IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sahath Ibunu

17th October 2025

Outline

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

Executive Summary

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

Introduction

- Project background and context

  This project develops a machine learning pipeline to predict the successful landing of a SpaceX Falcon 9 rocket's first stage. As reusability is the key to SpaceX's low launch cost of $62 million—compared to over $165 million for other providers—accurately predicting this outcome allows for precise cost assessment. The model can serve as a critical tool for competitors in formulating bids against SpaceX.

  Problems you want to find answers

  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - One-hot encoding was applied to categorical features
- 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 SpaceX dataset was compiled through a multi-source data collection approach to ensure comprehensive coverage of Falcon 9 launch records.

API Data Collection:

- Primary data was sourced directly from the official SpaceX API using HTTP GET requests
- JSON responses were parsed and converted into structured pandas DataFrames using the json_normalize() method
- This provided real-time, authoritative launch data including mission details and outcomes

Data Processing Pipeline:

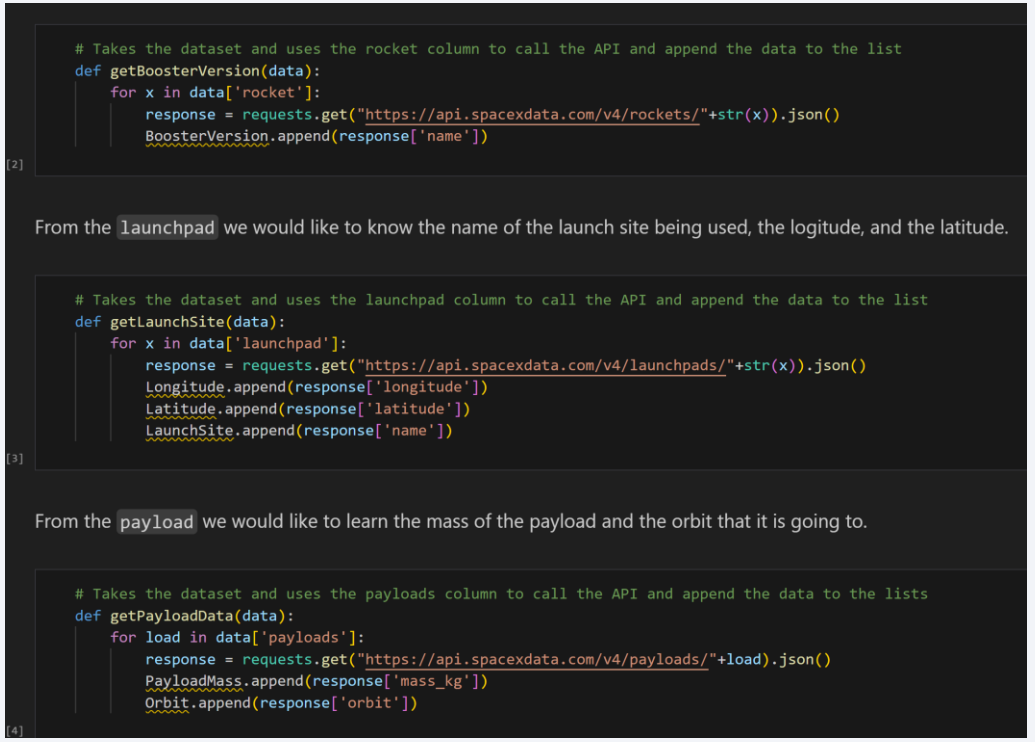
- Implemented comprehensive data cleaning procedures to handle missing values
- Performed data validation and imputation where necessary to maintain dataset integrity
- Standardized data formats across all records for consistent analysis

Supplementary Web Scraping:

- Enhanced the dataset with historical context through web scraping from Wikipedia
- Utilized BeautifulSoup to extract Falcon 9 launch records from HTML tables
- Transformed scraped tabular data into pandas DataFrames for seamless integration with API-sourced data
- This hybrid methodology combining official API data with curated historical records resulted in a robust, validated dataset suitable for comprehensive launch analysis and predictive modeling.

Data Collection – SpaceX API

- The dataset was constructed by executing GET requests to the SpaceX API to gather raw launch data. The collected information underwent comprehensive cleaning to ensure data quality and consistency. Basic data wrangling operations were performed to structure and format the dataset appropriately for analysis.The link to the notebook is
- https://github.com/Sahath787/Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb



```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

[2]

From the `launchpad` we would like to know the name of the launch site being used, the logitude, and the latitude.

```
# Takes the dataset and uses the launchpad column to call the API and append the data to the list
def getLaunchSite(data):
    for x in data['launchpad']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/"+str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```

[3]

From the `payload` we would like to learn the mass of the payload and the orbit that it is going to.

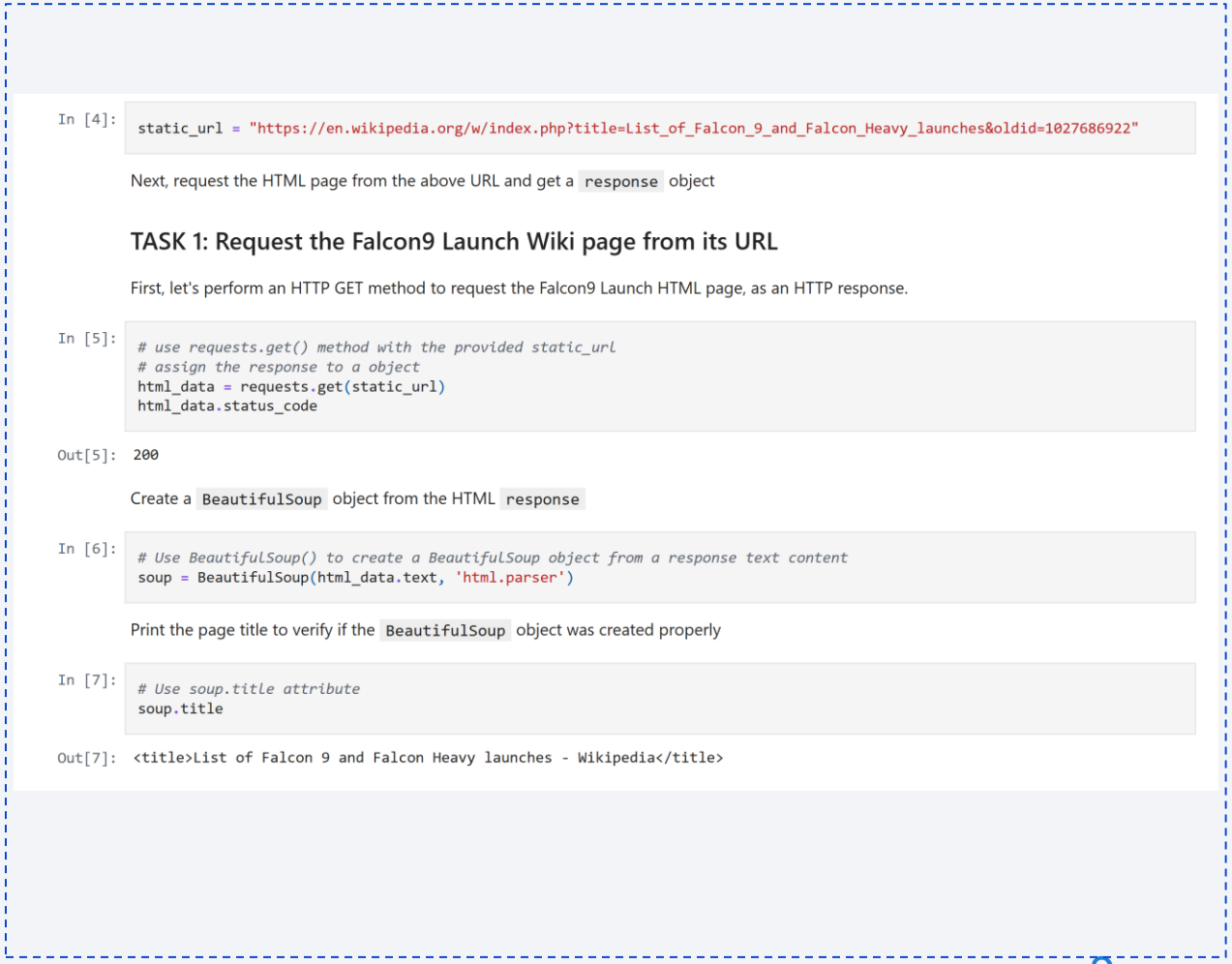
```
# Takes the dataset and uses the payloads column to call the API and append the data to the lists
def getPayloadData(data):
    for load in data['payloads']:
        response = requests.get("https://api.spacexdata.com/v4/payloads/"+load).json()
        PayloadMass.append(response['mass_kg'])
        Orbit.append(response['orbit'])
```

