Winning Space Race with Data Science

Arpita Sonparote
10.07.2025

Outline

Executive Summary

Summary of methodologies

- Data Collection
- Data Wrangling
- Exploratory Data Analysis with Data Visualization
- Exploratory Data Analysis with SQL
- Interactive Visual Analytics with Folium
- Building Interactive Dashboard with Plotly Dash
- Predictive Analysis

Summary of all results

- Exploratory Data Analysis Results
- Interactive Analytics results
- Predictive Analysis Results

Introduction

Project background and context

- SpaceX has emerged as a leader in the commercial space industry by making space travel significantly more affordable. Their Falcon 9 rocket launches are listed at $62 million each, which is much lower than the $165 million often charged by other providers. A major reason for this cost efficiency is their ability to reuse the rocket's first stage. If we can predict whether the first stage will land successfully, we can estimate the overall cost of a launch. Using publicly available data and machine learning techniques, this project aims to build a model that can forecast the likelihood of the first stage being reused and if the SpaceX Falcon 9 first stage will land successfully.

Problems you want to find answers

- What are the total successful launches by different sites?
- Has the success rate of landings improved over the years?
- What is the correlation between payload mass and success for all sites?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Retrieved data through SpaceX’s REST API
  - Supplemented with additional information scraped from Wikipedia
- Perform data preparation and cleaning
  - Filtered and cleaned the raw data, handled missing or incomplete values
  - Applied One-Hot Encoding to convert categorical data for binary classification
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Evaluated model performance to select the one with the best predictive accuracy

Data Collection

1
- We made a get request to the SpaceX REST API
- Convert the response to a .json file then to a Pandas DataFrame

2
- Use custom logic to clean the data
- Define lists for data to be stored in
- Call custom functions to retrieve data and fill the lists
- Use these lists and values in a dictionary and construct the dataset

3
- Create a Pandas DataFrame from the constructed dictionary dataset

4
- Filter the DataFrame to only include Falcon 9 launches
- Reset the FlightNumber column
- Replacing missing values of PayloadMass with the mean PayloadMass value

5
- GitHub Notebook Link:
- https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/Data%20Collection%20API.ipynb

Data Collection – SpaceX API

1

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

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

```
json_data = response.json()
data = pd.json_normalize(json_data)
```

2

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

3

```
launch_df = pd.DataFrame(launch_dict)
```

4

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] != 'Falcon 1']
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

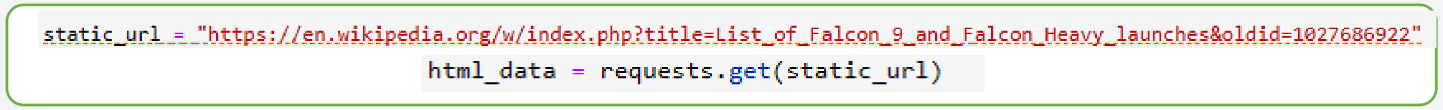
```
# Calculate the mean value of PayloadMass column
payload_mass_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean, inplace = True)
```

Data Collection - Scraping

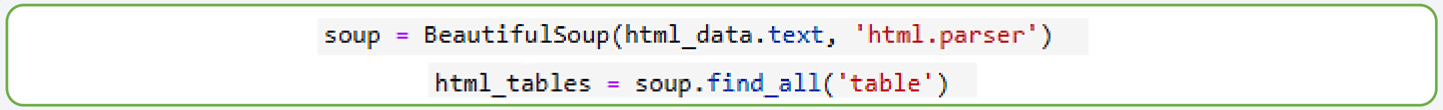
- Procedure Followed:
  I. Request the HTML page from the static URL
     - Assign the response to an object
  II. Create a BeautifulSoup object from the HTML response object
  III. Find all tables within the HTML page
  IV. Collect all column header names from the tables found within the HTML page
  V. Use the column names as keys in a dictionary
     - Use custom functions and logic to parse all launch tables to fill the dictionary values
  VI. Convert the dictionary to a Pandas DataFrame ready for export

- GitHub Link:

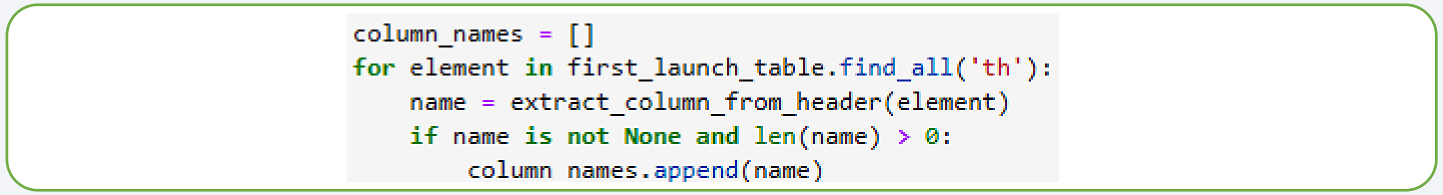
https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/Data%20Collection%20with%20Web%20Scraping.ipynb



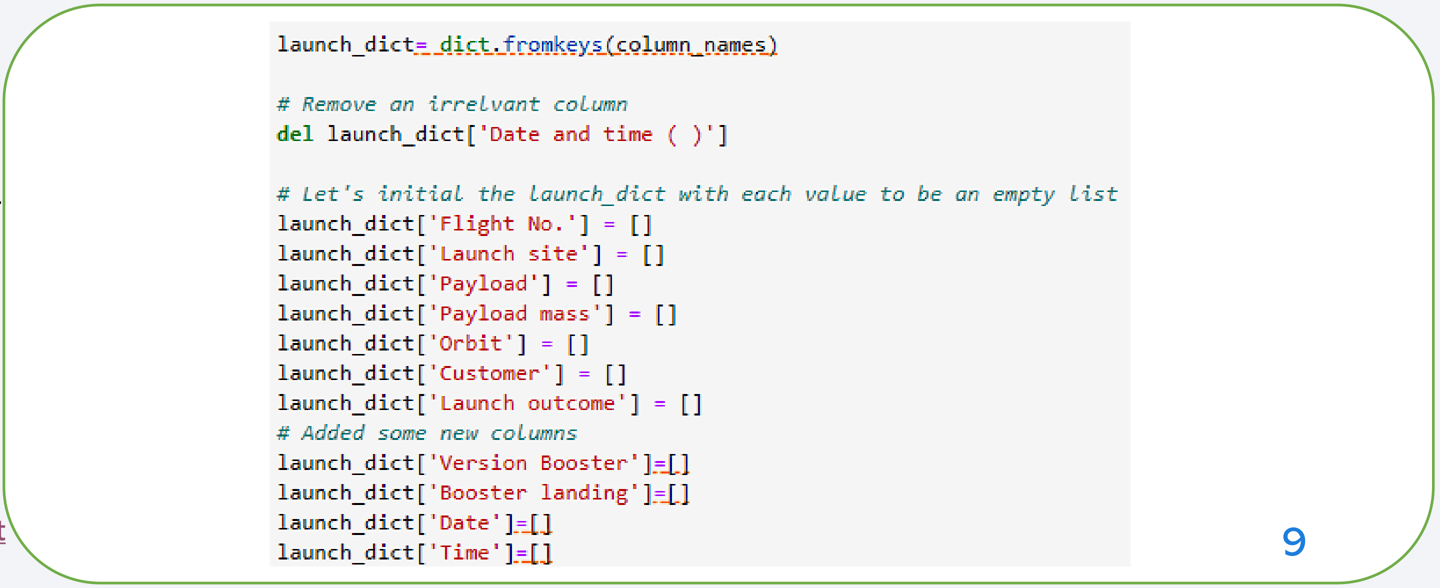
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
html_data = requests.get(static_url)
```



```
soup = BeautifulSoup(html_data.text, 'html.parser')
html_tables = soup.find_all('table')
```



```
column_names = []
for element in first_launch_table.find_all('th'):
    name = extract_column_from_header(element)
    if name is not None and len(name) > 0:
        column_names.append(name)
```



```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Data Wrangling

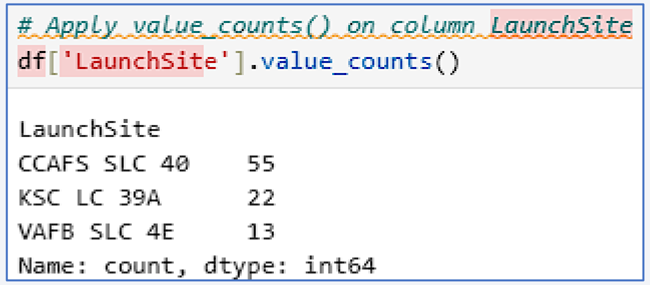
- The SpaceX dataset contains several Space X launch facilities, and each location is in the LaunchSite column.
- Each launch aims to a dedicated orbit, and some of the common orbit types are shown in the figure below. The orbit type is in the Orbit column.

1. Using the .value_counts() method to determine the following:
    - Number of launches on each site
    - Number and occurrence of each orbit
    - Number and occurrence of landing outcome per orbit type

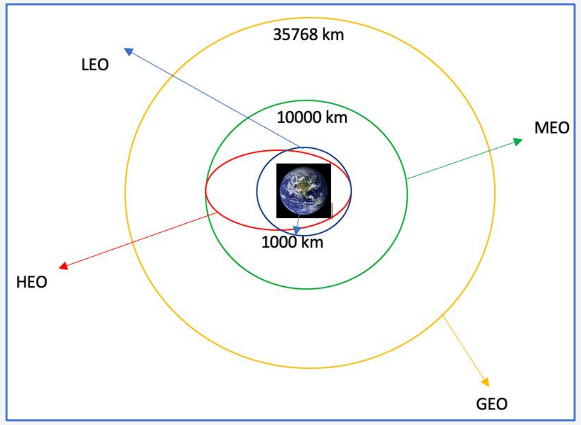
2. Create a landing outcome label from Outcome column

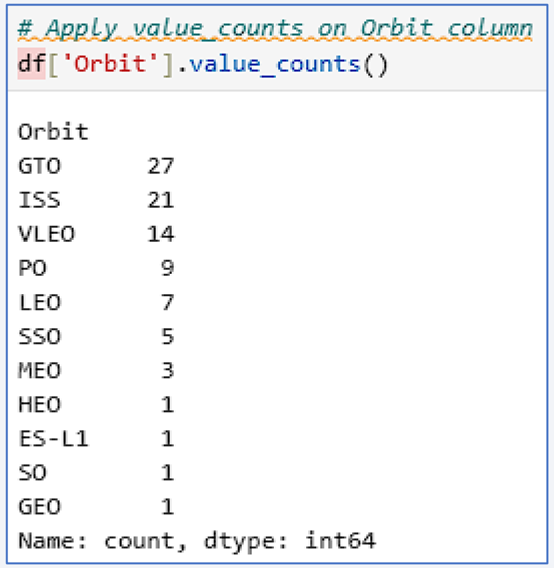
GitHub Link:

https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/Data%20Wrangling.ipynb

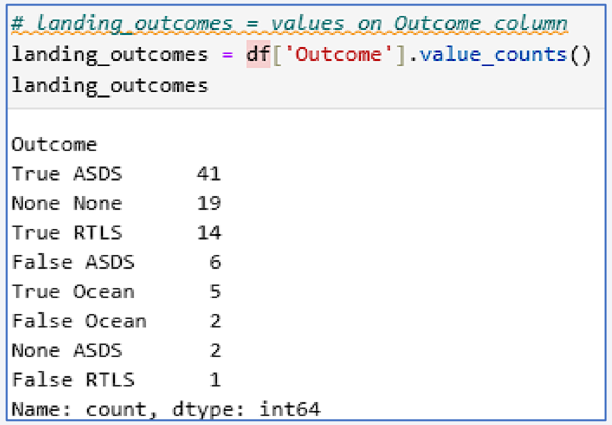


```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()

LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

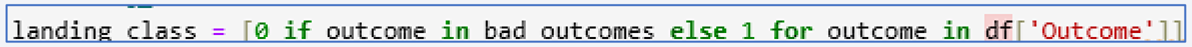




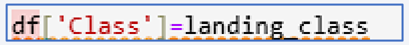
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()

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
```



```
# 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
```



```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
```



```
df['Class']=landing_class
```

