IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Michael Lee
November 27, 2022

Outline

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

Executive Summary

Summary of Methodologies:

The research attempts to identify the factors for a successful rocket landing. To make this determination, the research used the following methodologies. First, we collected data using SpaceX REST API and web scraping techniques. Second, we wrangled our collected data to create a success/fail outcome variable. Third, we explored our data with visualization techniques, considering factors such as payload, launch site, flight number, and yearly trend. Fourth, we analyzed our data with SQL, calculating such statistics as total payload, the payload range for successful launches, and total number of successful and failure mission outcomes. Fifth, we explored launch site success rates and proximities to geographic markers using an interactive map built with Folium. Sixth, we built a dashboard with Plotly Dash in order to show which launch site had the most success and to examine successful payload ranges. Lastly, we used predictive models – logistic regression, support vector machines (SVM), k-nearest neighbor (KNN), and decision trees – to predict landing outcomes. We found that for our data set that SVM model was the most accurate.

Summary of All Results:

- Exploratory data analysis results: The success rate of launches increases over time; KSC LC-39A has the highest success rate among the launch sites; orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.
- Interactive analytics demo in screenshots: Most of the launch sites are in proximity to the equator and all of the launch sites are close to the coast.
- Predictive analysis results: we found that the SVM is the best predictive model for our dataset.

Introduction

- Project background and context

  SpaceX, a leading company in the commercial space industry, strives to make space travel more cost effective. The company developed the Falcon 9 rocket, which costs $62 million per launch because of its novel reuse of the first stage; other companies in commercial space travel, which do not reuse the first stage, charge $165 million per launch. Therefore, assuming the first stage can be reused, determining the success of a first-stage landing can determine the cost per launch. We will therefore use public data and machine learning models to predict if SpaceX (or a competing company) can reuse the first stage.

- For this project, we will explore answers to the following questions:
  - How do variables such as payload mass, launch site, number of flights, and orbits affect the success of the first-stage landing?
  - Does the rate of successful landings increase over time?
  - What is the best predictive method that can be applied for binary classification?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
  - Using SpaceX REST API and Web Scraping from Wikipedia
- Perform data wrangling
  - Filtering the data; handling with missing value; applying One Hot Encoding to 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
  - Building, tuning, and evaluation of classification models to ensure the best results.

Data Collection

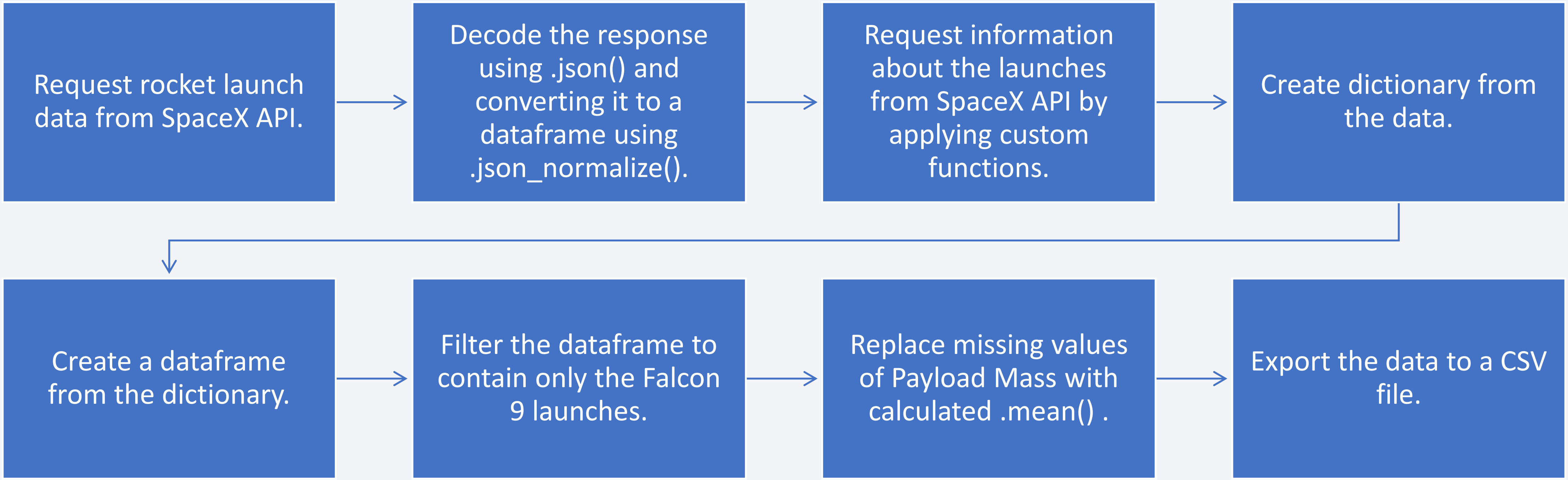
- The data collection process employed API requests from SpaceX REST API and Web Scraping data from a table in SpaceX's Wikipedia page.

- I used both data collection methods to obtain the needed information on the launches to conduct a more detailed analysis.

- I obtained the following data columns using SpaceX REST API:
  - FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude.

- I obtained the following data columns using Web Scraping:
  - Flight No., Launch Site, Payload, PayLoadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.

Data Collection – SpaceX API



Request rocket launch data from SpaceX API.

Decode the response using .json() and converting it to a dataframe using .json_normalize().

Request information about the launches from SpaceX API by applying custom functions.

Create dictionary from the data.

Create a dataframe from the dictionary.

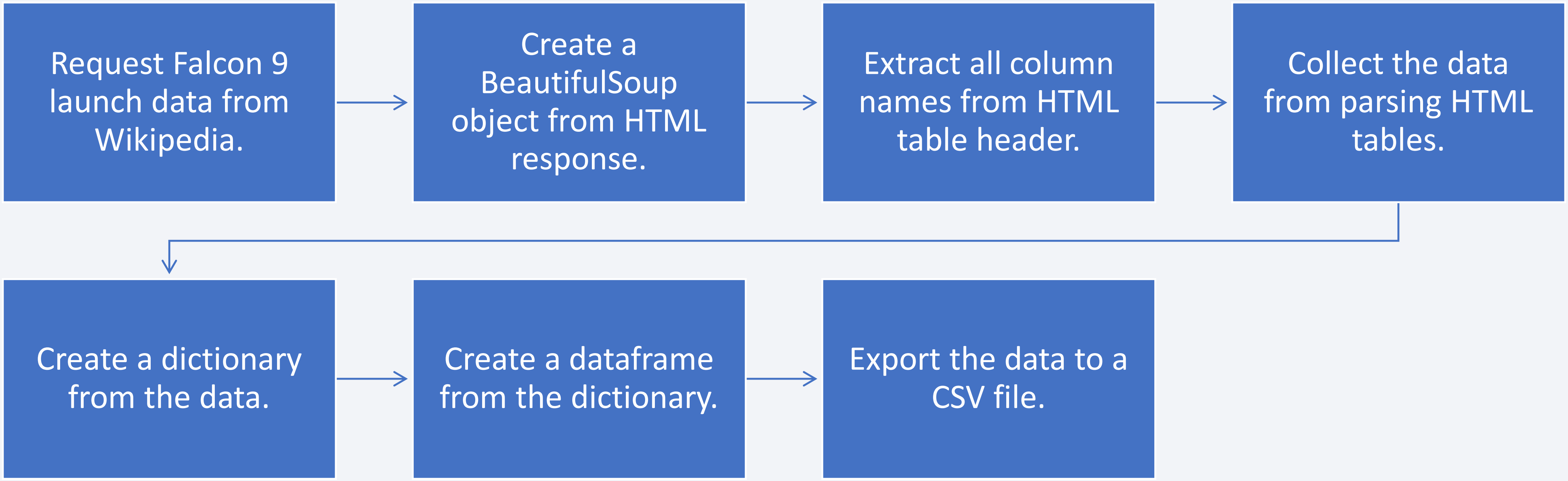
Filter the dataframe to contain only the Falcon 9 launches.

Replace missing values of Payload Mass with calculated .mean() .

Export the data to a CSV file.

[GitHub URL for SpaceX API Notebook]

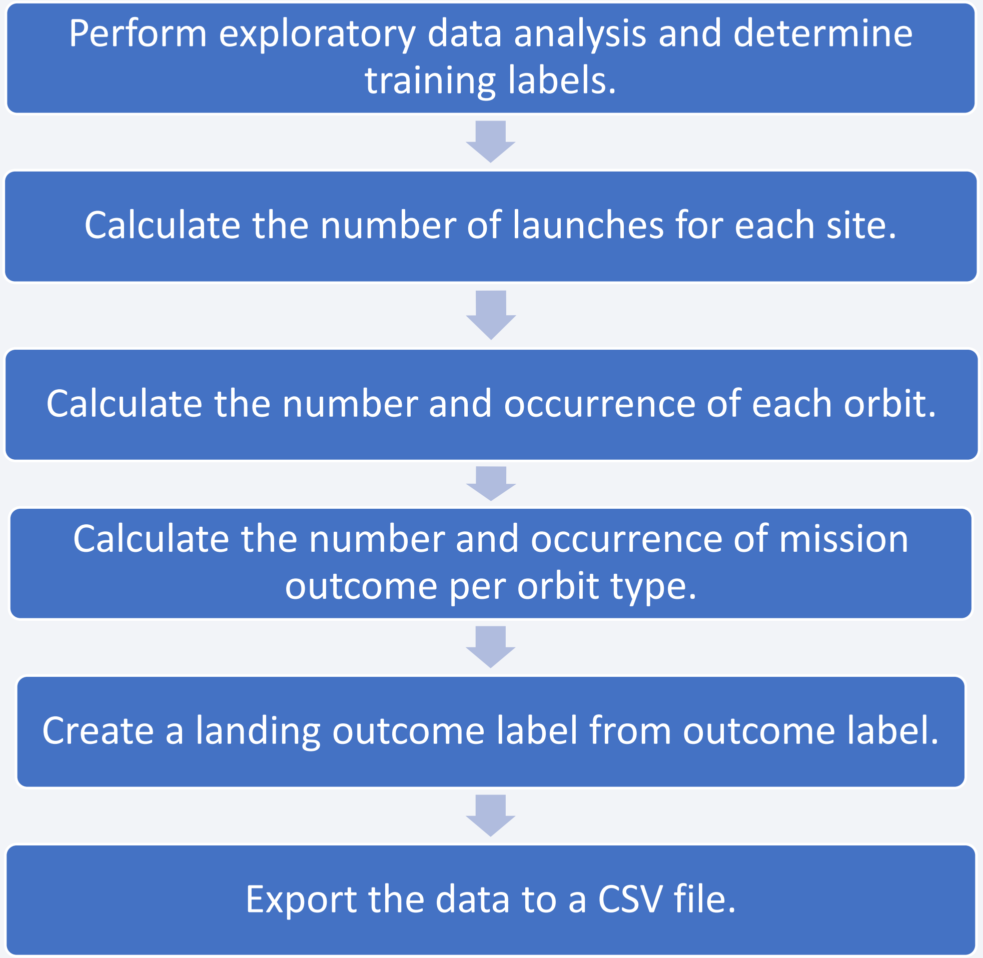
Data Collection – Web Scraping


Request Falcon 9 launch data from Wikipedia.
Create a BeautifulSoup object from HTML response.
Extract all column names from HTML table header.
Collect the data from parsing HTML tables.
Create a dictionary from the data.
Create a dataframe from the dictionary.
Export the data to a CSV file.

