IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Roimemy RALISON

1 April, 2025

Outline

- Executive Summary
- Introduction
- Section 1: Methodology
- Section 2: Insights drawn from EDA
- Section 3: Launch sites proximities analysis
- Section 4: Build a dashboard with Plotly Dash
- Section 5: Predictive analysis (classification)
- Conclusions

Executive Summary

This analysis evaluates SpaceX's historical launch performance, focusing on key metrics such as launch success rates and the impact of payload mass on success. To achieve this, the methodology involved systematic data collection, cleaning, and visualisation through interactive Dash components, including pie charts and scatter plots. As a result, a dynamic dashboard was developed, enabling users to filter data by launch site and payload range for a comprehensive review of SpaceX's performance. The findings reveal that SpaceX maintains a generally high success rate across all launch sites, with some variability observed across different locations. In addition, no significant correlation was found between payload mass and launch success, which suggests that other factors may influence outcomes. Overall, this analysis offers actionable insights for optimising future launch strategies.

Introduction

This report provides an in-depth analysis of SpaceX's launch performance over time, with a focus on understanding the factors that influence launch success. In particular, the study addresses two key questions:

- ✓ the consistency of SpaceX's success rates across various launch sites
- ✓ and the role of payload mass in determining launch outcomes.

Given the significant financial and operational investment involved in space missions, identifying the critical variables that influence launch success is essential for optimising future operations. The problems to solve include determining whether success rates vary significantly between launch sites and exploring the potential correlation between payload mass and mission success. By addressing these questions, the report aims to provide actionable insights that can guide SpaceX's strategic decisions, ensuring the continued success of its missions.

Section 1

Methodology

Methodology

1. The project begins with data collection using **web scraping** techniques to gather relevant launch information.

2. After collection, **data wrangling** is performed to clean, format, and structure the dataset for analysis.

3. The processed data is then explored through **exploratory data analysis (EDA)** using SQL, Pandas, and Matplotlib to identify patterns and trends.

4. **Interactive visual analytics** are developed using Folium for geospatial visualisations and Plotly Dash for **interactive dashboards**.

5. **Predictive analysis** is conducted by applying classification models to determine the likelihood of Falcon 9 first stage landing success.
   The models are built, tuned, and evaluated to ensure accuracy and reliability in forecasting outcomes.

Data Collection

The project uses launch data obtained from the SpaceX REST API, which provides detailed information about each mission. This includes the rocket used, the payload delivered, and specific launch and landing parameters, along with the final landing outcome. The objective is to analyse this data to predict whether SpaceX will attempt a rocket landing for a given launch.

Data Collection – SpaceX API

OBJECTIVES: Make a request to the SpaceX API and clean the requested data

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

TASK 2: Filter the dataframe to only include `Falcon 9` launches

TASK 3: Dealing with missing values

https://github.com/l1english/data-science-capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

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
launch_data = response.json()
df = pd.json_normalize(launch_data)
```

Using the dataframe `data` print the first 5 rows

```
# Get the head of the dataframe
df.head(5)
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	payloads	launchpad	flight_number	name	
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]	5e9e4502f5090995de566f86	1	FalconSat	24T2
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b6c3bb0006eeb1e2]	5e9e4502f5090995de566f86	2	DemoSat	21T0
							[{'time': 140, 'altitude': 35,	Residual stage 1								

Data Collection - Scraping

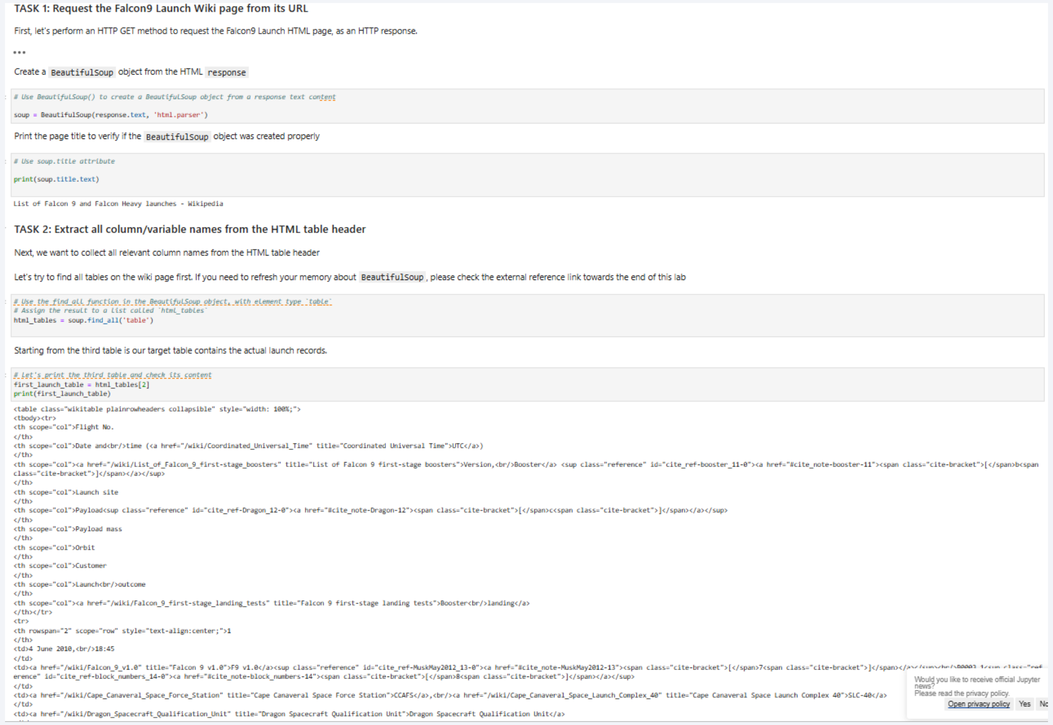
OBJECTIVES: Web scrap Falcon 9 launch records with BeautifulSoup

TASK 1: Request the Falcon 9 Launch Wiki page from its URL

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

TASK 3: Create a data frame by parsing the launch HTML tables

https://github.com/l1english/data-science-capstone/blob/main/jupyter-labs-webscraping.ipynb



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.

...

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

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

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

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

```
List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

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

```python
# 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.

```python
# 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%;">
<tbody><tr>
<th scope="col">Flight No.
</th>
<th scope="col">Date and<br/>time (<a href="/wiki/Coordinated_Universal_Time" title="Coordinated Universal Time">UTC</a>)
</th>
<th scope="col"><a href="/wiki/List_of_Falcon_9_first-stage_boosters" title="List of Falcon 9 first-stage boosters">Version,<br/>Booster</a> <sup class="reference" id="cite_ref-booster_11-0"><a href="#cite_note-booster-11"><span class="cite-bracket">[</span>b<span class="cite-bracket">]</span></a></sup>
</th>
<th scope="col">Launch site
</th>
<th scope="col">Payload<sup class="reference" id="cite_ref-Dragon_12-0"><a href="#cite_note-Dragon-12"><span class="cite-bracket">[</span>c<span class="cite-bracket">]</span></a></sup>
</th>
<th scope="col">Payload mass
</th>
<th scope="col">Orbit
</th>
<th scope="col">Customer
</th>
<th scope="col">Launch<br/>outcome
</th>
<th scope="col"><a href="/wiki/Falcon_9_first-stage_landing_tests" title="Falcon 9 first-stage landing tests">Booster<br/>landing</a>
</th></tr>
<tr>
<th rowspan="2" scope="row" style="text-align:center;">1
</th>
<td>4 June 2010,<br/>18:45
</td>
<td><a href="/wiki/Falcon_9_v1.0" title="Falcon 9 v1.0">F9 v1.0</a><sup class="reference" id="cite_ref-MuskMay2012_13-0"><a href="#cite_note-MuskMay2012-13"><span class="cite-bracket">[</span>7<span class="cite-bracket">]</span></a>
erence" id="cite_ref-block_numbers_14-0"><a href="#cite_note-block_numbers-14"><span class="cite-bracket">[</span>8<span class="cite-bracket">]</span></a></sup>
</td>
<td><a href="/wiki/Cape_Canaveral_Space_Force_Station" title="Cape Canaveral Space Force Station">CCAFS</a>,<br/><a href="/wiki/Cape_Canaveral_Space_Launch_Complex_40" title="Cape Canaveral Space Launch Complex 40">SLC-40</a>
</td>
<td><a href="/wiki/Dragon_Spacecraft_Qualification_Unit" title="Dragon Spacecraft Qualification Unit">Dragon Spacecraft Qualification Unit</a>
```

Would you like to receive official Jupyter news?
Please read the privacy policy.
Open privacy policy Yes No

Data Wrangling

OBJECTIVES: Perform exploratory data analysis and determine training labels

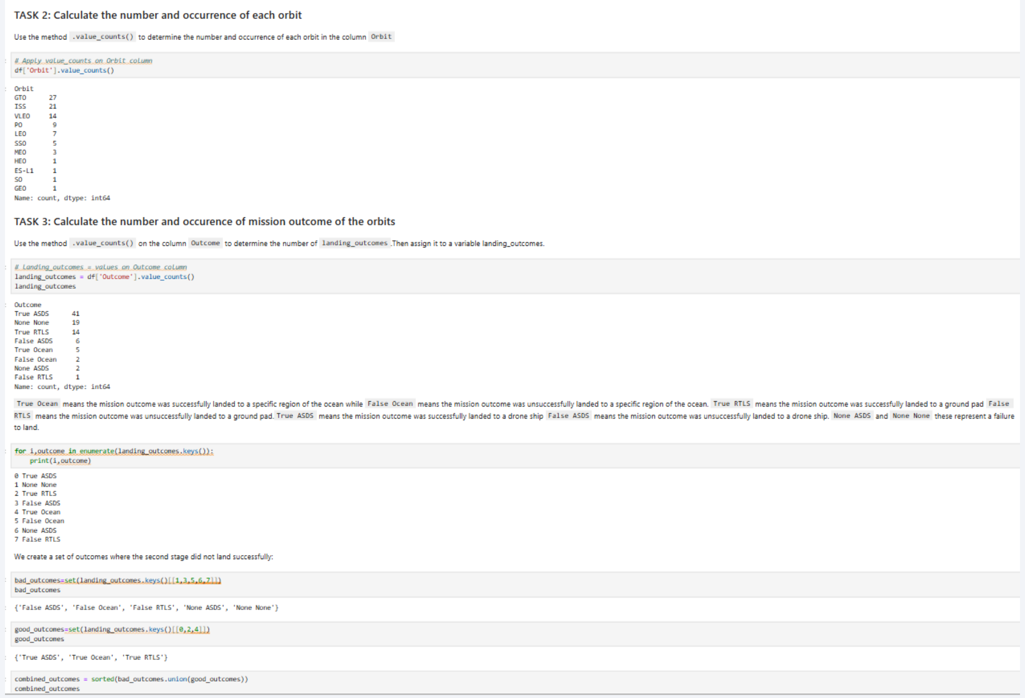
TASK 1: Calculate the number of launches on each site

