Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

- Summary of methodologies: Collected SpaceX launch data via REST API and Wikipedia scraping. Cleaned, filtered, and structured data using Python and Pandas. Performed Exploratory Data Analysis (EDA) with charts, SQL queries, and interactive visualizations using Folium and Plotly Dash. Built predictive models (Logistic Regression, SVM, KNN) to classify launch success.
- Summary of all results: Identified that KSC LC-39A is the most reliable launch site. Launch success improves with higher flight numbers and varies by payload mass and orbit type. Mid-range payloads with B4/B5 boosters have higher success rates, while heavy payloads with FT boosters are riskier. Predictive models achieved up to 83% accuracy in classifying launch outcomes.

Introduction

- SpaceX has revolutionized the space industry with reusable rockets and frequent launches. Understanding the historical performance of Falcon 9 rockets, payload outcomes, and launch site reliability can provide insights for predicting future launch success. The dataset includes detailed information on rockets, payloads, launch sites, core stages, and mission outcomes, collected via API requests and web scraping, then structured into Pandas DataFrames for analysis.
- Problems I want to find answers:

•How do launch site, rocket type, and payload mass affect launch success?
•Which launch sites and boosters are most reliable?
•How has the success rate evolved over time?
•Can predictive models accurately classify future launch outcomes based on historical data?
•More and more…

Section 1

Methodology

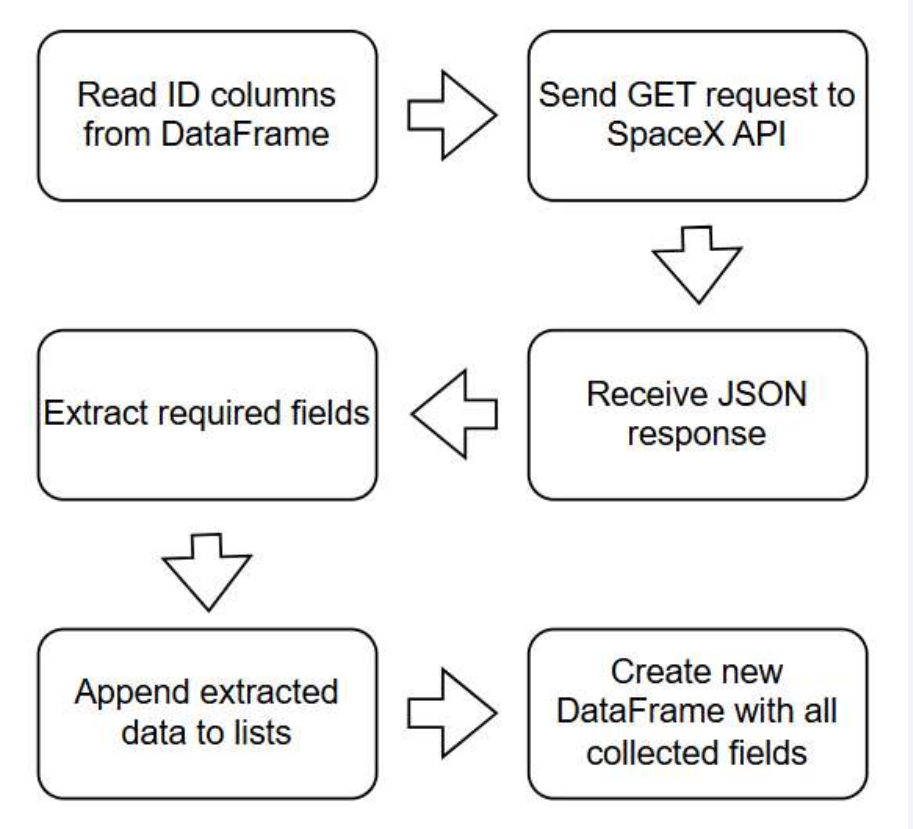
Methodology

Executive Summary

- Data collection methodology:
  - Data was gathered from the SpaceX API and supplemented with Wikipedia scraping.
- Perform data wrangling
  - Irrelevant columns were removed, and only Falcon launches were retained.
- 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

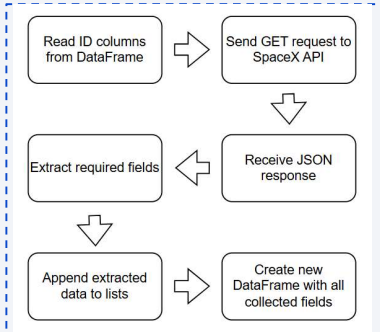
- The dataset was collected by reading the ID columns from the DataFrame, sending GET requests to the SpaceX API, and receiving JSON responses. Required fields were extracted from the responses and appended to lists, which were then combined to create a new DataFrame containing all collected data.



Data Collection – SpaceX API

- I retrieved rocket, launch site, payload, and core details from the official SpaceX REST API.
- I used the Python requests library to send GET requests to different API endpoints.
- Then, I processed and parsed the returned JSON objects and extracted the relevant fields into structured Python lists.
- Finally, I converted these lists into Pandas DataFrames for further analysis.

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
# Use json_normalize meethod to convert the json result into a dataframe
resp = response.json()
data = pd.json_normalize(resp)
```

Using the dataframe `data` print the first 5 rows

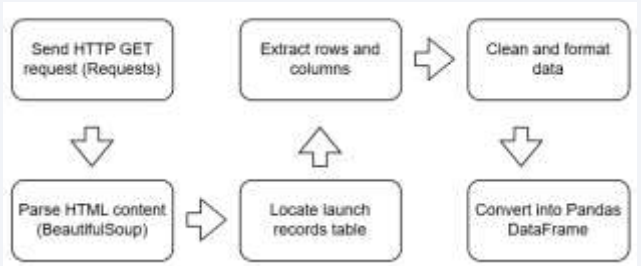
```
# Get the head of the dataframe
data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	payloads	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb0006eeb1e1]	5e9e4502f

Data Collection - Scraping

- First, I sent an HTTP GET request to the Falcon 9 Wikipedia page using the Requests library.
- Then, I parsed the HTML content with BeautifulSoup. I located the table containing Falcon 9 and Falcon Heavy launch records.
- From this table, I extracted rows and columns such as date, payload, orbit, and outcome.
- After cleaning and structuring the data, I stored it into Python lists.
- Finally, I converted these lists into a Pandas DataFrame.

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose



Next, request the HTML page from the above URL and get a `response` object

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

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

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(resp.content, 'html.parser')
```

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

```
# use soup.title attribute
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')
```

Starting from the third table is our target table contains the actual launch records.

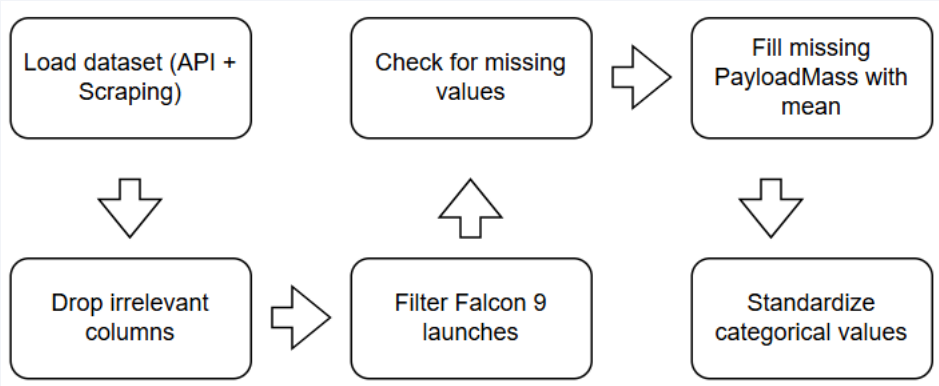
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
```

Data Wrangling

- First I loaded the data collected from the SpaceX API and Wikipedia scraping. Then I removed irrelevant columns such as links and extra metadata that were not useful for analysis. I also filtered the dataset to include only Falcon 9 launches removing the Falcon 1 records.

- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose



EDA with Data Visualization

- In the exploratory data analysis stage, I used different charts to uncover patterns and insights. Scatter plots were used to analyze relationships such as Flight Number vs. Launch Site and Payload Mass vs. Orbit, helping identify success trends. Bar charts allowed me to compare success rates across orbit types, while line charts revealed how the average launch success rate changed over the years. These visualizations helped me clearly understand correlations, performance, and trends in the data.

- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose

EDA with SQL

•I calculated the total payload mass carried by NASA (CRS).

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload", Customer FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

•I found the date of the first successful ground pad landing.

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

Hint:Use min function

```
%sql SELECT MIN(Date) AS "Successful Ground Pad Landing" FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

•I retrieved the boosters that had successful drone ship landings with payloads between 4000 and 6000 kg.

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 DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

•I counted and compared the total number of successful and failed mission outcomes.

Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.

```
%sql SELECT "Landing _Outcome", COUNT(*) as Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY "Landing _Outcome" ORDER BY Outcome_Count DESC;
```

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose :

Build an Interactive Map with Folium

- I created an interactive map using Folium to visualize all the launch sites. I added circle markers to highlight each site’s location and popup labels to display the site names. In addition, I included markers and text labels directly on the map to make the sites easier to identify without clicking. These map objects help in visually distinguishing between sites and understanding their geographical distribution.

- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

Build a Dashboard with Plotly Dash

- Summarize This dashboard shows SpaceX launch data with interactive charts. I included:
- **Launch Success Counts by Site** : a pie chart to compare success across launch sites.
- **Launch Success vs. Failure at KSC LC-39A** : a pie chart to see the success rate at the main site.
- **Payload Mass vs. Launch Success** : a scatter plot to explore how payload and booster type affect outcomes.
- These plots let users quickly understand which sites and boosters are most successful, and how payload size impacts launch risk. Interactions make it easy to filter and explore the data.

- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- I built and tested several models (Logistic Regression, SVM, Decision Tree, KNN), evaluated their accuracy, and compared results. Through this process, I identified Logistic Regression, SVM, and KNN as the best-performing models. The workflow is summarized with key steps and

Method	Conclusion
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.5
KNN	0.833333

- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

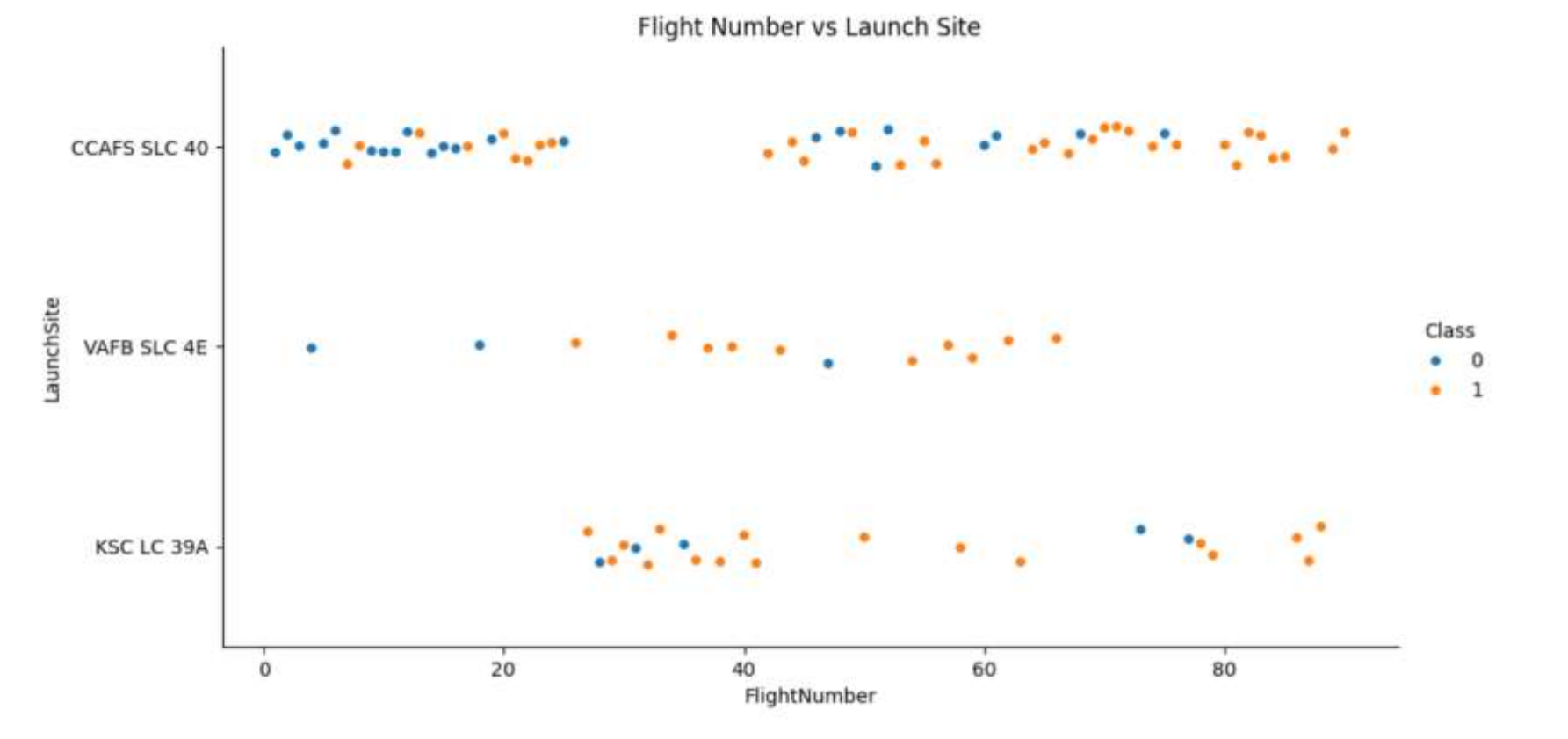
Results

- The results include key findings from exploratory data analysis, screenshots from the interactive analytics dashboard, and performance outcomes from predictive classification models. Together, these show both insights from the data and the effectiveness of the models.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

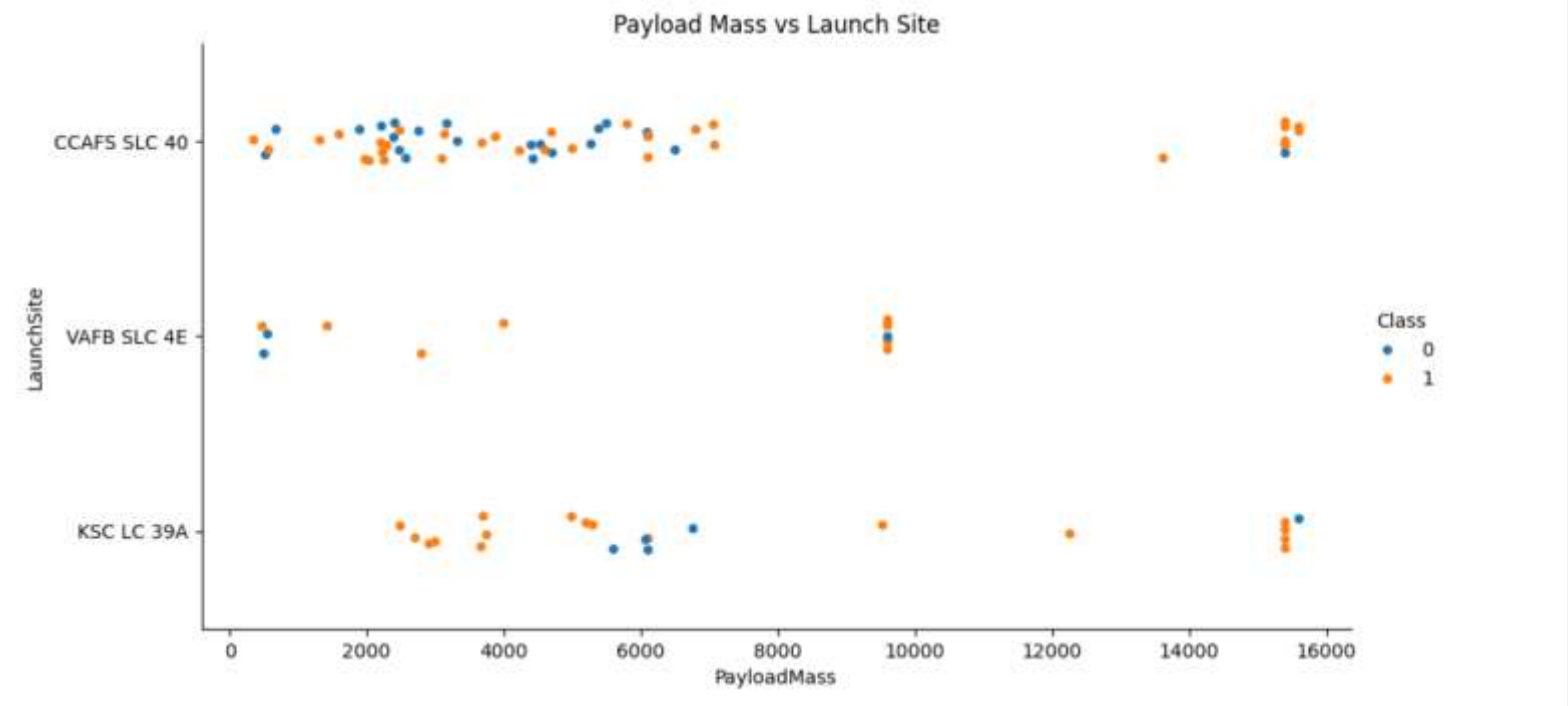
- Show a scatter plot of Flight Number vs. Launch Site