[4]

Data Collection - Scraping

- We applied web scraping techniques to extract Falcon 9 launch records using Beautiful Soup. We parsed the HTML table containing launch data and converted it into a structured pandas Data Frame for analysis.
- The link to the notebook is

https://github.com/Sahath787/Data-Science-Capstone/blob/main/Data%20Collection%20API.ipynb



In [4]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
html_data = requests.get(static_url)
html_data.status_code
```

Out[5]: 200

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

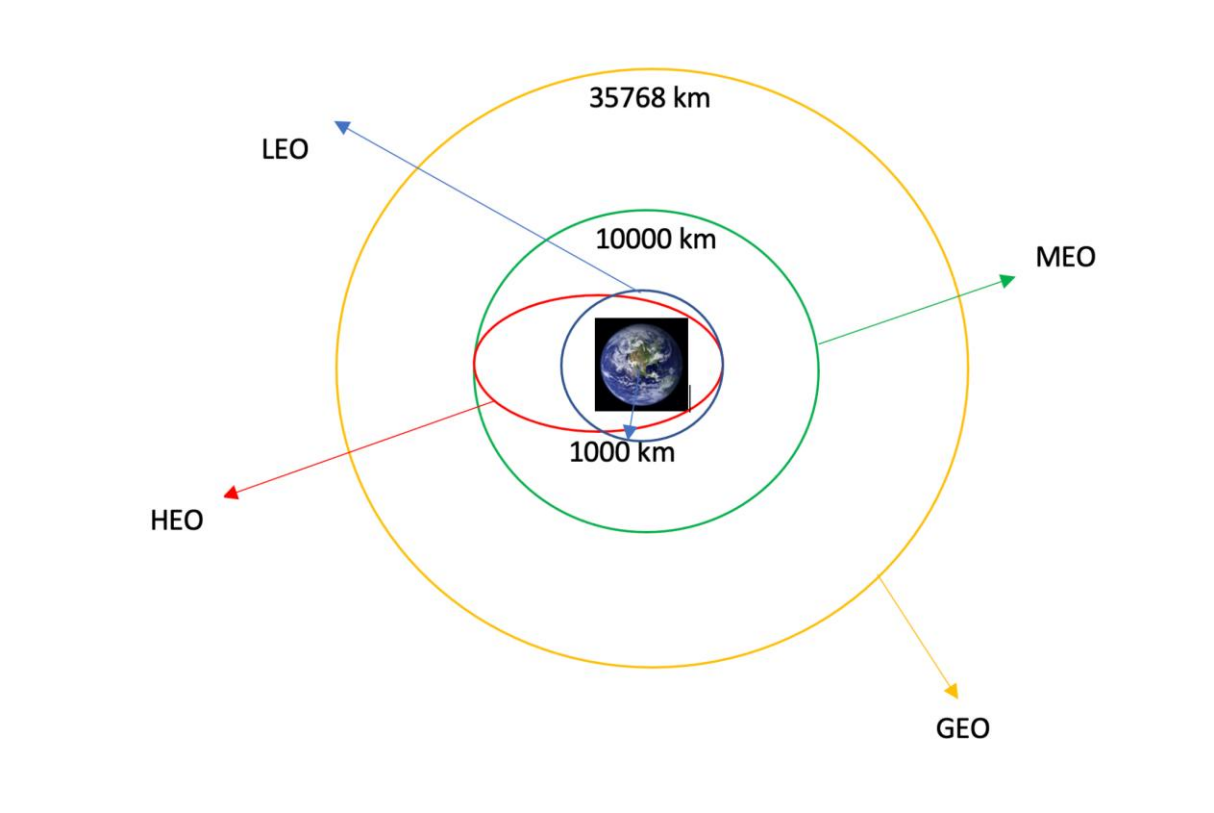
In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(html_data.text, 'html.parser')
```

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

In [7]:
```
# Use soup.title attribute
soup.title
```

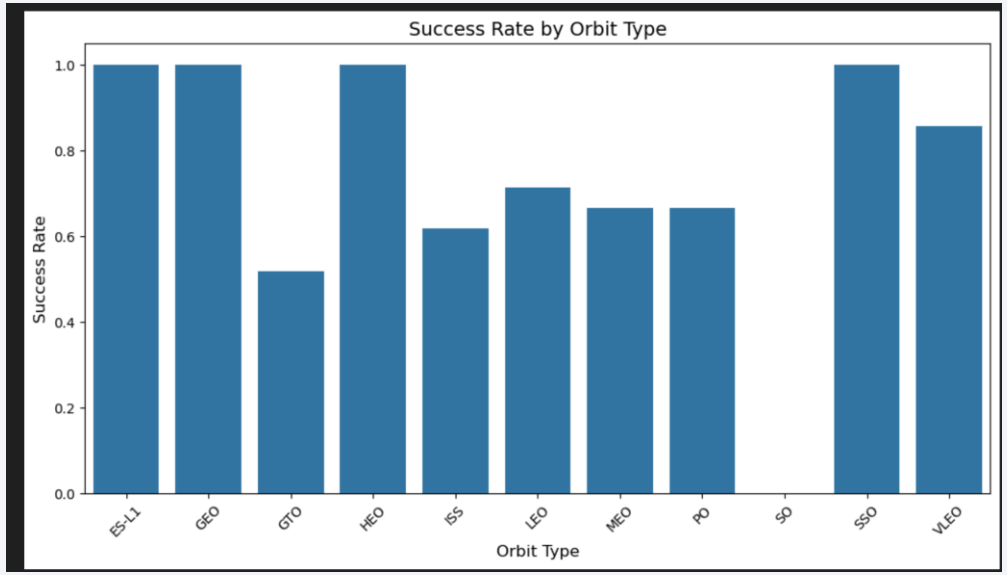
Out[7]: `<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>`

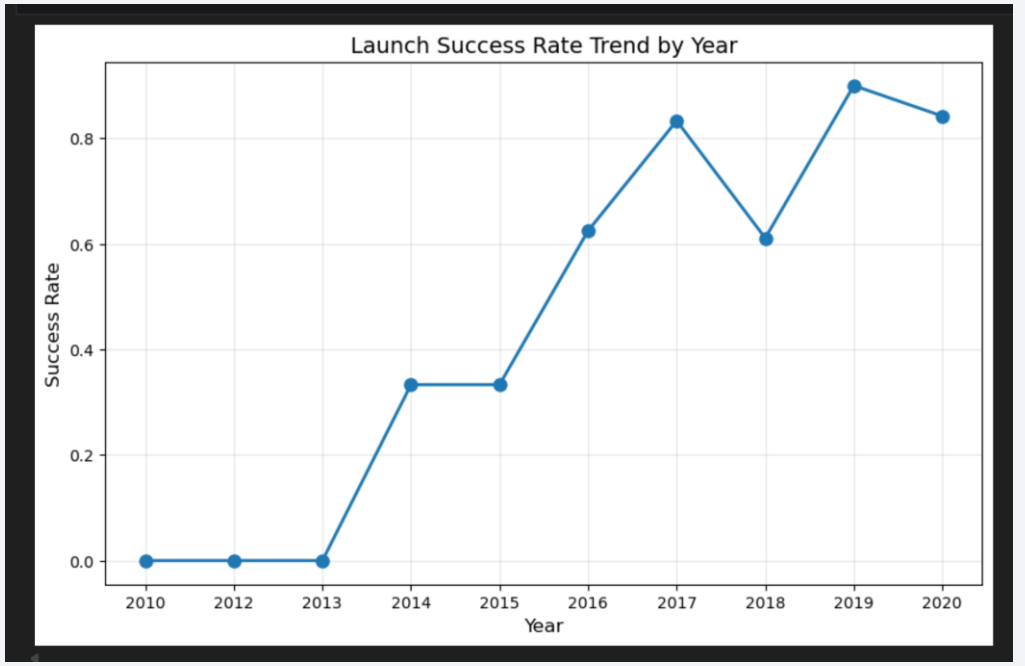
Data Wrangling



- We conducted exploratory data analysis to identify patterns and establish training labels for our predictive model. This included calculating launch frequency across different sites and analyzing the distribution and occurrence rates of various orbital destinations.
- We then engineered a binary classification label from the mission outcome column, encoding successful landings as '1' and unsuccessful attempts as '0', before exporting the processed dataset to CSV format for subsequent modeling.
- The link to the notebook is https://github.com/Sahath787/Data-Science-Capstone/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization





- **Unique launch site identification** using DISTINCT to catalog all SpaceX launch locations and their operational patterns
- **Payload capacity analysis** through aggregate functions (SUM, AVG, MAX) revealing booster performance limits and NASA mission specifications
- **Landing success tracking** combining date filtering, pattern matching, and grouping to identify improvement trends and optimal operational condition

- The link to the notebook is:
- https://github.com/Sahath787/Data-Science-Capstone/blob/main/EDA%20with%20Data%20Visualization.ipynb

EDA with SQL

- **SQL Analysis Summary:**

- **Task 1-2: Identified unique launch sites and filtered CCA-prefixed sites using DISTINCT and LIKE operators**
- **Task 3-5: Calculated NASA payload totals (45,596 kg), booster averages, and earliest successful ground landing dates**
- **Task 6-10: Analyzed successful drone ship landings, mission outcomes, max payload carriers, and 2015 failure patterns with date filtering and aggregation**
- The link to the notebook is https://github.com/Sahath787/Data-Science-Capstone/blob/main/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

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

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is https://github.com/Sahath787/Data-Science-Capstone/blob/main/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Predictive Analysis (Classification)

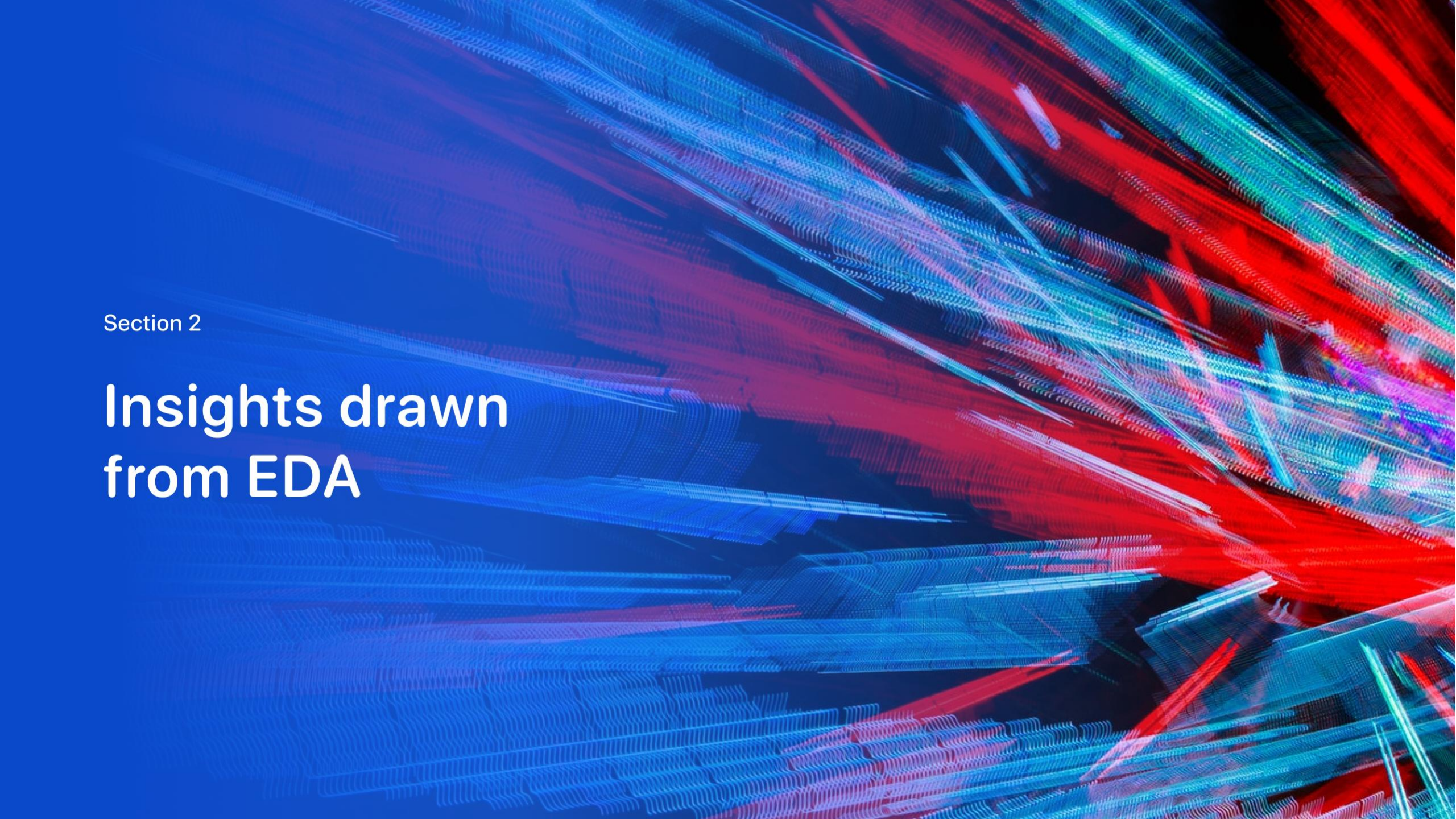
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.
- The link to the notebook is https://github.com/Sahath787/Data-Science-Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