TASK 2: Calculate the number and occurrence of each orbit

TASK 3: Calculate the number and occurrence of mission outcome of the orbits

TASK 4: Create a landing outcome label from Outcome column

https://github.com/l1english/data-science-capstone/blob/main/jupyter-spacex-datawrangling.ipynb



TASK 2: Calculate the number and occurrence of each orbit

Use the method `.value_counts()` to determine the number and occurrence of each orbit in the column `Orbit`

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

TASK 3: Calculate the number and occurence of mission outcome of the orbits

Use the method `.value_counts()` on the column `Outcome` to determine the number of `landing_outcomes` .Then assign it to a variable landing_outcomes.

```
# landing_outcomes = values on Outcome column
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

`True Ocean` means the mission outcome was successfully landed to a specific region of the ocean while `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean. `True RTLS` means the mission outcome was successfully landed to a ground pad `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad. `True ASDS` means the mission outcome was successfully landed to a drone ship `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship. `None ASDS` and `None None` these represent a failure to land.

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
```

```
0 True ASDS
1 None None
2 True RTLS
3 False ASDS
4 True Ocean
5 False Ocean
6 None ASDS
7 False RTLS
```

We create a set of outcomes where the second stage did not land successfully:

```
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
bad_outcomes
```

```
{'False ASDS', 'False Ocean', 'False RTLS', 'None ASDS', 'None None'}
```

```
good_outcomes=set(landing_outcomes.keys()[[0,2,4]])
good_outcomes
```

```
{'True ASDS', 'True Ocean', 'True RTLS'}
```

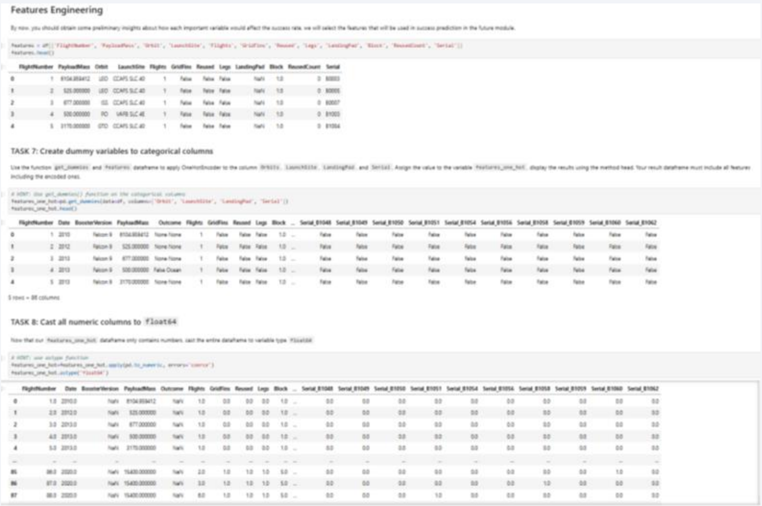
```
combined_outcomes = sorted(bad_outcomes.union(good_outcomes))
combined_outcomes
```

EDA with Data Visualization

OBJECTIVES: Perform exploratory data analysis (EDA) and feature engineering using `Pandas` and `Matplotlib`

- ➢ EDA offers insight about how different variables (e.g., launch site, payload mass, orbit) can affect the launch outcomes
- ➢ Features engineering creates and transform data features to improve model performance

https://github.com/l1english/data-science-capstone/blob/main/jupyter-labs-edadataviz.ipynb

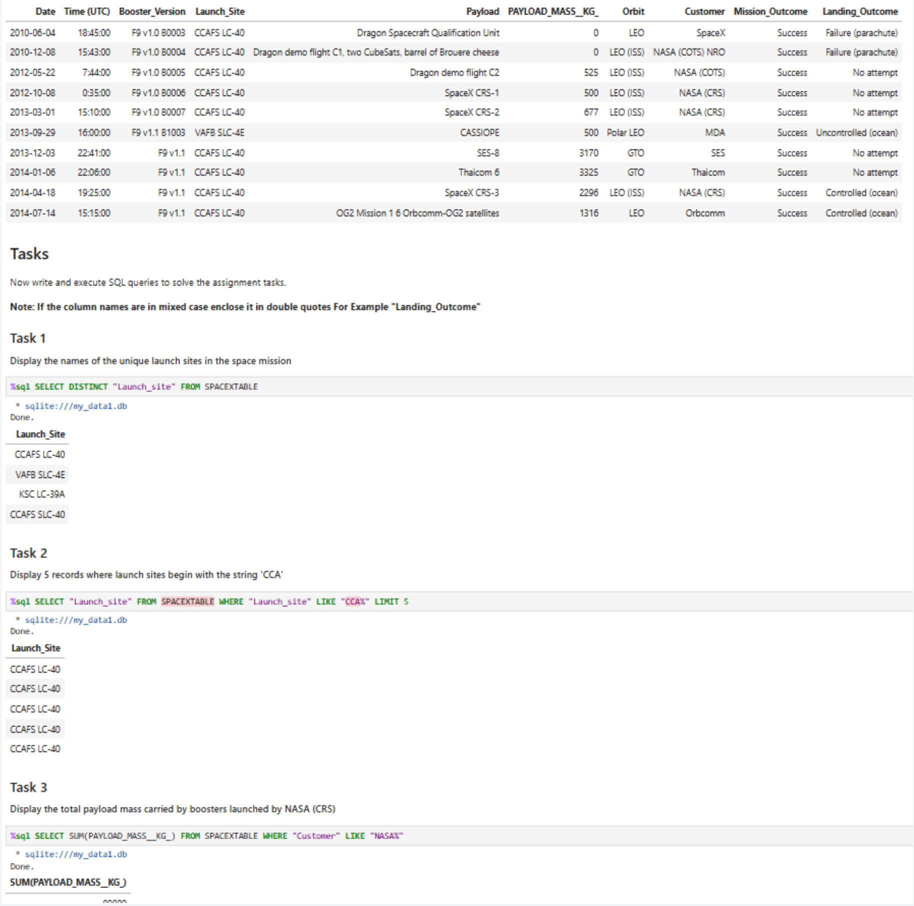


EDA with SQL

The SQL queries performed on the SpaceX data set include:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass, using a subquery
- 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.
- 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

https://github.com/l1english/data-science-capstone/blob/main/jupyter-labs-eda-sql.ipynb



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
2013-09-29	16:00:00	F9 v1.1 B1003	VAFB SLC-4E	CASSIOPE	500	Polar LEO	MDA	Success	Uncontrolled (ocean)
2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt
2014-01-06	22:06:00	F9 v1.1	CCAFS LC-40	Thaicom 6	3325	GTO	Thaicom	Success	No attempt
2014-04-18	19:25:00	F9 v1.1	CCAFS LC-40	SpaceX CRS-3	2296	LEO (ISS)	NASA (CRS)	Success	Controlled (ocean)
2014-07-14	15:15:00	F9 v1.1	CCAFS LC-40	OG2 Mission 1 6 Orbcomm-OG2 satellites	1316	LEO	Orbcomm	Success	Controlled (ocean)

Tasks

Now write and execute SQL queries to solve the assignment tasks.

Note: If the column names are in mixed case enclose it in double quotes For Example "Landing_Outcome"

Task 1

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

```
%sql SELECT DISTINCT "Launch_site" FROM SPACEXTABLE
```

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

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

Task 2

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

```
%sql SELECT "Launch_site" FROM SPACEXTABLE WHERE "Launch_site" LIKE "CCA%" LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Task 3

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer" LIKE "NASA%"
```

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

SUM(PAYLOAD_MASS__KG_)

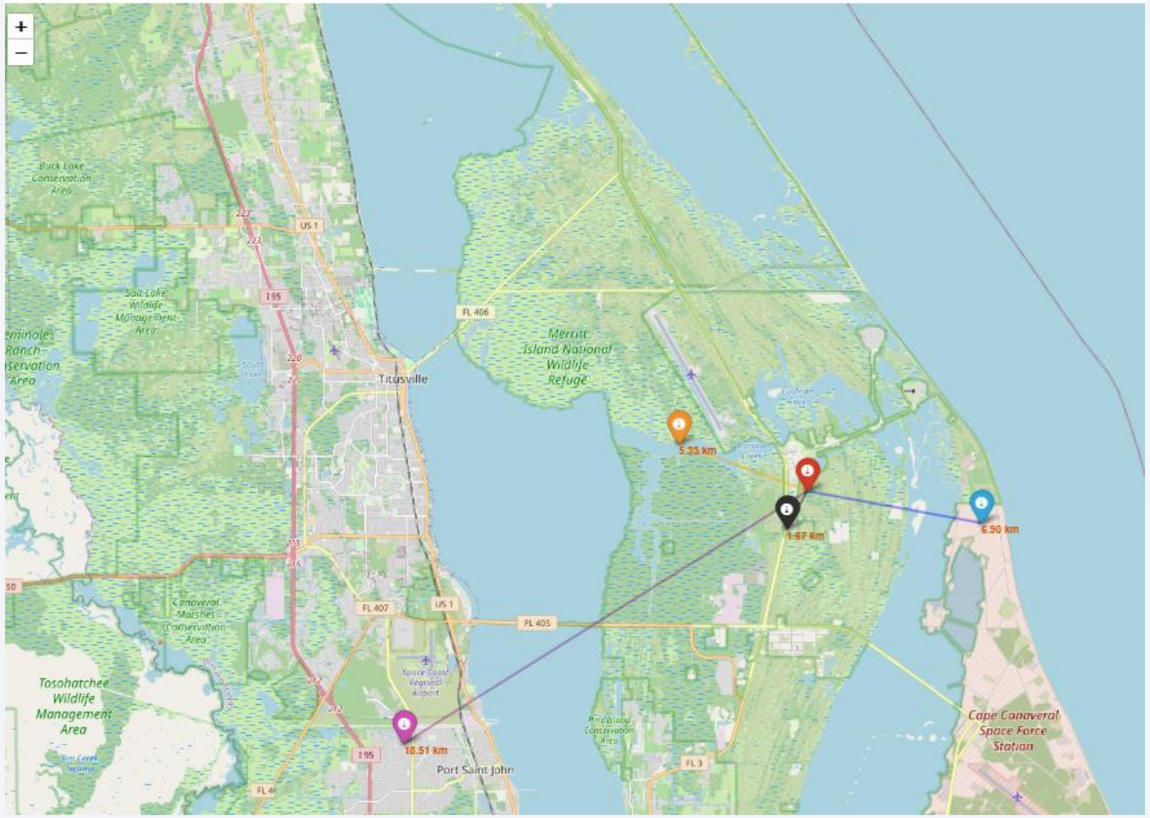
Build an Interactive Map with Folium

Folium was used to create an interactive visualisation that highlights the launch site locations and their spatial relationship with nearby features such as the coastline, city, railway, and highway.

