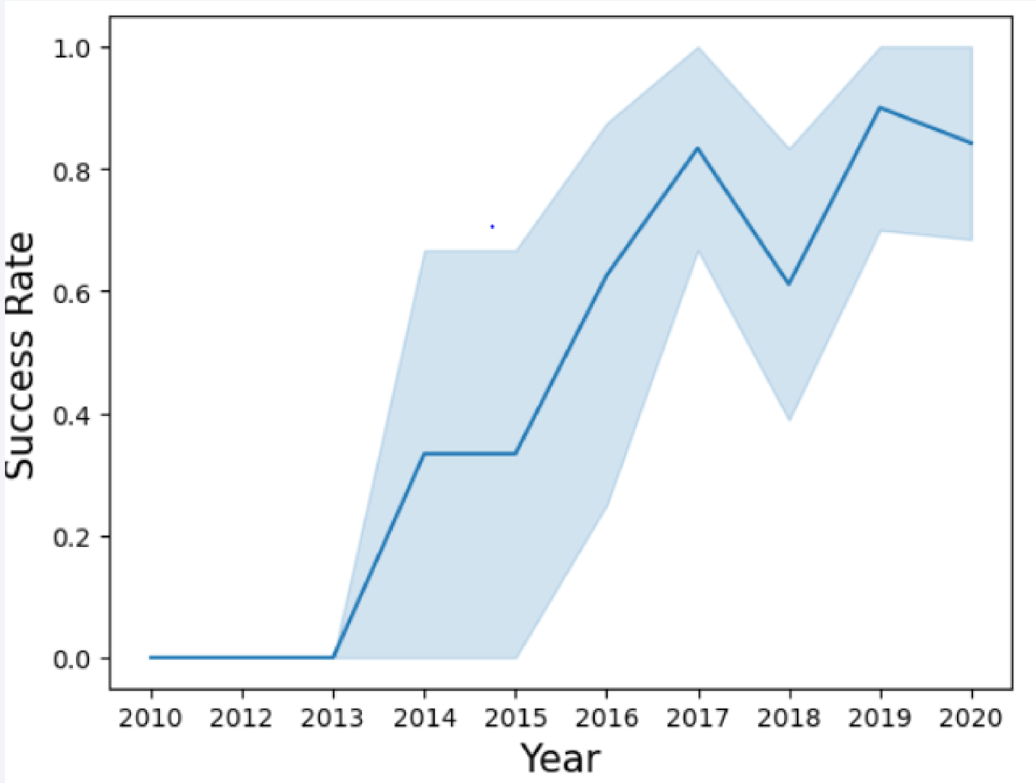
Payload vs. Orbit Type

- The GTO orbit has a mixed success rate, regardless of the payload mass.
- SSO orbit has the highest success rate.



Launch Success Yearly Trend

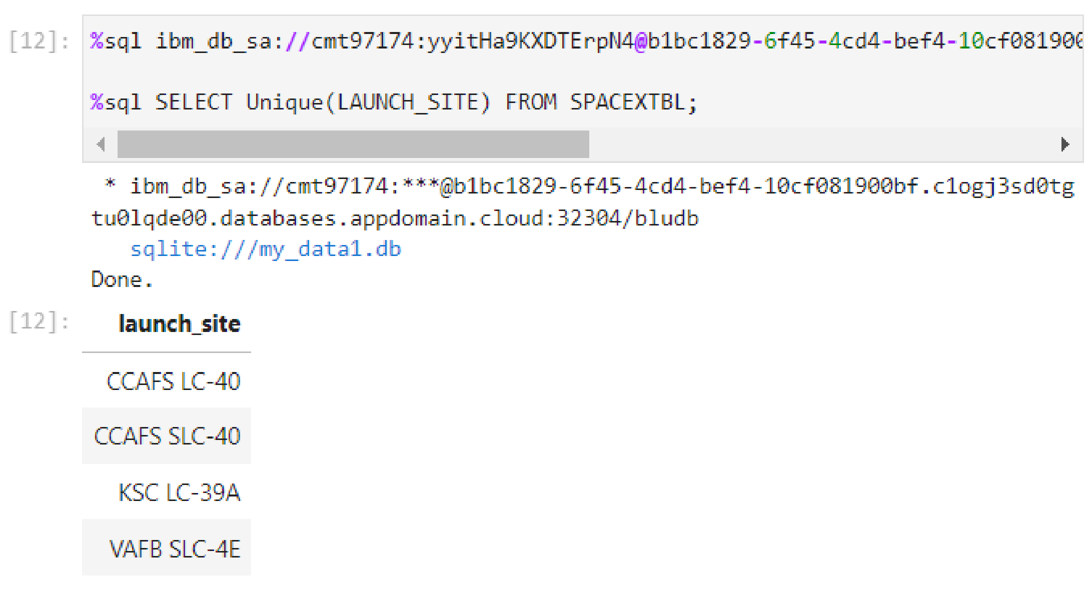
- The success rate improved from 2013-2017 and 2018-2019
- The success rate decreased from 2017-2018 and from 2019-2020
- Overall, the success rate has improved since 2013



All Launch Site Names

Launch sites

1. CCAFS LC-40
2. CCAFS SLC-40
3. KSC LC-39A
4. VAFB SLC-4E



```
[12]: %sql ibm_db_sa://cmt97174:yyitHa9KXDTErpN4@b1bc1829-6f45-4cd4-bef4-10cf08190

      %sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;
```

