IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mahshdid Kalantari
17/12/2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- **Summary of methodologies**
  1. Data Collection through API
  2. Data Collection with Web Scraping
  3. Data Wrangling
  4. Exploratory Data Analysis with SQL
  5. Exploratory Data Analysis with Data Visualization
  6. Interactive Visual Analytics with Folium
  7. Machine Learning Prediction
- **Summary of all results**
  1. Exploratory Data Analysis result
  2. Interactive analytics in screenshots
  3. Predictive Analytics result

Introduction

Project Background and Context

SpaceX prominently features Falcon 9 rocket launches on its website, boasting a cost of $62 million, a significantly lower figure compared to other providers whose costs can exceed $165 million per launch. The key factor contributing to these savings is SpaceX's ability to reuse the first stage of the rocket. Consequently, understanding whether the first stage will successfully land is crucial in estimating the overall cost of a launch. This knowledge is particularly valuable for competing companies seeking to bid against SpaceX for rocket launch contracts. The primary objective of this project is to develop a machine learning pipeline capable of predicting the successful landing of the first stage.

Introduction

Problems to Address

1. Determining Factors for Successful Landing:
    1. Identifying the key factors that influence the successful landing of the rocket's first stage.
    2. Analyzing the interplay between various features to comprehend their collective impact on the success rate of a landing.
2. Understanding Interactions Among Features:
    1. Investigating the relationships and interactions among different features to gain insights into how they contribute to or hinder the likelihood of a successful landing.
3. Operating Conditions for Success:
    1. Defining the specific operating conditions necessary to ensure a successful landing program.
    2. Examining the environmental, technical, and procedural requirements that optimize the probability of a secure and effective rocket stage landing.

Section 1

Methodology

Methodology

Executive Summary

- **Data Collection:** Data was gathered using SpaceX API and web scraping from Wikipedia.
- **Perform Data Wrangling:** A comprehensive data wrangling process was executed to ensure data cleanliness and reliability.
- **Apply One-Hot Encoding:** One-hot encoding was applied to categorical features to enhance their representation in subsequent analytical processes.
- **Perform Exploratory Data Analysis (EDA):** Exploration of data patterns was conducted using visualization techniques and SQL queries.
- **Perform Interactive Visual Analytics:** Interactive visual analytics was executed using Folium and Plotly Dash, providing dynamic and user-friendly representations.
- **Perform Predictive Analysis**: The project involved predictive analysis through the implementation of classification models.
- **Build, Tune, and Evaluate Classification Models:** The process included building, tuning, and evaluating classification models to ensure accuracy and robustness.

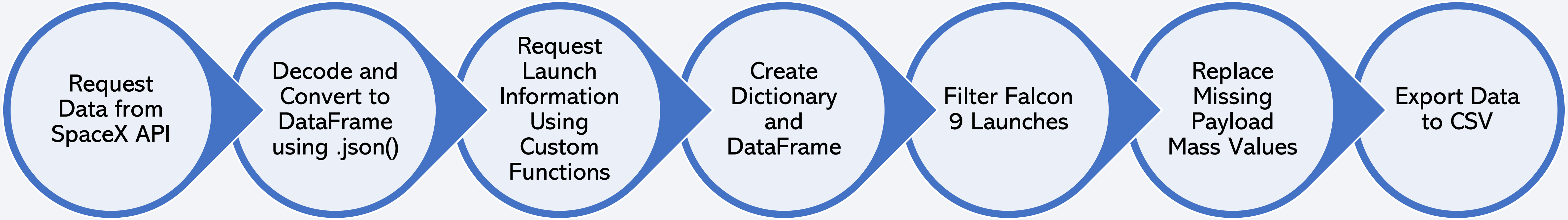
Data Collection

1. Initiate Collection: Start by collecting data through a GET request to the SpaceX API.
2. Decode and Transform: Decode the API response as JSON and convert it into a pandas dataframe using .json_normalize().
3. Clean Data: Address missing values and perform necessary data cleaning.
4. Web Scraping: Utilize BeautifulSoup for web scraping from Wikipedia to extract Falcon 9 launch records.
5. Extract and Parse: Extract launch records as an HTML table, then parse and convert it into a pandas dataframe.

Data Collection – SpaceX API

The link to the notebook is:

https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/jupyter_labs_spacex_data_collection_api%20.ipynb



Request Data from SpaceX API

Decode and Convert to DataFrame using .json()

Request Launch Information Using Custom Functions

Create Dictionary and DataFrame

Filter Falcon 9 Launches

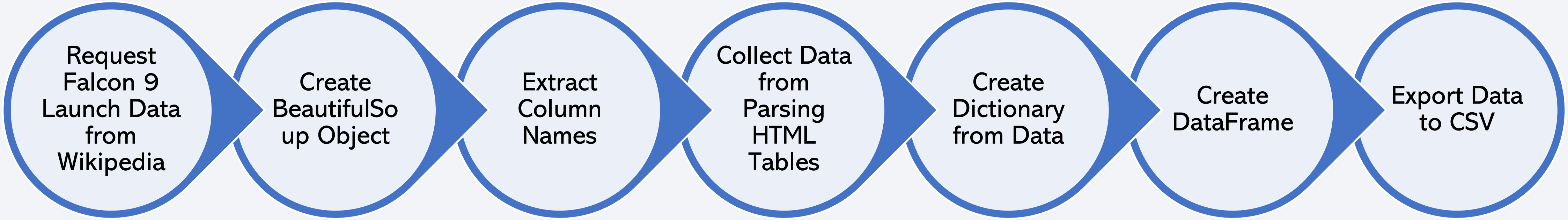
Replace Missing Payload Mass Values

Export Data to CSV

Data Collection - Scraping

The link to the notebook is:

https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/jupyter_labs_webscraping.ipynb



Request Falcon 9 Launch Data from Wikipedia

Create BeautifulSoup Object

Extract Column Names

Collect Data from Parsing HTML Tables

Create Dictionary from Data

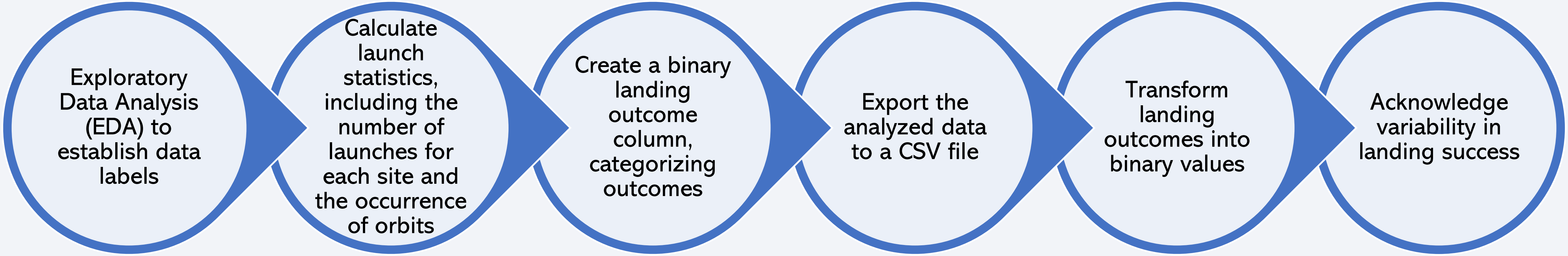
Create DataFrame

Export Data to CSV

Data Wrangling

The link to the notebook is:

https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/labs_jupyter_spacex_Data_wrangling.ipynb



Exploratory Data Analysis (EDA) to establish data labels

Calculate launch statistics, including the number of launches for each site and the occurrence of orbits

Create a binary landing outcome column, categorizing outcomes

Export the analyzed data to a CSV file

Transform landing outcomes into binary values

Acknowledge variability in landing success

EDA with Data Visualization

- **Charts for Exploration:**

1. Flight Number vs. Payload
2. Flight Number vs. Launch Site
3. Payload Mass (kg) vs. Launch Site
4. Payload Mass (kg) vs. Orbit type

- **EDA with Visualization**:

1. Utilized scatter plots to examine relationships, assessing potential relevance for machine learning if a pattern exists.
2. Employed bar charts to compare discrete categories, illustrating relationships between categories and measured values.

The link to the notebook is:

https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX IBM/blob/main/jupyter_labs_eda_dataviz.ipynb

EDA with SQL

- **Data Loading:** Loaded the SpaceX dataset into a PostgreSQL database directly within the Jupyter notebook.
- **EDA with SQL:** Utilized SQL queries for Exploratory Data Analysis (EDA) to gain insights from the data.
- **Extracted information:**

1. Names of unique launch sites in the space mission.
2. Total payload mass carried by boosters launched by NASA (CRS).
3. Average payload mass carried by booster version F9 v1.1.
4. Total number of successful and failed mission outcomes.
5. Details of failed landing outcomes on drone ships, including booster version and launch site names.

The link to the notebook is:

https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/jupyter_labs_eda_sql_coursera_sqllite.ipynb

Build an Interactive Map with Folium

Mapping Launch Sites:

- Marked all launch sites on a Folium map.
- Added map objects like markers, circles, and lines to visually represent the success or failure of launches for each site.

Outcome Classification:

- Assigned launch outcomes (failure or success) to classes 0 and 1, respectively.

Marker Clusters and Success Rate:

- Utilized color-labeled marker clusters to identify launch sites with relatively high success rates.

Distance Calculations:

- Calculated distances between launch sites and their proximities.

Spatial Analysis Questions Answered:

- Explored whether launch sites are near railways, highways, and coastlines.
- Investigated if launch sites maintain a certain distance from cities.

The link to the notebook is:
https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Interactive Dashboard:

- Developed an interactive dashboard using Plotly Dash for data visualization.

- Utilized pie charts to visually represent the distribution of total launches by specific launch sites.
- Outcome and Payload Mass Relationship:

- Created scatter graphs to explore the relationship between launch outcomes and payload mass (Kg) across different booster versions.

The link to the notebook is:
https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/SpaceX_Interactive_Visual_Analytics_Plotly.ipynb

Predictive Analysis (Classification)

- **Data Preparation:** Created a NumPy array from the "Class" column.
- **Standardization:** Standardized the data using StandardScaler.
- **Data Splitting:** Split the data into training and testing sets using train_test_split.
- **Parameter Optimization:** Employed GridSearchCV with cv=10 for parameter optimization.
- **Algorithm Evaluation:** Applied GridSearchCV on various algorithms, including Logistic Regression, Support Vector Machine, Decision Tree, and K-Nearest Neighbor.
- **Performance Metrics:** Calculated accuracy on the test data using .score() for all models.
- **Confusion Matrix:** Assessed the confusion matrix for each model.
- **Model Selection:** Identified the best model based on Jaccard Score, F1 Score, and Accuracy.

