IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nelson Amaro
13/07/2024

Outline

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

Executive Summary

Summary of methodologies

- Collecting data using SpaceX REST API and Web Scraping;
- Wrangling data to find patterns and to label supervised trained models;
- Explore data through some key factors, like payload, orbit, launch site, etc;
- Analyze data using SQL;
- Explore launch site success rates;
- Visualize launch sites with the most success;
- Build models in order to predict landing outcomes.

Introduction

- Space tourism is relatively new, so there is a lot of market for big tech companies to explore in this field, like SpaceX or Virgin Galactic, for instance. Today, this sort of tourism is only given access to a very few group of people.

- The next steps envelop making this sector more affordable and safer for people who dream of seeing the blue Earth from ‘the outside’ and make space travel cheaper for the companies themselves.

- Data modelling can have a huge impact in achieving these goals as it attempts to prevent and foretell the success and costs of such endeavors.

- One of the major issues presented to space tourism providers is the ability to predict whether the the first stage of a rocket lauch can be landed successfully and reused for future launches decreasing the overall cost of a rocket launch into outer space.

Section 1

Methodology

Methodology

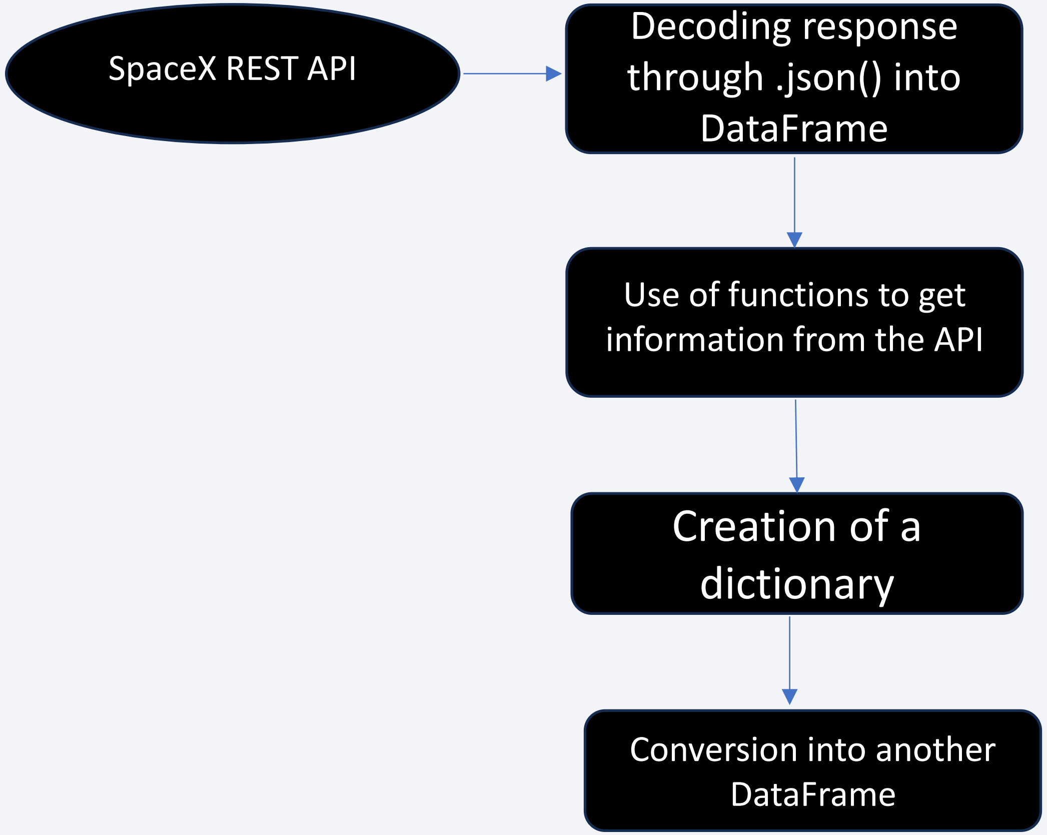
- **Collecting** data resorting to the SpaceX REST API and web scraping technicques;
- **Wrangling** data by filtering, dealing with missing values and using one hot encoding, thus preparing data to be worked and modelled;
- **Exploring** data with SQL and visualization tools like Matplotlib;
- **Visualizing** data with Folium and Dash;
- **Building** predictive and classification models.

Data Collection – SpaceX REST API

- First, we imported the necessary libraries – request, pandas, numpy and datetime;
- From getresquest, we requested data from the SpaceX REST API;
- Next, we decoded the response using .json();
- All the information in the json file was normalized and turned into a dataframe;
- We extracted information from the API again using functions, like getBoosterVerson, getLaunchSite, getPayLoadData and getCoreData;
- With that information, we created then a dictionary, called ‘launch_dict’;
- ‘launch_dict’ was transformed into a DataFrame called ‘df’;
- Filtered some information and replaced the missing values in ‘PayLoadMass’ column with its mean value

Data Collection – SpaceX API

- Flowchart – Data Collection SpaceX REST API



SpaceX REST API
Decoding response through .json() into DataFrame
Use of functions to get information from the API
Creation of a dictionary
Conversion into another DataFrame

- GitHub URL

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

Data Collection - Scraping

- Steps:

1. From Wikipedia, Falcon 9 launch data was requested;

2. BeautifulSoup object was created from An HTML response

3. Column names were extracted from HTML table header

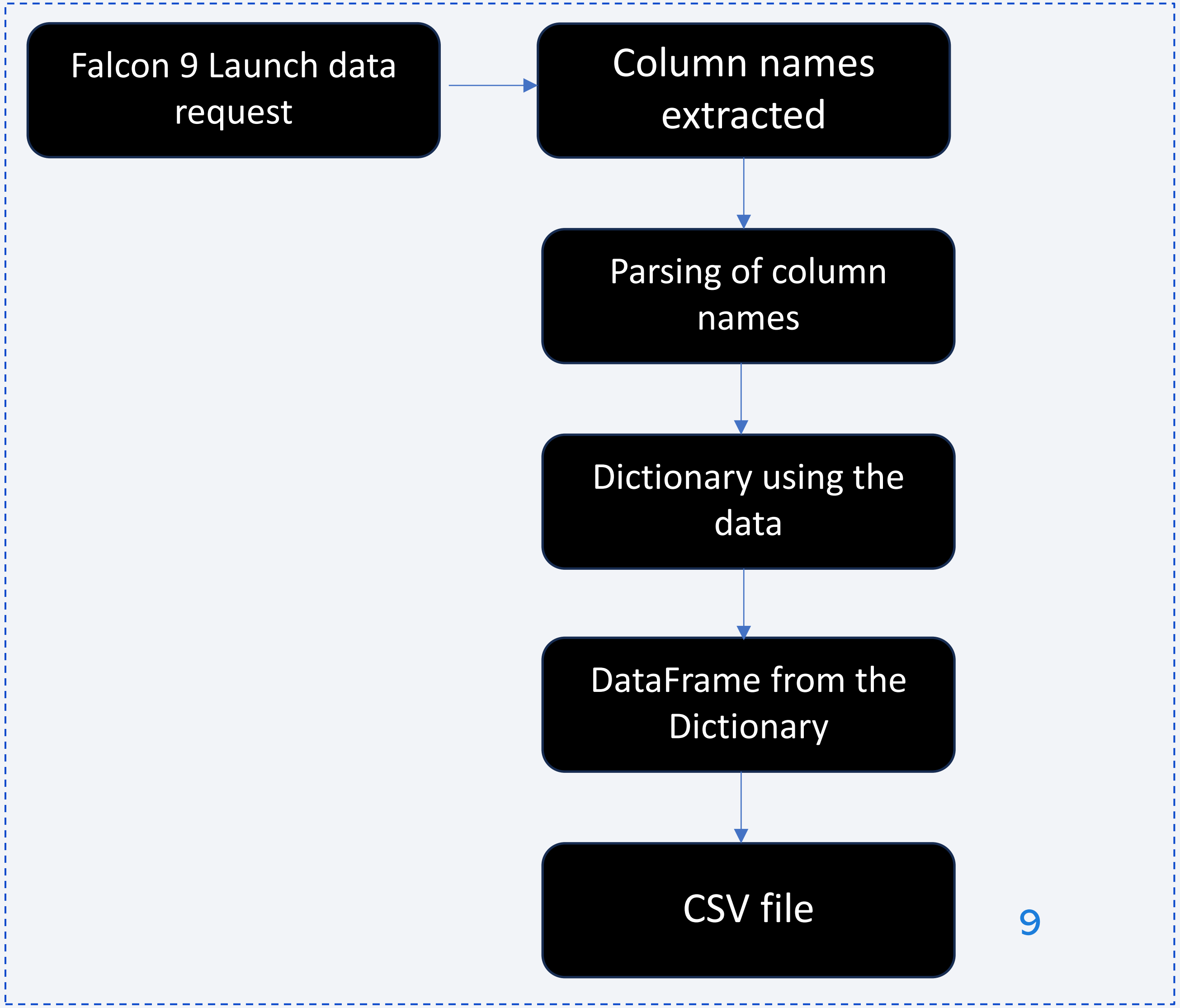
4. HTML tables were parsed to collect data

5. Created dictionary from the data

6. From the dictionary, created a DataFrame

7. Exported data to a CSV file

https://github.com/junioramaro/final_project/blob/main/data_collection_web_scraping_lab.ipynb


Falcon 9 Launch data request
Column names extracted
Parsing of column names
Dictionary using the data
DataFrame from the Dictionary
CSV file

Data Wrangling

- **Steps:**
- **Calculate:**
  - Number of launches at each site;
  - Number and occurrence at each orbit;
  - Number and occurrence of mission outcome per orbit type

- **Create binary landing outcome:**
  - **True ocean** – mission outcome landed successfully in a specific part of the ocean;
  - **False ocean** – mission outcome landed unsuccessfully in a specific part of the ocean;
  - **False RTLS** – mission outcome landed successfully on a ground pad;
  - **True RTLS** – mission outcome landed unsuccessfully on a ground pad;
  - **True ASDS** – mission outcome landed successfully on a drone ship;
  - **False ASDS** – mission outcome landed unsuccessfully on a drone ship;
  - Mission outcomes converted to 1 if successful

- *https://github.com/junioramaro/final_project/blob/main/data_wrangling.ipynb*

EDA with Data Visualization

Charts

- Flight number vs Payload mass (Kg)
- Flight number vs Launch site
- Orbit type vs Launch site
- Orbit type vs Payload mass (Kg)
- Orbit type vs Success rate

1. **Scatter plot** to see if there were a relationship between variables.
2. We also used **bar charts** to see the relationship between categorical variables.
3. **Line plots** were also used for discreet variables.

https://github.com/junioramaro/final_project/blob/main/eda_data_vizualization.ipynb

EDA with SQL

SQL Queries

- Names of unique launch sites;
- 5 examples of launch sites starting with the letters CCA;
- Total Payload mass (Kg) carried by boosters from NASA (CRA);
- Average Payload mass (Kg) carried by booster version F9 v1.1;
- Total number of successful and failed mission outcomes;
- Date when the first successful outcome in ground pad was achieved;
- Booster versions which have success in drone ship and have Payload mass (Kg) between 4000 and 6000;
- Booster versions which have have carried the maximum Payload mass (Kg);
- Records with month names, failure landing outcomes in drone ship, booster versions and launch sites in 2015;
- Count the landing outcomes between the dates 2010-06-04 and 2017-03-20

https://github.com/junioramaro/final_project/blob/main/eda-SQL.ipynb

Build an Interactive Map with Folium

- Used a blue circle at NASA Johnson Space Center with a pop-up label showing its name
- Used colored markers:

- Green for successful launches at each launching site;

- Red for unsuccessful launches at each launching site.

- Used colored lines to show the distance between launching site CCAFS SLC-40 and the closest city, railway, hightway and coastline

https://github.com/junioramaro/final_project/blob/main/launch_site_Folium.ipynb

Build a Dashboard with Plotly Dash

- **Dropdown list** of launch, enabling launch site selection
- **Pie chart**, showing percentage of successful and unsuccessful launch sites
- **Slider**, enabling selection of a Payload mass (Kg) range
- **Scatter chart**, showing the correlation between Payload mass (Kg) and launching success

- *https://github.com/junioramaro/final_project/blob/main/spacex_dash_app.py*

Predictive Analysis (Classification)

- Created a Numby array for column class in data;
- Standardized the data using StandardScaler;
- Split the data into train and test data;
- Found the best parameters using GridSearchCV object;
- Applied GridSearchCV on a few algorithms:

 - K-Nearest Neighbour (KNN);
 - Support Vector Machine (SVC);
 - Logistic Regression;
 - Decision Tree.

- Calculated accuracy using the score method;
- Analyzed the confusion matrix for all algorithms;
- Found the best model.