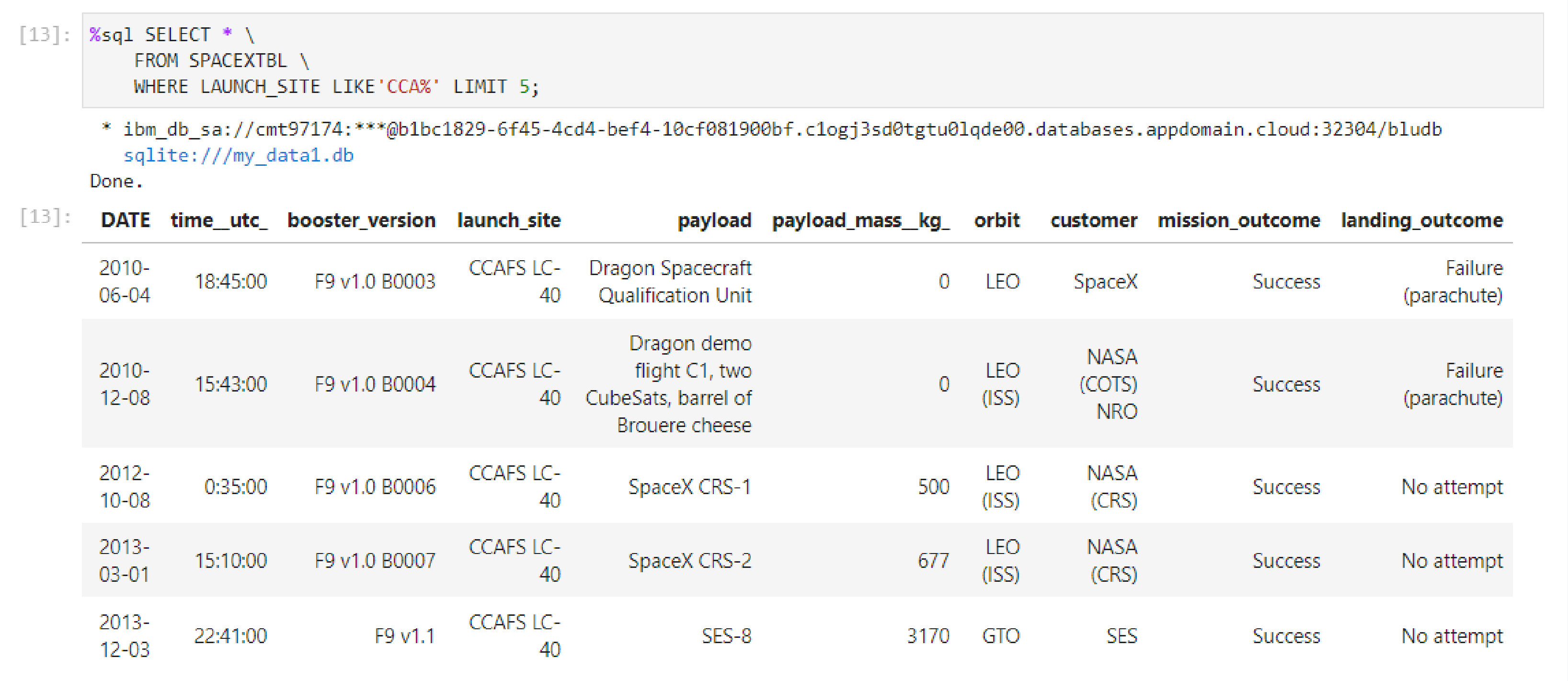
```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tg
tu0lqde00.databases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

[12]:

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

Launch Site Names Begin with 'CCA'

Records with Launch Site Starting with CCA



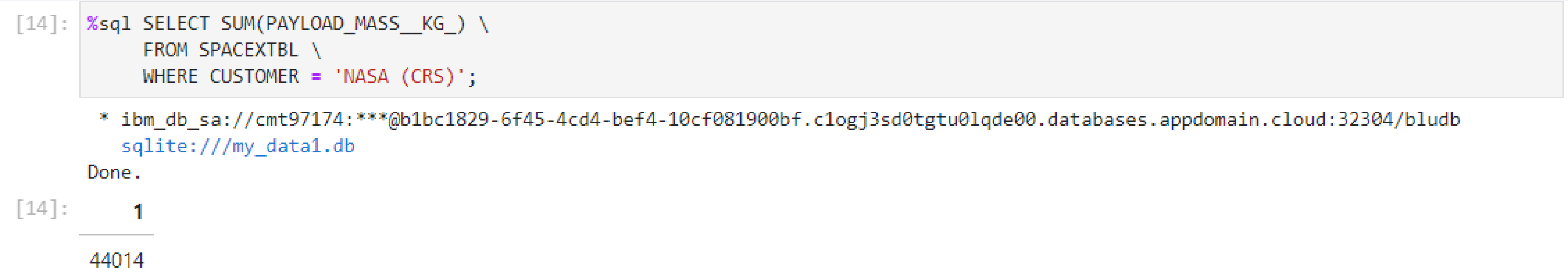
```
[13]: %sql SELECT * \
          FROM SPACEXTBL \
          WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;

 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

[13]:

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)
2012-10-08	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
2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Total Payload Mass

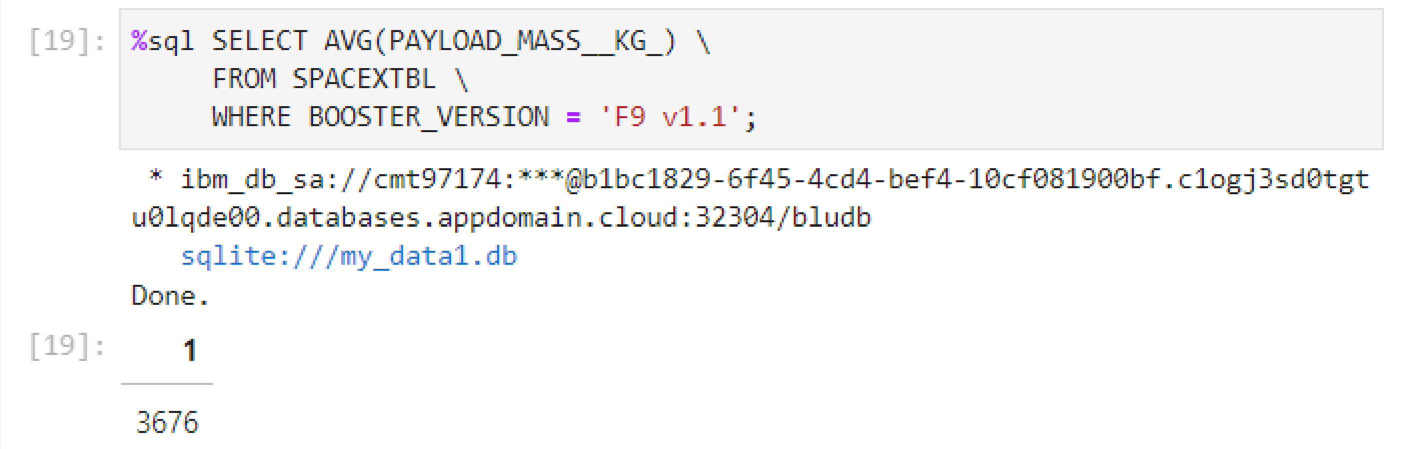
- 44014 kg (total) carried by boosters launched by NASA (CRS)



```
[14]: %sql SELECT SUM(PAYLOAD_MASS__KG_) \
           FROM SPACEXTBL \
           WHERE CUSTOMER = 'NASA (CRS)';

       * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
         sqlite:///my_data1.db
      Done.

[14]:      1
      44014
```

Average Payload Mass by F9 v1.1

- 3676 kg (average) carried by booster version F9 v1.1



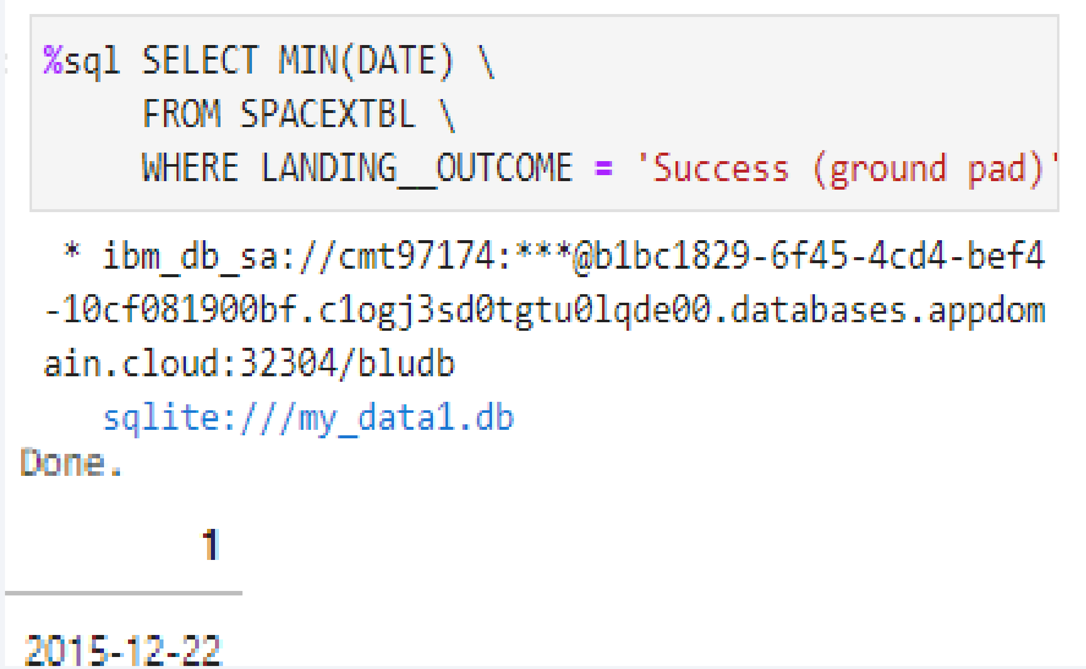
```
[19]: %sql SELECT AVG(PAYLOAD_MASS__KG_) \
           FROM SPACEXTBL \
           WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgt
u0lqde00.databases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