EDA with Data Visualization

- Perform exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib
  1. Exploratory Data Analysis
  2. Preparing Data Feature Engineering

- Charts Used:
  1. Catplot function to visualize the relationship between Flight Number and Launch Site using scatter point chart
  2. Scatter point chart to visualize the relationship between Payload Mass and Launch Site
  3. Bar chart to visualize the relationship between success rate of each orbit type
  4. Scatter point chart to visualize the relationship between FlightNumber and Orbit type
  5. Scatter point chart to visualize the relationship between Payload Mass and Orbit type
  6. Line chart to visualize the launch success yearly trend

- Other tasks done:
  1. Create dummy variables to categorical columns
  2. Cast all numeric columns to float64

- GitHub Link:

https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/EDA%20with%20DAta%20Visualization.ipynb

EDA with SQL

- To gather some information about the dataset, some SQL queries were performed.
- The SQL queries performed on dataset were used to:
  - 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 succesful landing outcome in ground pad was acheived.
  - 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 all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function
  - 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

- GitHub Link:

https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- The following steps were taken to visualize the launch data on an interactive map
- **TASK 1:** Mark all launch sites on a map
  1. Initialize the map using a Folium Map object
  2. Add a folium.Circle and folium.Marker for each launch site on the launch map
- **TASK 2:** Mark the success/failed launches for each site on the map
  - As many launches have the same coordinates, it makes sense to cluster them together
  - Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red
  - To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object
  - Create an icon as a text label, assigning the icon_color as the marker_colour determined previously
- **TASK 3:** Calculate the distances between a launch site to its proximities
  - To explore the proximities of launch sites, calculations of distances between points can be made using the Lat and Long values
  - After marking a point using the Lat and Long values, create a folium.Marker object to show the distance
  - To display the distance line between two points, draw a folium.PolyLine and add this to the map

- GitHub Link:

https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- The following plots were added to a Plotly Dash dashboard to have an interactive visualization of data:
  - Pie chart (px.pie()) showing the total successful launches per site
    1. This makes it clear to see which sites are most successful
    2. The chart could also be filtered (using a dcc.Dropdown() object) to see the success/failure ratio for an individual site
  - Scatter graph (px.scatter()) to show the correlation between outcome (success or not) and payload mass (kg)
    1. This could be filtered (using a RangeSlider() object) by ranges of payload masses
    2. It could also be filtered by booster version
- Tasks Performed:
  - TASK 1: Add a Launch Site Drop-down Input Component
  - TASK 2: Add a callback function to render success-pie-chart based on selected site dropdown
  - TASK 3: Add a Range Slider to Select Payload
  - TASK 4: Add a callback function to render the success-payload-scatter-chart scatter plot
- GitHub Link:

https://github.com/Arpita368/Applied-Data-Science-Capstone/blob/45f69e83d2503be6dfe5769e1baf10da46fcab9b/spacex-dash-app.py

Predictive Analysis (Classification)

The following steps summarize how we built, evaluated, and selected the best model to predict SpaceX launch outcomes:

Data Preparation & Model Building

- Cleaned and transformed launch data (standardization, encoding)
- Split dataset into training and test sets using train_test_split()
- Chose suitable classification algorithms (e.g., Logistic Regression, SVM, Decision Tree, KNN)
- For each model, applied GridSearchCV to tune hyperparameters and improve performance

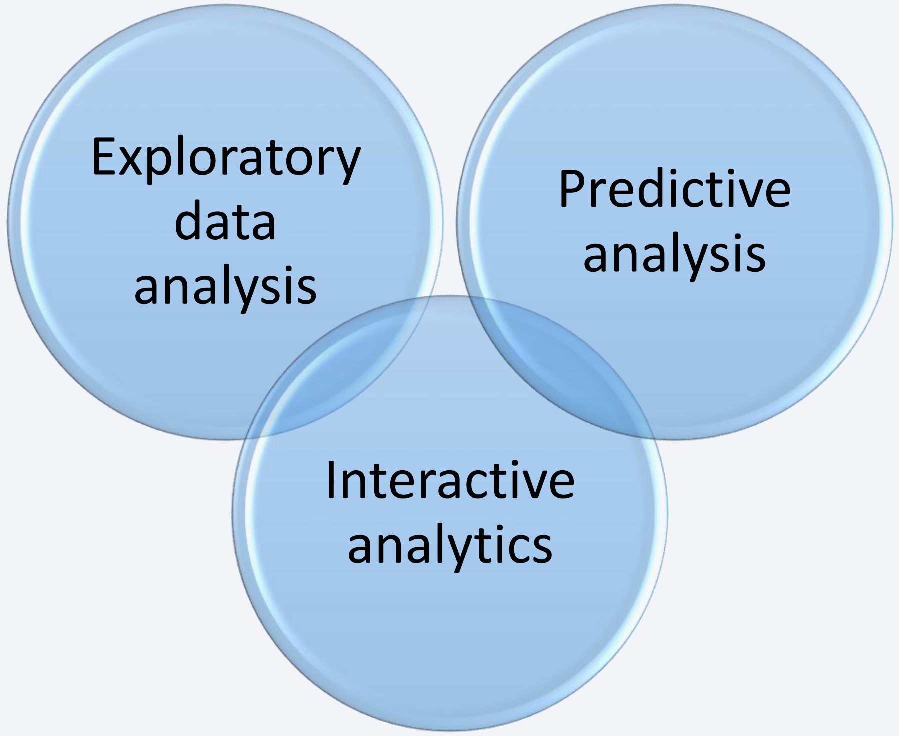
Model Evaluation

- Trained models on the training set and evaluated them using cross-validation
- Compared models based on their accuracy, precision, and recall scores
- Visualized the performance using confusion matrices and classification reports

Selecting the Best Model

- Reviewed accuracy scores across all models
- Identified the top-performing algorithm with the highest cross-validation accuracy
- This selected model is used for final predictions and insights

Results


Exploratory data analysis
Predictive analysis
Interactive analytics

SCALABLE PATH

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

Purpose:

To examine how frequently each launch site has been used over the sequence of SpaceX launches.

Explanation:

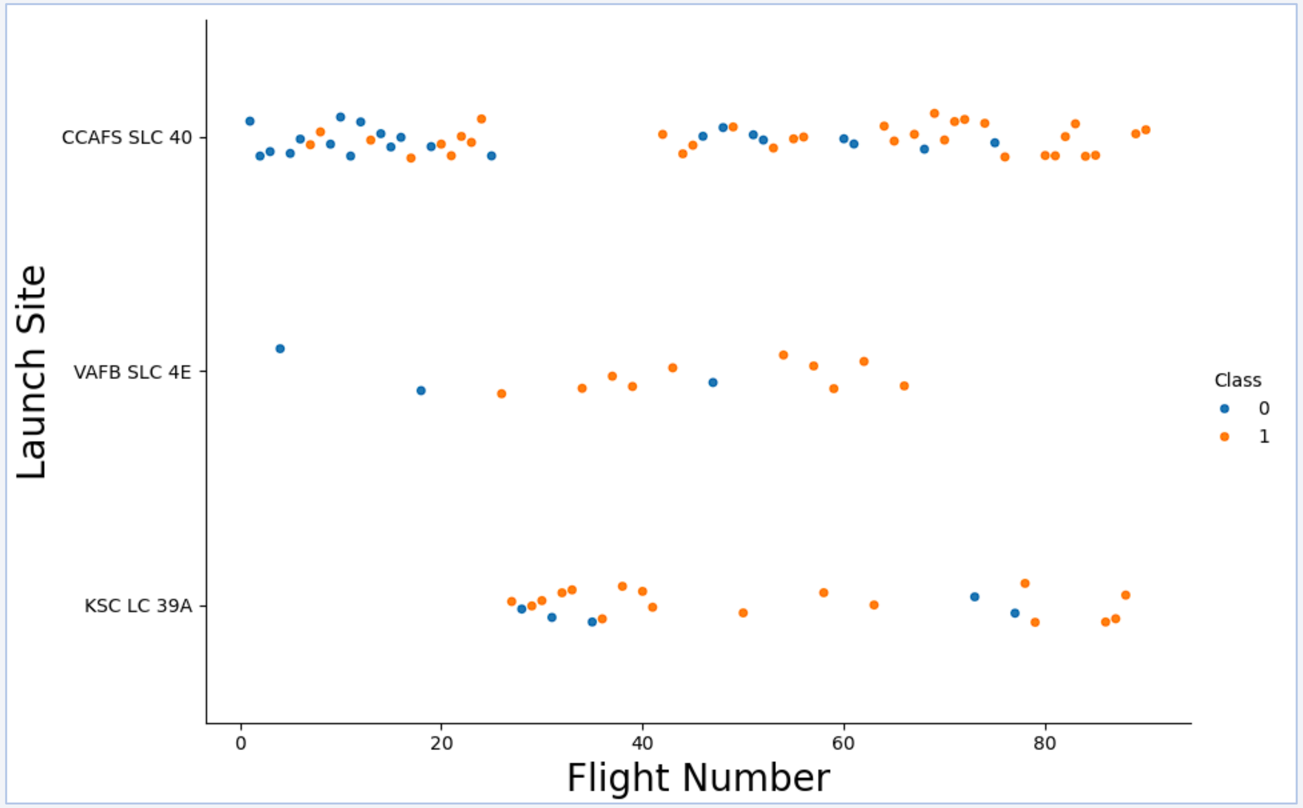
I. The chart shows that CCAFS SLC-40 has been used consistently from early to later flights, reflecting its reliability and central role in SpaceX operations.

II. KSC LC-39A appears more often in the later flight numbers, indicating its use for advanced or high-priority missions as technology matured.

III. VAFB SLC-4E has fewer launches, suggesting it's reserved for specific mission types.

IV. The overall upward pattern shows that with each successive flight, mission outcomes tend to improve a likely result of learning and refinement over time.

V. It also hints at site specialization, with some locations dedicated to frequent launches while others support occasional, specialized missions.