- *https://github.com/junioramaro/final_project/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb*

Results

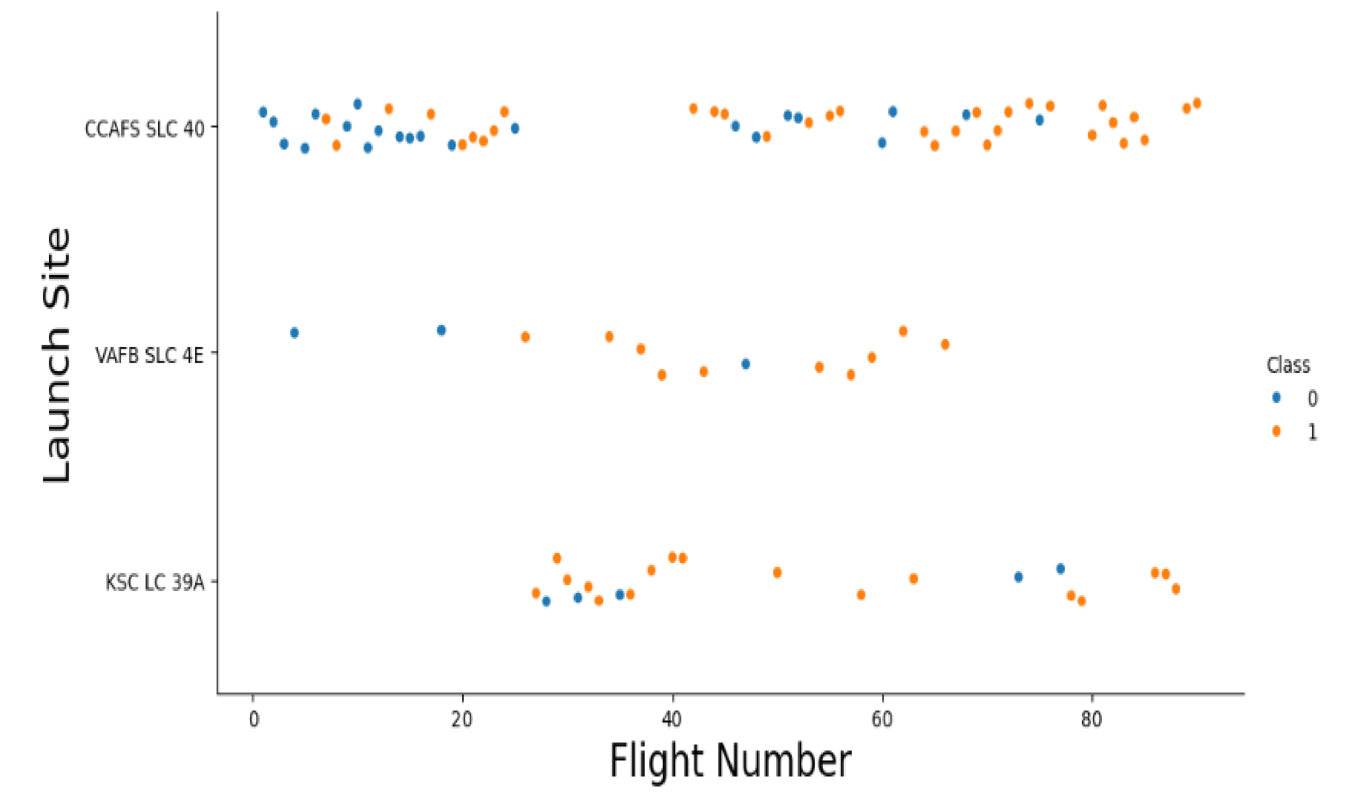
- **Exploratory data analysis results**
- Launch success has improved over time;
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate.
- **Interactive analytics demo in screenshots**
- In Section 3.
- **Predictive analysis results**
- Best model was found to be the Decision Tree model.

Section 2

Insights drawn from EDA

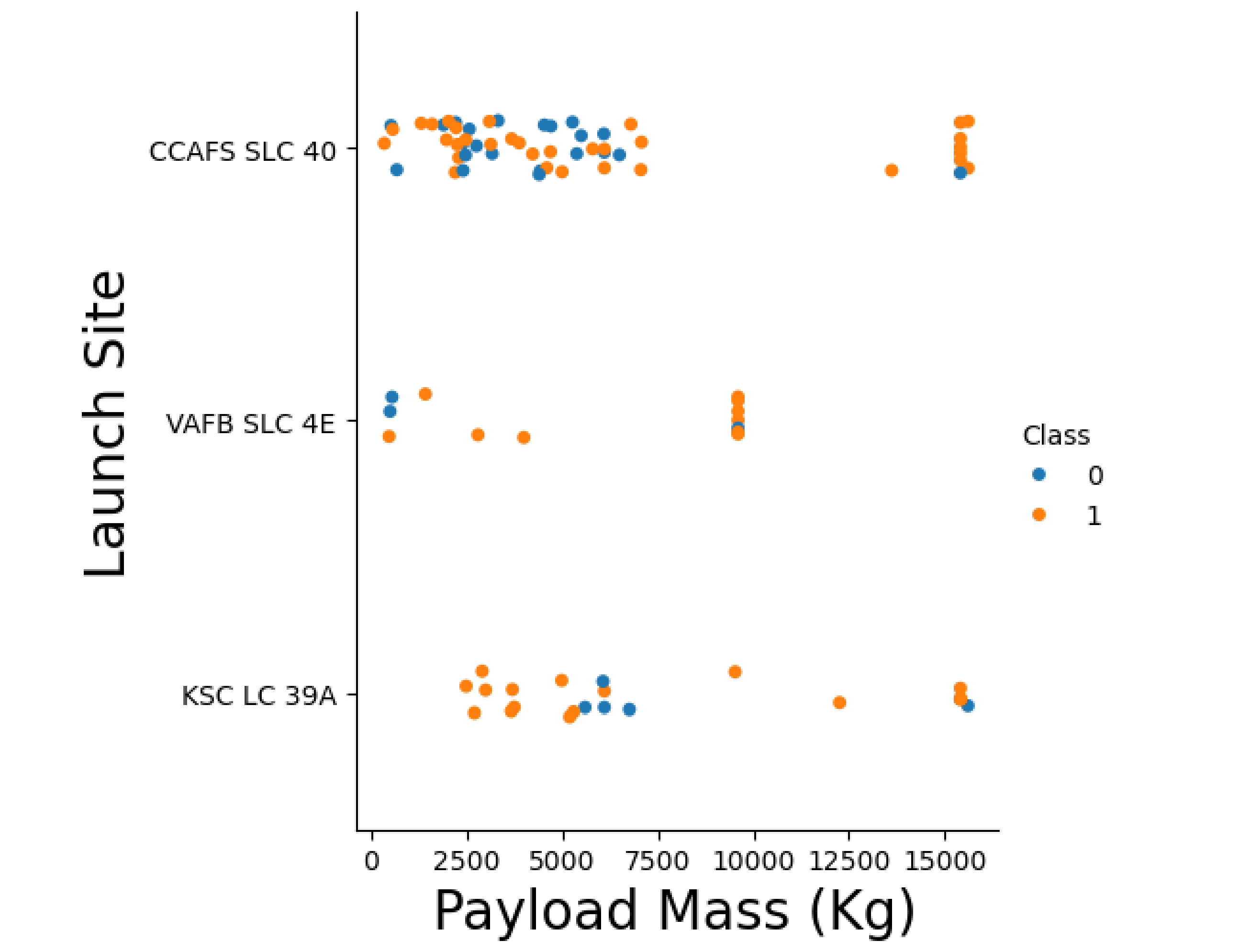
Flight Number vs. Launch Site

- Blue dots – fail; orange dots – success;
- There's an increase in the number of orange dots with the Flight number, suggesting an increasing success with time;
- It seems that KSC LC 39A and VAFB SLC 4E have higher success rates than CCAFS SLC 40.



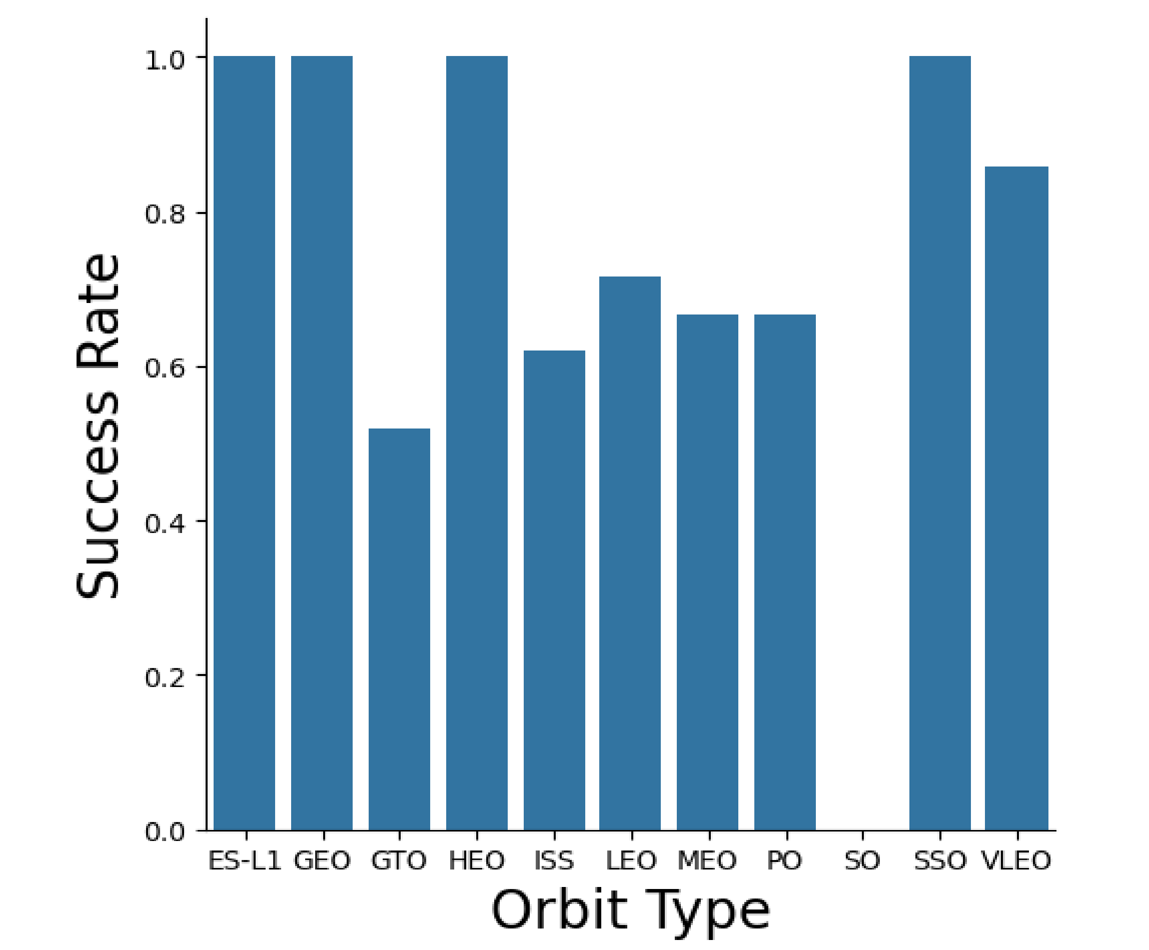
Payload vs. Launch Site

- There’s an increase in success rates with the Payload Mass (Kg);
- Most launches were done below the 7500Kg threshold.
- Above 7500Kg, success rates increase substantially;
- Below 5000Kg, KSC LC 39A has 100% success.



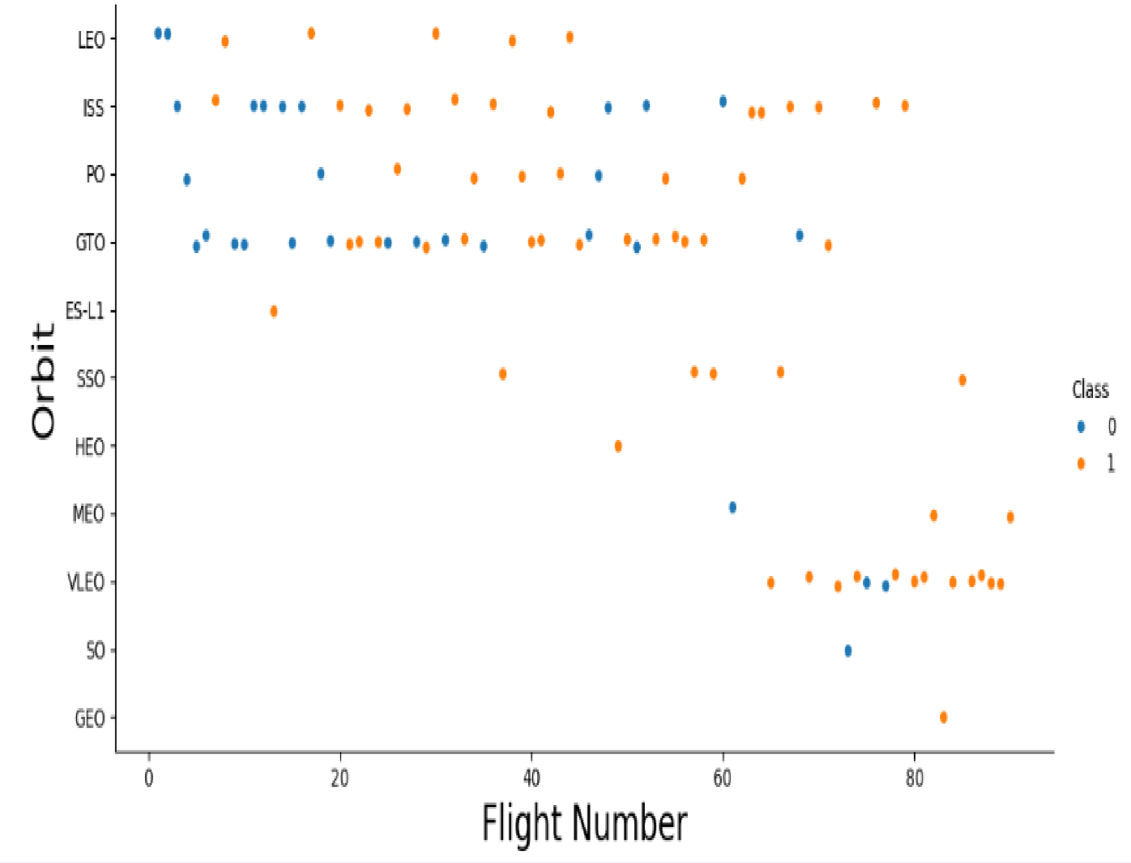
Success Rate vs. Orbit Type

- Orbits ES-L1, GEO, HEO and SSO have a success rate of 100%;
- On the other hand, SO orbit show 0% of success rate.