GitHub URL for Data Collection with Web Scraping Notebook

Data Wrangling

- In the data set, the booster did not always land successfully: a landing could be attempted but could fail because of an accident.
- True Ocean meant that the mission outcome had a successful landing to a specific region of the ocean; a False Ocean meant that the mission had an unsuccessful landing to a specific region.
- True RTLS meant that the mission outcome had a successful landing on a ground pad; a False RTLS meant that the mission had an unsuccessful landing to a ground pad.
- True ASDS meant that the mission outcome had a successful landing on a drone ship; False ASDS meant that the mission had an unsuccessful landing on a drone ship.
- We convert those outcomes into training labels with “1” meaning the booster successfully landed, and “0” meaning that the landing was unsuccessful.



Perform exploratory data analysis and determine training labels.

Calculate the number of launches for 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 label.

Export the data to a CSV file.

GitHub URL for Data Wrangling Notebook

EDA with Data Visualization

The following charts were plotted: Flight Number versus Payload; Flight Number versus Launch Site; Payload Mass versus Launch Site; Orbit Type versus Success Rate; Flight Number verses Orbit Type; Payload Mass versus Orbit Type; and Success Rate Year Trend.

Scatter plots show the relationship between variables. If a relationship exists, they could be useful in a machine learning model. Bar charts show comparisons among discrete categories. The goal of a bar chart is to show the relationships among the specific categories being compared and a measured value.

GitHub URL for EDA with Data Visualization Notebook

EDA with SQL

SQL Queries Performed:

- Displaying the names of the unique launch sites in the space mission.
- Displaying 5 records where launch sites begin the string ‘CCA’.
- Displaying the total payload mass carried by boosters launched by NASA (CRS).
- Displaying the 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 drop ship, their booster versions and launch site names for the months 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.

GitHub URL for EDA with SQL Notebook

Build an Interactive Map with Folium

- Markers of all Launch Sites:
  - Added Marker with Circle, Pop-up Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
  - Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to the Equator and coasts.
- Colored markers of the launch outcomes for each Launch Site:
  - Added colored markers of success (green) and failed (red) launches using Marker Cluster to identify which launch sites have relatively high success rates.
- Distances between a Launch Site to its proximities:
  - Added colored lines to show distances between the Launch Site CCAFS SLC-40, as an example, and its proximity to the nearest railway, highway, coastline, and closest city.

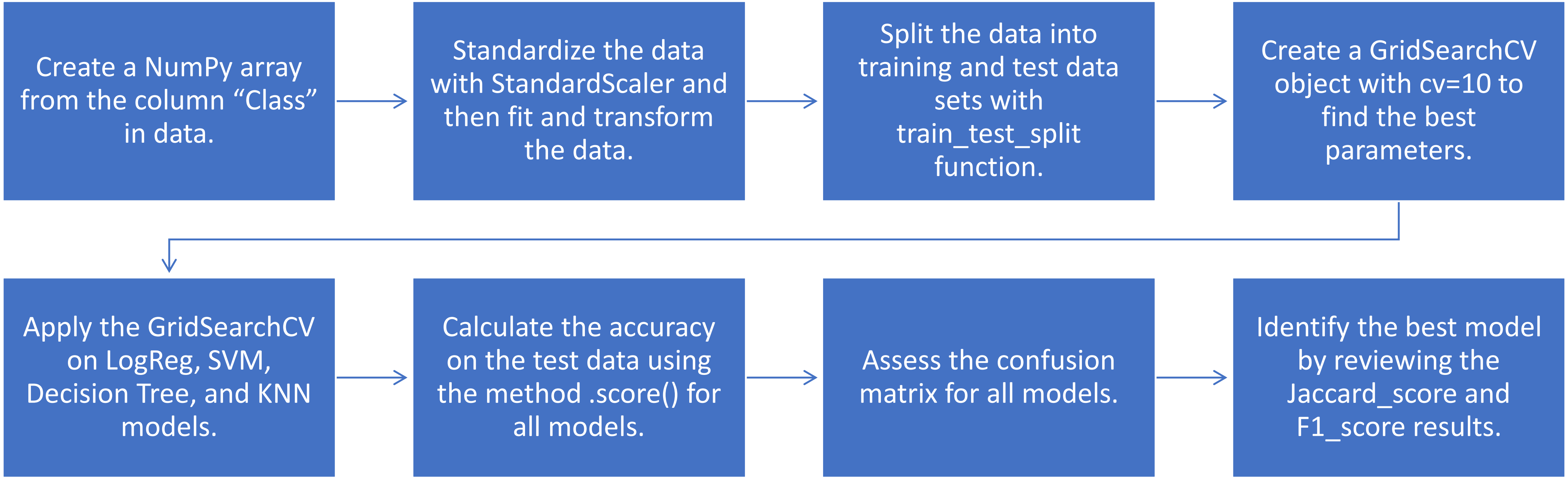
GitHub URL for Interactive Map with Folium (pdf of Notebook) [1]

Build a Dashboard with Plotly Dash

- Launch Sites Dropdown List:
  - Added a dropdown list to enable Launch Site selection.
- Pie Chart showing Successful Launches (All Sites/Certain Site):
  - Added a pie to show the total successful launches count for all sites and the success versus failed counts for the site, if a specific Launch Site was selected.
- Slider of Payload Mass Range:
  - Added a slider to select Payload range.
- Scatter Chart of Payload Mass versus Success Rate for the different Booster Versions:
  - Added a scatter chart to show the correlation between Payload and Launch Success.

GitHub URL for Plotly Dash App [2]

Predictive Analysis (Classification)



Create a NumPy array from the column “Class” in data.

Standardize the data with StandardScaler and then fit and transform the data.

Split the data into training and test data sets with train_test_split function.

Create a GridSearchCV object with cv=10 to find the best parameters.

Apply the GridSearchCV on LogReg, SVM, Decision Tree, and KNN models.

Calculate the accuracy on the test data using the method .score() for all models.

Assess the confusion matrix for all models.

Identify the best model by reviewing the Jaccard_score and F1_score results.

[GitHub URL for Machine Learning Prediction Lab]()

Results

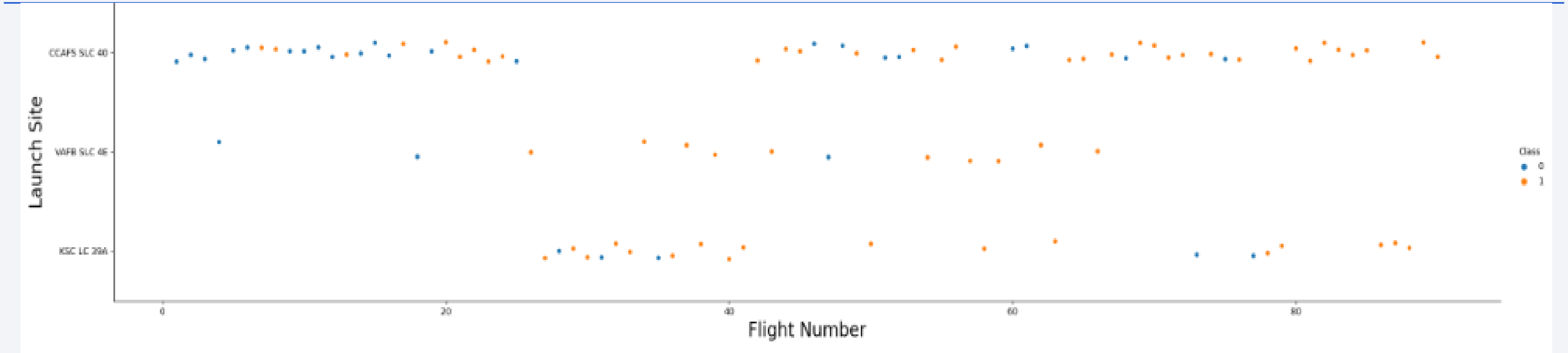
Sections 2, 3, 4, and 5 will present the results of this research in more detail, but we summarize some of our results below:

- Exploratory data analysis results:
  - The success rate of launches increases over time.
  - KSC LC-39A has the highest success rate among the launch sites.
  - Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.
- Interactive analytics demo in screenshots: Most of the launch sites are in proximity to the equator and all of the launch sites are close to the coast.
- Predictive analysis results: we found that SVM is the best predictive model for the dataset

Section 2

Insights drawn from EDA

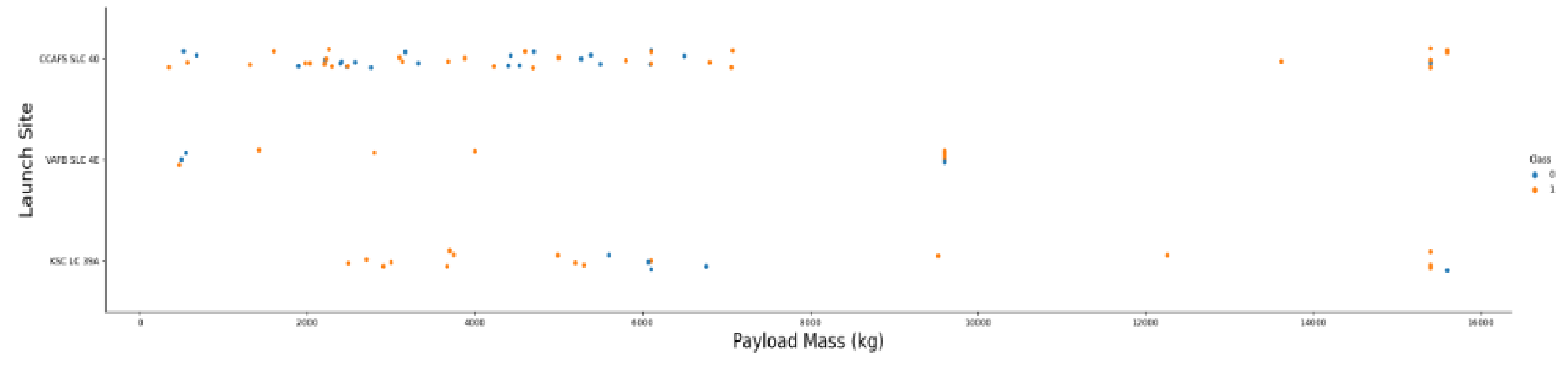
Flight Number versus Launch Site



Explanation:

- The earliest flights tended to fail while the latest flights tended to succeed.
- The CCAFS SLC 40 launch comprises about half of the launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- From the plot, we can infer that new launches have a higher success rate.