Payload vs. Launch Site

Purpose:
To analyze how payload size varies depending in the launch site

Explanation:

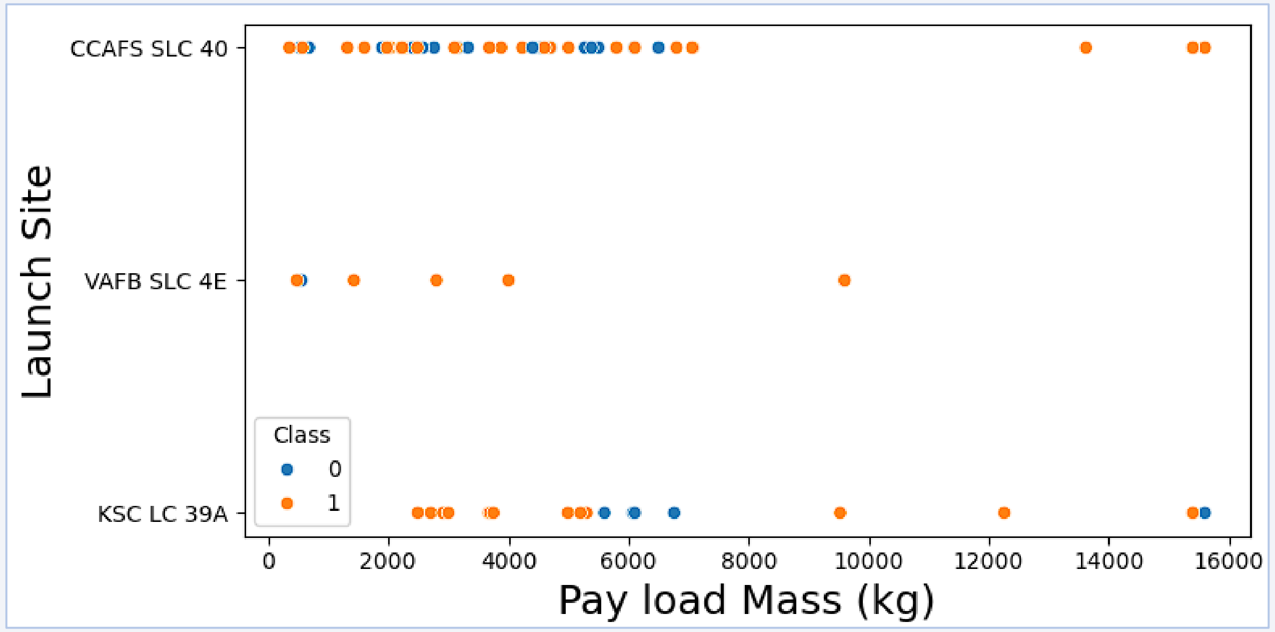
I. The plot indicates that KSC LC-39A often launches heavier payloads reflecting its capability to support large missions, possibly involving human spaceflight or high-value satellites.

II. CCAFS SLC-40 handles a wider range of payload masses, pointing to its versatility.

III. VAFB SLC-4E launches relatively lighter payloads, possibly due to geographic or structural constraints.

IV. Some outliers with very heavy payloads stand out these may be test missions or involve special-purpose satellites.

V. The plot supports the idea that **infrastructure, location, and mission type** drive payload allocation to each site.



Success Rate vs. Orbit Type

Purpose:
To compare how often missions to different orbits succeed, revealing which type of missions are more prone to failure.

Explanation:

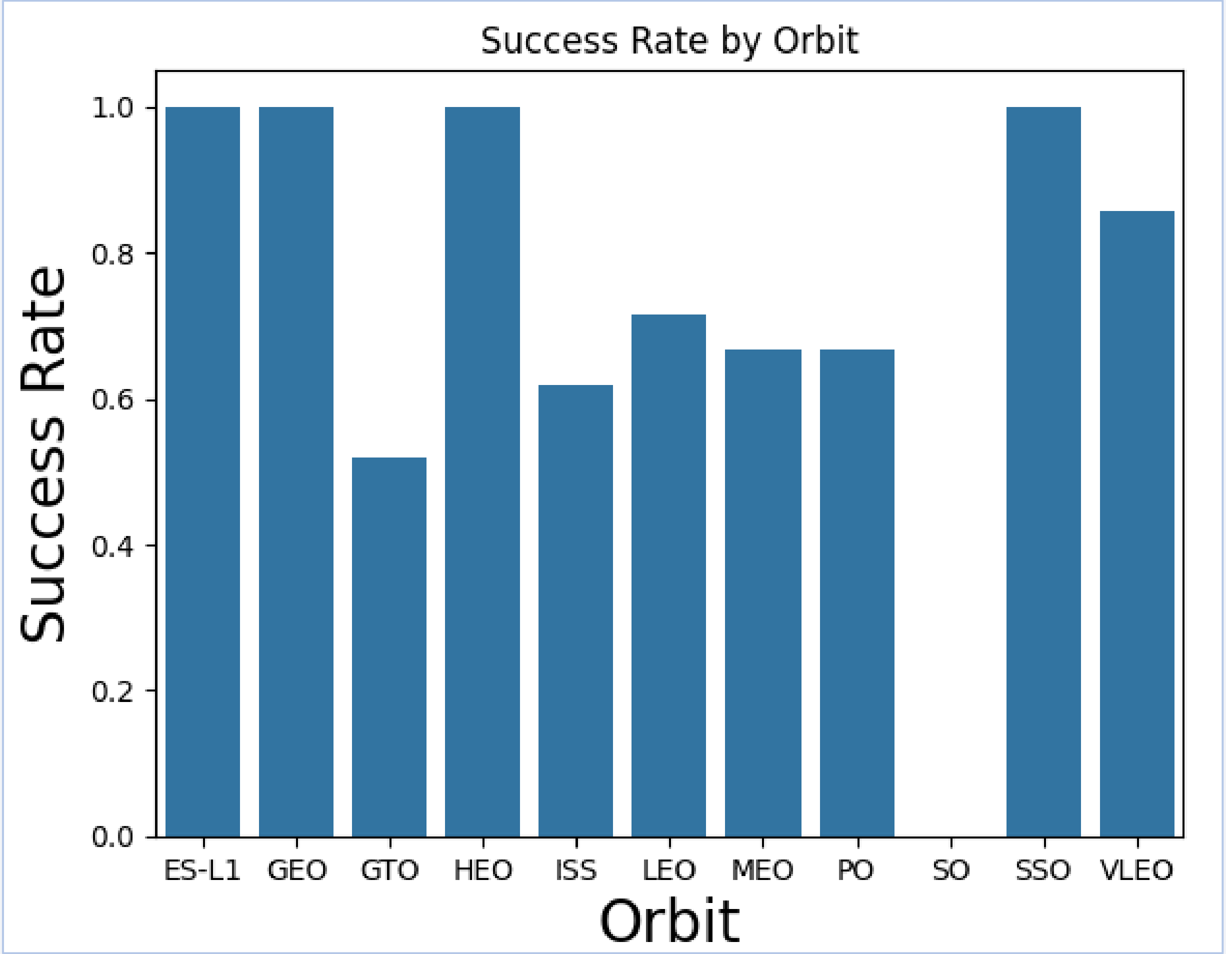
I. Orbits like GEO, ES-L1, HEO and SSO show high or even perfect success rates indicating mature technology and well-practiced mission profiles.

II. Other than GEO, ES-L1, HEO and SSO, LEO missions are also largely successful, likely due to their proximity and simplicity.

III. SO orbit has a noticeably lower success rate, suggesting greater technical difficulty or experimental nature.

IV. This breakdown helps identify which mission types are more reliable and where further improvements may be needed.

V. Additionally, newer orbits with low counts may distort the success rate if just one failure occurs, so it's important to consider the total number of launches per orbit.



Flight Number vs. Orbit Type

Purpose:
To explore how the choice of orbit has evolved throughout SpaceX's mission history.

Explanation:

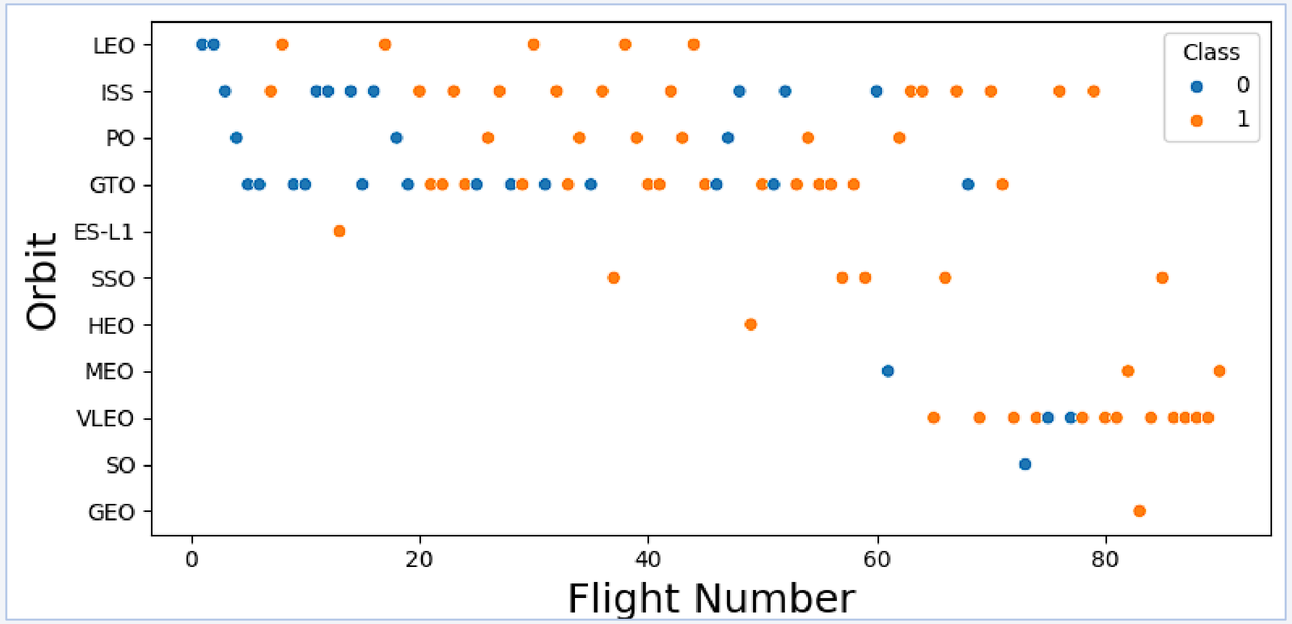
I. Early missions mostly targeted LEO, indicating initial focus on simpler, lower-risk missions.

II. Over time, flights to more complex orbits like GEO and SSO appear a sign of increasing technical capabilities.

III. Missions to the ISS begin in the middle flight numbers and continue steadily, reflecting NASA partnerships and resupply contracts.

IV. The transition in orbit types over the flight timeline shows SpaceX's growth from short-range to long-range and high-value missions.

V. This evolution also aligns with advancements in booster versions and reusability, which enable more ambitious orbital destinations.



Payload vs. Orbit Type

Purpose:
To explore how different orbit types are associated with variations in payload mass and success

Explanation:

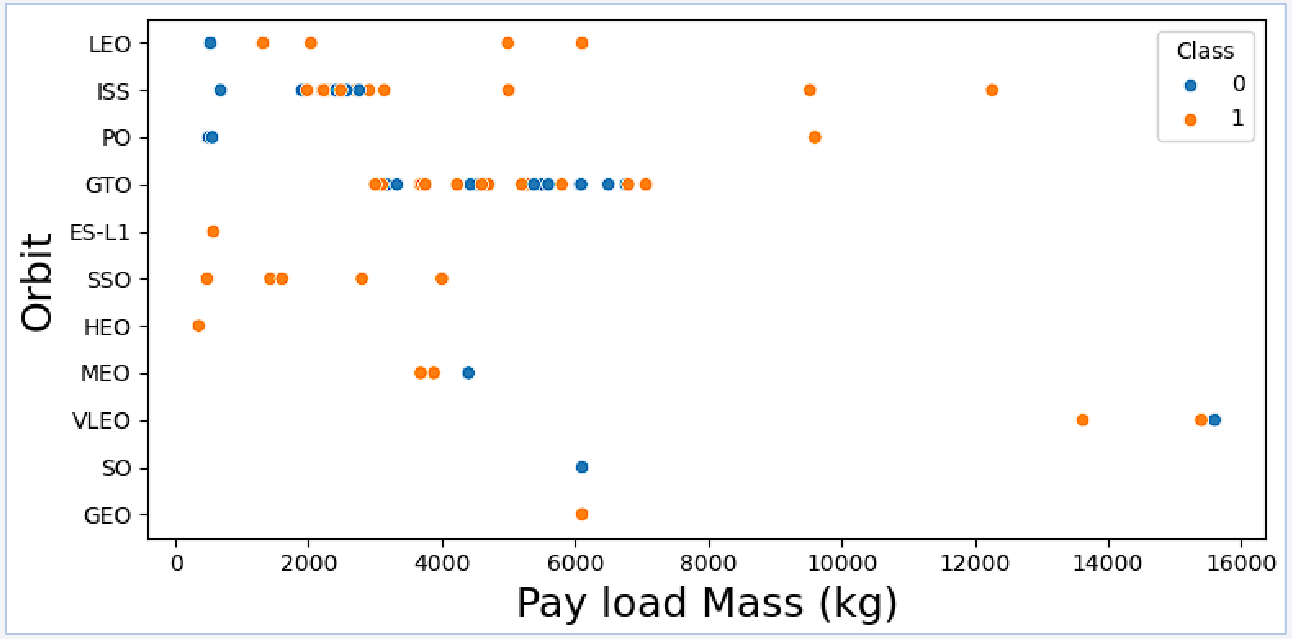
I. Certain orbit types like ISS, LEO, and PO tend to carry heavier payloads and show higher success.