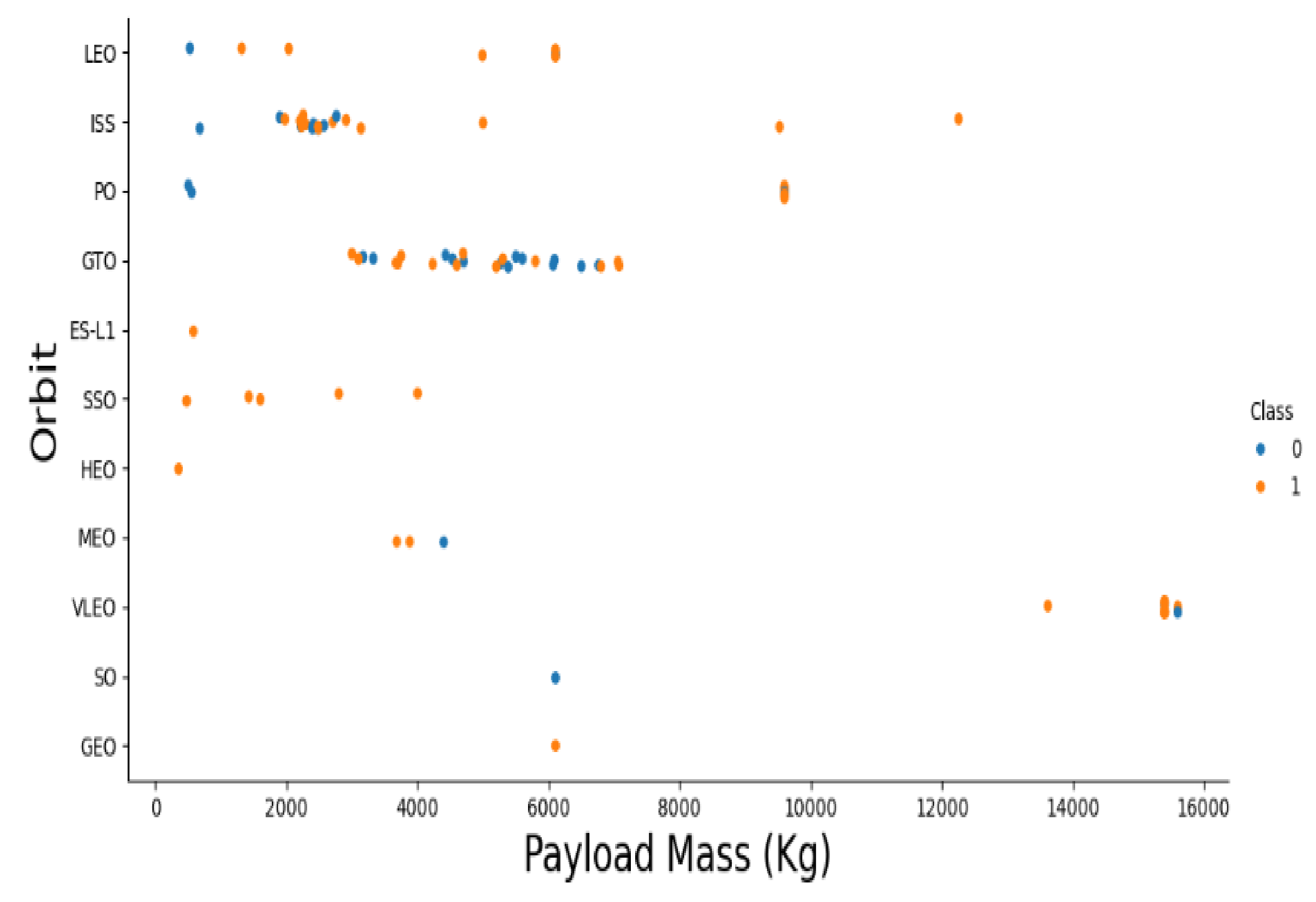
Flight Number vs. Orbit Type

- From a general point of view, with the increase in the number of flights, the number os success landings also increases, with the exception being the GTO orbit.
- SSO orbit does have a 100% of landing success rate.



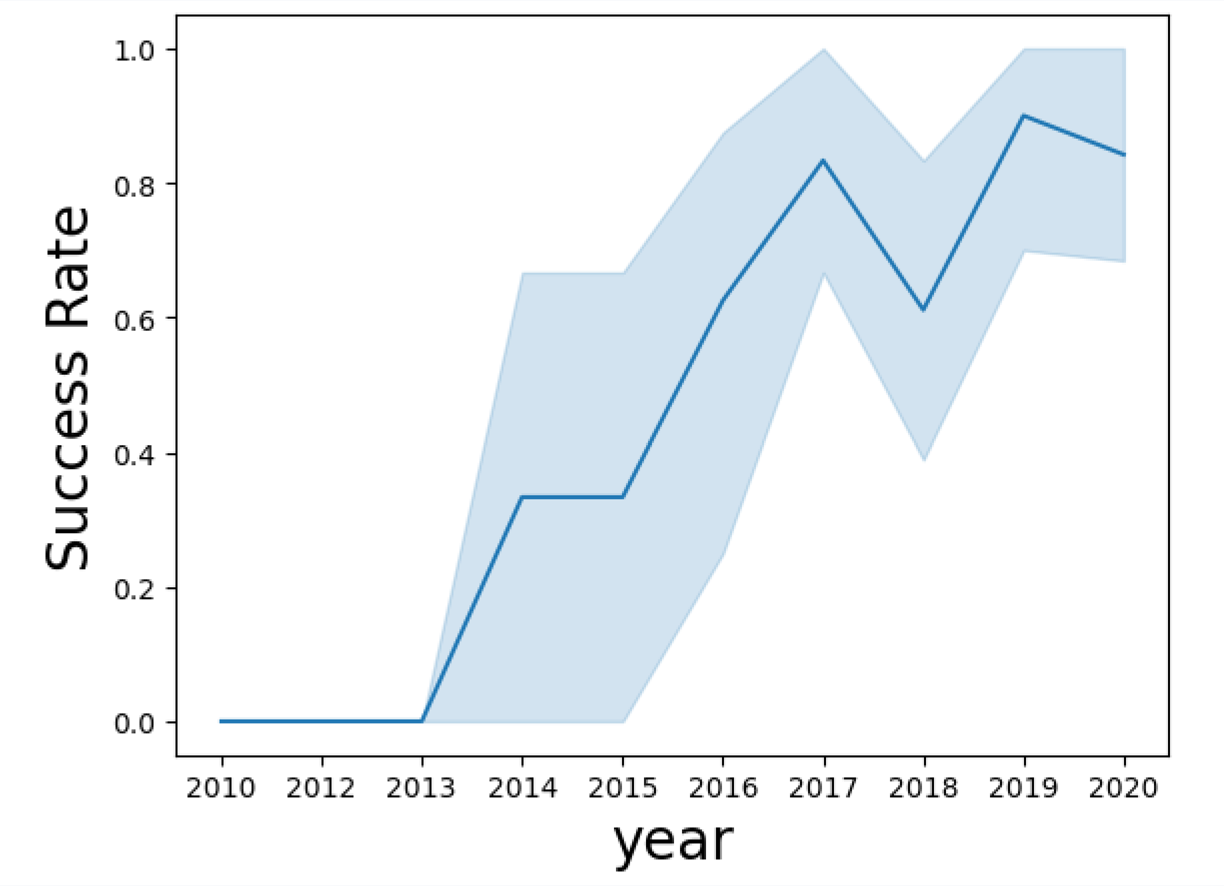
Payload vs. Orbit Type

- Overall, with the increase in the Payload Mass (Kg), there's an increase in the number os successful landings;
- GTO orbit seems to be the one where this trend does not occur;
- SSO orbit has a successful landing rate of 100%.



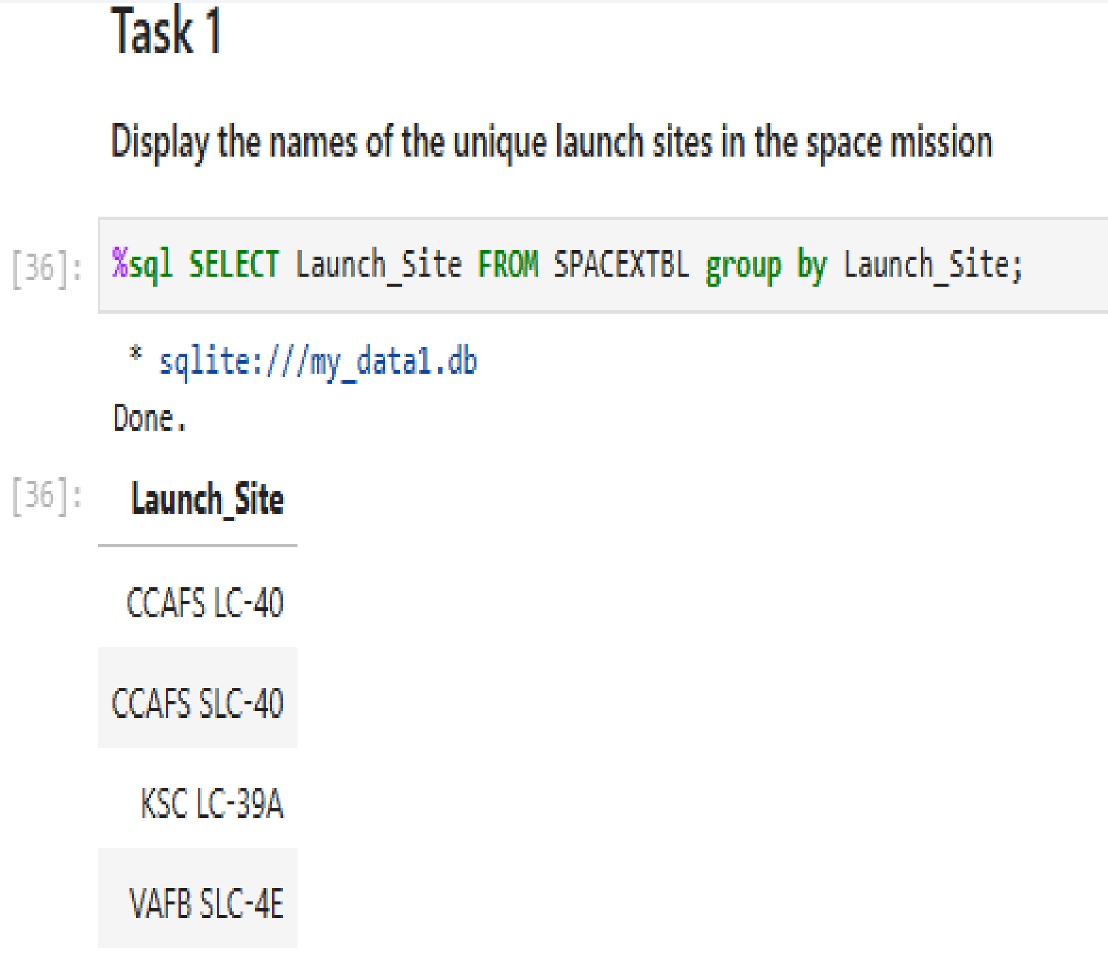
Launch Success Yearly Trend

- As time goes by, the success rate increases.



All Launch Site Names

- According to the query used, the names of the Launch Sites are:
  - CCAFS LC-40;
  - CCAFS SLC-40;
  - KSC LC 39A
  - VAFB SLC 4E
- I selected the column ‘Launch_Site and grouped by ‘Launch_Site in order to have just the names of the Launch Site;
- The query ‘Unique’ didn’t work, I don’t know why.