The link to the notebook is:
https://github.com/Kalmahsh/Data-Science-Capstone-SpaceX-IBM/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20.ipynb

Results

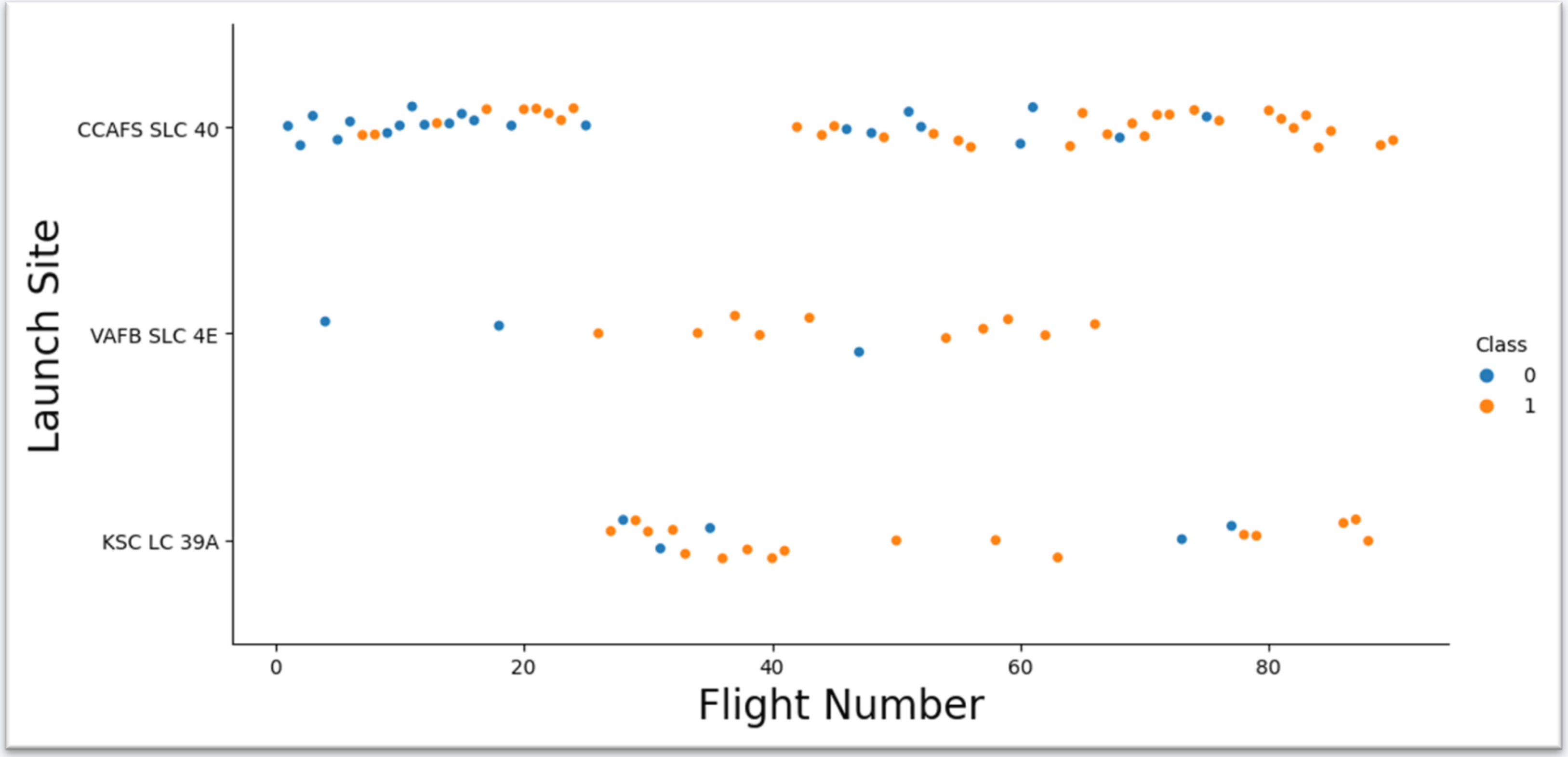
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

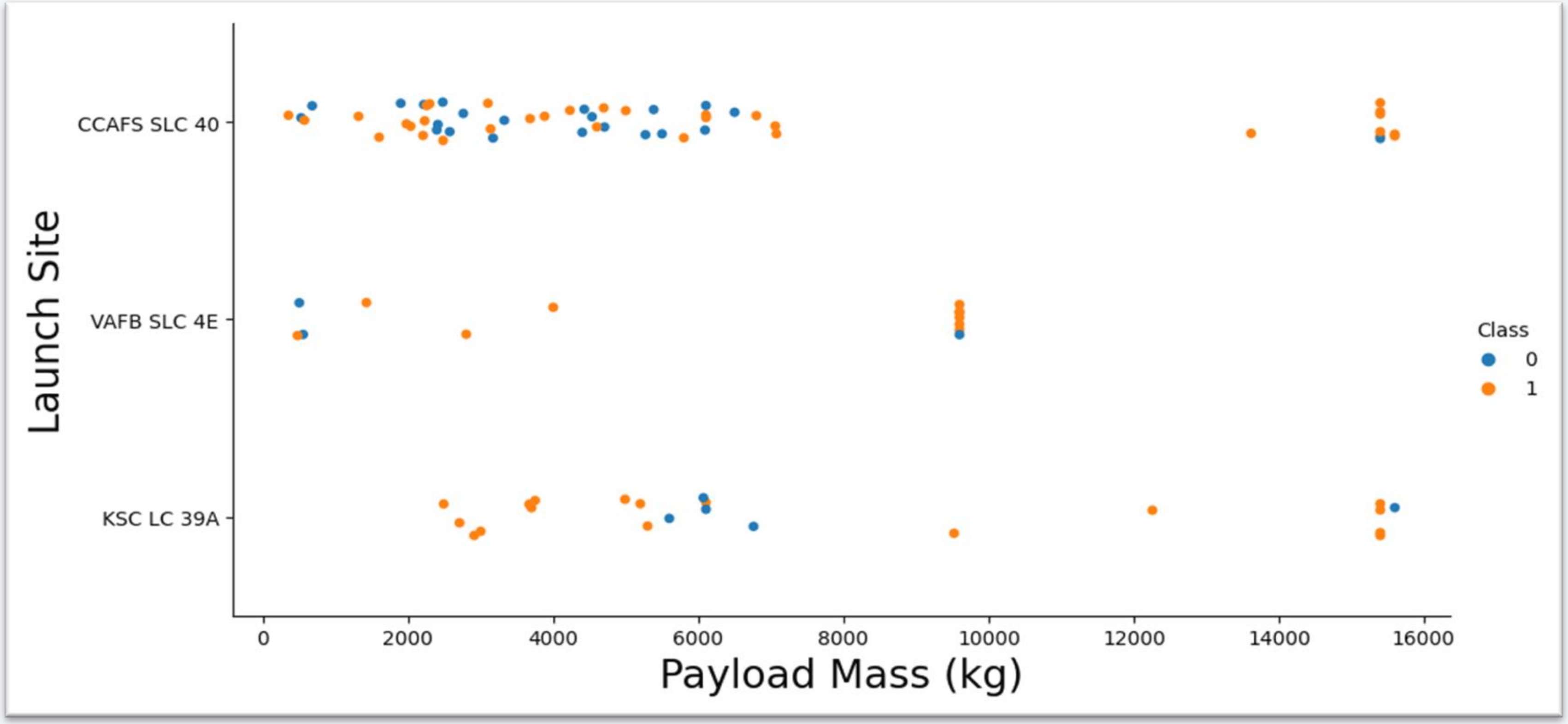
Flight Number vs. Launch Site

- Approximately half of the launches originated from the CCAFS SLC 40 launch site.
- Launch sites such as VAFB SLC 4E and KSC LC 39A showed higher success rates.
- From these observations, it can be inferred that newer launches tend to have a higher success rate.



Payload vs. Launch Site

- Most launces with a payload greater than 7,000 kg were successful
- KSC LC 39A has a 100% success rate for launches less than 5,500 kg
- VAFB SKC 4E has not launched anything greater than ~10,000 kg