https://github.com/l1english/data-science-capstone/blob/main/jupyter-labs-folium.ipynb

```
add_feature(coastline_coord, "Closest Coastline", "blue")
add_feature(closest_city_coord, "Closest City", "purple")
add_feature(closest_railway_coord, "Closest Railway", "black")
add_feature(closest_highway_coord, "Closest Highway", "orange")

site_map
```



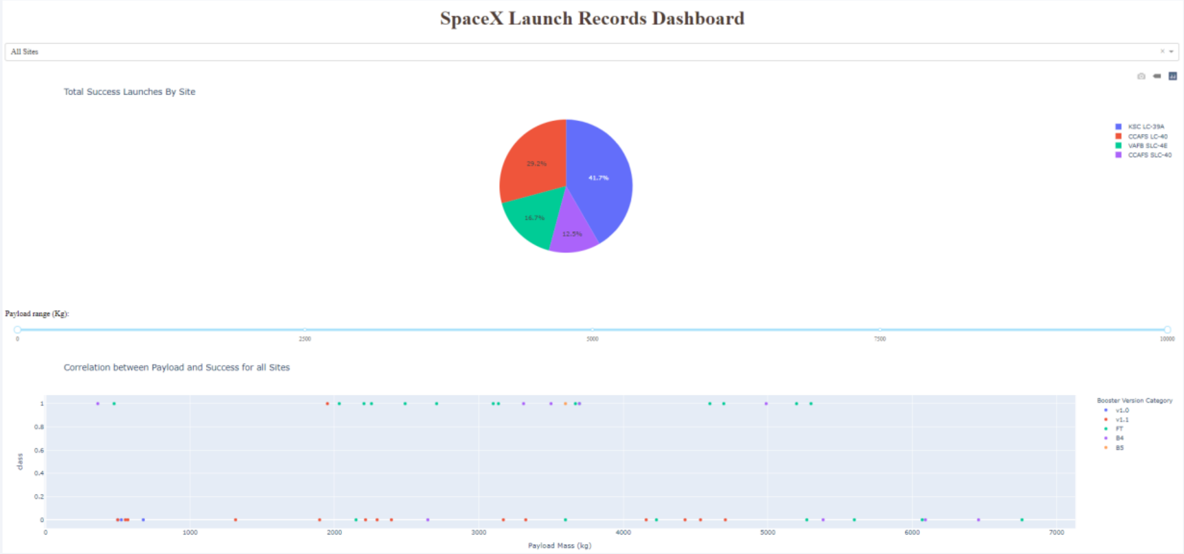
Build a Dashboard with Plotly Dash

The dashboard uses a dropdown to select launch sites and a range slider to filter payload mass.

It includes a pie chart showing success rates and a scatter plot revealing payload-success correlations. These interactive components help users explore SpaceX launch data, compare site performance, and analyse booster version outcomes effectively.

Dashboard code:

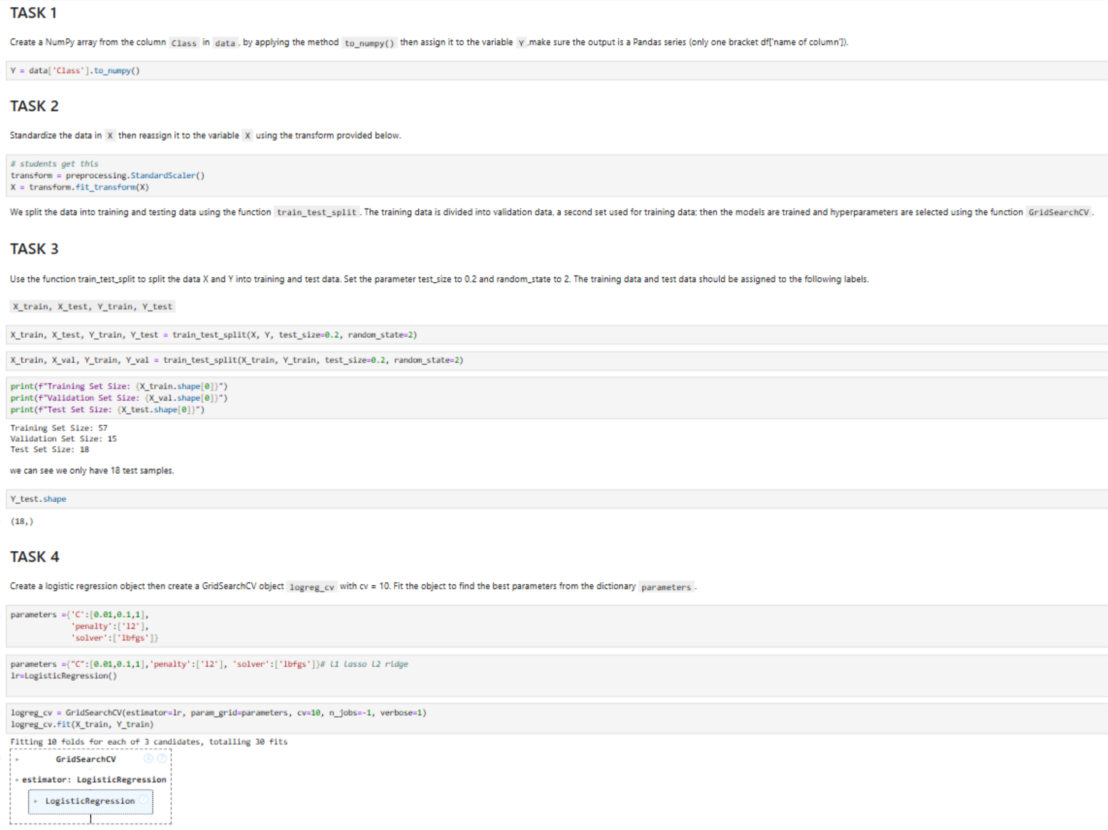
https://github.com/l1english/data-science-capstone/blob/main/spacex-dash-final.py



Predictive Analysis (Classification)

OBJECTIVES: Perform an EDA and determine training labels by:

- Creating a column for the class
- Standardising the data
- Splitting it into training and test data
- Finding the best performing method using test data



TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable `Y` ,make sure the output is a Pandas series (only one bracket df['name of column']).

```
Y = data['Class'].to_numpy()
```

TASK 2

Standardize the data in `X` then reassign it to the variable `X` using the transform provided below.

```
# students get this
transform = preprocessing.StandardScaler()
X = transform.fit_transform(X)
```

We split the data into training and testing data using the function `train_test_split`. The training data is divided into validation data, a second set used for training data; then the models are trained and hyperparameters are selected using the function `GridSearchCV`.

TASK 3

Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2. The training data and test data should be assigned to the following labels.

`X_train, X_test, Y_train, Y_test`

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=2)
```

```
X_train, X_val, Y_train, Y_val = train_test_split(X_train, Y_train, test_size=0.2, random_state=2)
```

```
print(f"Training Set Size: {X_train.shape[0]}")
print(f"Validation Set Size: {X_val.shape[0]}")
print(f"Test Set Size: {X_test.shape[0]}")
```

```
Training Set Size: 57
Validation Set Size: 15
Test Set Size: 18
```

we can see we only have 18 test samples.

```
Y_test.shape
```

```
(18,)
```

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`.

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

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}# l1 lasso l2 ridge
lr=LogisticRegression()
```

```
logreg_cv = GridSearchCV(estimator=lr, param_grid=parameters, cv=10, n_jobs=-1, verbose=1)
logreg_cv.fit(X_train, Y_train)
```

```
Fitting 10 folds for each of 3 candidates, totalling 30 fits
```

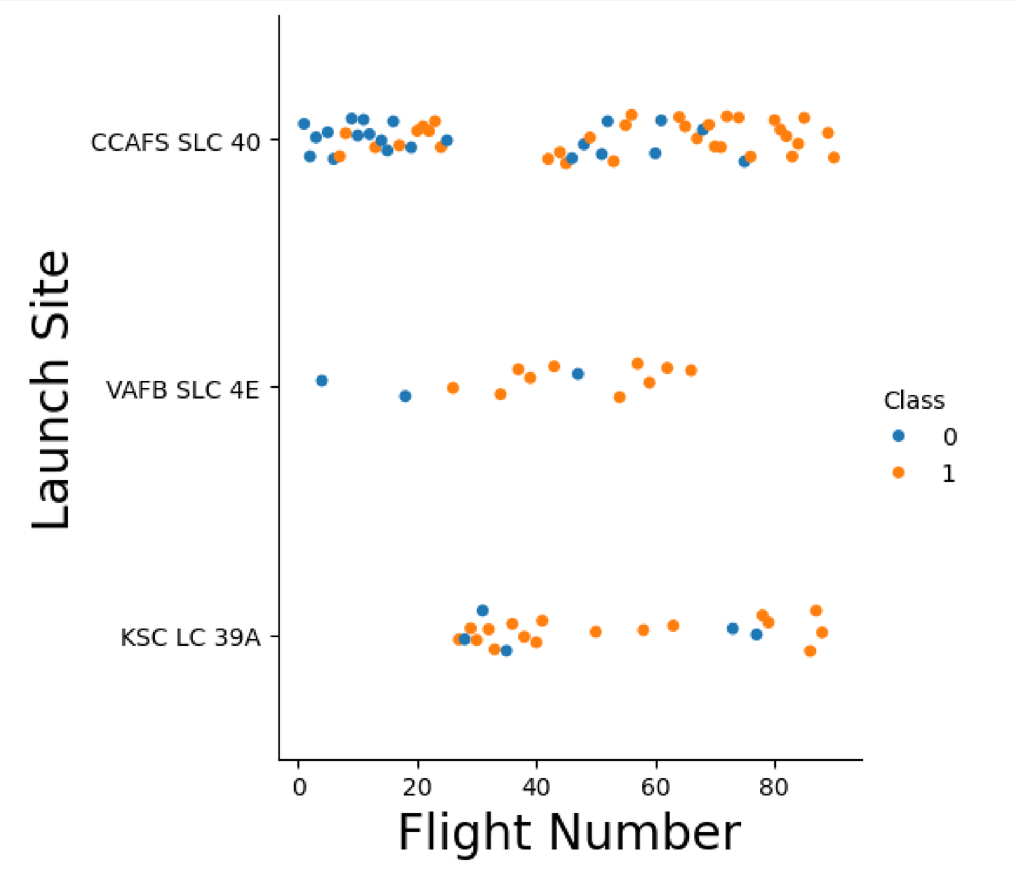
```
GridSearchCV
estimator: LogisticRegression
LogisticRegression
```