Task 1

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

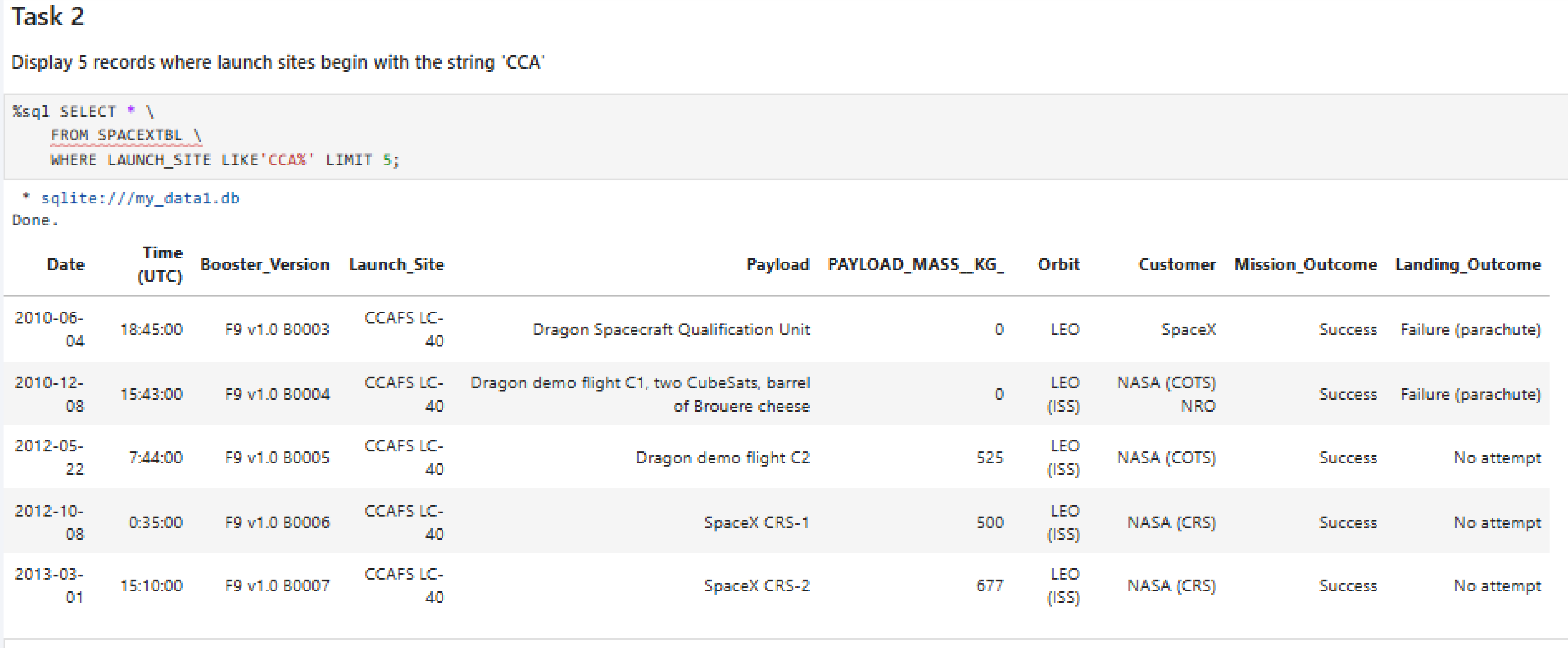
```
[36]: %sql SELECT Launch_Site FROM SPACEXTBL group by Launch_Site;
```

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

[36]:

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

Launch Site Names Begin with 'CCA'



Task 2

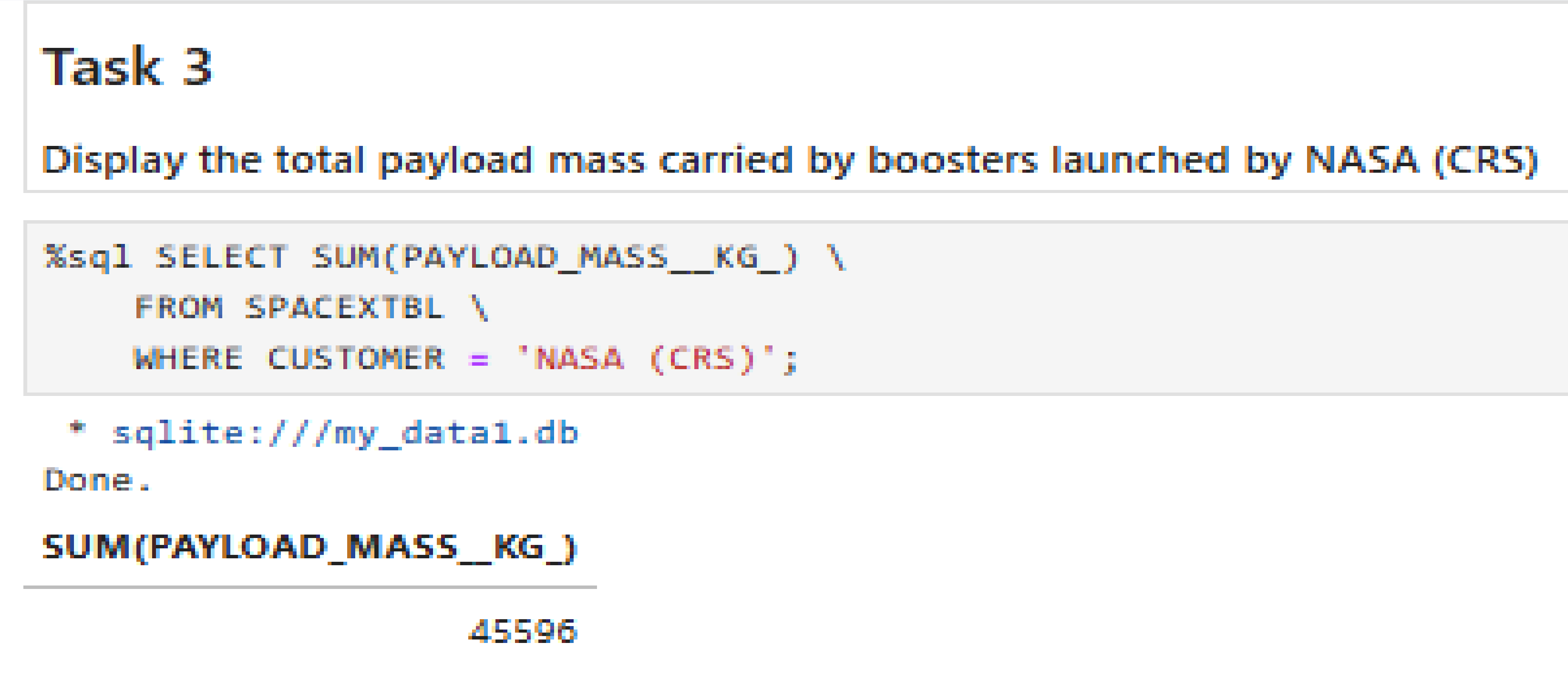
Display 5 records where launch sites begin with the string 'CCA'

```
%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



Task 3

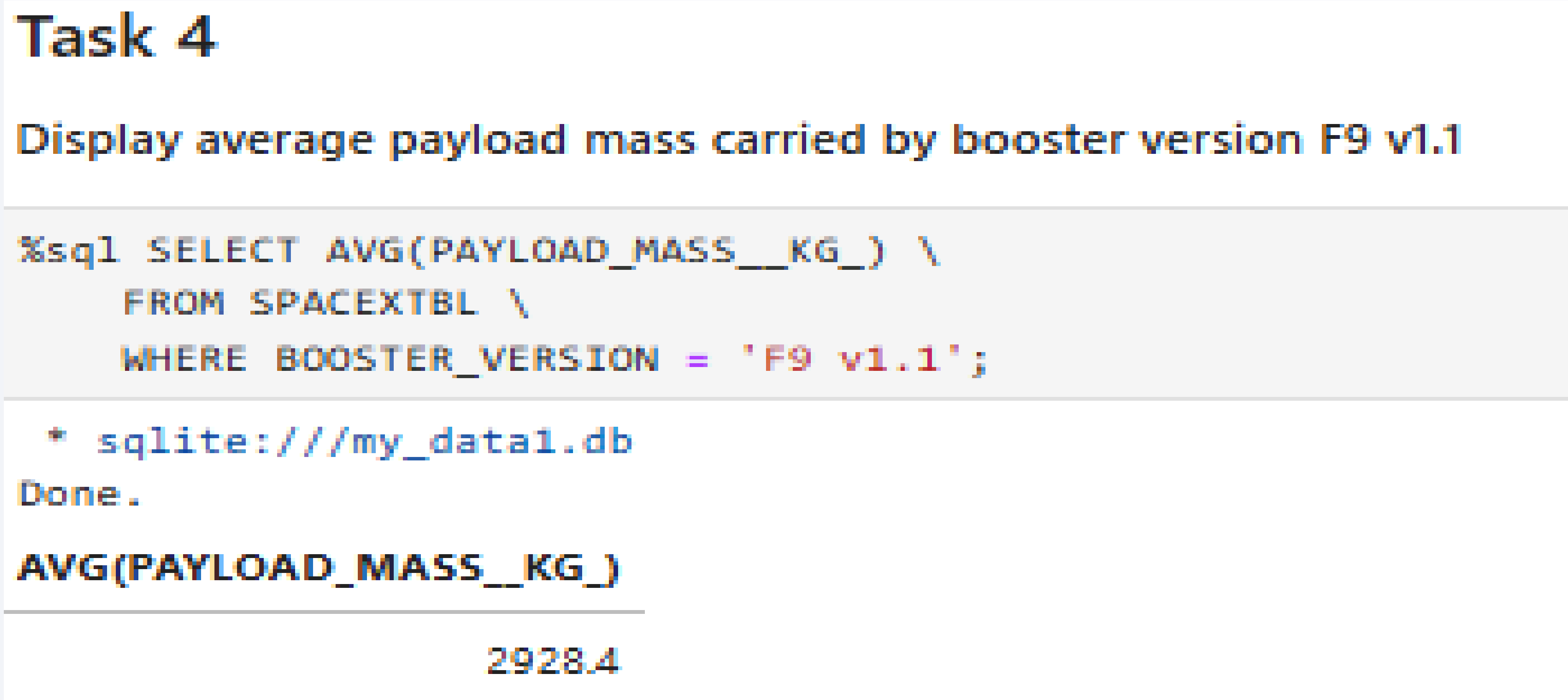
Display the total payload mass carried by boosters launched by NASA (CRS)

