IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sergio R.

13/07/2023

Executive Summary

- Summary of methodologies
  - Web Scrapping for data acquisition using the SpaceX and Wikipedia API
  - Data Wrangling, Data Visualizations and SQL exploration
  - Dashboard for interactive Data Analysis
  - Prediction and Classification generation using Machine Learning
- Summary of all results
  - Gathered Data from the web
  - Understanding through EDA
  - Insight gained thru Machine learning methods

Outline – Table of Contents

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

Introduction

- Project background and context
  - Space Y that would like to compete with SpaceX founded by Billionaire industrialist Allon Musk.
  - The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets.

- Problems you want to find answers
  - Determine price of each launch
  - Gather information about Space X and Create dashboards
  - Predict if Space X will reuse the first Stage

Section 1

Methodology

Methodology

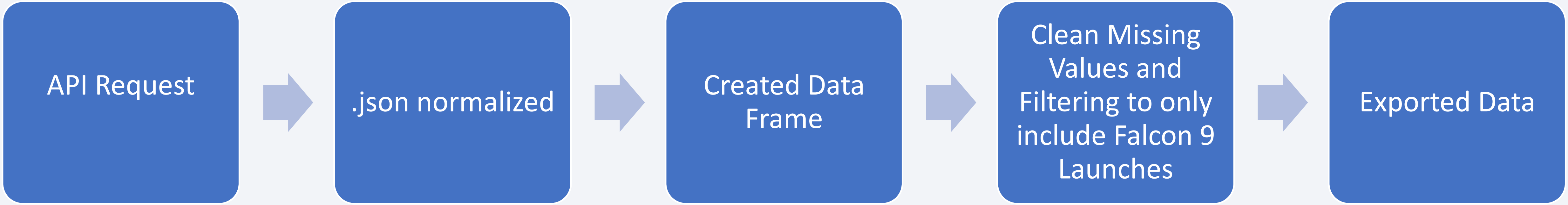
Executive Summary

- Data collection methodology:
  - Data was retrieved from using APIs
    - https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
    - https://api.spacexdata.com/v4/launches/past
- Perform data wrangling
  - Parsed Data, handled missing Values, used One hot encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Normalized data, created training sets, Evaluating using different models and found best

Data Collection

- The data collection process involved utilizing a combination of API requests from SpaceX's REST API and performing web scraping on a table within SpaceX's Wikipedia entry. We employed both of these methods to gather comprehensive information about the launches for a more thorough analysis. The following data columns were acquired using the SpaceX REST API: FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, and Latitude. On the other hand, the following data columns were obtained through web scraping from Wikipedia: Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, and Time.

Data Collection – SpaceX API



API Request

.json normalized

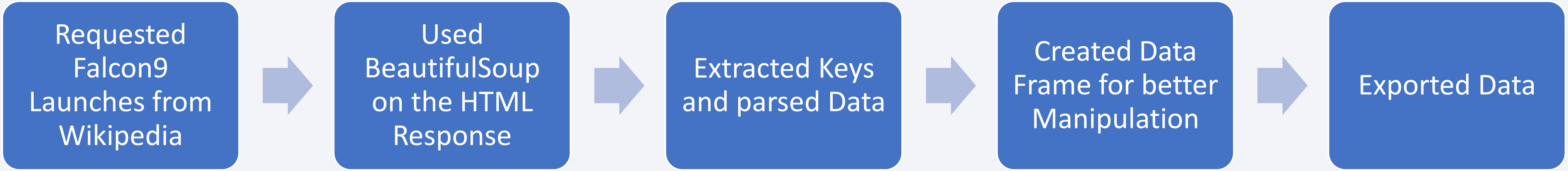
Created Data Frame

Clean Missing Values and Filtering to only include Falcon 9 Launches

Exported Data

- GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/jupyter-labs-spacex-data-collection-api%20(2).ipynb

Data Collection - Scraping



Requested Falcon9 Launches from Wikipedia

Used BeautifulSoup on the HTML Response

Extracted Keys and parsed Data

Created Data Frame for better Manipulation

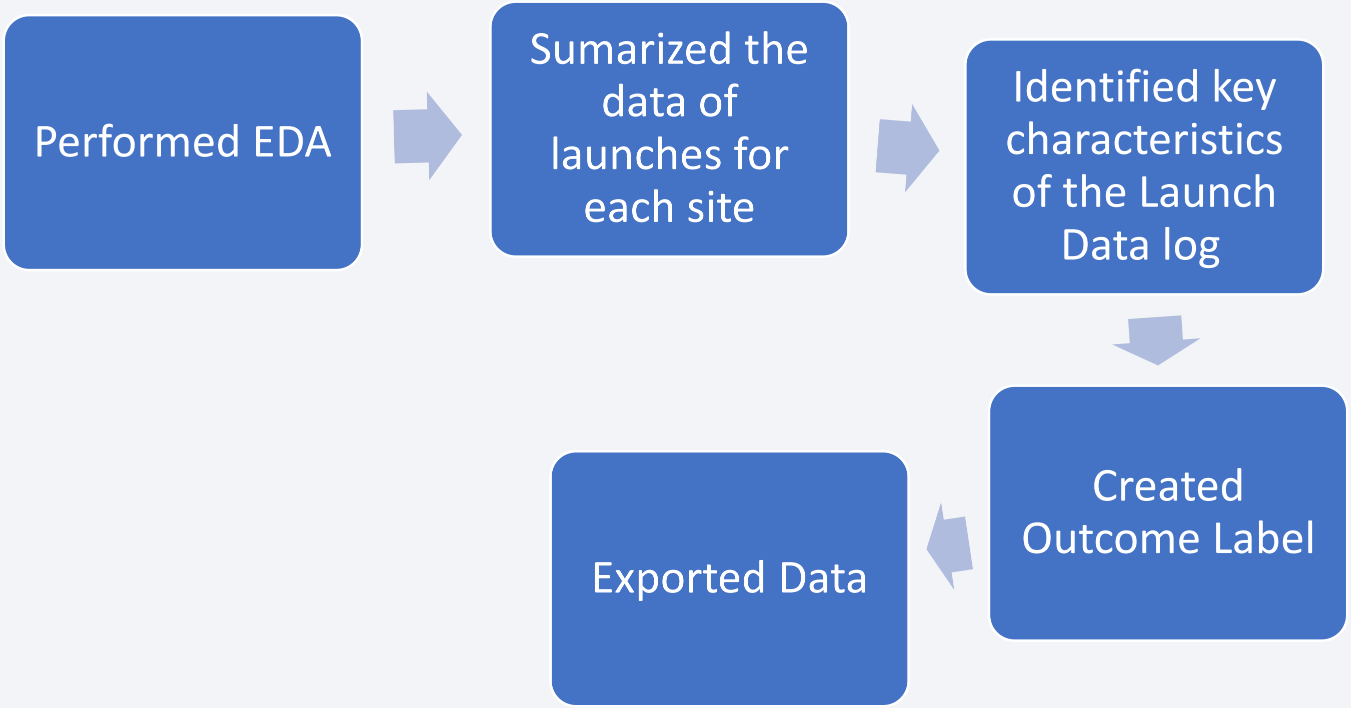
Exported Data

- GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/jupyter-labs-webscraping%20(1).ipynb

Data Wrangling

- The data set contains various instances of unsuccessful booster landings. These failures can occur due to accidents or other factors. For instance, "True Ocean" indicates a successful landing in a specific ocean region, while "False Ocean" means an unsuccessful landing in a specific ocean region. Similarly, "True RTLS" denotes a successful landing on a ground pad, while "False RTLS" signifies an unsuccessful landing on a ground pad. "True ASDS" represents a successful landing on a drone ship, whereas "False ASDS" indicates an unsuccessful landing on a drone ship.

- Transformed continuous Data into discreet Data


Performed EDA

Sumarized the data of launches for each site

Identified key characteristics of the Launch Data log

Created Outcome Label

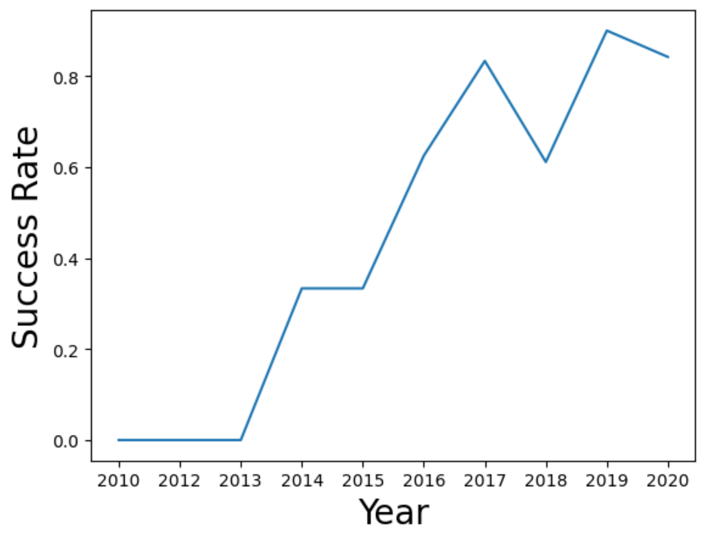
Exported Data

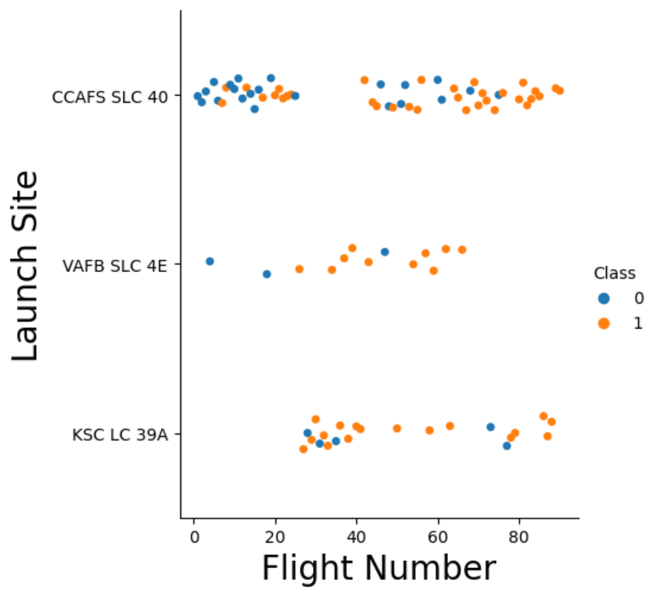
GitHub URL
https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb

EDA with Data Visualization

Data exploration involved the use of **scatterplots** and **barplots** to visualize the connections between different features. The scatterplots depicted the relationships between Flight Number and Payload Mass, Launch Site and Flight Number, Launch Site and Payload Mass, Orbit Type and Flight Number, Payload and Orbit. Meanwhile, the bar charts displayed Flight Number vs. Payload Mass, Flight Number vs. Launch Site, Payload Mass vs. Launch Site, Orbit Type vs. Success Rate, Flight Number vs. Orbit Type, Payload Mass vs. Orbit Type, and Success Rate Yearly Trend.

Scatterplots provided **insights** into variable relationships that could be utilized in machine learning models. On the other hand, bar charts facilitated comparisons between discrete categories to highlight the relationship between specific categories and measured values. Lastly**, line charts were employed to observe trends in data over time, particularly in time series data.**





- GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite%20(1).ipynb

EDA with SQL

- - Retrieving the names of the unique launch sites in the space mission
- - Obtaining the top 5 launch sites whose names begin with the string 'CCA'
- - Calculating the total payload mass carried by boosters launched by NASA (CRS)
- - Calculating the average payload mass carried by booster version F9 vl. 1
- - Finding the date of the first successful landing outcome on a ground pad
- - Retrieving the names of the boosters that successfully landed on a drone ship and had a payload mass between 4000 and 6000 kg
- - Determining the total number of successful and failure mission outcomes
- - Retrieving the names of the booster versions that carried the maximum payload mass
- - Identifying the failed landing outcomes on a drone ship, along with their booster versions and launch site names for the year 2015
- - Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the dates 2010-06-04 and 2017-03-20 in descending order

GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- In the folium map, several map objects were created and added for different purposes:
- Markers of all Launch Sites:
  - A Marker with a Circle, Popup Label, and Text Label of NASA Johnson Space Center was added, using its latitude and longitude coordinates as the start location.
  - Markers with Circles, Popup Labels, and Text Labels of all Launch Sites were added, using their latitude and longitude coordinates. This was done to visually represent their geographical locations and illustrate their proximity to the Equator and coasts.
- Colored Markers of the launch outcomes for each Launch Site:
  - Colored Markers indicating success (Green) and failed (Red) launches were added. A Marker Cluster was used to group them together. This addition helps to identify which launch sites have relatively high success rates based on the color distribution.
- Distances between a Launch Site and its proximities:
  - Colored Lines were added to show the distances between a specific Launch Site (e.g., KSC LC-39A) and its nearby features such as Railway, Highway, Coastline, and Closest City. These lines visually represent the distances and provide insights into the proximity of the launch site to various important landmarks.
- Overall, these map objects were added to enhance the visual representation of the launch sites and their associated information. They provide a comprehensive overview of the launch sites' locations, success rates, and distances to nearby features, allowing for better analysis and understanding of the data.

GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/lab_jupyter_launch_site_location.jupyterlite%20(1).ipynb

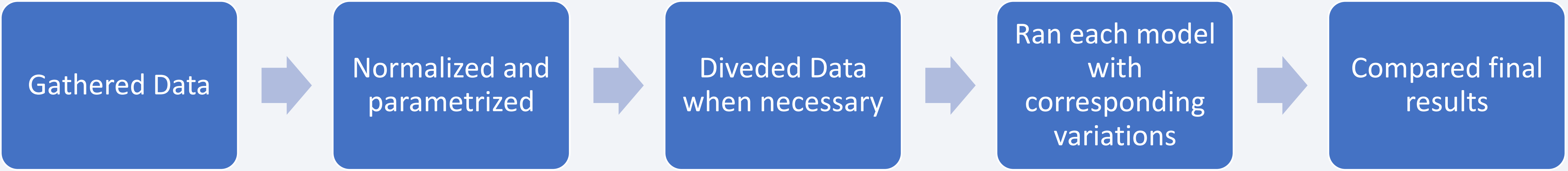
Build a Dashboard with Plotly Dash

- In the dashboard, the following plots, graphs, and interactions were added:
- 1. Launch Sites Dropdown List: A dropdown list was included to enable the selection of a launch site. This allows users to choose a specific launch site for analysis and comparison.
- 2. Pie Chart showing Success Launches (All Sites/Certain Site): A pie chart was included to display the total count of successful launches for all sites. Additionally, if a specific launch site is selected from the dropdown list, the pie chart shows the success versus failure counts for that particular site. This provides an overview of the success rates and allows users to compare the performance of different launch sites.
- 3. Slider of Payload Mass Range: A slider was added to enable the selection of a payload mass range. Users can adjust the slider to define a specific range of payload masses for analysis.
- 4. Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions: A scatter chart was included to visualize the relationship between payload mass and launch success rate. This chart specifically focuses on different booster versions. It allows users to examine the correlation between payload mass and the likelihood of a successful launch, providing insights into the performance of different booster versions.
- The plots and interactions were added to provide a comprehensive analysis of launch data. The dropdown list facilitates site-specific analysis, allowing users to compare the success rates of different launch sites. The pie chart offers a clear visualization of success and failure counts for both all sites and a specific launch site. The payload mass range slider enables users to narrow down the analysis based on specific payload requirements. Finally, the scatter chart offers insights into the relationship between payload mass and launch success, specifically focusing on the different booster versions. Overall, these plots and interactions provide a comprehensive and interactive dashboard to explore and analyze the launch data from various perspectives.

GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- In the evaluation of four classification models (logistic regression, support vector machine, decision tree, and k nearest neighbors), the initial analysis based on the test set did not provide conclusive evidence of the best-performing method. This uncertainty could be attributed to the small sample size of the test set, which consisted of only 18 samples. To address this limitation, all models were subsequently tested on the entire dataset. The results from the entire dataset revealed that the decision tree model not only achieved higher scores but also exhibited the highest accuracy among all the models.



Gathered Data

Normalized and parametrized

Diveded Data when necessary

Ran each model with corresponding variations

Compared final results

GitHub URL https://github.com/sseerrggiiooo/Final-Delivery-IBM-/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

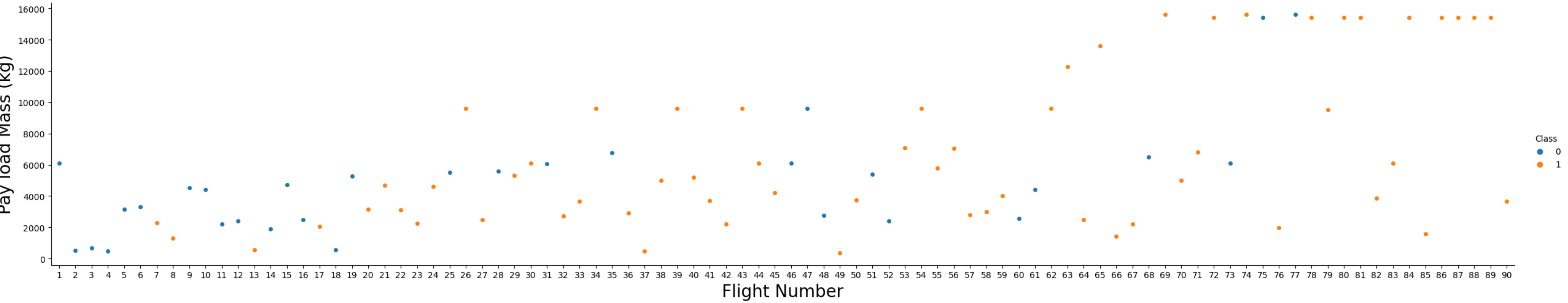
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

- Scores and Accuracy Test for ML models

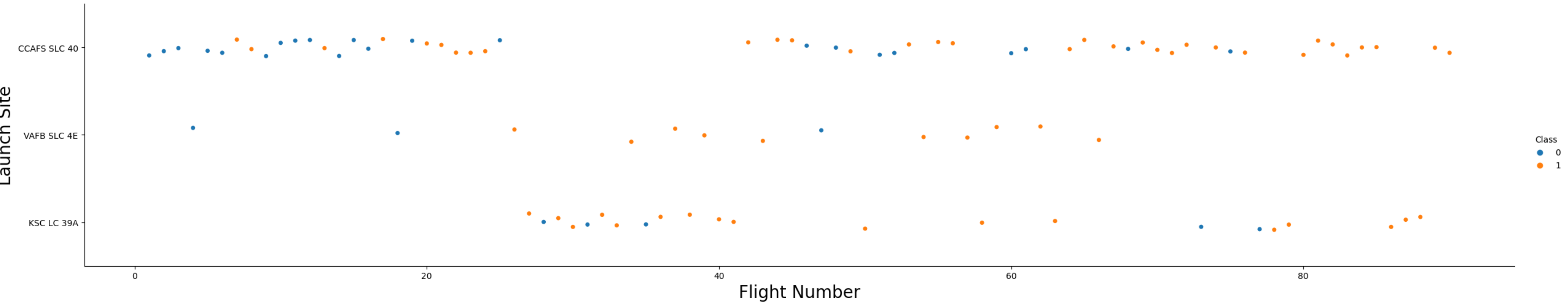
Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



scatter plot with explanations

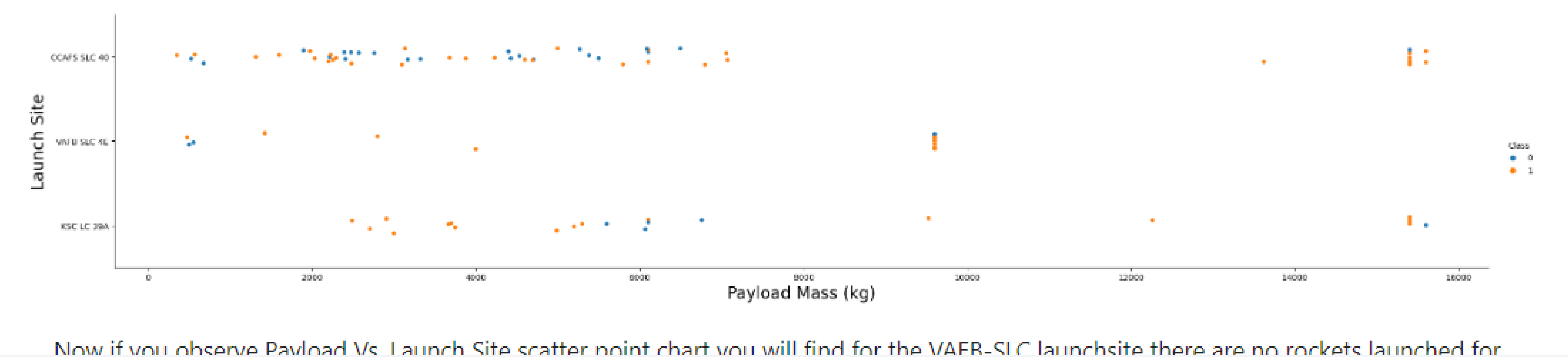


Flight Number vs. Launch Site

- The given information states that earlier flights were unsuccessful while the latest flights were successful. The launch site CCAFS SLC 40 had approximately half of all launches, but VAFB SLC 4E and KSC LC 39A had higher success rates. It can be assumed that each new launch has a higher chance of success. The plot confirms that the current best launch site is CCAFS SLC 40, where most recent launches have been successful. In second place is VAFB SLC 4E, followed by KSC LC 39A. Furthermore, the overall success rate has improved over time.

Payload vs. Launch Site

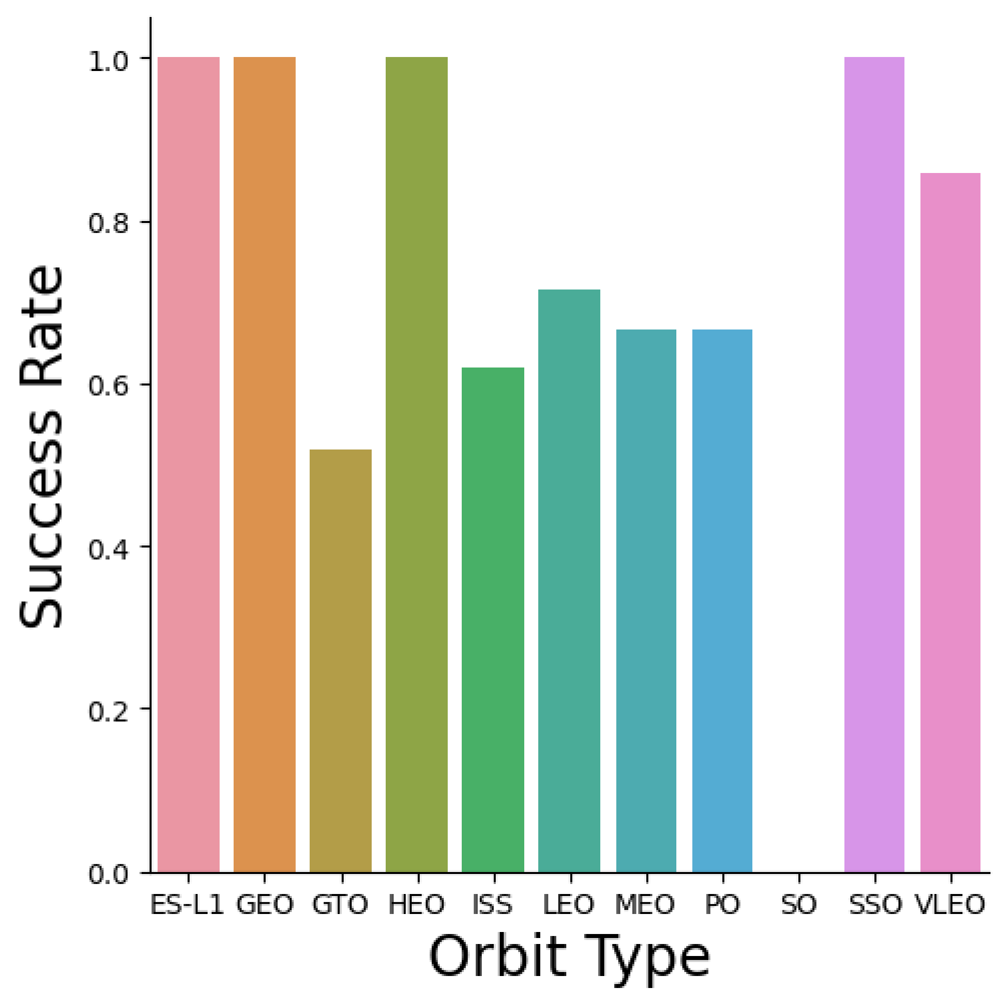
- Explanation
  The success rate of rocket launches is influenced by the payload mass. Launches with higher payload masses tend to have higher success rates. Specifically, launches with payload masses over 7000 kg have mostly been successful. Furthermore, KSC LC 39A has a 100% success rate for payload masses under 5500 kg. Payloads weighing over 9000 kg have excellent success rates, comparable to the weight of a school bus. Finally, payloads exceeding 12,000 kg appear to be feasible only at the CCAFS SLC 40 and KSC LC 39A launch sites.