- This scatter plot compares flight numbers with launch sites, showing landing outcomes in blue (unsuccessful) and orange (successful). It highlights how success rates vary across sites and improve with increasing flight numbers.

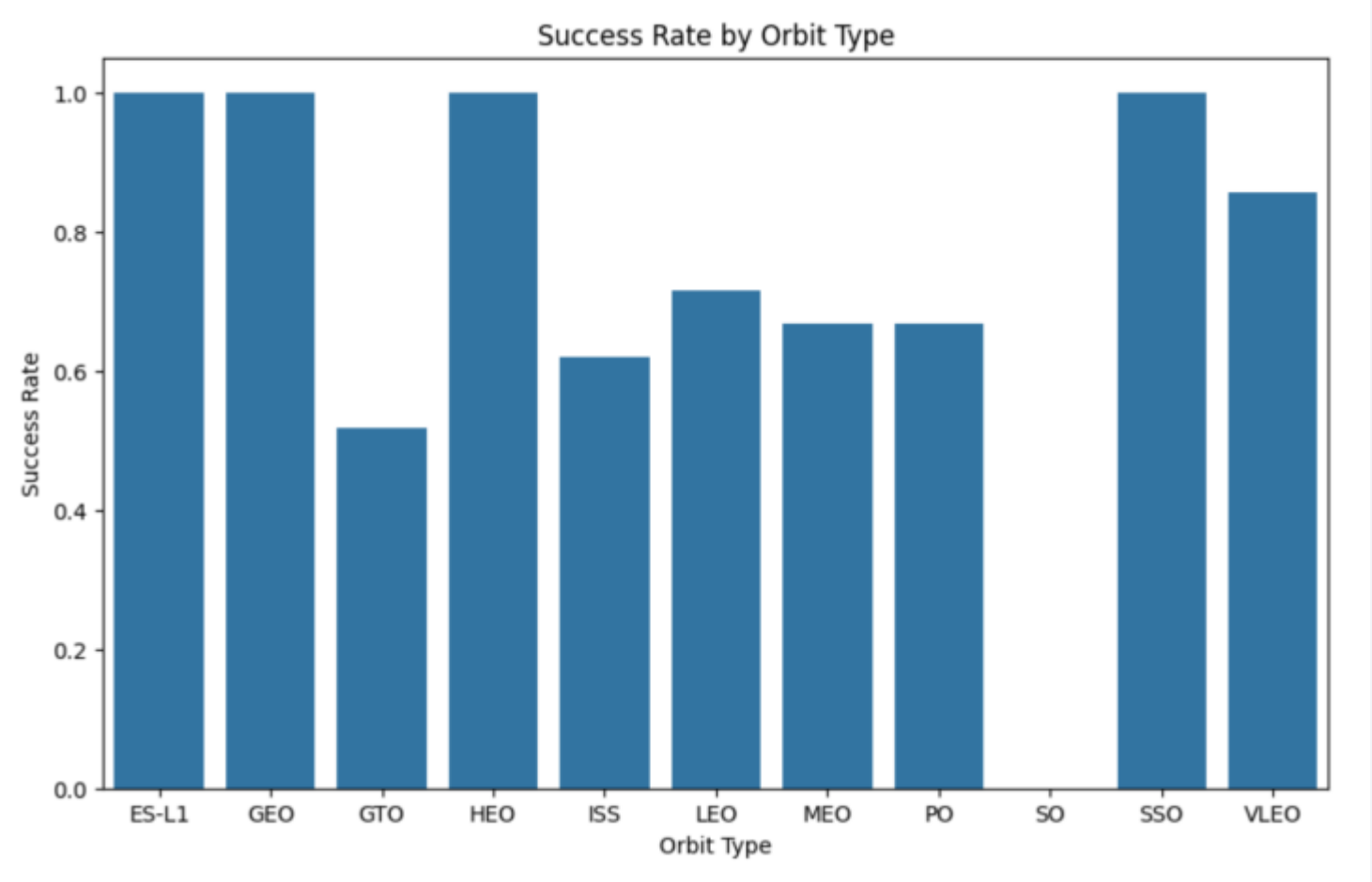
Payload vs. Launch Site

- This scatter plot shows the relationship between Payload Mass and Launch Site, with each point representing a flight outcome. The x-axis indicates payload mass, while the y-axis shows the launch sites. Blue dots (0) mark unsuccessful landings, and orange dots (1) mark successful ones. The chart highlights how landing success varies with payload size across different launch sites.



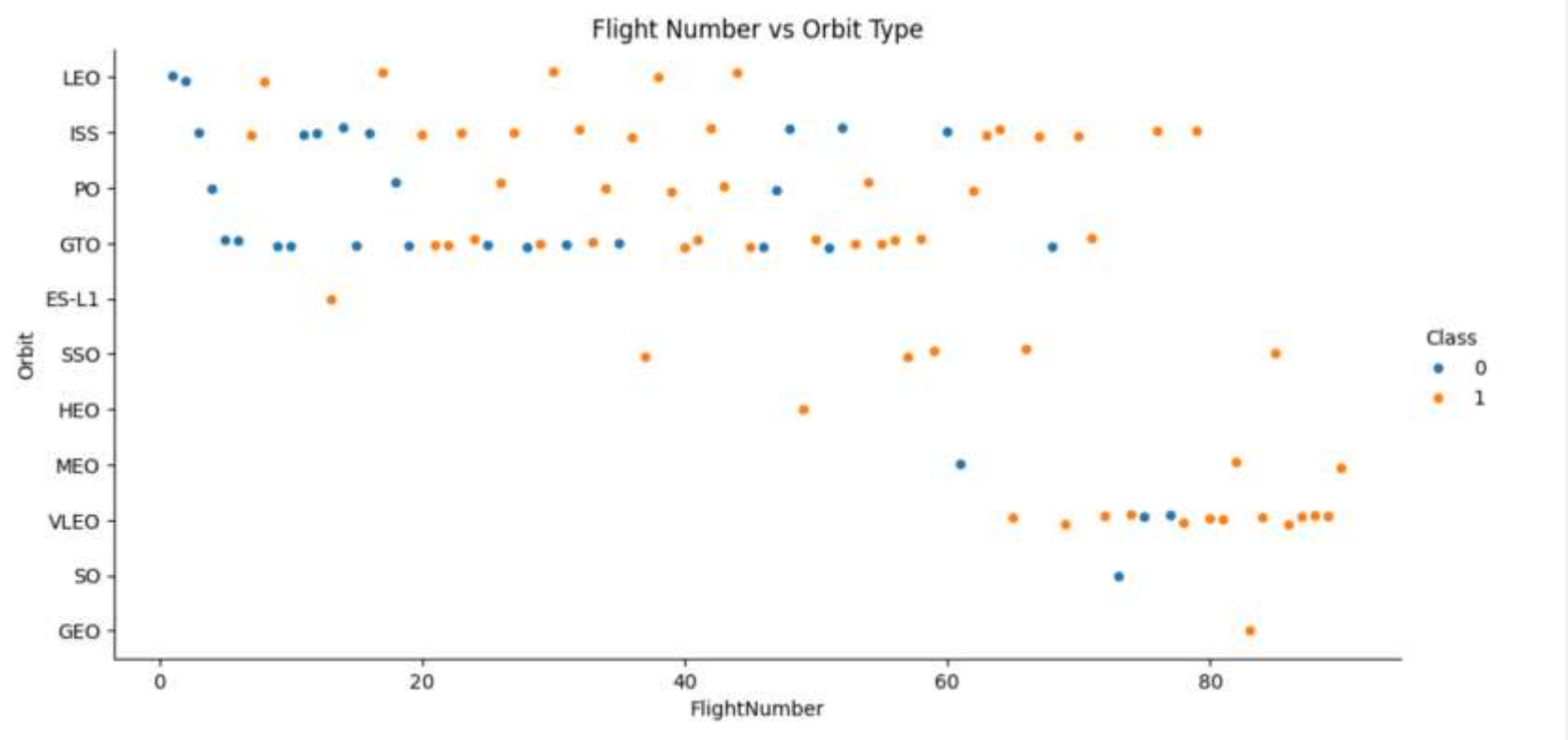
Success Rate vs. Orbit Type

- This is the bar chart for Success Rate vs. Orbit Type. It shows the success rate for each orbit type, with ES-L1, GEO, HEO, and SSO having the highest success rate (1.0), while GTO and SO has the lowest success rate among them.



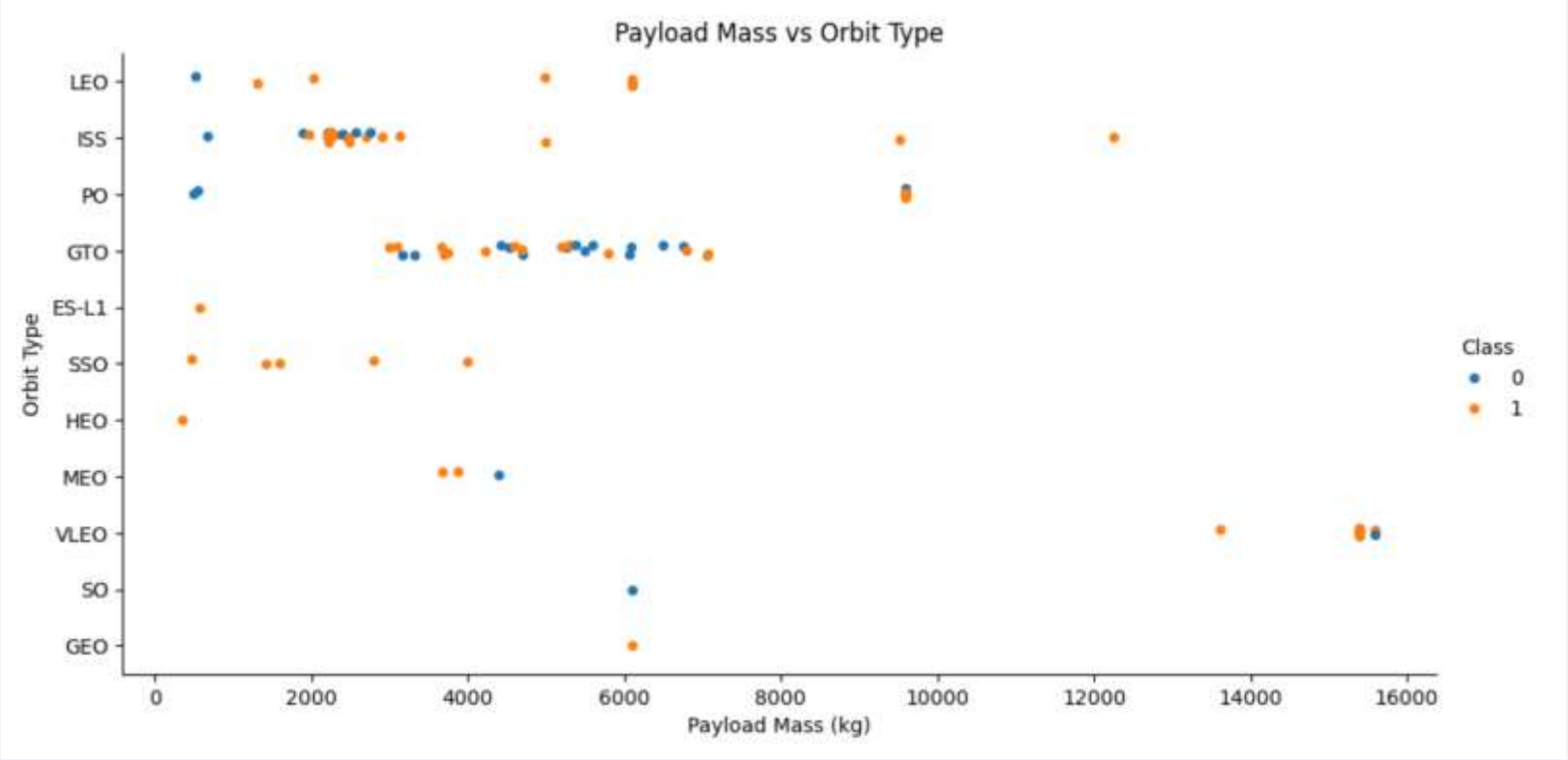
Flight Number vs. Orbit Type

- This is the scatter plot for Flight Number vs. Orbit Type. Each point represents a flight, with the x-axis showing the flight number and the y-axis showing the orbit type. The colors indicate the class: blue (0) and orange (1). We can see how different orbit types are distributed across flight numbers, and how success/failure (class) varies depending on the orbit.



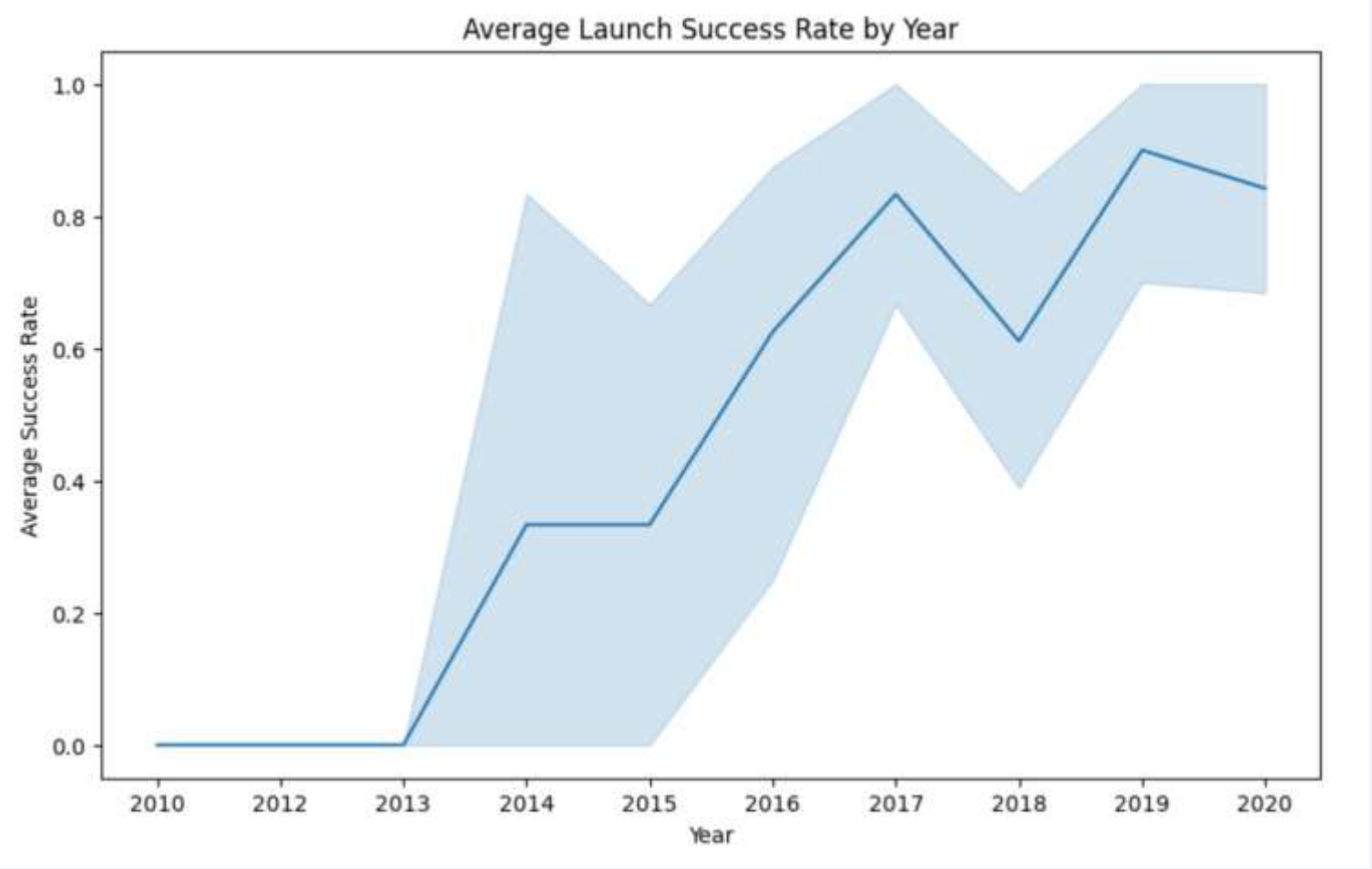
Payload vs. Orbit Type

- This is the scatter plot for Payload Mass vs. Orbit Type. Each point represents a flight, with the x-axis showing the Payload Mass(kg) and y-axis showing the orbit type. The colours indicate the class: blue (0) and orange (1).



Launch Success Yearly Trend

- In this line chart we can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

- Find the names of the unique launch sites

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

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch Site" FROM SPACEXTBL;
```