```
%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



Task 4

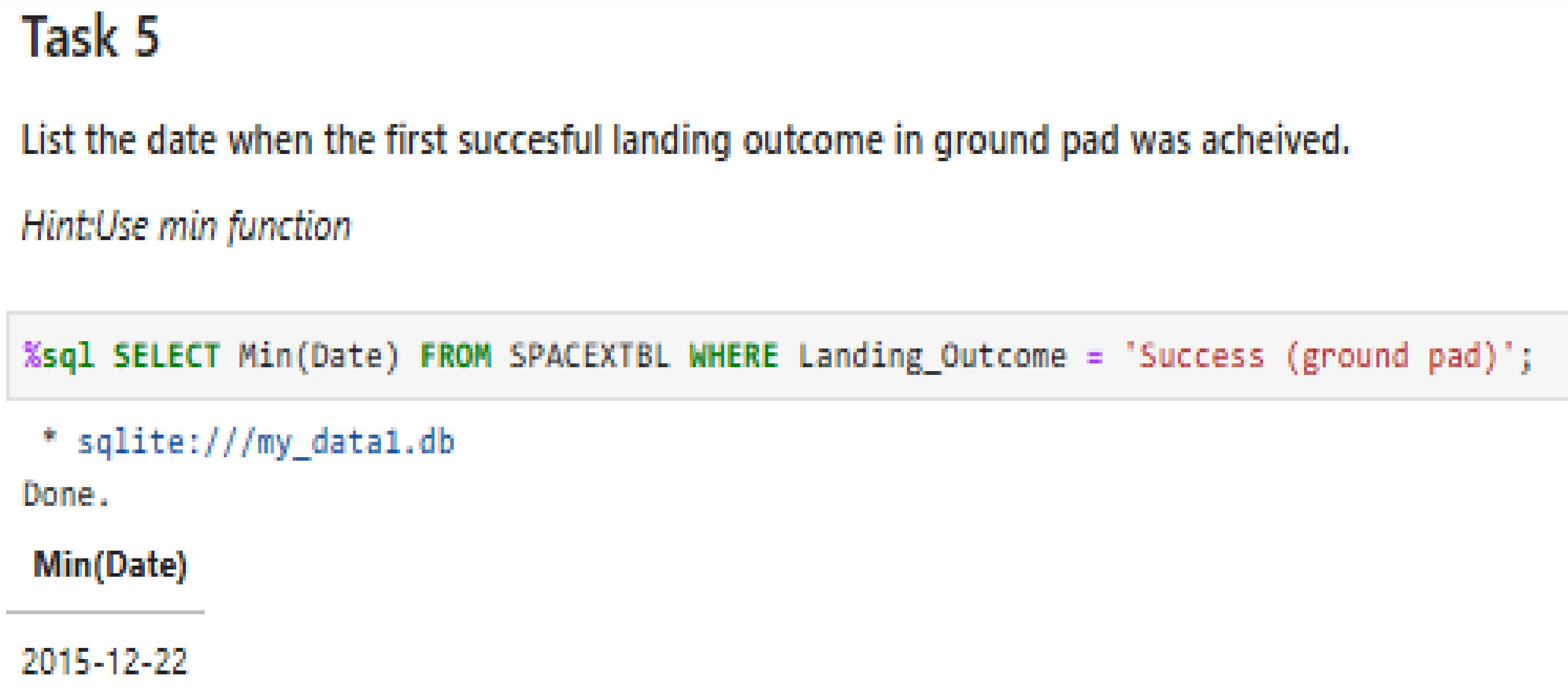
Display average payload mass carried by booster version F9 v1.1

```
%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



Task 5

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

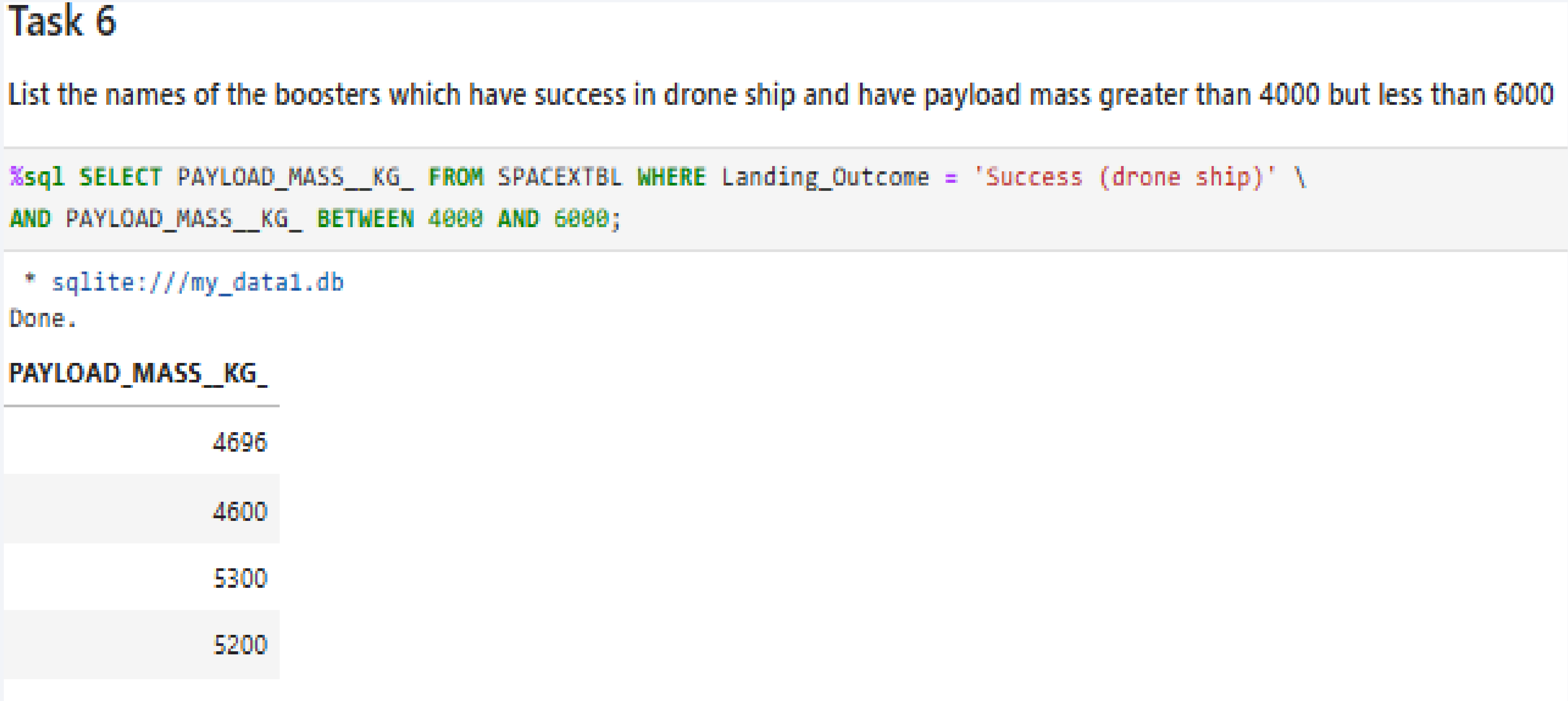
Hint:Use min function

```
%sql SELECT Min(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000



Task 6

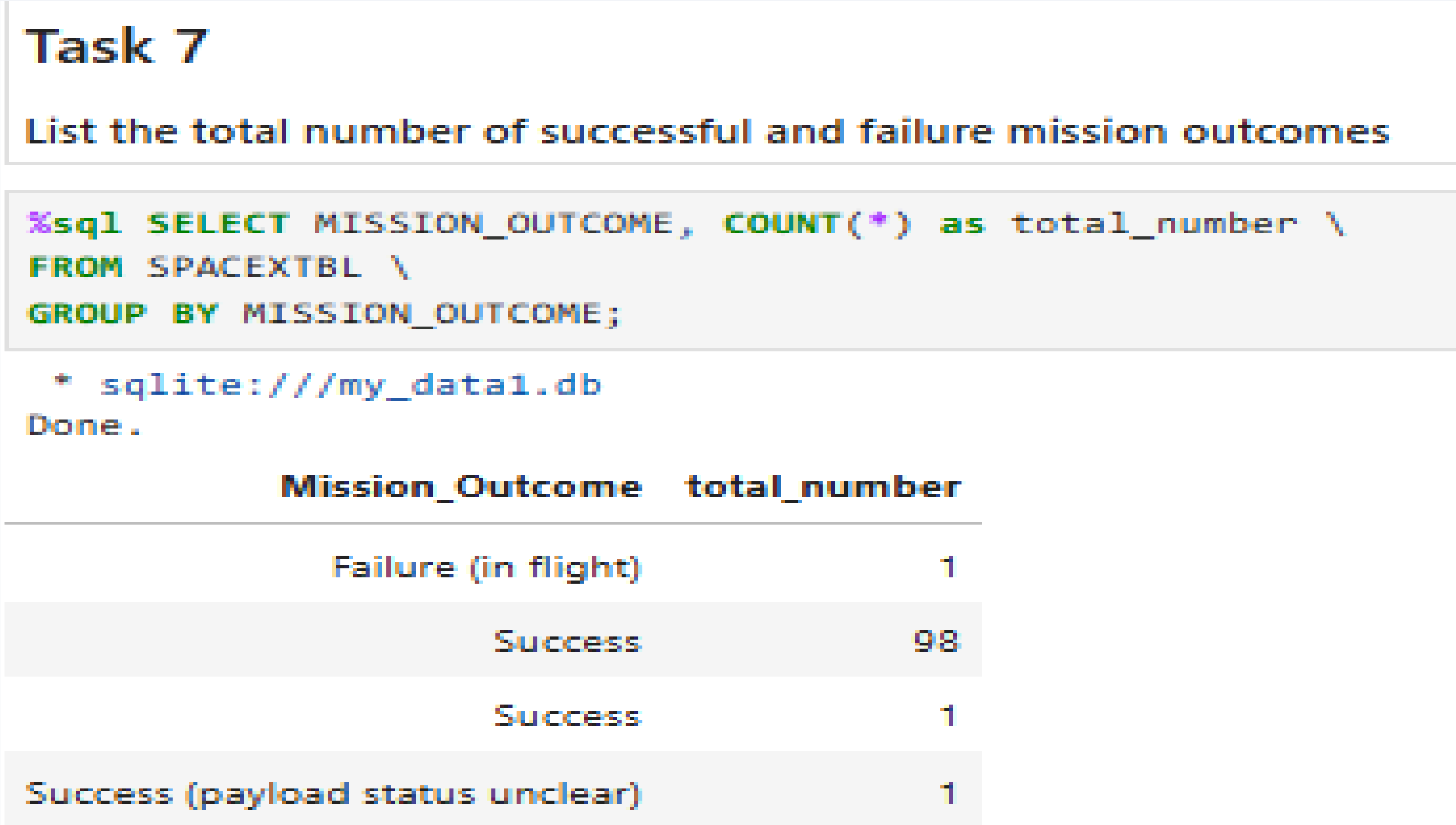
List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