Now if you observe Payload Vs Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for

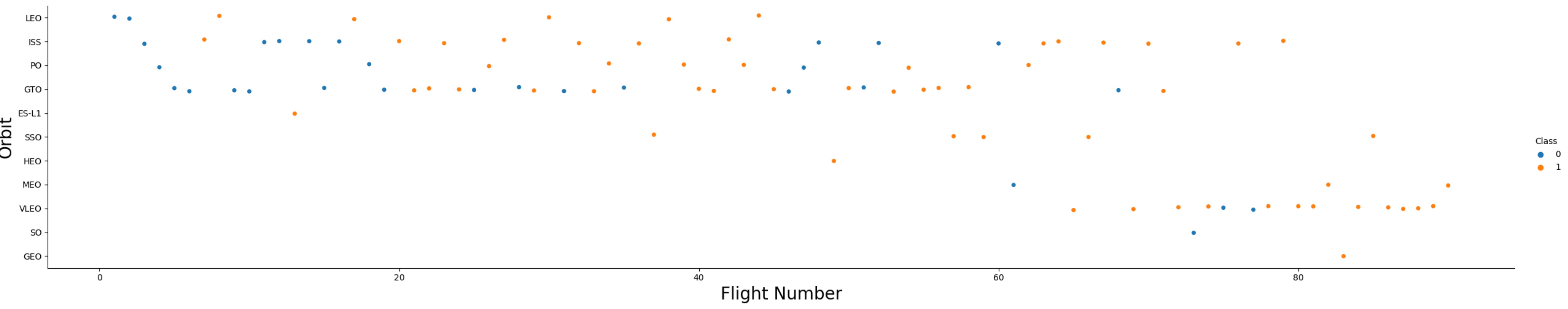
Success Rate vs. Orbit Type

- Explanation
- The success rates of different orbits can be categorized as follows: Orbits with a 100% success rate include ES-LI, CEO, HEO, and SSO. Orbits with a 0% success rate are not mentioned. Orbits with success rates between 50% and 85% include GTO, lss, LEO, MEO, and POR. The orbits with the highest success rates are ES-LI, GEO, HEO, followed by VLEO (above 80%) and LEO (above 70%).



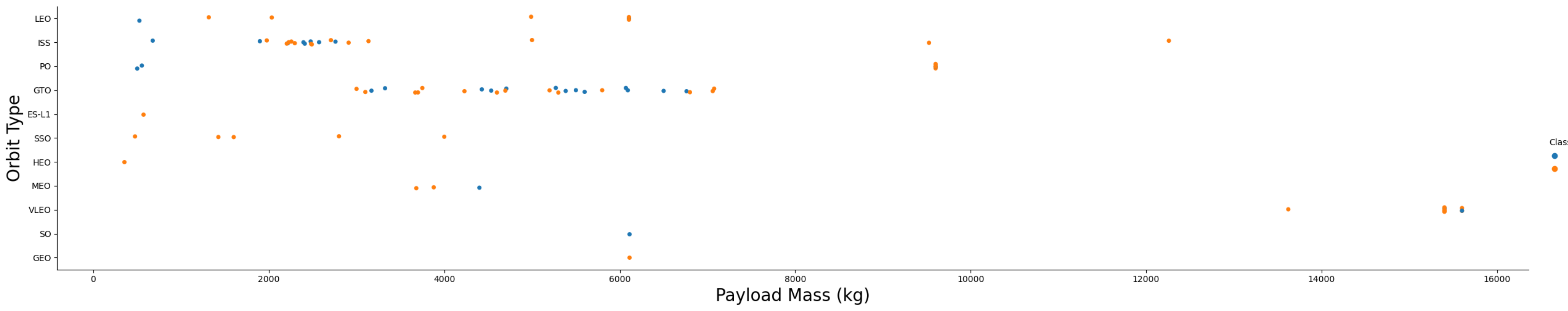
Flight Number vs. Orbit Type

- In summary, the success rate of missions in low Earth orbit (LEO) is influenced by the number of flights, whereas there is no correlation between flight number and success rate in geostationary transfer orbit (GTO). However, overall, the success rate has shown improvement over time in all orbits. The increasing frequency of missions in very low Earth orbit (VLEO) suggests that it presents a promising new business opportunity.



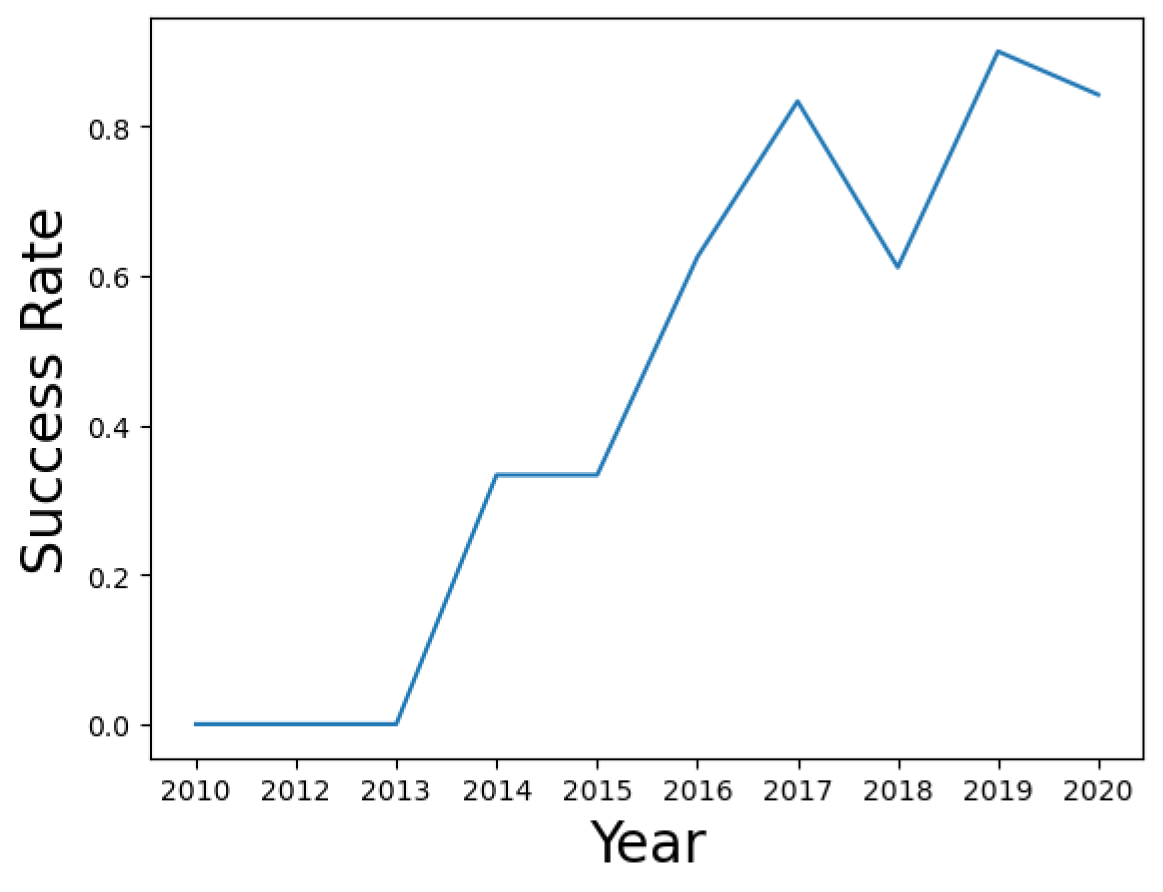
Payload vs. Orbit Type

- Summary: The weight of payloads affects different types of orbits differently. Heavy payloads have a negative impact on GTO (Geostationary Transfer Orbit) orbits and a positive impact on GTO and Polar LEO (Low Earth Orbit) orbits. Surprisingly, there seems to be no correlation between payload weight and the success rate of achieving GTO orbit. The International Space Station (ISS) has the widest range of payload capabilities and a high success rate. On the other hand, there are relatively fewer launches to SO (Sun-synchronous Orbit) and GEO (Geostationary Orbit) orbits.



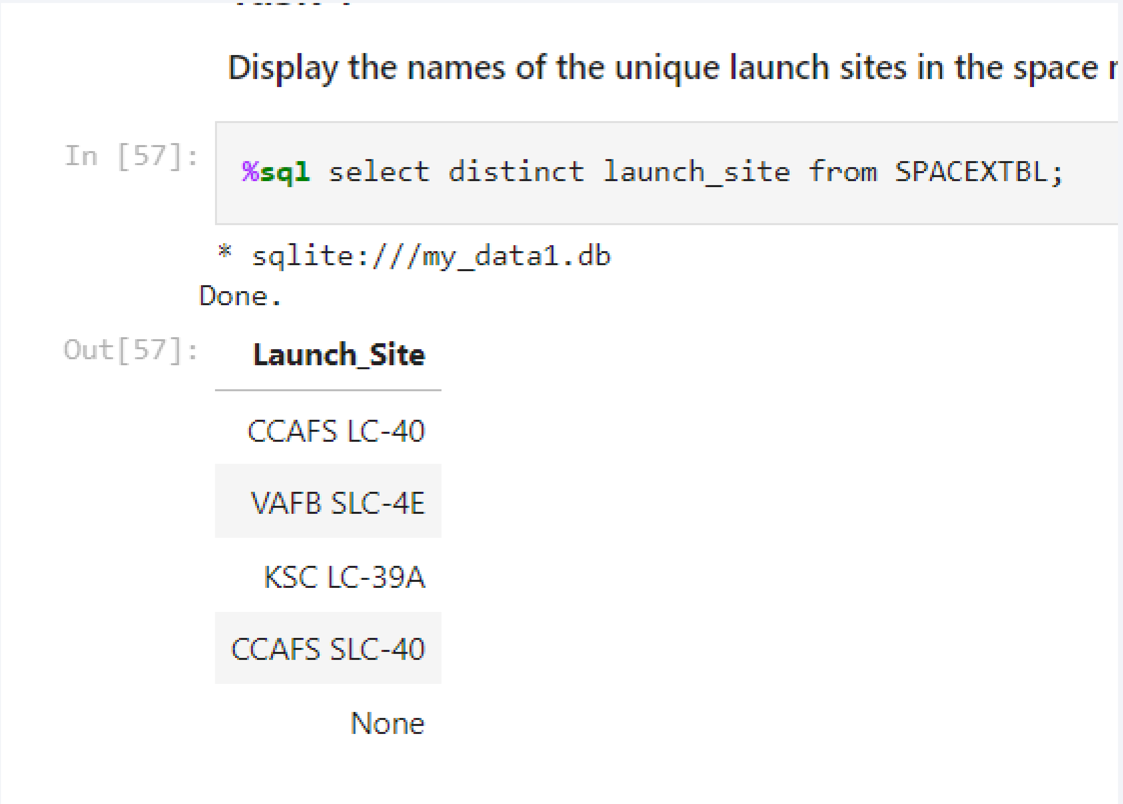
Launch Success Yearly Trend

Almost every year does the success rate get better



All Launch Site Names

- Unique launch sites



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

In [57]:
```
%sql select distinct launch_site from SPACEXTBL;
```

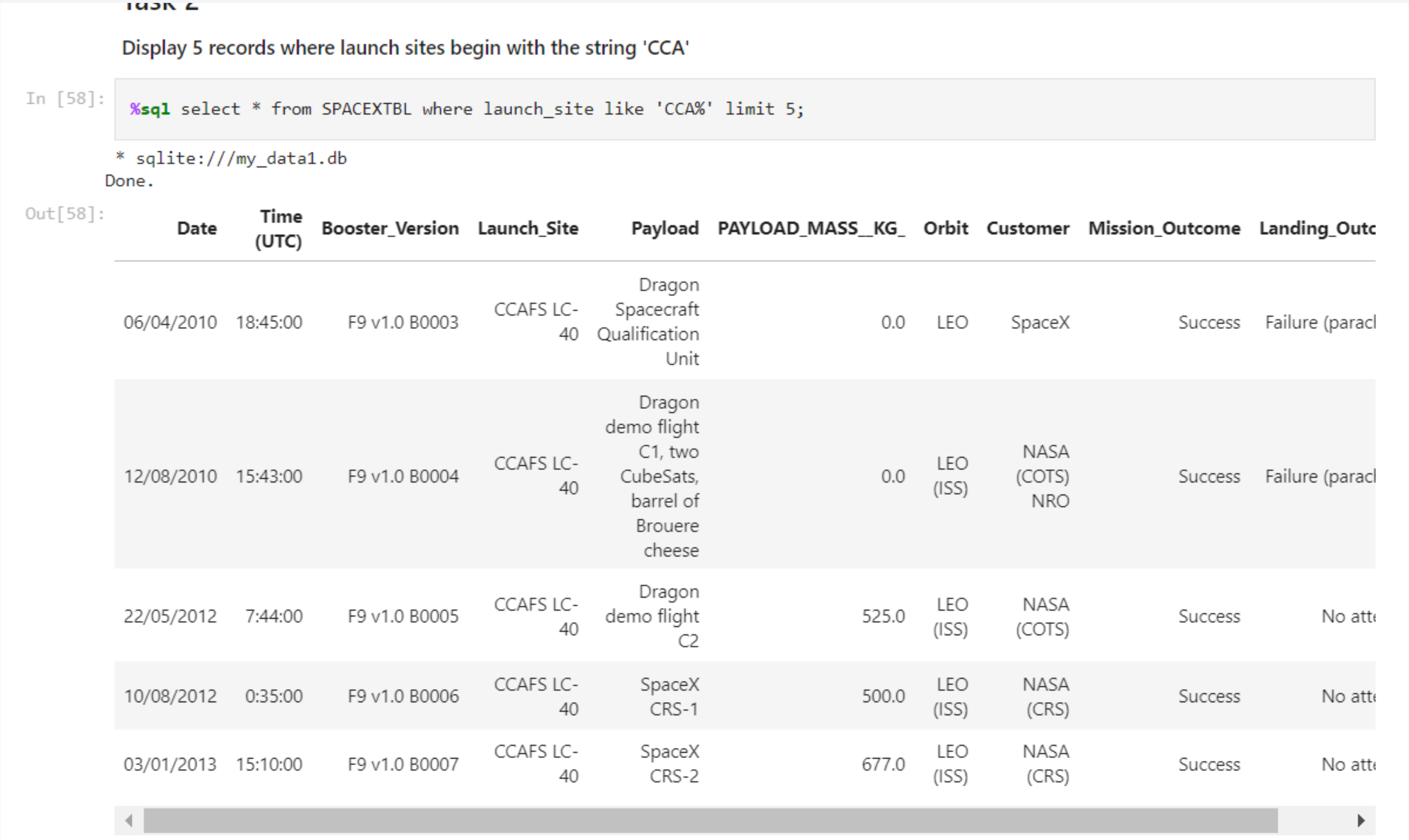
```
 * sqlite:///my_data1.db
Done.
```

Out[57]:

