IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Khwezi Duba

20 June 2024

Outline

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

Executive Summary

- Summary of methodologies
    - Data Collection through API
    - Data Collection with Web Scraping
    - Data Wrangling
    - Exploratory Data Analysis with SQL
    - Exploratory Data Analysis with Data Visualization
    - Interactive Visual Analytics with Folium
    - Machine Learning Prediction
- Summary of all results
    - Exploratory Data Analysis result
    - Interactive analytics in screenshots
    - Predictive Analytics result

Introduction

- Project background and context
  - Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

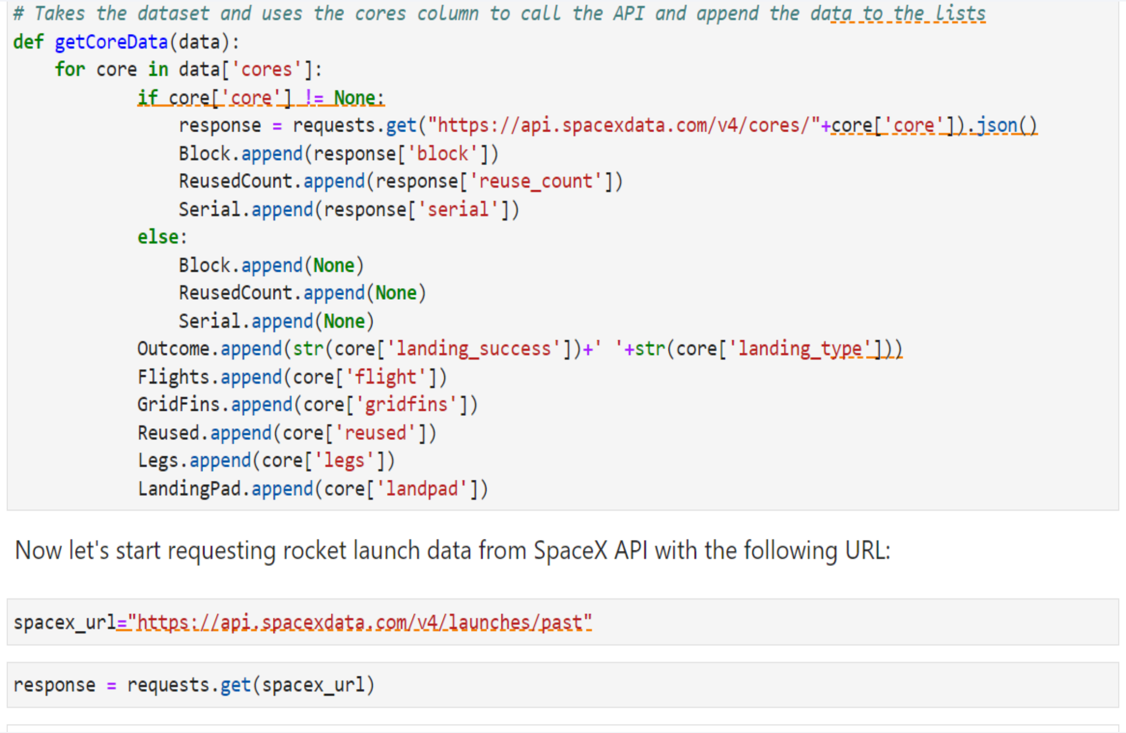
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- Data collection with SpaceX REST calls using key phrases and flowcharts.
- Get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting
- Link to the notebook : https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/Lab%201-spacex-data-collection-api.ipynb



```
# Takes the dataset and uses the cores column to call the API and append the data to the lists
def getCoreData(data):
    for core in data['cores']:
            if core['core'] != None:
                response = requests.get("https://api.spacexdata.com/v4/cores/"+core['core']).json()
                Block.append(response['block'])
                ReusedCount.append(response['reuse_count'])
                Serial.append(response['serial'])
            else:
                Block.append(None)
                ReusedCount.append(None)
                Serial.append(None)
            Outcome.append(str(core['landing_success'])+' '+str(core['landing_type']))
            Flights.append(core['flight'])
            GridFins.append(core['gridfins'])
            Reused.append(core['reused'])
            Legs.append(core['legs'])
            LandingPad.append(core['landpad'])
```

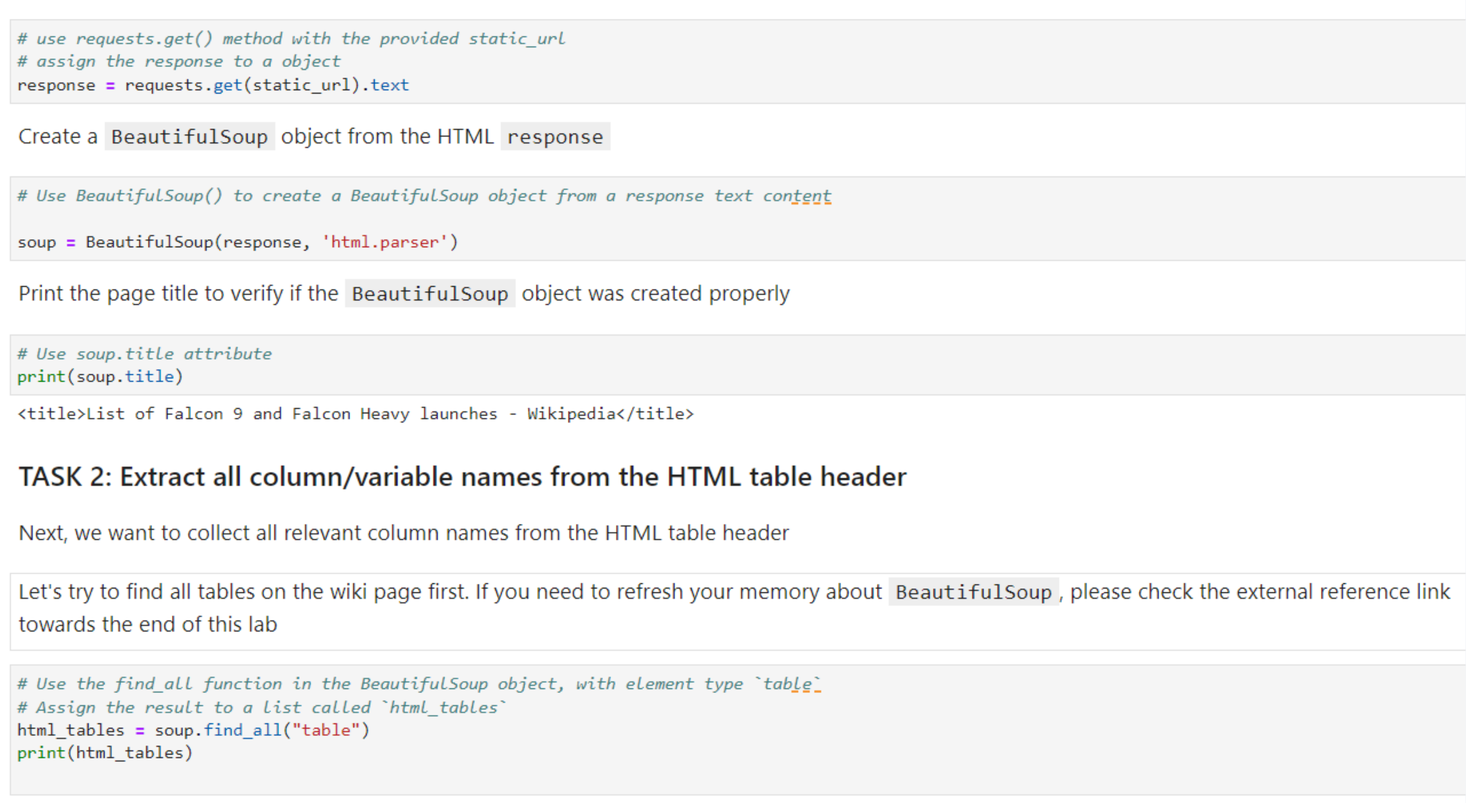
Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

```
response = requests.get(spacex_url)
```

Data Collection - Scraping

- I applied web scrapping to web scrape Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- Link to the notebook : https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/jupyter-labs-webscraping.ipynb



```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup = BeautifulSoup(response, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

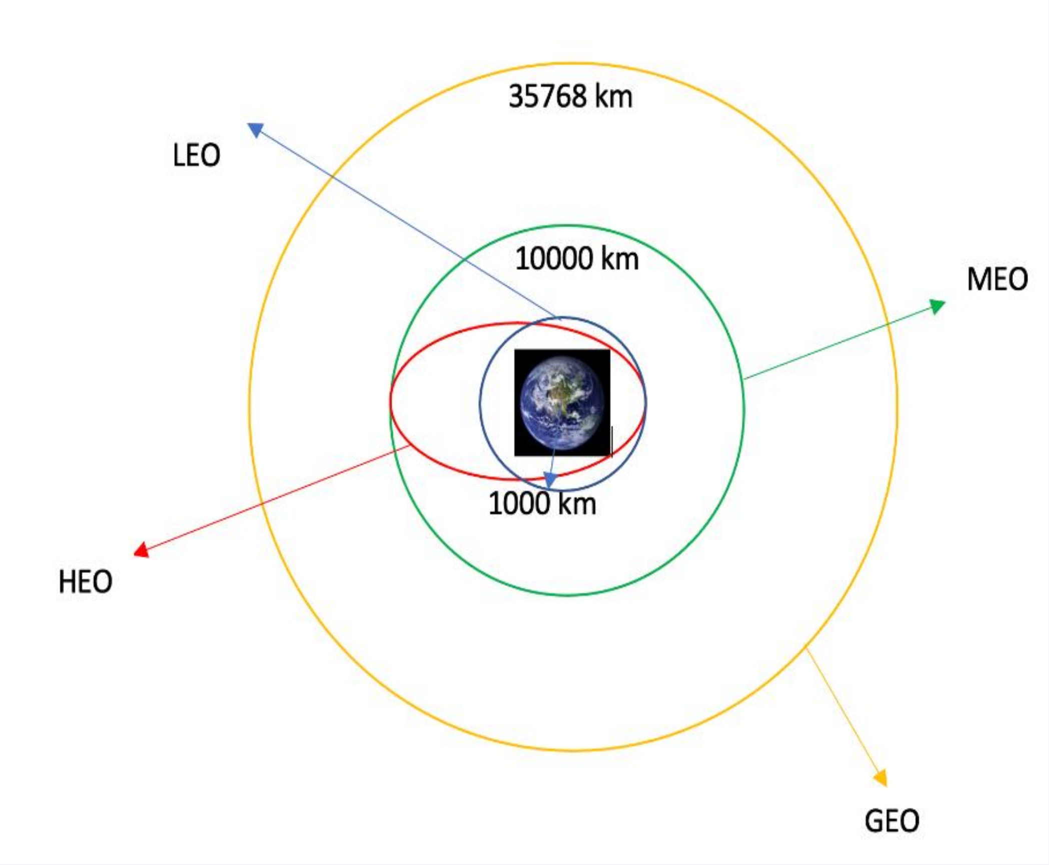
Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
print(html_tables)
```

Data Wrangling

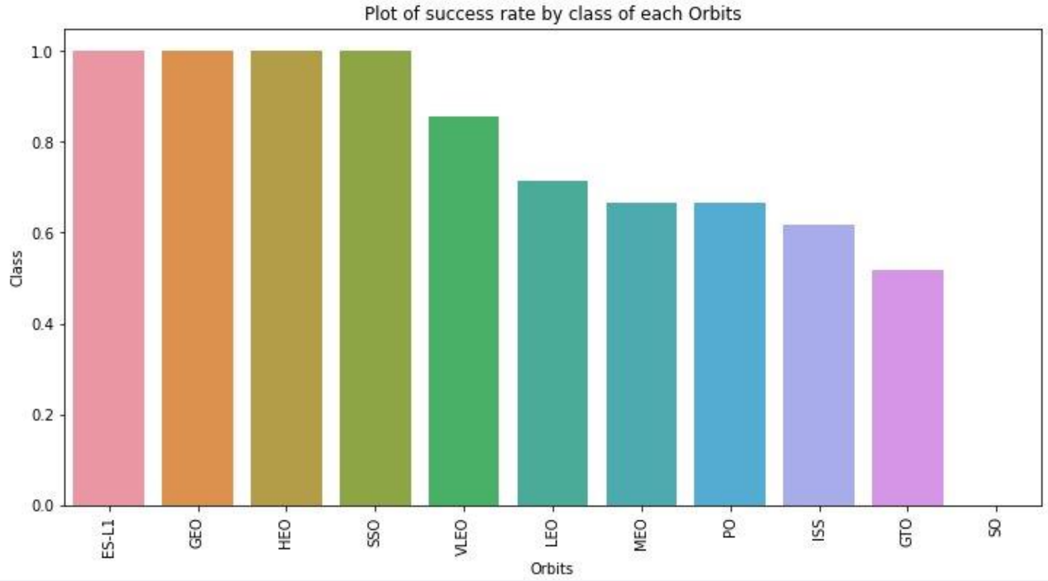
- I performed exploratory data analysis and determined the training labels.
- I calculated the number of launches at each site, and the number and occurrence of each orbits
- I then created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook : https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/lab%202-jupyter-spacex-Data%20wrangling.ipynb