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

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

PAYLOAD_MASS__KG_
4696
4600
5300
5200

Total Number of Successful and Failure Mission Outcomes



Task 7

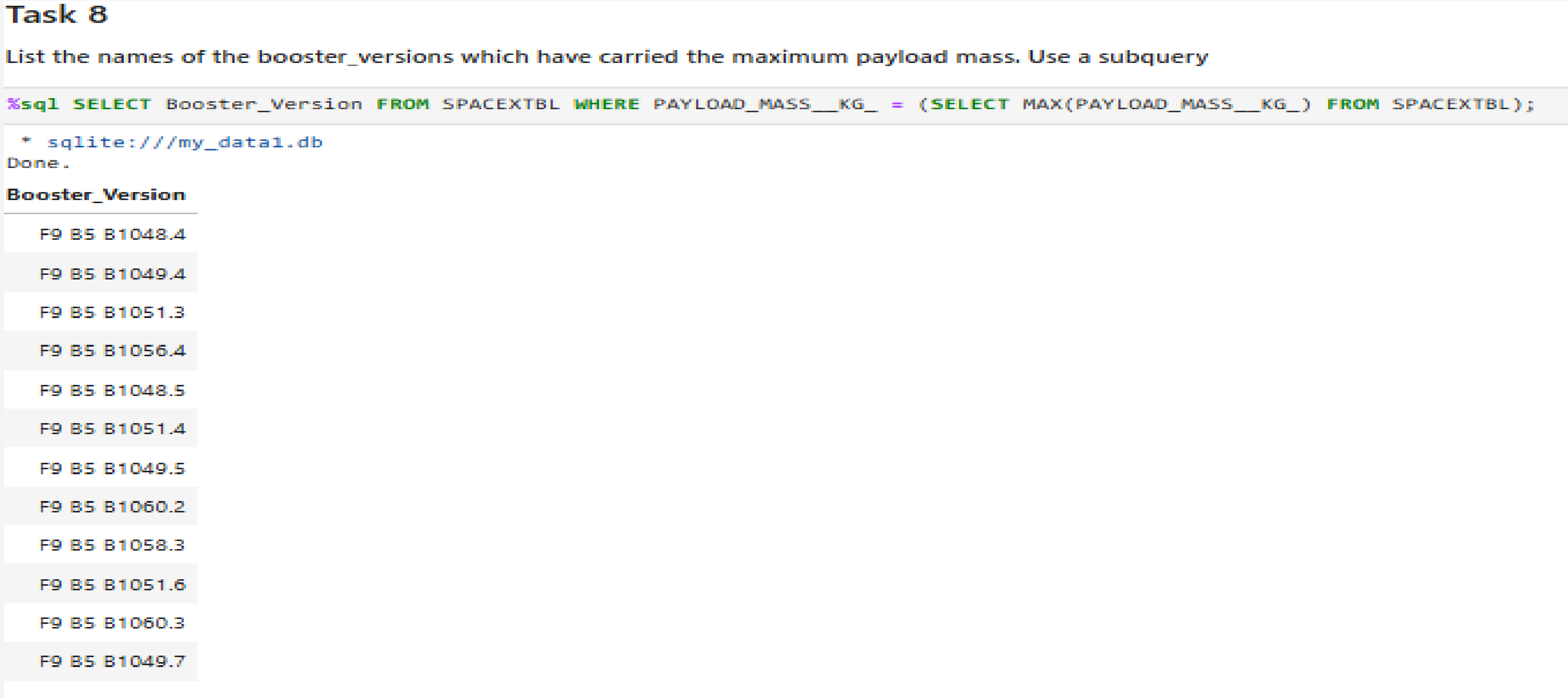
List the total number of successful and failure mission outcomes

```
%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



Task 8

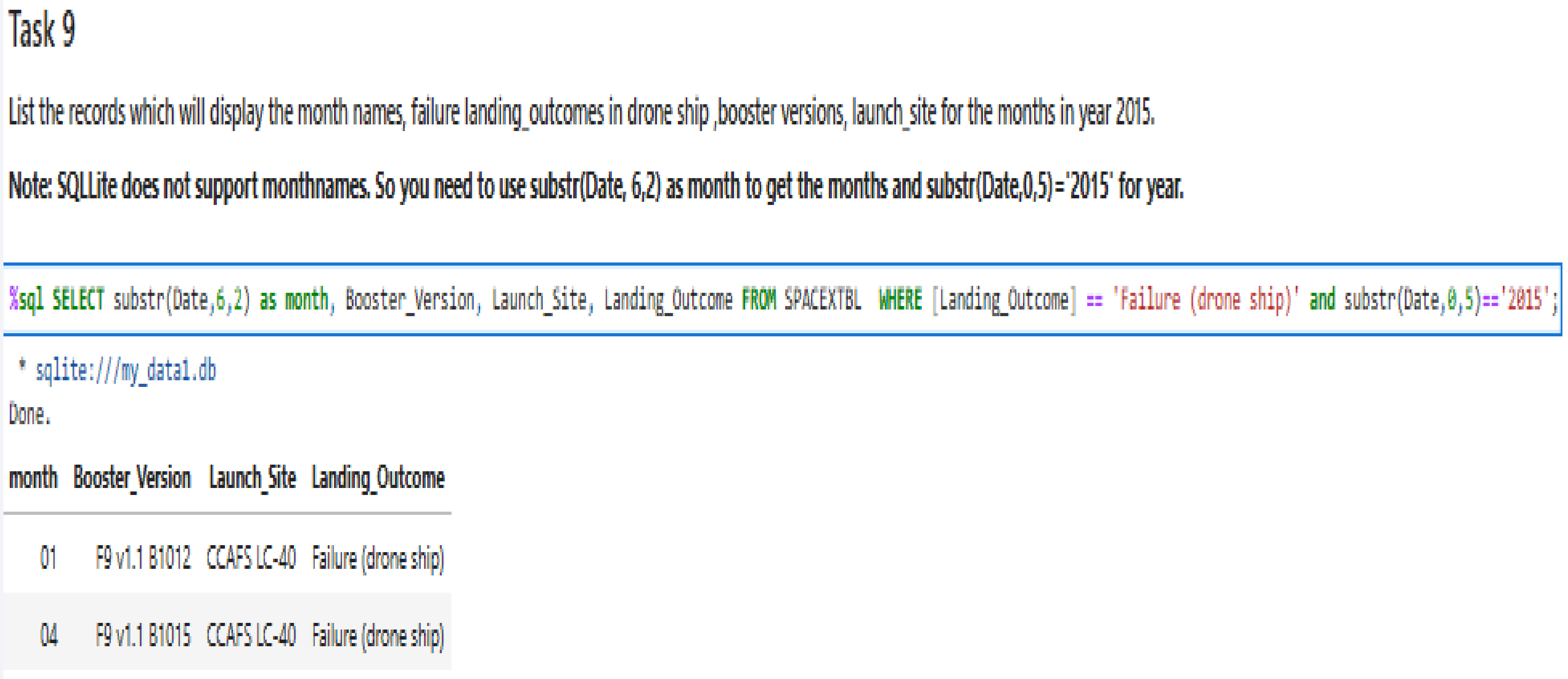
List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%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



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.

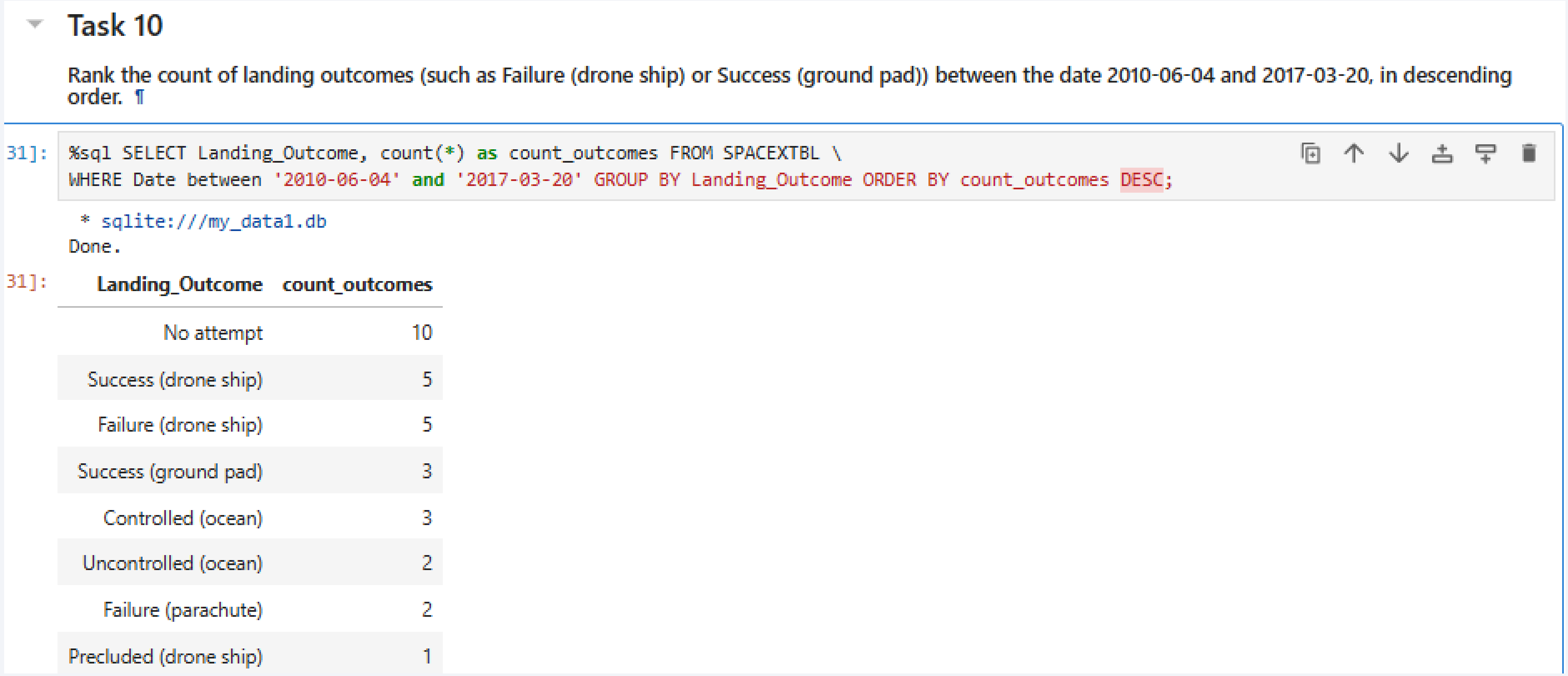
Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date,6,2) as month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL  WHERE [Landing_Outcome] == 'Failure (drone ship)' and substr(Date,0,5)=='2015';
```

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

month	Booster_Version	Launch_Site	Landing_Outcome
01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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



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. ¶

31]:

```
%sql SELECT Landing_Outcome, count(*) as count_outcomes FROM SPACEXTBL \
WHERE Date between '2010-06-04' and '2017-03-20' GROUP BY Landing_Outcome ORDER BY count_outcomes DESC;
```

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

31]:

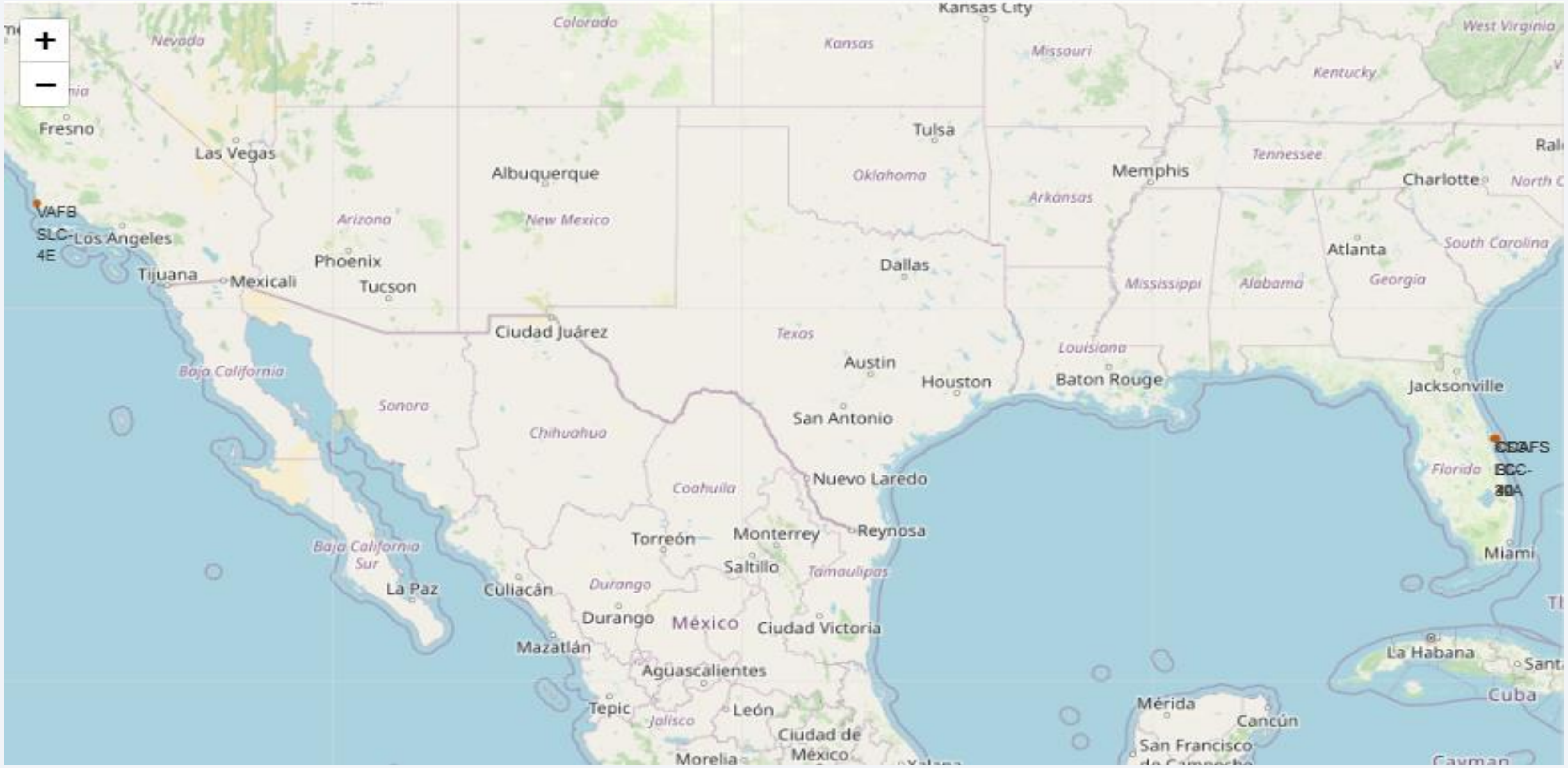
Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

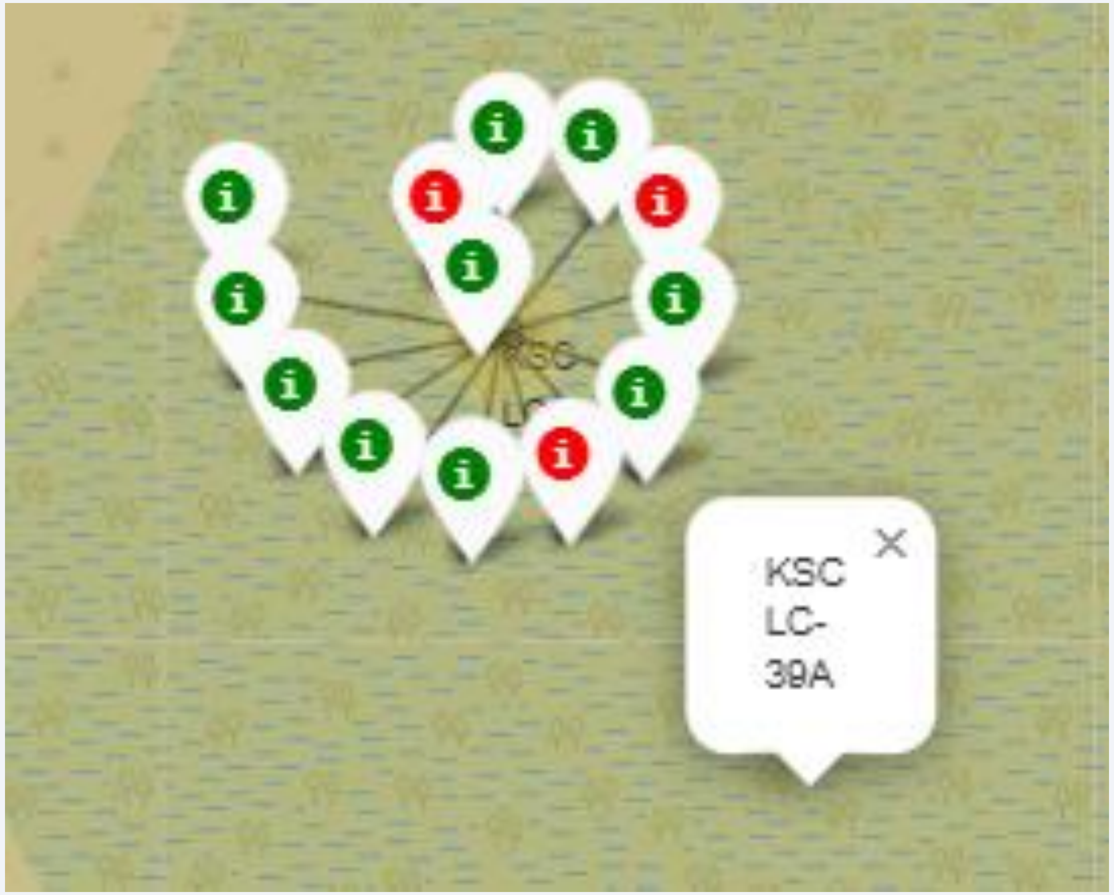
Launch Sites

From the map, we can see that the location sites are on both sides of the American southern coast, relatively closed to the Equator. This helps saving costs in fuel consumption, since launching closer to the Equator you have higher initial speed thanks to Earth’s rotation.



