IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raghavendar V
20/12/23

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
  - SpaceX is Space Exploration Technologies Corp, mainly a launch service provider, defence contractor and satellite communications company. It launches many rockets carrying various payloads and we know that rocket science is a expensive business!
  - Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - Reasons behind first stage success or failure
  - 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

Data Collection

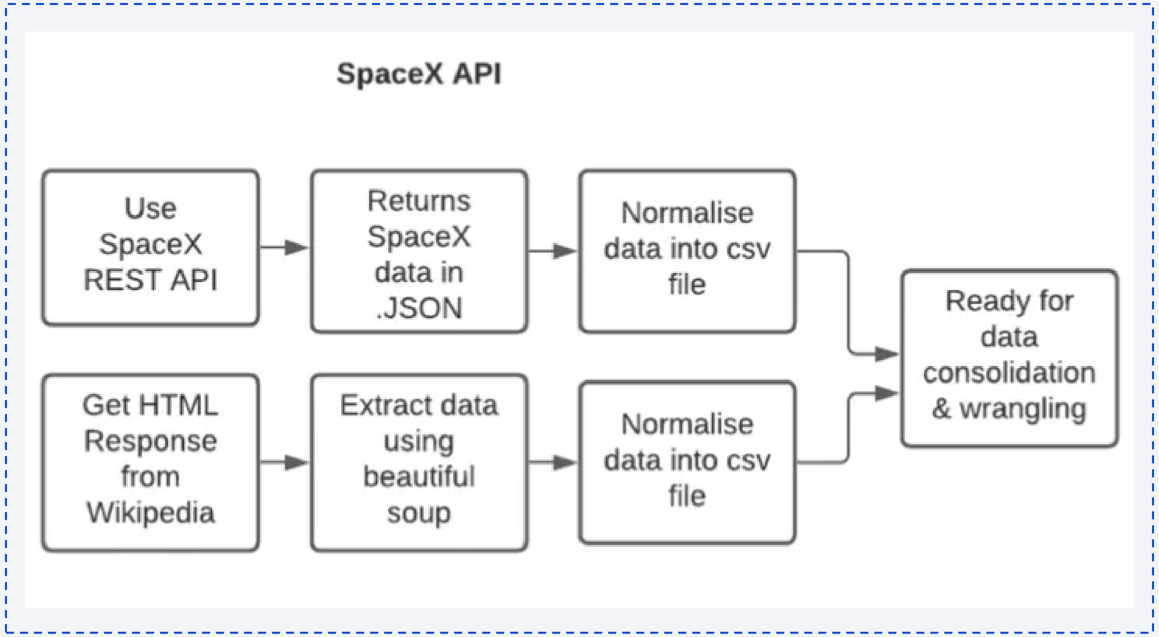
We employed two primary methods for data collection:

- Firstly, utilizing SpaceX's API, we gathered essential data, including booster details, launch sites, and payload information.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - Handling missing data, especially in payload and launch site features, we opted for the replace-with-mean method.