Success Rate vs. Orbit Type

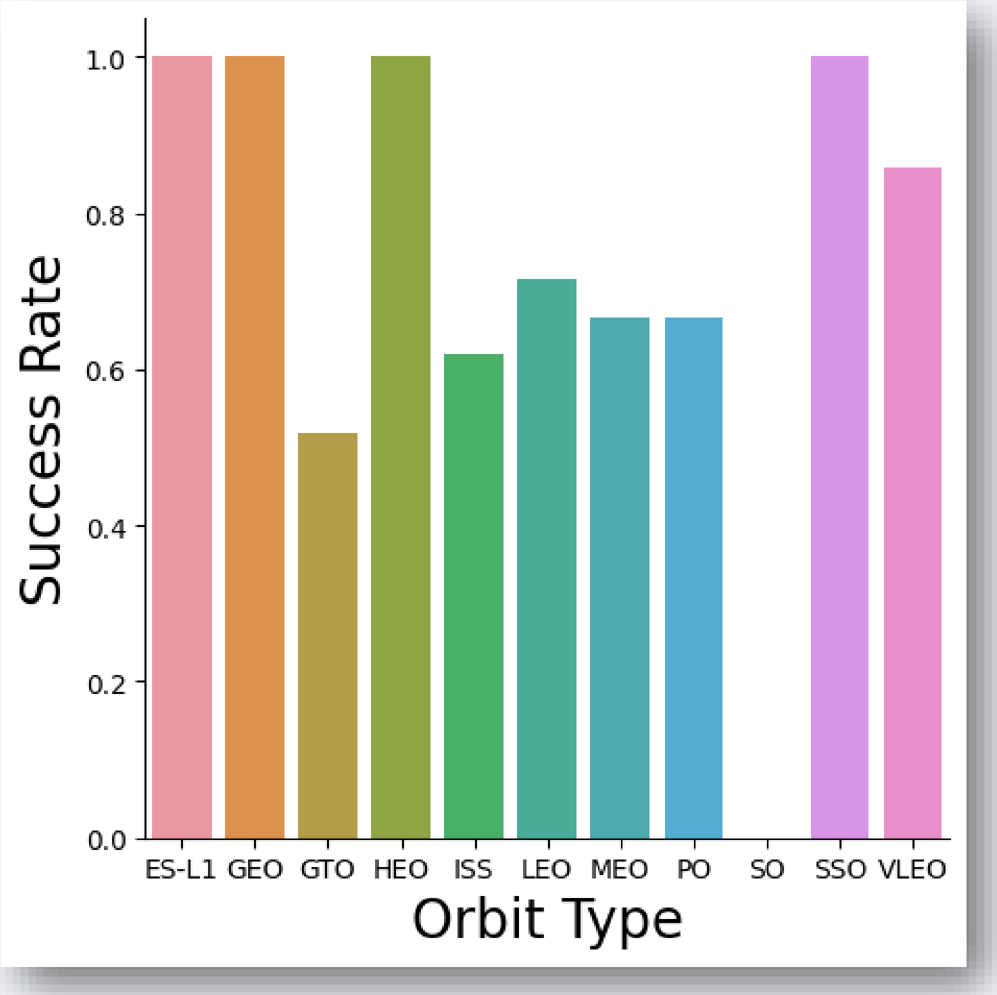
- **100% Success Rate:**

ES-L1, GEO, HEO and SSO

- **50%-80% Success Rate:** GTO, *ISS, LEO, MEO, PO*

- **0% Success Rate:**

SO



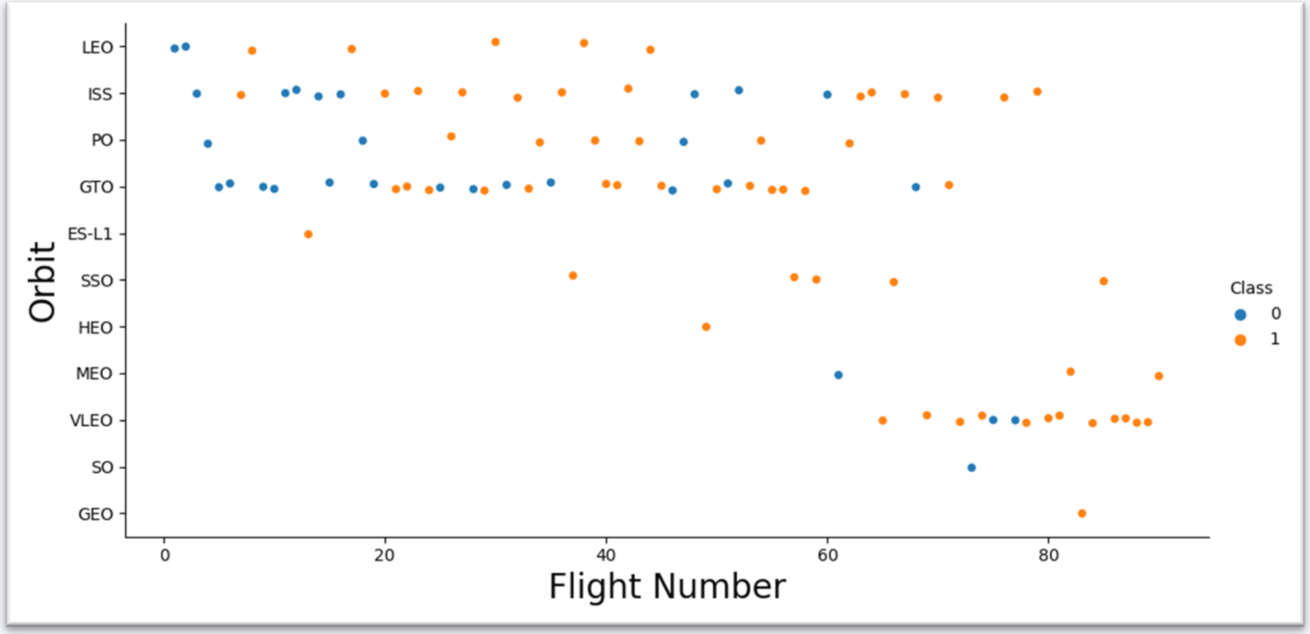
Flight Number vs. Orbit Type

- The success rate typically

increases with the number of flights for each orbit

- This relationship is highly apparent for the LEO orbit

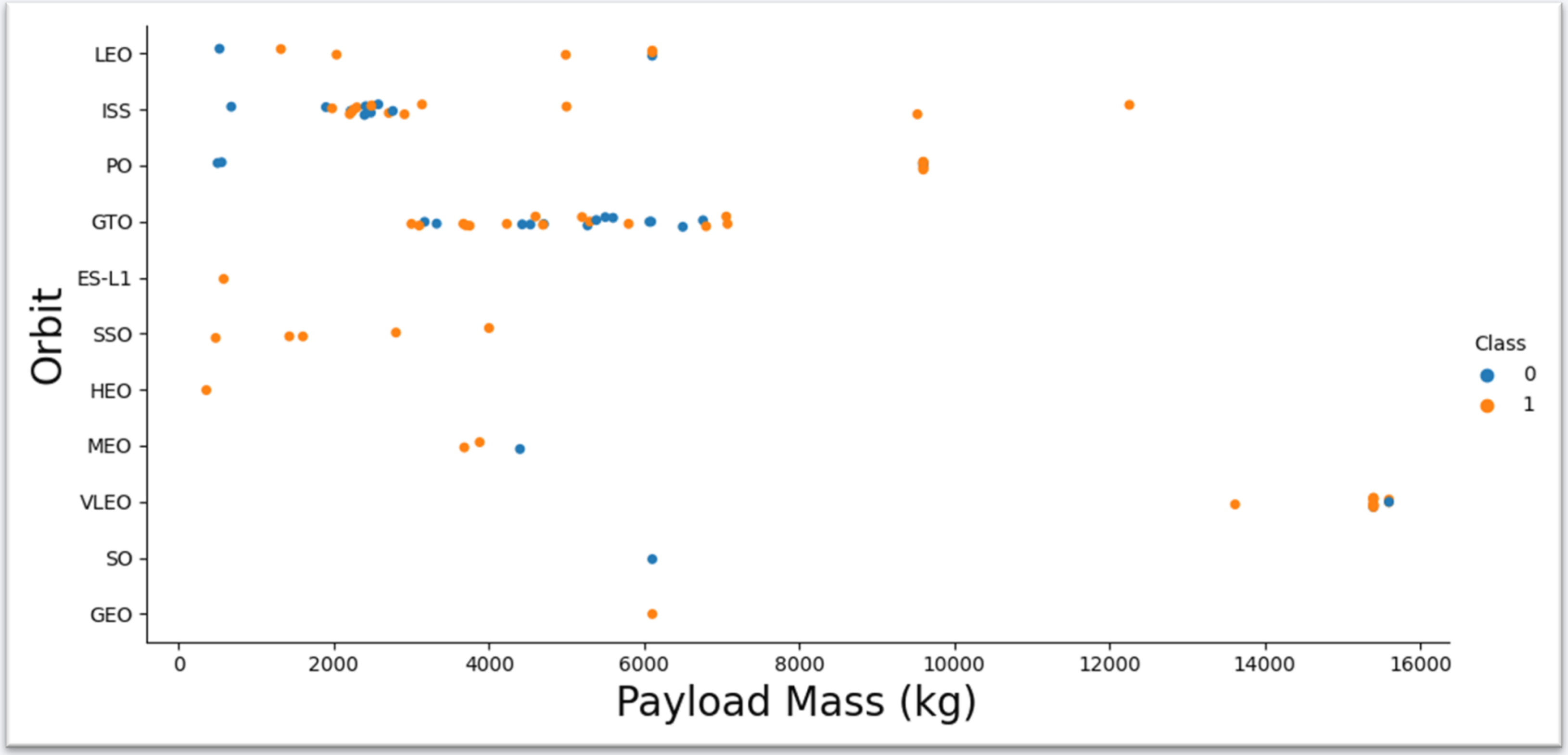
- The GTO orbit, however, does not follow this trend



Payload vs. Orbit Type

- Heavy payloads are better with LEO, ISS and PO orbits

- The GTO orbit has mixed success with heavier payloads

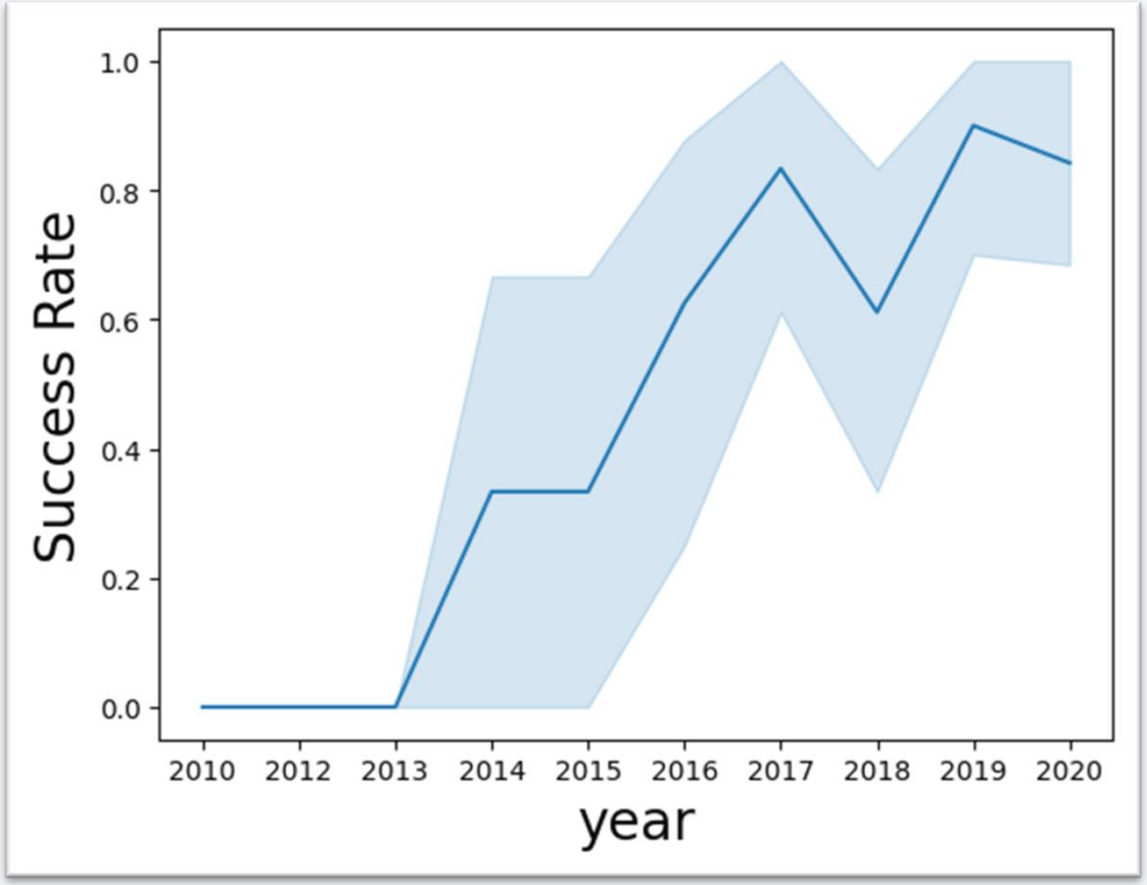


Launch Success Yearly Trend

- The success rate improved from 2013-2017 and 2018-2019

- The success rate decreased from 2017-2018 and from 2019-2020

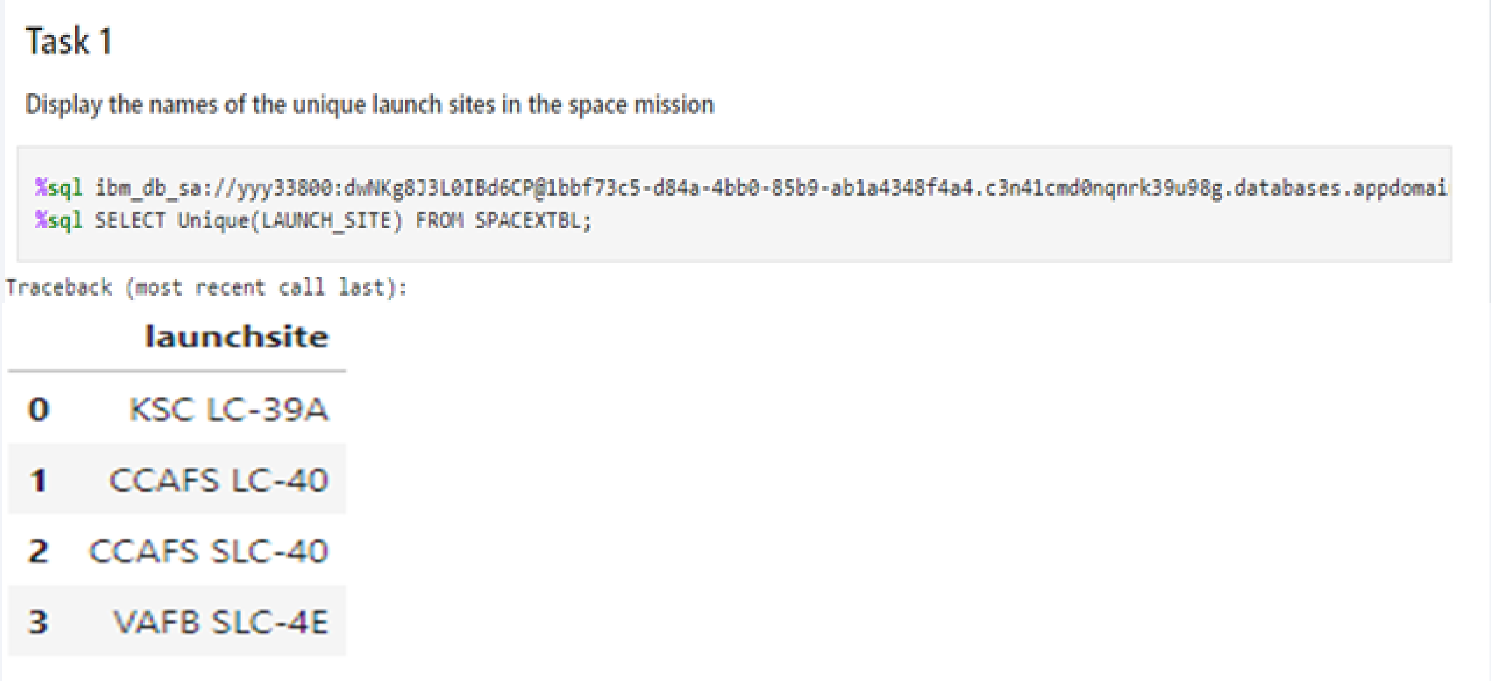
- Overall, the success rate has improved since 2013