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'

- Launch sites begin with `CCA` the list is comprehensive



Task 2

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

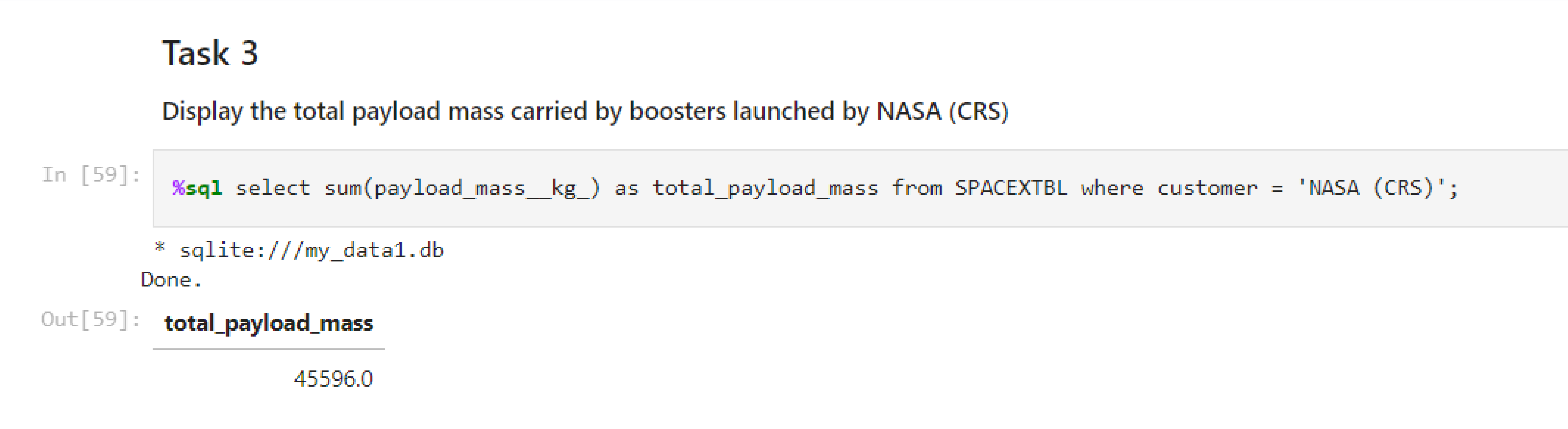
```
In [58]:  %sql select * from SPACEXTBL where launch_site like 'CCA%' limit 5;

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

Out[58]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No atte
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No atte
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No atte

Total Payload Mass

- These are the lines that display the Total Payload Mass



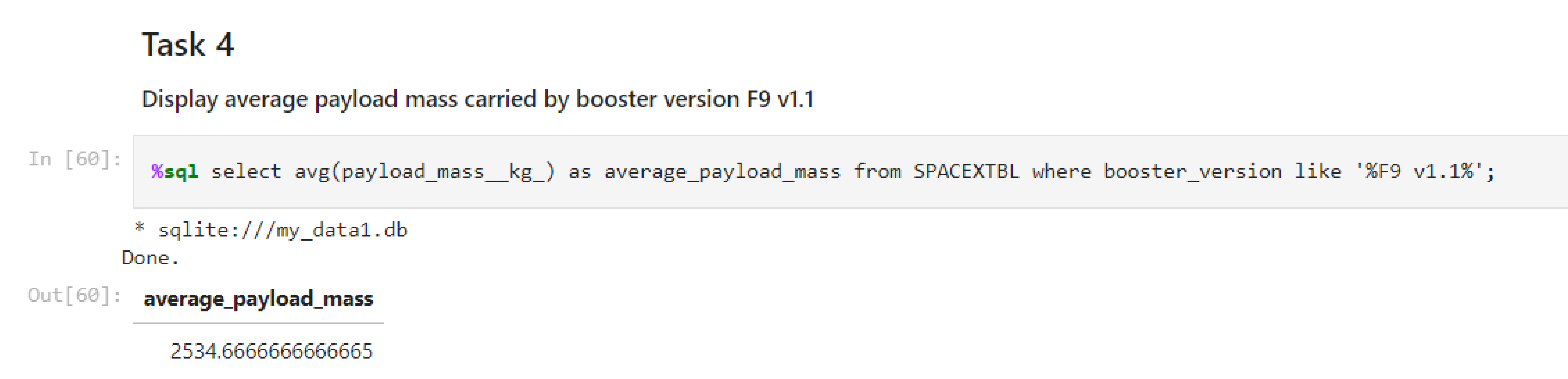
Task 3

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

```
In [59]: %sql select sum(payload_mass__kg_) as total_payload_mass from SPACEXTBL where customer = 'NASA (CRS)';

 * sqlite:///my_data1.db
Done.

Out[59]: total_payload_mass
                    45596.0
```

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.



Task 4

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

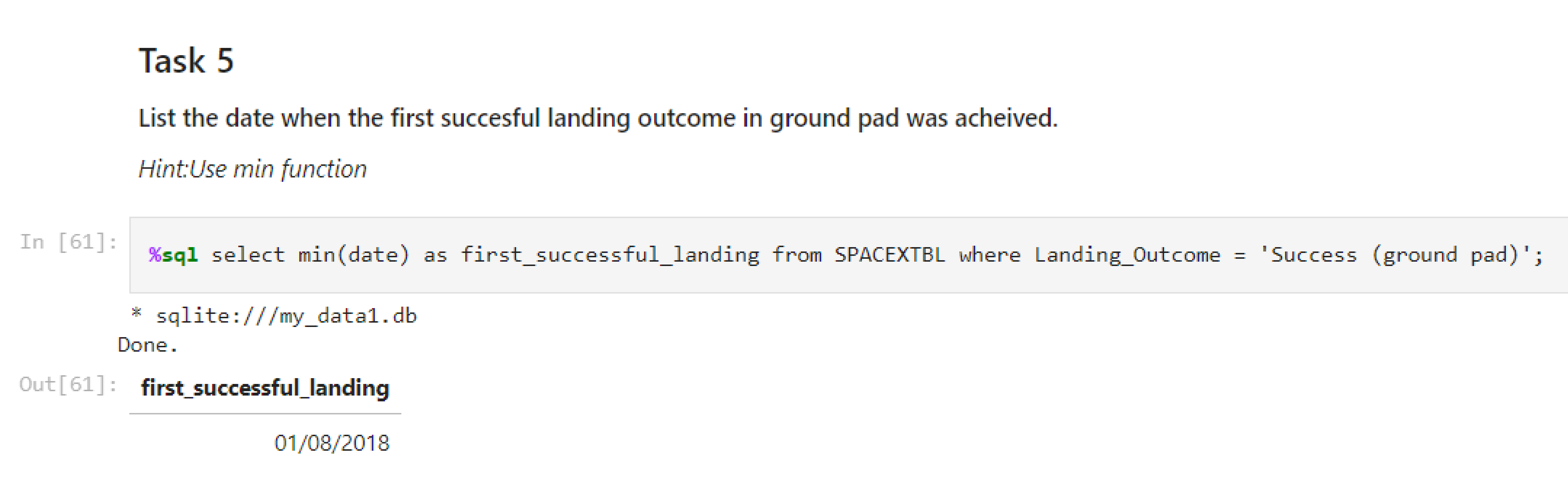
```
In [60]: %sql select avg(payload_mass__kg_) as average_payload_mass from SPACEXTBL where booster_version like '%F9 v1.1%';
```

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

```
Out[60]: average_payload_mass
           2534.6666666666665
```

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad



Task 5

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

Hint:Use min function

```
In [61]: %sql select min(date) as first_successful_landing from SPACEXTBL where Landing_Outcome = 'Success (ground pad)';
```

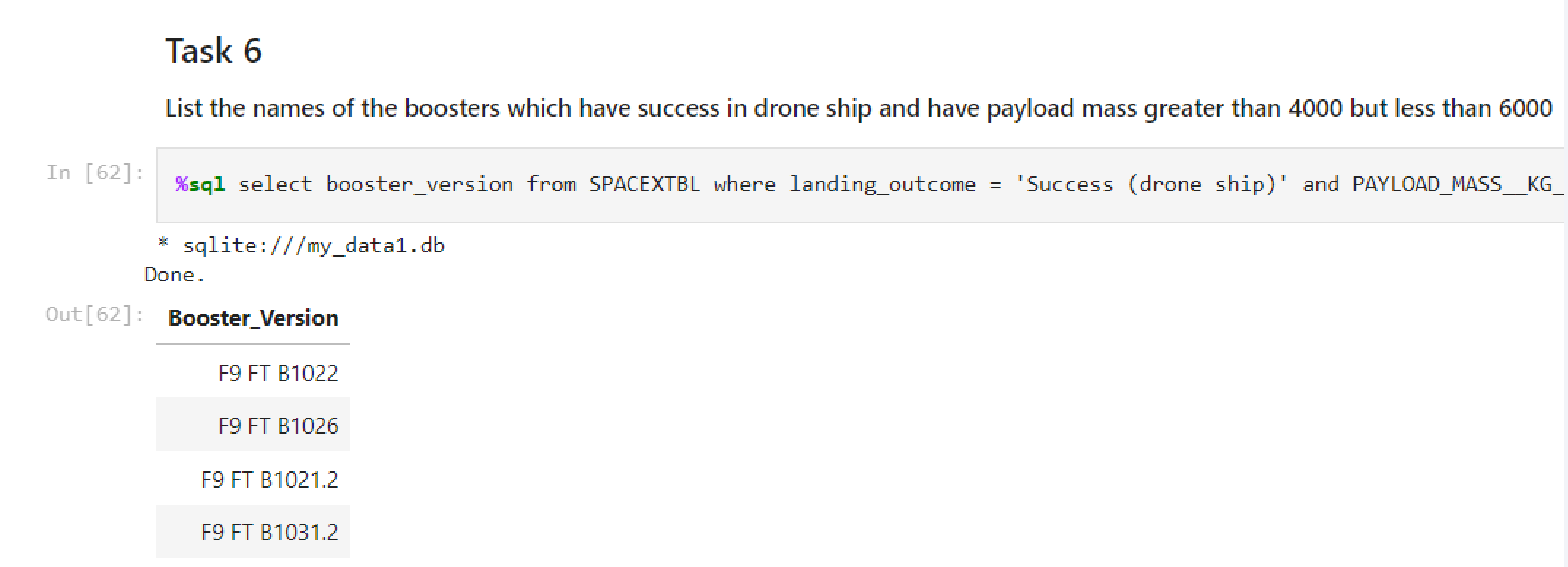
```
 * sqlite:///my_data1.db
Done.
```

Out[61]:

first_successful_landing
01/08/2018

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had 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