```
[19]:    1
       ----
       3676
```

First Successful Ground Landing Date

- First successful Landing in Ground Pad – 22/12/2015



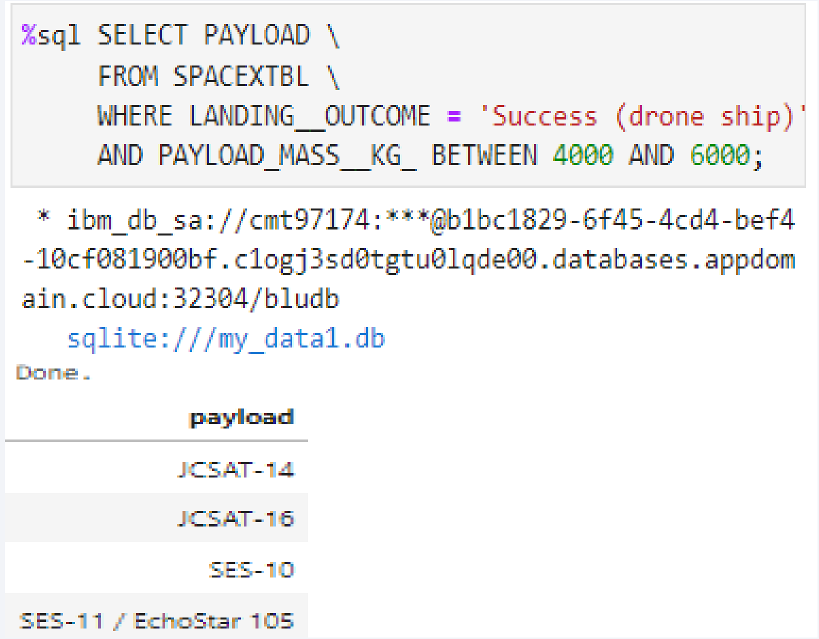
```
%sql SELECT MIN(DATE) \
     FROM SPACEXTBL \
     WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4
-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdom
ain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.

         1
----------
2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

Booster mass greater than 4,000 but less than 6,000

- JSCAT-14
- JSCAT-16
- SES-10
- SES-11 / EchoStar 105



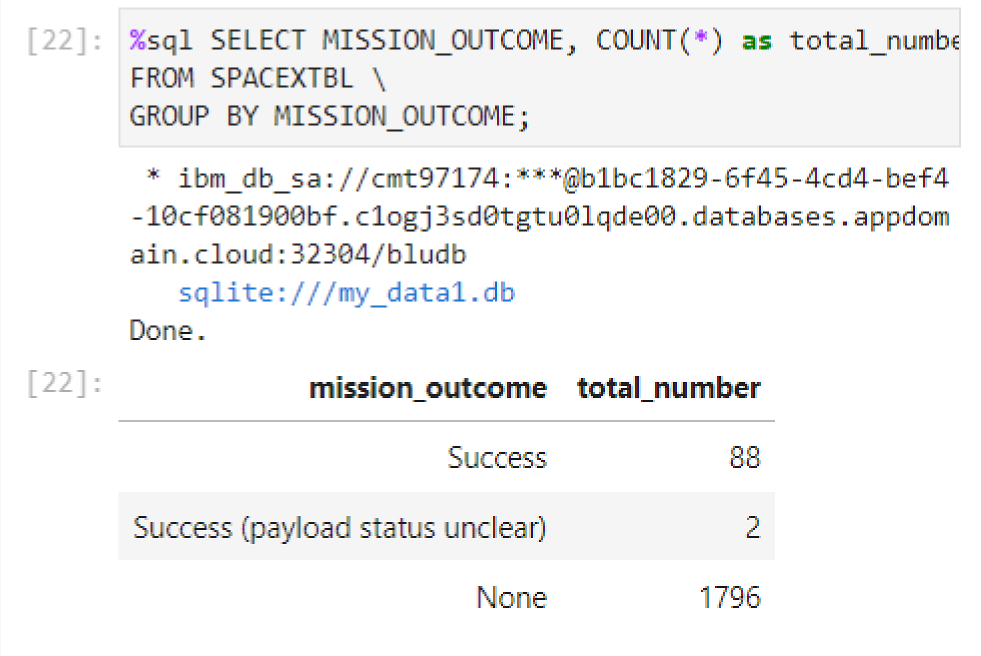
```
%sql SELECT PAYLOAD \
     FROM SPACEXTBL \
     WHERE LANDING__OUTCOME = 'Success (drone ship)'
     AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

- 88 Success
- 2 Success (payload status unclear)



```
[22]: %sql SELECT MISSION_OUTCOME, COUNT(*) as total_numbe
      FROM SPACEXTBL \
      GROUP BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4
-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdom
ain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

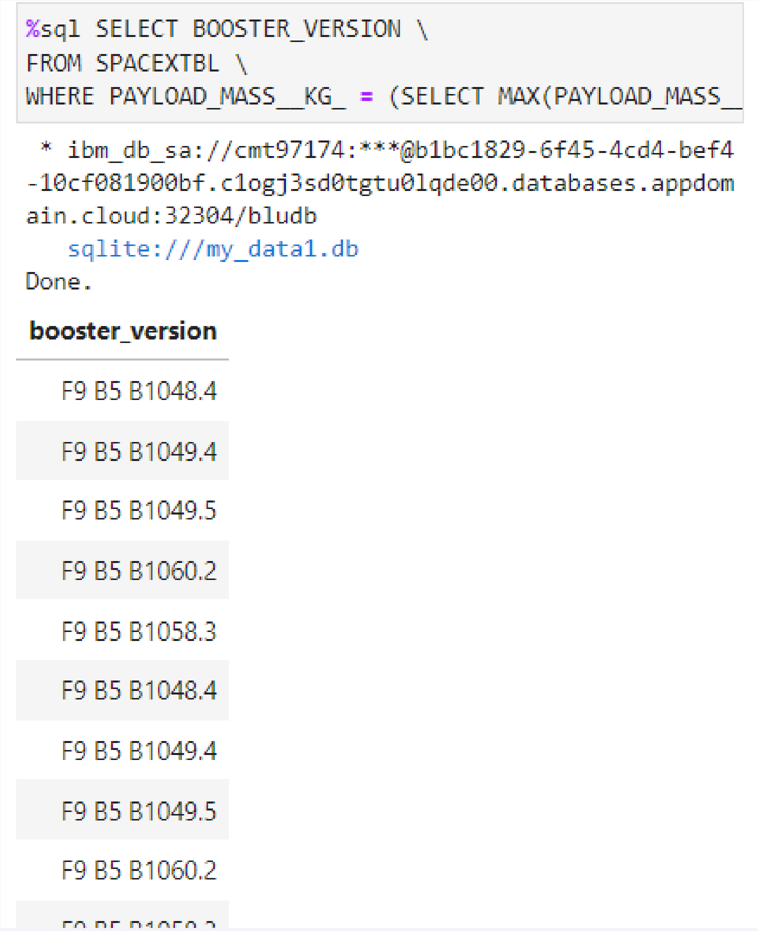
[22]:

mission_outcome	total_number
Success	88
Success (payload status unclear)	2
None	1796