All Launch Site Names

Launch Site Names

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



Task 1

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

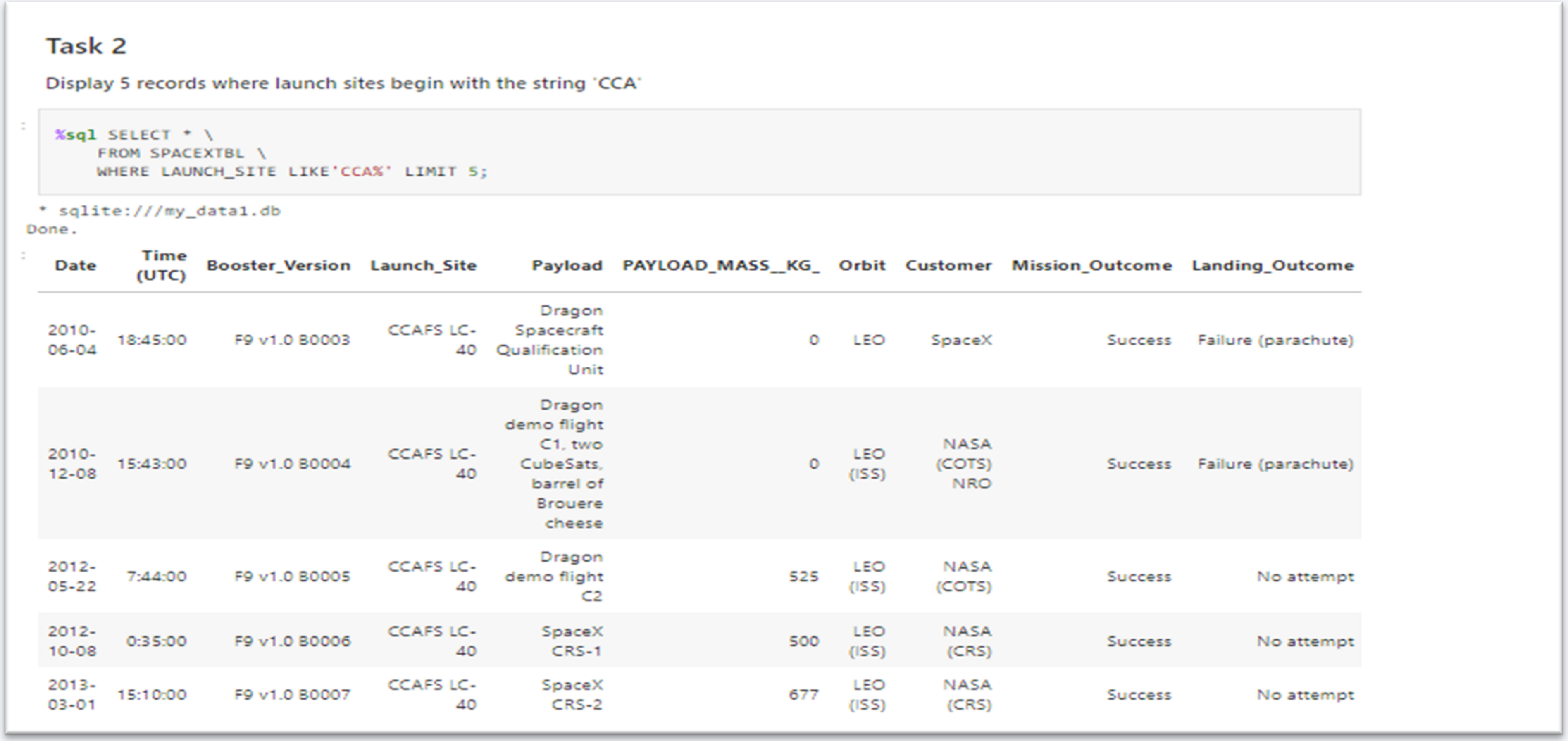
```
%sql ibm_db_sa://yyy33800:dwNKg8J3L0IBd6CP@1bbf73c5-d84a-4bb0-85b9-ab1a4348f4a4.c3n41cmd0nqnrk39u98g.databases.appdomai
%sql SELECT Unique(LAUNCH_SITE) FROM SPACEXTBL;
```

```
Traceback (most recent call last):
```

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

Launch Site Names Begin with 'CCA'

Applied the query above to display 5 records where launch sites begin with `CCA`



Task 2

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

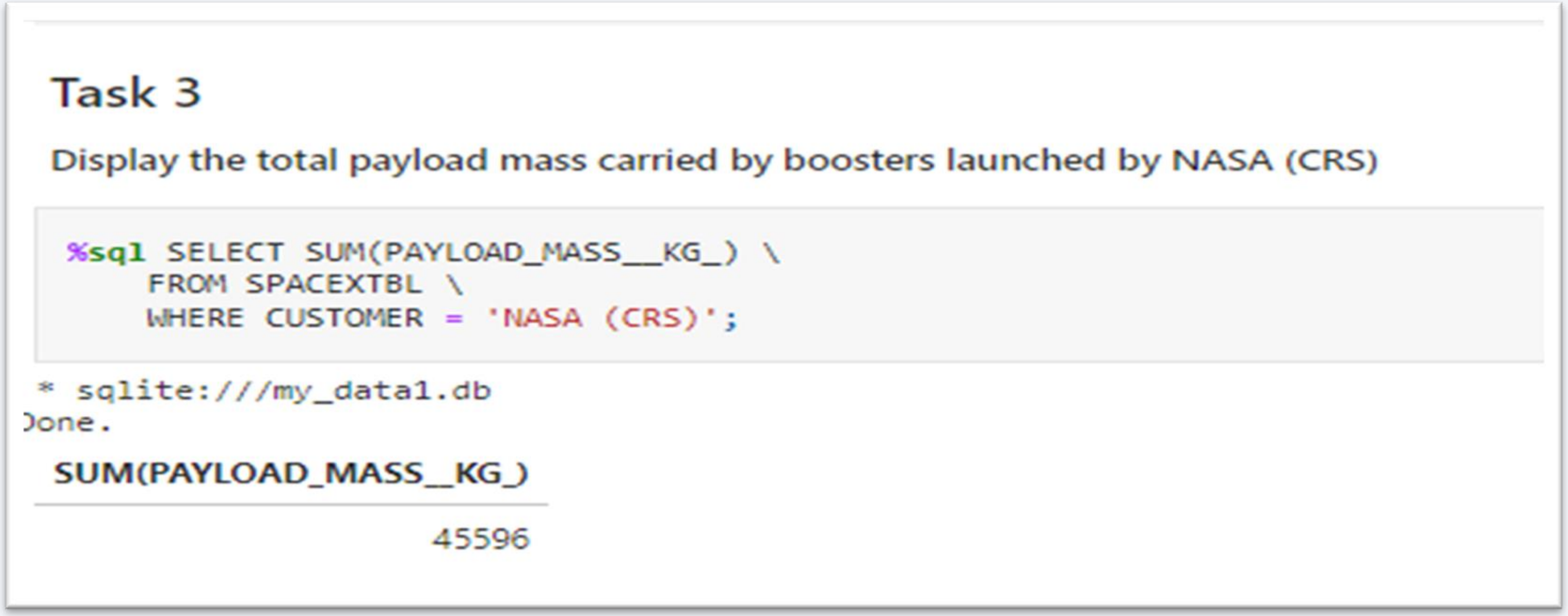
```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

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

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

Calculated the total payload carried by boosters from NASA as 45596 using the query below



Task 3

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

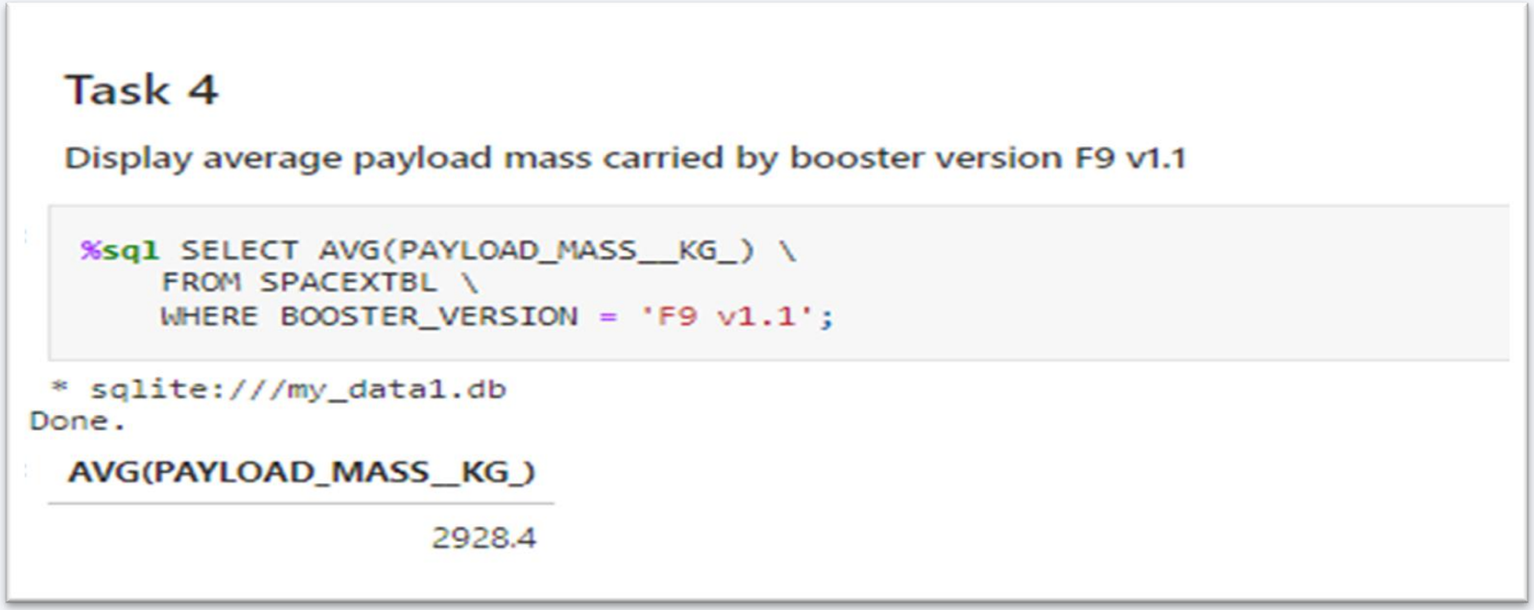
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE CUSTOMER = 'NASA (CRS)';
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Calculated the average payload mass carried by booster version F9 v1.1



Task 4

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

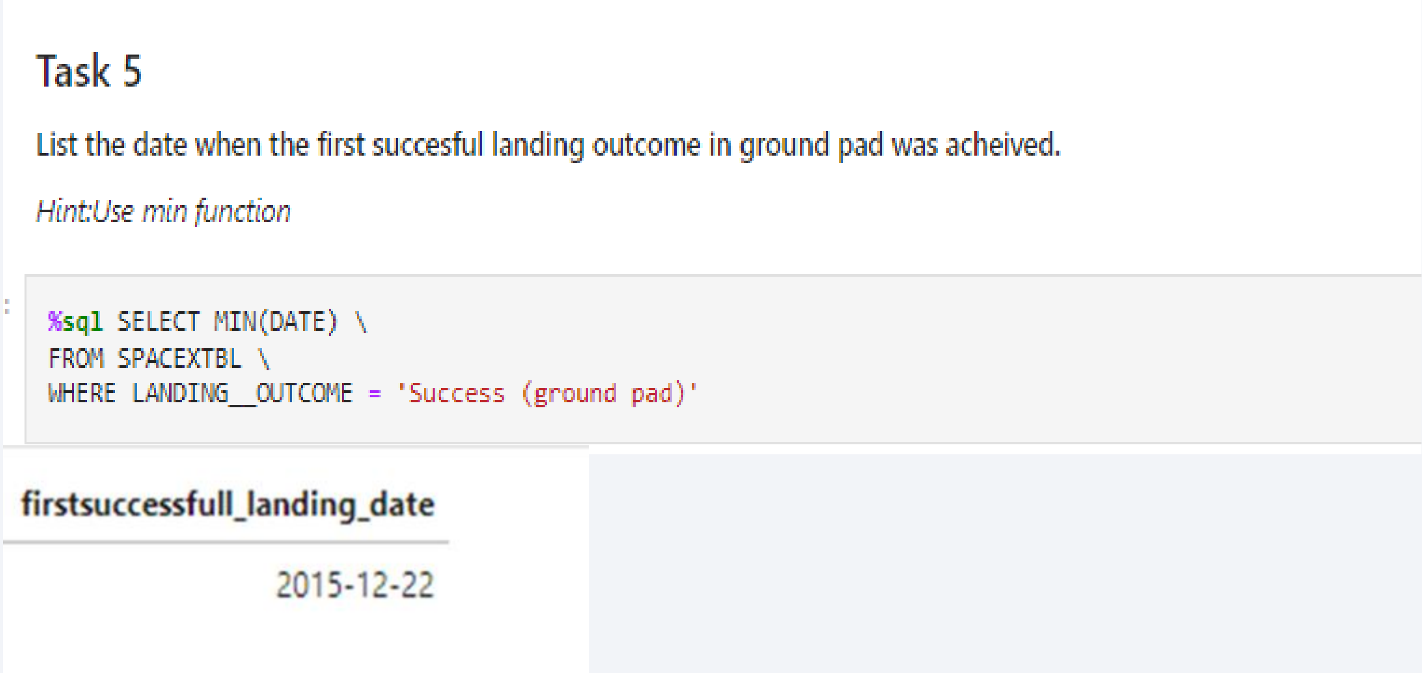
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015



Task 5

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

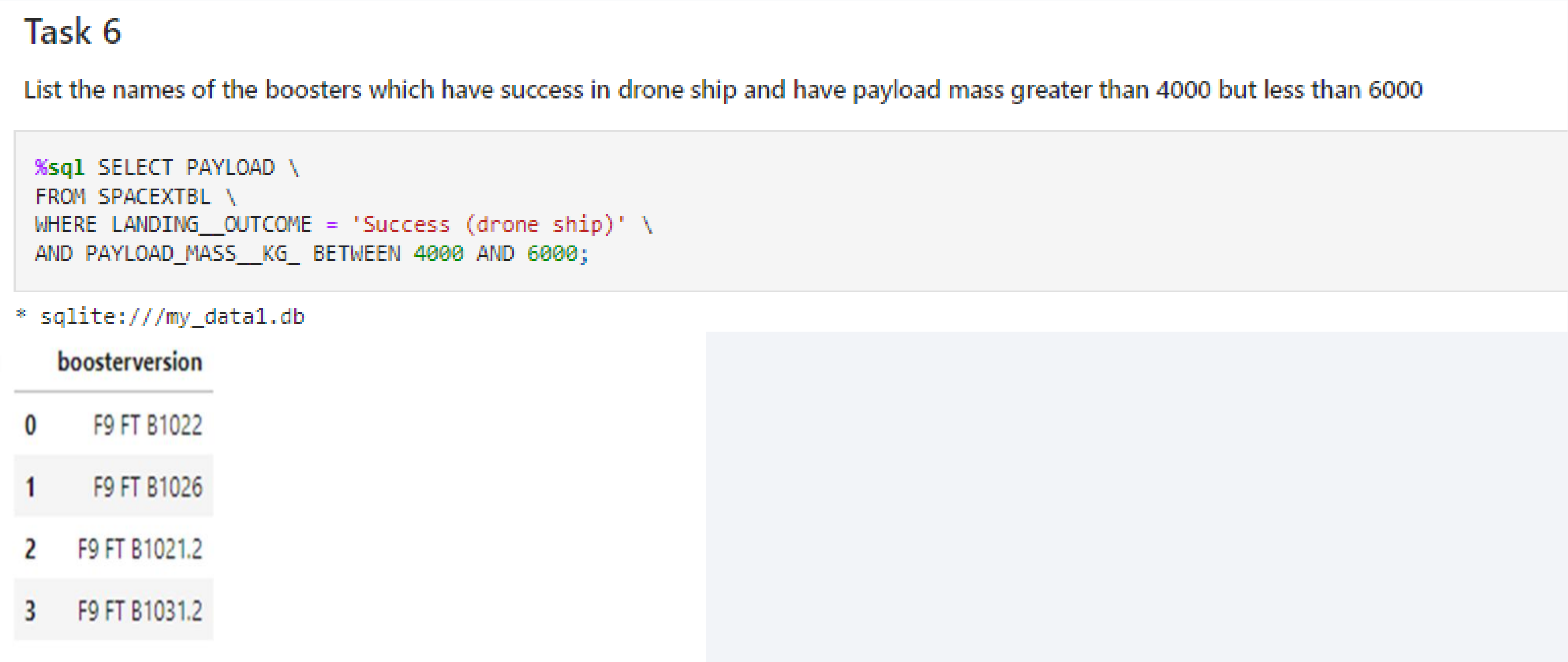
Hint:Use min function

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

firstsuccessfull_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Applied the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 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

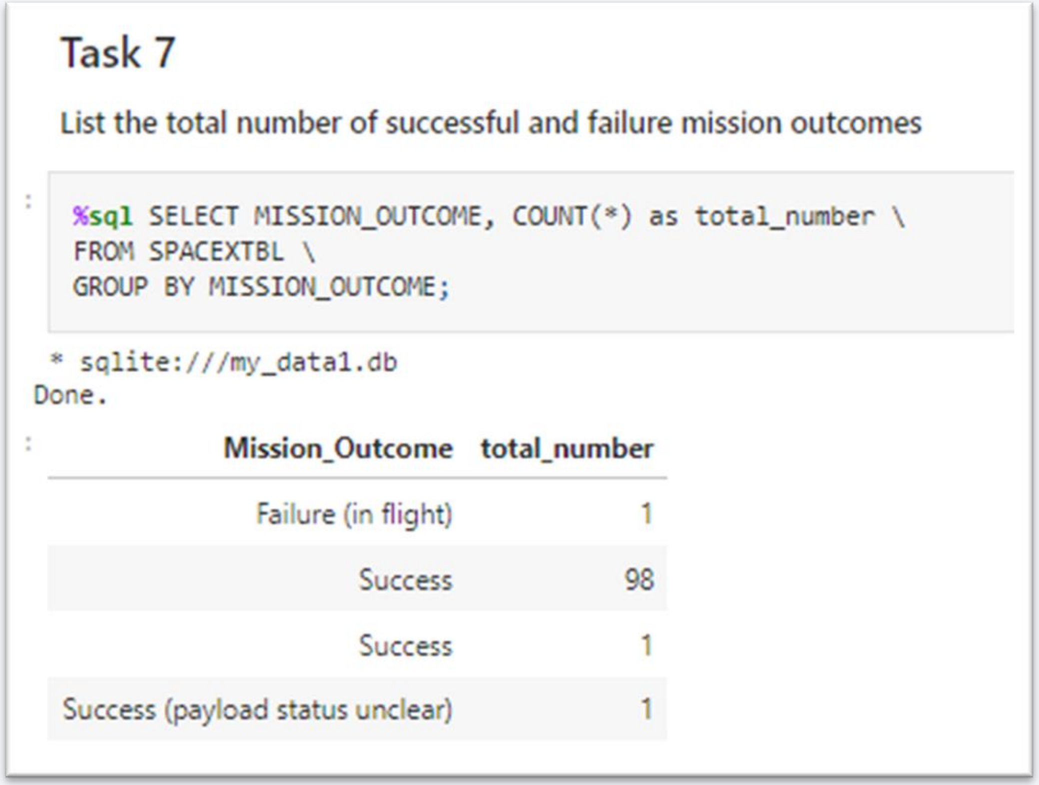
```
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