```
In [62]: %sql select booster_version from SPACEXTBL where landing_outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_
```

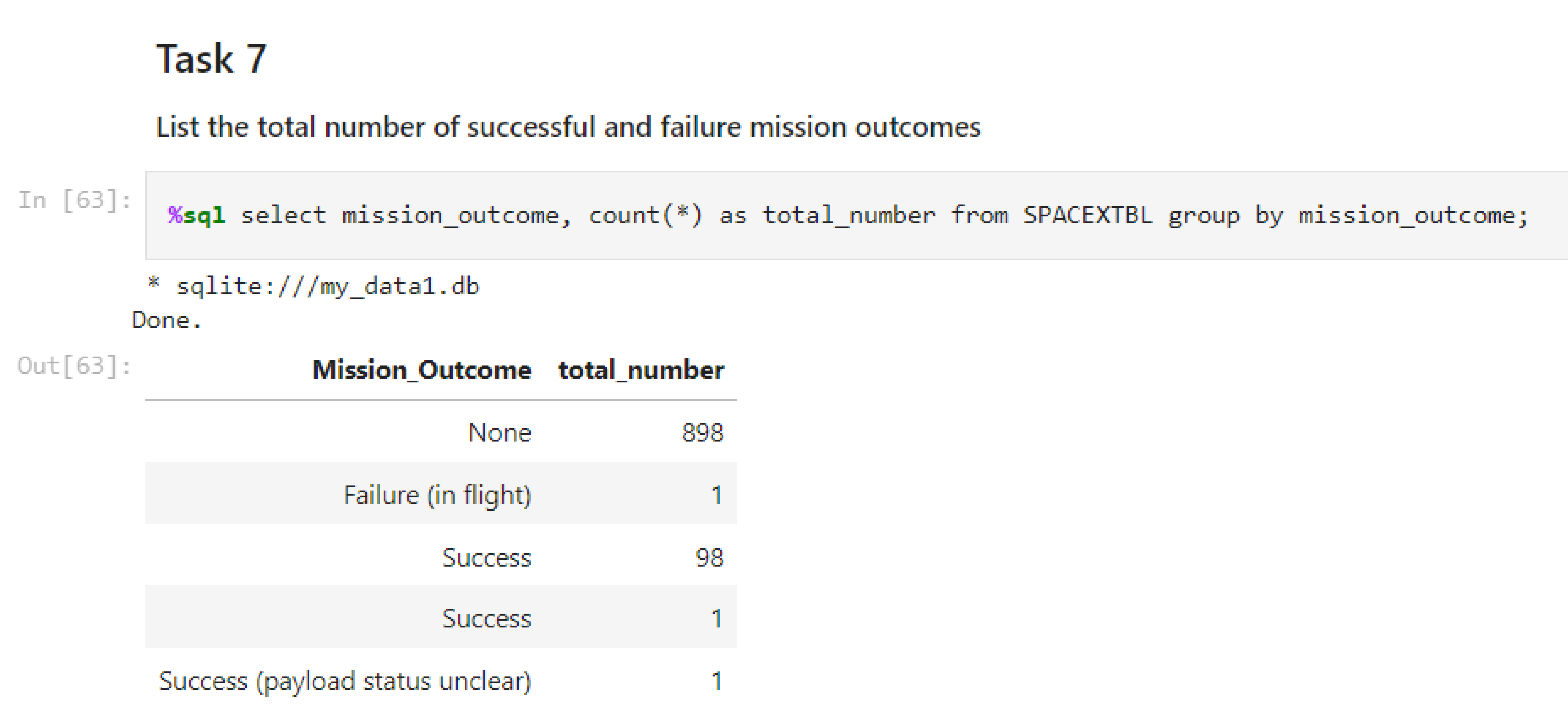
```
 * sqlite:///my_data1.db
Done.
```

Out[62]:

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

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



Task 7

List the total number of successful and failure mission outcomes

```
In [63]:  %sql select mission_outcome, count(*) as total_number from SPACEXTBL group by mission_outcome;
```

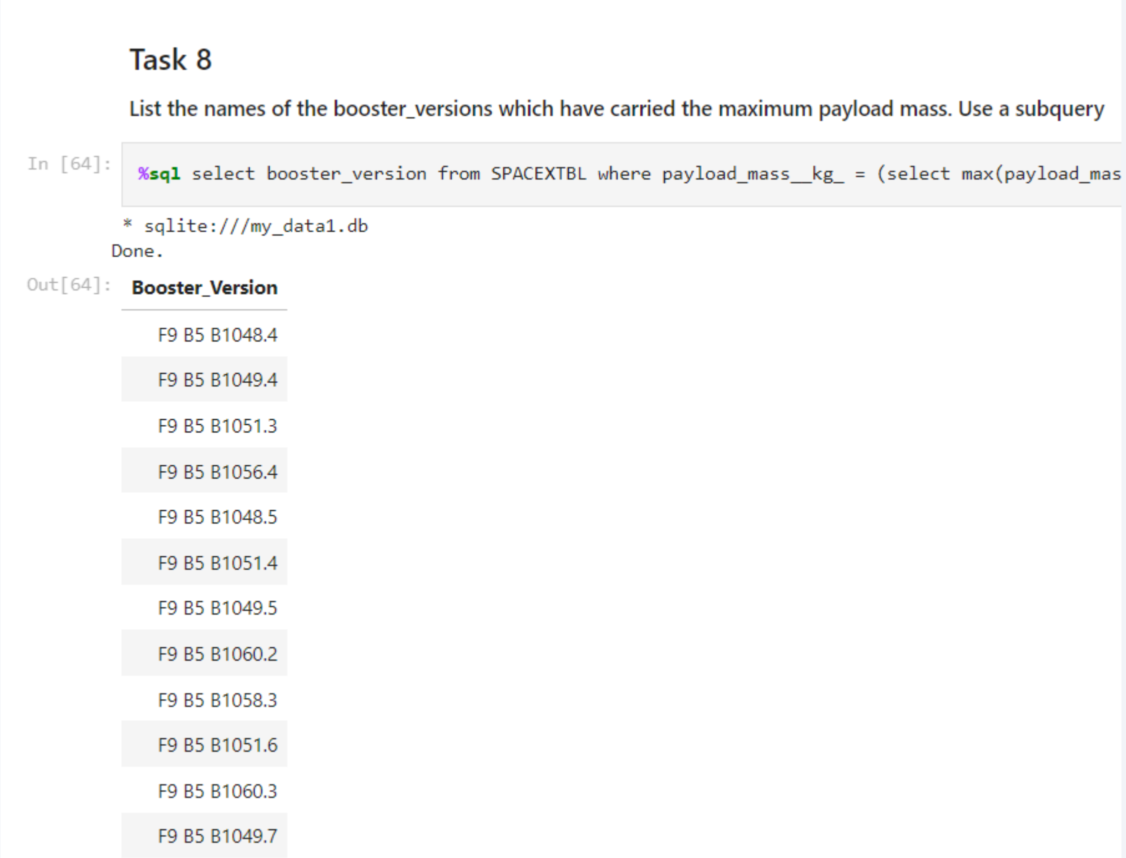
```
 * sqlite:///my_data1.db
Done.
```

Out[63]:

Mission_Outcome	total_number
None	898
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass



Task 8

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

In [64]:
```
%sql select booster_version from SPACEXTBL where payload_mass__kg_ = (select max(payload_mas
```

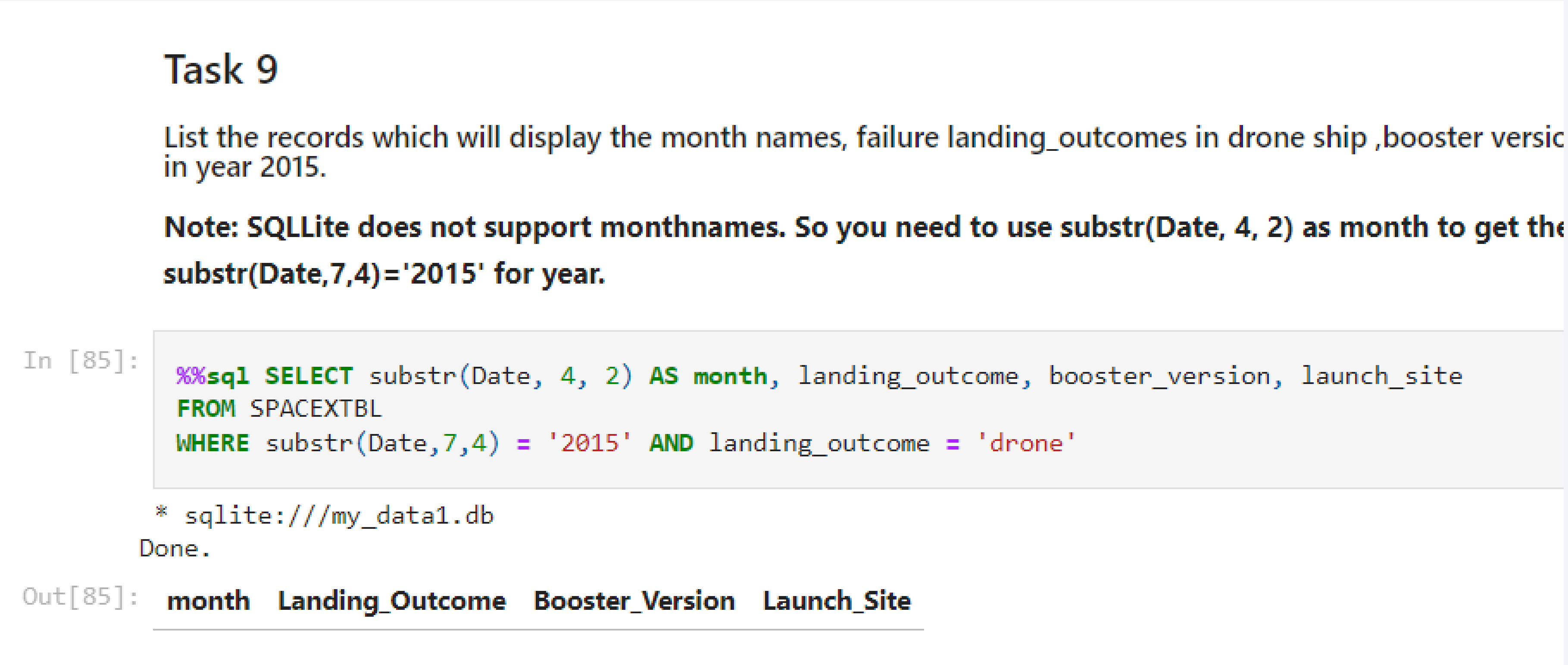
```
 * sqlite:///my_data1.db
Done.
```

Out[64]:

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

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015>>> THERE WERE NO RECORDS IN THE CSV



Task 9

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versic in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the substr(Date,7,4)='2015' for year.

```
In [85]: %%sql SELECT substr(Date, 4, 2) AS month, landing_outcome, booster_version, launch_site
         FROM SPACEXTBL
         WHERE substr(Date,7,4) = '2015' AND landing_outcome = 'drone'
```

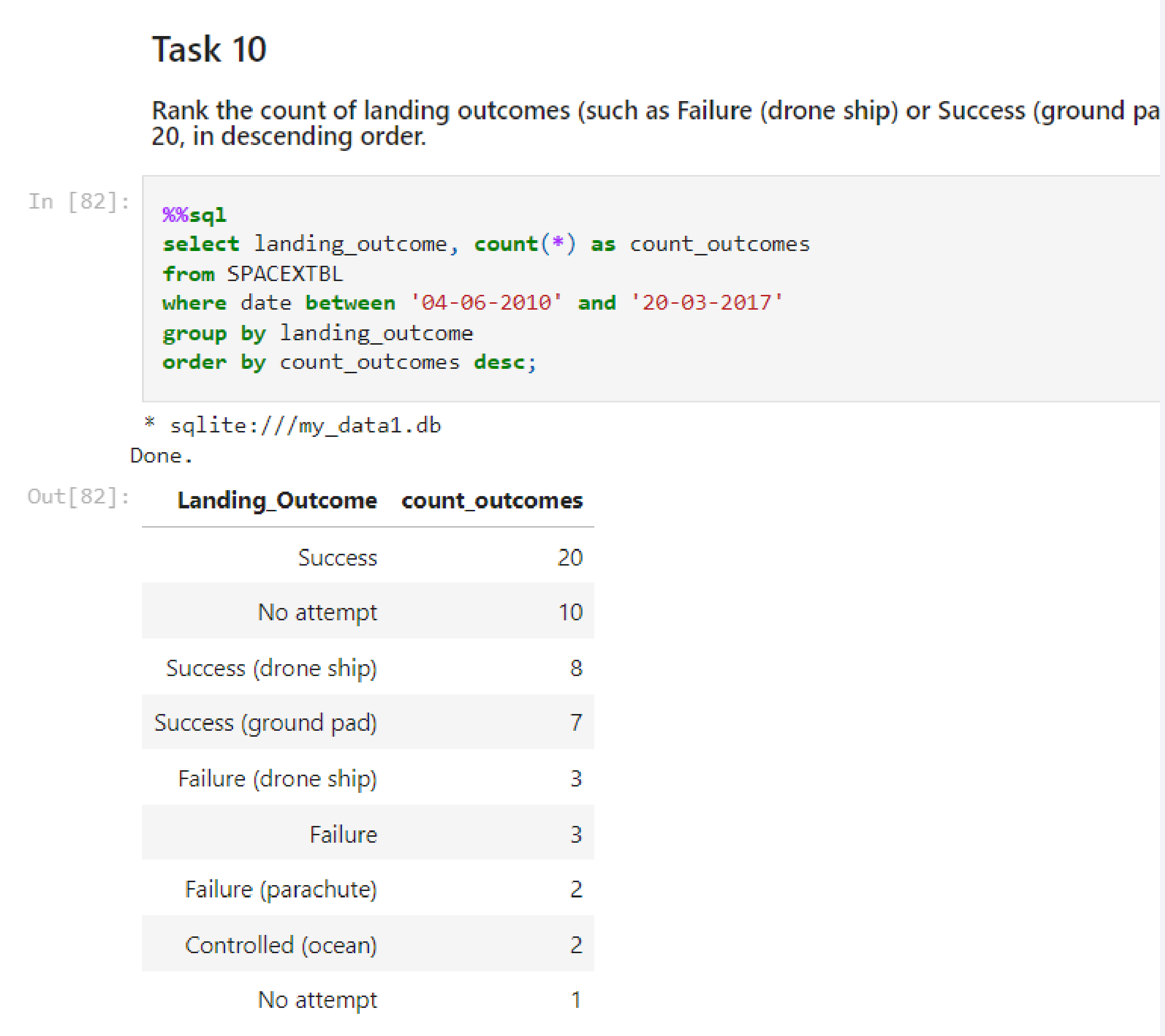
```
 * sqlite:///my_data1.db
Done.
```

Out[85]:

month	Landing_Outcome	Booster_Version	Launch_Site

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

- 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



Task 10

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pa
20, in descending order.

In [82]:

```
%%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
Done.
```

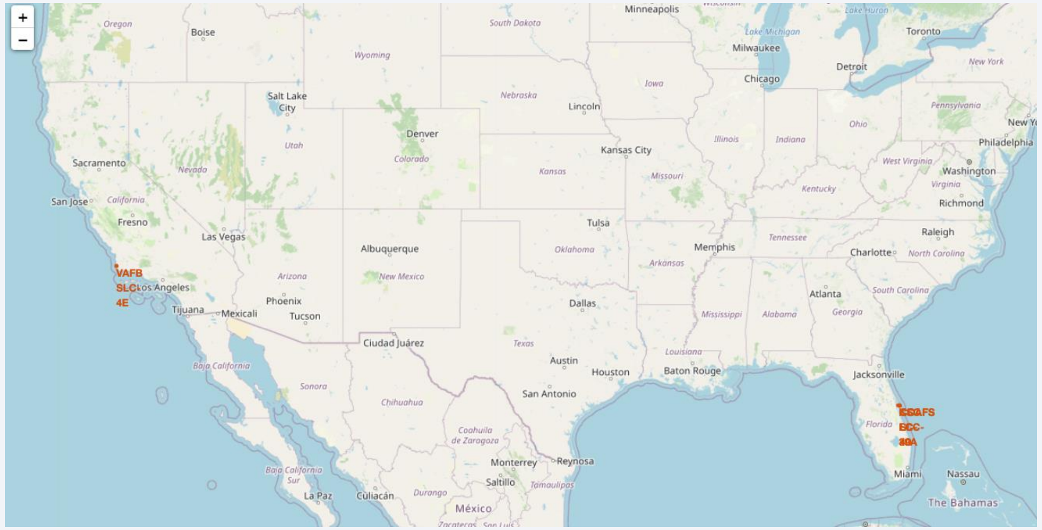
Out[82]:

Landing_Outcome	count_outcomes
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

Launch Sites Proximities Analysis

Location of all Launches

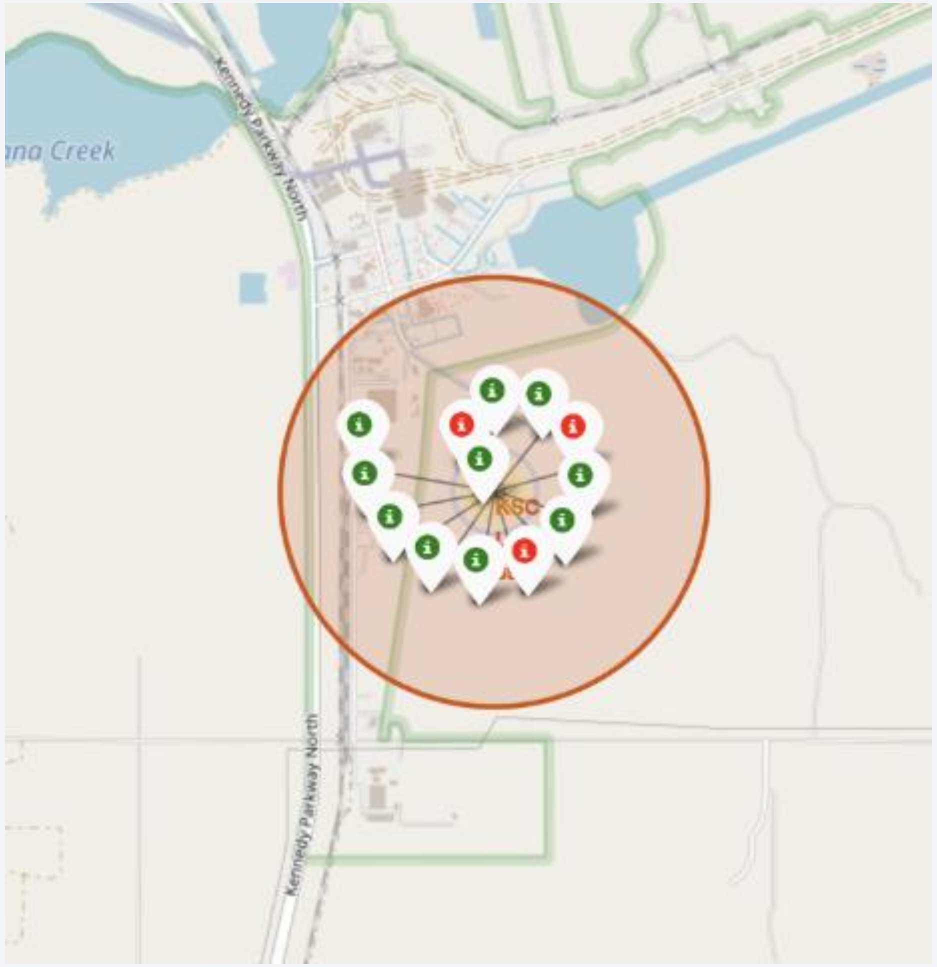


- Summary: Launch sites are often located near the equator due to the higher speed of the land in that region. Objects on the equator are already moving at a fast speed, and when a spacecraft is launched from there, it maintains its initial speed due to inertia. This high speed helps the spacecraft stay in orbit. Additionally, launch sites are situated close to the coast to minimize the risk of debris falling or exploding near populated areas.

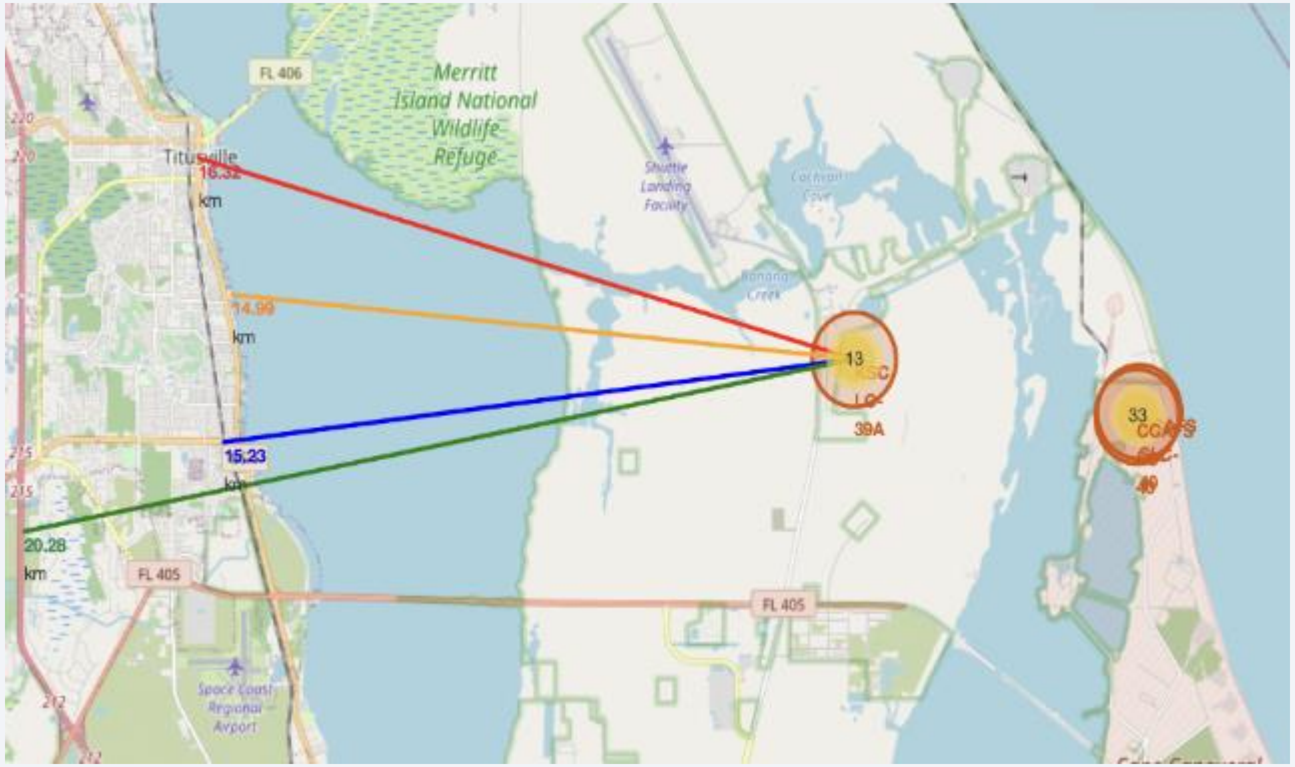
Labels by color of Launch Sites

- Summary: The markers' colors indicate the success or failure of launch sites, with green representing successful launches and red representing failed launches. The launch site KSC LC-39A has a notably high success rate.

- Rephrased: By observing the color-coded markers, it becomes simple to distinguish launch sites with high success rates. A green marker indicates a successful launch, while a red marker indicates a failed one. Notably, the launch site KSC LC-39A boasts an exceptionally high success rate.



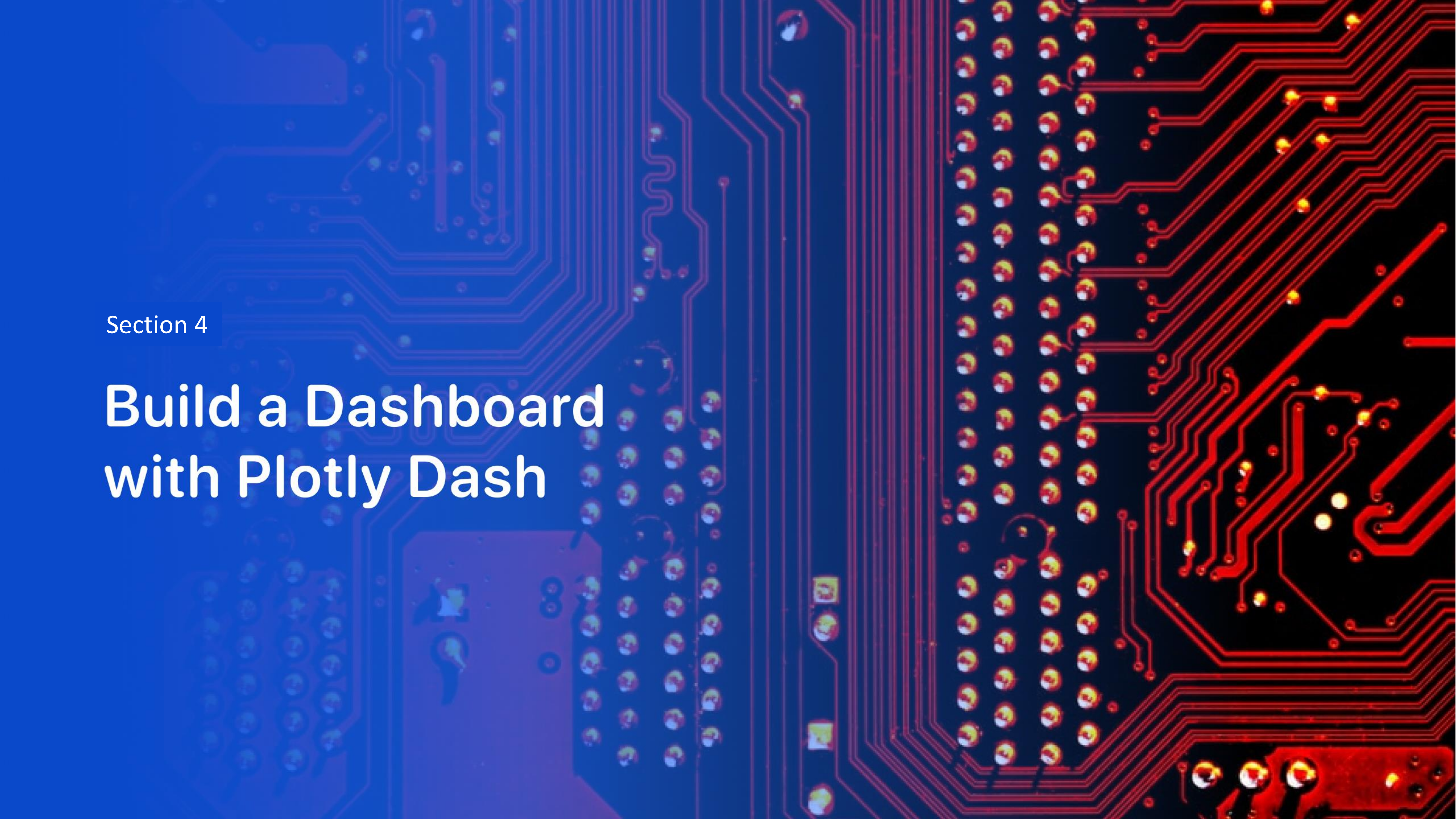
Location Selection due to logistical support



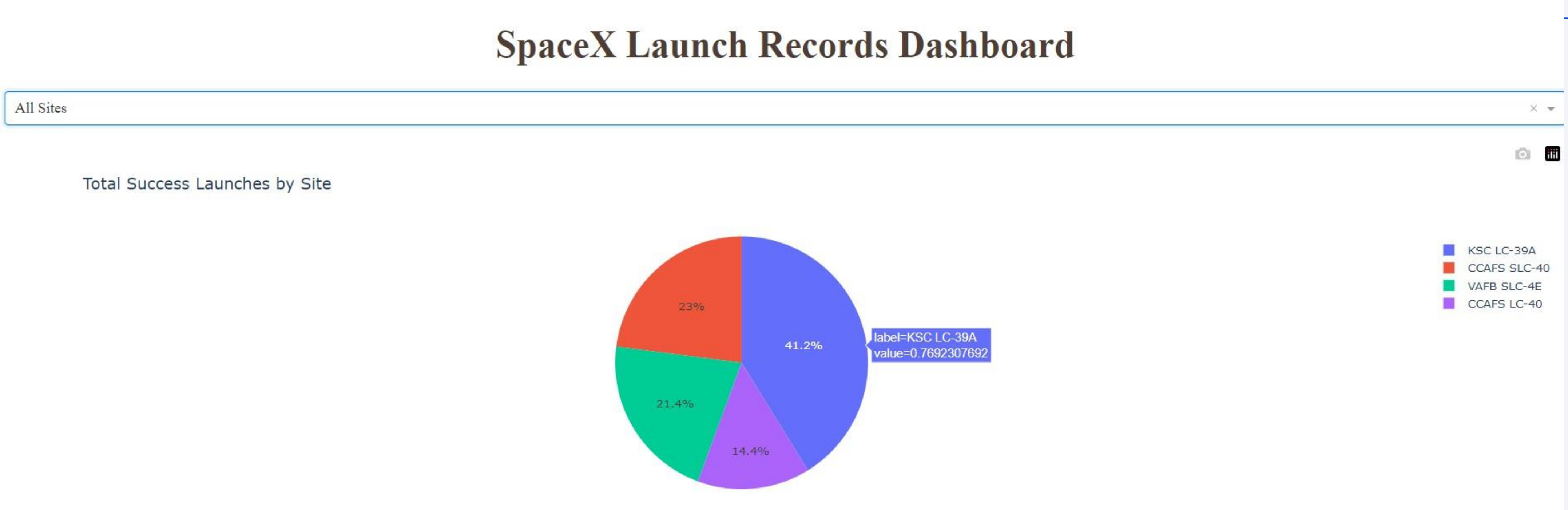
- The launch site, KSC LC-39A, is located in close proximity to various transportation and geographical features. It is approximately 15.23 km away from a railway, 20.28 km away from a highway, and 14.99 km away from the coastline. Additionally, the site is relatively close to the nearest city, Titusville, at a distance of 16.32 km. This poses a potential risk as a failed rocket traveling at high speeds could reach these populated areas within seconds, posing a danger to the residents.