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

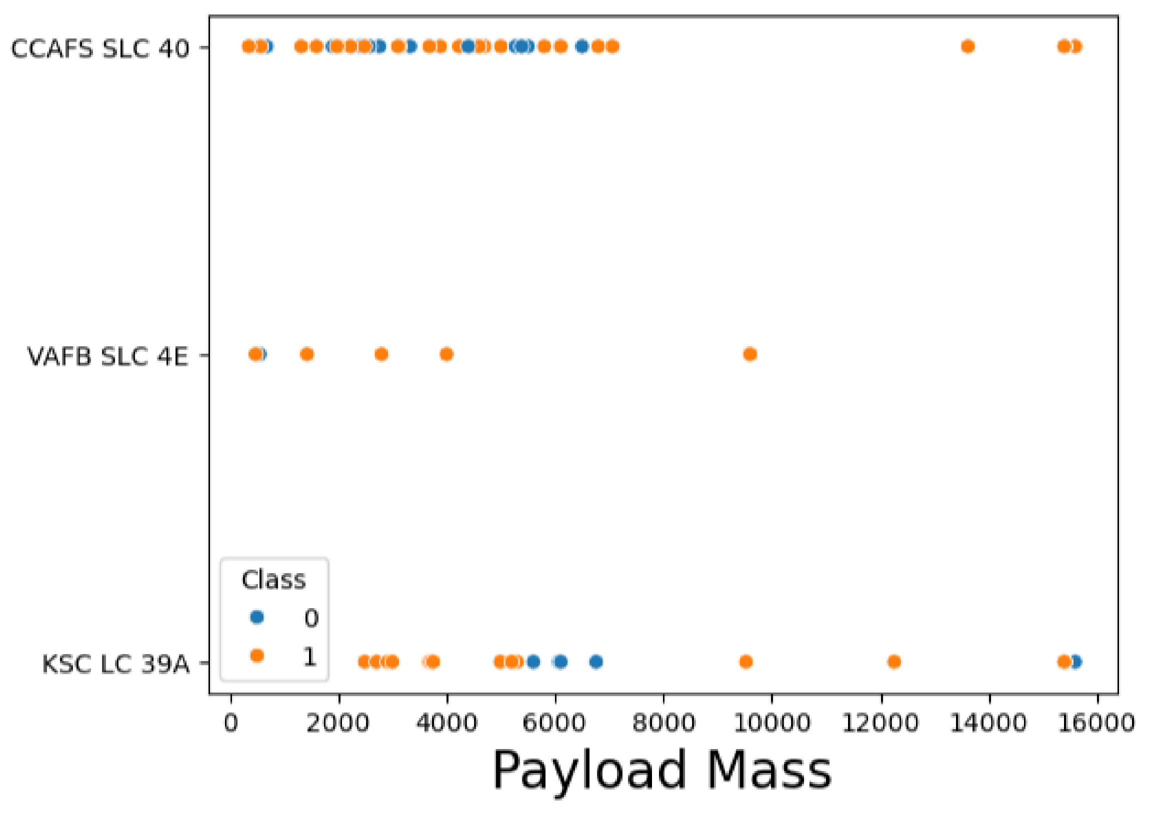
Launches are concentrated around a few key facilities, with CCAFS SLC 40 and KSC LC 39A leading in activity. Over time, missions have become increasingly successful, as indicated by the dominance of successful outcomes (Class 1) in higher flight numbers. Early efforts saw a mix of results across sites, but operational reliability improved significantly with experience.



Payload vs. Launch Site

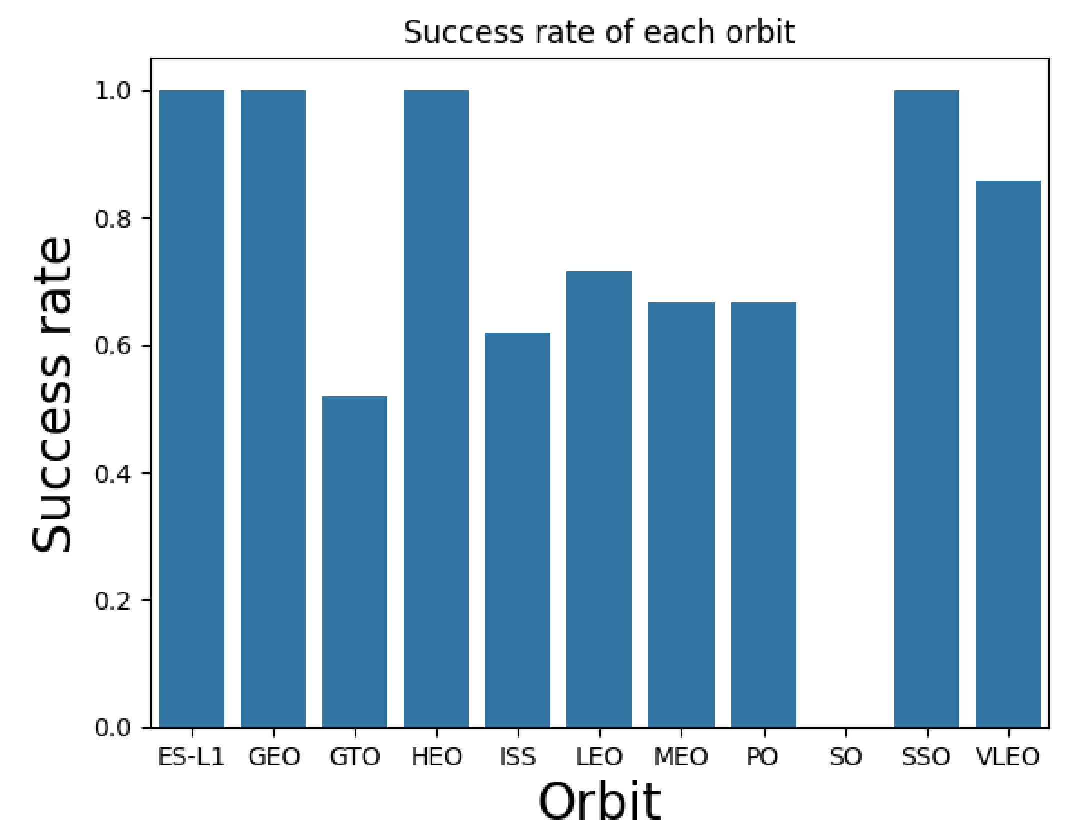
CCAFS SLC 40 and KSC LC 39A have handled the heaviest payloads, reflecting their capability for large-scale missions. Failures appear more often among lower-mass launches, possibly due to earlier testing phases or less complex payloads.

As the payload mass increases, so does the consistency of successful launches, highlighting the maturation of launch procedures.



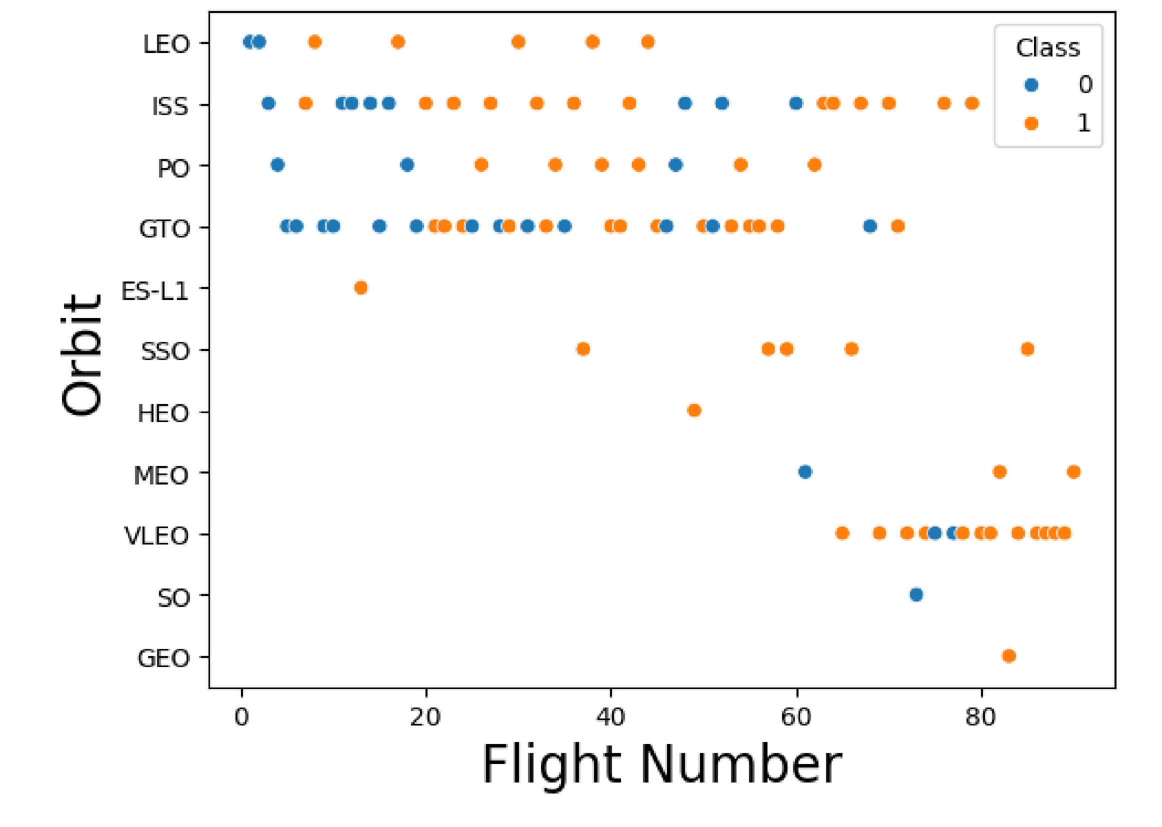
Success Rate vs. Orbit Type

Missions targeting ES-L1, GEO, and SSO achieved nearly perfect reliability, whereas MEO and HEO exhibited more variability. These figures suggest a strong track record in standard orbital paths, with room for growth in less conventional or more technically demanding orbits.