	boosterversion
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

- 1 Failure in Flight
- 99 Success
- 1 Success (payload status unclear)



Task 7

List the total number of successful and failure mission outcomes

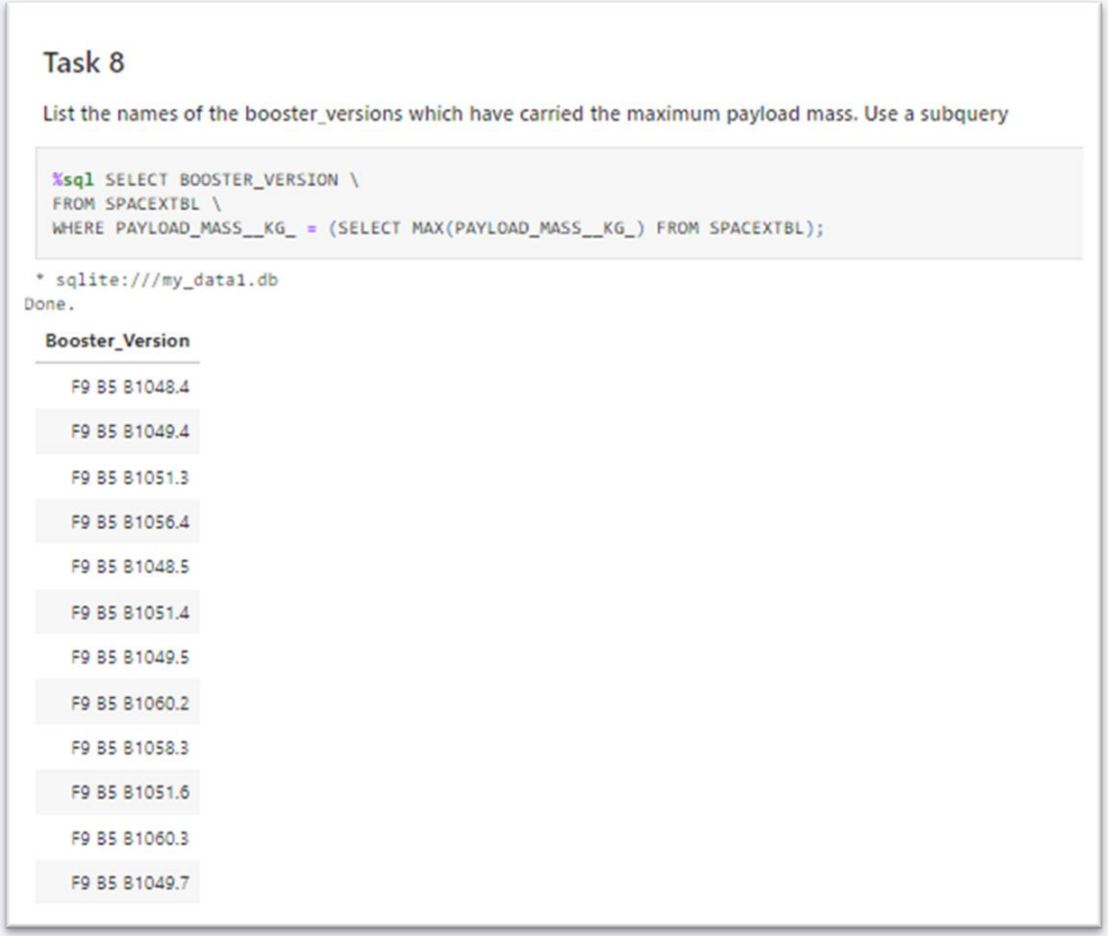
```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

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

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

Boosters Carried Maximum Payload

Determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.



Task 8

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

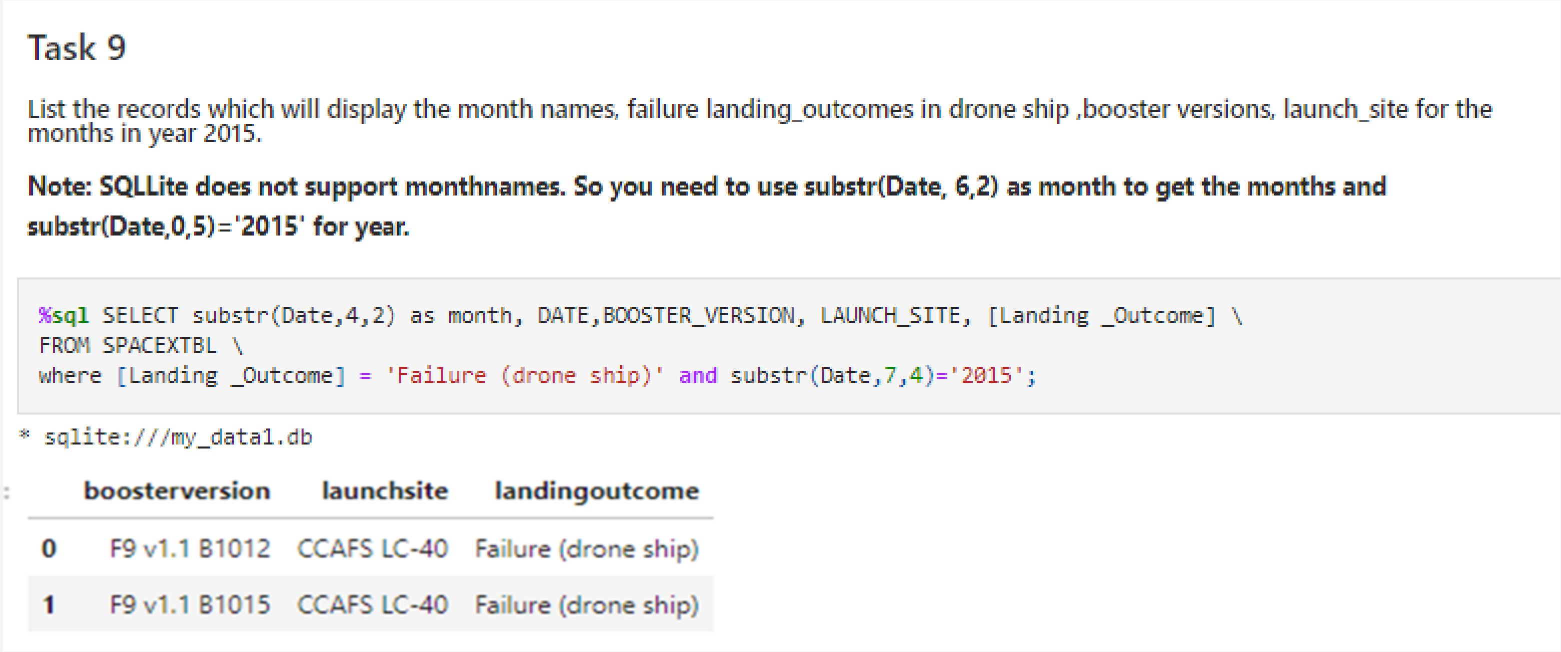
```
%sql SELECT BOOSTER_VERSION \
FROM SPACEXTBL \
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

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

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

Used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for 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.