Payload versus Launch Site



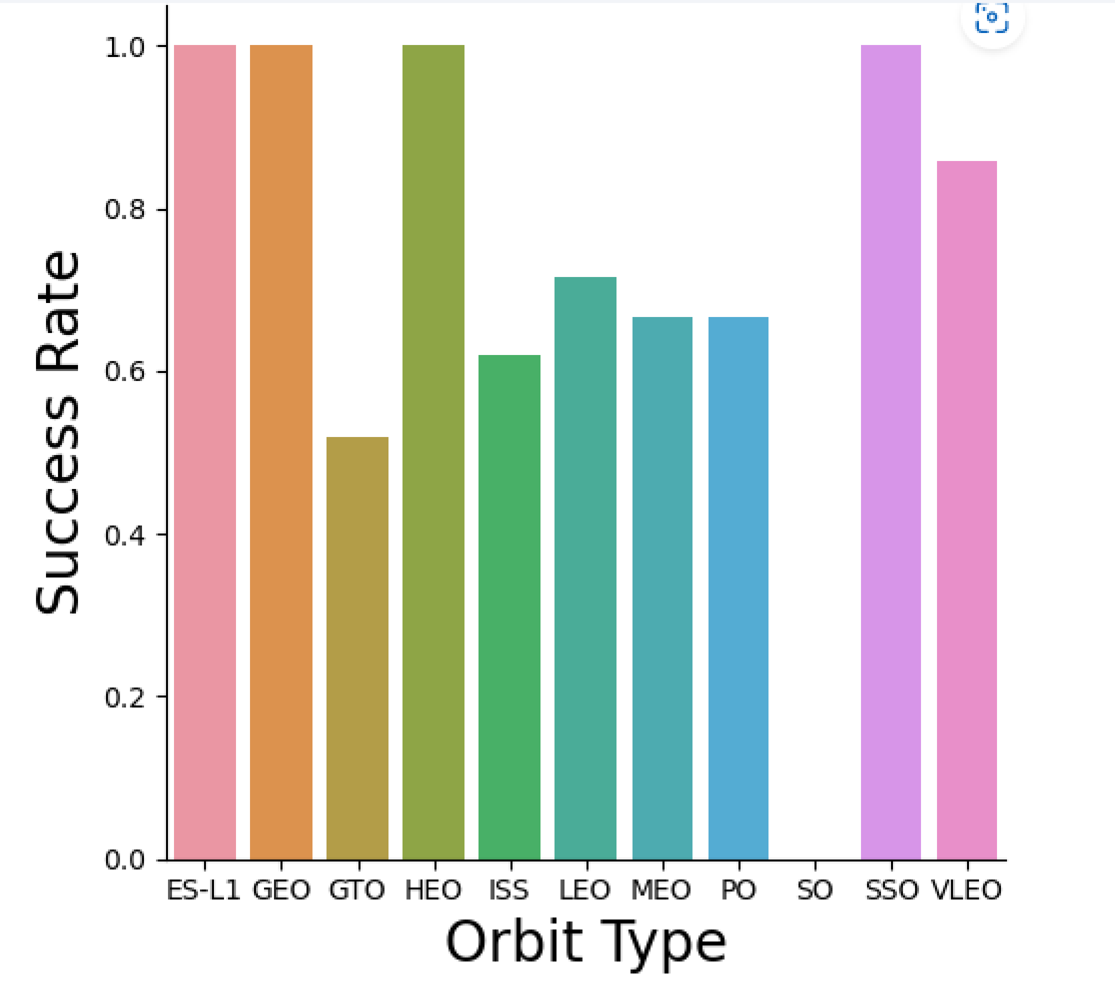
Explanation:

- For every launch site, the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.

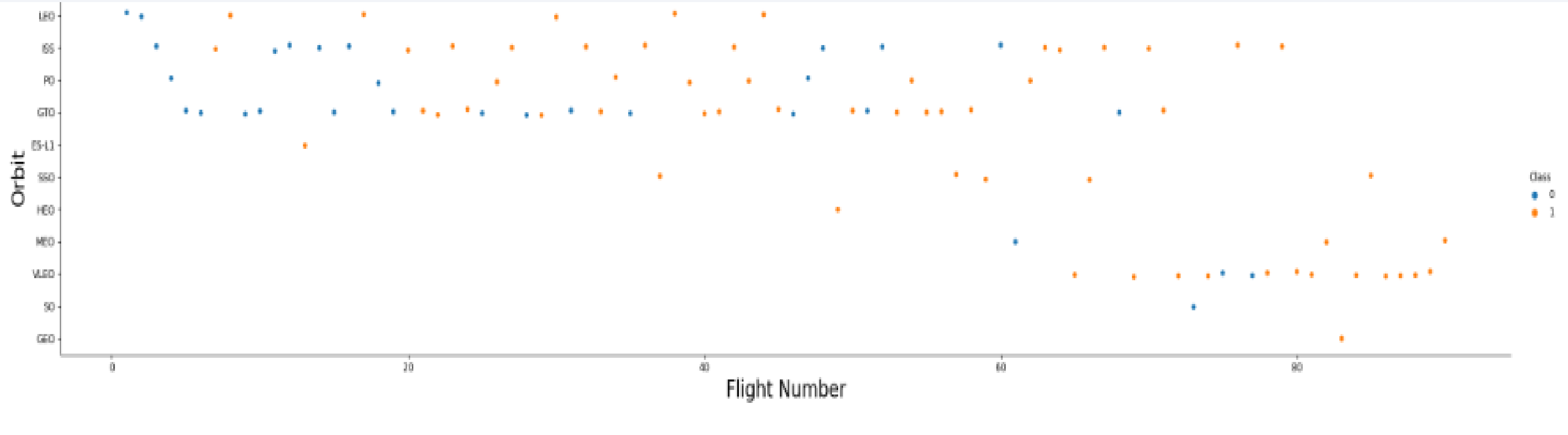
Success Rate versus Orbit Type

Explanation:

- Orbits with 100% success rate: ES-LI, GEO, HEO, SSO
- Orbits with 0% success rate: SO
- Orbits with a success rate between 50% and 85%: GTO, ISS, LEO, MEO, PO

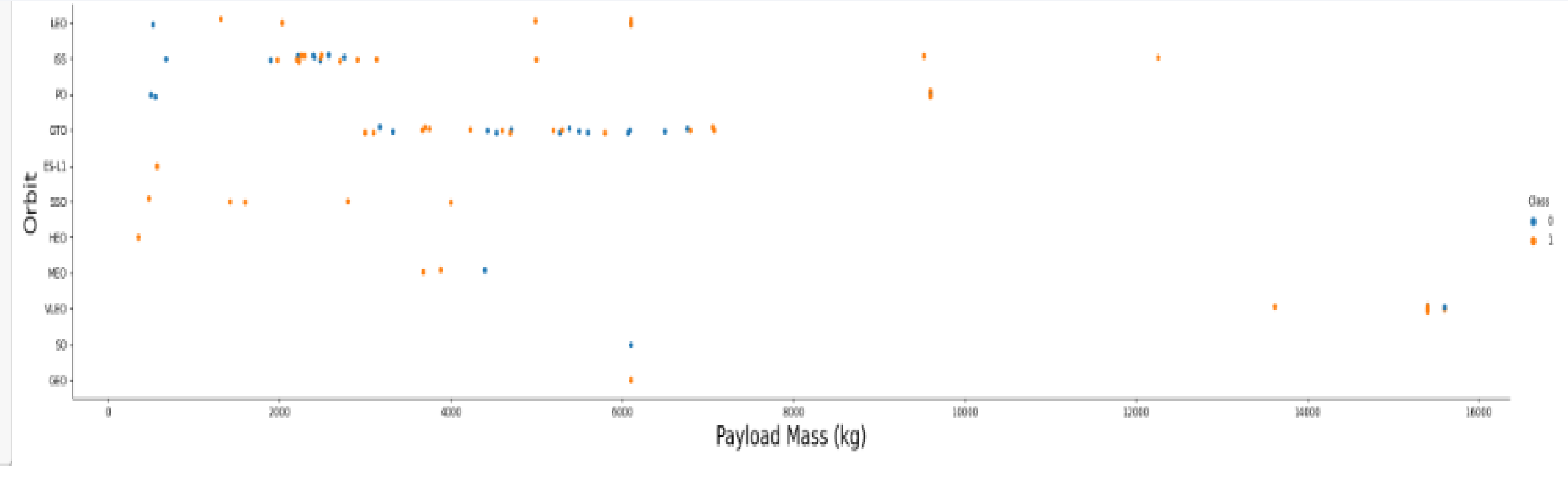


Flight Number versus Orbit Type



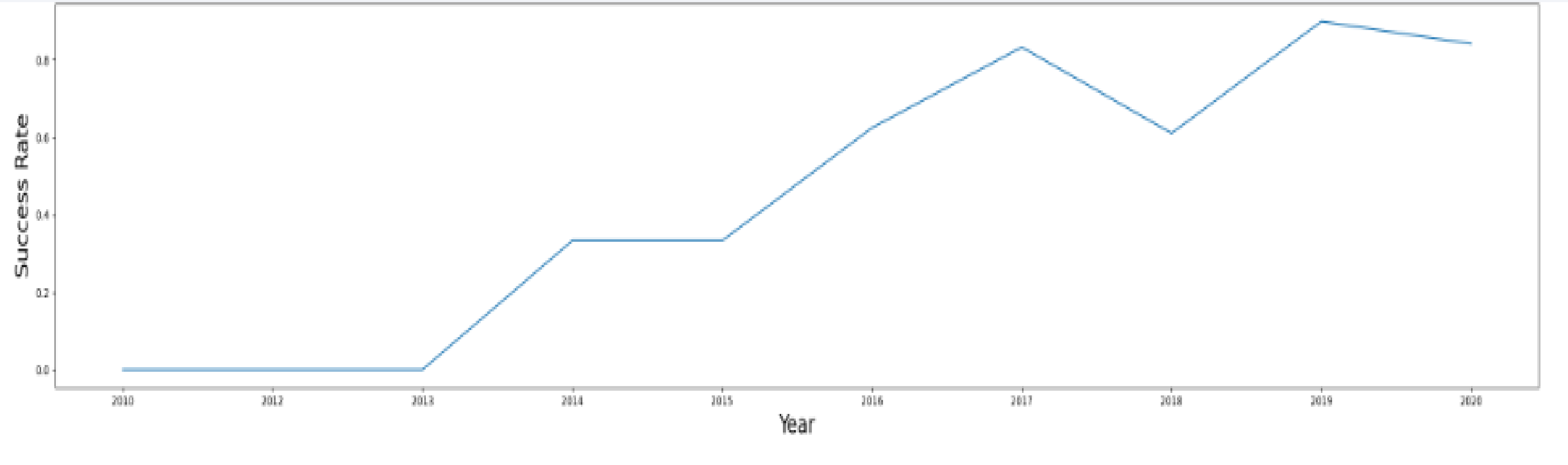
Explanation: In the LEO orbit, success appears to be related to the number of flights. However, this relationship disappears when in GTO orbit.

Payload Mass versus Orbit Type



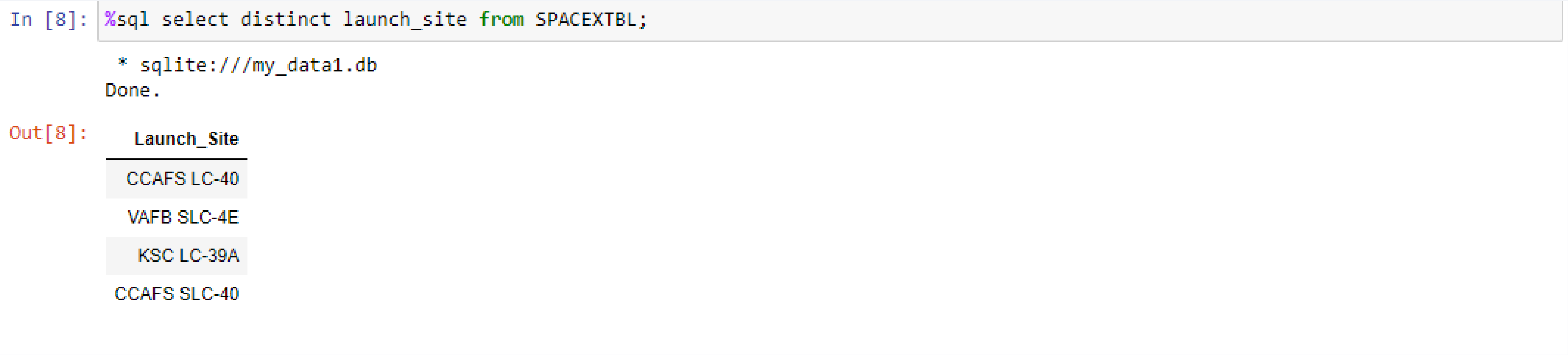
Explanation: Heavy payloads have a negative influence on GTO orbits and positive influences on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend



Explanation: The success rate increased from 2013 to 2020.