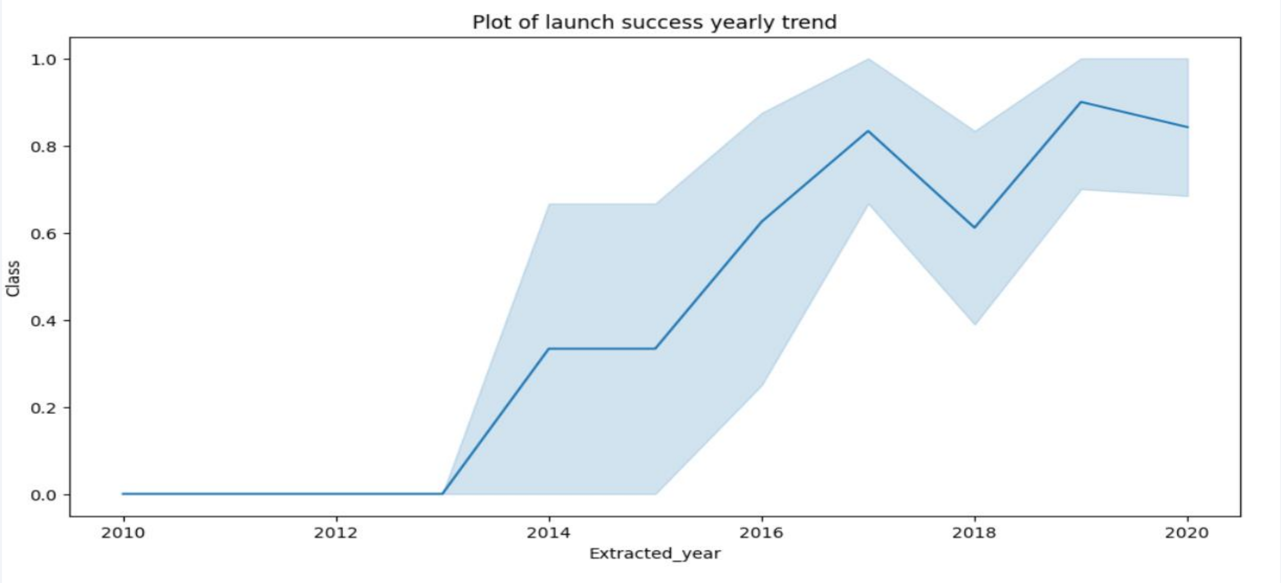


EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type,
  the launch success yearly trend.

Link to notebook :https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/edadataviz.ipynb





EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- Link to notebook : https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities. We answered some question for instance:
    - Are launch sites near railways, highways and coastlines.
    - Do launch sites keep certain distance away from cities.

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Plotted pie charts showing the total launches by a certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- Link: https://github.com/Odwa125/IBM_DS_CAPSTONE/blob/main/edadataviz.ipynb

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- Built different machine learning models and tune different hyperparameters using GridSearchCV.
- used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- Found the best performing classification model.

Results

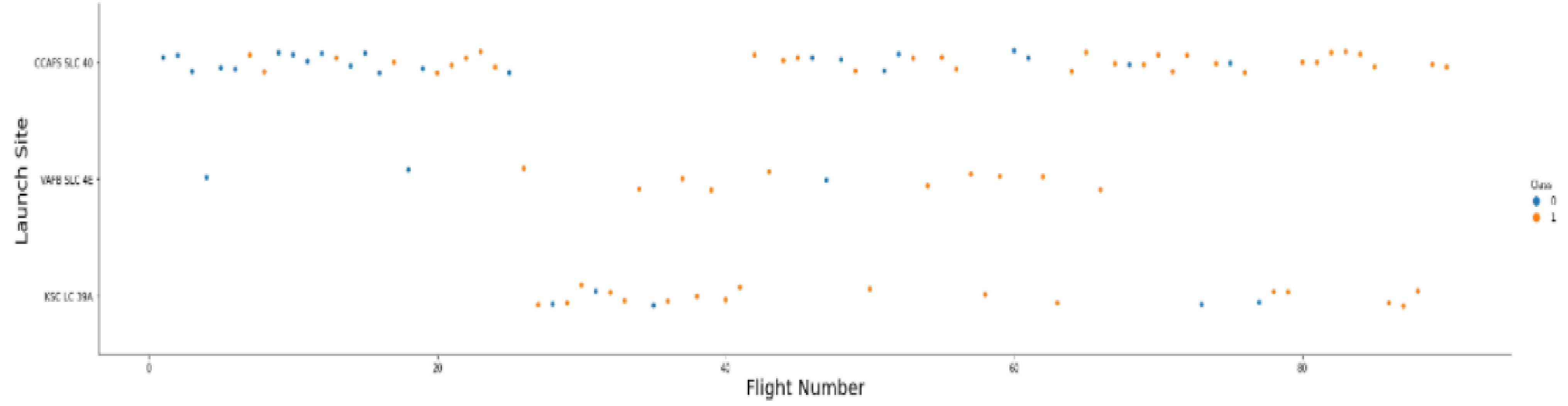
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

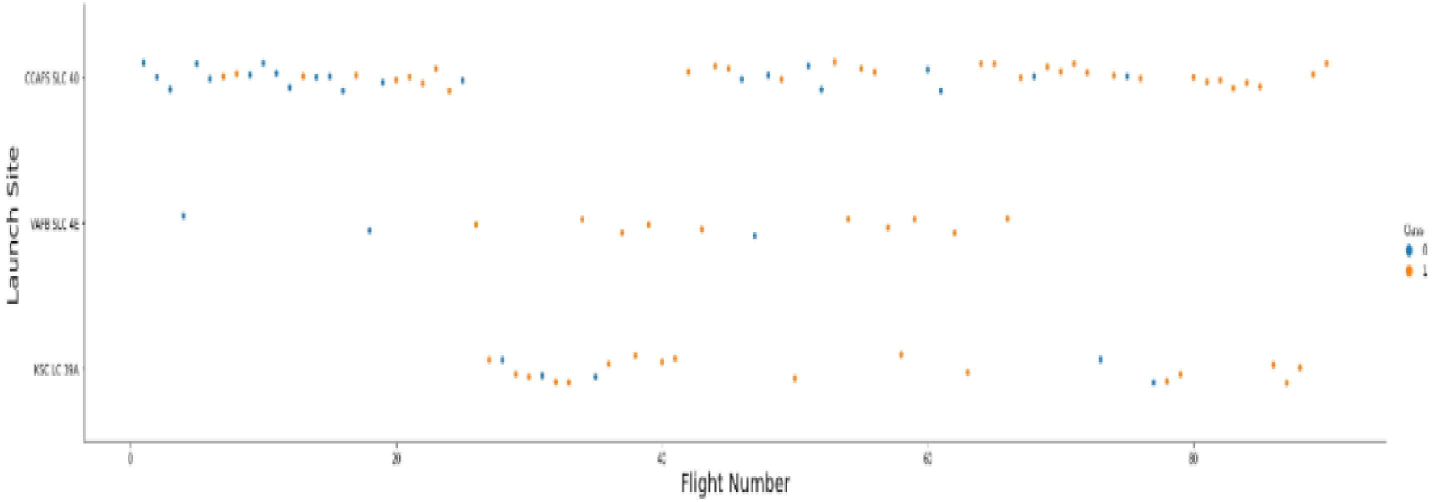
Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



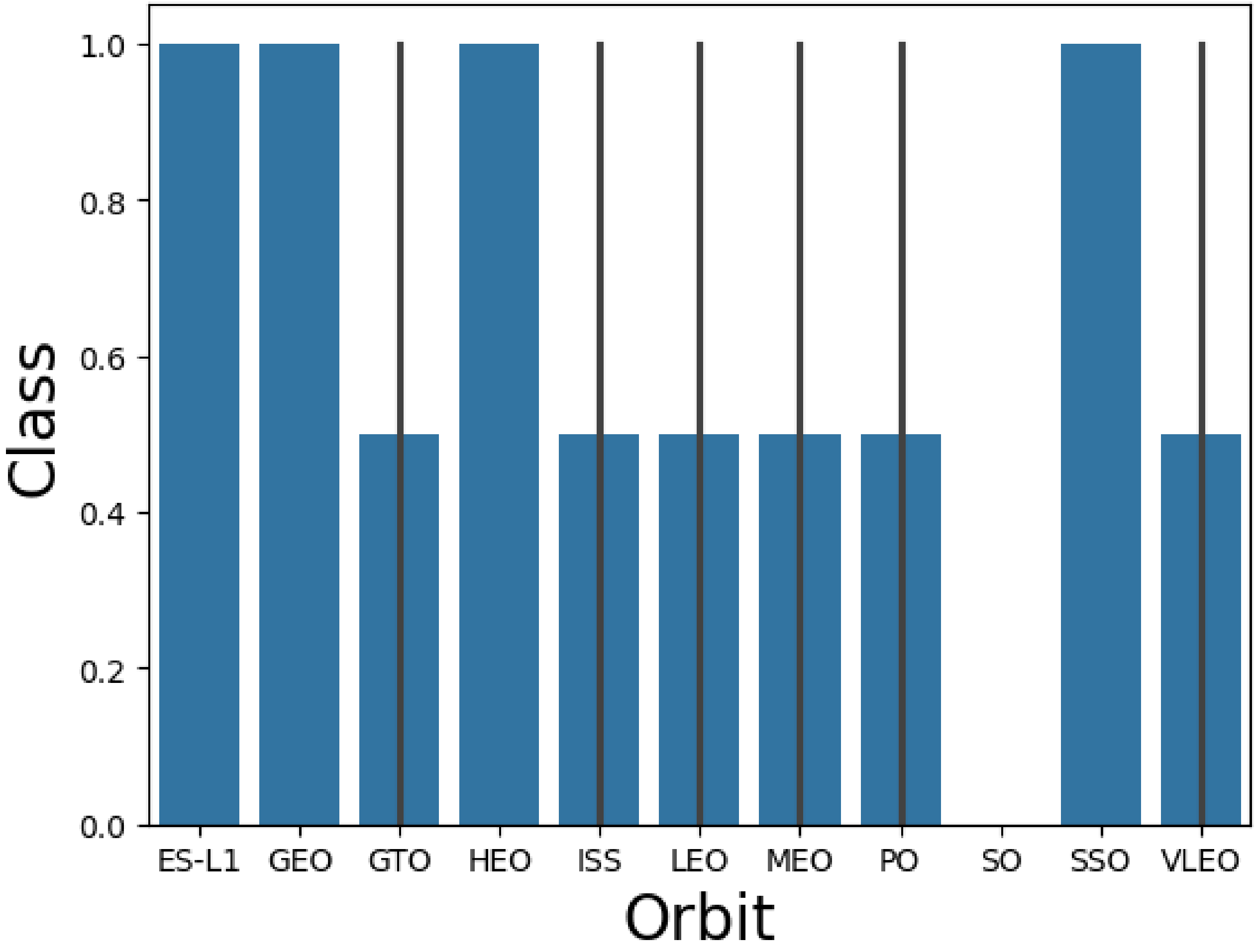
Payload vs. Launch Site

The greater the payload mass for the launch site CCAFC SLC 40 the higher the success rate for the rocket



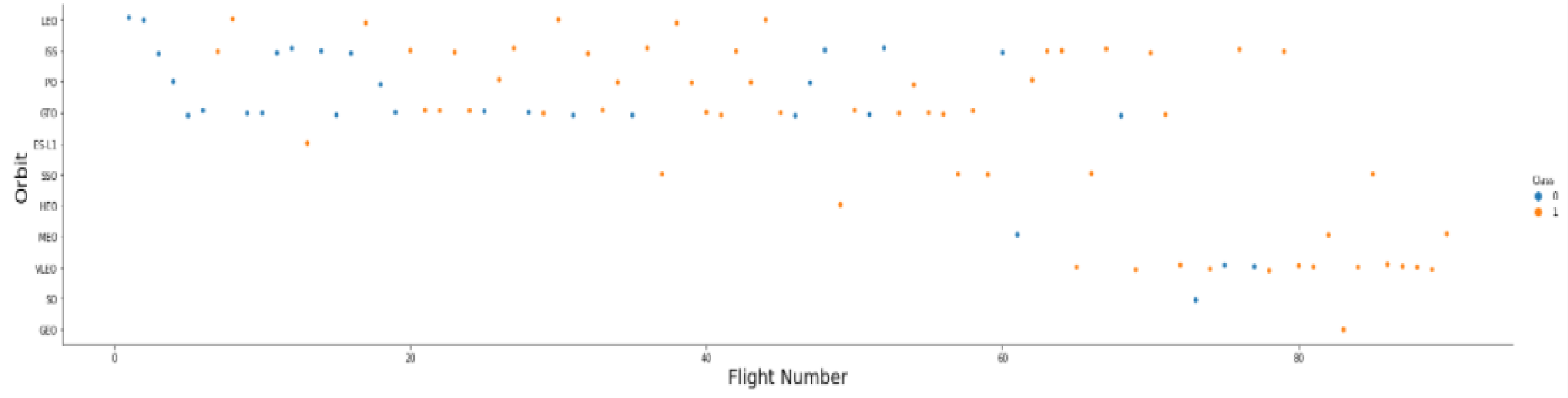
Success Rate vs. Orbit Type

- From the bar graph, it's evident that ES-L1, GEO, HEO and SSO orbits exhibited the highest success rates.
- This observation underscores the reliability and effectiveness of these orbits in facilitating successful launches, highlighting their significance in space mission planning and execution.