Results

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

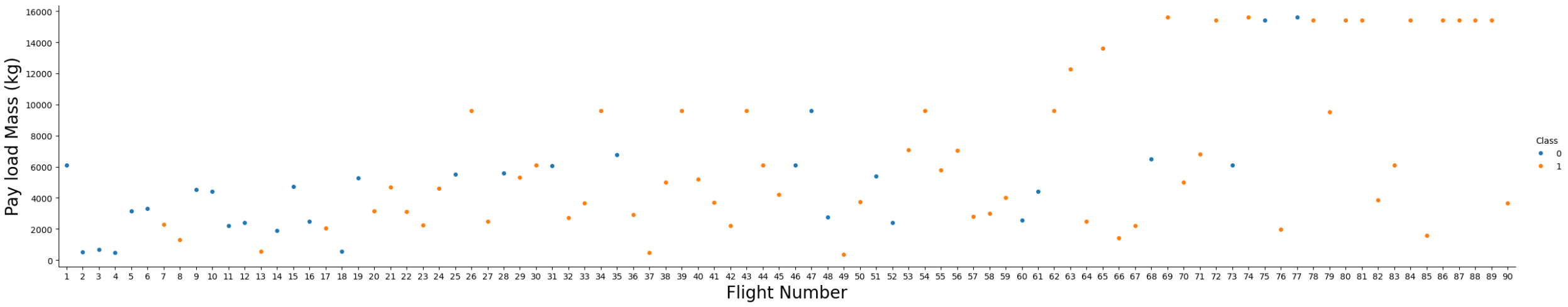
Section 2

Insights drawn from EDA



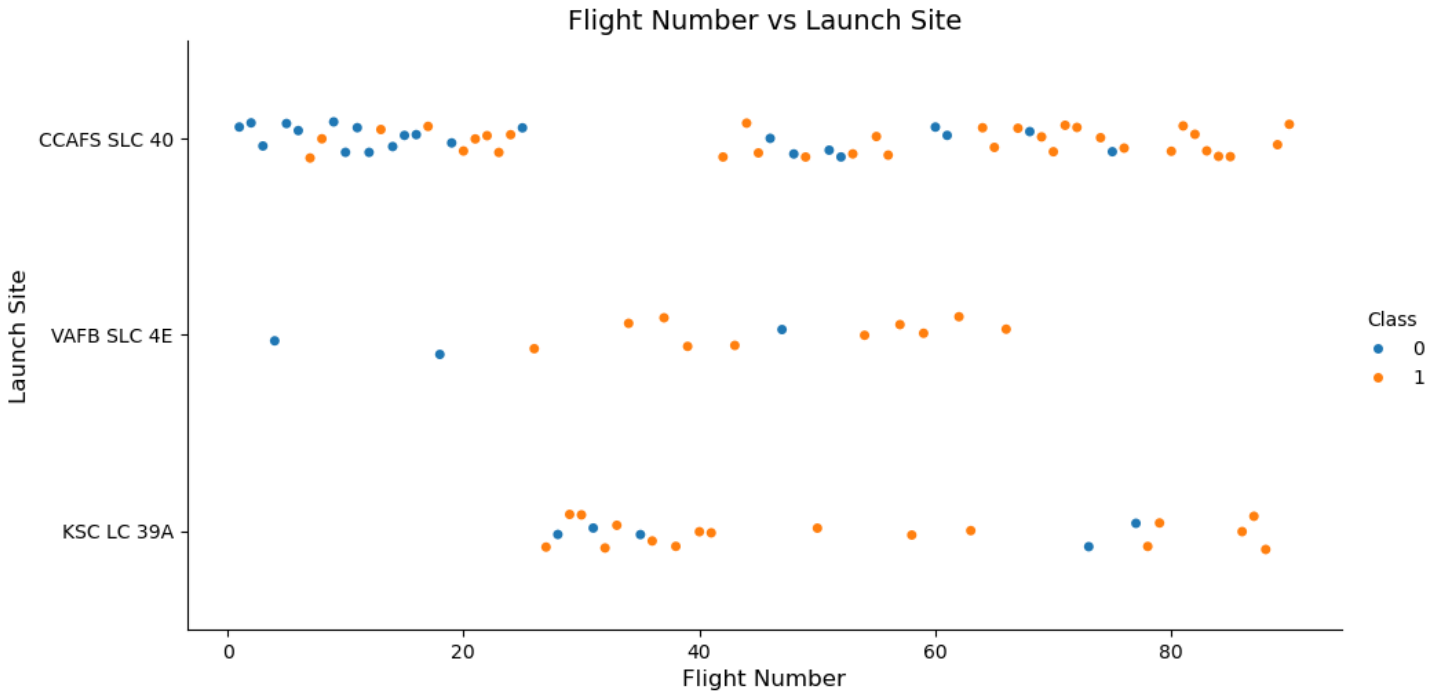
Flight Number vs. Launch Site

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



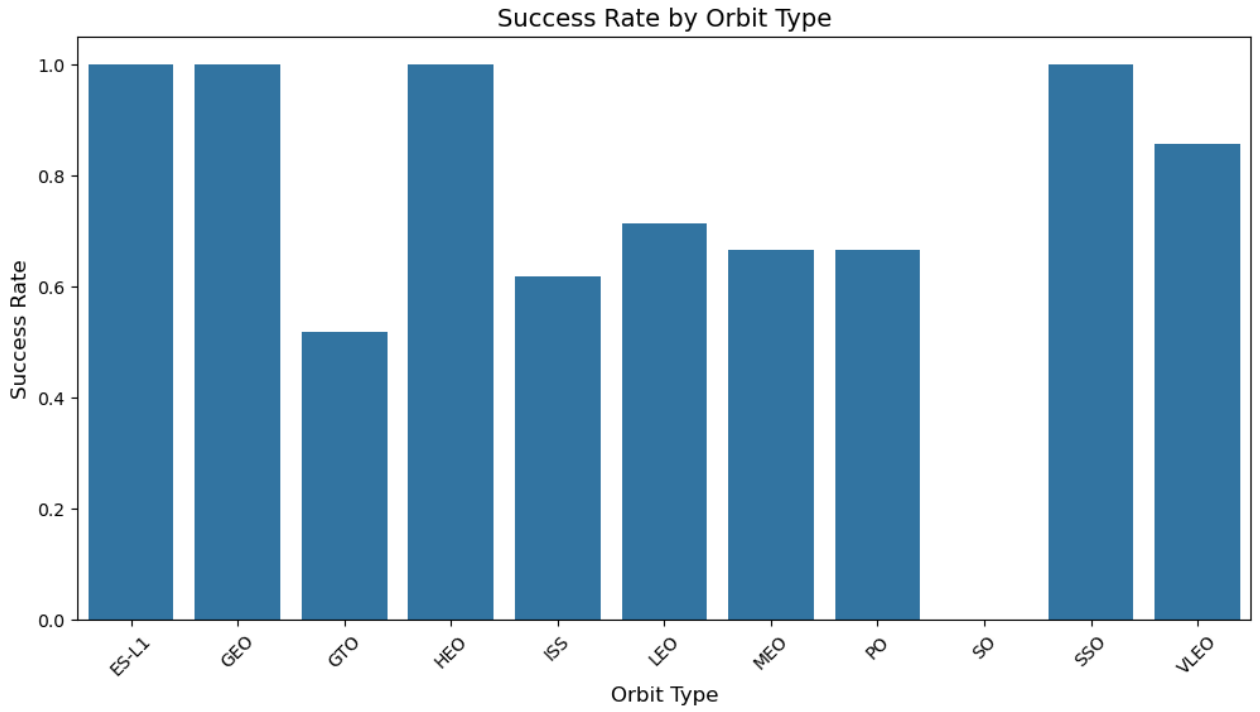
Payload vs. Launch Site

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



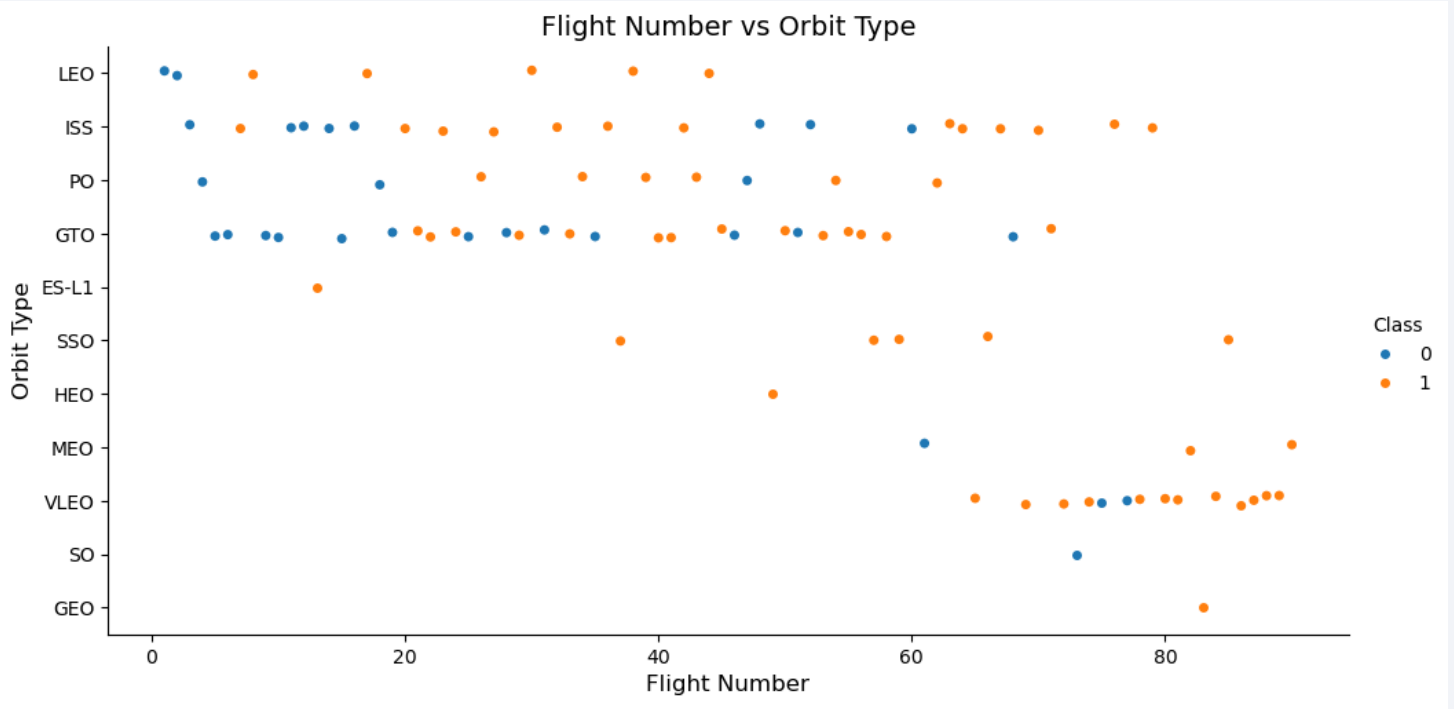
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.