All Launch Site Names



```
In [8]: %sql select distinct launch_site from SPACEXTBL;
```

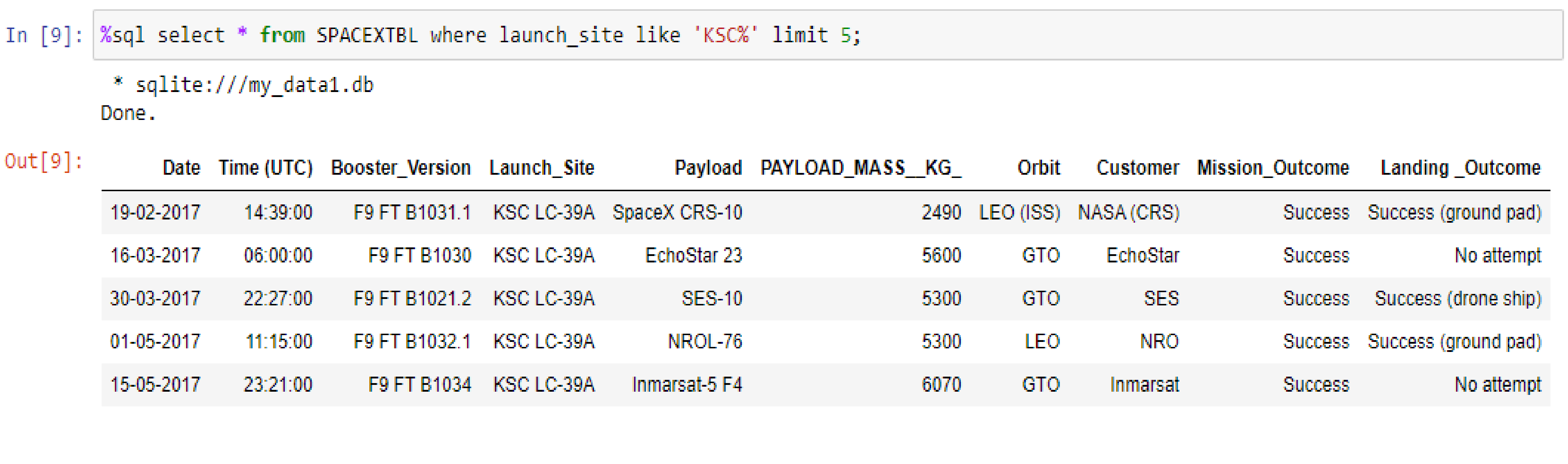
```
 * sqlite:///my_data1.db
Done.
```

Out[8]:

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

Explanation: The names of the unique launch sites in the space mission are displayed above.

Launch Site Names Begin with 'KSC'

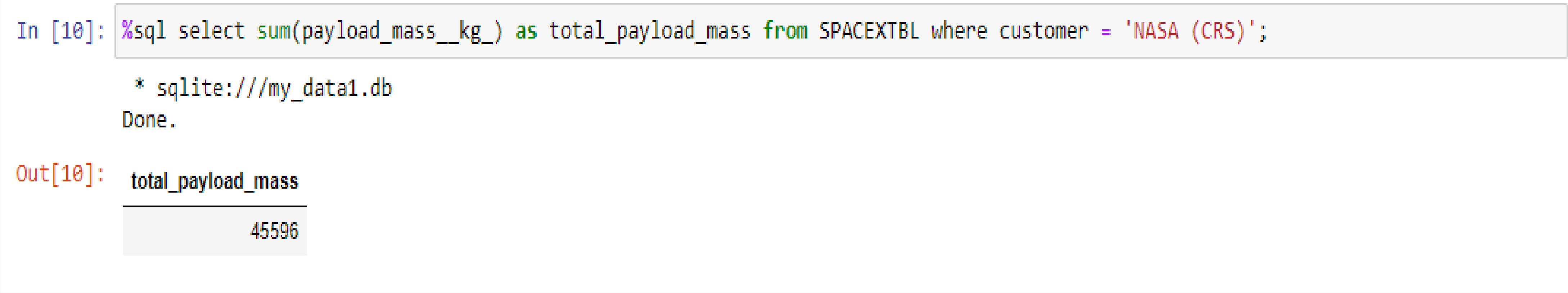


```
In [9]: %sql select * from SPACEXTBL where launch_site like 'KSC%' limit 5;

         * sqlite:///my_data1.db
        Done.

Out[9]:
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
19-02-2017	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
16-03-2017	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
30-03-2017	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
01-05-2017	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
15-05-2017	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Explanation: Displaying the 5 records where the launch site being the string ‘KSC’.

Total Payload Mass

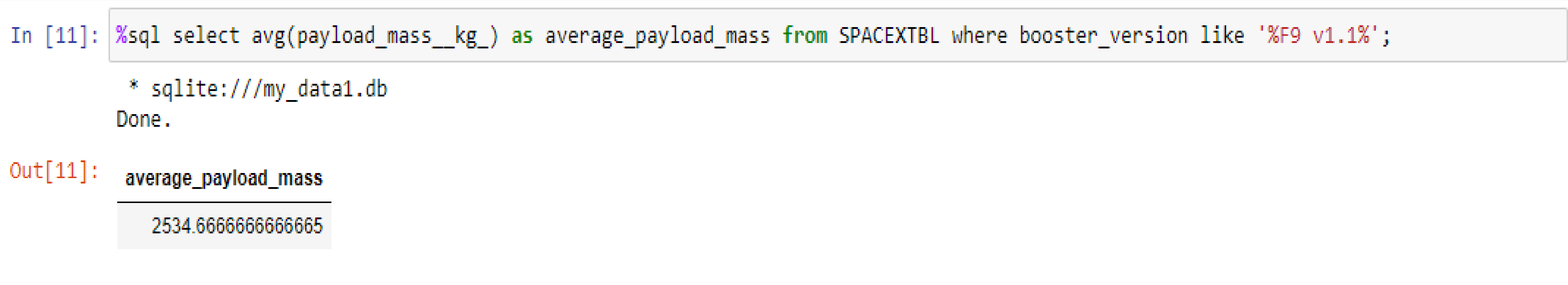


```
In [10]: %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)';

          * sqlite:///my_data1.db
         Done.

Out[10]:
```

total_payload_mass
45596

Explanation: The total payload mass carried by the boosters launched by NASA is 45,596 kg and is displayed above.

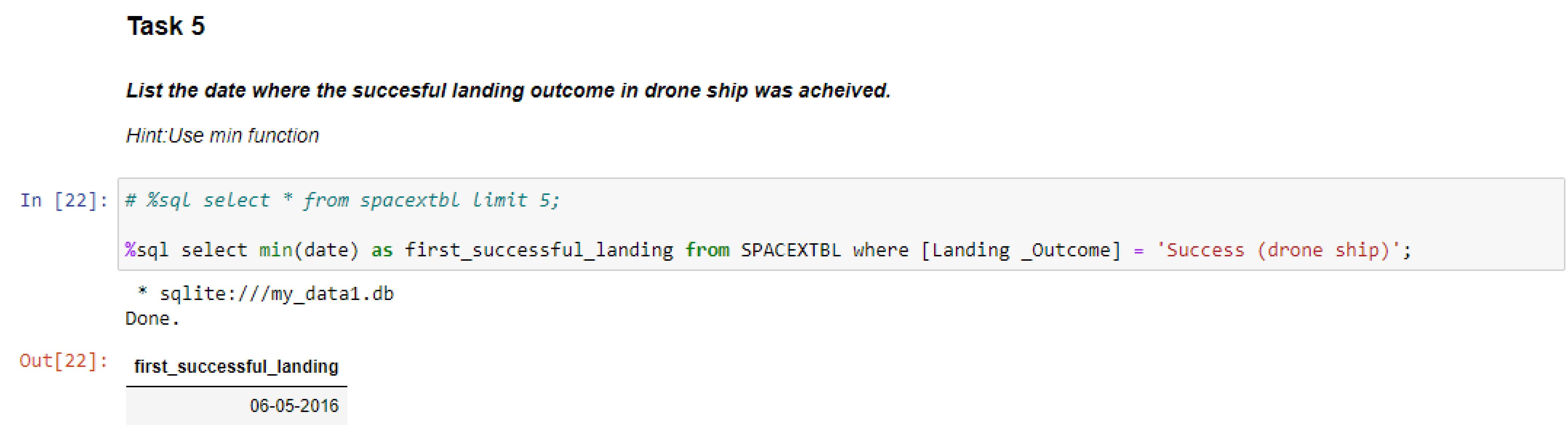
Average Payload Mass by F9 v1.1



```
In [11]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%';

          * sqlite:///my_data1.db
         Done.

Out[11]:  average_payload_mass
          2534.6666666666665
```

Explanation: The average payload mass carried by booster version F9 v1.1 is 2,534.67 kg and is displayed above.

First Successful Drone Ship Landing Date



Task 5

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

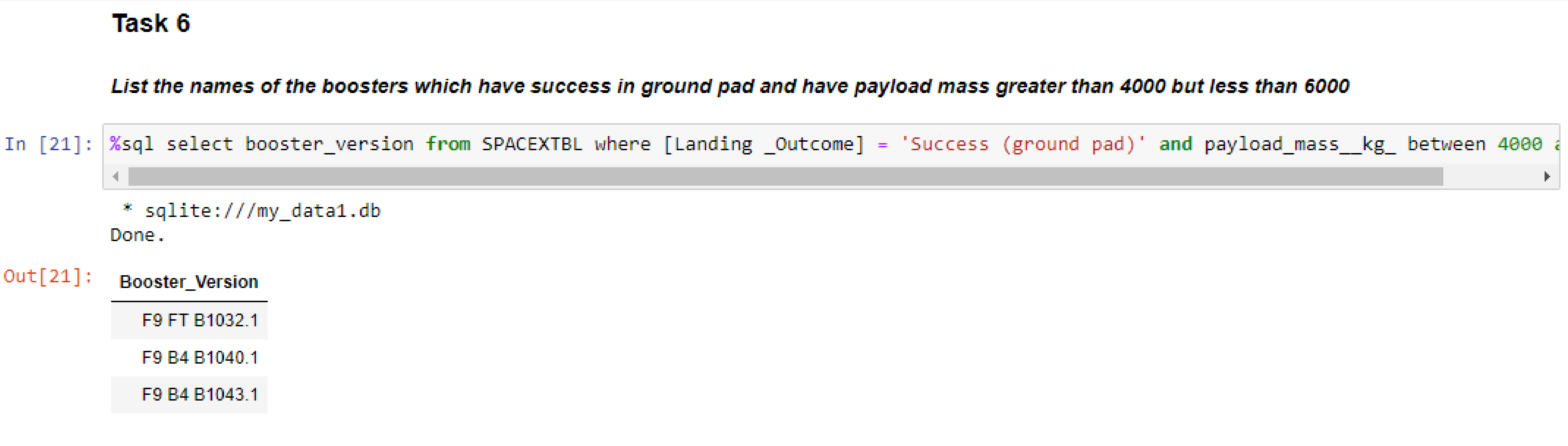
```
In [22]: # %sql select * from spacextbl limit 5;

         %sql select min(date) as first_successful_landing from SPACEXTBL where [Landing _Outcome] = 'Success (drone ship)';

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

Out[22]:

first_successful_landing
06-05-2016

Explanation: The date of the first successful landing outcome in the drone ship is displayed above. [3]

Successful Ground Pad Landings with Payload between 4000 kg and 6000 kg



Task 6

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

In [21]:
```
%sql select booster_version from SPACEXTBL where [Landing _Outcome] = 'Success (ground pad)' and payload_mass__kg_ between 4000 a
```