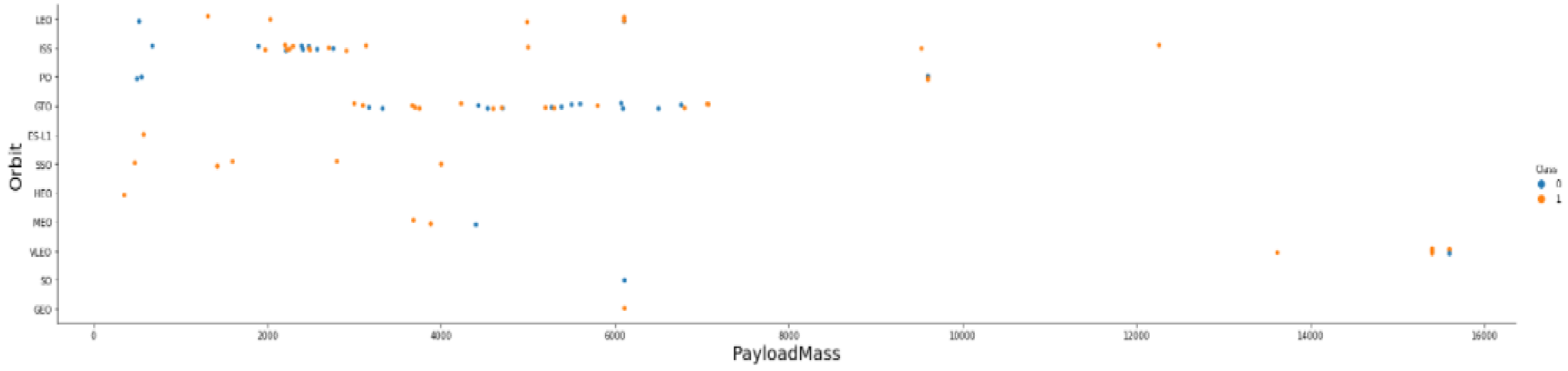
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



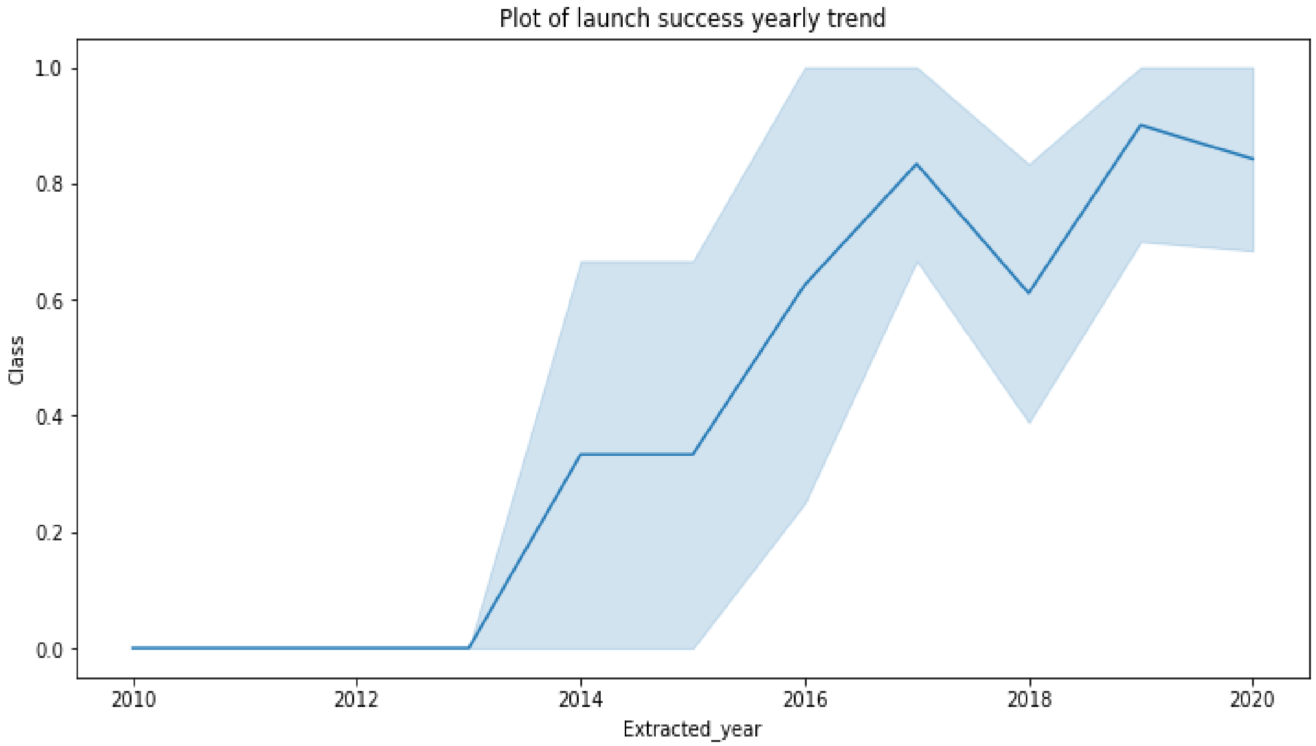
Payload vs. Orbit Type

- With heavy payloads, successful landings are more frequent in orbits such as PO, LEO, and ISS. This observation suggests a correlation between payload weight and successful landings in specific orbits, highlighting potential factors influencing successful landings in space missions.



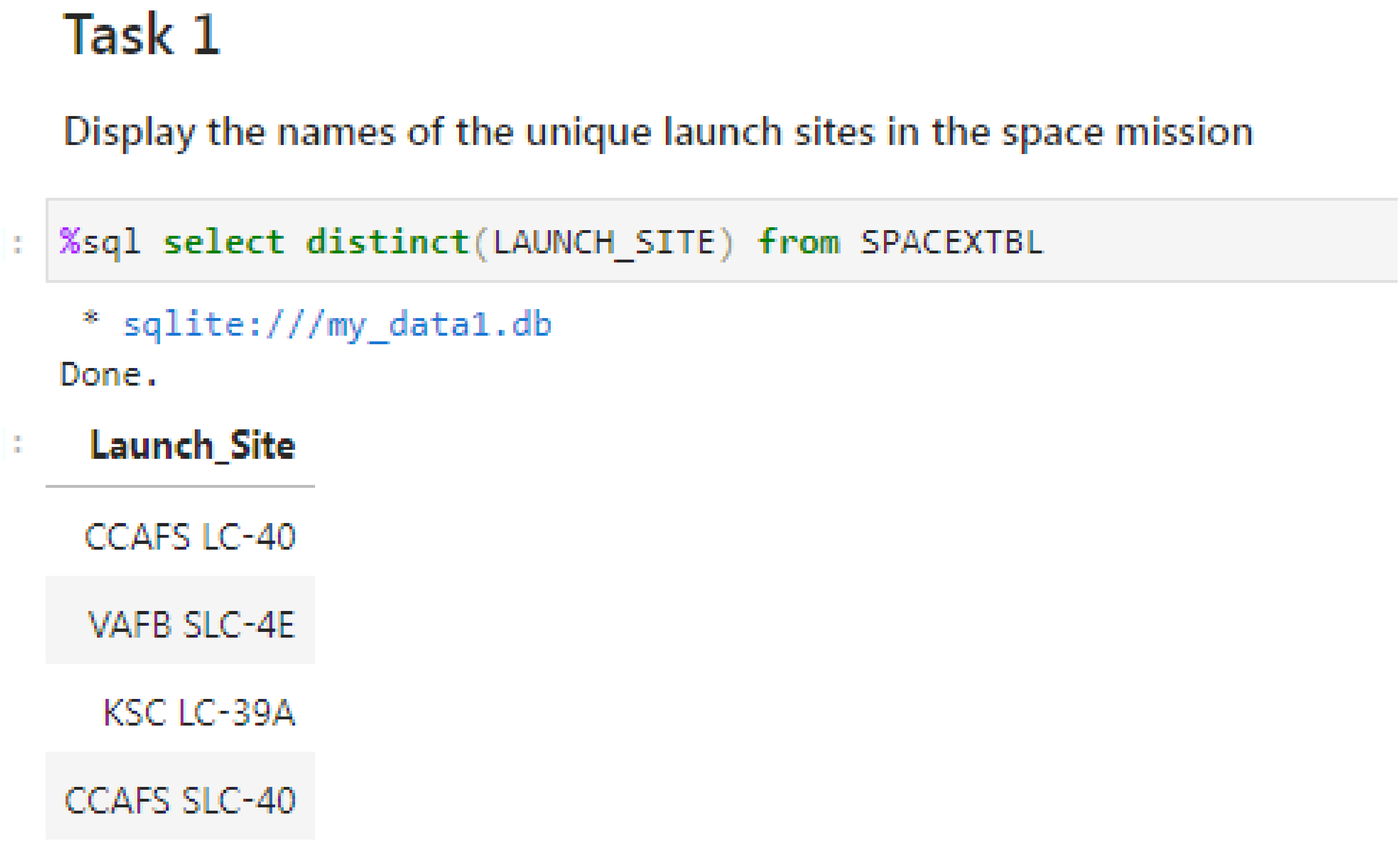
Launch Success Yearly Trend

- Based on the plot, we can see a continuous increase in success rates from 2013 to 2020.
- This trend indicates a positive trajectory in launch success over time, reflecting advancements and improvements in launch operations."