- Secondly, we performed web scraping on Wikipedia pages related to Falcon 9 and Falcon Heavy launches using BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

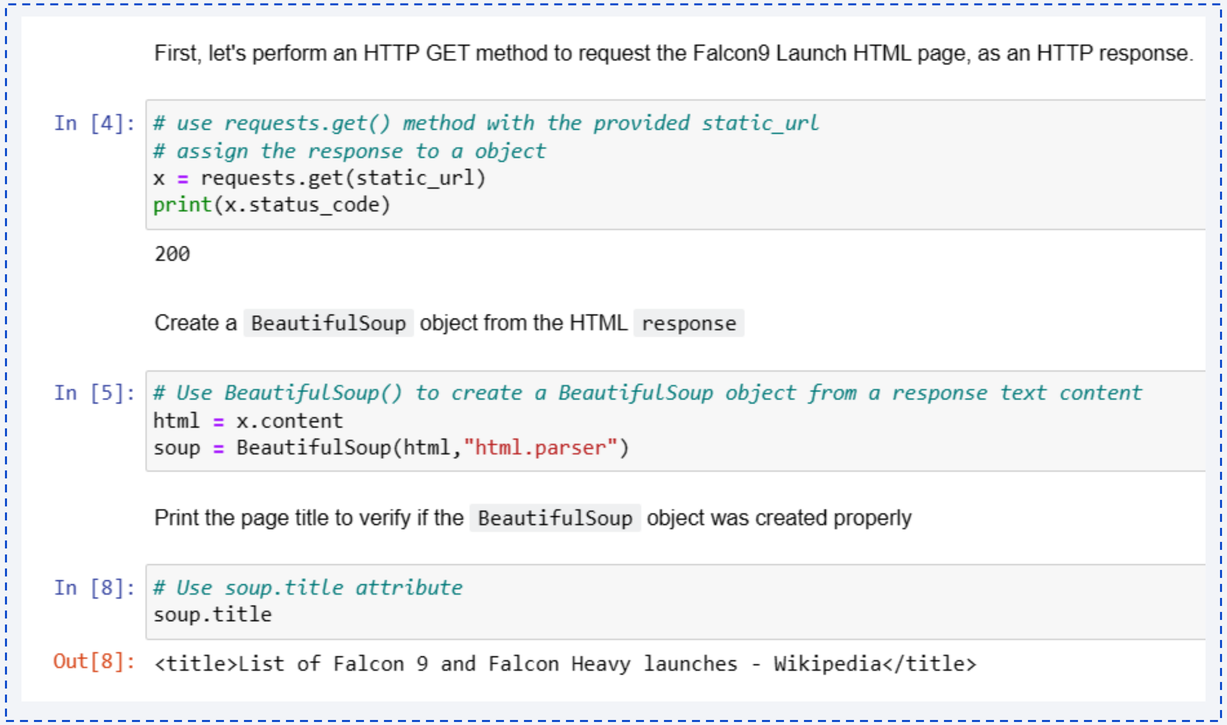
Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/01_jupyter-labs-spacex-data-collection-api.ipynb


SpaceX API
Use SpaceX REST API
Returns SpaceX data in .JSON
Normalise data into csv file
Get HTML Response from Wikipedia
Extract data using beautiful soup
Normalise data into csv file
Ready for data consolidation & wrangling

Data Collection - Scraping

- We utilized web scraping techniques with BeautifulSoup to extract Falcon 9 launch records.
- After parsing the table, we transformed the data into a Pandas dataframe for further analysis.
- GitHub URL:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/02_jupyter-labs-webscraping.ipynb