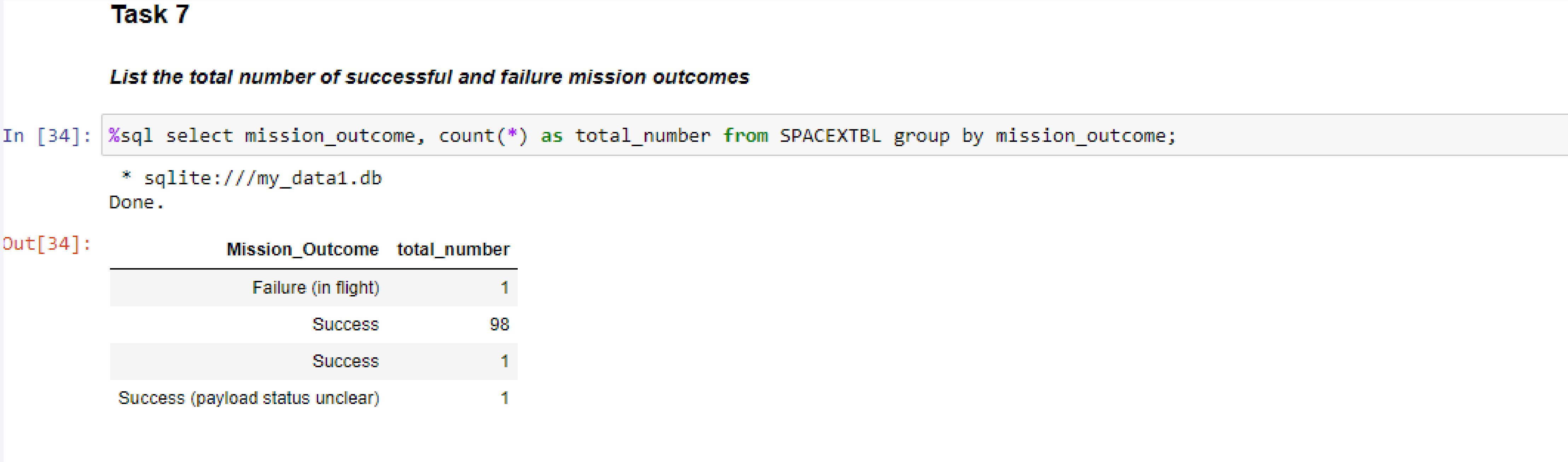
```
 * sqlite:///my_data1.db
Done.
```

Out[21]:

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Explanation: The names of the boosters with successful ground pad landings and have a payload mass greater than 4000 kg but less than 6000 kg are displayed above. [4]

Total Number of Successful and Failure Mission Outcomes



Task 7

List the total number of successful and failure mission outcomes

```
In [34]: %sql select mission_outcome, count(*) as total_number from SPACEXTBL group by mission_outcome;
```

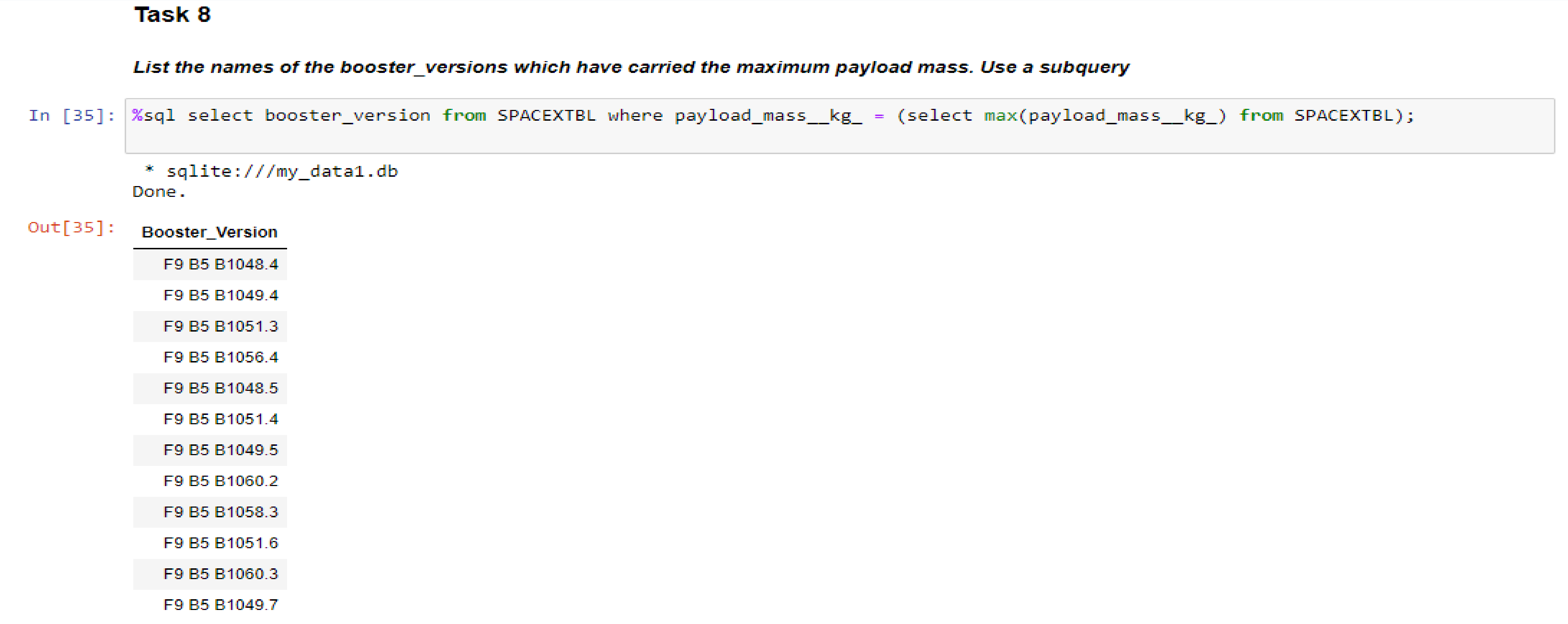
```
 * sqlite:///my_data1.db
Done.
```

Out[34]:

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

Explanation: The total number of successful and failure mission outcomes are listed above.

Boosters Carried Maximum Payload



Task 8

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

```
In [35]: %sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXTBL);
```

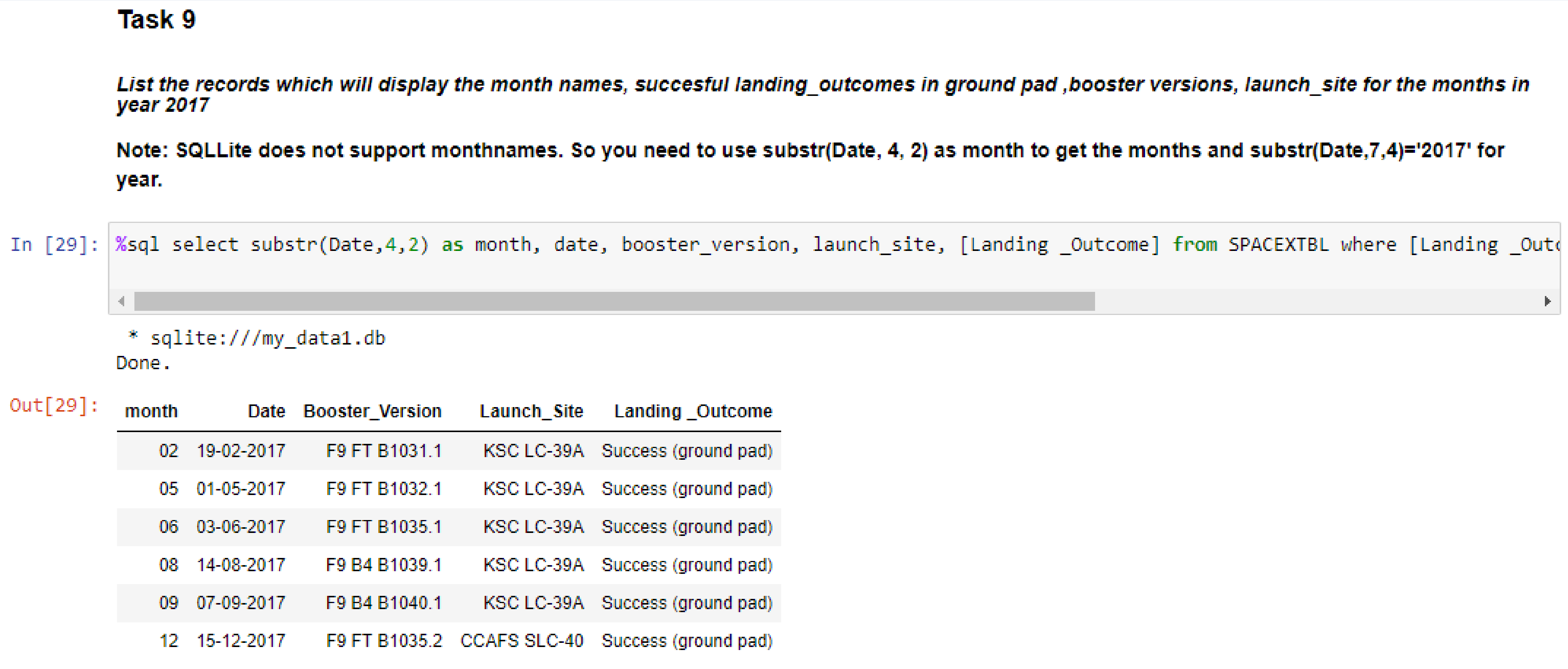
```
 * sqlite:///my_data1.db
Done.
```

Out[35]:

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

Explanation: The names of booster versions which have carried the maximum payload mass are displayed above.

2017 Launch Records



Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2017' for year.

In [29]:

```
%sql select substr(Date,4,2) as month, date, booster_version, launch_site, [Landing _Outcome] from SPACEXTBL where [Landing _Outc
```