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

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

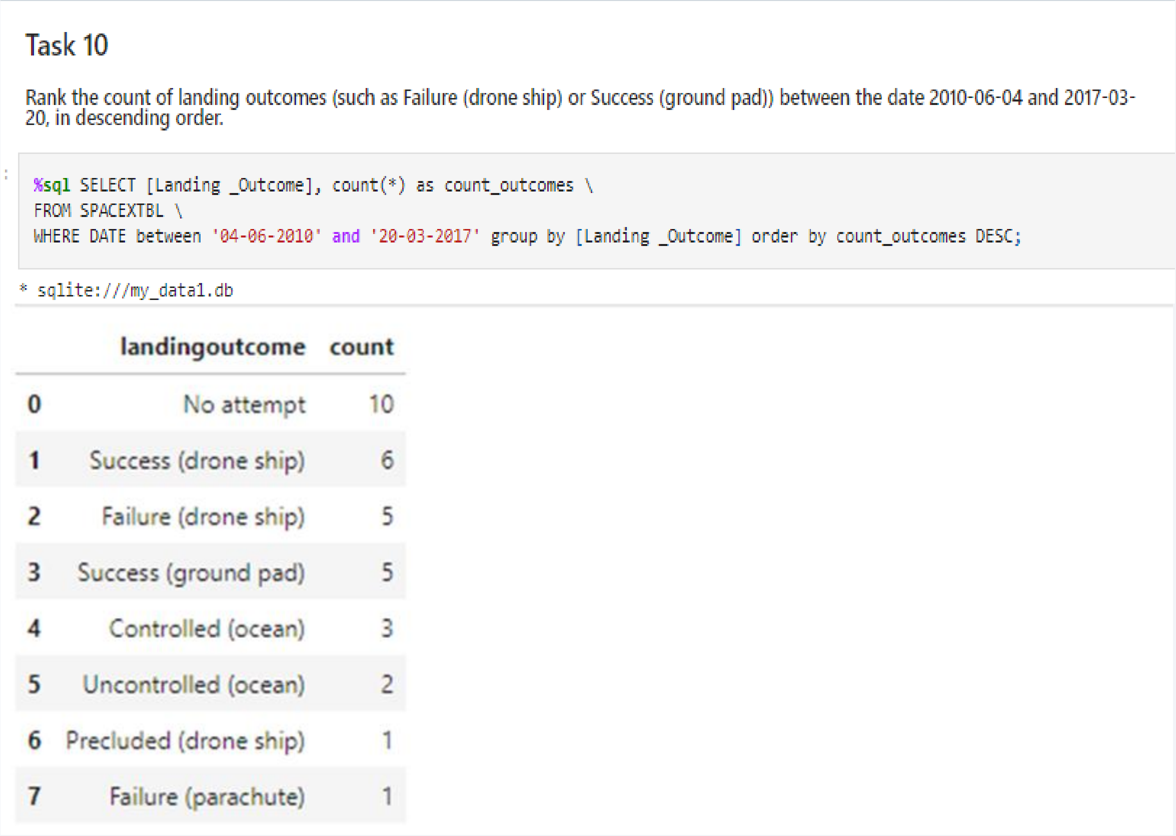
	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Utilized the **WHERE** clause to filter landing outcomes for the period **BETWEEN** June 4, 2010, and March 20, 2010.

Applied the **GROUP BY** clause to group the landing outcomes.

Used the **ORDER BY** clause to arrange the grouped landing outcomes 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.

```
%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 count_outcomes DESC;
```

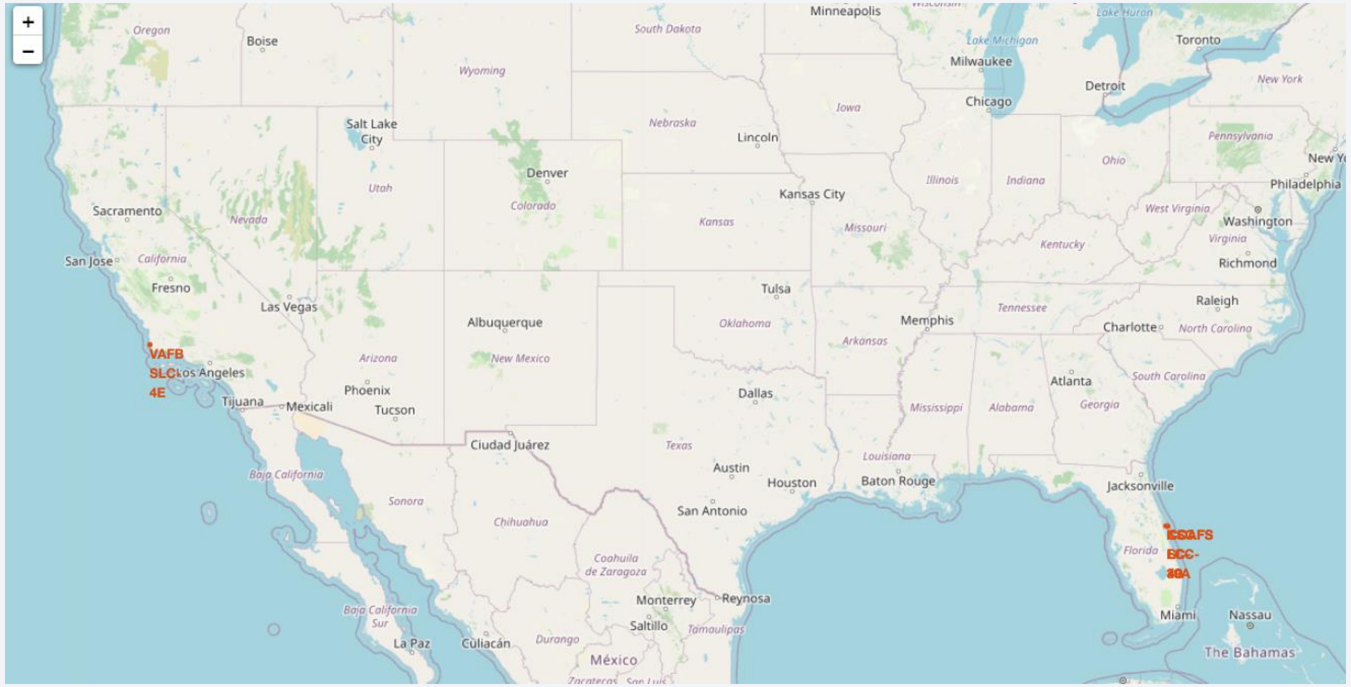
```
* sqlite:///my_data1.db
```

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

Section 3

Launch Sites Proximities Analysis

Launch Sites

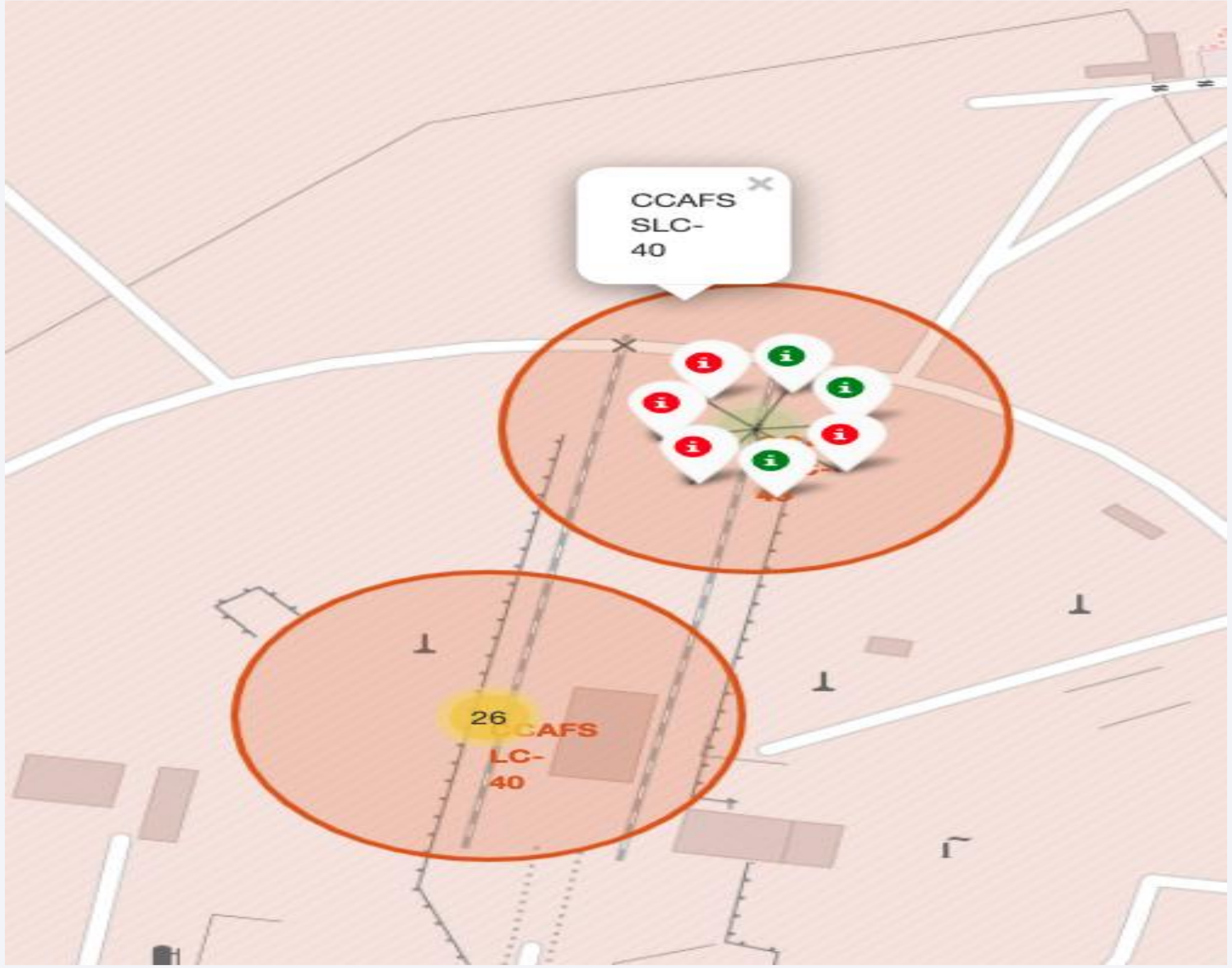


- Near Equator: the closer the launch site to the equator, the easier it is to launch to equatorial orbit, and the more help you get from Earth's rotation for a prograde orbit. Rockets launched from sites near the equator get an

- Due to the rotational speed of earth - that helps to save the cost of putting in extra fuel and boosters.