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

```
In [4]: # use requests.get() method with the provided static_url
        # assign the response to a object
        x = requests.get(static_url)
        print(x.status_code)

        200
```

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

```
In [5]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        html = x.content
        soup = BeautifulSoup(html,"html.parser")
```

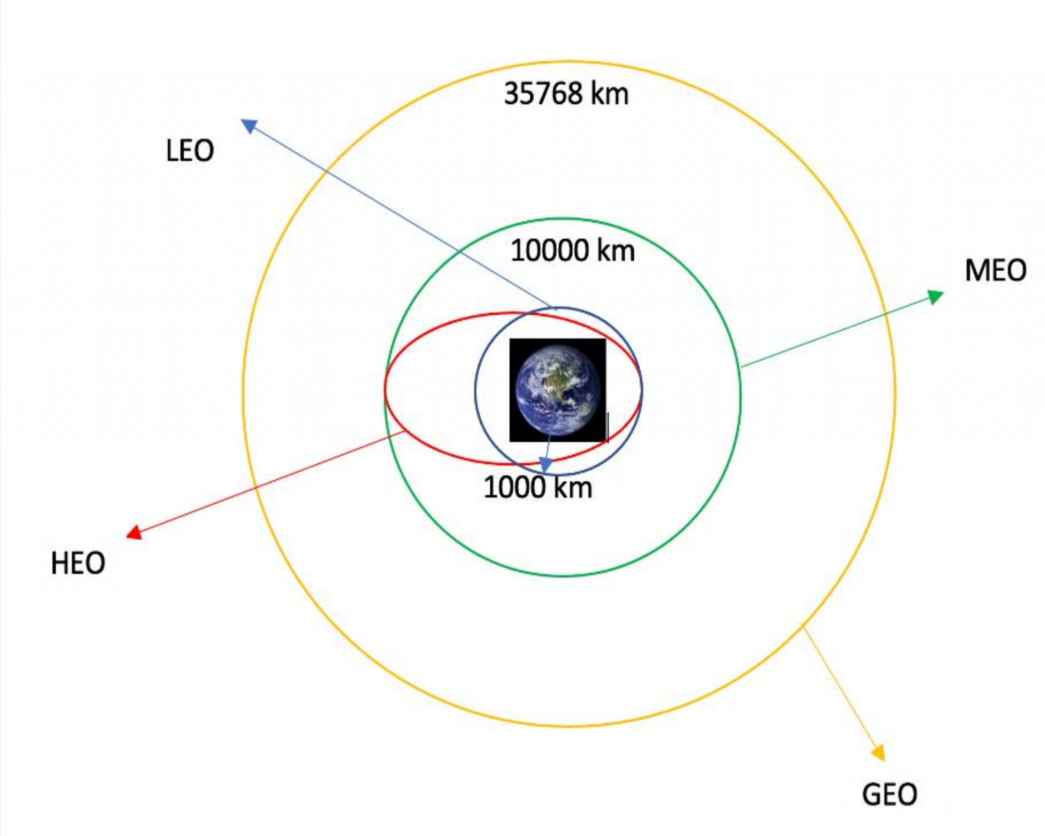
Print the page title to verify if the `BeautifulSoup` object was created properly

```
In [8]: # Use soup.title attribute
        soup.title

Out[8]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

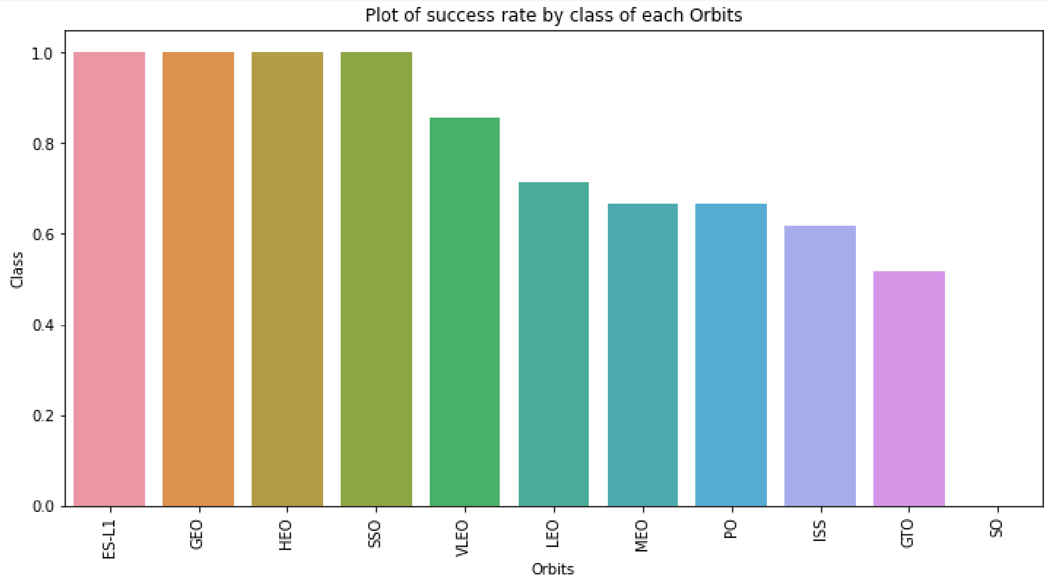
Data Wrangling

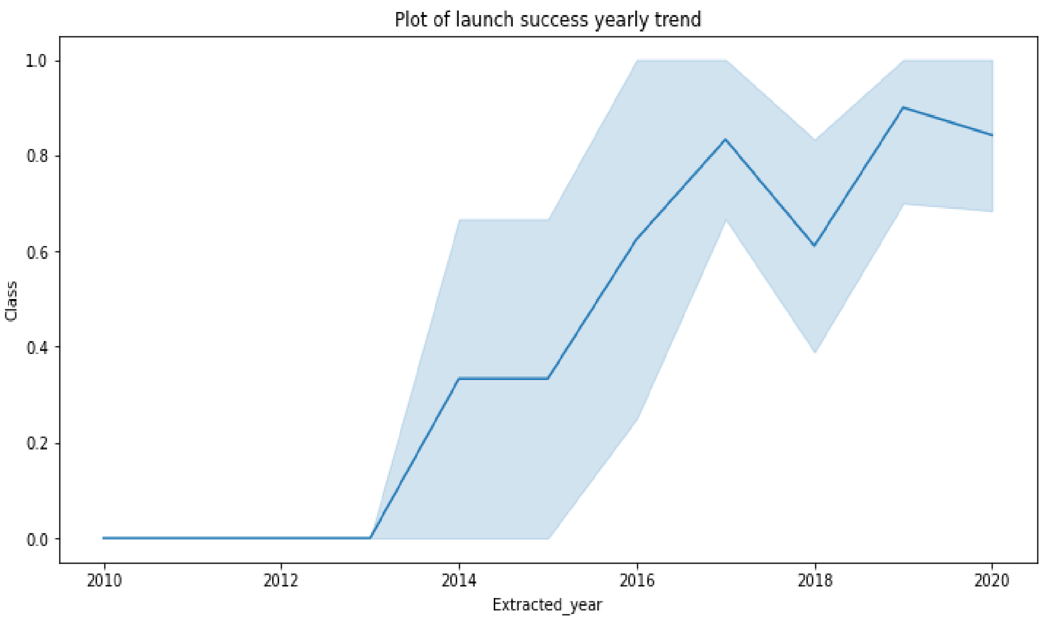
- The data wrangling phase involved transforming raw data into a usable form for analysis.