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

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

- I wrote a SQL query using the %sql command to retrieve the unique launch site names from the dataset. By applying the DISTINCT keyword on the Launch_Site column, I was able to remove duplicates and list each site only once.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

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

- I used the LIKE 'CCA%' condition on the Launch_Site column to filter records that start with "CCA".
  Then, I added LIMIT 5 to display only the first 5 rows.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload", Customer FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

Total Payload	Customer
45596	NASA (CRS)

- I wrote a SQL query using the SUM() aggregate function on the PAYLOAD_MASS__KG_ column to calculate the total payload mass.
  I filtered the records by setting the Customer column equal to 'NASA (CRS)'.
  This gave me the total payload mass carried by NASA (CRS) boosters, which was 45,596 kg.

Average Payload Mass by F9 v1.1

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

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

- I calculated the average payload mass for booster version F9 v1.1 using the AVG() aggregate function. The query filtered records where Booster_Version started with 'F9 v1.1'.
- The result showed that the average payload mass was approximately 2534.6 kg for booster F9 v1.1 B1003, launched for customer MDA.

First Successful Ground Landing Date

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

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

Hint:Use min function

```
%sql SELECT MIN(Date) AS "Successful Ground Pad Landing" FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

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

Successful Ground Pad Landing
2015-12-22

- I queried the database to find the earliest date when a SpaceX rocket successfully landed on a ground pad by selecting the minimum date (MIN(Date)) from the SPACEXTBL table where the Landing_Outcome column equals 'Success (ground pad)'. The result shows that the first successful ground landing occurred on December 22, 2015.

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

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 DISTINCT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTBL WHERE "Landing _Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version	PAYLOAD_MASS__KG_

- I wrote a SQL query to list the booster versions that had successful drone ship landings and carried a payload mass between 4000 and 6000 kg.
  I used the WHERE condition with "Landing _Outcome" = 'Success (drone ship)' and applied numeric filters on PAYLOAD_MASS__KG_.
  Although the query executed without errors, the result returned an empty table.
  This indicates that in the dataset, there are no records matching these conditions.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

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

- I wrote a SQL query using the %sql command to calculate the total number of each mission outcome. By selecting the Mission_Outcome column and using COUNT() while grouping by Mission_Outcome, I was able to see how many missions succeeded and how many failed. The query results show that most missions were successful, with only a few failures.

Boosters Carried Maximum Payload

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

List all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.

```
%sql SELECT Booster_Version, Payload_Mass__kg_  FROM SPACEXTBL WHERE Payload_Mass__kg_ = (SELECT MAX(Payload_Mass__kg_) FROM SPACEXTBL);
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

- I wrote a SQL query using the %sql command to find the boosters that carried the maximum payload mass. I selected the Booster_Version and Payload_Mass__kg_ columns and used a subquery to identify the rows where Payload_Mass__kg_ equals the maximum value in the table. The results show that several boosters.