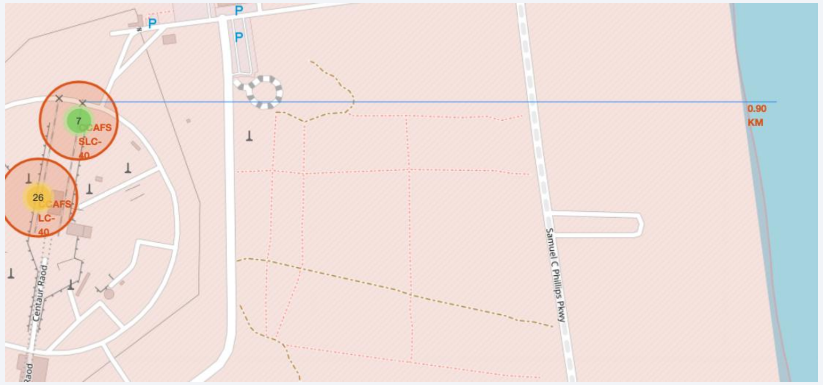
Markers showing launch sites with color labels

Green markers for successful launches
Red markers for unsuccessful launches



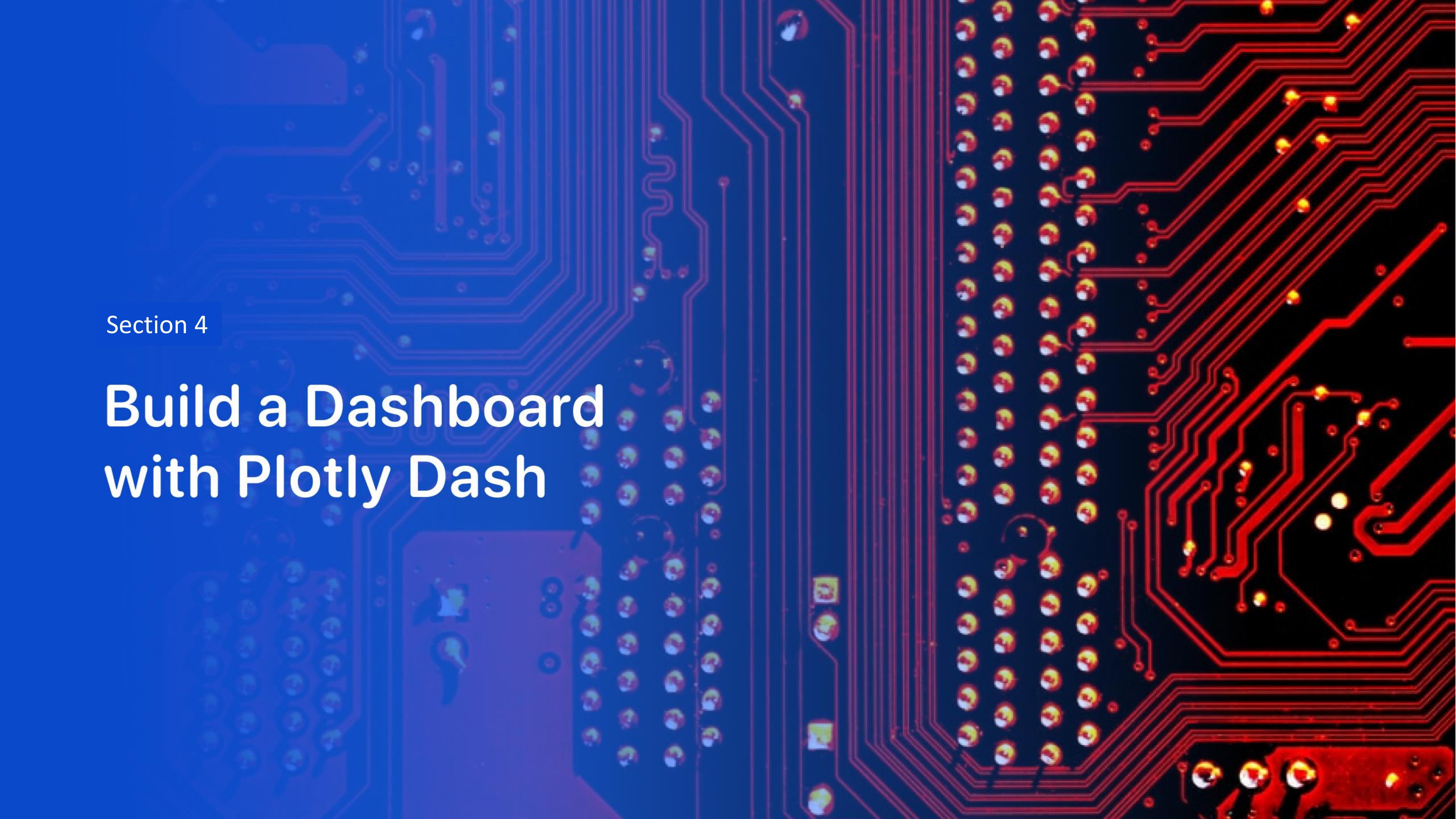
Launch Site distance to landmarks

- Critical for ensuring that spent rocket stages or failed launches do not fall along the launch path, preventing potential harm to people or property.
- Establishes exclusion zones around launch sites to maintain security, keep unauthorized individuals away, and ensure the safety of people in the vicinity.
- Requires launch sites to be positioned away from areas that could be damaged by a failed launch. Simultaneously, they should remain close enough to roads, railways, and docks for efficient transportation of people and materials to and from the launch site in support of launch activities.

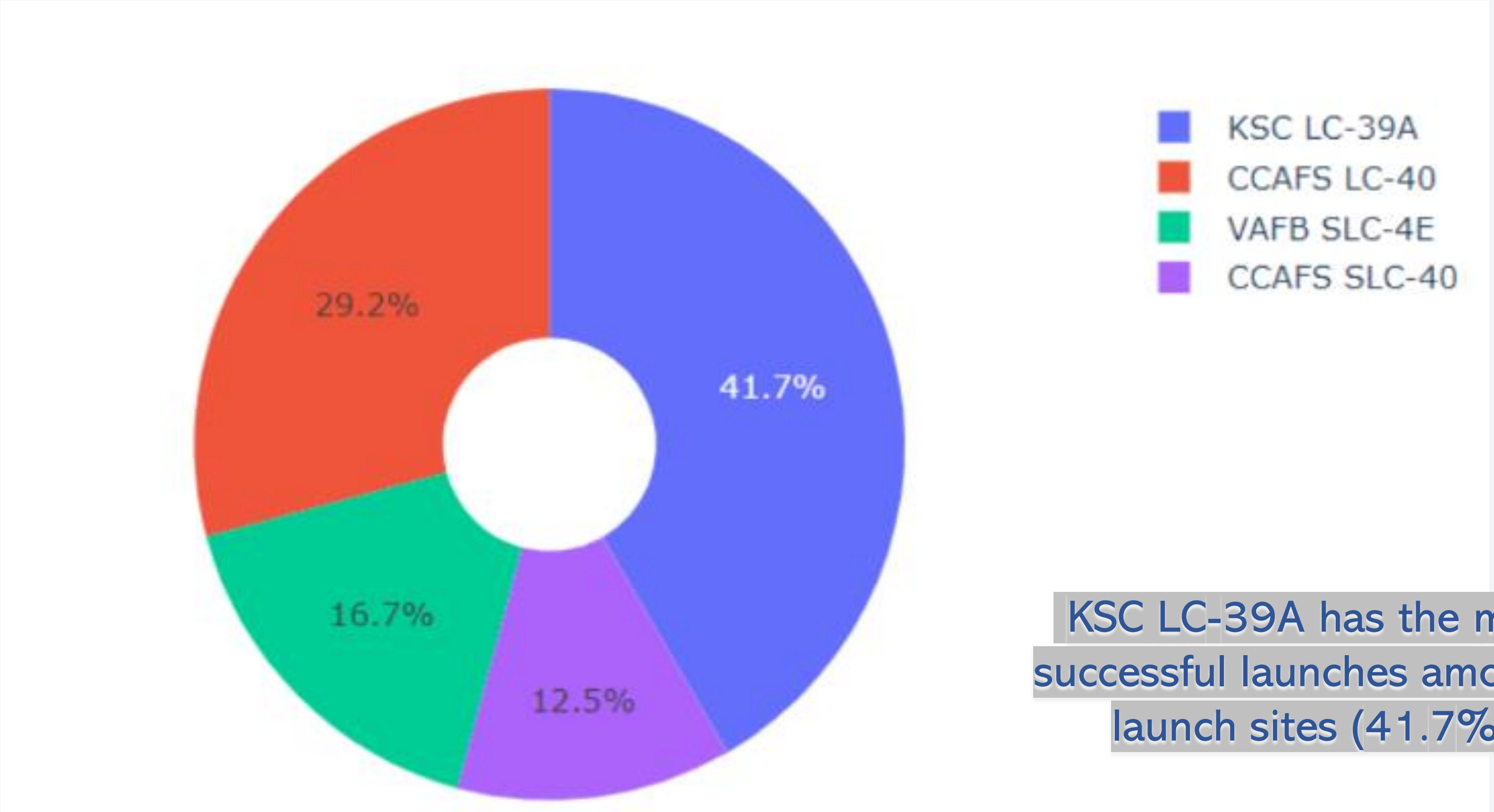


Section 4

Build a Dashboard with Plotly Dash



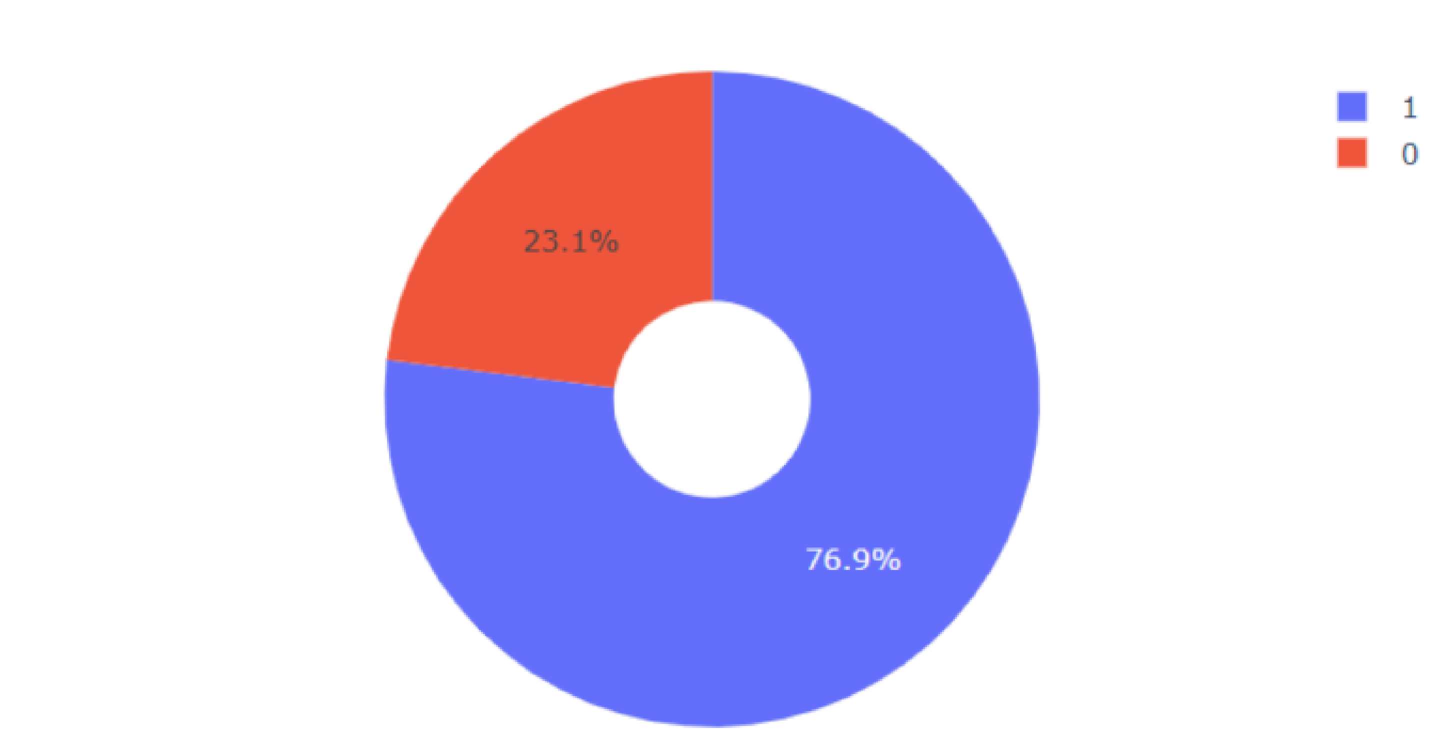
Launch Success by Site



KSC LC-39A has the most successful launches amongst launch sites (41.7%)