- This included checking data types, mapping variables, and introducing new columns to facilitate analysis.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- GitHub URL :
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/03_labs-jupyter-spacex-Data%20wrangling.ipynb



EDA with Data Visualization

- Through the application of aggregate functions, we gained valuable insights. Graphical representations using Matplotlib and Seaborn aided in preparing data feature engineering.
- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- GitHub URL:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/05_jupyter-labs-eda-dataviz.ipynb





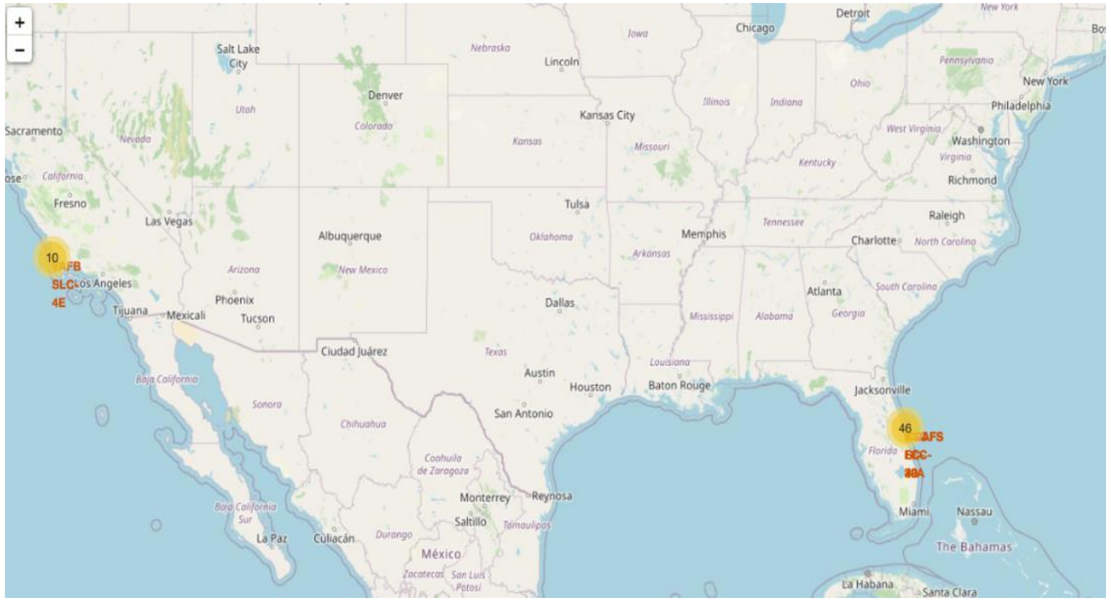
EDA with SQL

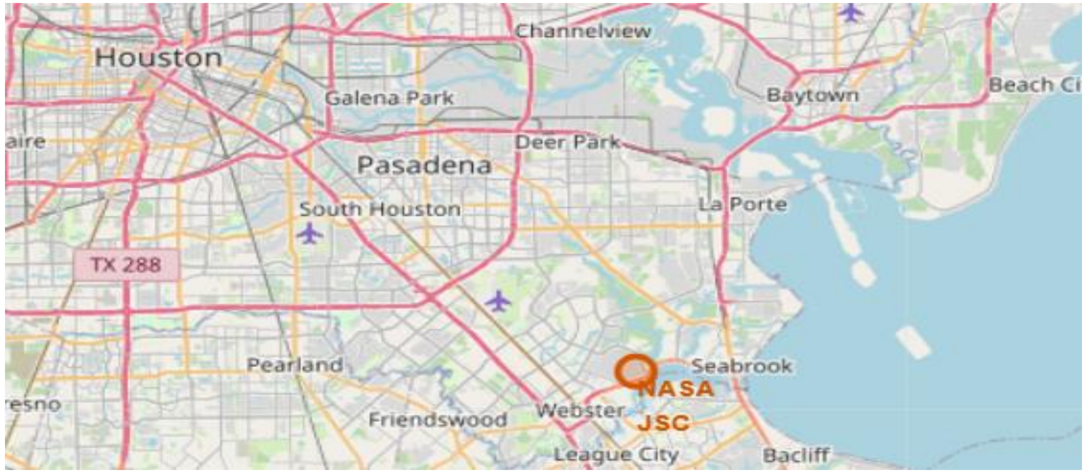
- In this case, I did not use the jupyter notebook, whereas I directly imported the dataset in Microsoft SQL server and started querying the data there.
- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- GitHub URL of the sql file:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/04_EDA_spacex.sql

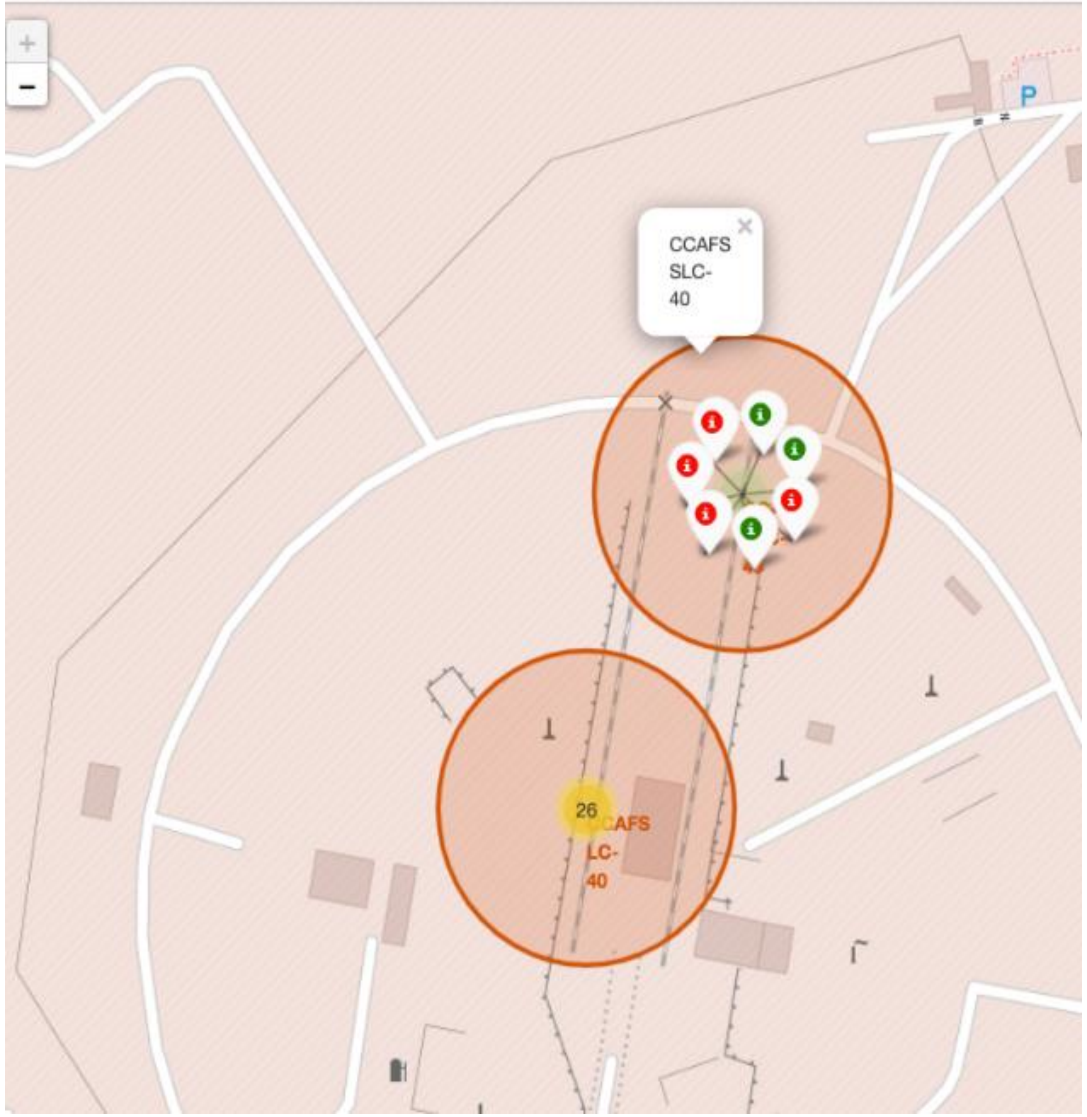
Build an Interactive Map with Folium

- Beyond numeric features, we delved into external and geographical factors critical to our analysis.
- Employing the Folium library, we analysed launch site locations, considering factors like rocket trajectories and proximity. Marking locations with latitude and longitude from the dataset, we uncovered new insights, particularly in the quest for optimal launch site locations.