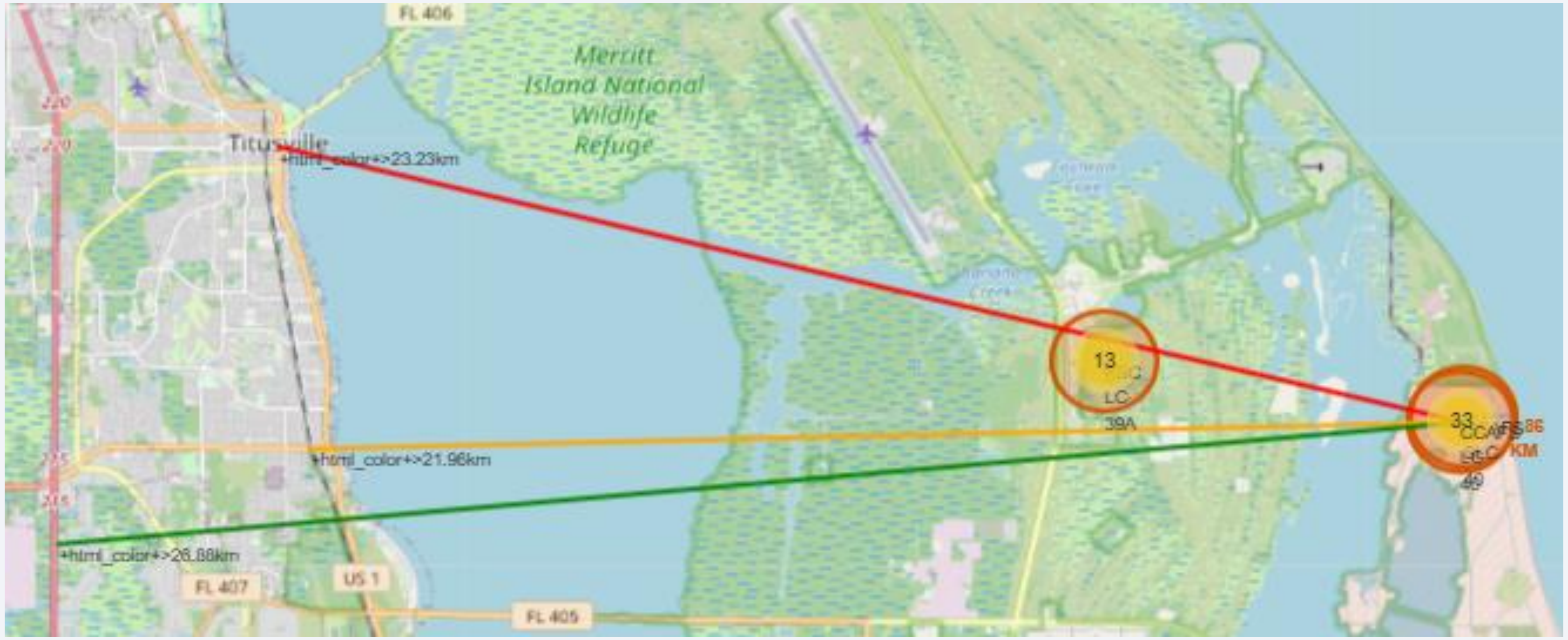
Launch Outcomes at KSC LC-39A

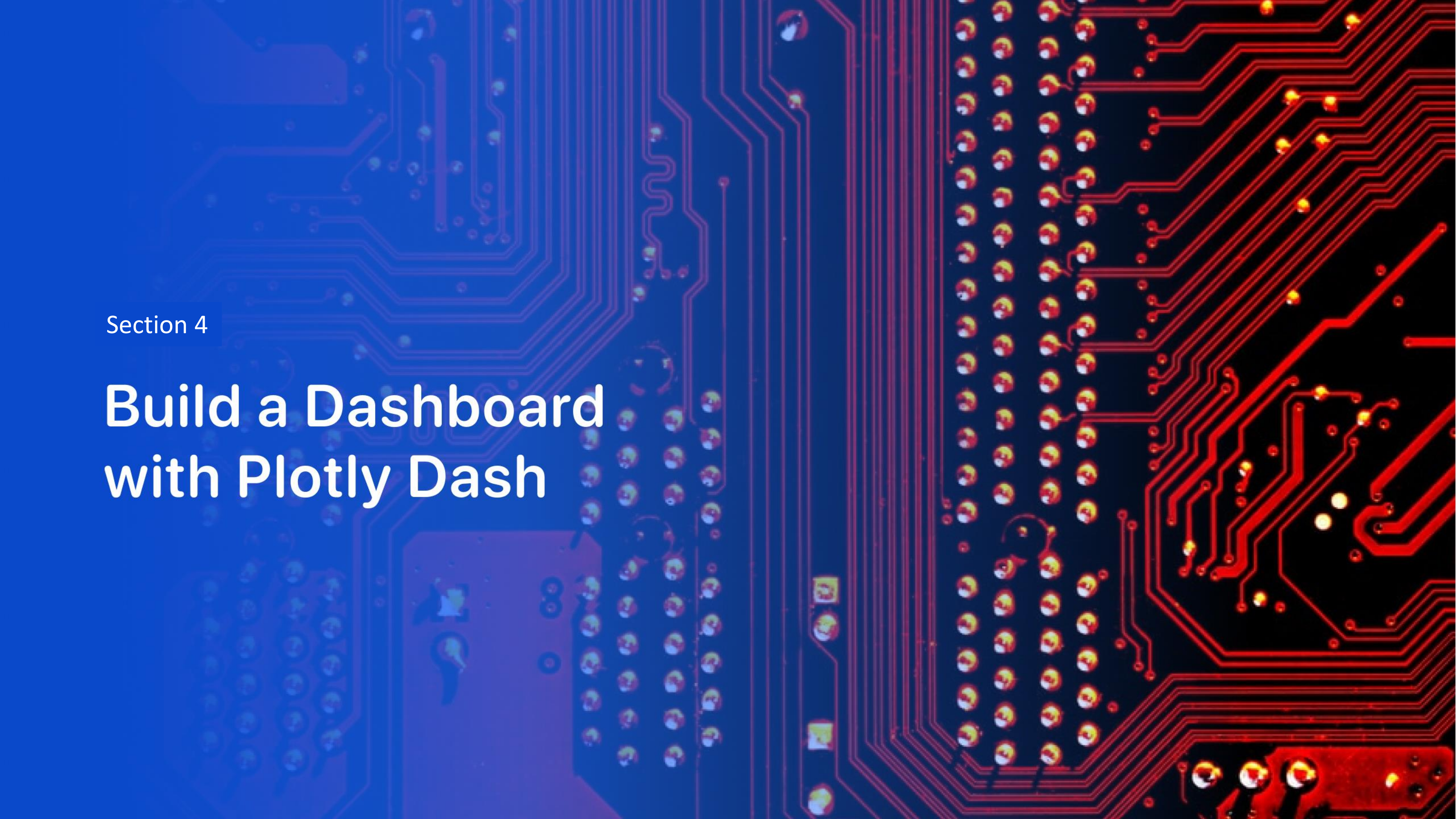
- Most launch outcomes at KSC LC-39A are in green, showing high rates of successful launches – 10/13, corresponding to almost 77%.



Distance to proximities from CCAFS SLC-40

- **Distance to the city**: 23.34Km
- **Distance to railway**: 21.96Km
- **Distance to Highway**: 26.88Km
- **Distance to coastline**: 0.86Km

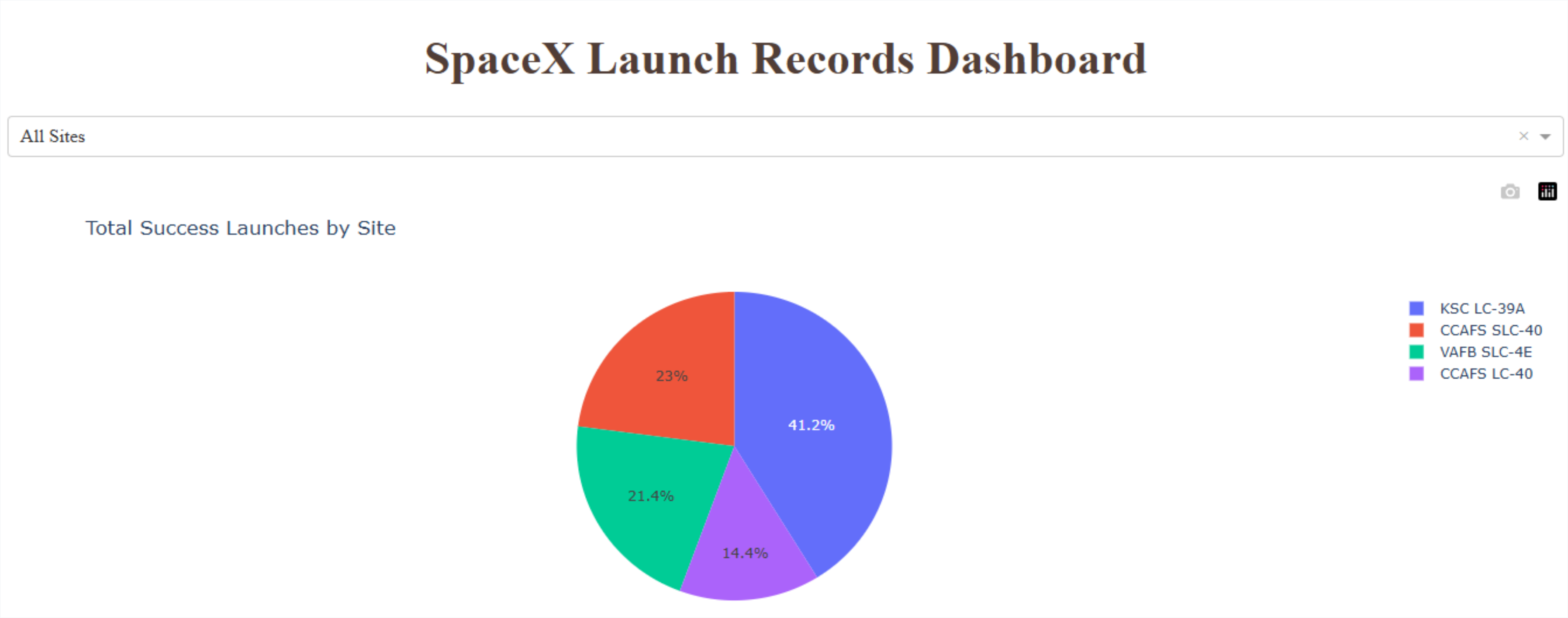




Section 4

Build a Dashboard with Plotly Dash

Launch Success



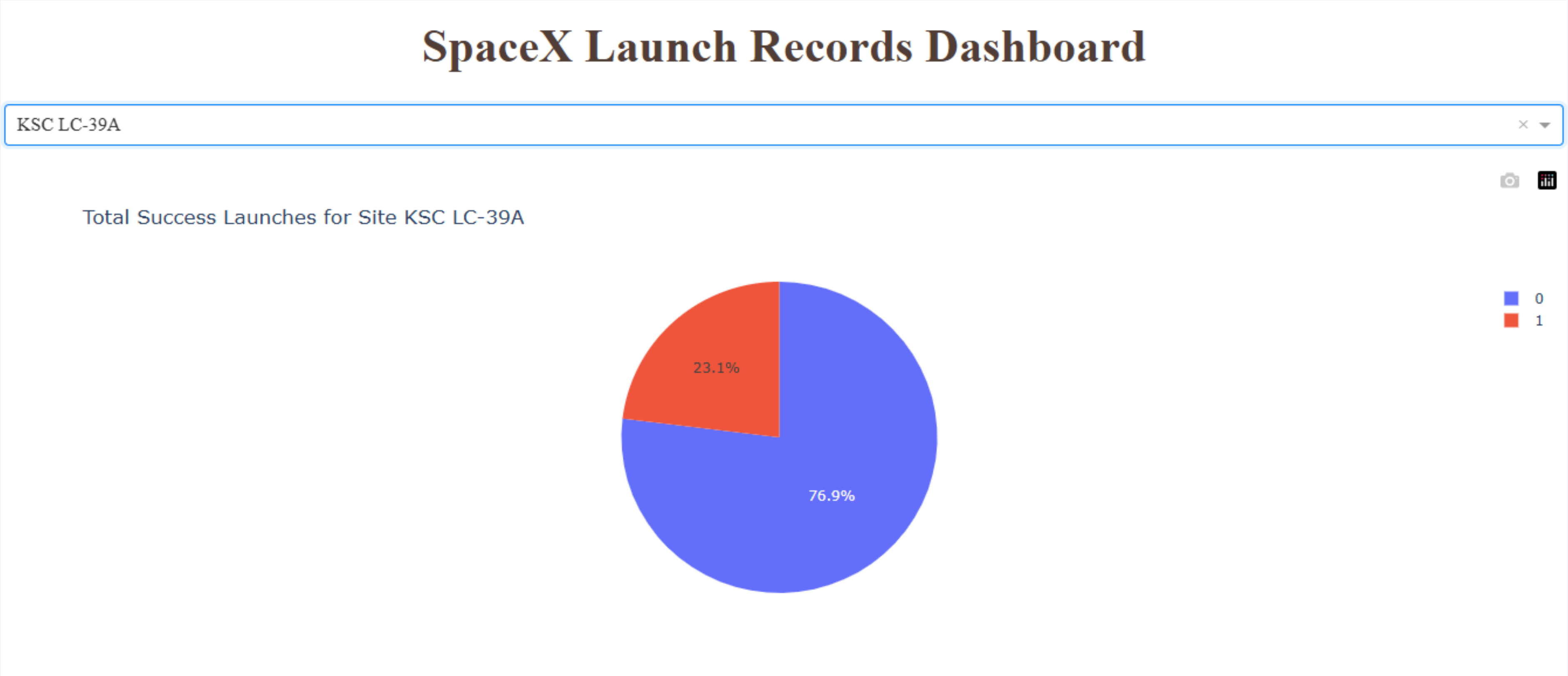
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by Site

- KSC LC-39A is the site with more launch sites, with 41% of them being successful.
- On the other hand, CCAFS LC-40 is the site with less successful launches with 14.4%.