Flight Number vs. Orbit Type

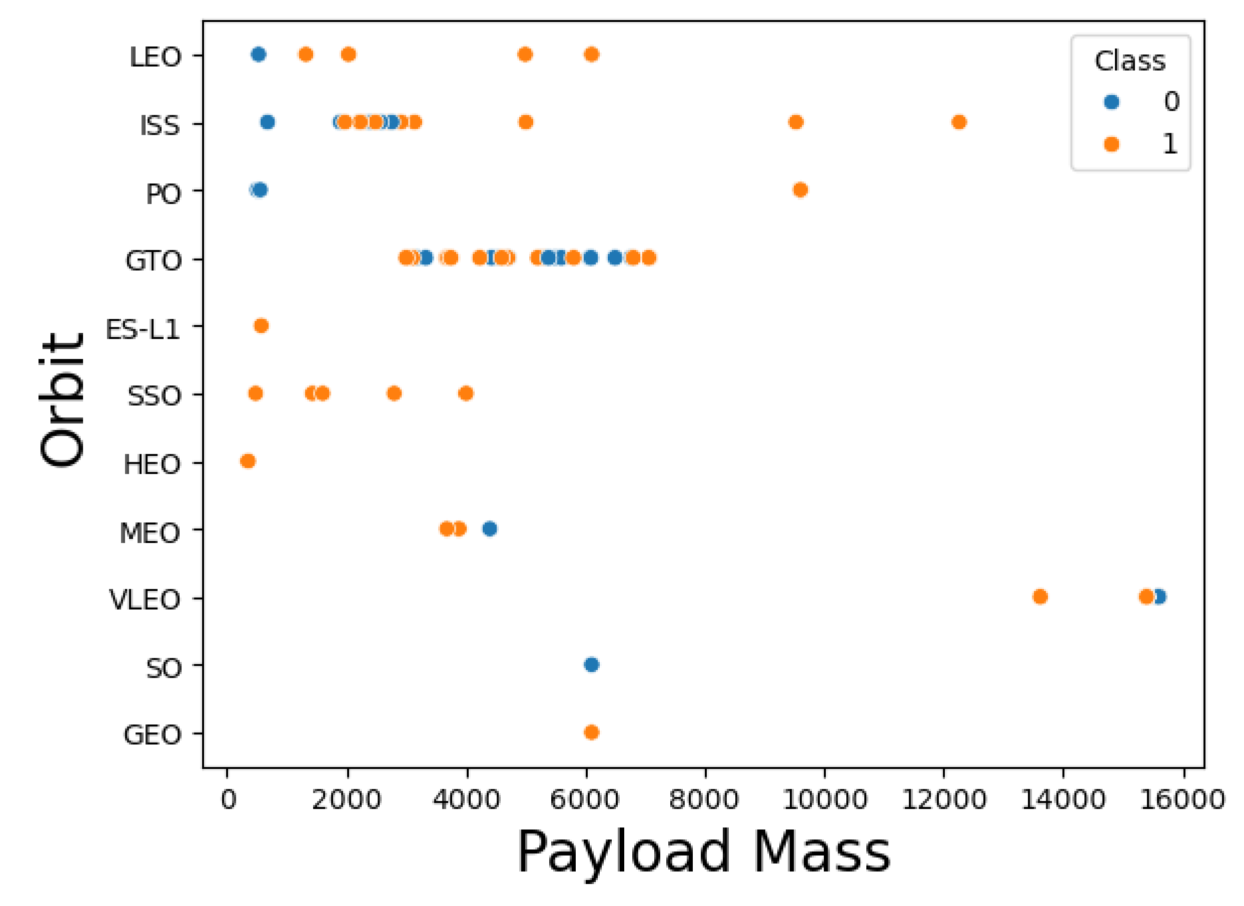
As launch experience increased, the diversity of orbital destinations expanded. Early flights focused mainly on LEO, but later missions ventured into more specialised orbits such as GEO, ES-L1, and VLEO. This development reflects both technological advancement and growing mission complexity, corresponding with SpaceX's evolving capabilities.



Payload vs. Orbit Type

Heavier payloads are primarily sent to destinations like GTO and ISS, while LEO accommodates the broadest range of masses. Unsuccessful missions are mostly clustered around mid-weight payloads and are dispersed across several orbit types.

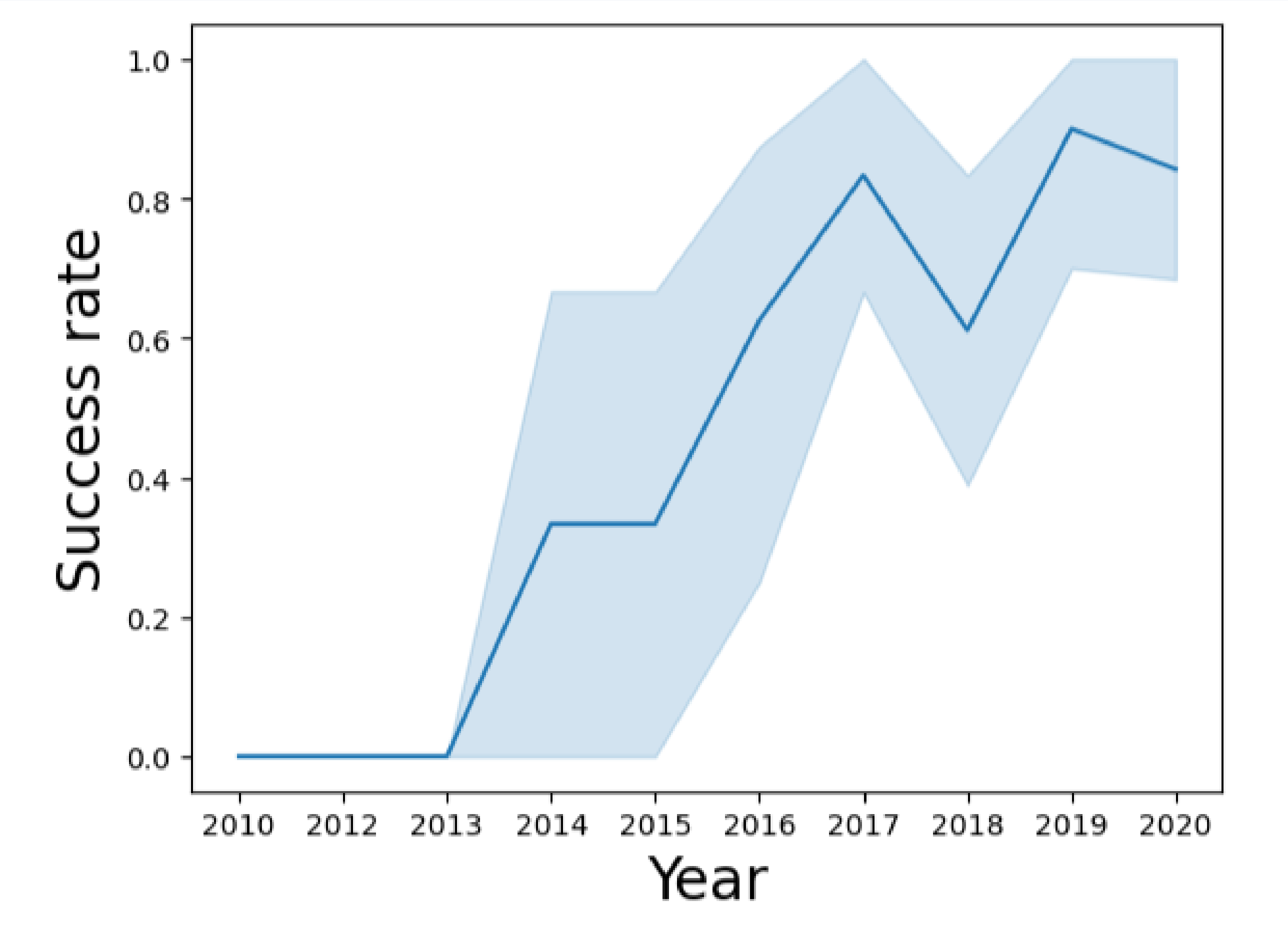
The dominance of LEO in both frequency and payload diversity underscores its strategic importance in mission planning.



Launch Success Yearly Trend

Between 2010 and 2014, outcomes varied, reflecting a period of experimentation and adjustment. From 2015 onward, performance stabilised with a consistent upward trend, culminating in high success rates by the end of the decade.

The improvement suggests effective refinement of both hardware and launch strategy.



All Launch Site Names

The query selecting distinct launch sites confirms that **CCAFS LC-40 (Cape Canaveral Air Force Station, Launch Complex 40)**, **VAFB SLC-4E (Vandenberg Air Force Base, Space Launch Complex 4E)**, and **KSC LC-39A (Kennedy Space Center, Launch Complex 39A)** were the primary locations used for operations, consistent with observed launch patterns and highlighting SpaceX's reliance on established coastal launch infrastructure.

```
%sql SELECT DISTINCT "Launch_site" FROM SPACEXTABLE
```

```
 * 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'

The total payload mass for NASA-related missions amounts to 99,980 kg, reflecting the agency's significant reliance on SpaceX services for heavy-lift capabilities.

```
%sql SELECT "Launch_site" FROM SPACEXTABLE WHERE "Launch_site" LIKE "CCA%" LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

The total payload mass for NASA-related missions amounts to 99,980 kg, reflecting the agency's significant reliance on SpaceX services for heavy-lift capabilities.