- 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.
- GitHub URL:
    - https://github.com/RaghavendarV/IBM-Capstone/blob/main/06_lab_jupyter_launch_site_location.ipynb

Build an Interactive Map with Folium

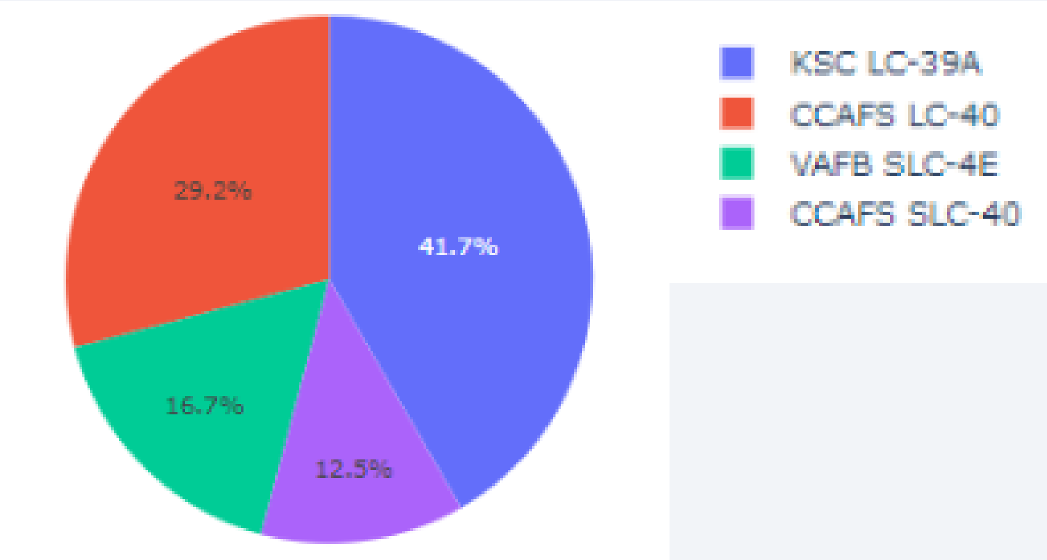






Build a Dashboard with Plotly Dash

- We developed an interactive dashboard using Plotly Dash, showcasing pie charts that illustrate the total launches at specific sites.
- Additionally, we created scatter graphs to visualize the relationship between the outcome and payload mass (in kilograms) for different booster versions.
- GitHub URL of python file:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/07_spacex_interactive_dash.py

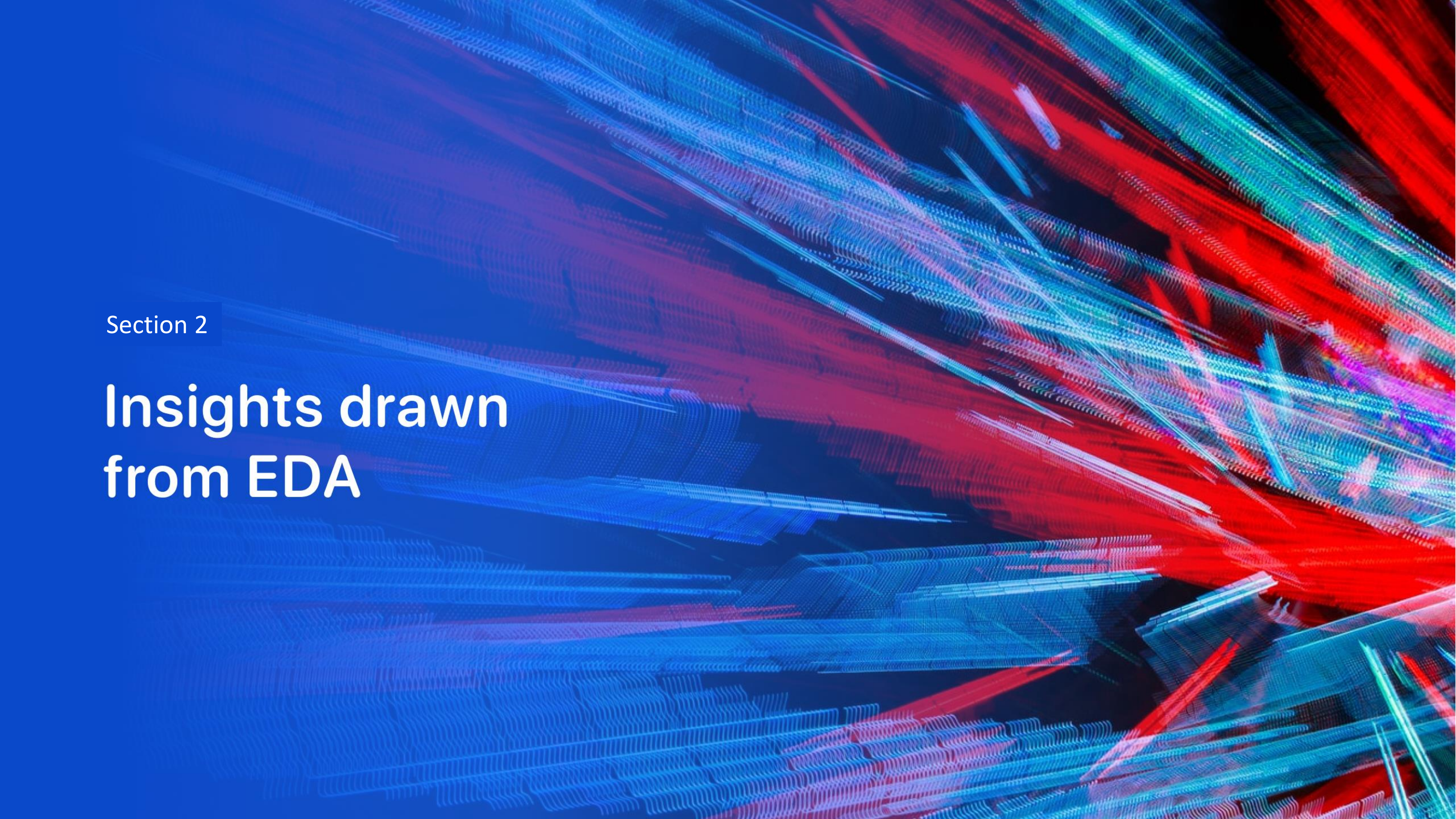


Predictive Analysis (Classification)

- In the final stage, we aimed to predict mission outcomes using machine learning.