Launch Success Rate of KSC LC-39A

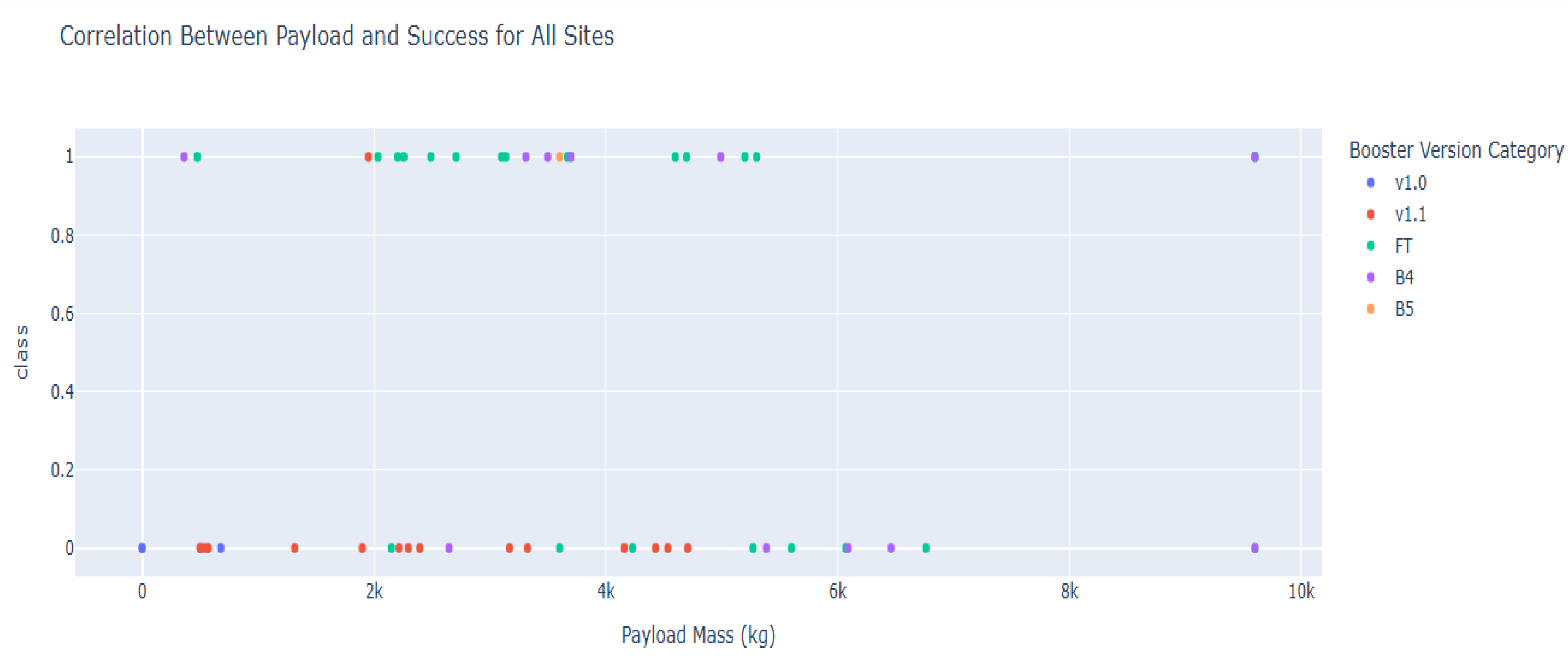


SpaceX Launch Records Dashboard

KSC LC-39A

- Site KSC LC-39A has a success rate of almost 77%.

Success by Payload Mass (Kg) and Booster Version

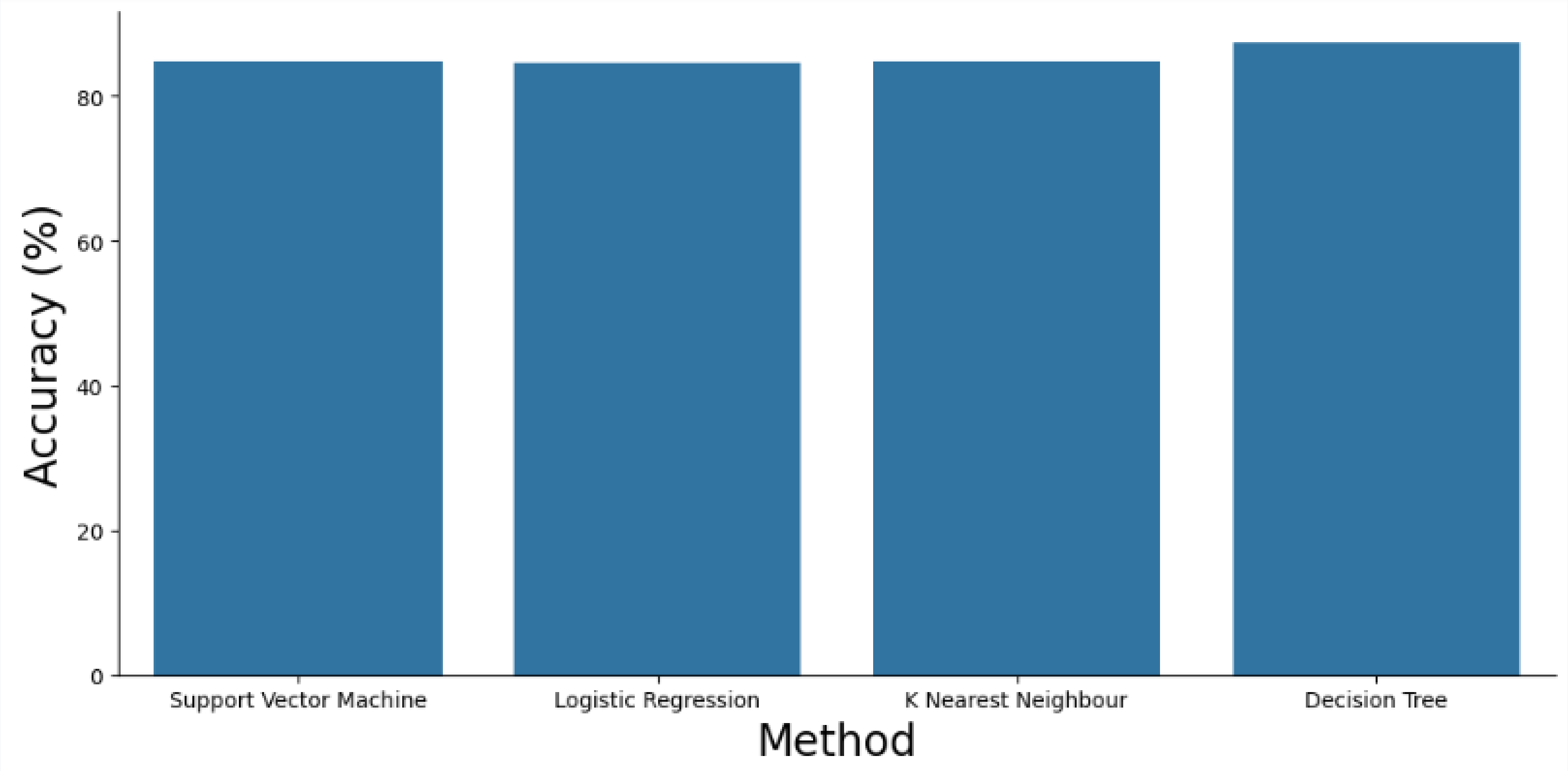


- Best Payload Mass (Kg) range is between 2000 and 4000.
- Best Booster Version is FT, which is the one in green.

Section 5

Predictive Analysis (Classification)

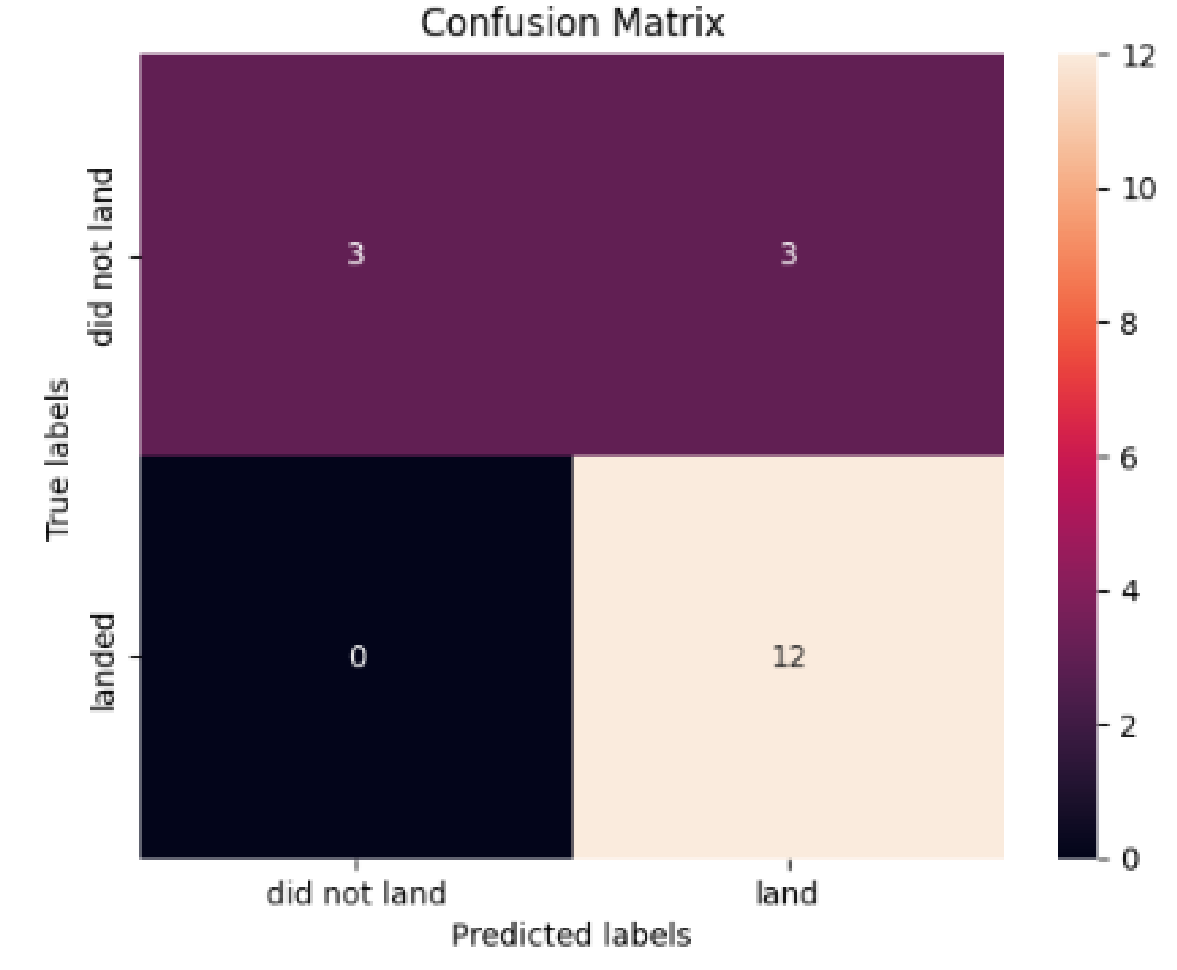
Classification Accuracy



- Decision Tree Classification method is slightly better than the other 3, with 87% accuracy.

Confusion Matrix

- Confusion matrices are useful to synthesize a model's performance;
- False Positives are a Type 1 Error;
- We have:
    - 12 True Positives
    - 0 False Negatives
    - 3 False Positives
    - 3 True Negatives



Conclusions

- Launch site KSC LA-39A is the best launch site of them all;
- Orbits ES-L1, GEO, HEO and SSO have a success rate of 100%;
- Launch sites are all near the coast so the debris falls into the water, and near the equator to make the most of earth's rotational speed, saving fuel costs;
- There's an increase in launch success with the passing of time;
- Best classification method is Decision Tree, having slightly better results than the other methods.

Thank you!