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Customer" LIKE "NASA%"
```

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

SUM(PAYLOAD_MASS__KG_)
99980

Average Payload Mass by F9 v1.1

The average payload mass carried by F9 v1.1 boosters is approximately 2,928.4 kg, illustrating the model's efficiency in handling medium-weight payloads with consistency.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE "Booster_Version"="F9 v1.1"
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

This screenshot shows the date of the first successful Falcon 9 landing on a ground pad, which marked a key milestone in SpaceX's reusable rocket programme.

```
%sql SELECT "Date", "Landing_Outcome" FROM SPACEXTABLE WHERE "Landing_Outcome" LIKE "Success%" LIMIT 1
```

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

Date	Landing_Outcome
2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

Here is a list of booster names that successfully landed on drone ships while carrying mid-weight payloads. Highlights launch consistency within this weight range.

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "Landing_Outcome"="Success (drone ship)" AND "PAYLOAD_MASS__KG_"> 4000 AND "PAYLOAD_MASS__KG_"< 6000
```

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

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

Total Number of Successful and Failure Mission Outcomes

Here is a summary of the total count of all successful and failed missions (regardless of the landing outcomes). This result offers a clear picture of SpaceX's overall mission success rate.

```
%sql SELECT Mission_Outcome, COUNT(*) AS Count FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

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

Boosters Carried Maximum Payload

The following query identifies the booster versions that carried the heaviest payloads. This demonstrates SpaceX's capability in high-load missions.

```
%sql SELECT "Booster_Version" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE);
```

```
 * 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

Here is the record of two failed drone ship landings occurred in 2015, along with the corresponding booster versions and launch sites.

This highlights early-stage challenges in achieving consistent drone ship recovery during that year.

```
%sql SELECT CASE WHEN substr("Date", 6, 2) = '01' THEN 'January' WHEN substr("Date", 6, 2) = '02' THEN 'February'WHEN substr("Date", 6, 2) = '03
```

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

Month_Name	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

This ranking of landing outcomes (e.g., success/failure types) shows progress in landing technology and reliability.

```
%sql SELECT Landing_Outcome, COUNT(*) AS Landing_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'GROUP BY Landing_Outcome
```

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

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

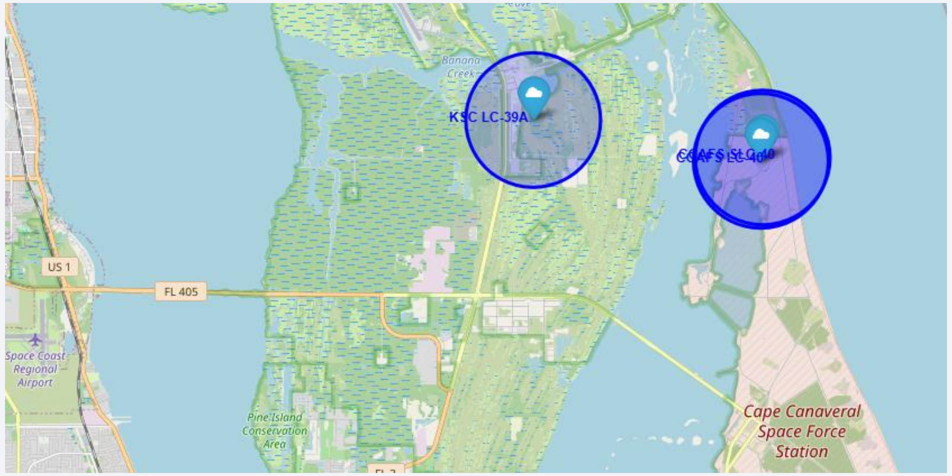
Section 3

Launch Sites Proximities Analysis

Launch Sites Locations in the US

The geographic distribution of the launch sites is a strategic advantage for SpaceX, giving them the flexibility to handle a wide variety of missions, reduce operational risk by spreading their launches across the country, and optimise their overall space launch strategy.

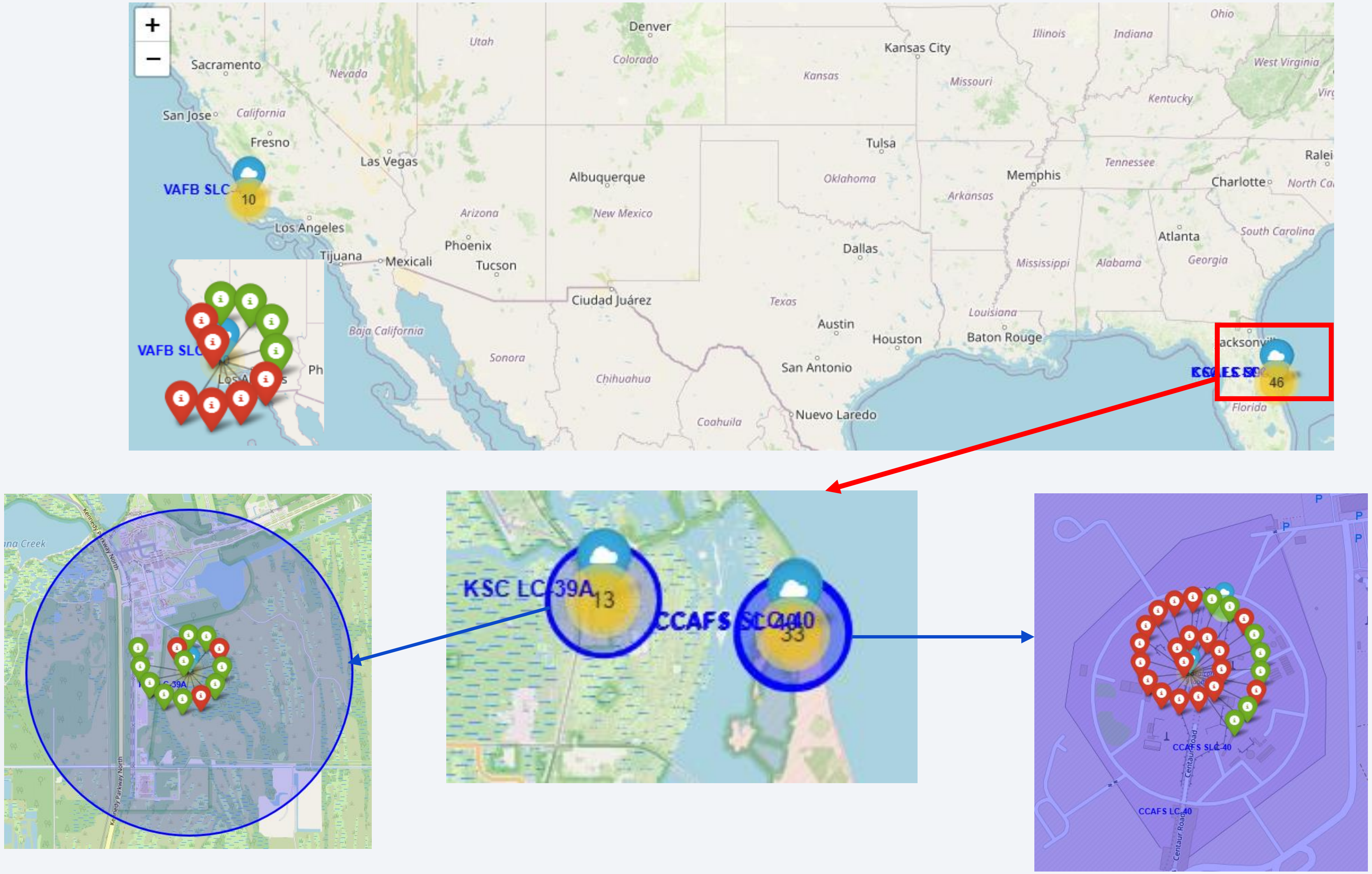
- ✓ **Florida Launch Sites (CCAFS LC-40, CCAFS SLC-40, KSC LC-39A)** are concentrated along the East Coast, taking advantage of the open ocean to the east for safer launches. The proximity of these sites suggests **coordination and infrastructure around the Cape Canaveral and Kennedy Space Centre area**, which has long been a major hub for space launches.



SpaceX Launch Map

SpaceX's launch sites show varying success rates.

CCAFS LC-40 had initial failures (red), while KSC LC-39A demonstrated higher success (green) over time. VAFB SLC-4E also saw strong performance, ideal for polar orbits.

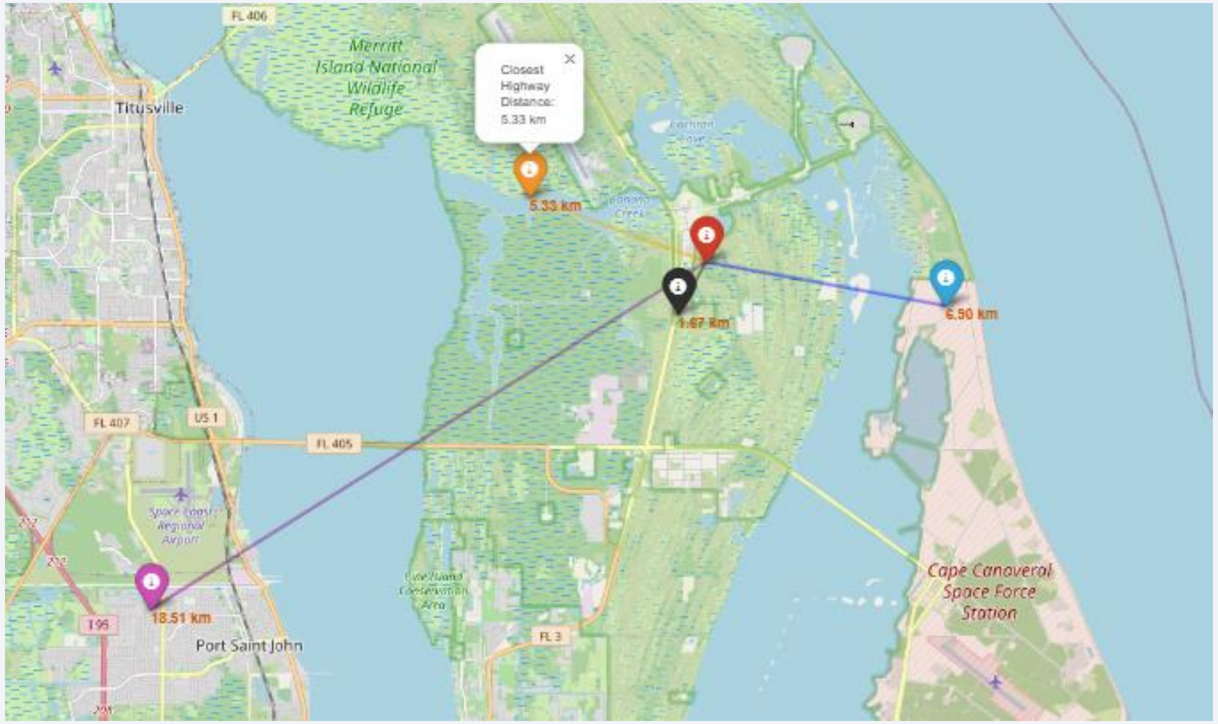


Interactive Map of KSC L39-A

The interactive map built with Folium visually highlights the proximity of Kennedy Space Center (KSC LC-39A) to key geographical features:

- **KSC (red) is located close to the coastline (blue)**, supporting safe rocket launches over the ocean.
- **Titusville (purple)**, the nearest city, lies to the west, showing the site's relative remoteness from dense population.
- **NASA Parkway and the NASA Railroad (black)** are both located nearby, indicating strong transport links critical for equipment and personnel access.