Section 4

Build a Dashboard with Plotly Dash



Successful Launches by Site

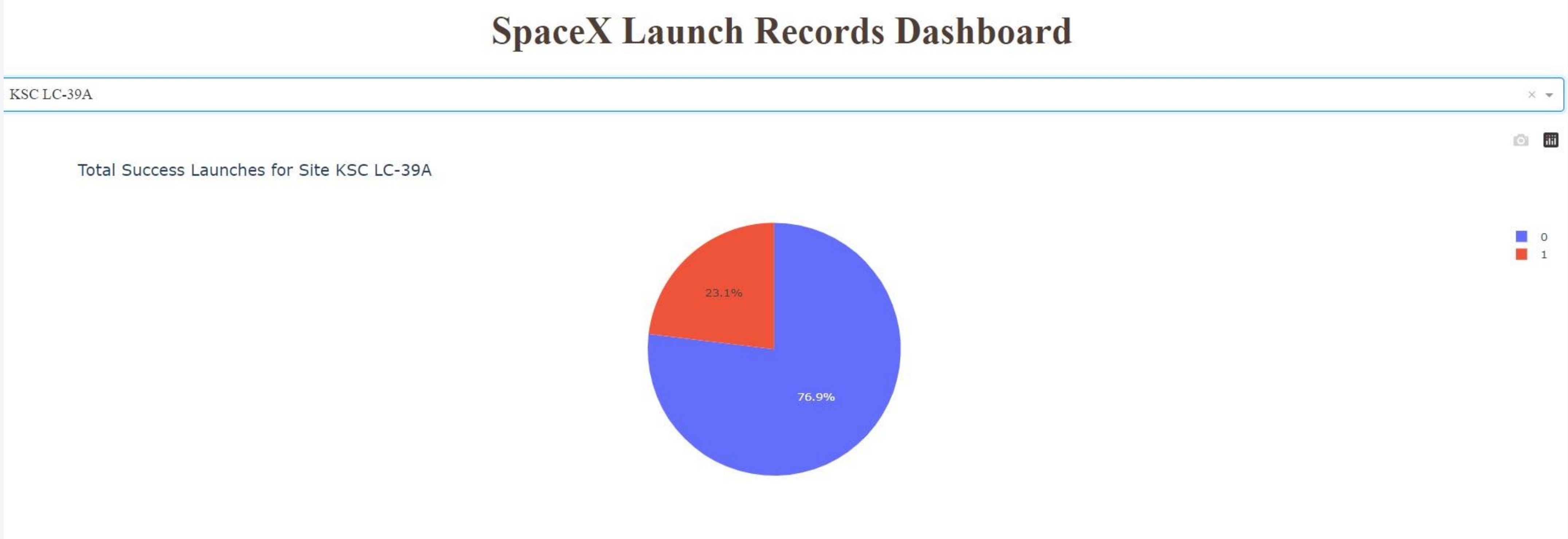


SpaceX Launch Records Dashboard

All Sites

- Explanation:
- The chart clearly shows that from all the sites, KSC LC-39A has the most
- successful launches.

Launch site with highest launch success ratio



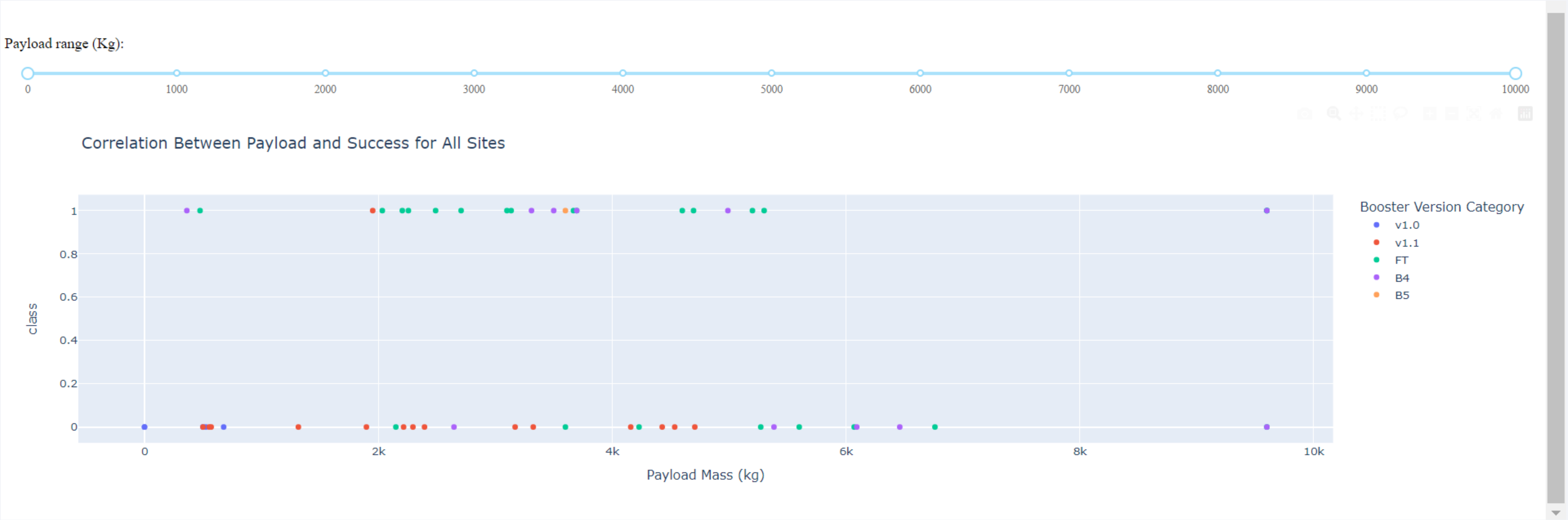
SpaceX Launch Records Dashboard

KSC LC-39A

Total Success Launches for Site KSC LC-39A

- Ca 80% of launches are successful in this site.

Payload Mass vs. Launch Outcome for all sites

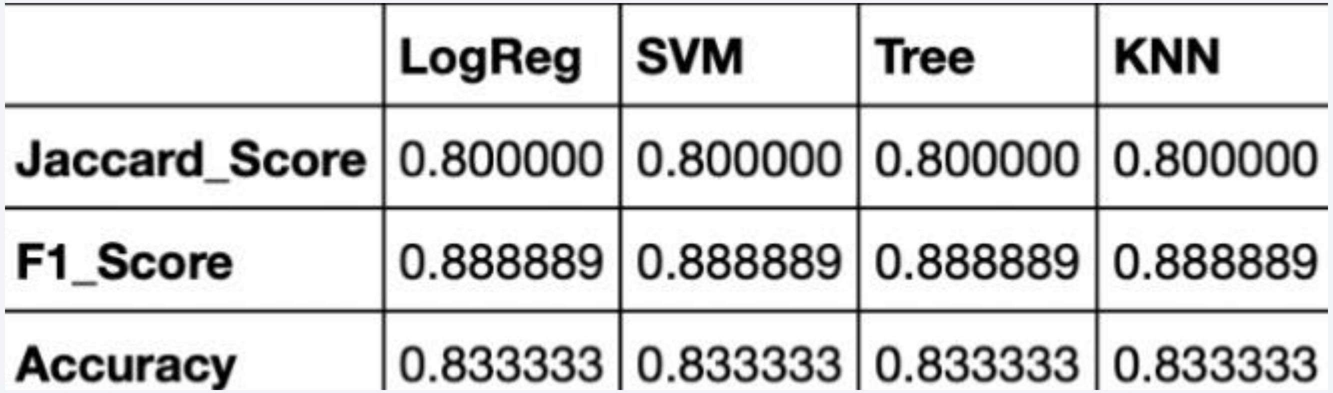


- There are more succeses under the 6k payload range

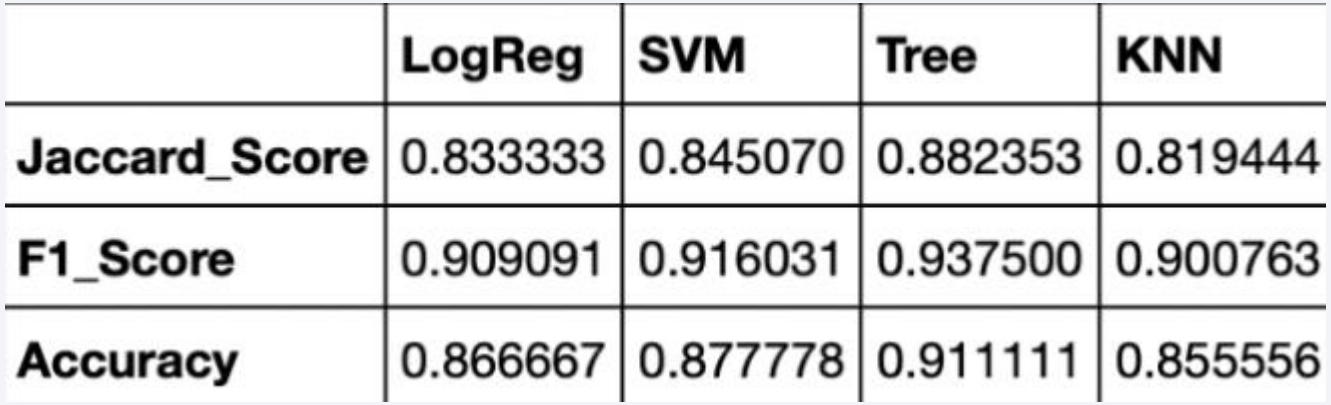
Section 5

Predictive Analysis (Classification)

Classification Accuracy

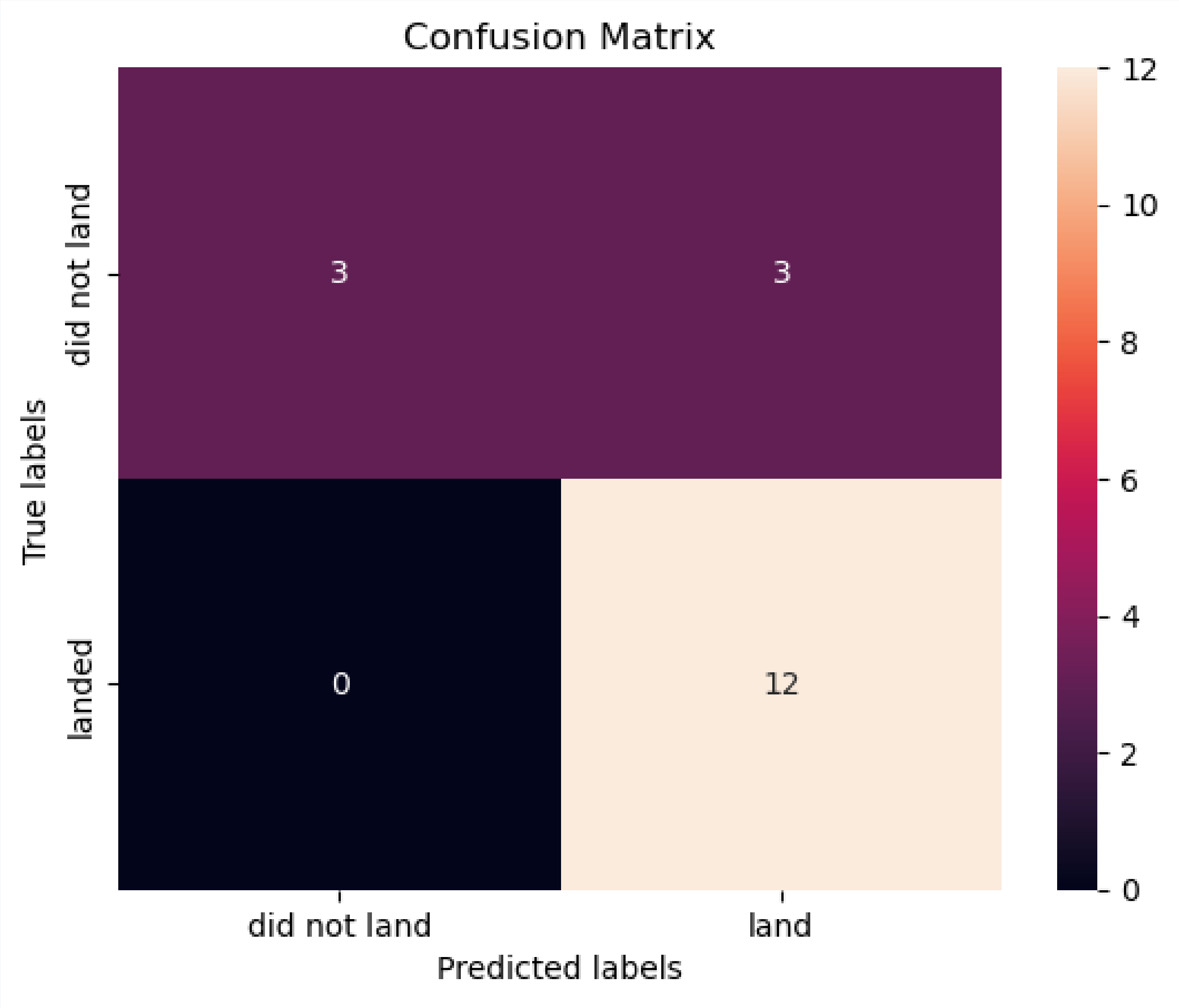


	LogReg	SVM	Tree	KNN
Jaccard_Score	0.800000	0.800000	0.800000	0.800000
F1_Score	0.888889	0.888889	0.888889	0.888889
Accuracy	0.833333	0.833333	0.833333	0.833333



	LogReg	SVM	Tree	KNN
Jaccard_Score	0.833333	0.845070	0.882353	0.819444
F1_Score	0.909091	0.916031	0.937500	0.900763
Accuracy	0.866667	0.877778	0.911111	0.855556

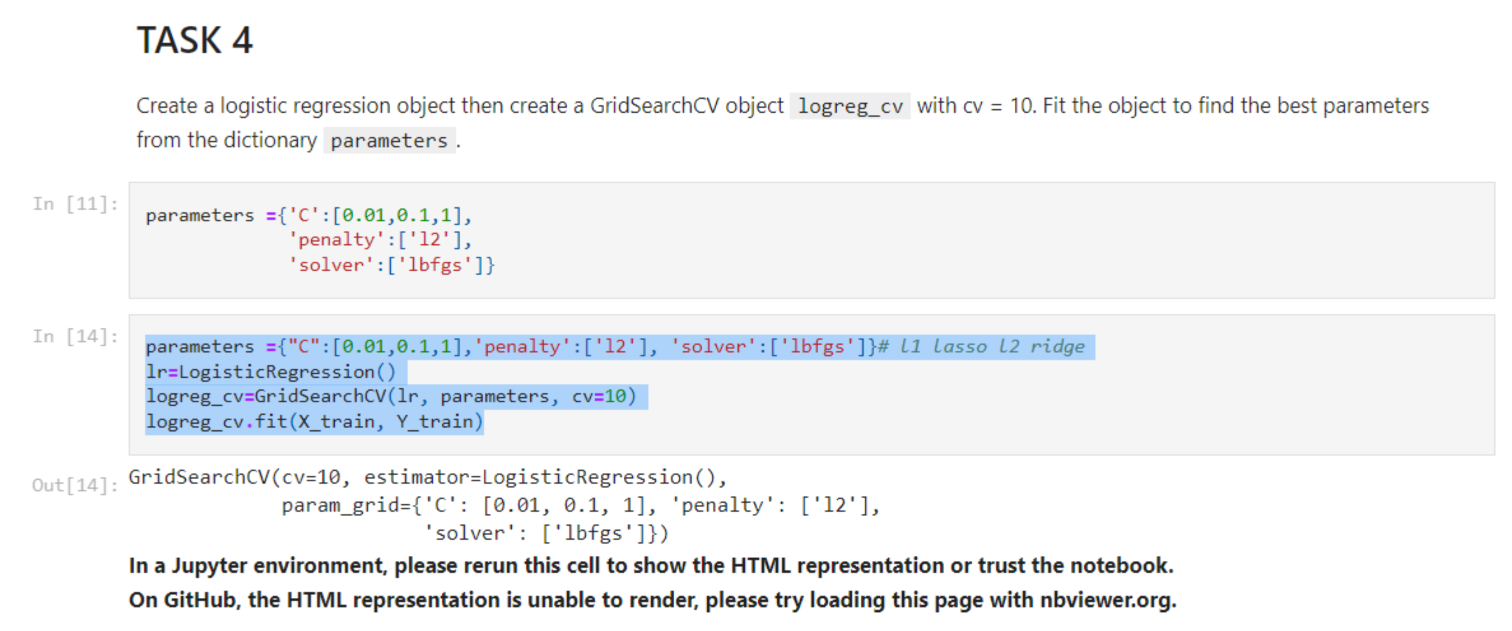
Confusion Matrix



Conclusions

- - Decision Tree Model is the optimal algorithm for this dataset.
- - Launches with lower payload mass perform better than those with larger payload mass.
- - Most launch sites are located near the Equator and close to the coast.
- - Launch success rate increases over the years.
- - KSC LC-39A has the highest success rate among all launch sites.
- - Orbits ES-LI, GEO, HEO, and SSO have a 100% success rate.
- - Various data sources were analyzed to refine conclusions.
- - KSC LC-39A is the recommended launch site.
- - Launches with a payload above 7,000kg are less risky.
- - Successful landing outcomes improve over time, reflecting advancements in processes and rockets.