Boosters Carried Maximum Payload

Booster which have carried the maximum payload mass

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



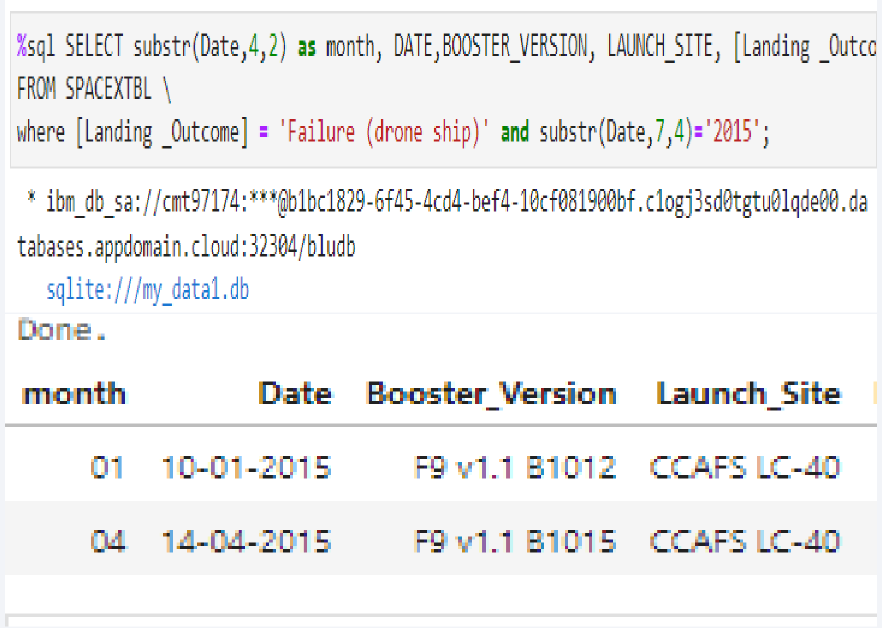
```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4
-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdom
ain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

2015 Launch Records

Showing month, date, booster version and launch site



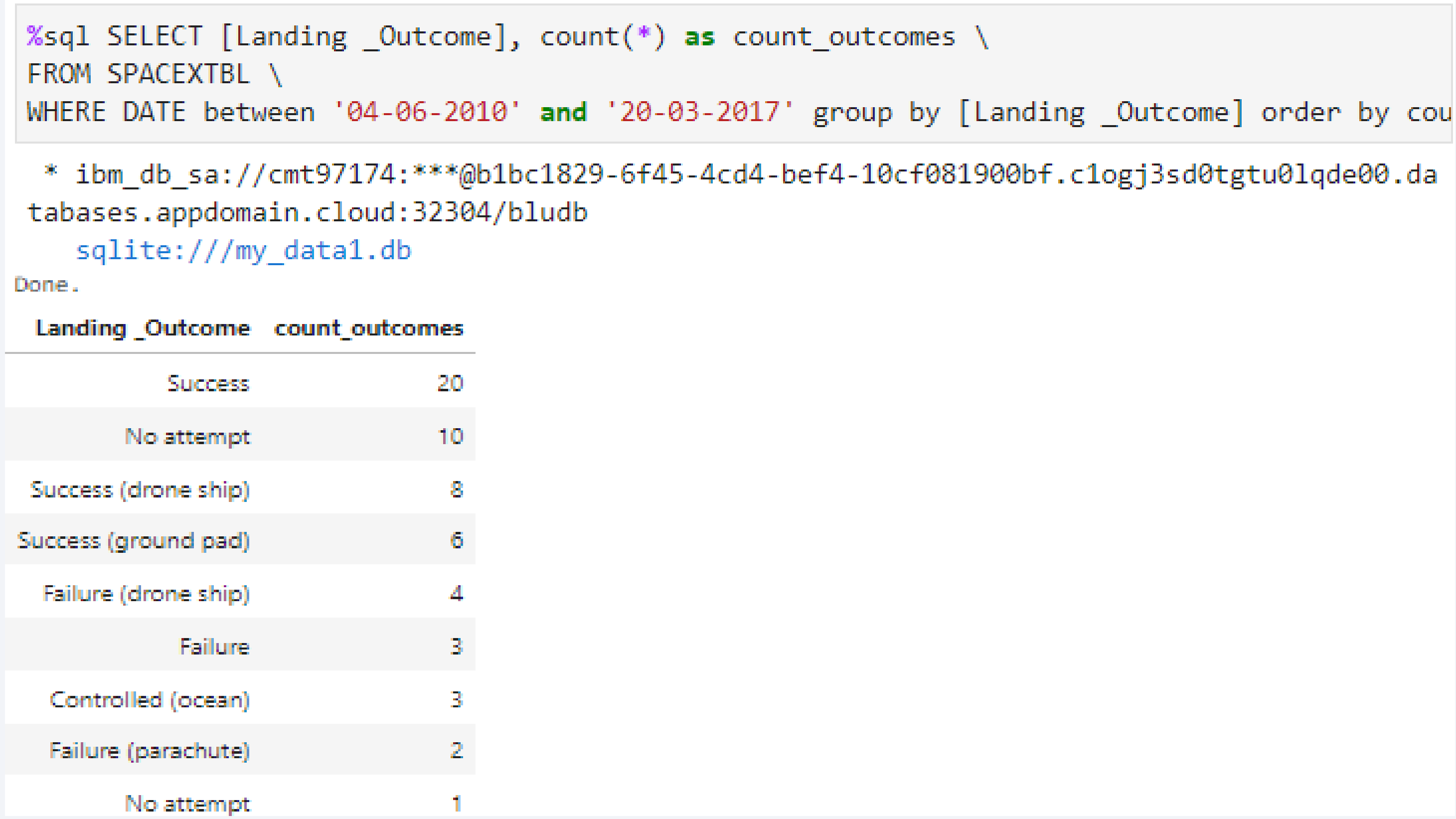
```
%sql SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outco
FROM SPACEXTBL \
where [Landing _Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.da
tabases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

month	Date	Booster_Version	Launch_Site
01	10-01-2015	F9 v1.1 B1012	CCAFS LC-40
04	14-04-2015	F9 v1.1 B1015	CCAFS LC-40

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

- Rank the of landing outcomes



```
%sql SELECT [Landing _Outcome], count(*) as count_outcomes \
FROM SPACEXTBL \
WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by cou
```

```
 * ibm_db_sa://cmt97174:***@b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.da
tabases.appdomain.cloud:32304/bludb
   sqlite:///my_data1.db
Done.
```

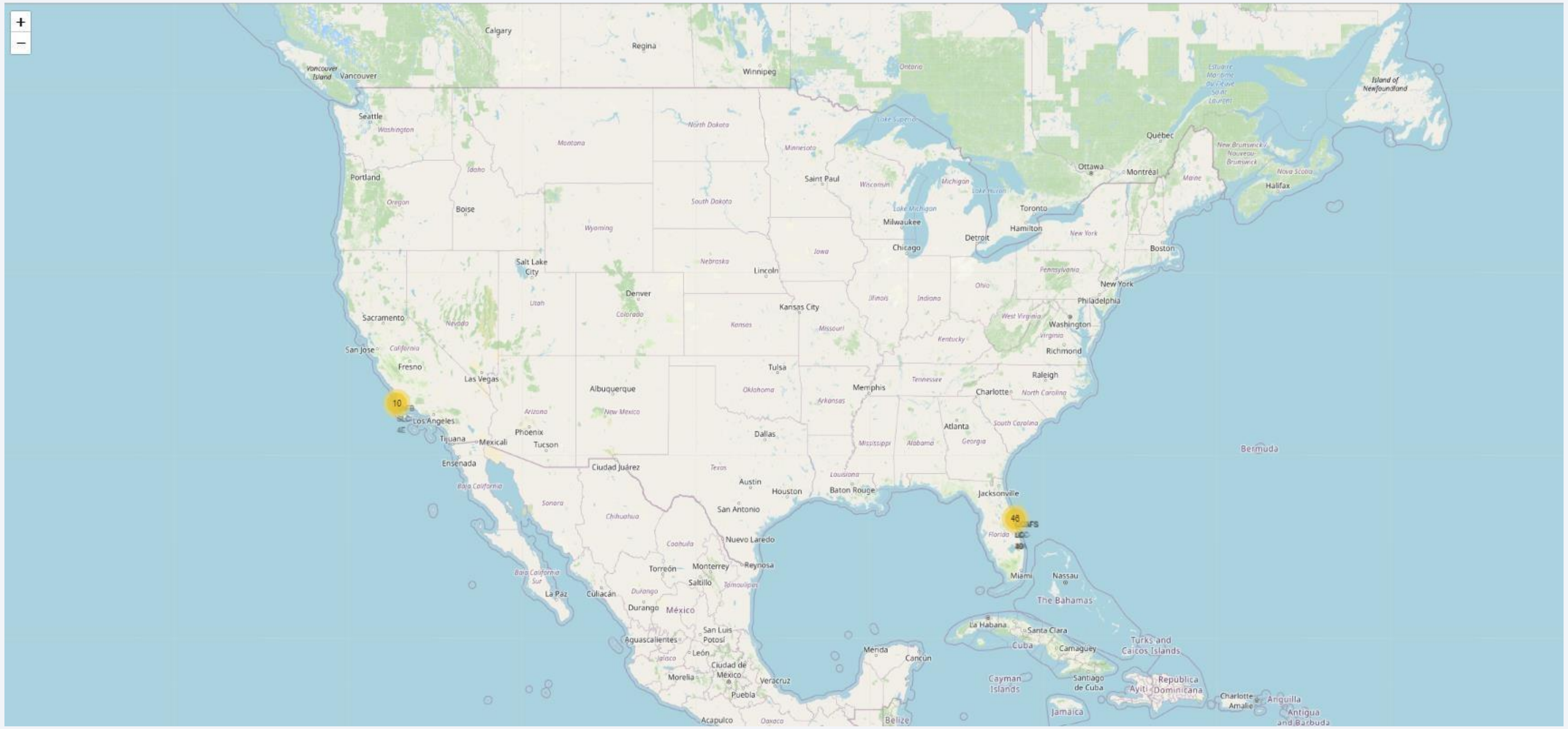
Landing _Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