II. Although PO has limited data, its missions appear to support heavy payloads.

III. For GTO (Geostationary Transfer Orbit), there is no clear trend between payload mass and success rate likely due to mission complexity.

IV. VLEO (Very Low Earth Orbit) missions also handle heavier payloads, consistent with expectations for short, low-orbit launches.

V. These trends support the idea that mission type and destination strongly influence payload mass.



Launch Success Yearly Trend

Purpose:
To understand how SpaceX's success rate has changed over time.

Explanation:

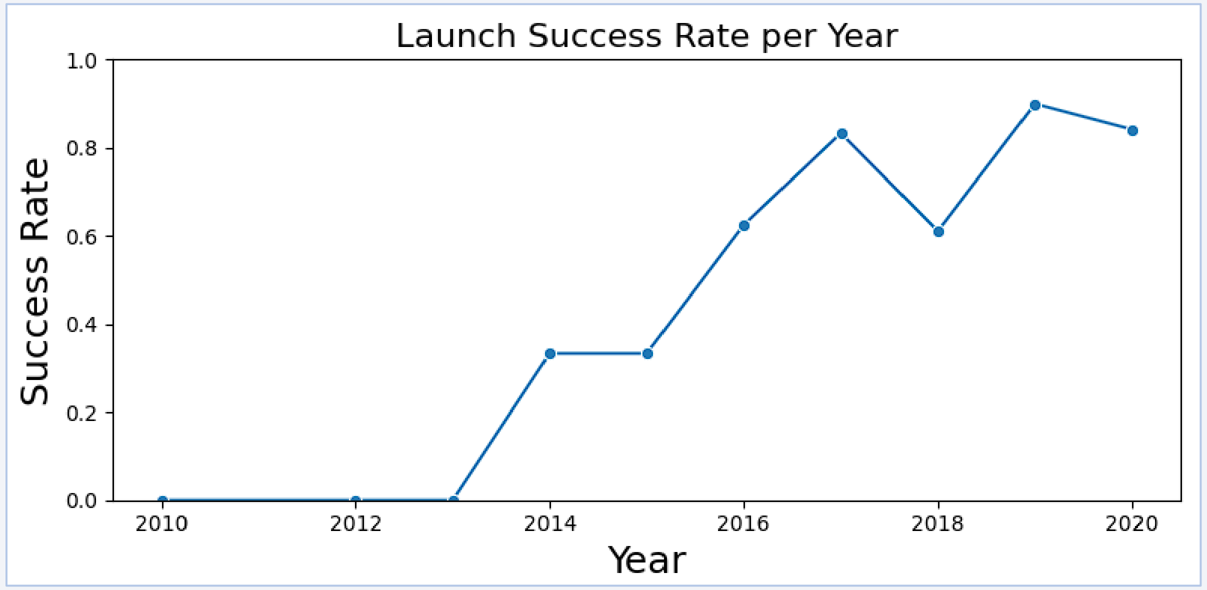
I. From 2010 to 2013, all launches failed or were unsuccessful, indicating an experimental phase.

II. Starting from 2014, the success rate began to improve significantly as technology and reliability matured.

III. There were brief setbacks in 2018 and 2020, but overall performance kept rising.

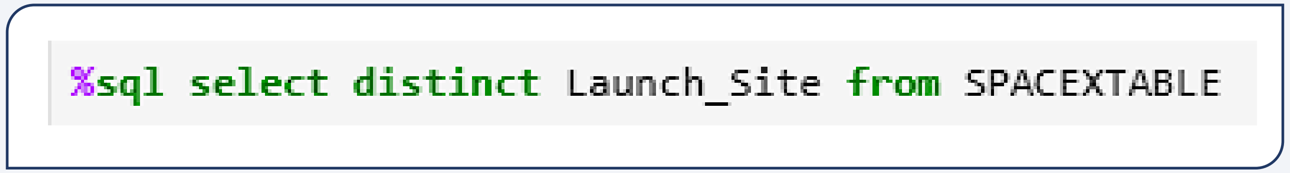
IV. Since 2016, SpaceX consistently achieved a >50% success rate, showing steady growth and operational excellence.

V. This upward trend reflects SpaceX's learning curve, process refinement, and innovation in launch operations.

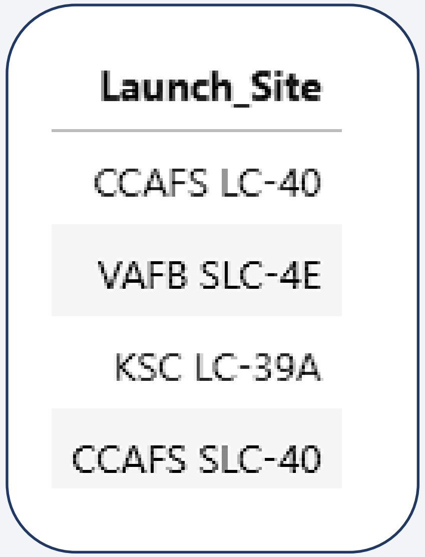


All Launch Site Names

- Find the names of the unique launch sites



```
%sql select distinct Launch_Site from SPACEXTABLE
```

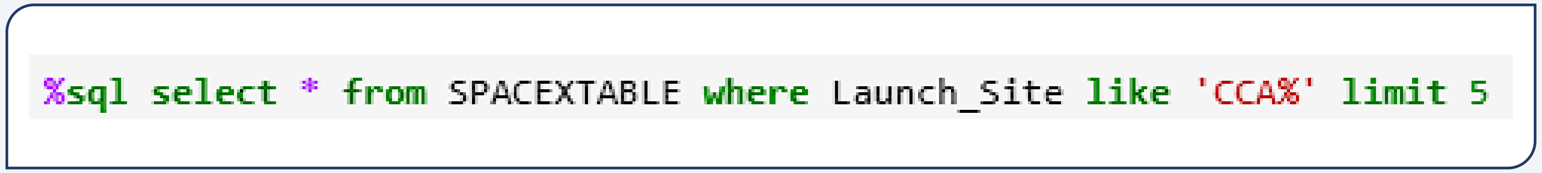


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

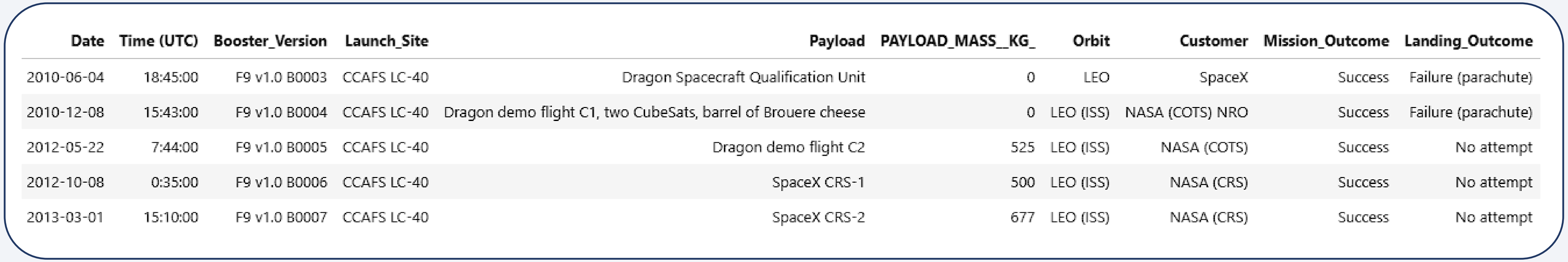
- **Explanation**: This query displays the names of unique launch sites. The DISTINCT keyword ensures that duplicate entries are removed, so each launch site appears only once in the result.

Launch Site Names Begin with 'CCA'

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



```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5
```

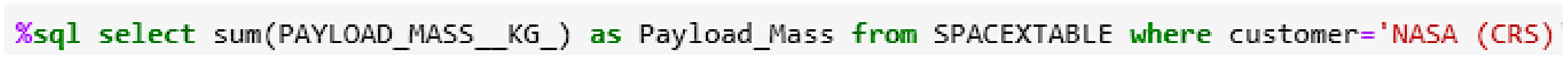


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

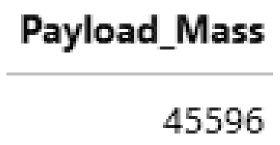
- **Explanation:** The ‘LIKE CCA%’ condition filters for launch sites beginning with ‘CCA’, and ‘LIMIT 5’ restricts the output to 5 rows.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA



```
%sql select sum(PAYLOAD_MASS__KG_) as Payload_Mass from SPACEXTABLE where customer='NASA (CRS)
```

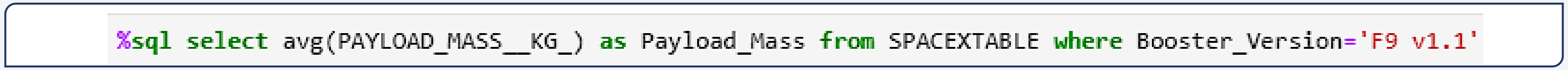


```
Payload_Mass
       45596
```

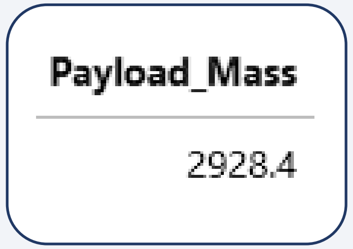
- **Explanation:** This query calculates the total payload mass (in kg) for launches where the customer was NASA (CRS).
- SUM(PAYLOAD_MASS_KG_) adds up the payloads.
- The WHERE clause filters records to only include ‘NASA (CRS)’ missions.
- The result is labelled as ‘Payload_Mass’.

Average Payload Mass by F9 v1.1

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



```
%sql select avg(PAYLOAD_MASS__KG_) as Payload_Mass from SPACEXTABLE where Booster_Version='F9 v1.1'
```

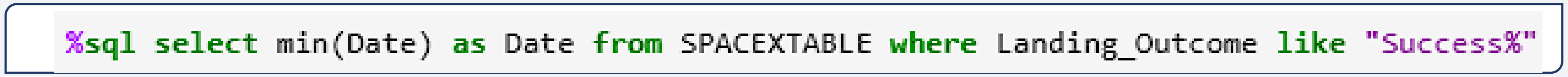


Payload_Mass
2928.4

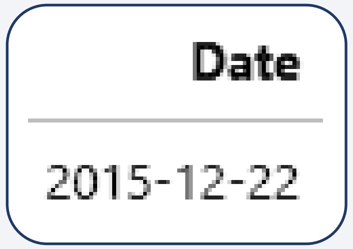
- **Explanation:** This query calculates the average payload mass (in kg) for all launches that used the F9 v1.1 booster version.
- AVG(PAYLOAD_MASS__KG_) returns the mean of the payload masses.
- The WHERE clause filters the dataset to include only rows with Booster_Version = 'F9 v1.1'.
- The result is labeled as 'Payload_Mass'.