Distance markers and connecting lines effectively illustrate spatial relationships, enabling intuitive understanding of KSC's logistical and safety context.



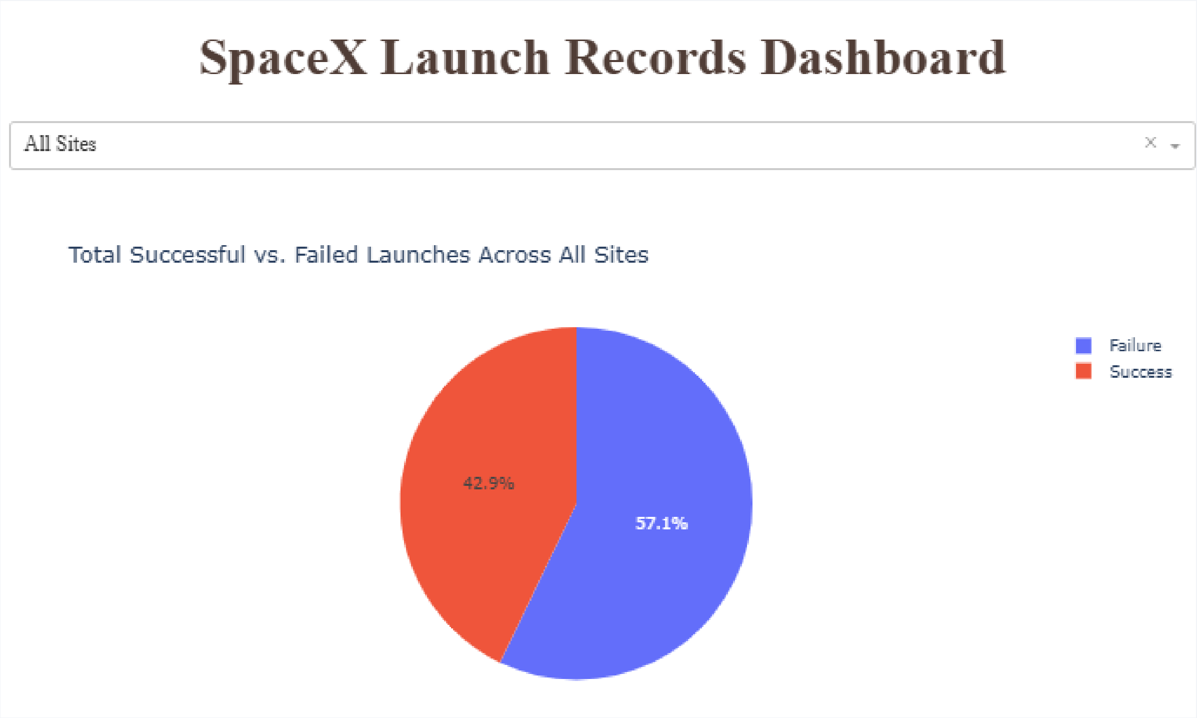
Section 4

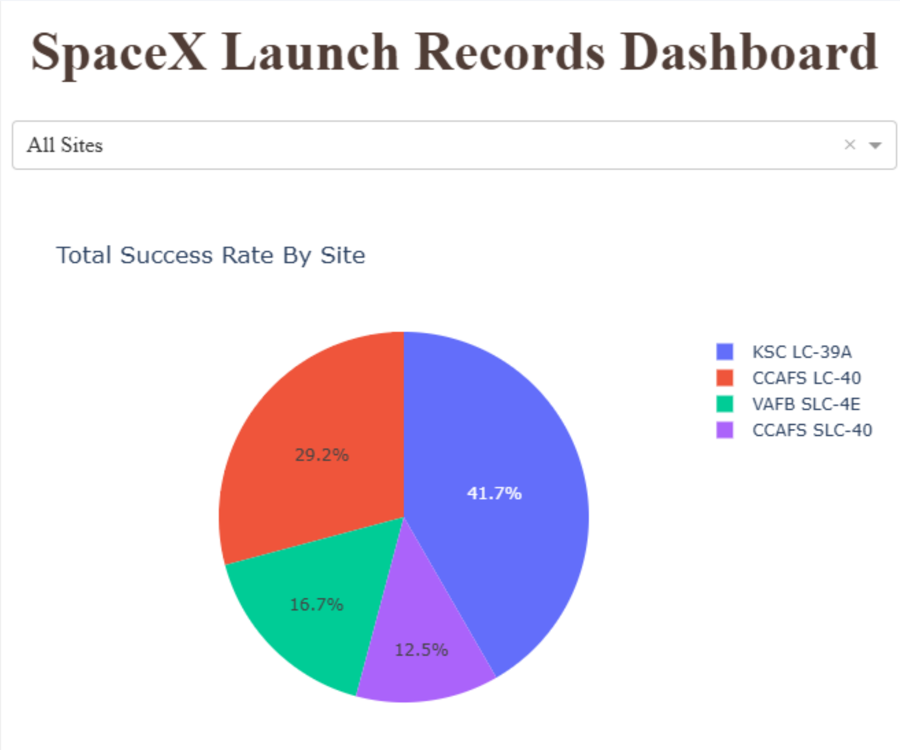
Build a Dashboard with Plotly Dash

Launch Successes vs. Failures

- Out of all launches across all sites, **57.1% were successful** and **42.9% resulted in failure**

The **Kennedy Space Center (KSC) Launch Complex 39A (LC-39A)** has the highest success rate of **41.7%** among all the sites, followed by **Cape Canaveral Space Force Station (CCAFS) Launch Complex 40 (LC-40)** with a success rate of **29.2%**. This suggests that KSC LC-39A is currently the most reliable site for SpaceX launches.

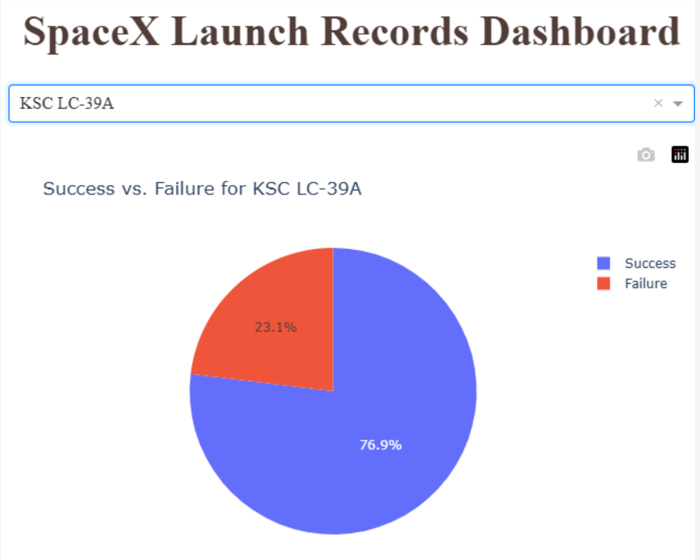


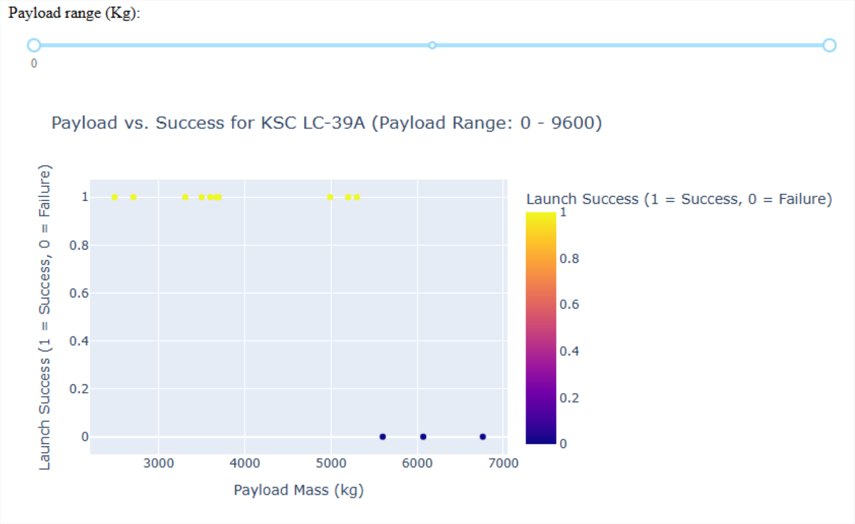


The Success of KSC LC-39A

KSC LC-39A has a notably higher **success rate** (76.9%) compared to the average across all launch sites (57.1%), suggesting that this site may have superior infrastructure or more optimized launch conditions.