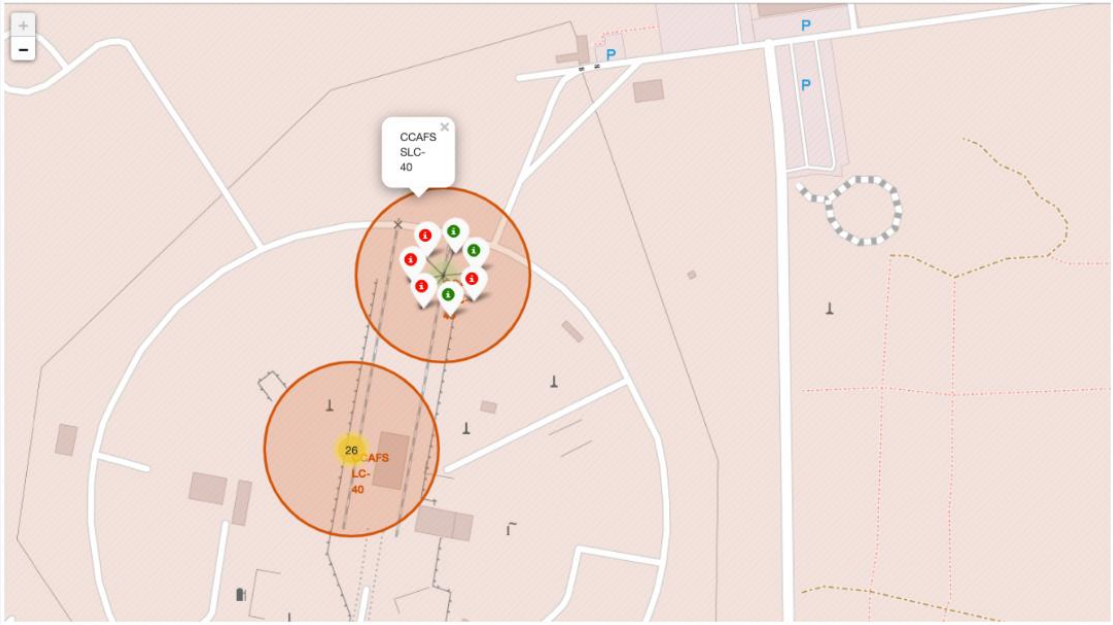
Launch Sites

Two main launch sites.



Launch outcomes

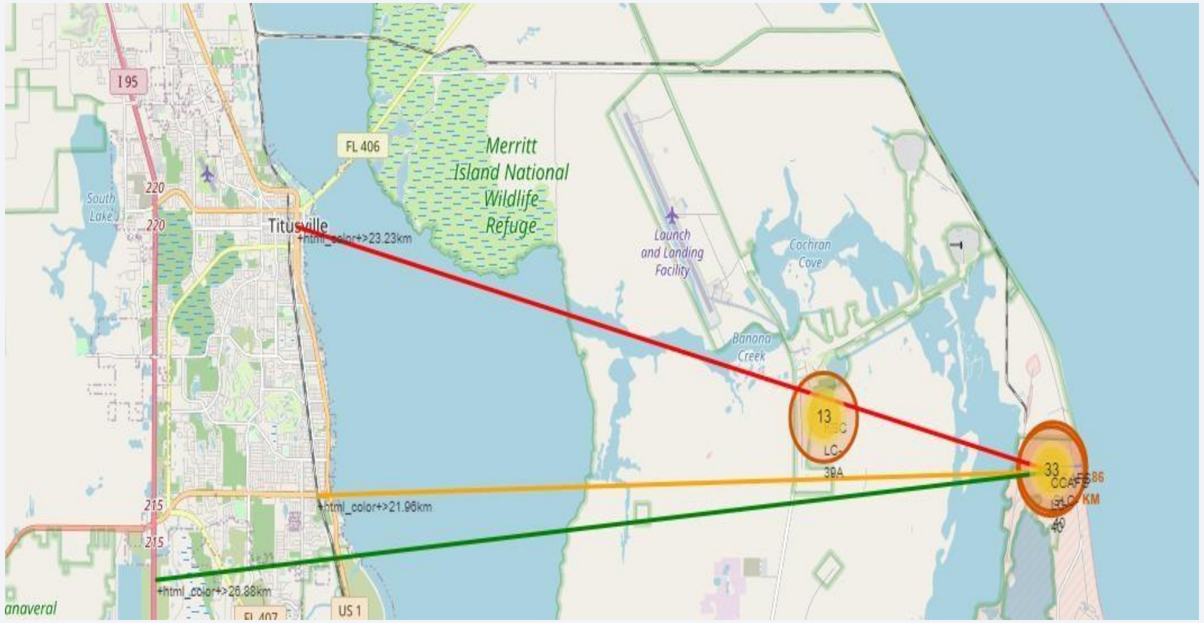
- Green markers for successful launches
- Red markers for unsuccessful launches
- Launch site CCAFS SLC-40 has a 3/7 success rate (42.9%)