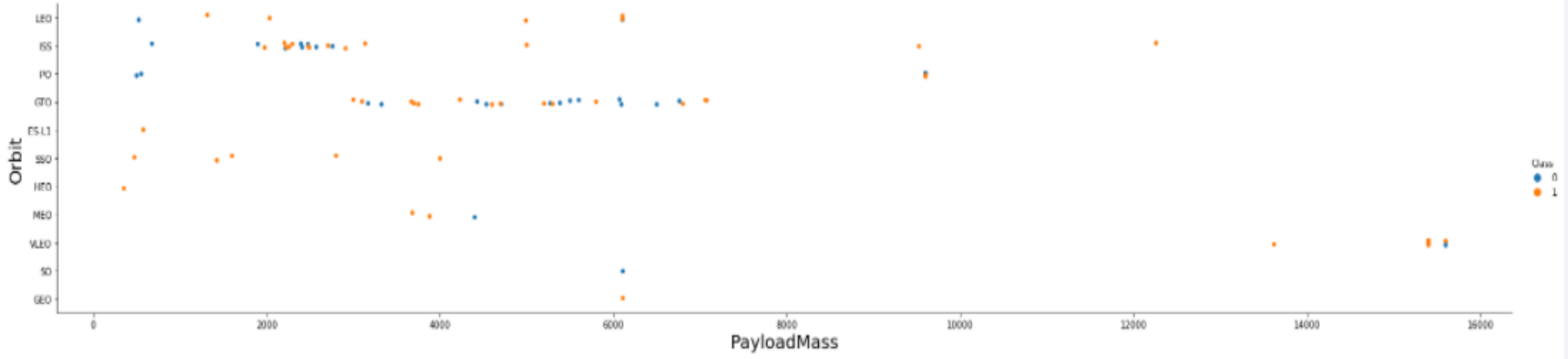
Flight Number vs. Orbit Type

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



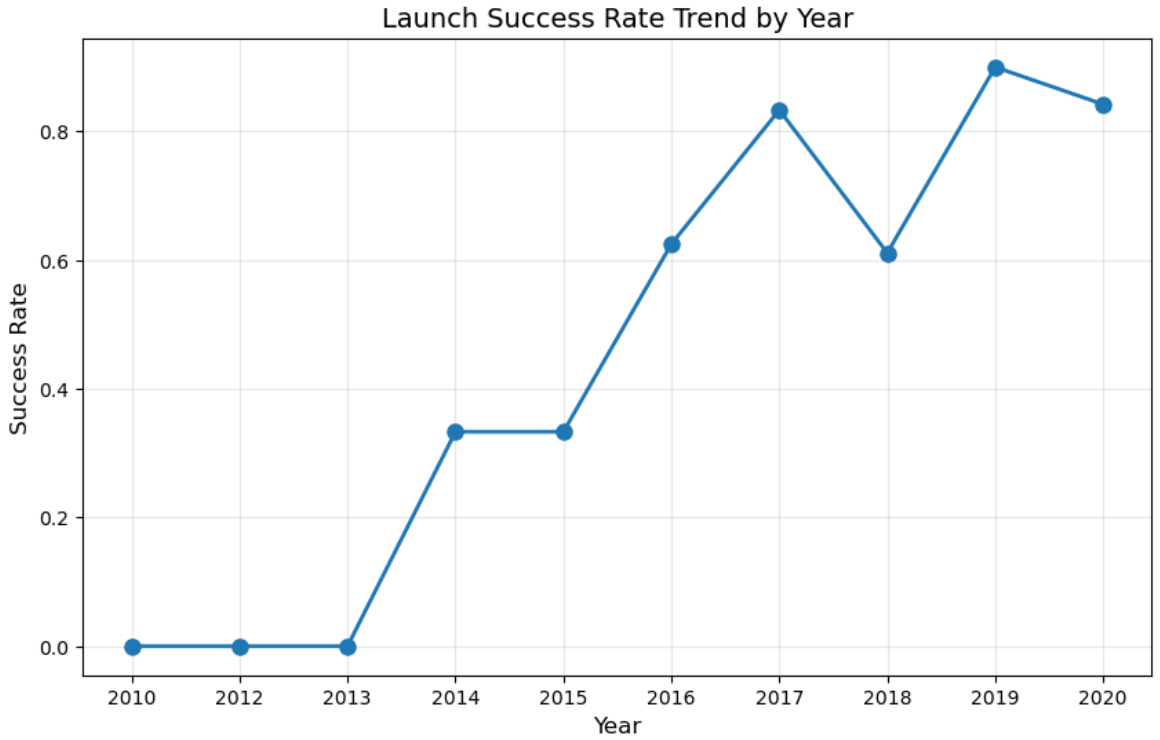
Payload vs. Orbit Type

- We can observe that with heavy payloads, the successful landing are more for PO, LEO and ISS orbits.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

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

In [10]:
```
task_1 = '''
        SELECT DISTINCT LaunchSite
        FROM SpaceX
'''
create_pandas_df(task_1, database=conn)
```

Out[10]:

	launchsite
0	KSC LC-39A
1	CCAFS LC-40
2	CCAFS SLC-40
3	VAFB SLC-4E

Launch Site Names Begin with 'CCA'

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

Generate + Code + Markdown

```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

[12] Python

```
 * 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

- The query filtered and displayed the first 5 launch records from sites starting with 'CCA' to analyze mission characteristics.

Total Payload Mass

- The query calculated that NASA missions carried a total payload mass of 45,596 kg across all SpaceX booster launches.

Task 3

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

```
%%sql
SELECT SUM("Payload_Mass__kg_") as Total_Payload_Mass
FROM SPACEXTABLE
WHERE "Customer" LIKE '%NASA%' OR "Customer" LIKE '%CRS%';
```

Python

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

Total_Payload_Mass
107010

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2928.4

Task 4

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

```
%%sql
SELECT AVG("Payload_Mass__kg_") as Average_Payload_Mass
FROM SPACEXTABLE
WHERE "Booster_Version" = 'F9 v1.1';
```

[14]

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

Average_Payload_Mass
2928.4

First Successful Ground Landing Date

- We observed that the dates of the first successful landing outcome on ground pad was 22nd December 2015

Task 5

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

Hint:Use min function

```
%%sql
SELECT MIN(Date) as First_Successful_Ground_Landing
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE '%Success%'
AND "Landing_Outcome" LIKE '%ground pad%';
```

[15]

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

First_Successful_Ground_Landing
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%%sql
SELECT DISTINCT "Booster_Version"
FROM SPACEXTABLE
WHERE "Landing_Outcome" LIKE '%Success%'
AND "Landing_Outcome" LIKE '%drone ship%'
AND "Payload_Mass__kg_" > 4000
AND "Payload_Mass__kg_" < 6000;
```

[16] Python

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

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

- The query filtered for boosters that successfully landed on drone ships while carrying payloads between 4,000-6,000 kg using combined WHERE and AND conditions.

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

[17]

```
%%sql
SELECT
    CASE
        WHEN "Mission_Outcome" LIKE '%Success%' THEN 'Success'
        ELSE 'Failure'
    END as Outcome_Type,
    COUNT(*) as Count
FROM SPACEXTABLE
GROUP BY Outcome_Type;
```

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

Outcome_Type	Count
Failure	1
Success	100

- We used wildcard like '%' to filter for WHERE Mission Outcome was a success or a failure.

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

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

[18]

```
 * 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

```
%%sql
SELECT
    substr(Date, 6, 2) as month,
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE substr(Date, 0, 5) = '2015'
AND "Landing_Outcome" LIKE '%Failure%'
AND "Landing_Outcome" LIKE '%drone ship%';
```

[19]

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

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

- The query combined WHERE, LIKE, AND, and BETWEEN conditions to filter failed drone ship landings with their booster versions and launch sites specifically for the year 2015.

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

```
%%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;
```

[20]

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

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

- The query selected landing outcomes with their counts and filtered for records between June 4, 2010 and March 20, 2017 using date range conditions.