- We loaded the data using numpy and pandas, Focusing on the first stage of Falcon 9 landings, we split the data into training and test sets.
- 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.
- GitHub URL:
  - https://github.com/RaghavendarV/IBM-Capstone/blob/main/08_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

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

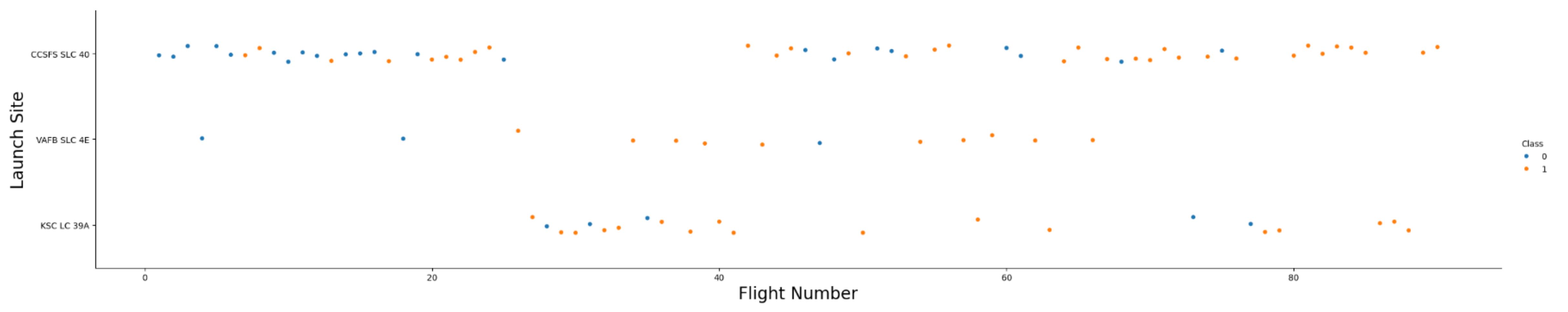


Section 2

Insights drawn from EDA

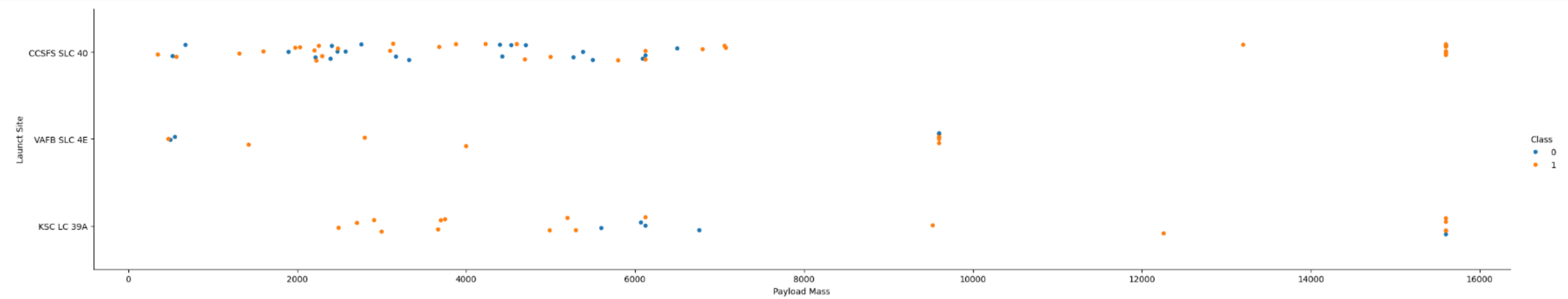
Flight Number vs. Launch Site

- scatter plot of Flight Number vs. Launch Site:
- The plot revealed a positive correlation between the number of flights at a launch site and the success rate at that site, indicating that a higher flight count corresponds to a greater success rate.