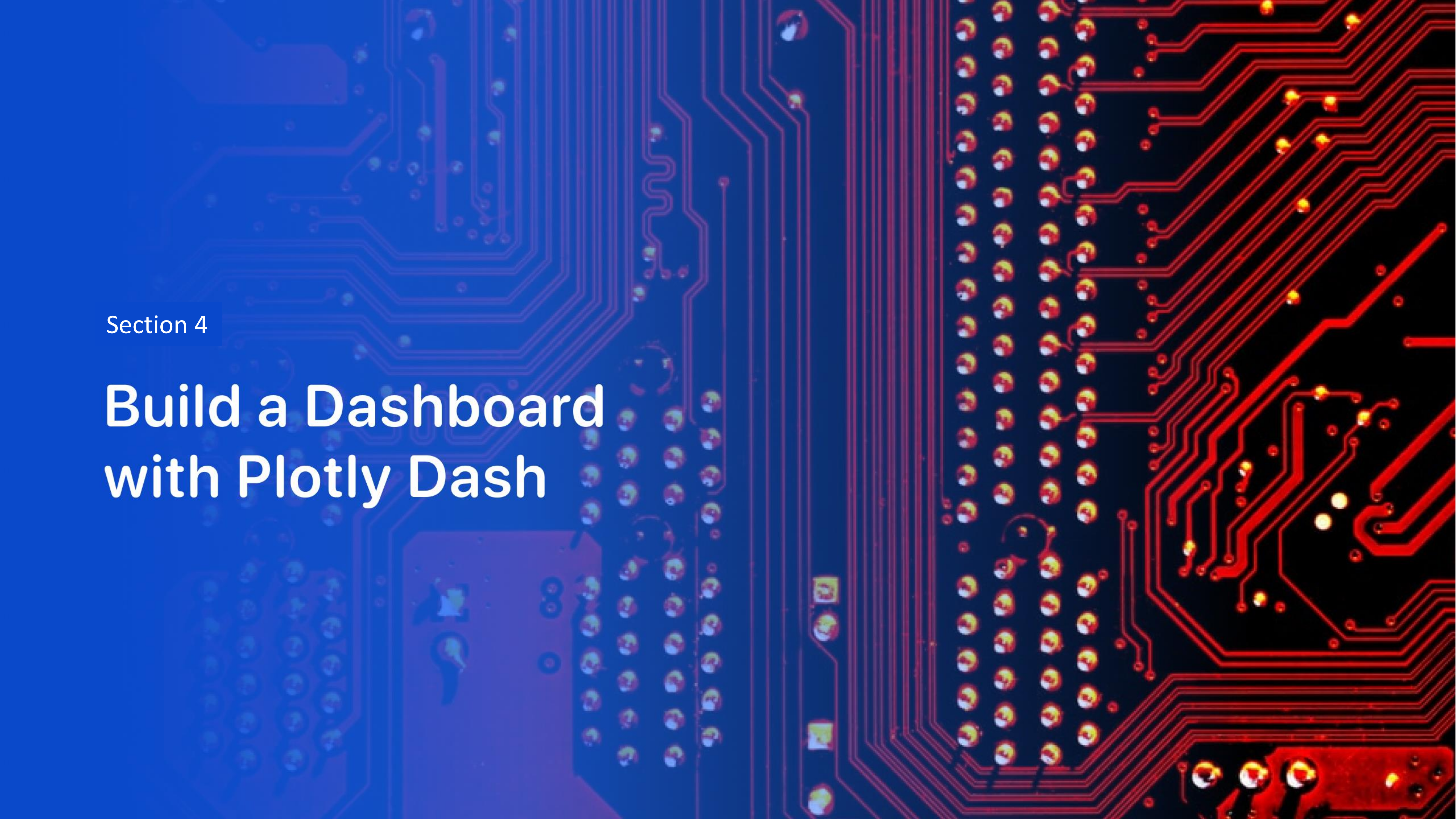


Distance to Proximities

CCAFS SLC-40

- 86 km from nearest coastline
- 21.96 km from nearest railway
- 23.23 km from nearest city
- 26.88 km from nearest highway



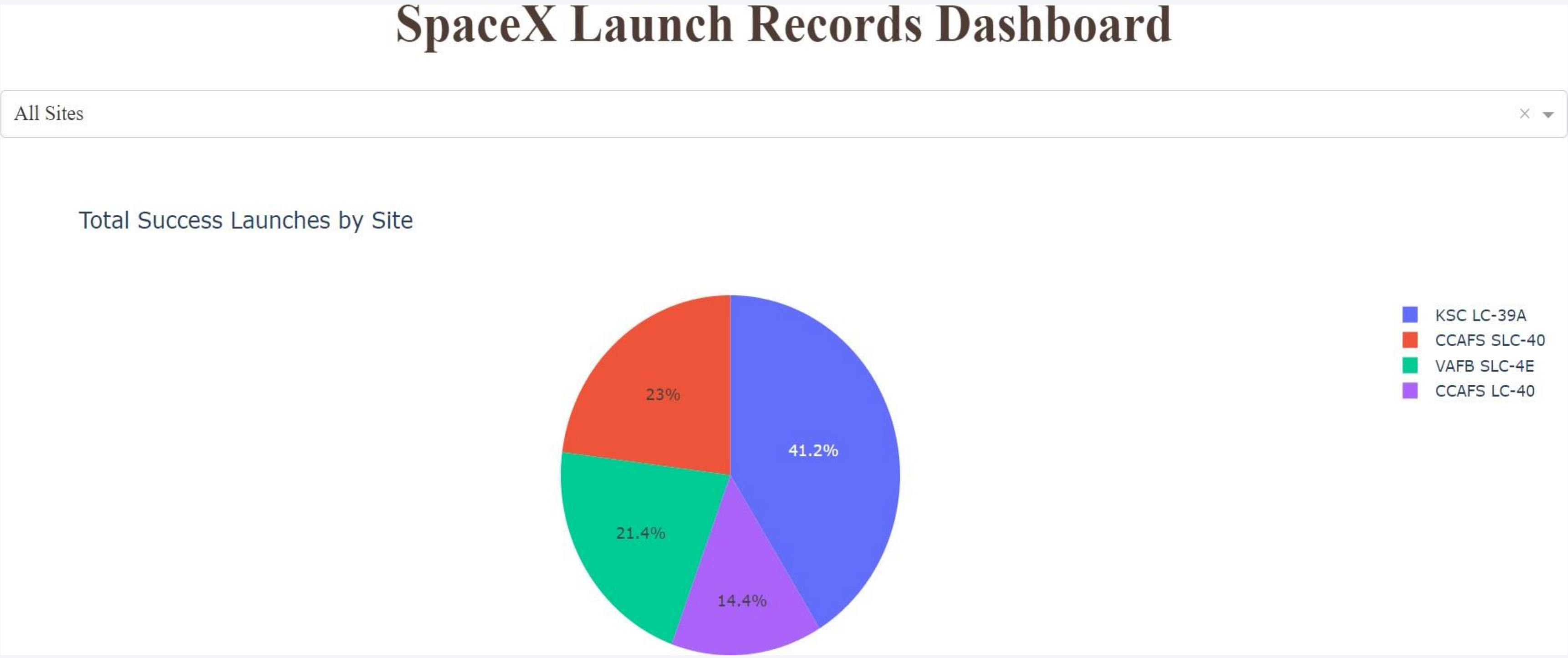


Section 4

Build a Dashboard with Plotly Dash

Launch success by sites

- KSC LC-39A has the most successful launches amongst launch sites (41.2%)