First Successful Ground Landing Date

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



```
%sql select min(Date) as Date from SPACEXTABLE where Landing_Outcome like "Success%"
```

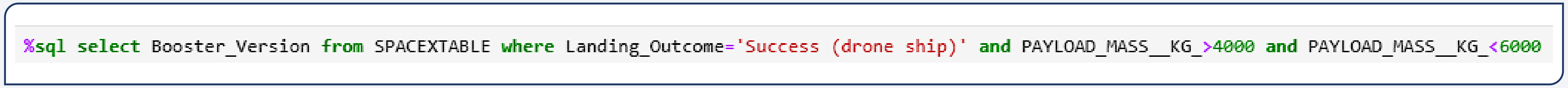


Date
2015-12-22

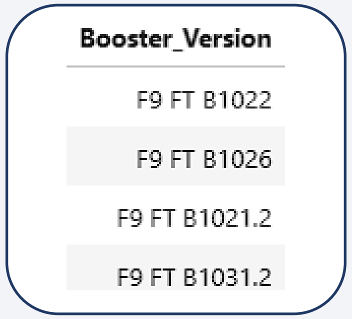
- **Explanation:** This query finds the earliest (minimum) launch date for which the landing was successful.
- MIN(Date) returns the earliest date.
- The WHERE clause filters rows where Landing_Outcome starts with "Success", covering outcomes like "Success (drone ship)", "Success (ground pad)", etc.
- The result is labeled as Date.

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



```
%sql select Booster_Version from SPACEXTABLE where Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_>4000 and PAYLOAD_MASS__KG_<6000
```

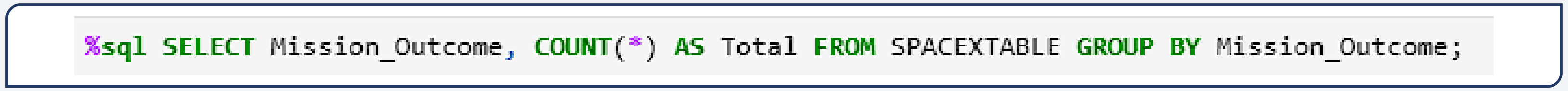


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

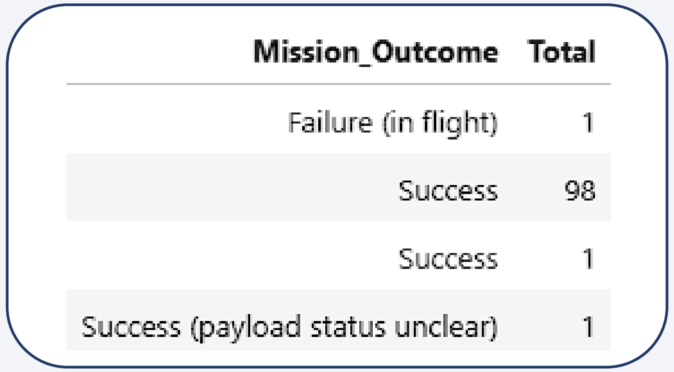
- **Explanation:** This query retrieves the booster versions used in launches that:
- Had a successful drone ship landing,
- Carried a payload mass between 4000 kg and 6000 kg.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes



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

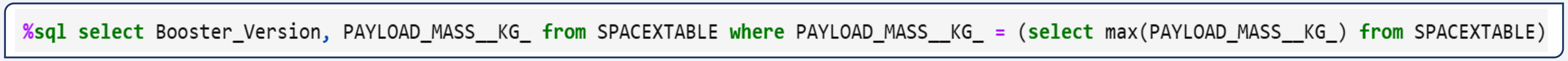


Mission_Outcome	Total
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

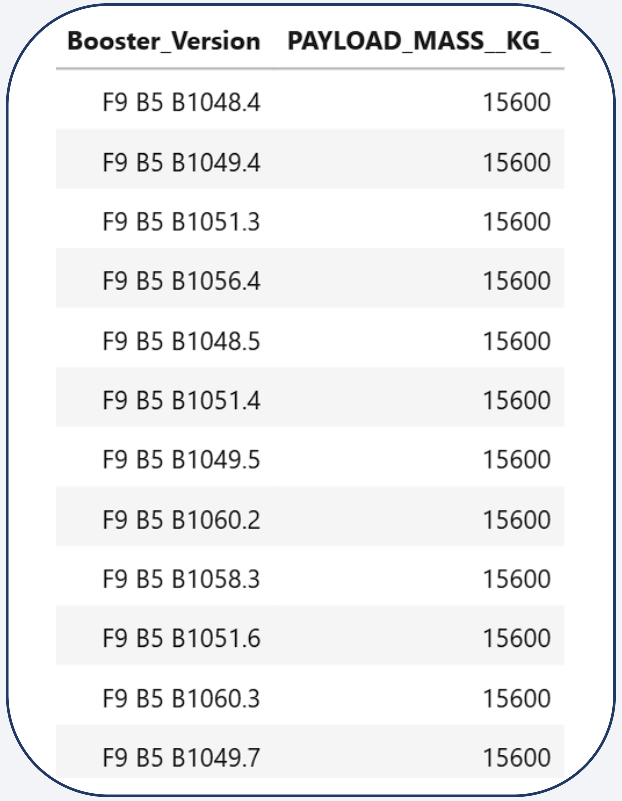
- **Explanation:** This query counts how many launches had each type of **mission outcome**.
- GROUP BY Mission_Outcome groups the records by unique outcome types.
- COUNT(*) counts how many times each outcome occurred.
- The result is labeled as Total for each mission outcome category.

Boosters Carried Maximum Payload

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



```
%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

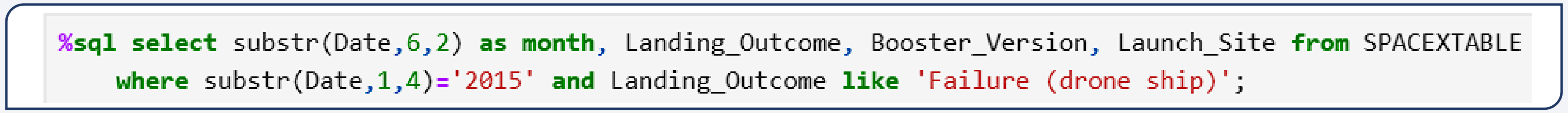


Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

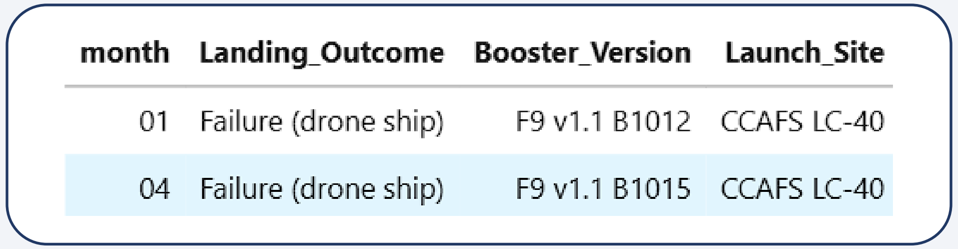
- **Explanation:** This query returns the booster version and the corresponding maximum payload mass ever launched in the dataset.
- The inner query SELECT MAX(...) finds the highest payload mass.
- The outer query fetches the booster version(s) that carried that maximum payload.

2015 Launch Records

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



```
%sql select substr(Date,6,2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE
    where substr(Date,1,4)='2015' and Landing_Outcome like 'Failure (drone ship)';
```

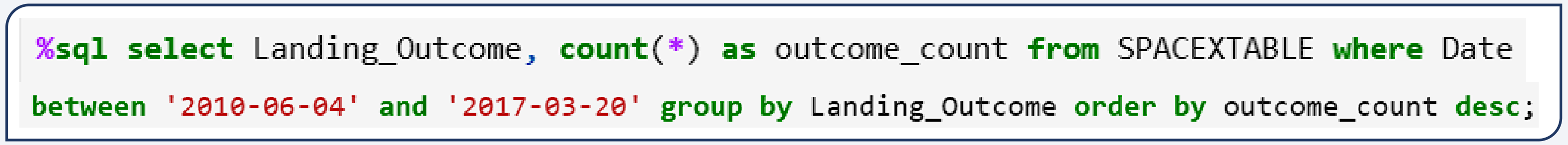


month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

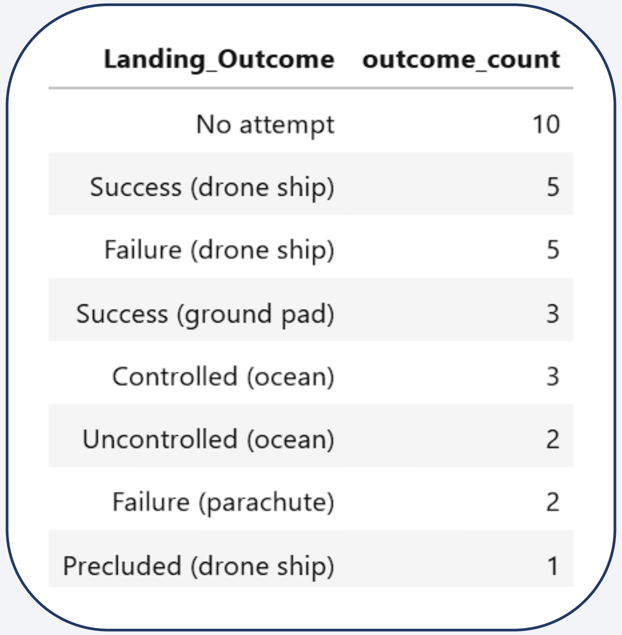
- **Explanation:** This query filters the dataset to show all failed drone ship landings that happened in 2015, and displays:
- The month of each failed landing (SUBSTR(Date,6,2) extracts the 2-digit month),
- The landing outcome (which will all match 'Failure (drone ship)'),
- The booster version used and the launch site where the launch took place.

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