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

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,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';
 * sqlite:///my_data1.db
Done.
```

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	"Landing _Outcome"

- I wrote a SQL query using the %sql command to list the booster versions launch site names and failed drone ship landings in 2015. I used the substr(Date,7,4) function to filter the year as 2015 and applied a condition on the Landing_Outcome column to select only 'Failure (drone ship)'.

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

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 CASE WHEN "Landing_Outcome" LIKE 'Success%' THEN 'Success' WHEN "Landing_Outcome" LIKE 'Failure%' THEN 'Failure' ELSE "Landing_Outcome" END AS Outcome_Category, COUNT(*) AS Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04'
```

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

Outcome_Category	Outcome_Count
No attempt	10
Success	8
Failure	7
Controlled (ocean)	3
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- I wrote a SQL query using the %sql command to rank the count of landing outcomes between June 4, 2010 and March 20, 2017. I used a CASE WHEN statement to group different variations of outcomes into broader categories such as Success and Failure, then applied COUNT() and ordered the results in descending order.

Section 3

Launch Sites Proximities Analysis

Launch Sites Locations on Global Map

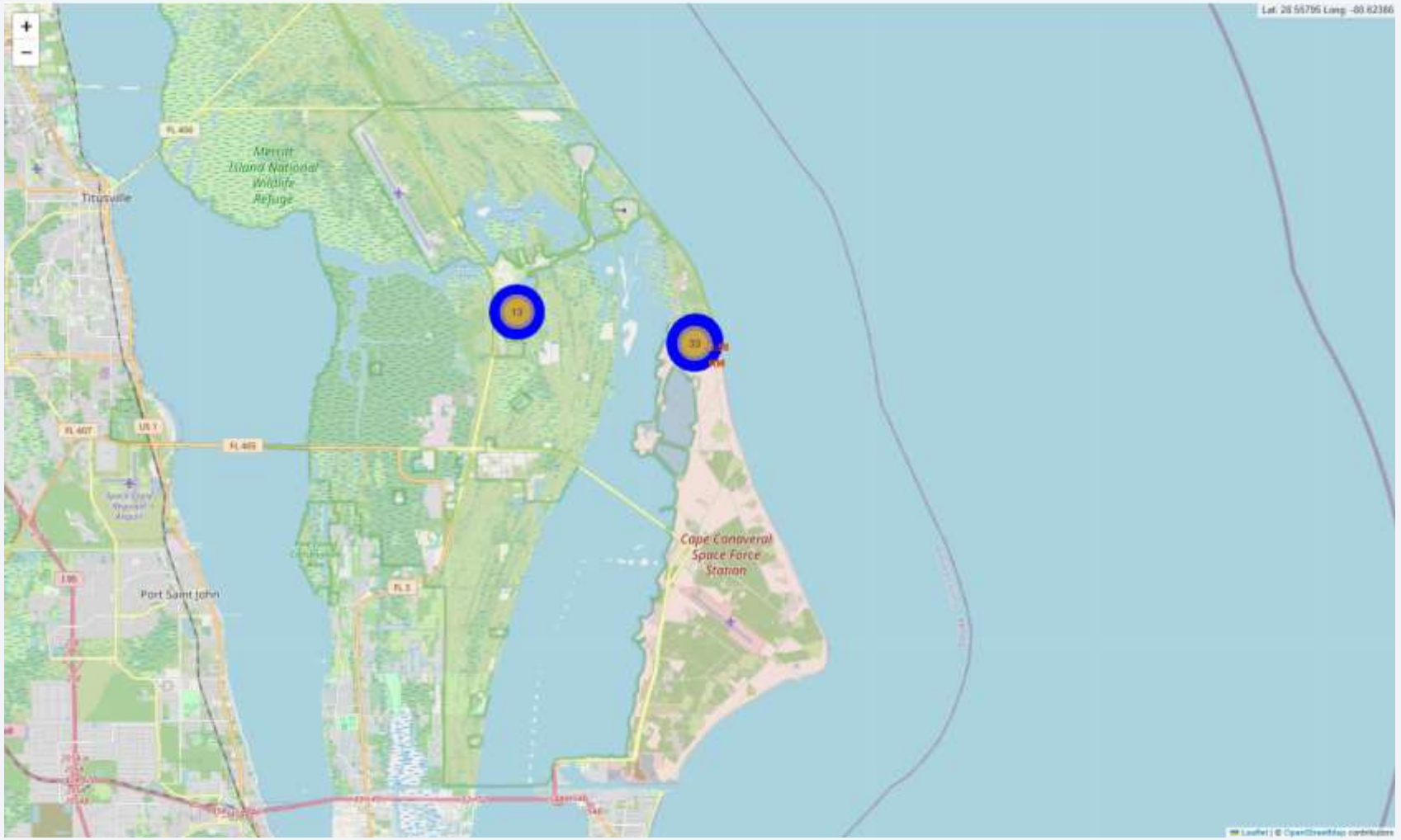
- This map displays all SpaceX launch sites around the world using blue circle markers. Each circle is placed at the site's coordinates and includes a popup label with the launch site's name.
  This visualization helps to identify the geographical distribution of launch sites and provides a global overview.

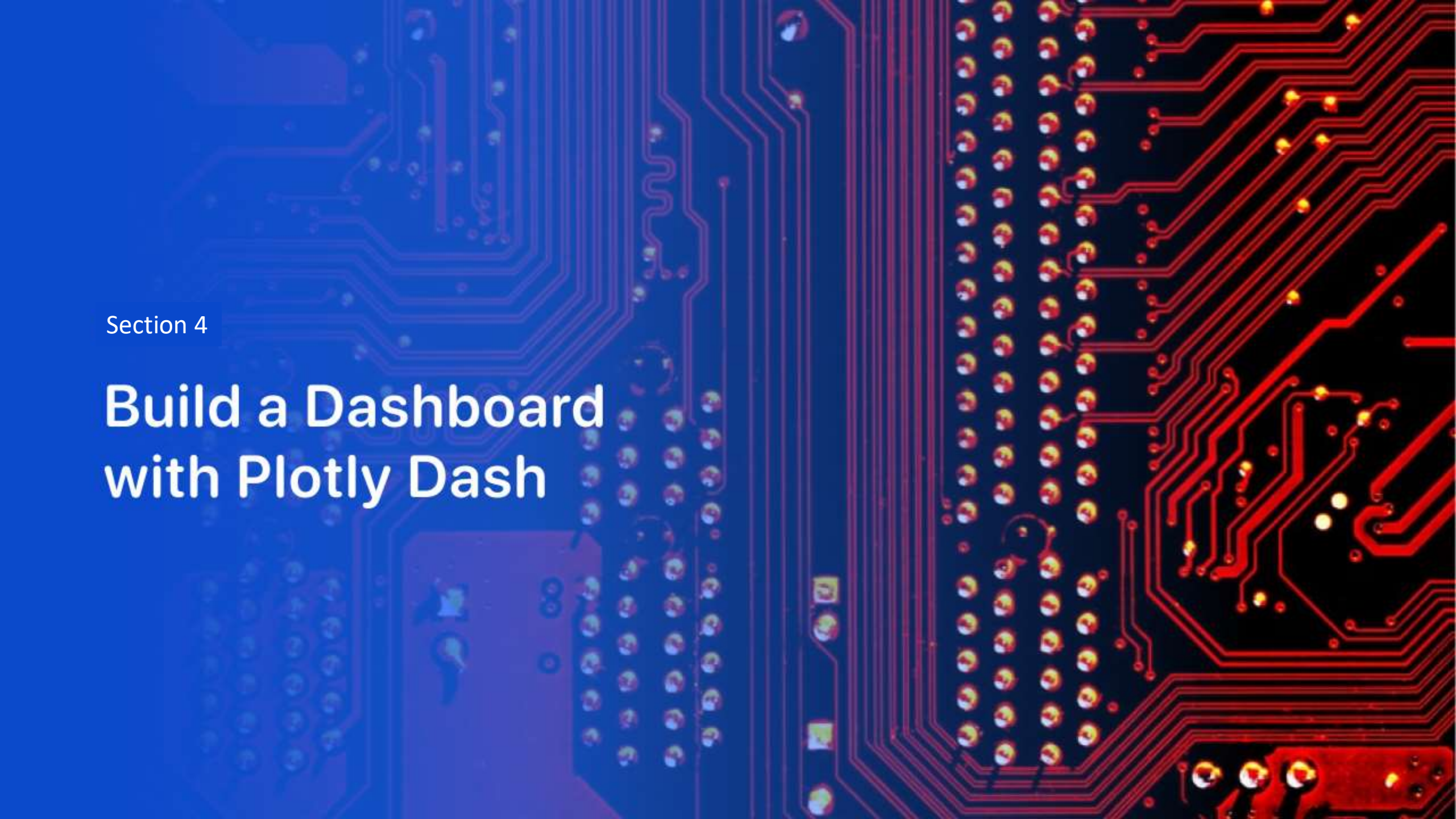
Launch Outcomes with Color-Coded Markers