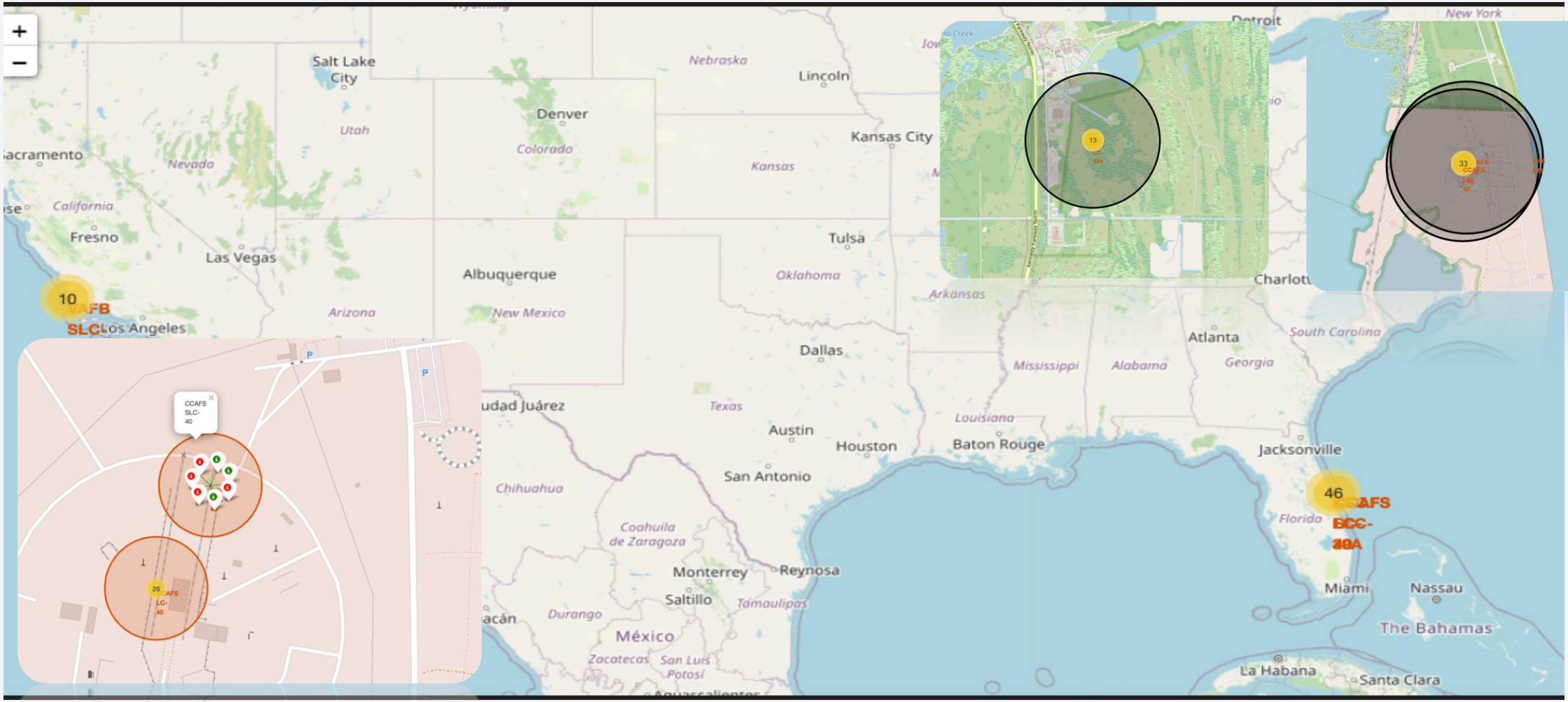
- The query grouped landing outcomes by type and ordered them in descending frequency to identify the most common landing results.

Section 4

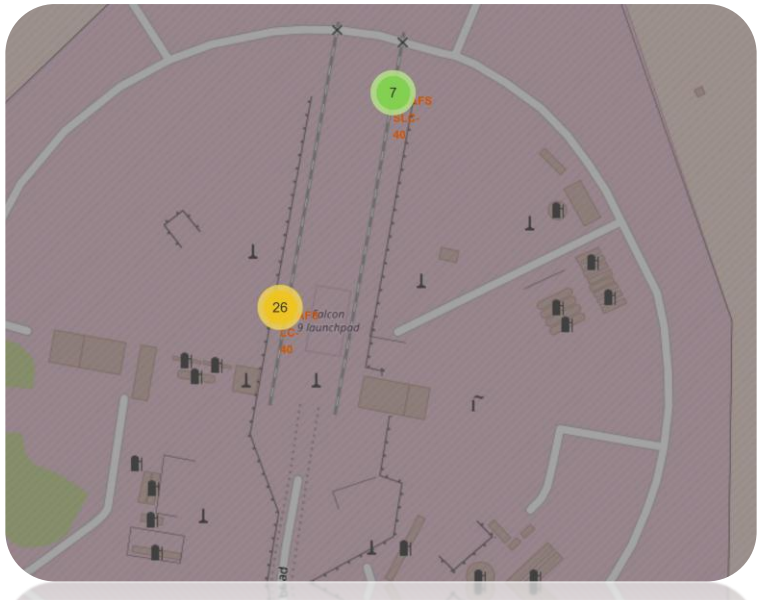
Launch Sites Proximities Analysis

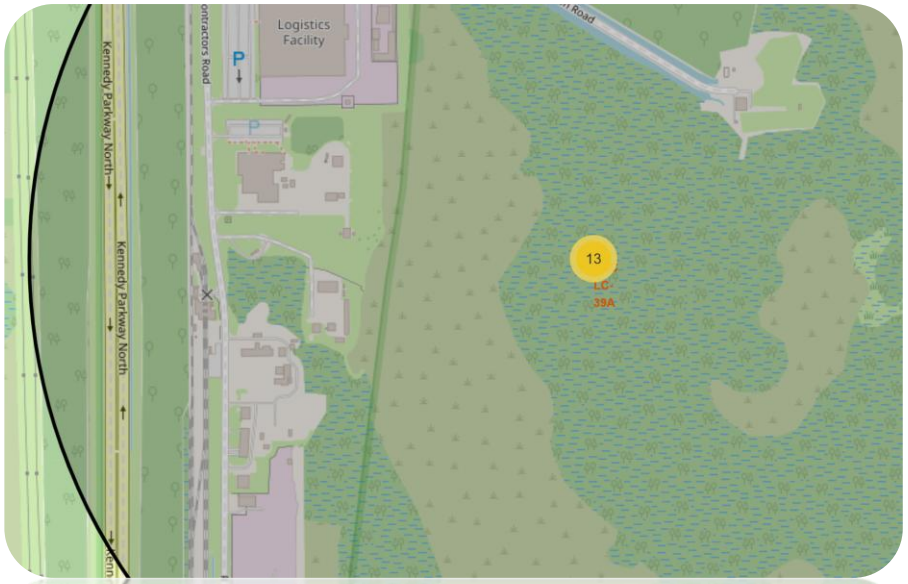
All launch sites global map markers

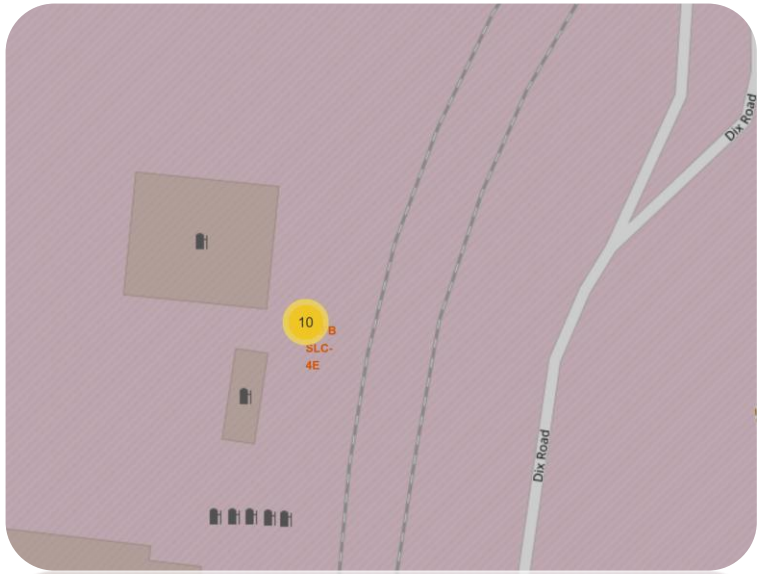
Markers showing launch sites with color labels



Launch Site distance to landmarks







Coastline: ✅ Yes, all sites directly coastal for safety and ocean recovery operations.