All Launch Site Names

- We applied the DISTINCT keyword to display unique launch sites exclusively from the SpaceX dataset.
- This approach helped us identify and focus solely on distinct launch sites, eliminating duplicate entries and streamlining our analysis of SpaceX's launch locations



Task 1

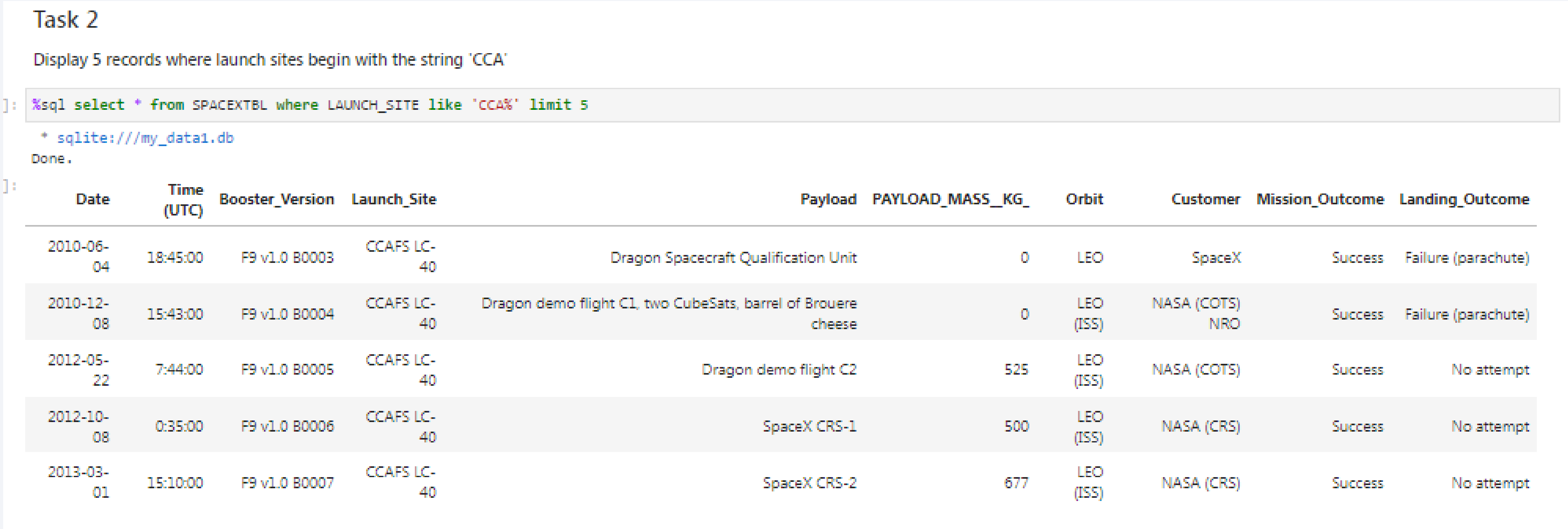
Display the names of the unique launch sites in the space mission

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

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

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

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
```

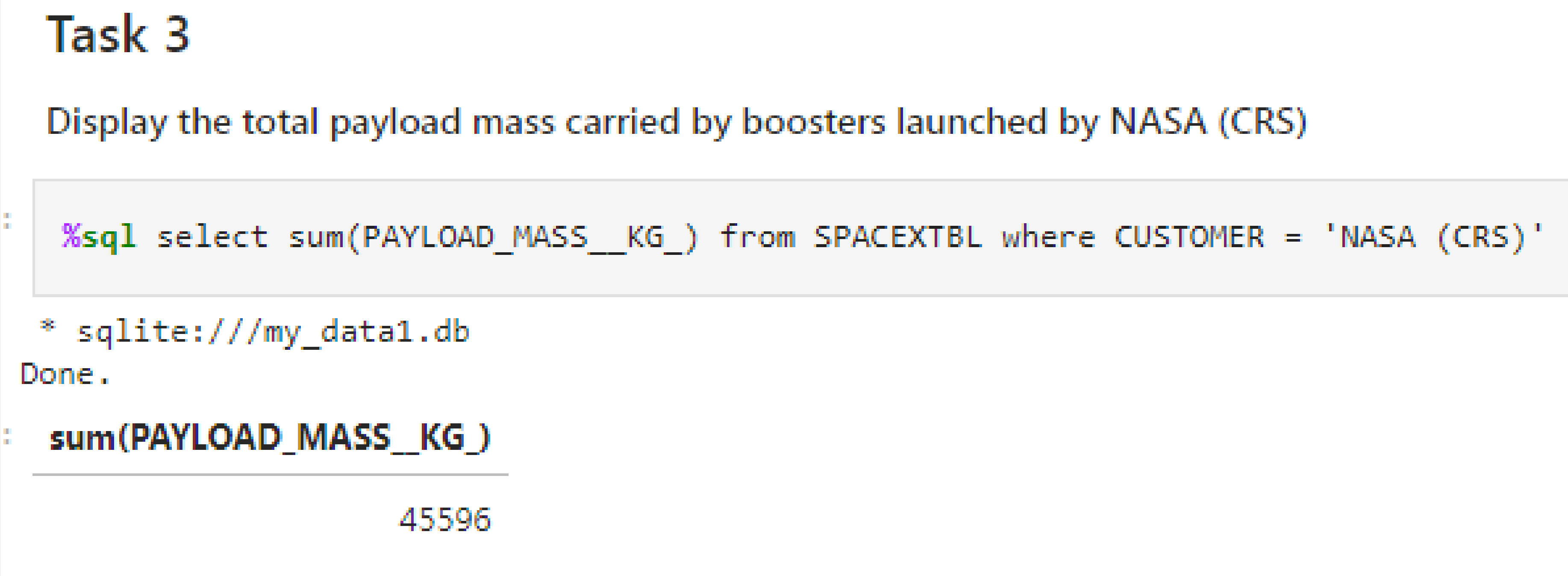
```
 * sqlite:///my_data1.db
Done.
```

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

- We utilized the previous query to showcase five records where launch sites start with 'CCA'. This targeted approach allowed us to specifically highlight instances related to launch sites with the specified prefix, providing focused insights into SpaceX's launch operations at these locations.

Total Payload Mass

- We 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)

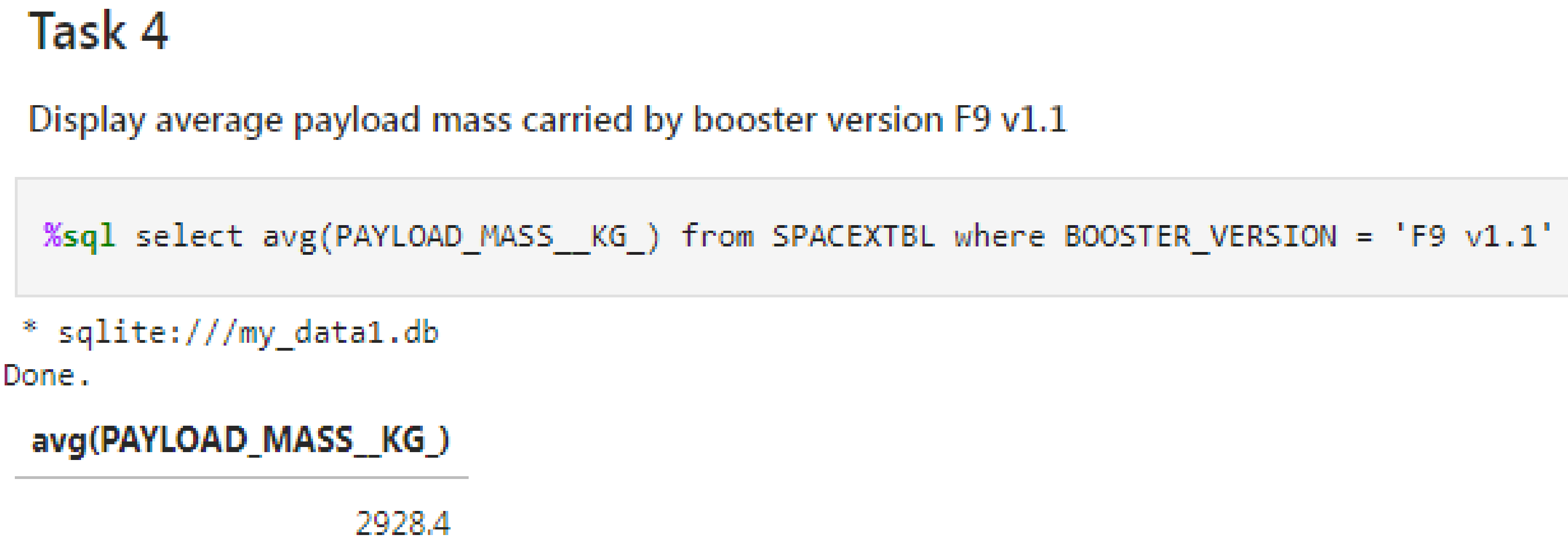
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We determined that the average payload mass carried by booster version F9 v1.1 was 2928.4. This calculation offers valuable insights into the performance capabilities of this specific booster version, aiding in our analysis of payload capacities across different iterations.



Task 4

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

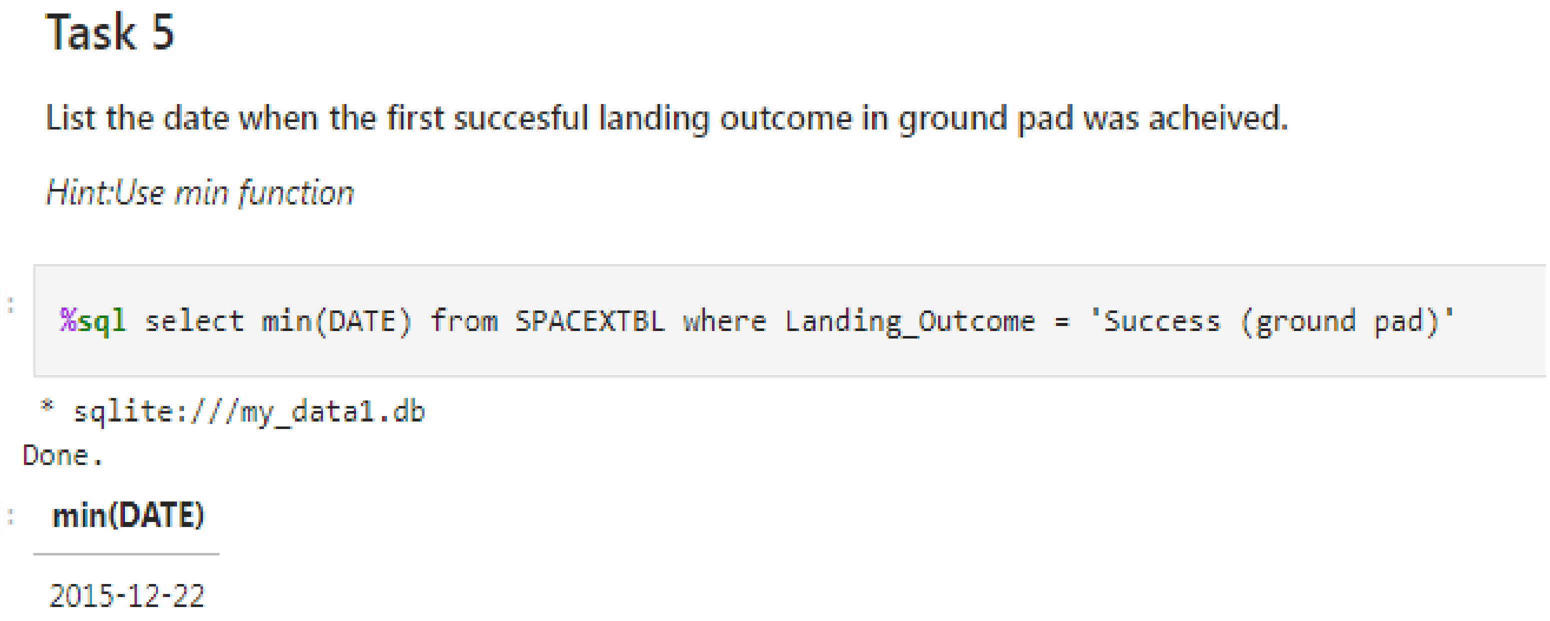
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- We noted that the first successful landing on a ground pad occurred on December 22, 2015.
- This milestone marked a significant achievement in our analysis, providing a key historical data point for understanding the progression of successful landings in our dataset.



Task 5

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

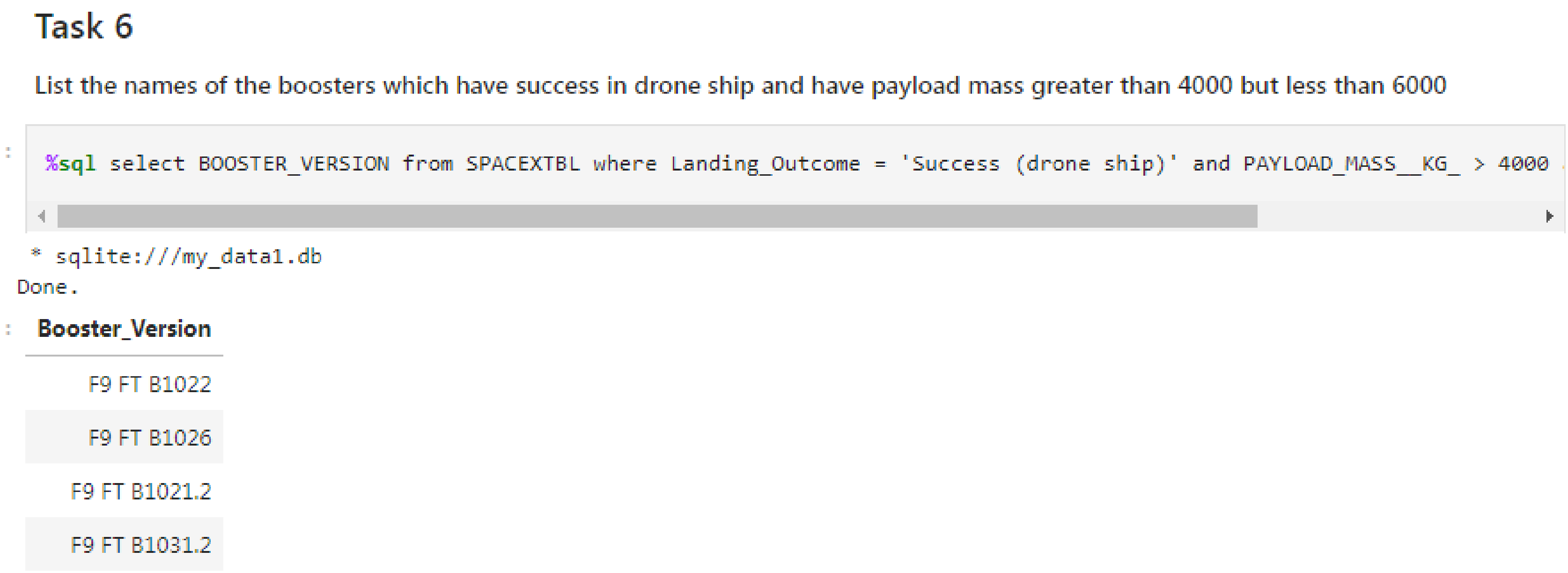
Hint:Use min function