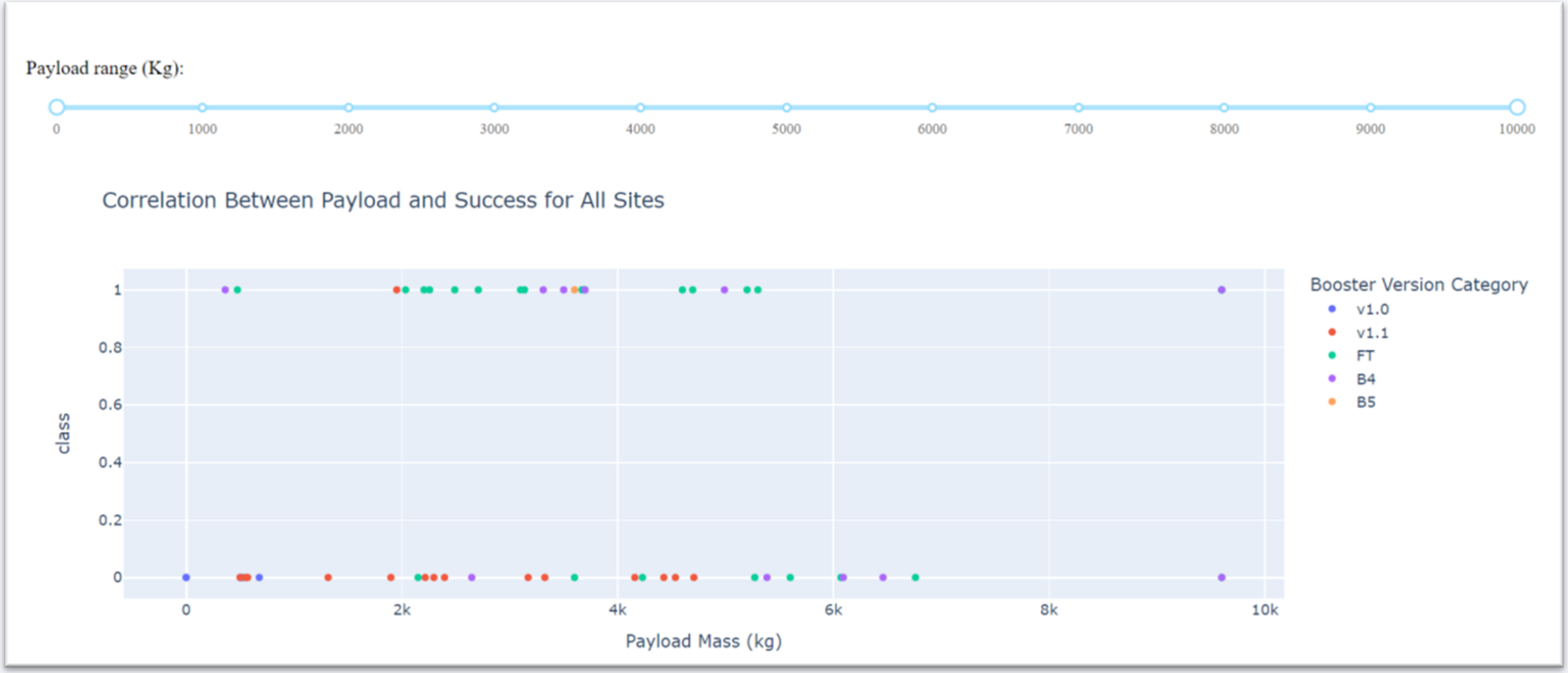
Launch Success (KSC LC-29A)

- KSC LC-39A has the highest success rate amongst launch sites (76.9%)
- 10 successful launches and 3 failed launches



Payload Mass and Success

- Payloads between 2,000 kg and 5,000 kg have the highest success rate
- 1 indicating successful outcome and 0 indicating an unsuccessful outcome

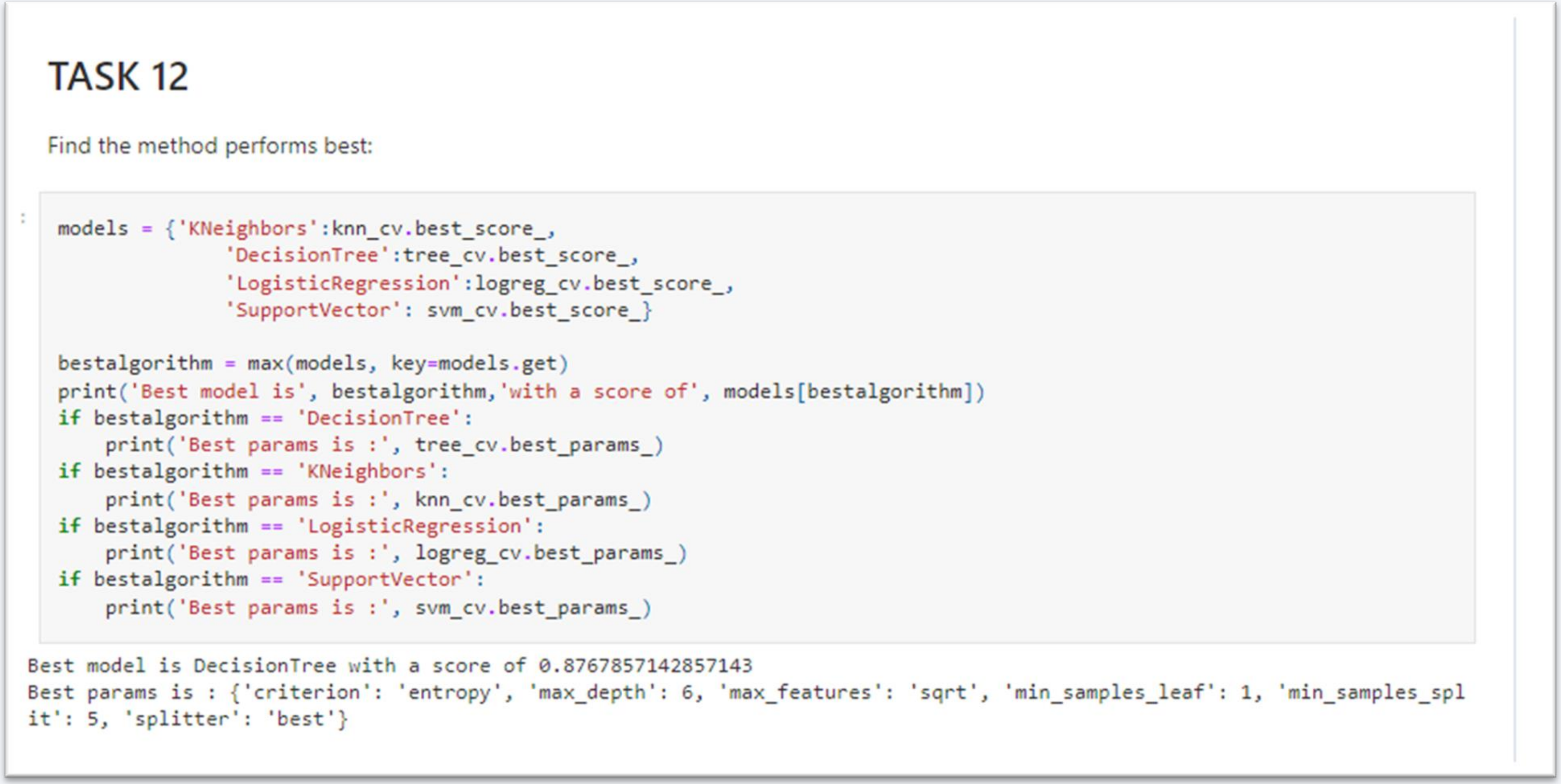


Section 5

Predictive Analysis (Classification)

Classification Accuracy

All the models performed at about the same level and had the same scores and accuracy. This is likely due to the small dataset. The Decision Tree model slightly outperformed the rest when looking at .best_score_



TASK 12

Find the method performs best:

```
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_)
```

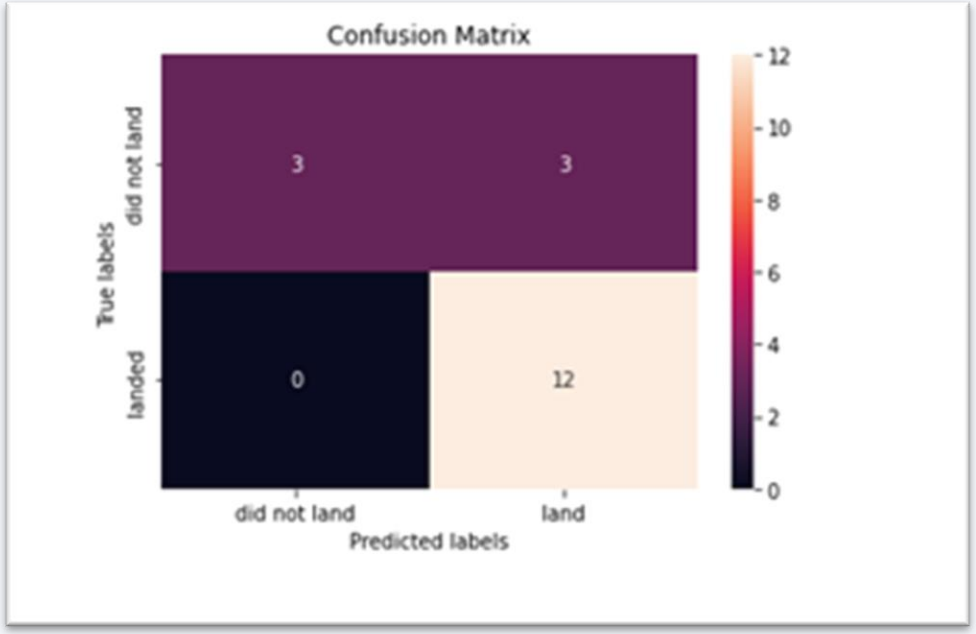
```
Best model is DecisionTree with a score of 0.8767857142857143
Best params is : {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_spl
it': 5, 'splitter': 'best'}
```

Confusion Matrix

A confusion matrix provides a comprehensive overview of a classification algorithm's performance.

The observed confusion matrices were consistent across different evaluations.

The presence of false positives (Type 1 errors) is a concern, indicating instances where the algorithm incorrectly identifies negative cases as positive.



Confusion matrix outputs:

True Positive (TP): 12

True Negative (TN): 3

False Positive (FP): 3

False Negative (FN): 0

Precision (Precision = TP / (TP + FP)): 0.80

Recall (Recall = TP / (TP + FN)): 1.00

F1 Score (F1 Score = 2 * (Precision * Recall) / (Precision + Recall)): 0.89

Accuracy (Accuracy = (TP + TN) / (TP + TN + FP + FN)): 0.833

Conclusions

Launch Success Trends:

- Higher flight amounts at a launch site correlate with greater success rates.
- Launch success has shown a consistent upward trend since 2013, peaking in 2020.

Orbital Success Rates:

- Orbits ES-L1, GEO, HEO, SSO, and VLEO demonstrated exceptionally high success rates.

Top Performing Launch Site:

- KSC LC-39A emerged as the top-performing site with the highest number of successful launches.

Machine Learning Algorithm:

- The Decision Tree classifier proved to be the most effective machine learning algorithm.

Conclusions

Research Insights:

- Equator Advantage: Launch sites near the equator benefit from Earth's rotational speed, reducing costs.
- Coastal Proximity: All launch sites are strategically located close to coastlines.
- Launch Success Improvement Over Time: A consistent improvement in launch success rates over the years.
- KSC LC-39A Success Rate: Notably high success rates, reaching 100% for specific payload ranges.
- Orbits Success: Certain orbits consistently achieved a 100% success rate.
- Payload Mass Impact: Higher payload masses are associated with higher success rates.

Thank you!