```
%sql select Landing_Outcome, count(*) as outcome_count from SPACEXTABLE where Date
between '2010-06-04' and '2017-03-20' group by Landing_Outcome order by outcome_count desc;
```



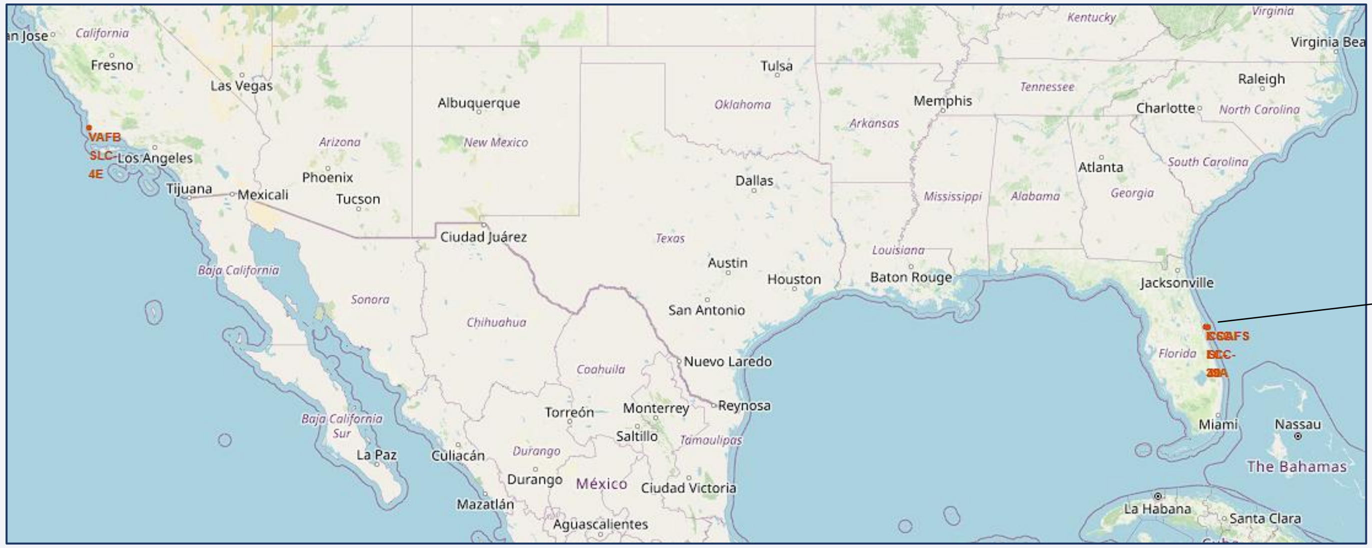
Landing_Outcome	outcome_count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

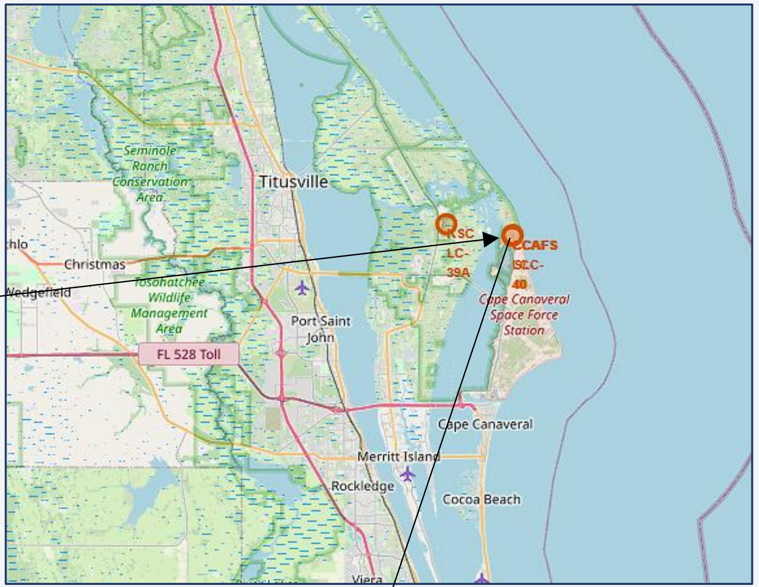
- **Explanation:** This query counts how many times each landing outcome occurred between June 4, 2010 and March 20, 2017.
- COUNT(*) counts the number of records for each landing outcome.
- GROUP BY Landing_Outcome groups results by unique landing outcomes.
- ORDER BY outcome_count DESC sorts the results from most to least frequent.

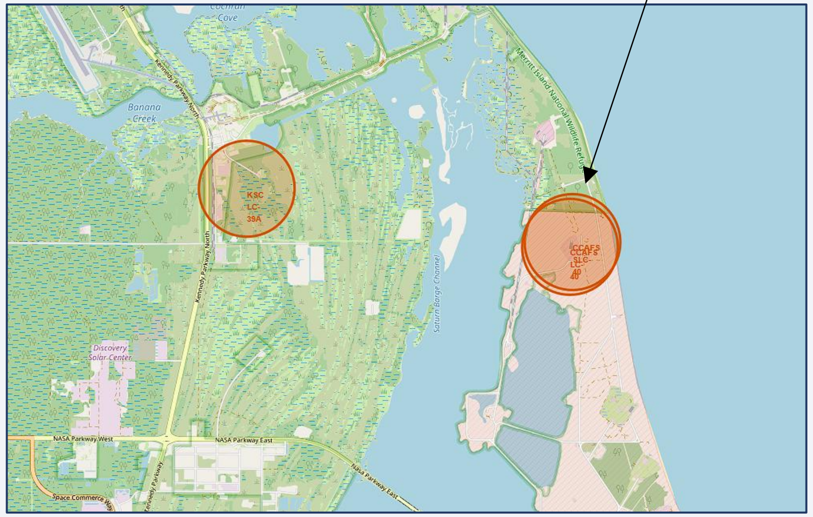
Section 3

Launch Sites Proximities Analysis

All Launch Sites on Map

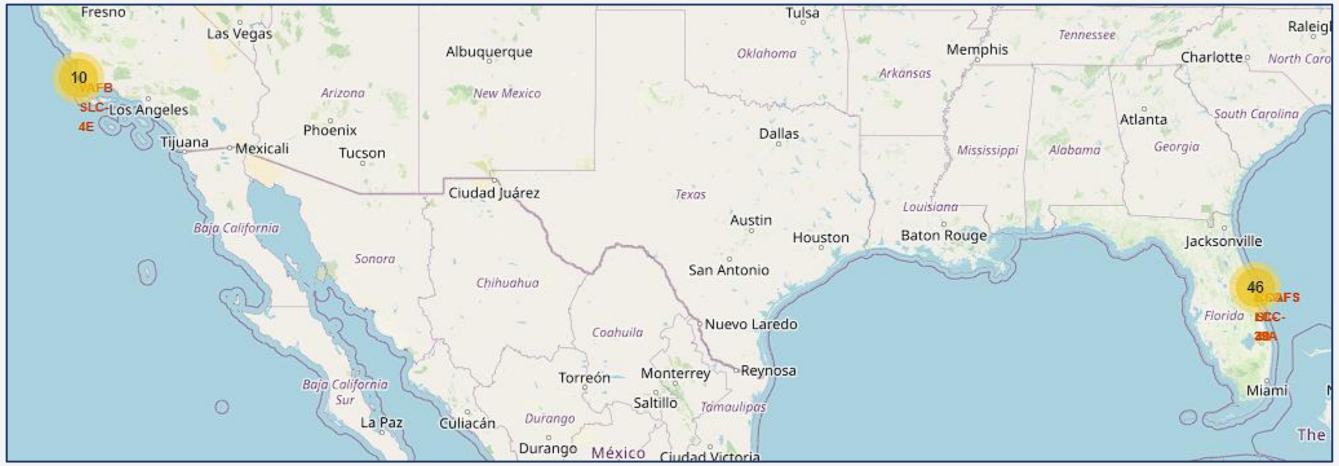




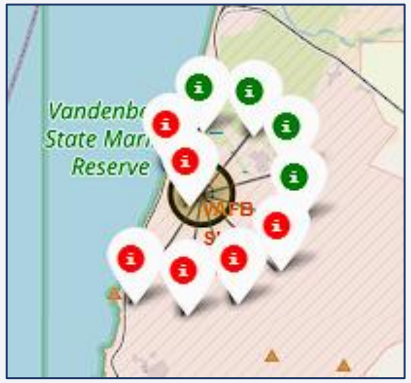


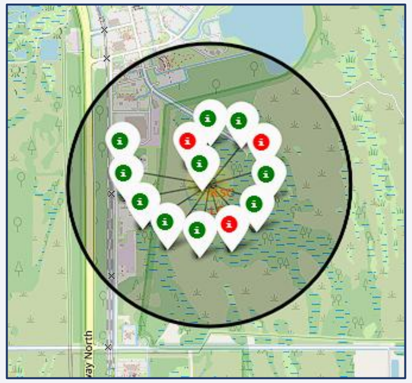
All SpaceX's launch sites are on the coast of the United States of America, specifically Florida and California

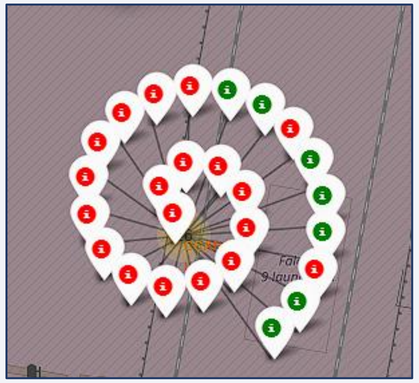
All The Success/ Failed Launches for Each Site

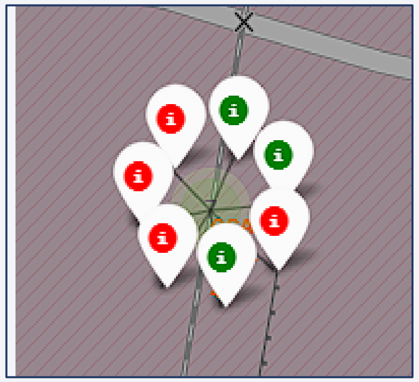


Launches have been clustered into clusters, successful launches are indicated by green icons while the failed launches are indicated by red icons.


VAFB SLC-4E

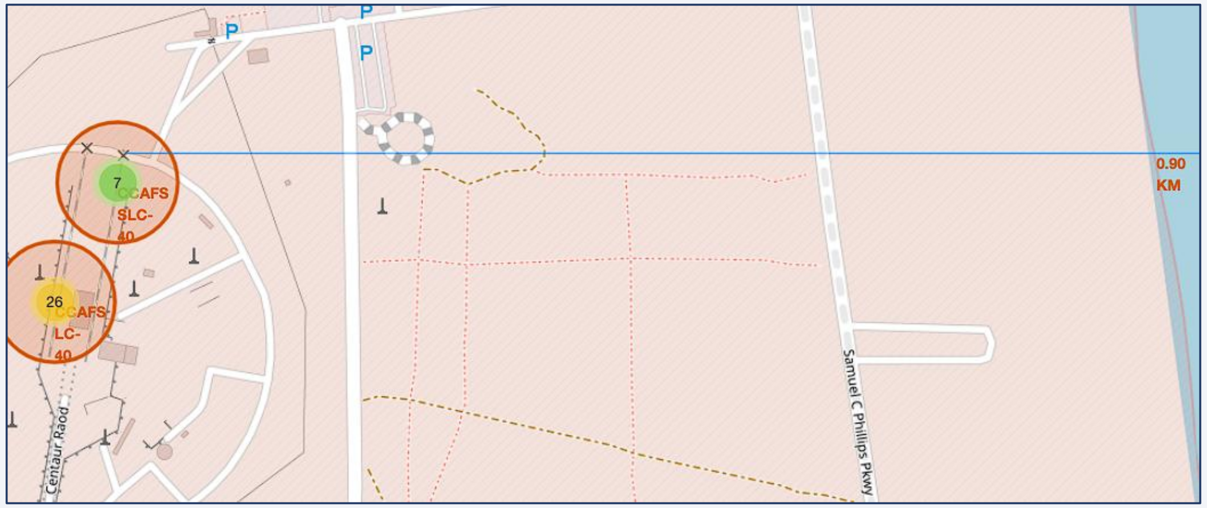

KSC LC-39A

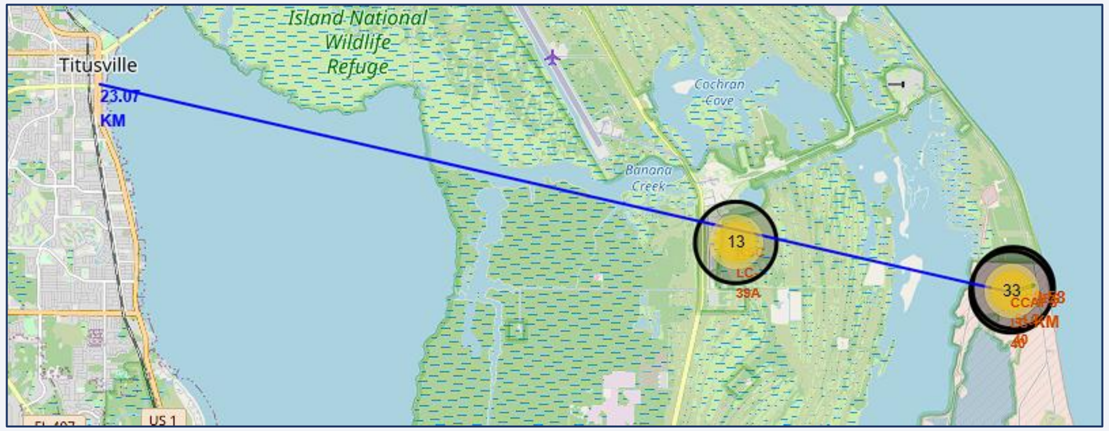

CCAFS SLC-40

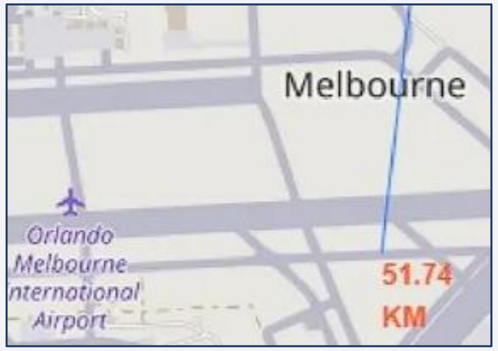

CCAFS SLC-40

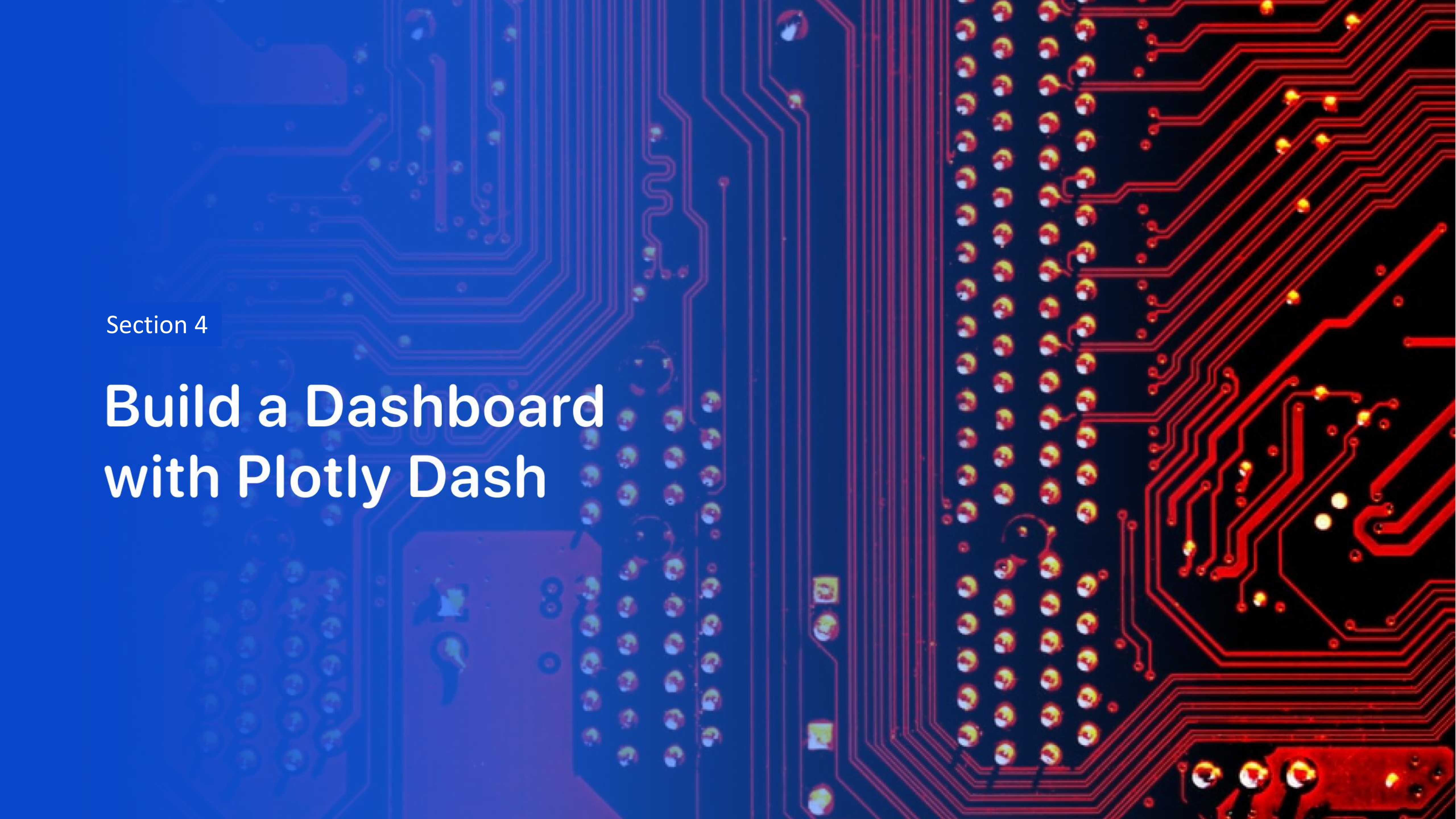