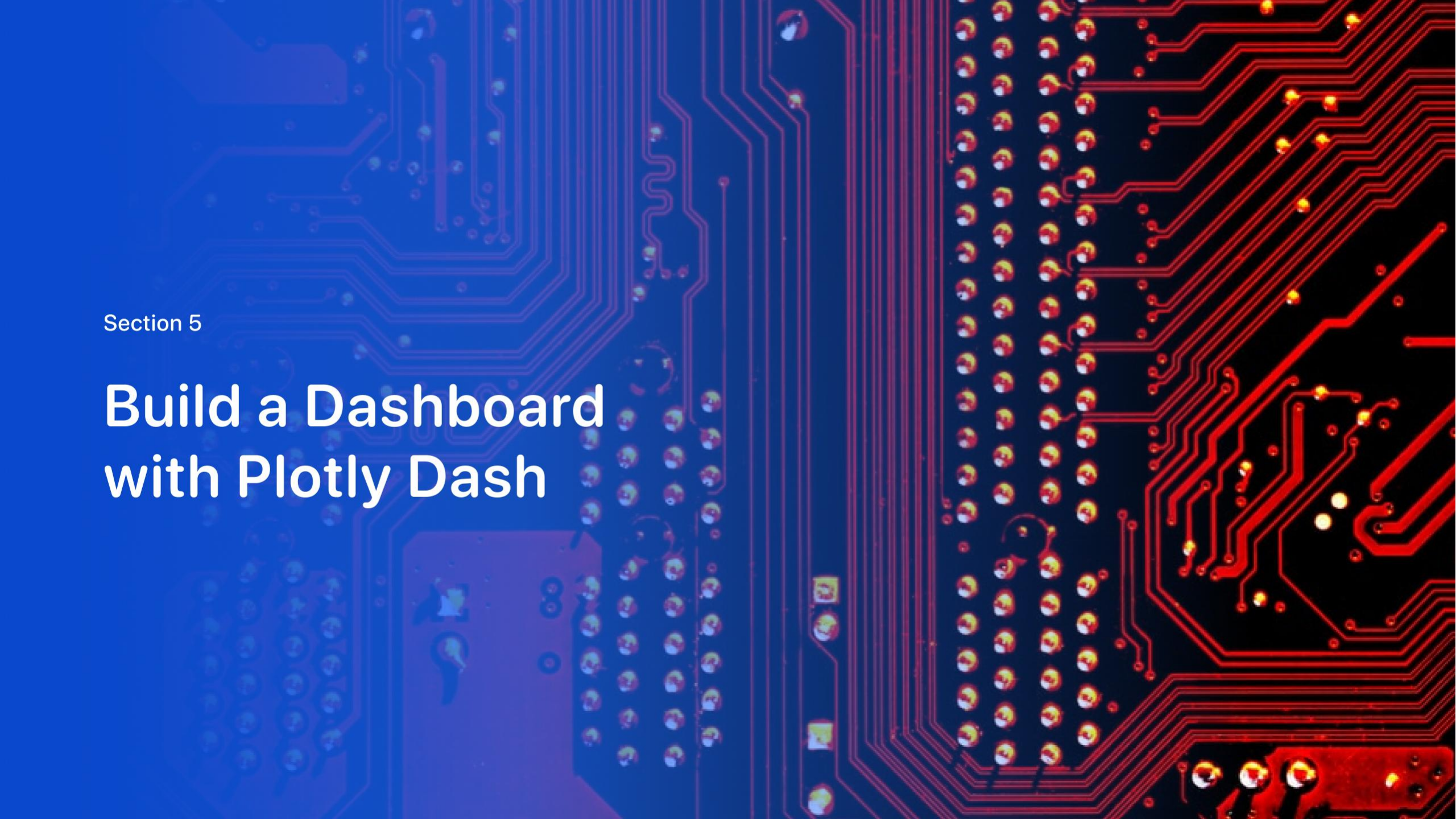
Highways: ✅ Yes, close highway access for efficient logistics and rapid transport.

Railways: ❌ No, not typically close as road transport better suits rocket components.

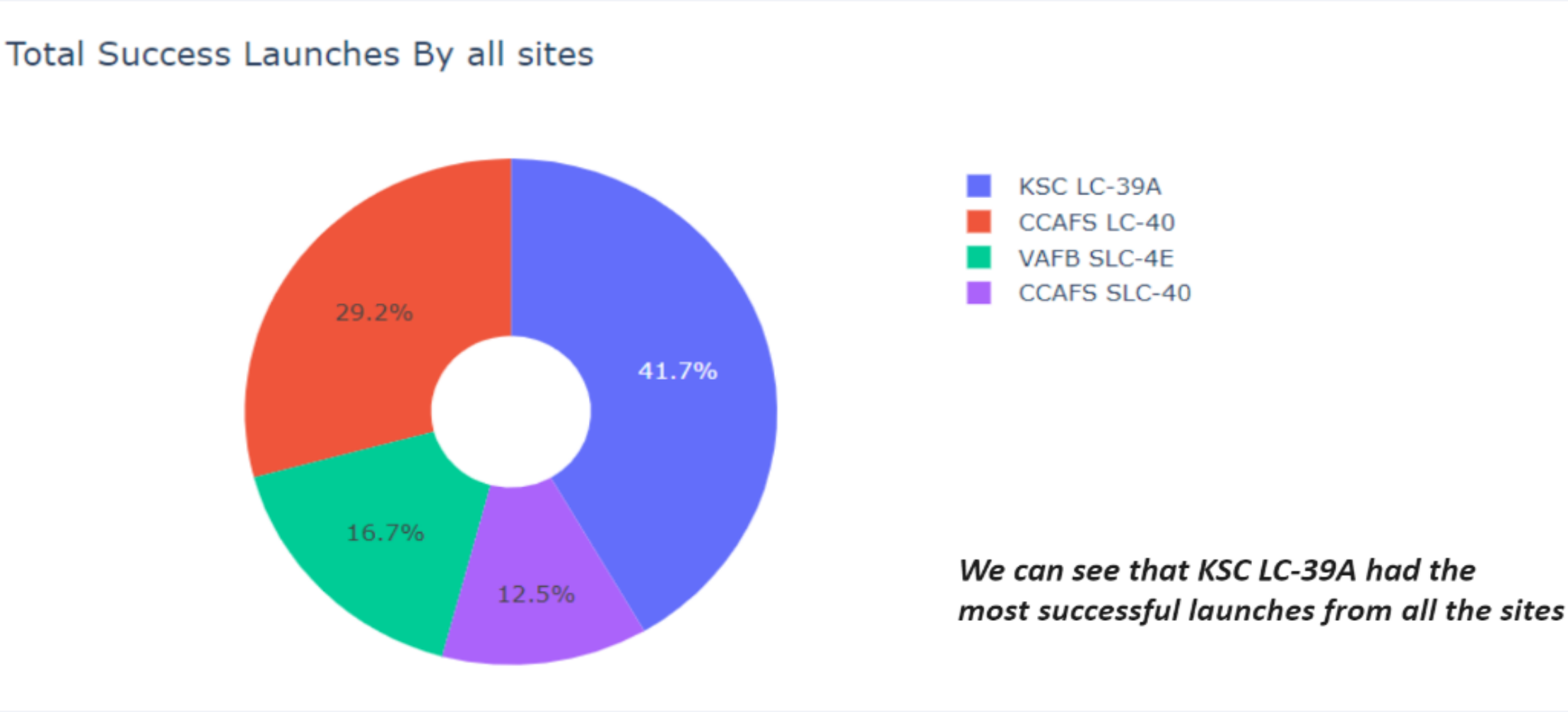
Cities: ✅ Yes, maintained safe distances from urban centers for risk mitigation.