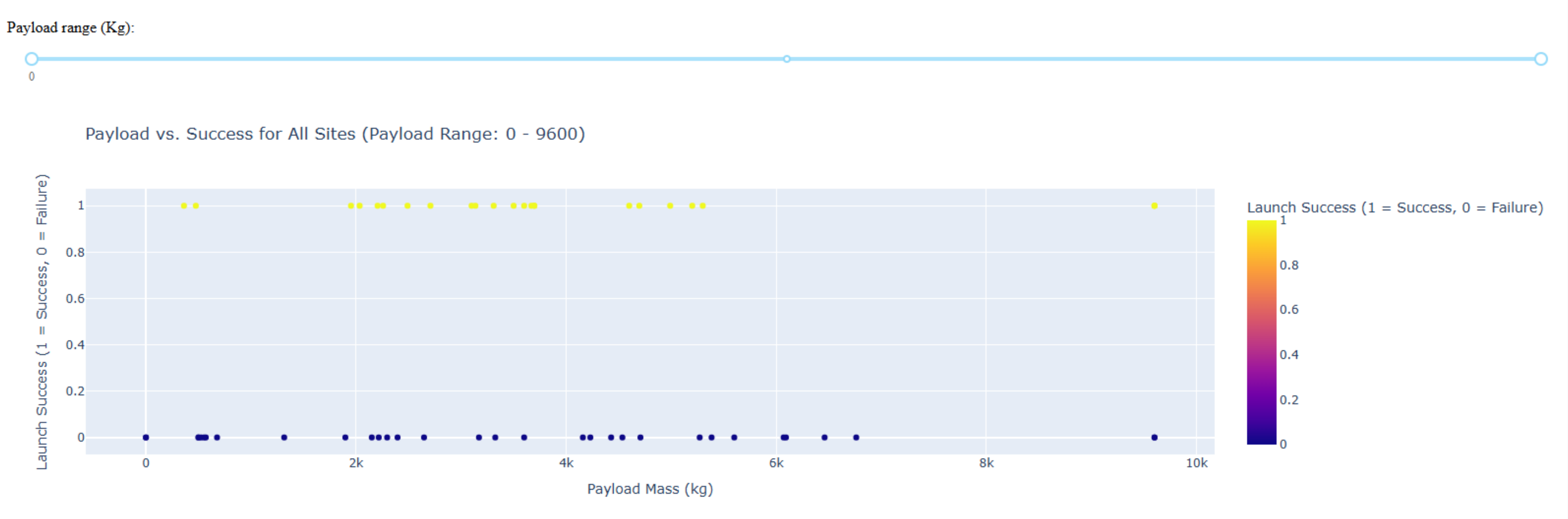
Additionally, launches with a **payload mass between 3000-4000 kg** appear to have a **higher success rate**, indicating that this payload range is particularly well-suited for SpaceX rockets, possibly due to better optimisation, engineering, or a favourable weight-to-power ratio for the rockets.





Correlation between Payload Mass and Mission Success

This scatter plot demonstrates that SpaceX launches with **payloads between 2,000 and 4,000 kg (2-4 metric tons) have the highest success rate.** This suggests an optimal range for SpaceX payloads in terms of mission success. Launches with payloads closer to 0 kg or near the maximum 9,600 kg (9.6 metric tons) have lower success rates. This indicates that there might be challenges in achieving successful launches with extremely light or very heavy payloads.



Section 5

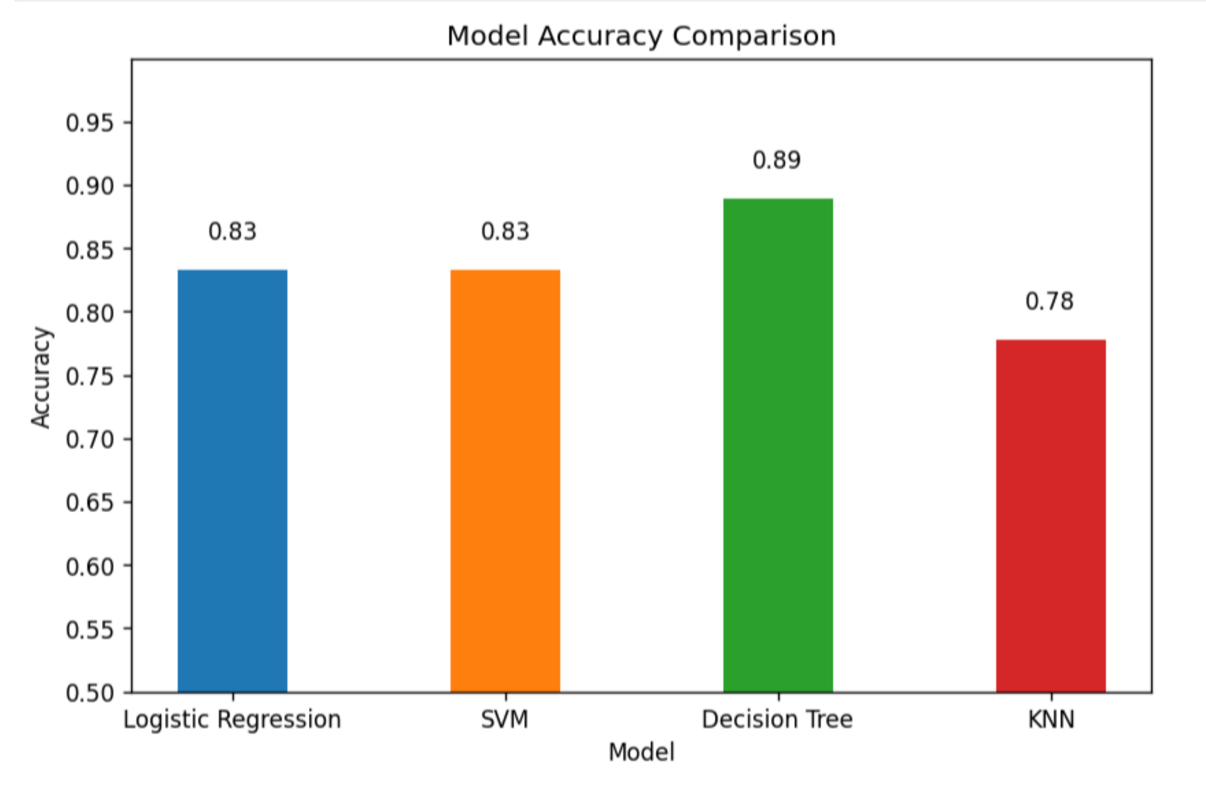
Predictive Analysis (Classification)

Classification Accuracy

The bar chart shows the test accuracy of four classification models: **Decision Tree, Logistic Regression, Support Vector Machine (SVM),** and **K-Nearest Neighbors (KNN)**.

The Decision Tree model achieved the highest classification accuracy (89%), indicating it was the most effective in predicting whether the Falcon 9 first stage would successfully land.

Logistic Regression and SVM performed well but slightly below the Decision Tree (83%), while KNN showed lower accuracy in comparison (78%).



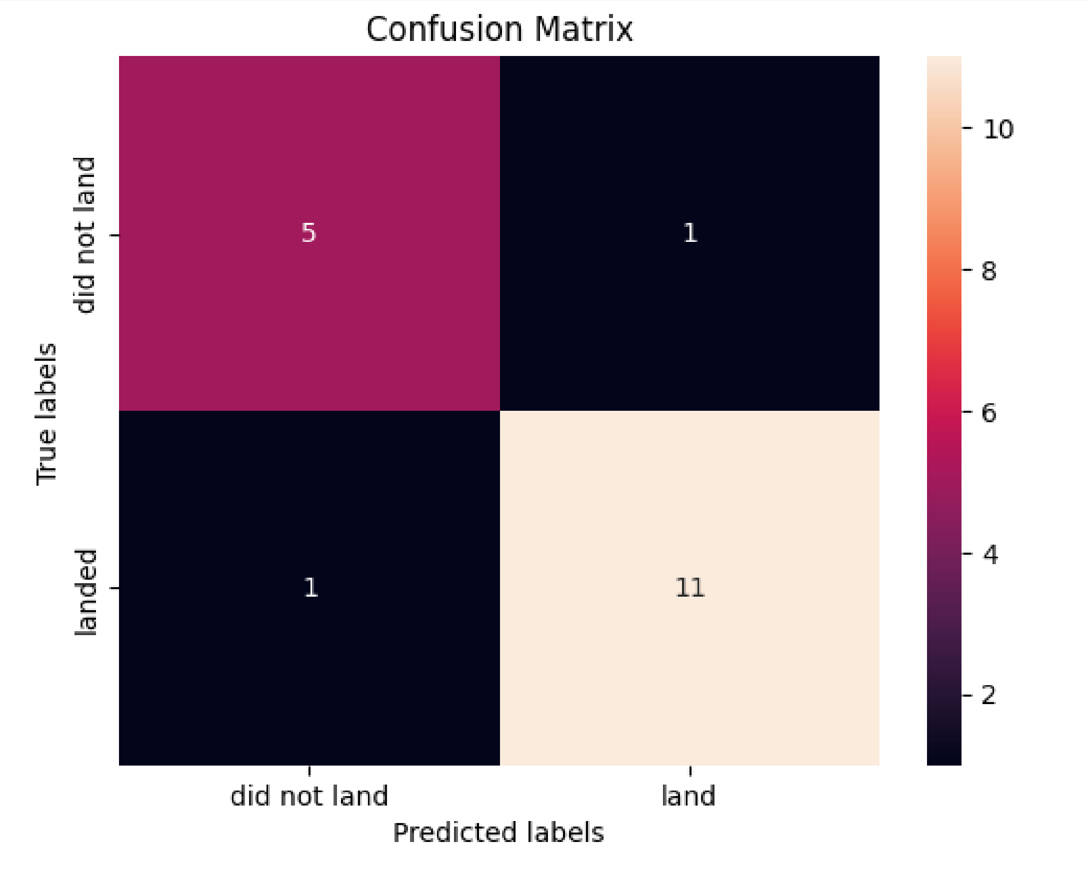
Confusion Matrix

Overview:

➢ Examining the confusion matrix, we see that logistic regression can distinguish between the different classes. We see that the problem is false positives.

✓ **True positive - 11** (True label is landed, Predicted label is also landed)

✓ **False positive - 5** (True label is not landed, Predicted label is landed)



Conclusions

This EDA analysis revealed key patterns in SpaceX launch performance and identified factors influencing mission success, offering guidance for future strategies.

- **KSC LC-39A** had the highest success rate.
- Optimal payload mass range: **3,000–4,000 kg**.
- **LEO** and **GEO** orbits had the most consistent outcomes.
- **Decision Tree** proved the most accurate model for prediction.

These findings can help SpaceX plan future launches more effectively by choosing the best launch sites, payload ranges, and predicting outcomes more reliably.

Thank you!