Proximities of Launch Sites to Other Points of Interest

The distances between a launch site to its proximities such as the nearest city, railways, coastline, highways:





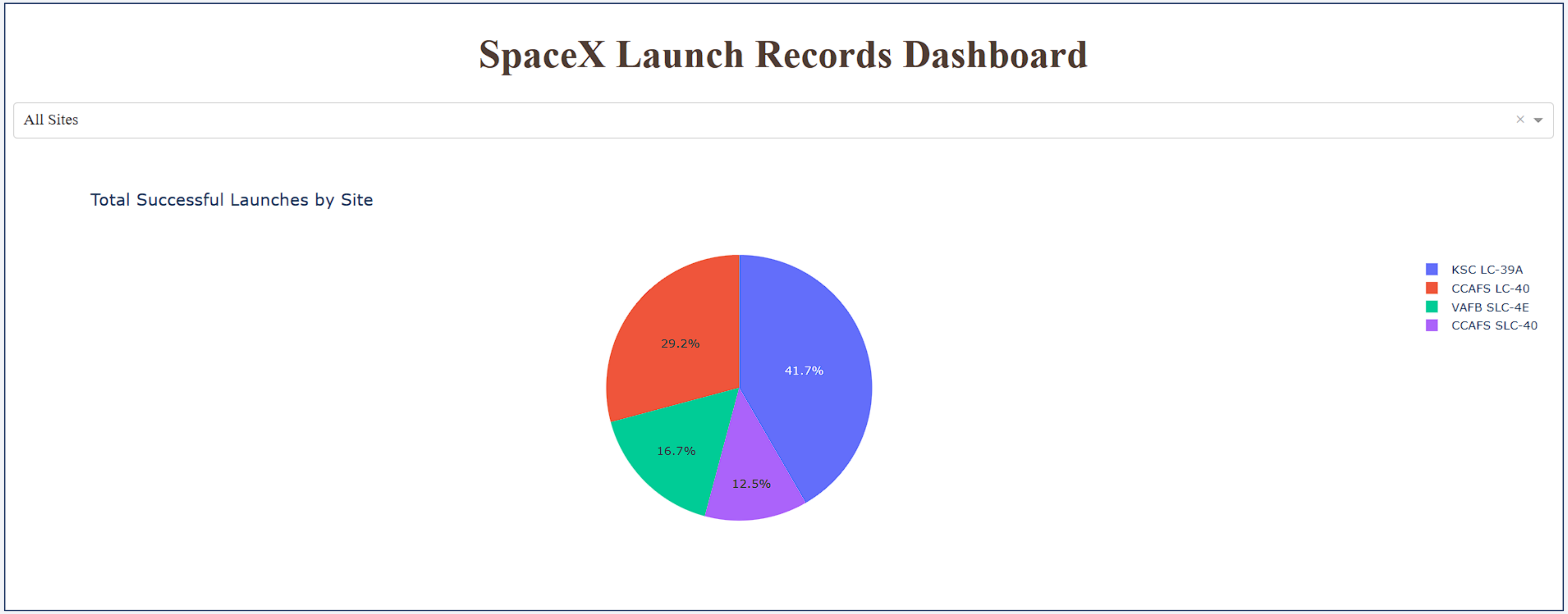




Section 4

Build a Dashboard with Plotly Dash

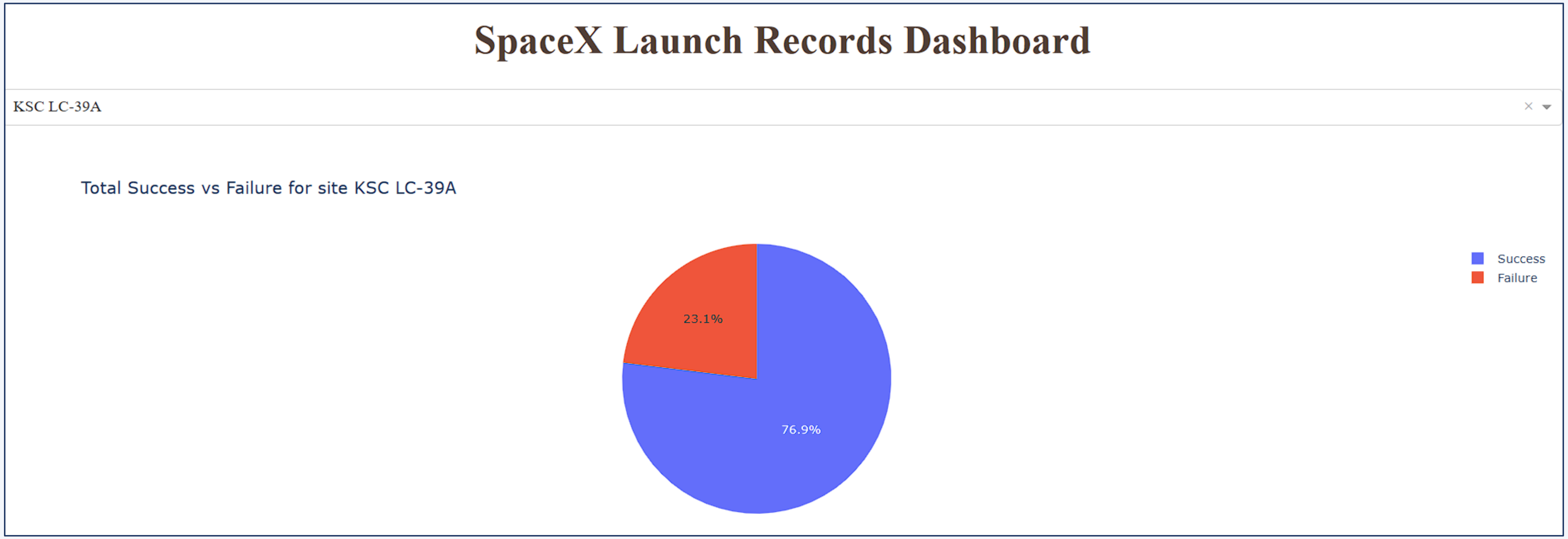
Total Successful Launches by Sites


SpaceX Launch Records Dashboard

All Sites

The launch site KSC LC-39A had the most successful launches, with a success rate of 41.7%

Successful and Failed Launches for KSC LC-39A

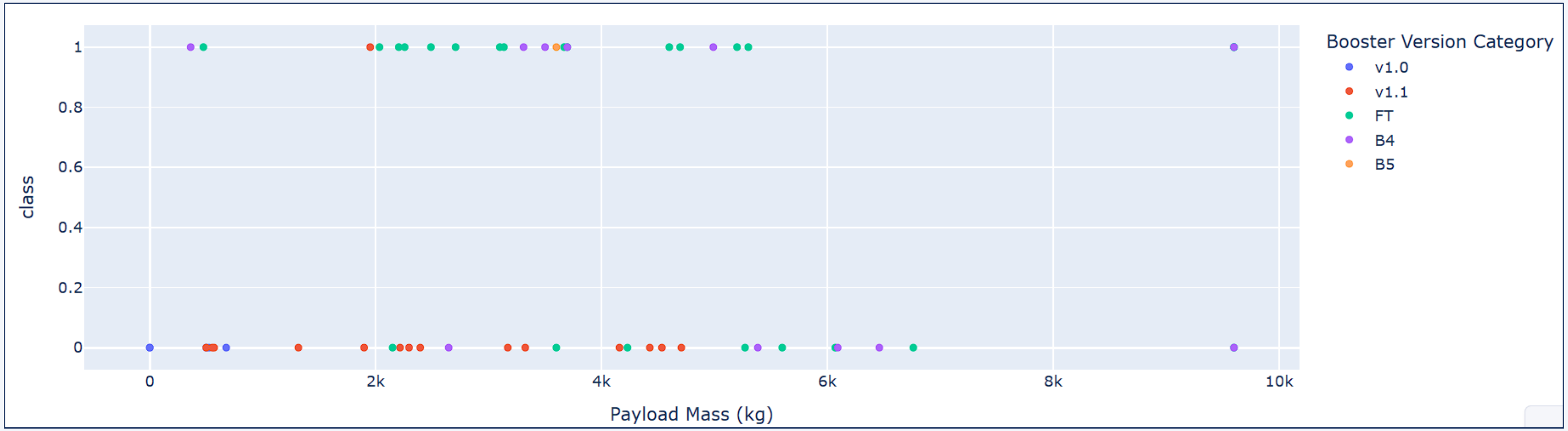

SpaceX Launch Records Dashboard

KSC LC-39A

Total Success vs Failure for site KSC LC-39A

The launch site KSC LC-39A had the success rate of 76.9% while the failure rate is only 23.1%

Correlation Between Payload and Success for All Sites



- Launches with low payloads (<4000 kg) show a higher success rate compared to launches with massive payloads (>4000 kg).
- Some booster versions like v1.0 and B5 have not been used for heavy payload launches, which may suggest capability or reliability limitations.
- There appears to be a correlation between increased payload and decreased launch success, highlighting the challenge of launching heavier payloads.

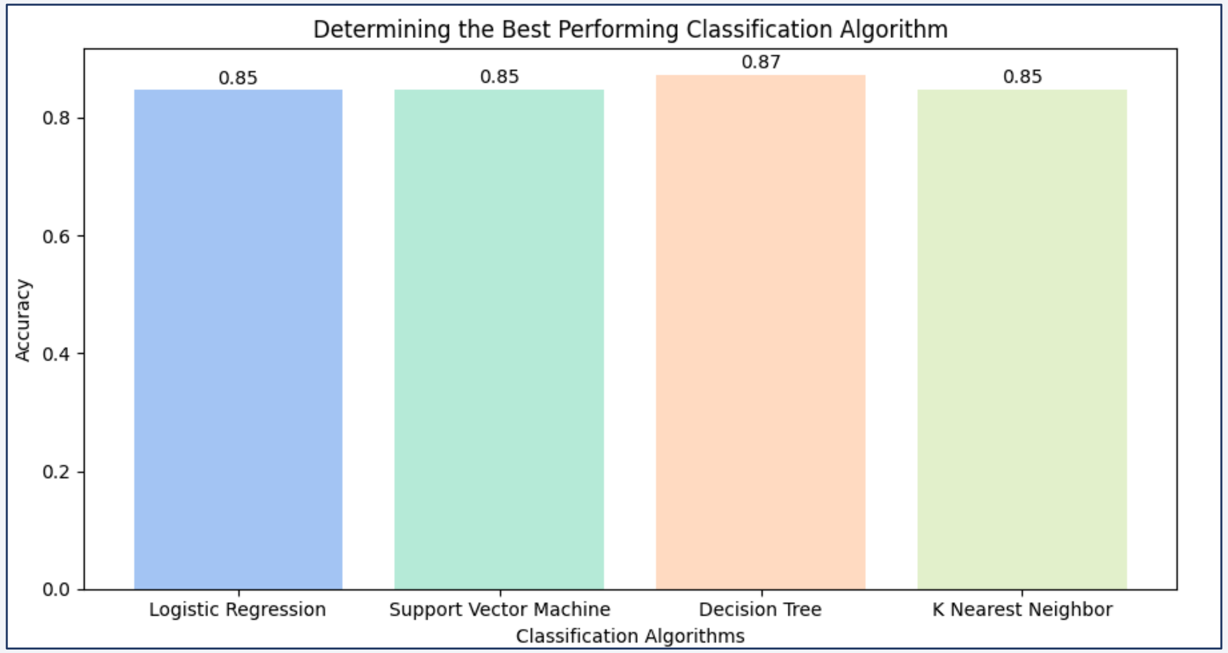
Section 5

Predictive Analysis (Classification)

Classification Accuracy

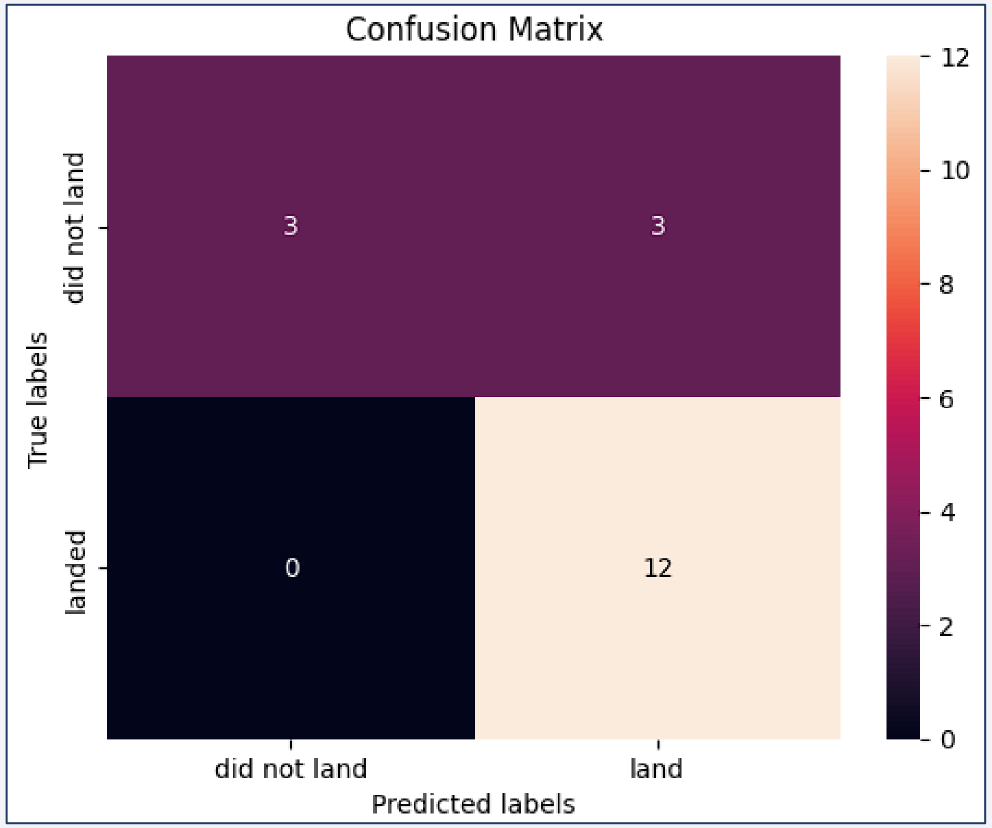
- The bar chart shows the accuracy of different classification models and we ca state that the accuracy of Decision Tree Algorithm is the highest among all other models.