- This map shows clustered launch records represented by circular markers. Each marker cluster indicates the number of launches in that area. By clustering the data, we can easily visualize the density of launches at each site and identify regions with higher activity.

Launch Site Proximity to Coastline and Infrastructure

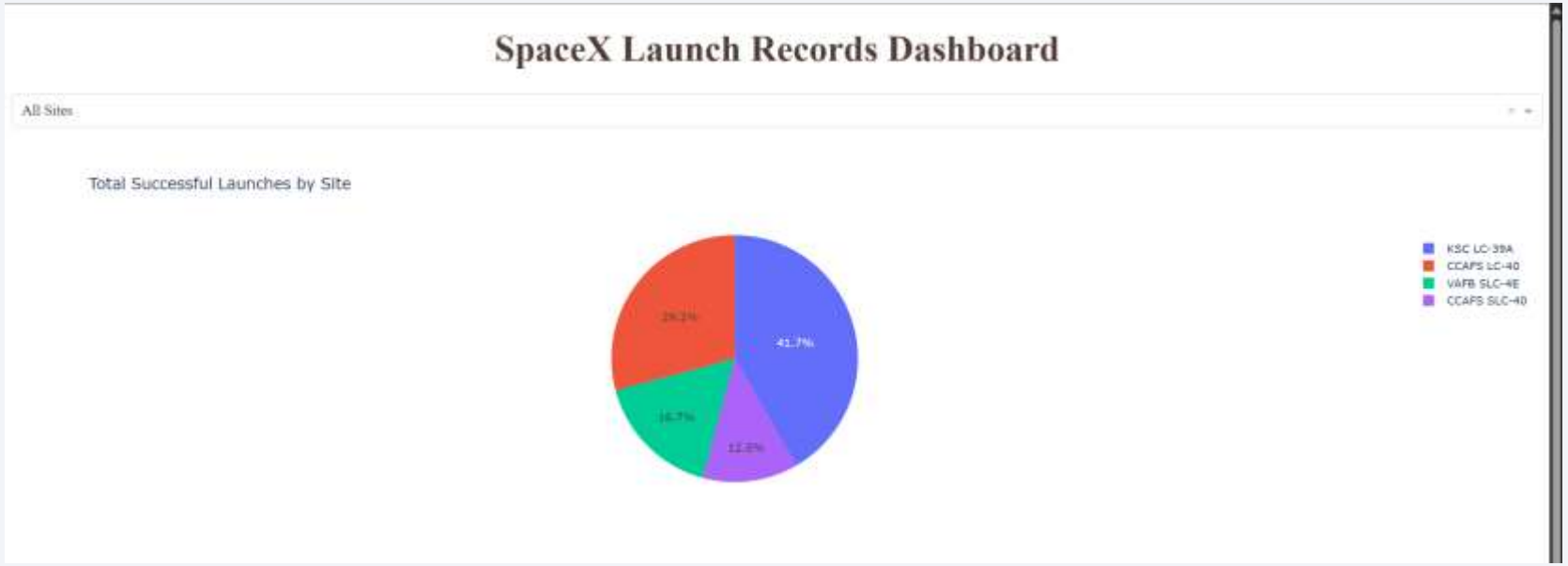
- This map highlights selected launch sites and their proximity to key infrastructure such as the coastline, highways, nearby cities, and airports. Distances are indicated with numeric markers, making it clear how launch facilities are positioned relative to surrounding features.
- This analysis is important because launch sites require open space and safe distances from populated areas or transportation lines.