```
%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 BOOSTER_VERSION from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000
```

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

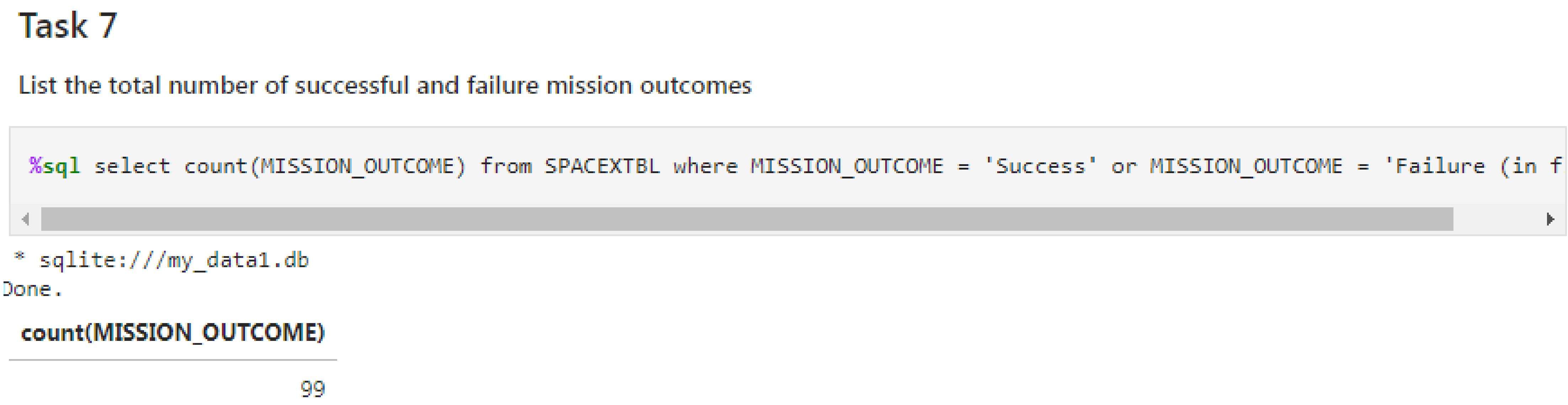
Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

- We utilized the WHERE clause to filter boosters that successfully landed on a drone ship. Applying the AND condition, we further narrowed down to identify successful landings with a payload mass exceeding 4000 but less than 6000.

  This approach enabled us to focus specifically on boosters meeting these criteria, facilitating a targeted analysis of successful landings within the specified payload range

Total Number of Successful and Failure Mission Outcomes

- We used wildcard like ‘%’ to filter for **WHERE** MissionOutcome was a success or a failure.



Task 7

List the total number of successful and failure mission outcomes

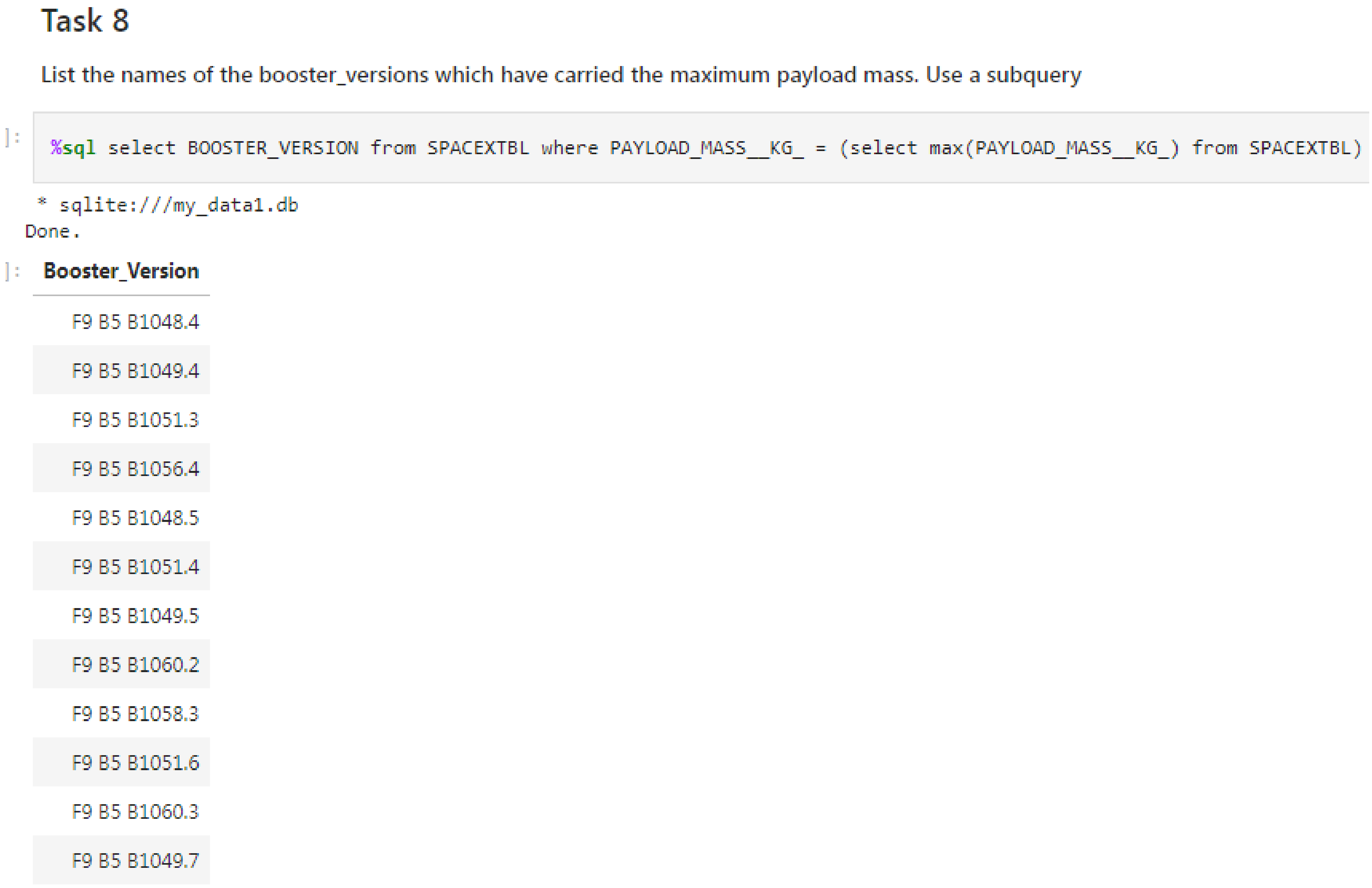
```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in f
```

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

count(MISSION_OUTCOME)
99

Boosters Carried Maximum Payload

- We used a subquery within the WHERE clause alongside the MAX() function to identify the booster that carried the maximum payload.
- This approach allowed us to pinpoint the specific booster with the highest payload capacity, streamlining our analysis and providing valuable insights into the performance capabilities of different boosters.



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)
```

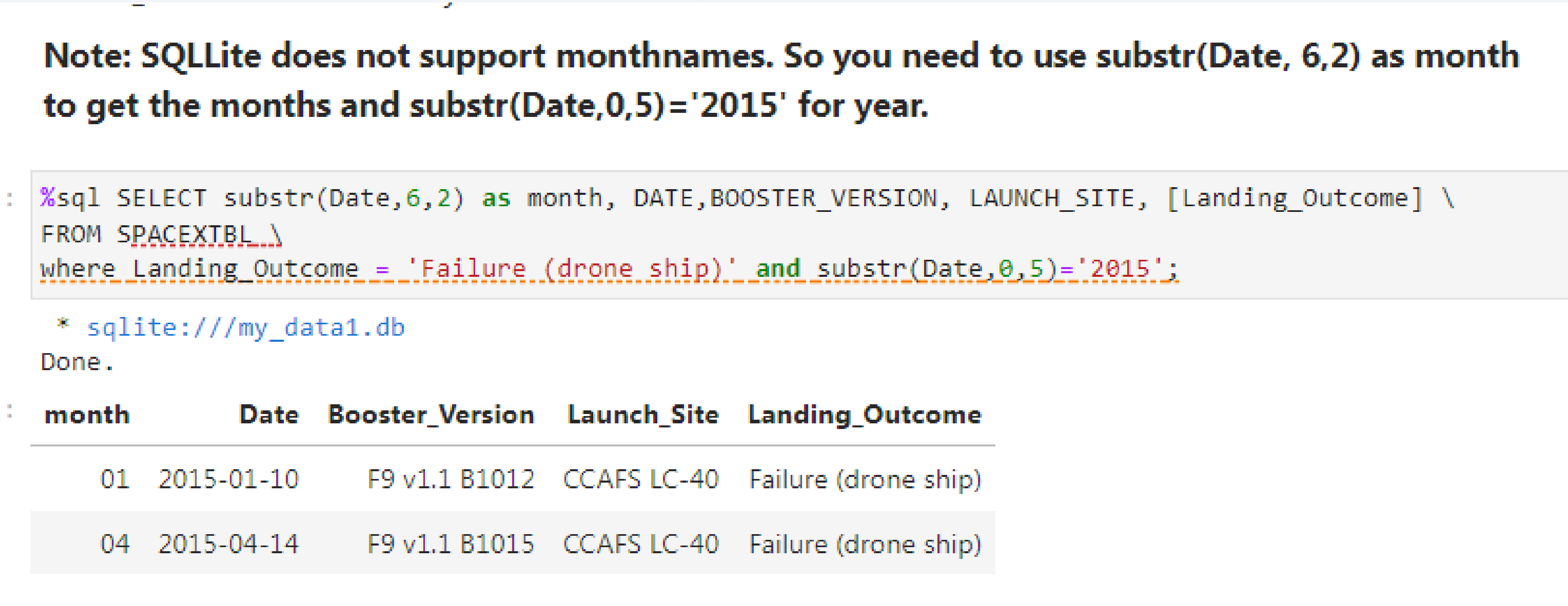
```
 * sqlite:///my_data1.db
Done.
```

]:

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

- We 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



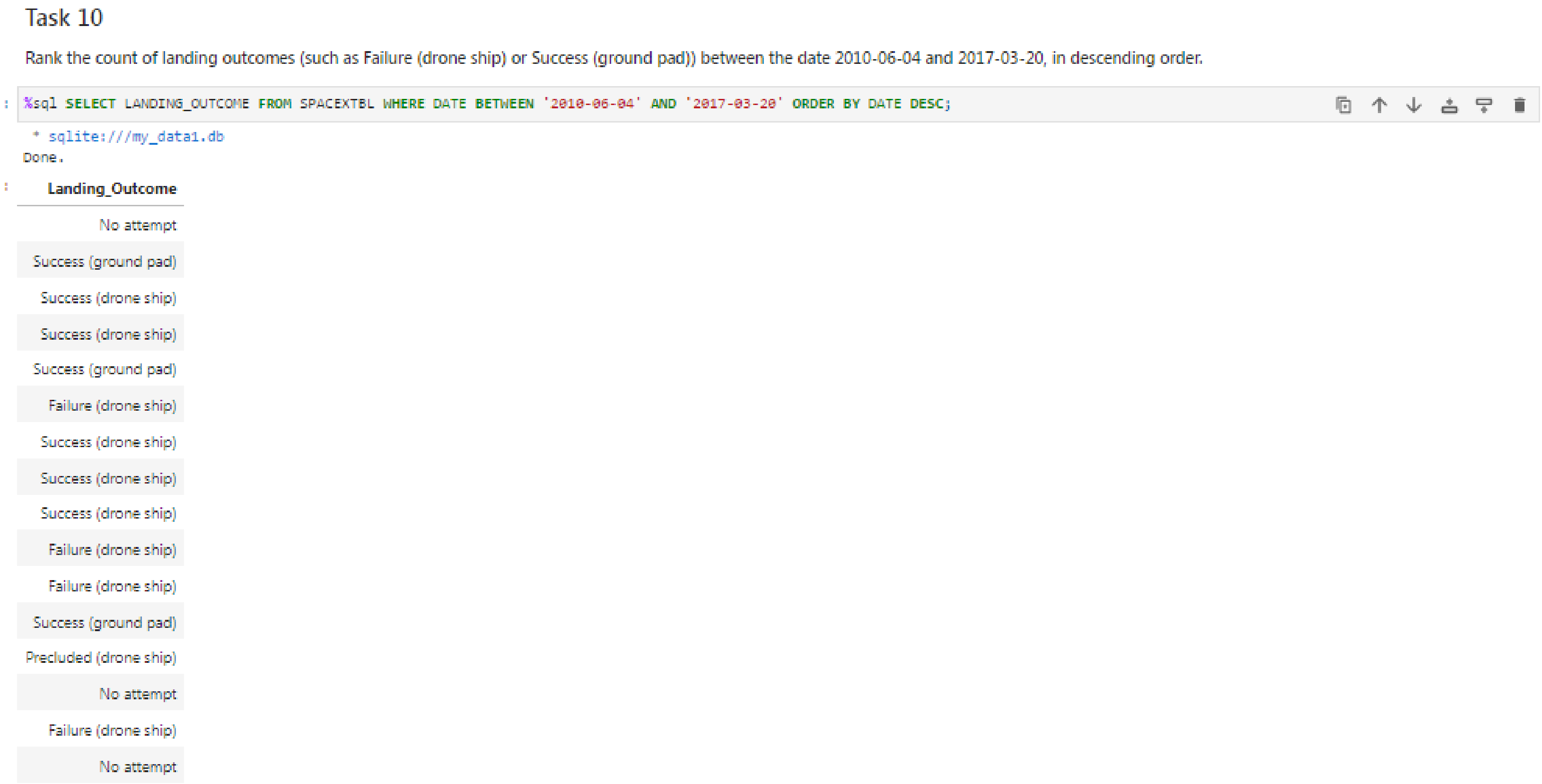
Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date,6,2) as month, DATE,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	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	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.

```
%sql SELECT LANDING_OUTCOME FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY DATE DESC;
```

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

Landing_Outcome
No attempt
Success (ground pad)
Success (drone ship)
Success (drone ship)
Success (ground pad)
Failure (drone ship)
Success (drone ship)
Success (drone ship)
Success (drone ship)
Failure (drone ship)
Failure (drone ship)
Success (ground pad)
Precluded (drone ship)
No attempt
Failure (drone ship)
No attempt

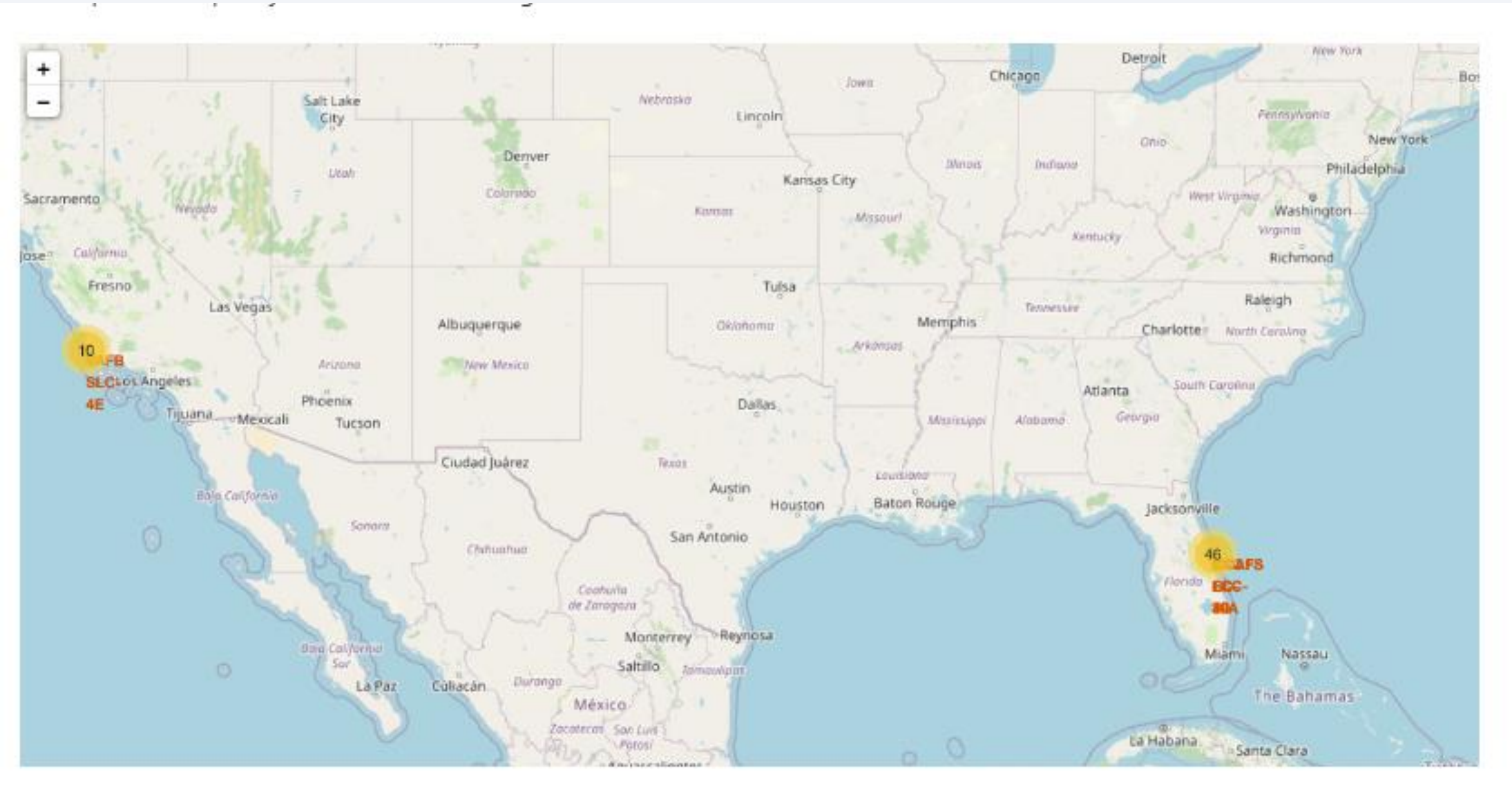
Selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.

We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

Section 4

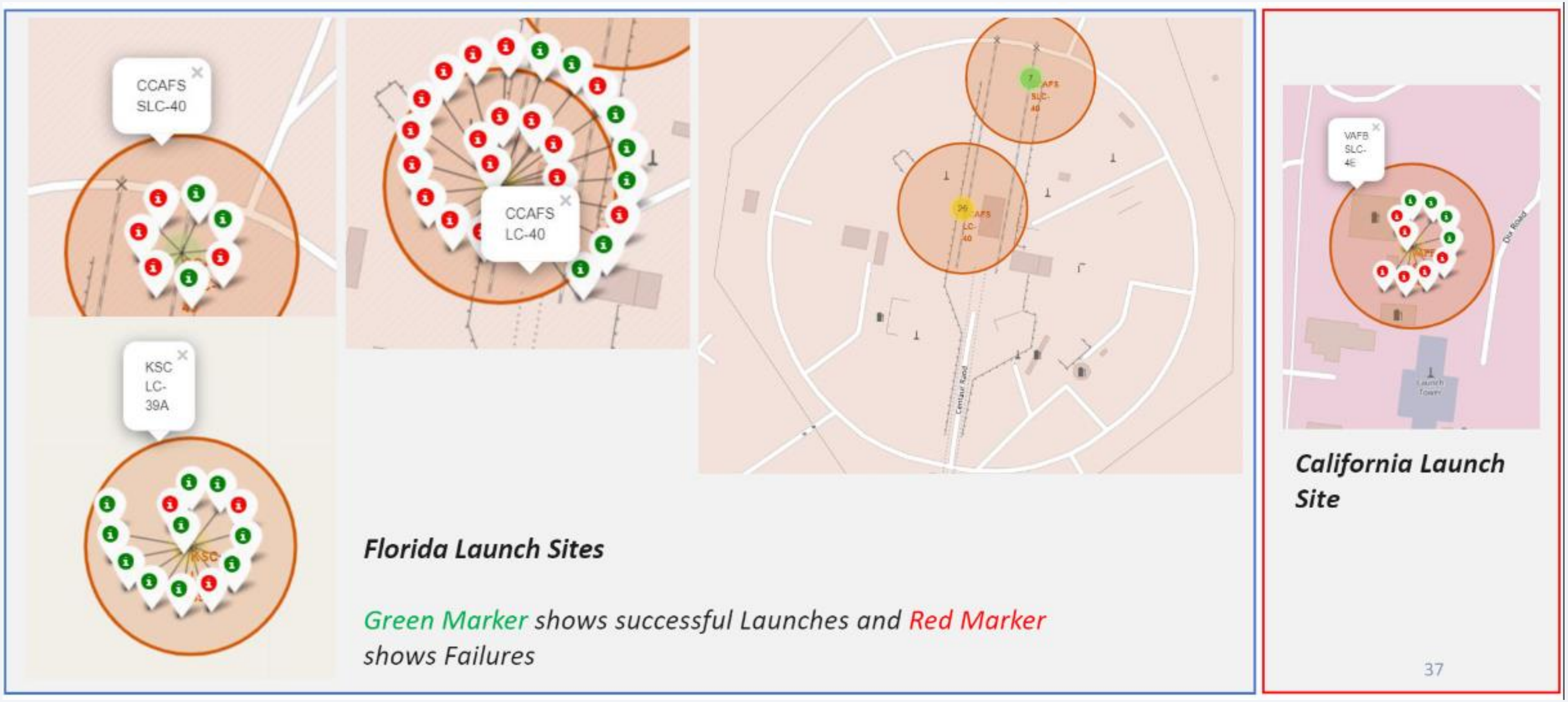
Launch Sites Proximities Analysis

All launch sites global map markers



SpaceX launch sites are in the United States of America coasts. Florida and California.