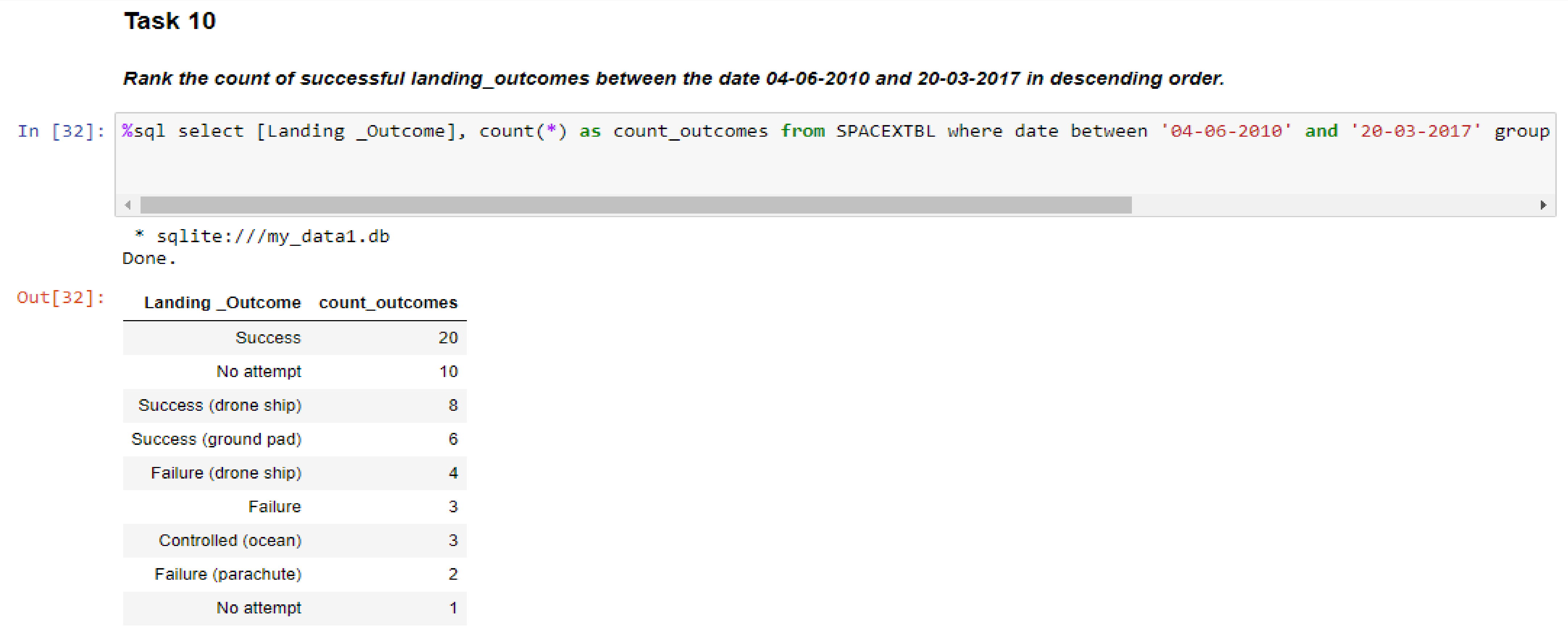
```
 * sqlite:///my_data1.db
Done.
```

Out[29]:

month	Date	Booster_Version	Launch_Site	Landing _Outcome
02	19-02-2017	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
05	01-05-2017	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
06	03-06-2017	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
08	14-08-2017	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
09	07-09-2017	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
12	15-12-2017	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

Explanation: The successful ground pad landing outcomes, their booster versions, and launch site names for the month in year 2017 are listed above. [5]

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



Task 10

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

```
In [32]: %sql select [Landing _Outcome], count(*) as count_outcomes from SPACEXTBL where date between '04-06-2010' and '20-03-2017' group
```

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

Out[32]:

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

Explanation: Ranking the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order.

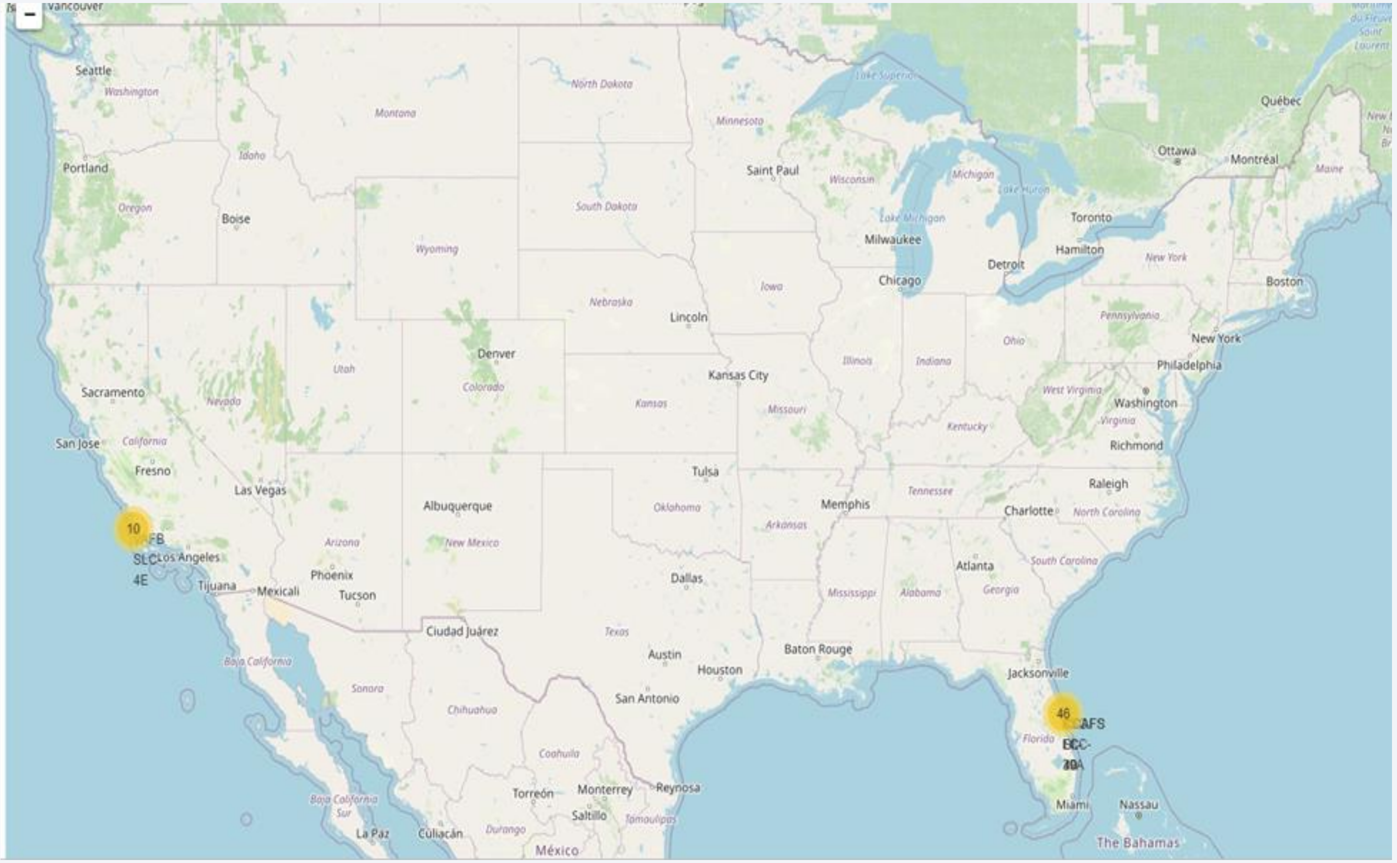
Section 3

Launch Sites Proximities Analysis

Launch Site Location Markers on a US Map

Explanation:

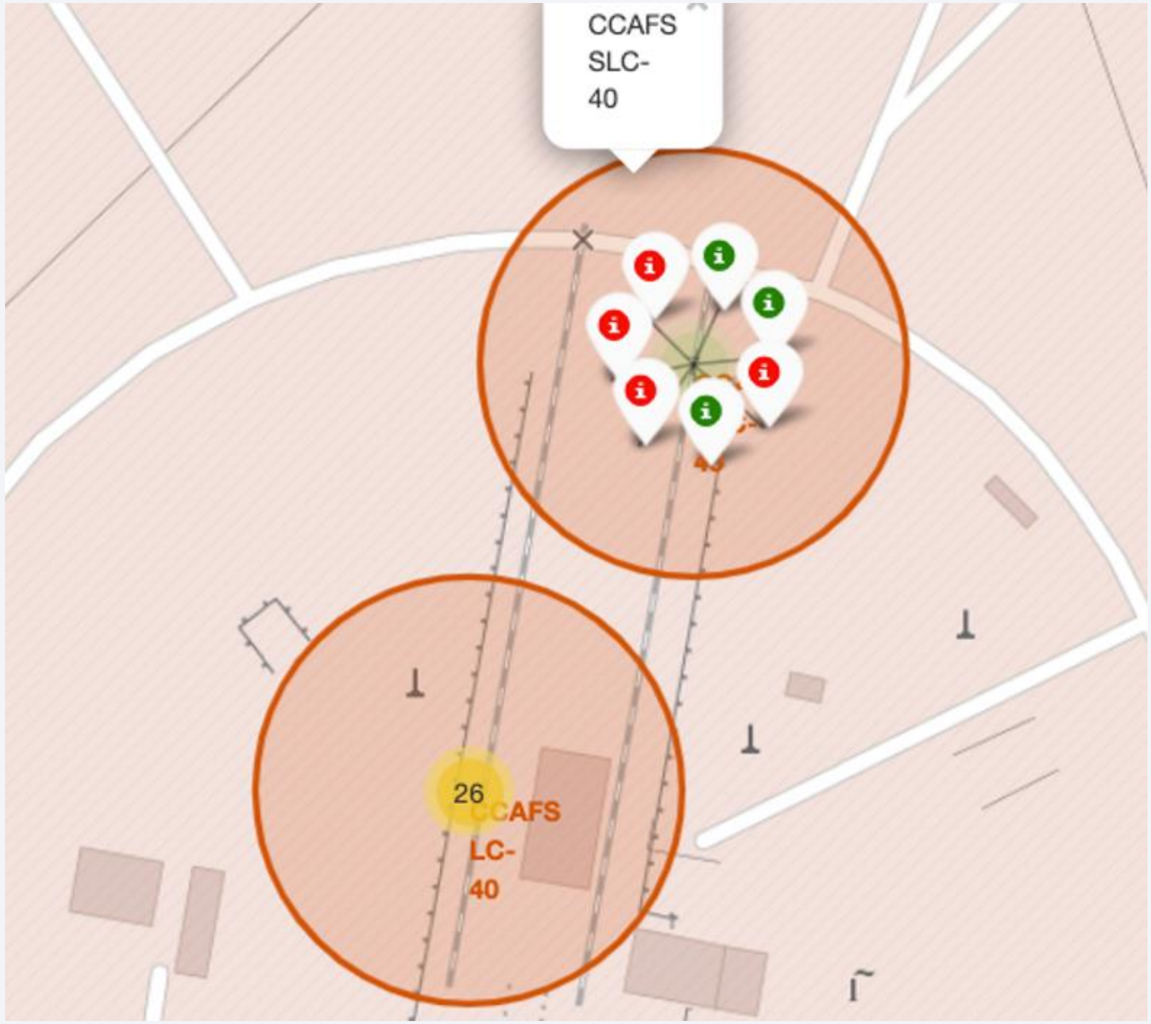
- Most of the launch sites are close to the equator. This is because the Earth is rotating faster at the equator (approximately 1670 km/hour) than at other latitudes. This high rotational velocity will help a rocket launched near the equator achieve and stay in orbit.
- All launch sites are near the coast to minimize the risk of debris damaging people or property in the event of a mission failure.