Section 4
Build a Dashboard with Plotly Dash

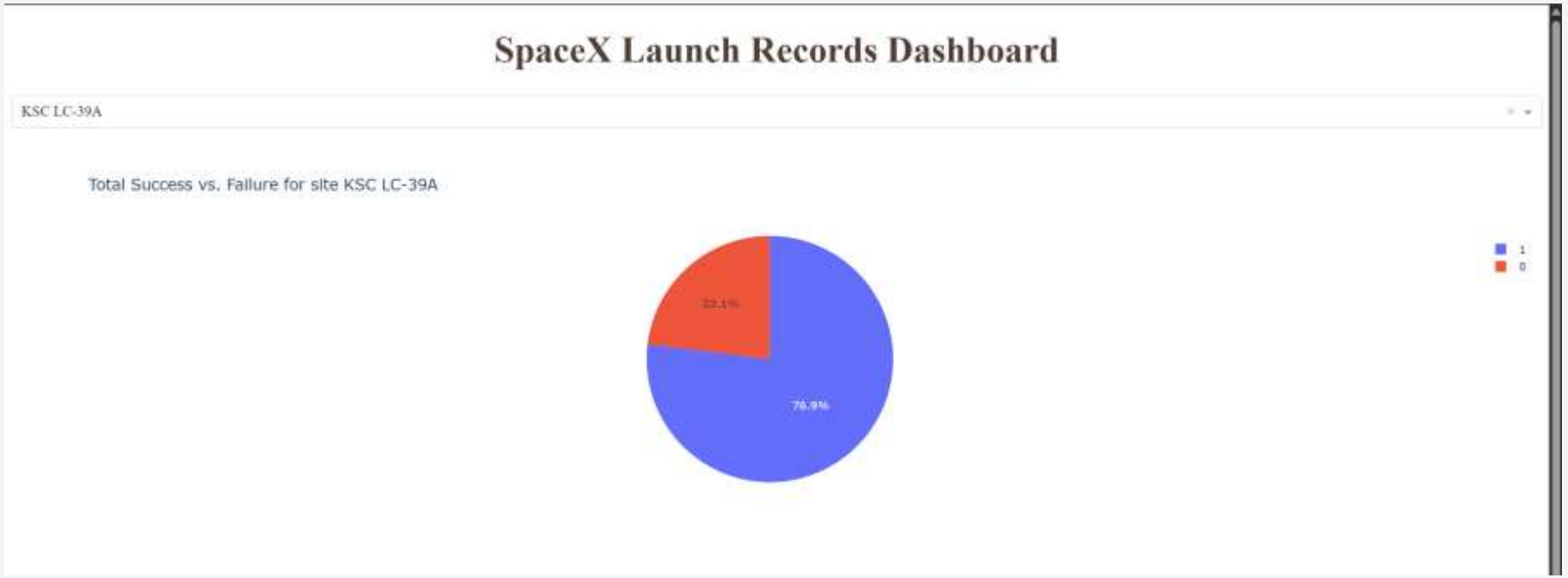
Launch Success Counts by Site

- This screenshot shows the overall launch success distribution across all SpaceX launch sites. The data is visualized in a pie chart where each colour represents a different launch site.
- From the chart, we can see that:
- KSC LC-39A accounts for the largest share of successful launches, making up about 41.7%.
- CCAFS LC-40 follows with around 29.2%.
- VAFB SLC-4E contributes about 16.7%.
- CCAFS SLC-40 has the smallest share, about 12.5%.
- This breakdown highlights that most successful missions were launched from KSC LC-39A, showing its importance as the primary site for SpaceX launches.



Launch Success vs Failure KSC LC-39A

- This screenshot shows the launch outcome distribution for KSC LC-39A, the site with the highest launch success ratio.
- The pie chart clearly illustrates that:
- 76.9% of launches were successful (blue section).
- 23.1% of launches failed (red section).



Correlation between Payload Mass and Launch Success at KSC LC-39A

- The scatter plot shows payload mass vs. launch success at KSC LC-39A, with booster types in colour: FT = blue, B4 = red, B5 = green.
- Key points:
- Most launches with mid-range payloads (2,500–5,500 kg) succeed.
- FT boosters have some failures at high payload.
- B4 and B5 boosters are mostly successful.
- In short: mid-range payloads with B4 or B5 boosters succeed most, while heavy payloads with FT boosters fail a bit more.

Section 5

Predictive Analysis (Classification)

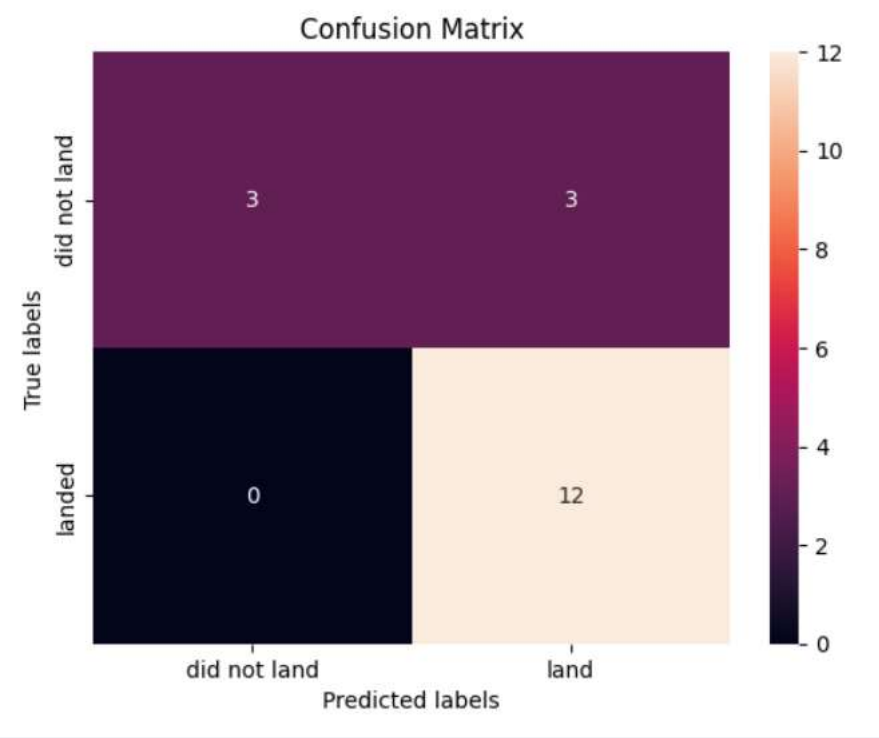
Classification Accuracy

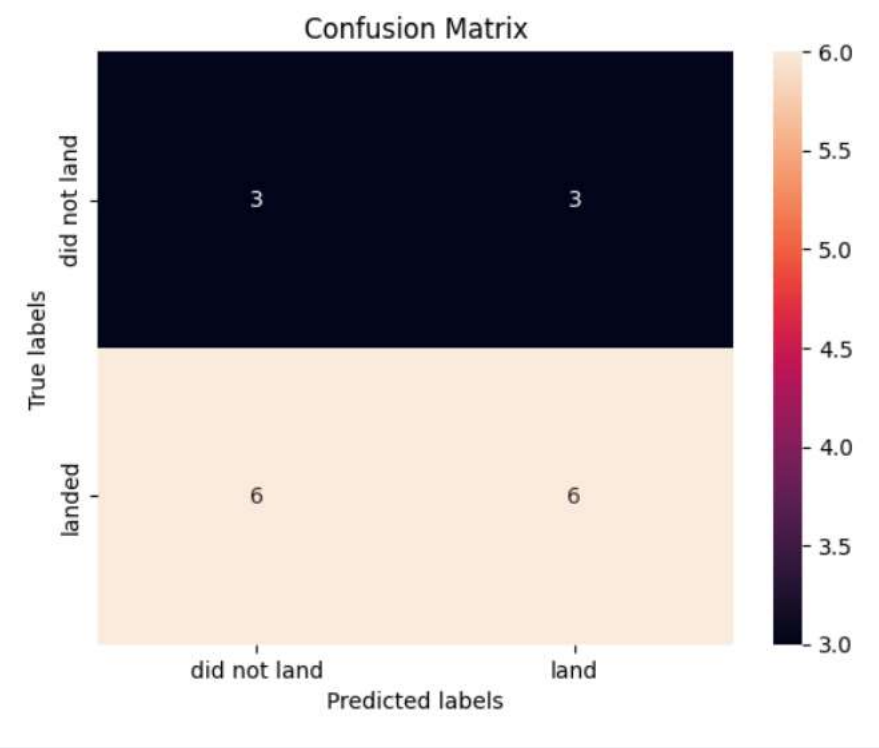
- This table shows the accuracy of all the classification models I built. Logistic Regression, SVM, and KNN all perform the best with an accuracy of 83.3%, while the Decision Tree model is lower at 50%.
- In short: Logistic Regression, SVM, and KNN are the top performers.

Method	Conclusion
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.5
KNN	0.833333

Confusion Matrix

- Decision Tree (50% Accuracy): Struggles to classify correctly, with many misclassifications.
- Logistic Regression, SVM, KNN (83.3% Accuracy): Perform much better, correctly identifying most successful landings with fewer errors.
- Best models: Logistic Regression, SVM, and KNN





Conclusions

- Launch success rates improve over time and with higher flight numbers.
- Success varies across launch sites, with KSC LC-39A being the most reliable.
- Payload mass and orbit type both influence landing success.
- Mid-range payloads with B4/B5 boosters perform best, while heavy payloads with FT boosters are riskier.
- Logistic Regression, SVM, and KNN are the best models (83.3% accuracy), outperforming Decision Tree.

Thank you!