- Decision Tree is the best performing model with an accuracy of 0.87



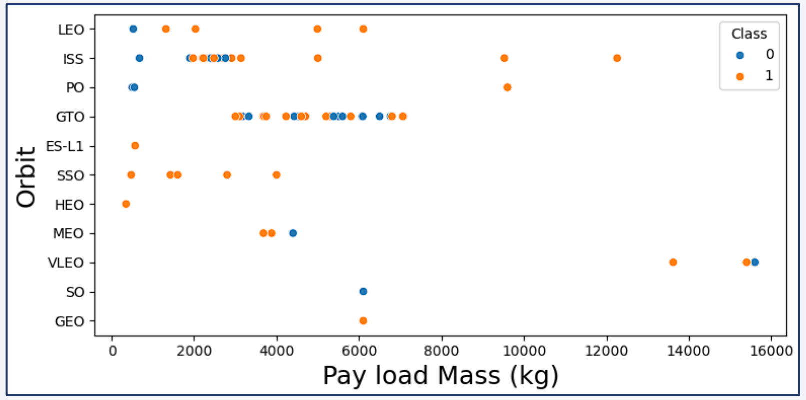
Confusion Matrix

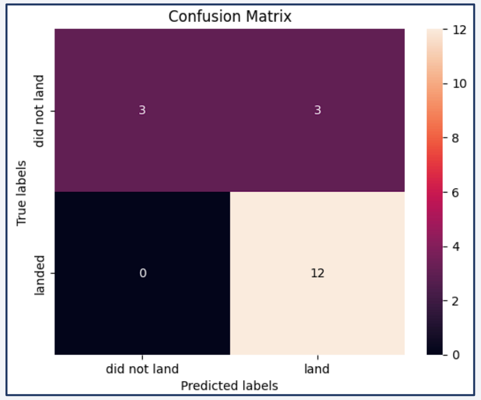
- The matrix shows 3 true negatives, 3 false positives, 12 true positives, and 0 false negatives, out of a total of 18 results.
- This means the model incorrectly predicted 3 launches as successful when they were actually not
- However, it correctly predicted all 12 actual landings and 3 actual non-landings, resulting in 15 correct classifications out of 18.
- This corresponds to an accuracy of 83.33%

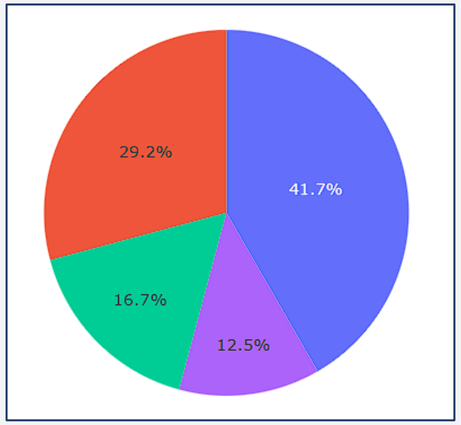


Conclusions

- This study aimed to forecast the landing success of the Falcon 9 rocket's first stage to help estimate launch costs effectively.
- Various features, such as orbit type and payload weight, were analyzed to understand their influence on mission outcomes.
- Multiple machine learning models were trained using historical launch data to uncover patterns and build predictive tools.
- Among all models tested, the Decision Tree classifier stood out with the highest accuracy (83.33%), correctly classifying 15 out of 18 cases based on the confusion matrix.
- While launch success has improved over time, especially at experienced sites, factors like payload size and orbit type continue to impact results highlighting the importance of data-driven decision-making in aerospace missions.







Appendix

- **GitHub Repository Link:**

  https://github.com/Arpita368/Applied-Data-Science-Capstone

Thank you!