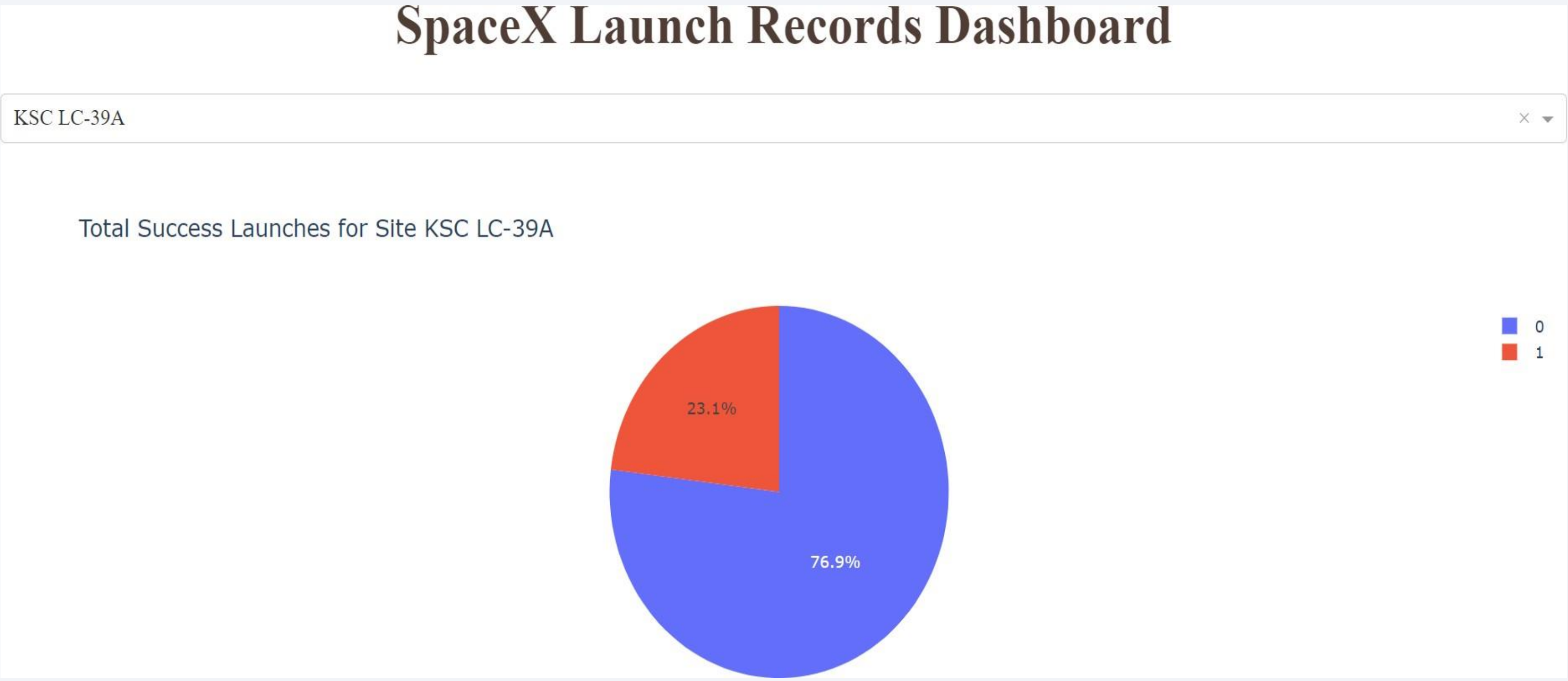


SpaceX Launch Records Dashboard

All Sites

Launch Success (KSC LC-29A)

- KSC LC-39A has the highest success rate amongst launch sites (76.9%)



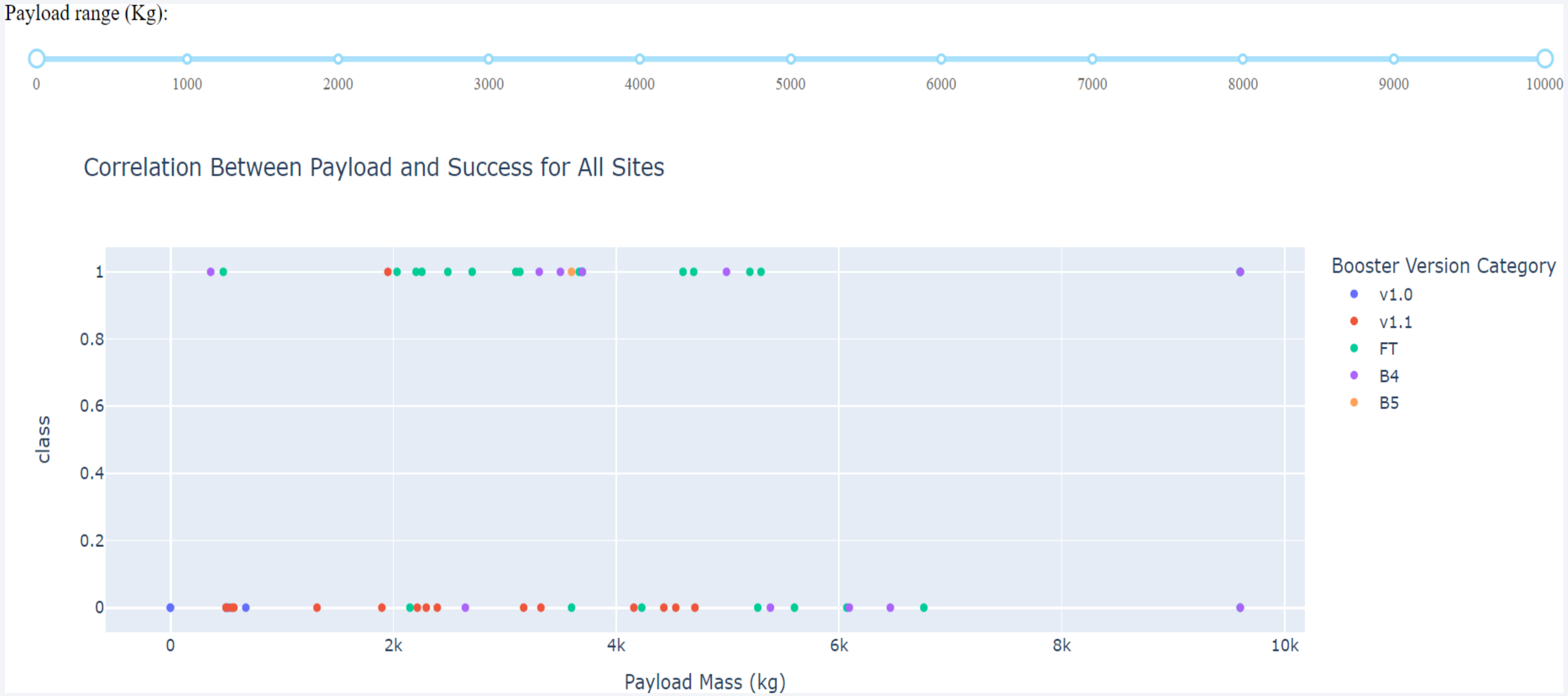
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site KSC LC-39A

Payload Mass and Success

- Payloads between 2,000 kg and 5,000 kg have the highest success rate

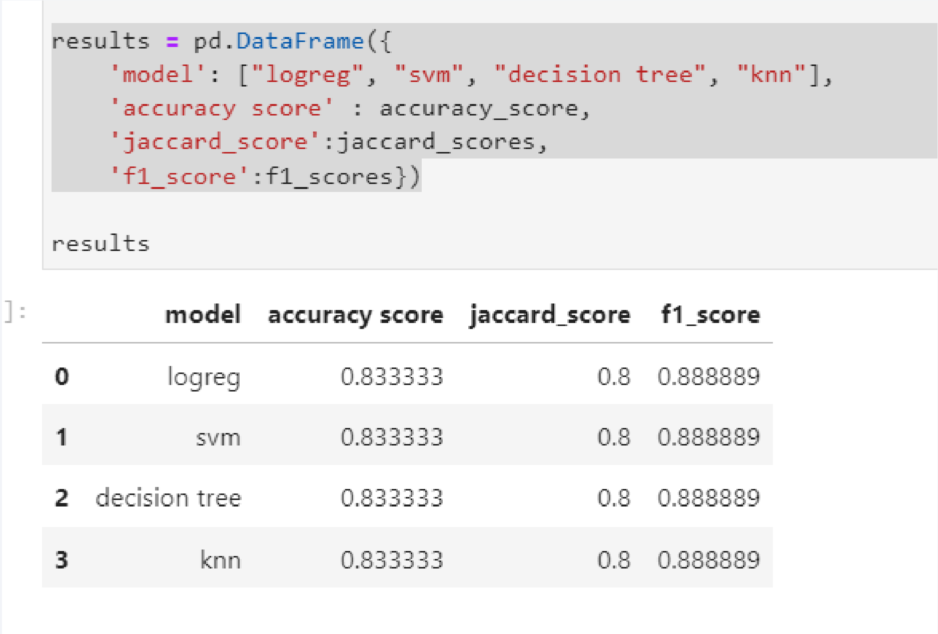


Section 5

Predictive Analysis (Classification)