Section 5

Build a Dashboard with Plotly Dash

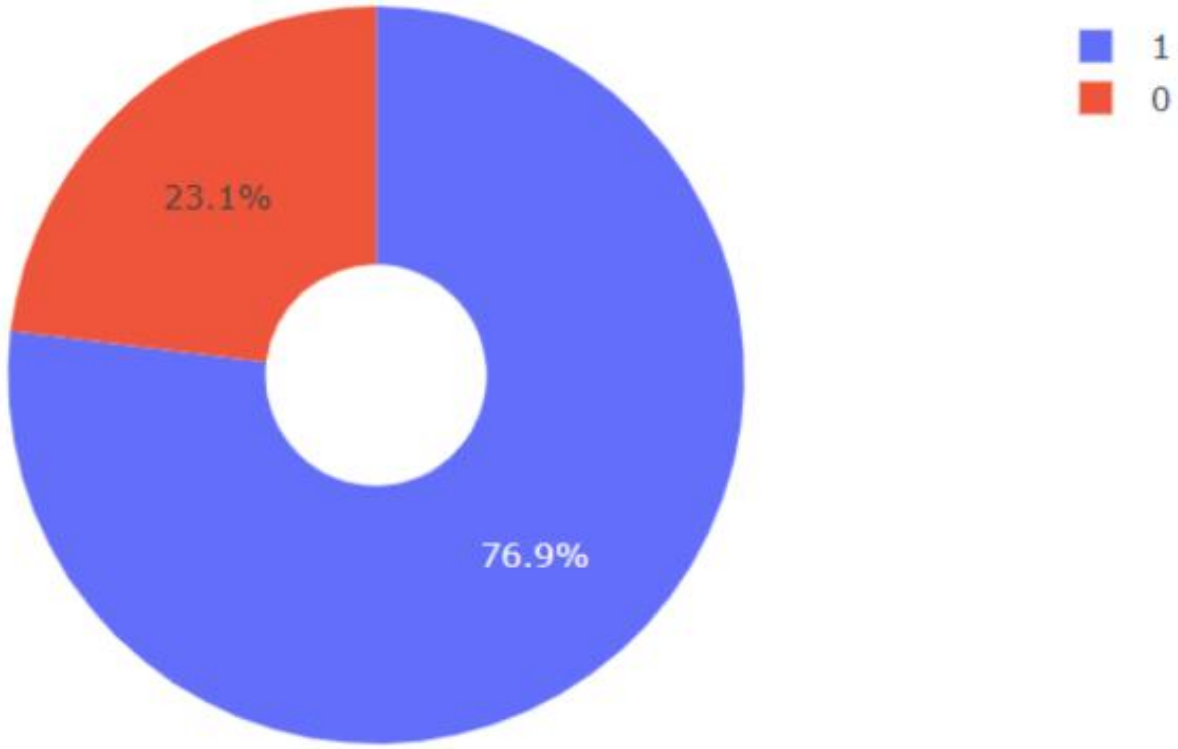


Pie chart showing the success percentage achieved by each launch site



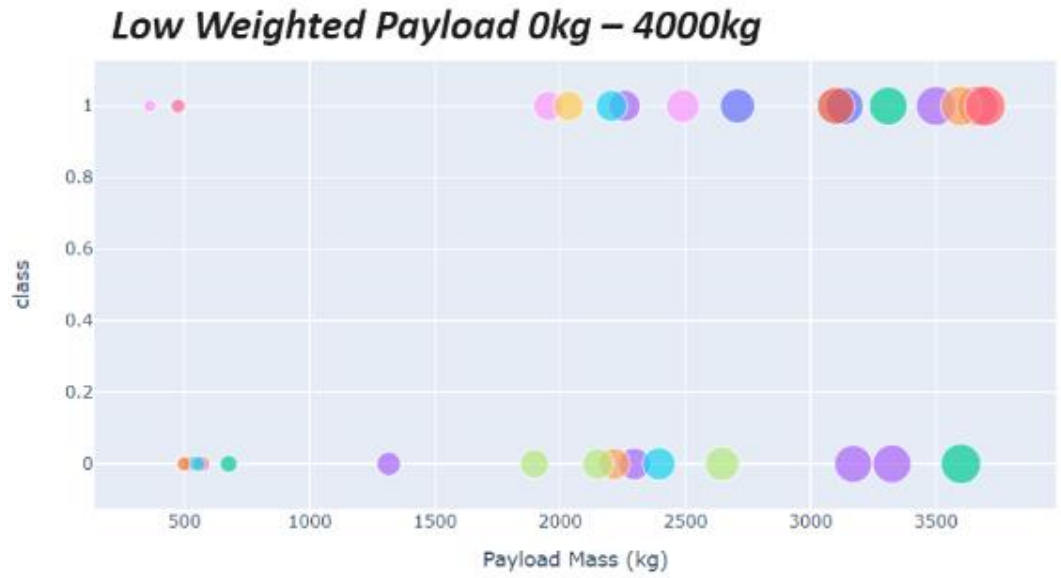
We can see that KSC LC-39A had the most successful launches from all the sites

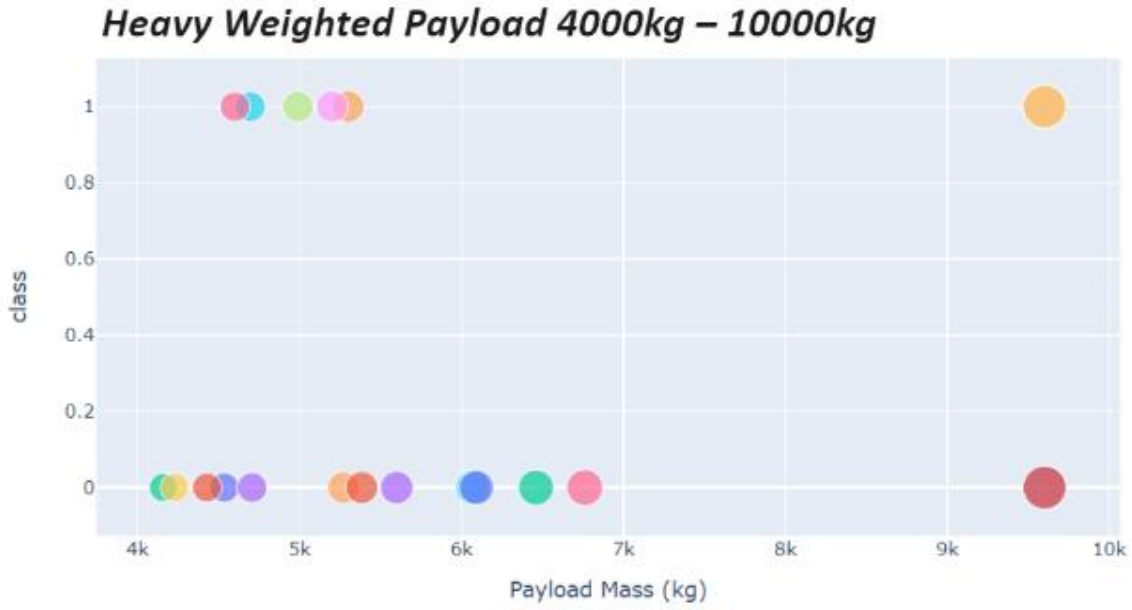
Pie chart showing the Launch site with the highest launch success ratio



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

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





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

Section 6

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy

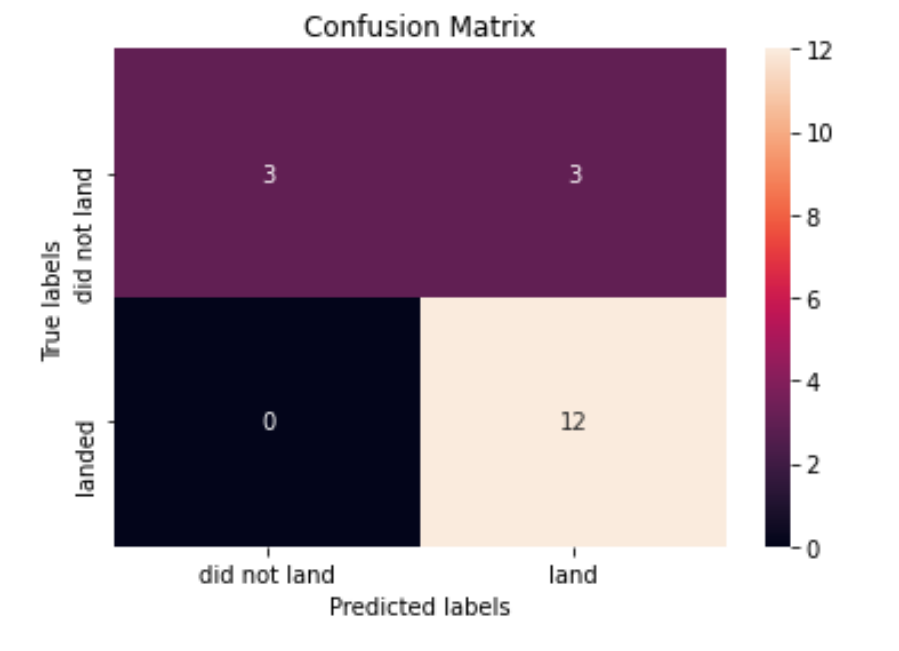
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'splitter': 'random'}
```

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!