Color-Labeled Launch Outcomes on the Map

Explanation:

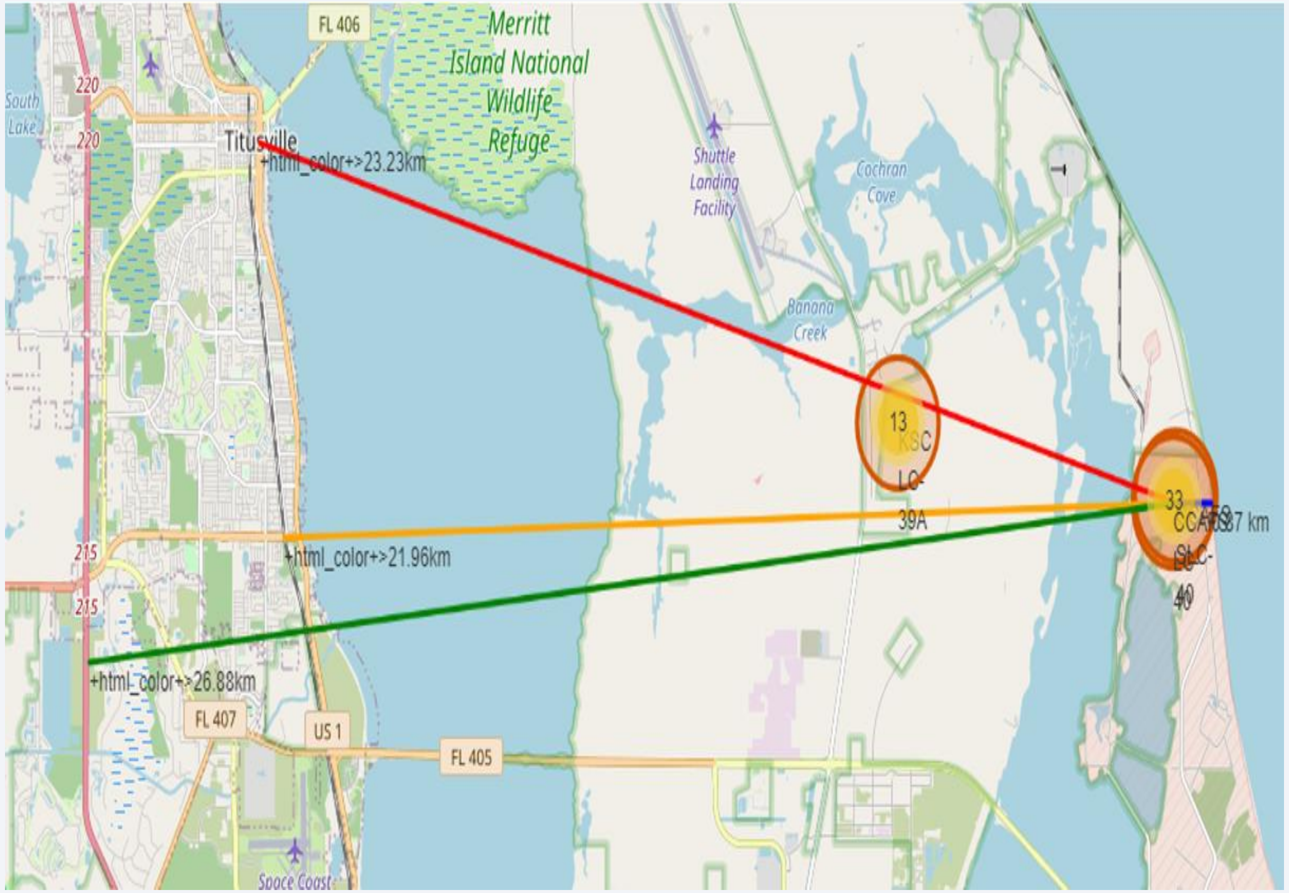
- Using the color-labeled markers, we can identify which launch sites have relatively high success rates, with green marker indicating successful launch and red marker indicating a failed launch.
- Launch site CCAFS SLC-40 has a relatively low success rate (3/7 or 42.9%).

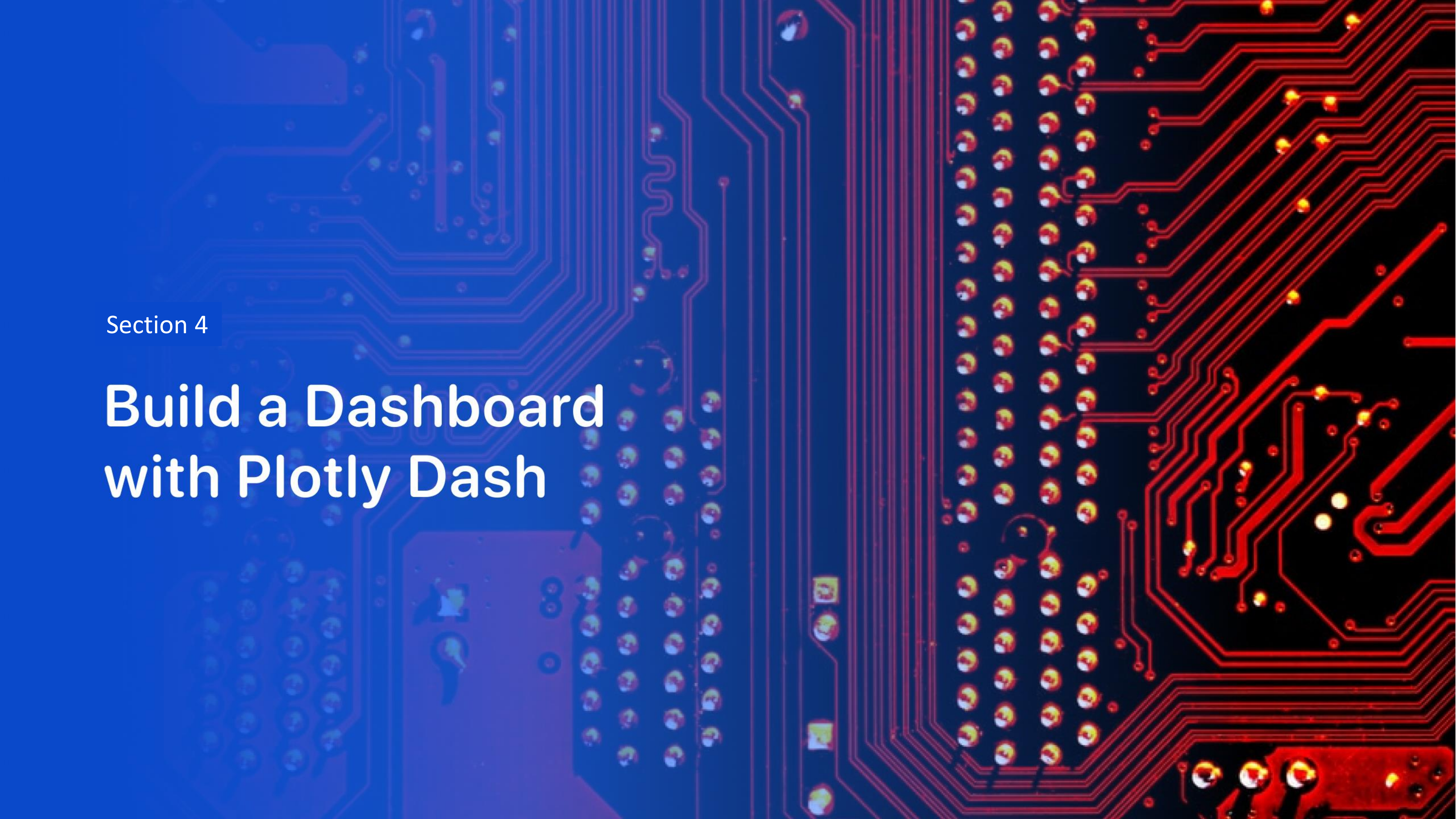


Distance from CCAFS SLC-40 to Proximities

Explanation:

- From visual analysis of CCAFS SLC-40, we see that it is:
  - 21.96 km from the nearest railway
  - 26.88 km from the nearest highway
  - 0.866 km from the nearest coastline
  - 23.23 from the nearest city
- A failed rocket can cover distances such as 15 to 20 kilometers in a few seconds, making them potentially dangerous in densely populated areas.

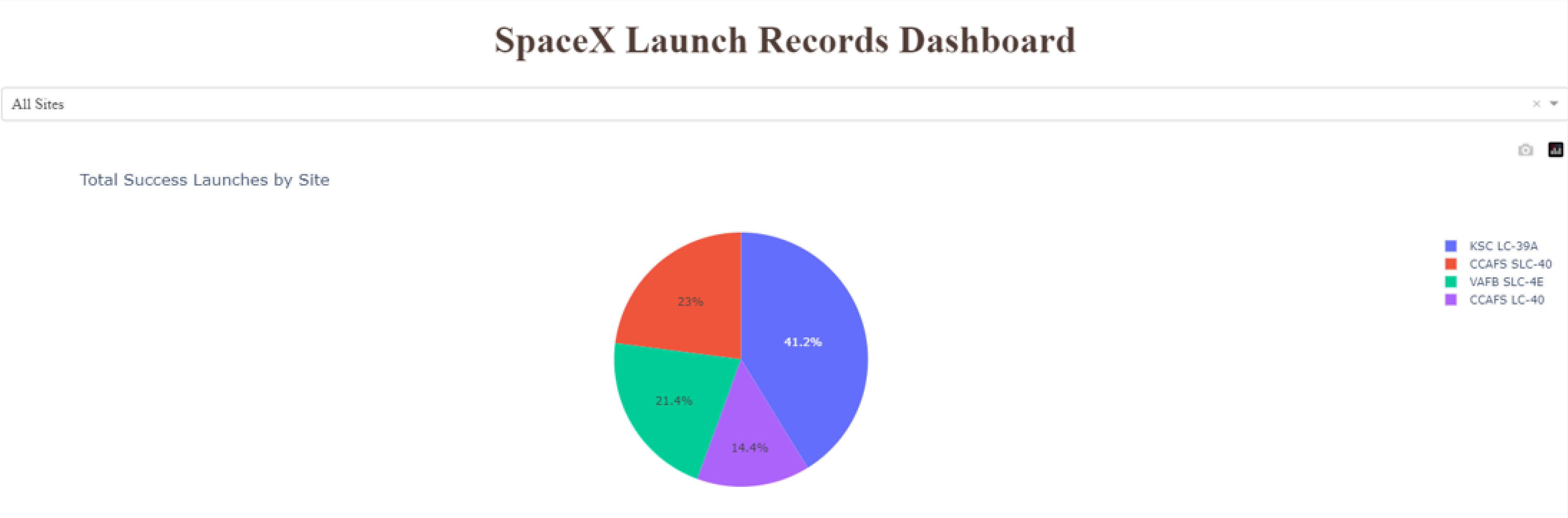




Section 4

Build a Dashboard with Plotly Dash

Launch Success Count for All Sites



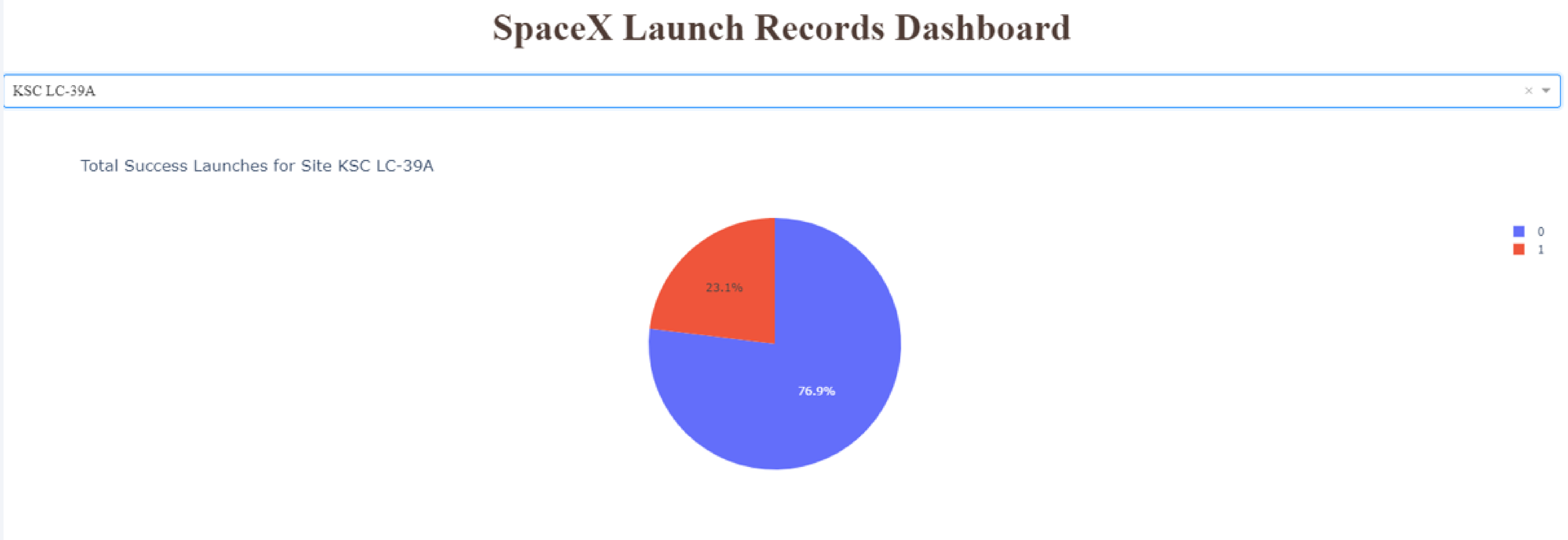
SpaceX Launch Records Dashboard

All Sites

Total Success Launches by Site

Explanation: The pie chart shows that KSC LC-39A has the most successful launches among the launch sites.