Markers showing launch sites with color labels

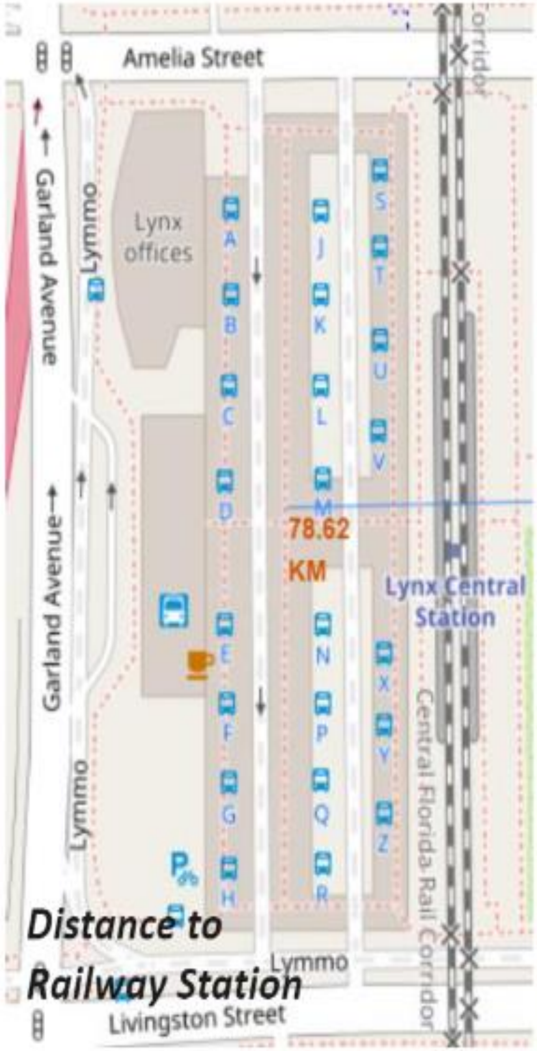


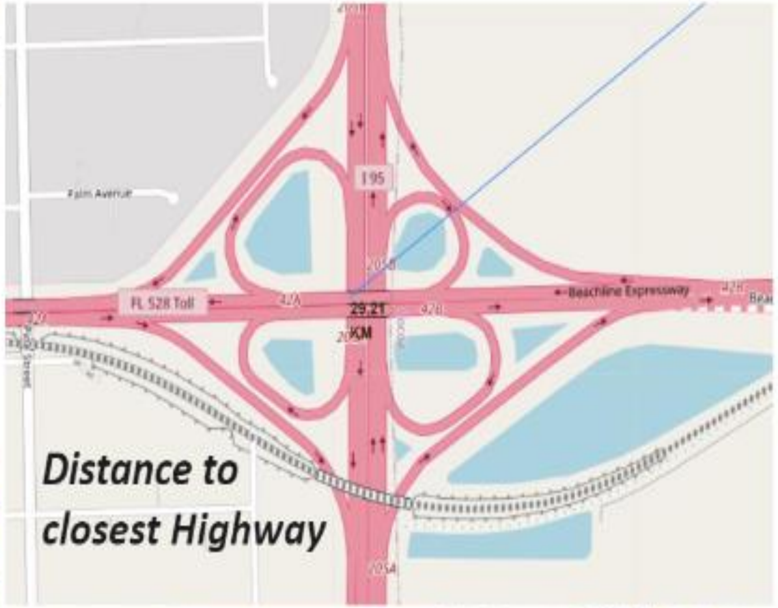
Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

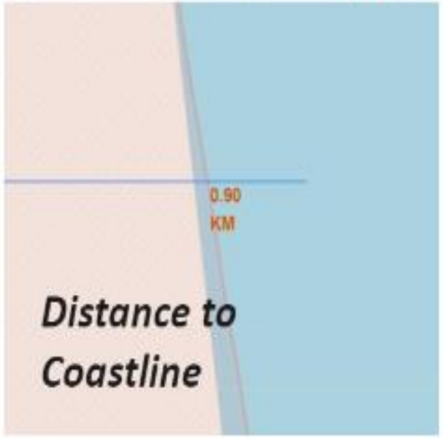
California Launch Site

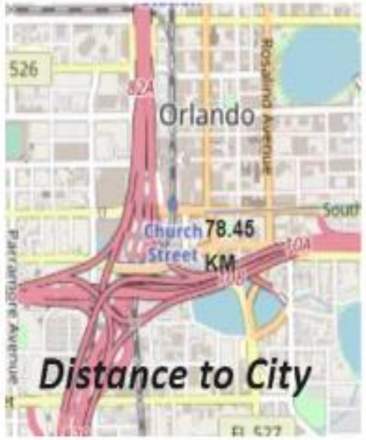
Launch Site distance to landmarks



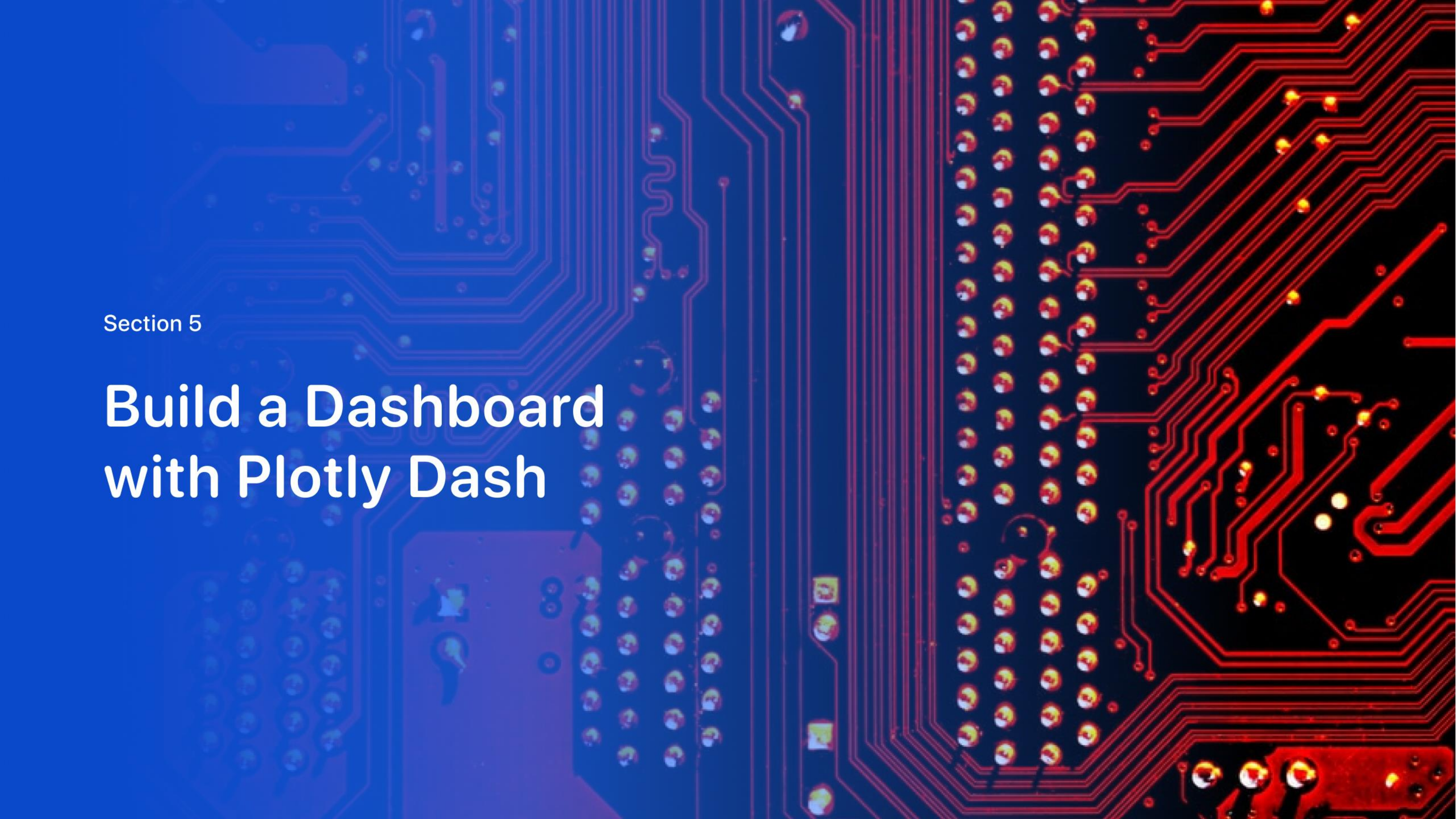


7
CCAFS
SLC-
40
26
CCAFS
LC-
40
Distance to coast





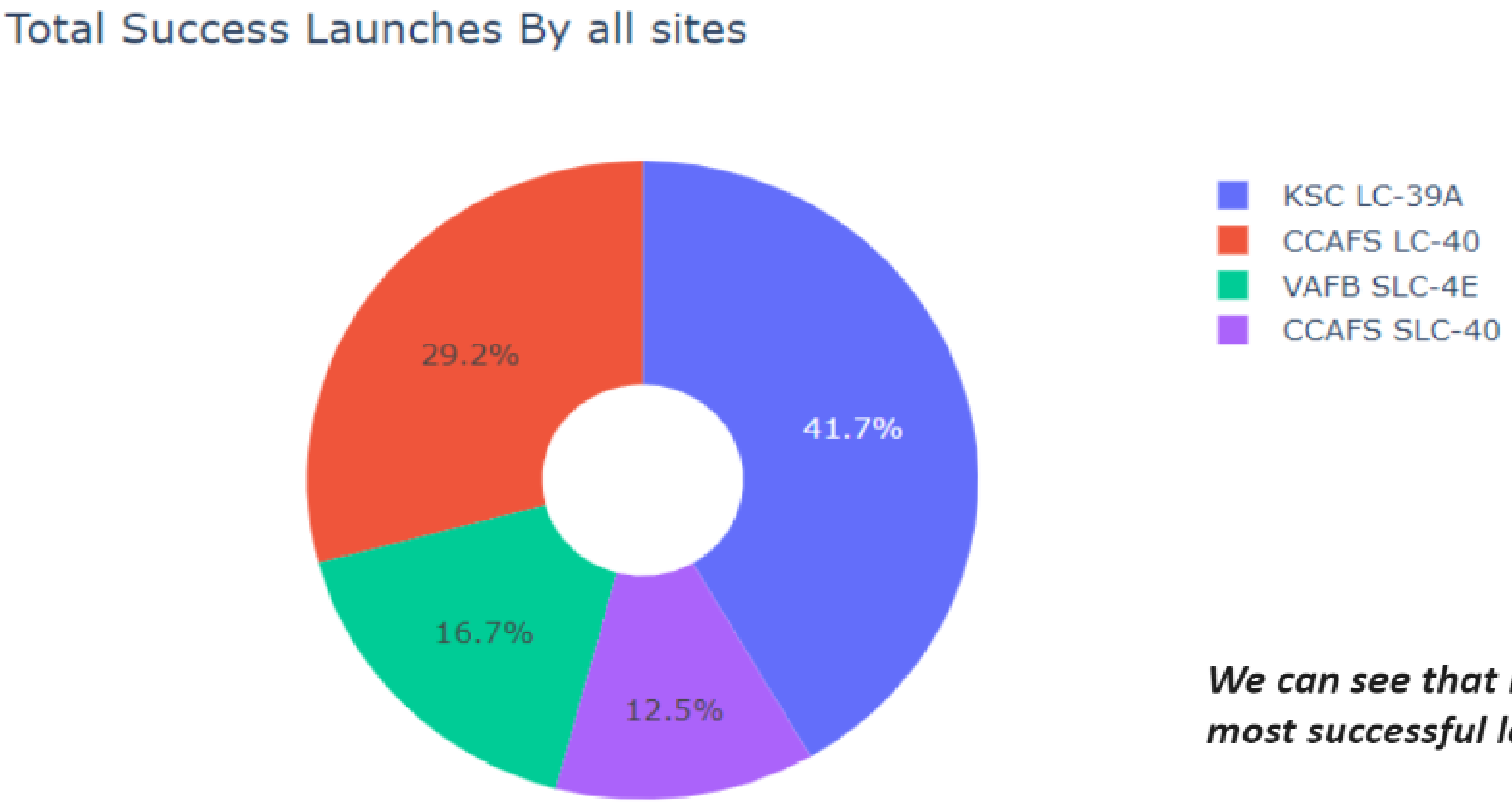
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes



Section 5

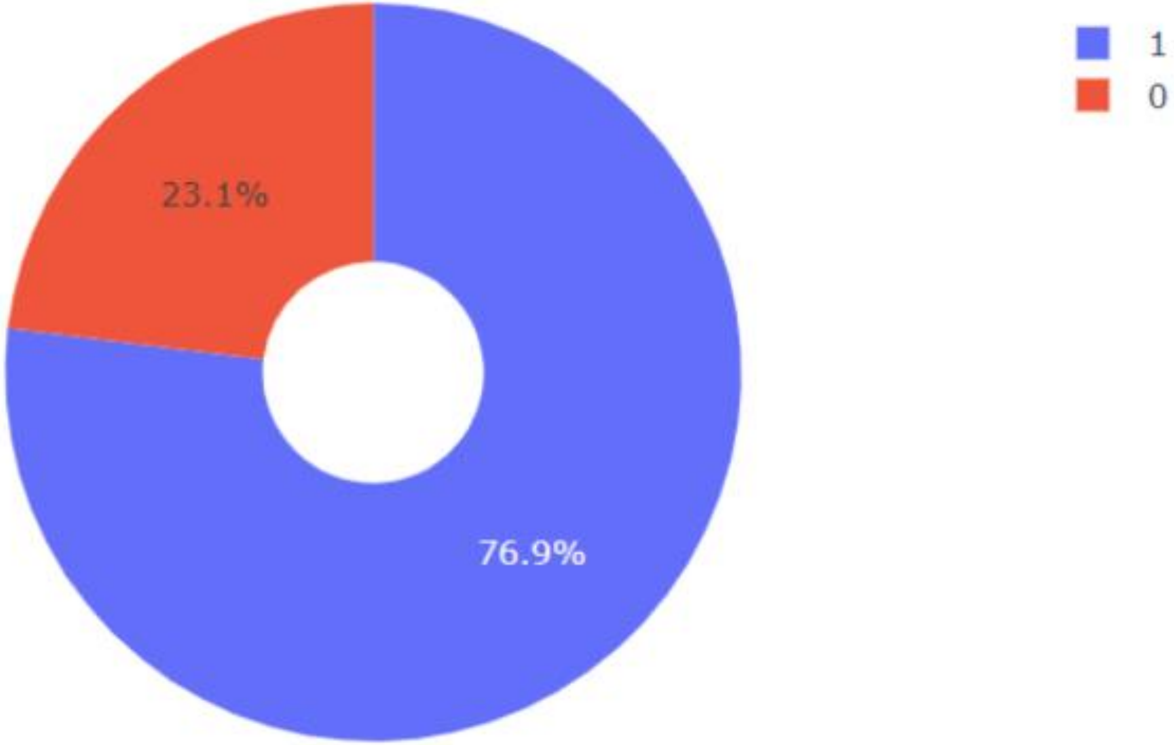
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



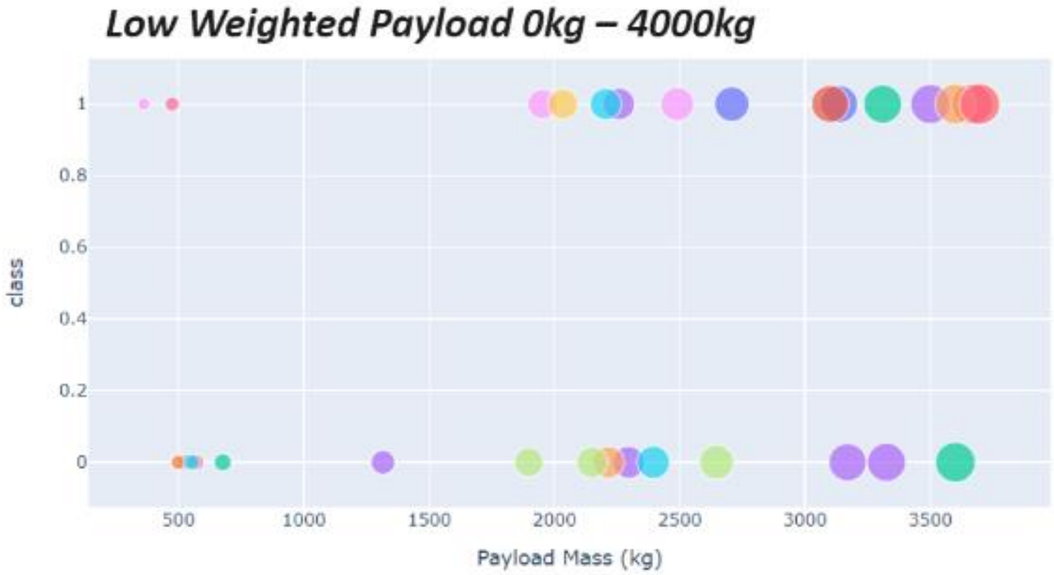
We can see that KSC LC-39A had the most successful launches from all the sites

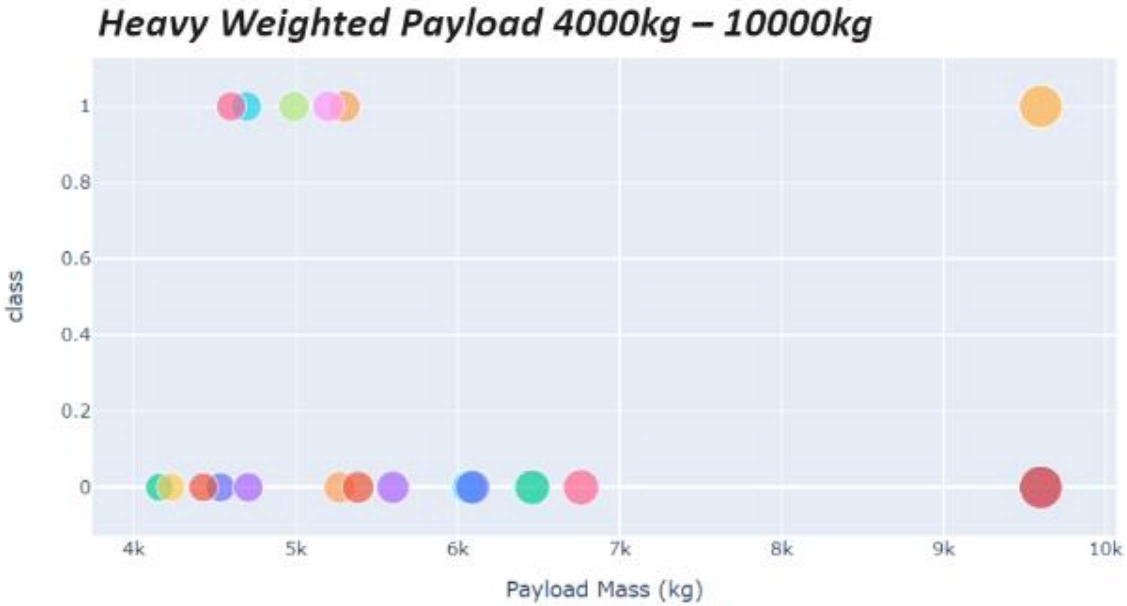
Pie chart showing the Launch site with the highest launch success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





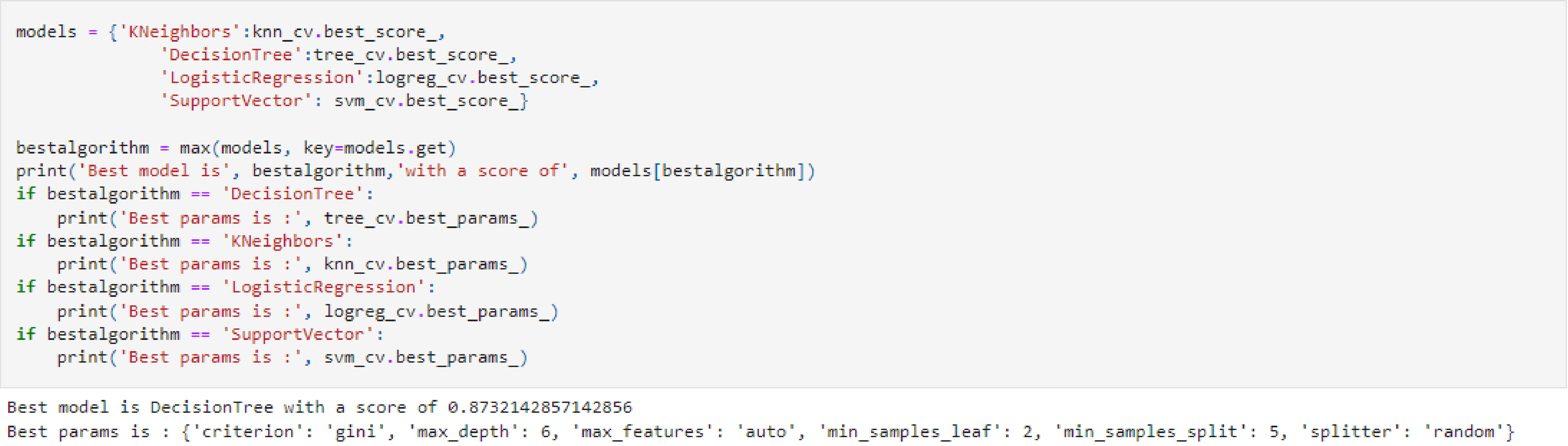
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy
- Best parameters are shown below the code.



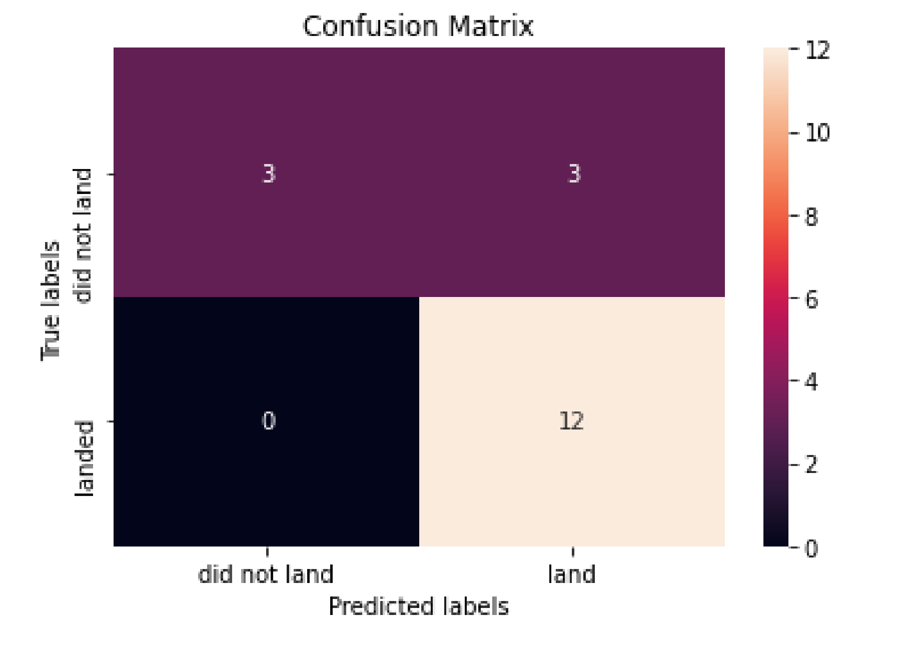
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

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 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix generated by the decision tree classifier demonstrates its capacity to distinguish between various classes effectively.
- However, a significant concern arises from false positives, where instances of unsuccessful landings are erroneously classified as successful by the classifier.
- This misclassification can lead to misleading results and impact the overall performance and reliability of the classifier's predictions, especially in scenarios where accurate distinction is critical



Conclusions

- Our analysis of space launch data reveals compelling insights into the dynamics of launch success rates and the effectiveness of machine learning algorithms in predicting these outcomes. One notable observation is the positive correlation between flight frequency and success rates at launch sites, particularly prominent from 2013 to 2020. This period saw an increase in the number of launches, accompanied by a corresponding rise in success rates. This correlation suggests that increased launch activity may contribute to enhanced operational efficiency and improved success outcomes.
- Among the various orbital trajectories, including ES-L1, GEO, HEO, SSO, and VLEO, we observed consistently high success rates. These orbits are critical for a range of missions, from Earth observation to communication and scientific exploration. Their reliability in facilitating successful launches underscores their importance in space mission planning and execution.
- In evaluating launch sites, KSC LC-39A emerged as a standout performer with the highest number of successful launches. This site's operational efficiency and track record of success make it a preferred choice for space missions, highlighting the significance of infrastructure and operational processes in achieving successful outcomes.

Conclusions - continues

- Our analysis also delved into the effectiveness of machine learning algorithms in predicting launch success probabilities. Among the algorithms evaluated, the Decision Tree classifier demonstrated exceptional performance and emerged as the most effective tool for this task. Its ability to analyze complex data sets and identify key factors contributing to launch success sets it apart as a reliable and robust solution.
- The Decision Tree classifier operates by recursively partitioning the data based on features that lead to the most significant information gain, resulting in a predictive model that accurately predicts launch outcomes. This capability makes it a valuable tool for informed decision-making in space launch operations, allowing stakeholders to assess and mitigate risks effectively.
- Furthermore, the classifier's ability to handle diverse data sets and adapt to evolving launch conditions enhances its utility in optimizing launch strategies. By identifying critical factors that influence launch success, such as weather conditions, technical readiness, and payload specifications, the classifier enables stakeholders to make informed decisions that maximize mission success rates.
- In conclusion, our analysis underscores the importance of data-driven insights and machine learning algorithms in space launch operations. The positive correlation between flight frequency and success rates, combined with the effectiveness of the Decision Tree classifier, highlights the potential for data-driven approaches to enhance operational efficiency, mitigate risks, and ensure mission success in the space industry

Thank you!