- - Using a Decision Tree Classifier can predict successful landings and enhance profitability.

Appendix



TASK 4

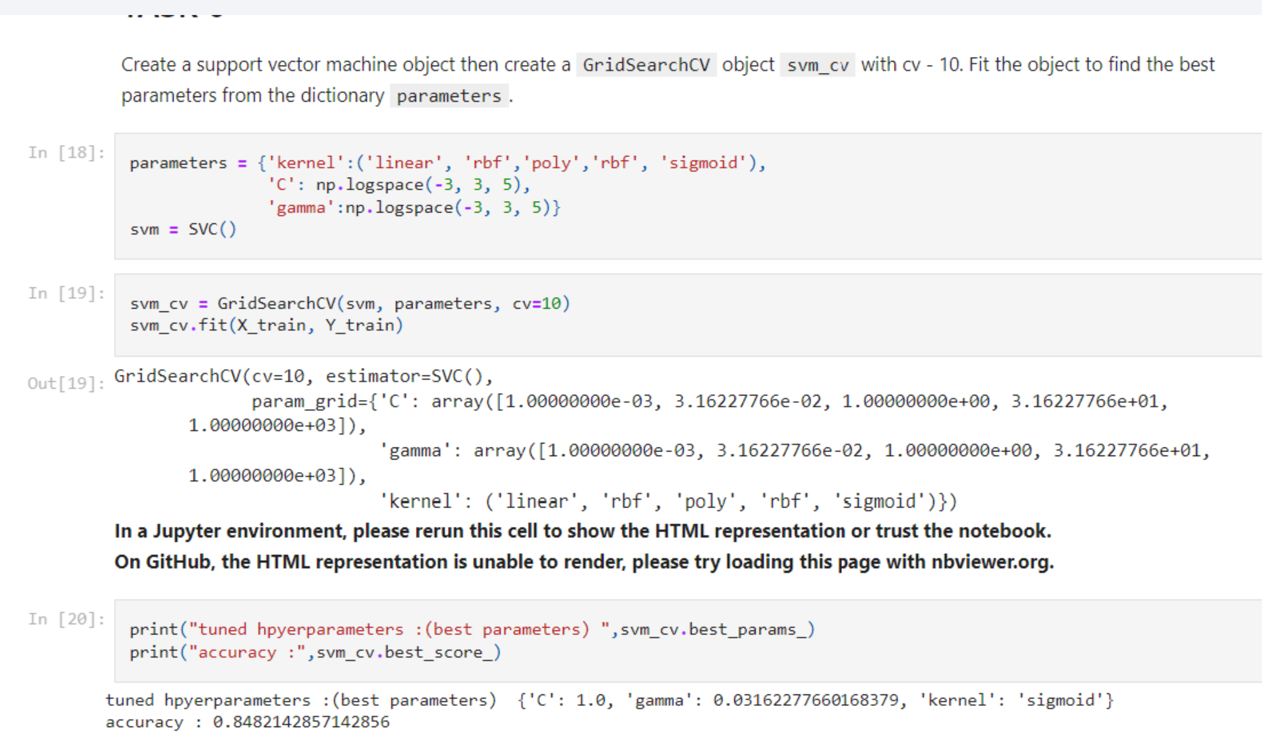
Create a logistic regression object then create a GridSearchCV object `logreg_cv` with cv = 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
In [11]: parameters ={'C':[0.01,0.1,1],
                      'penalty':['l2'],
                      'solver':['lbfgs']}
```

```
In [14]: parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
         lr=LogisticRegression()
         logreg_cv=GridSearchCV(lr, parameters, cv=10)
         logreg_cv.fit(X_train, Y_train)
```

```
Out[14]: GridSearchCV(cv=10, estimator=LogisticRegression(),
                      param_grid={'C': [0.01, 0.1, 1], 'penalty': ['l2'],
                                  'solver': ['lbfgs']})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.



Create a support vector machine object then create a `GridSearchCV` object `svm_cv` with cv - 10. Fit the object to find the best parameters from the dictionary `parameters`.

```
In [18]: parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'),
                       'C': np.logspace(-3, 3, 5),
                       'gamma':np.logspace(-3, 3, 5)}
         svm = SVC()
```

```
In [19]: svm_cv = GridSearchCV(svm, parameters, cv=10)
         svm_cv.fit(X_train, Y_train)
```

```
Out[19]: GridSearchCV(cv=10, estimator=SVC(),
                      param_grid={'C': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
                1.00000000e+03]),
                                  'gamma': array([1.00000000e-03, 3.16227766e-02, 1.00000000e+00, 3.16227766e+01,
                1.00000000e+03]),
                                  'kernel': ('linear', 'rbf', 'poly', 'rbf', 'sigmoid')})
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

```
In [20]: print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
         print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

Thank you!