Launch Site with the Highest Launch Success Ratio



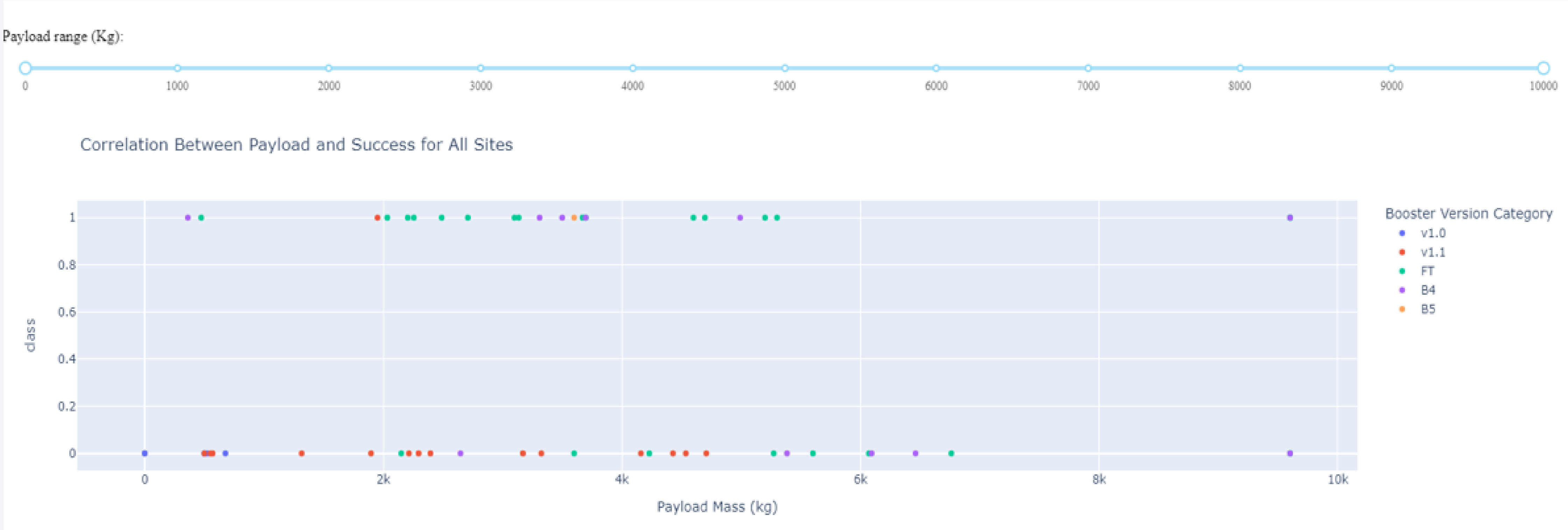
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site KSC LC-39A

Explanation: KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and 3 failed launches.

Payload Mass versus Launch Outcome for All Sites



Explanation: The chart shows that payloads between 2000 kg and 5000 kg have the highest success rate.

Section 5

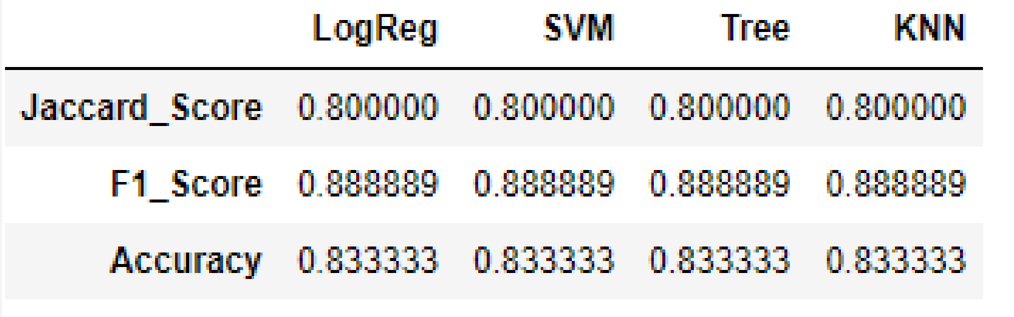
Predictive Analysis (Classification)

Classification Accuracy

Explanation:

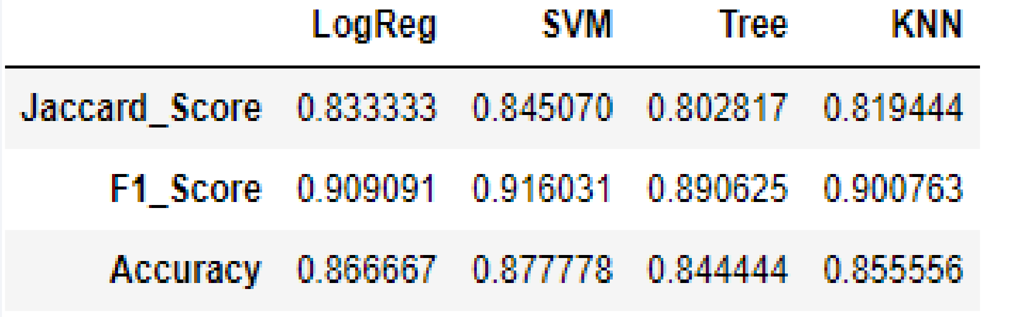
- Based on the scores of the test set, we cannot identify which model performed best, as all four models had the same scores and accuracy. This may be due to the small dataset used.
- When using the entire dataset, the best model is the SVM, which had highest scores and the highest accuracy.

Scores and Accuracy of the Test Set



	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333

Scores and Accuracy of the Entire Data Set

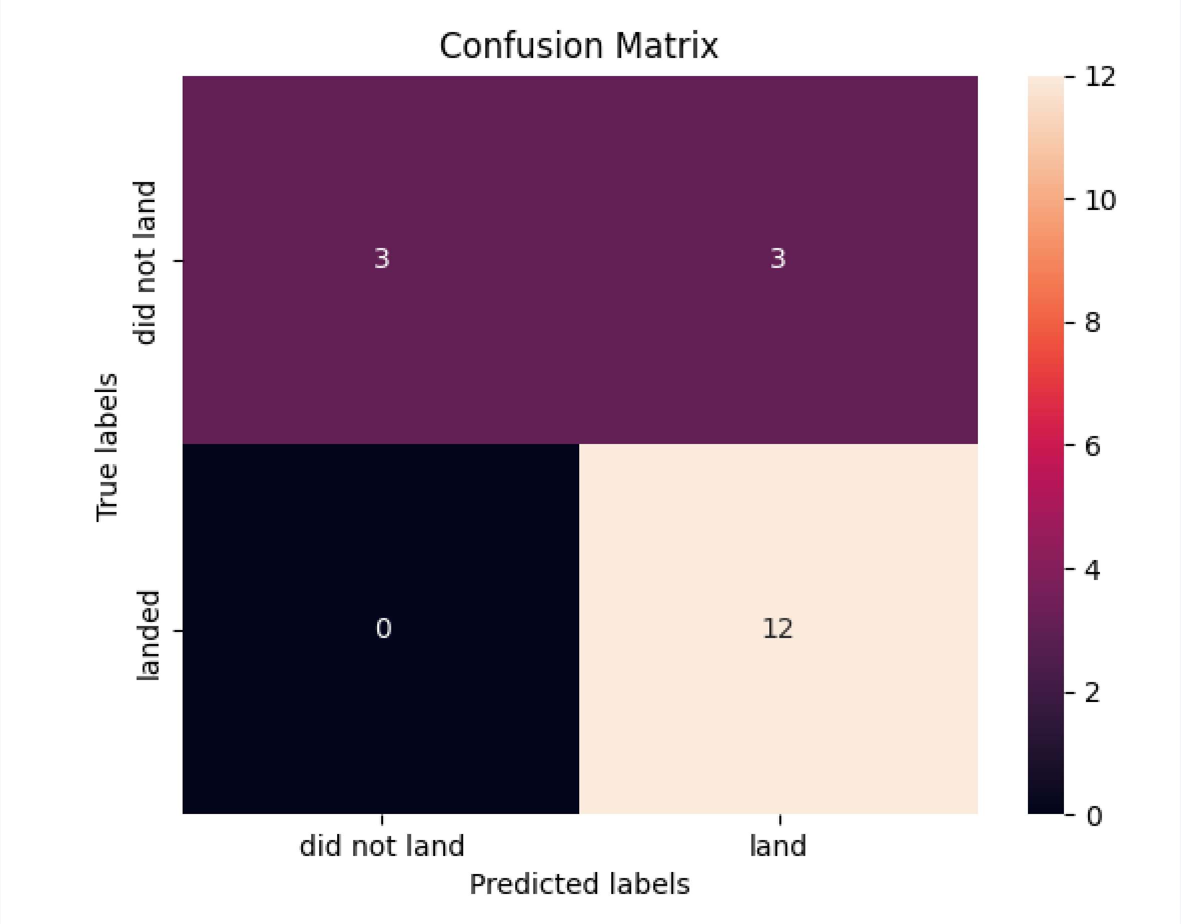


	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.802817	0.819444
F1_Score	0.909091	0.916031	0.890625	0.900763
Accuracy	0.866667	0.877778	0.844444	0.855556

Confusion Matrix

Explanation:

- As shown in the confusion matrix of the SVM, the most accurate model, distinctions can be made among the different outcomes. However, this model has some difficulty with false positives.



Conclusion

Our initial research gave us some results to apply to future research:

- SVM is the best predictive model for the dataset.
- Most of the launch sites are in proximity to the equator and all of the launch sites are close to the coast.
- The success rate of launches increases over time.
- KSC LC-39A has the highest success rate among the launch sites.
- Orbits ES-L1, GEO, HEO, and SSO have a 100% success rate.

For future research, we can build upon our initial results by employing a larger data set and apply the SVM model and other success factors to those data. We can then determine if the same observations that we made in this initial study are generalizable to a larger scale data set. Also, we can see if our results can be replicated by other data scientists, which would aid in our research into booster rocket technology and their cost effectiveness.

Appendix

Endnotes:

[1] I linked the pdf version of the Interactive Map with Folium Notebook because more of the Notebook could be viewed in the pdf version (due to GitHub's limitations in supporting Folium maps). The link to actual Interactive Map with Folium is here: GitHub link for Interactive Map with Folium Notebook.

[2] Because the lab for Plotly Dash was not conducted in Jupyter Lab, I could not save the notebook. Therefore, I made the link to the Python code used for the lab.

[3] On page 28, I show the first successful drone ship landing since it was required to do during the lab. The template had "First Successful Ground Landing Date," which I did not follow.

[4] On page 29, I show the successful ground pad landings with payload between 4000 kg and 6000 kg since it was required to do during the lab. The template had "Successful Drone Ship Landing with Payload between 4000 and 6000."

[5] On page 32, I queried 2017 launch records since that was what was required in the lab. The template had "2015 Launch Records" in the heading.

Thank you!