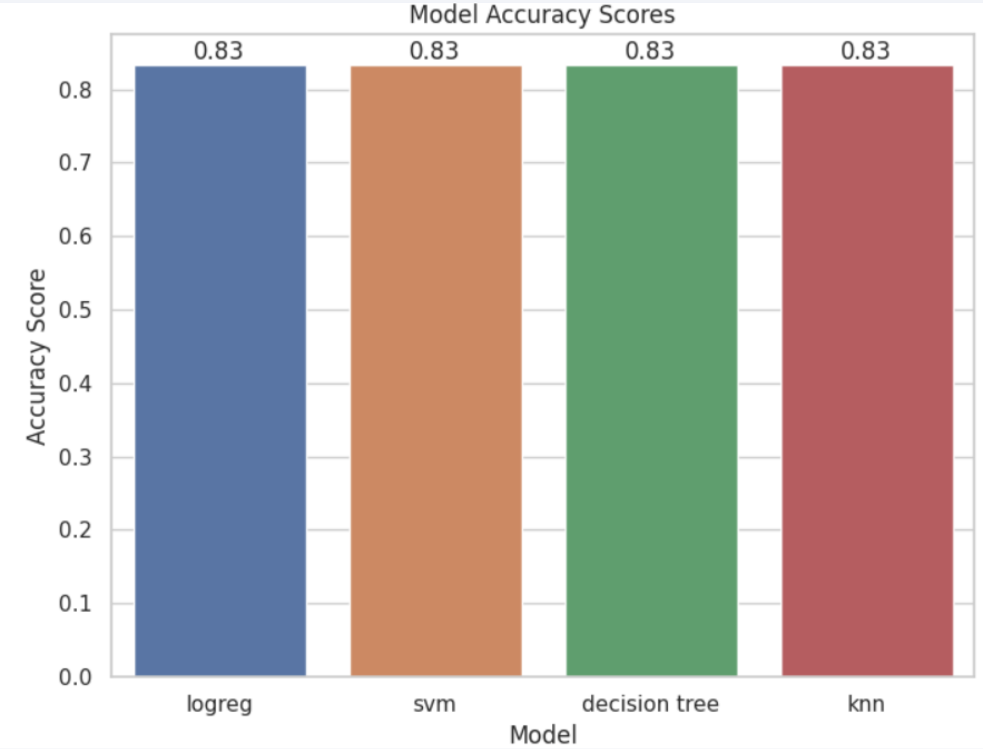
Classification Accuracy

- **All** the **models** performed at about the same level and had the **same accuracy, Jaccard and F1 scores.**



```
results = pd.DataFrame({
    'model': ["logreg", "svm", "decision tree", "knn"],
    'accuracy score' : accuracy_score,
    'jaccard_score':jaccard_scores,
    'f1_score':f1_scores})

results
```

	model	accuracy score	jaccard_score	f1_score
0	logreg	0.833333	0.8	0.888889
1	svm	0.833333	0.8	0.888889
2	decision tree	0.833333	0.8	0.888889
3	knn	0.833333	0.8	0.888889



Confusion Matrix

All confusion matrix outputs are identical in each method.

12 - True positive

3 - True negative

3 - False positive

0 - False Negative

Precision = TP / (TP + FP)

- 12 / 15 = .80

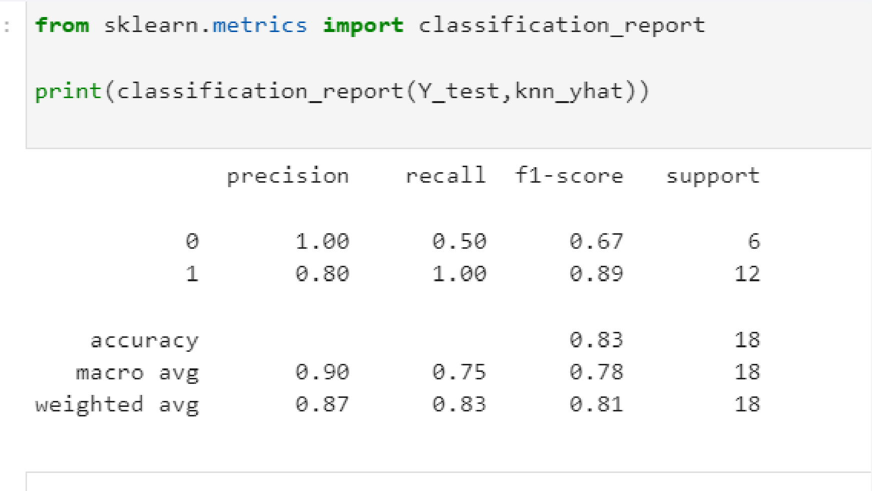
Recall = TP / (TP + FN)

- 12 / 12 = 1

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

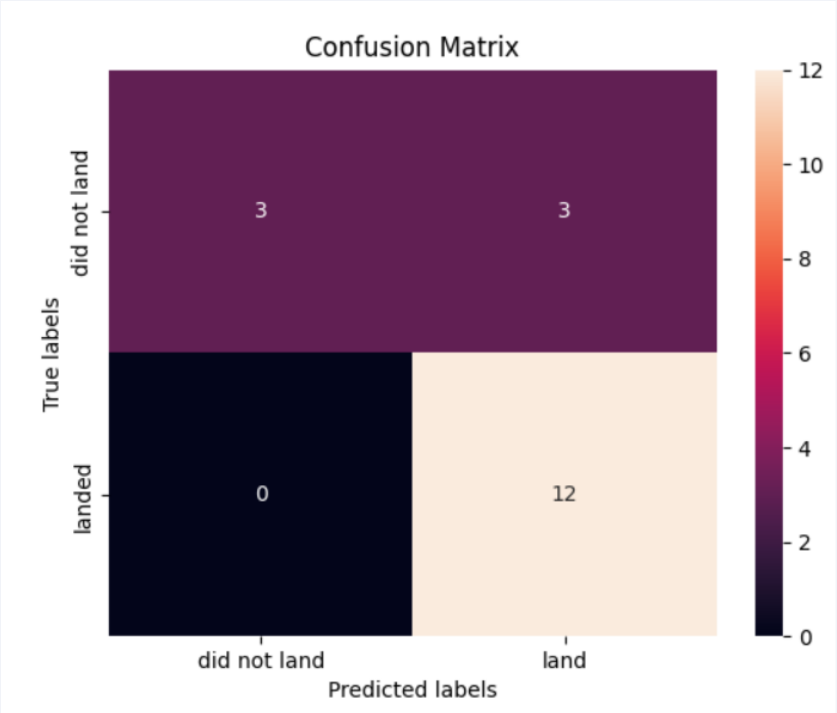
- 2 * (.8 * 1) / (.8 + 1) = .89

Accuracy = (TP + TN) / (TP + TN + FP + FN) = .833



```
from sklearn.metrics import classification_report

print(classification_report(Y_test,knn_yhat))
```

```
              precision    recall  f1-score   support

           0       1.00      0.50      0.67         6
           1       0.80      1.00      0.89        12

    accuracy                           0.83        18
   macro avg       0.90      0.75      0.78        18
weighted avg       0.87      0.83      0.81        18
```



Conclusions

- Logistic regression, support vector machine (SVM), decision tree and K-nearest neighbor (KNN) models had same accuracy levels (accuracy, F1 and Jaccard scores)
- All the launch sites are located close to the coast.
- Launch success has increased over time.
- KSC LC-39A has the highest success rate among the launch sites, with a 100% success rate for launches weighing less than 5,500 kg.
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.
- Across all launch sites, the higher the payload mass (in kg), the higher the success rate.

Appendix

- GitHub URL of the overall project - https://github.com/MadhawaHulangamuwa/IBM-Applied-Data-Science-Capstone.git

Thank you!