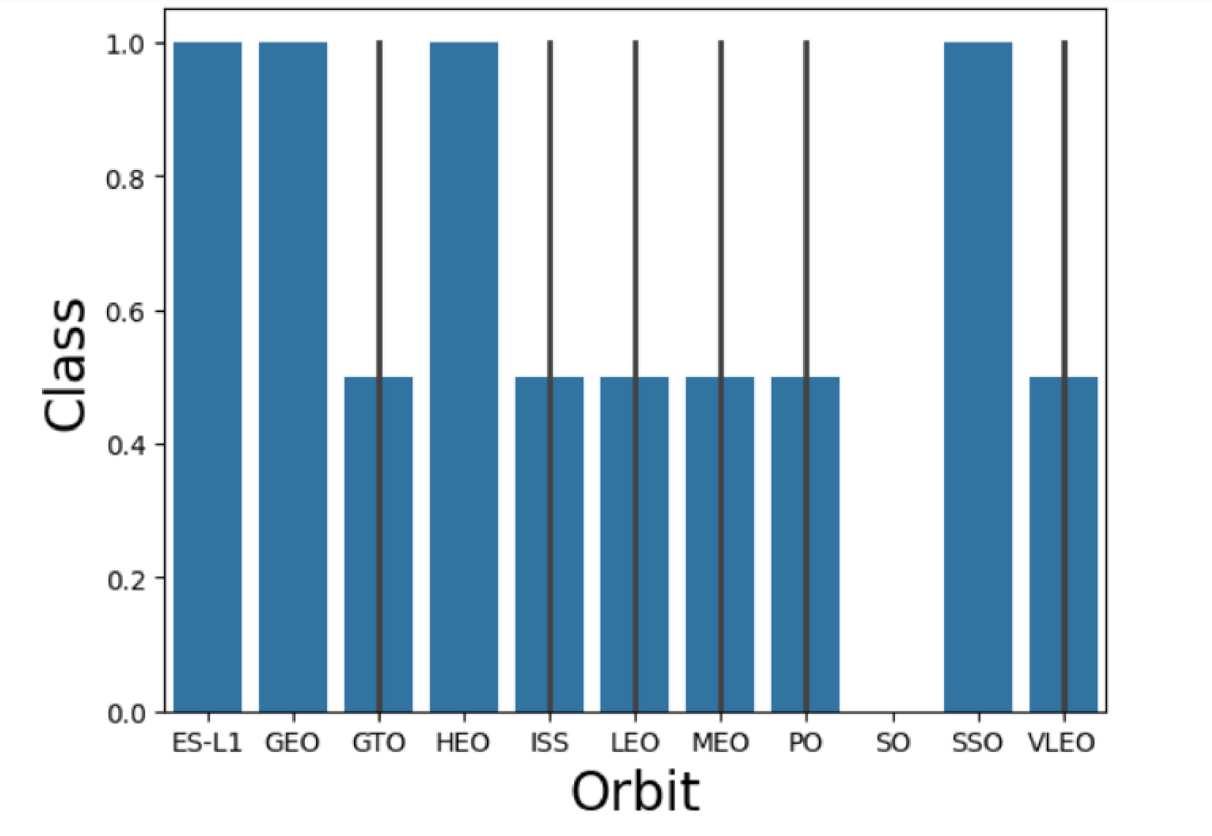
Payload vs. Launch Site

- scatter plot of Payload vs. Launch Site
- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.



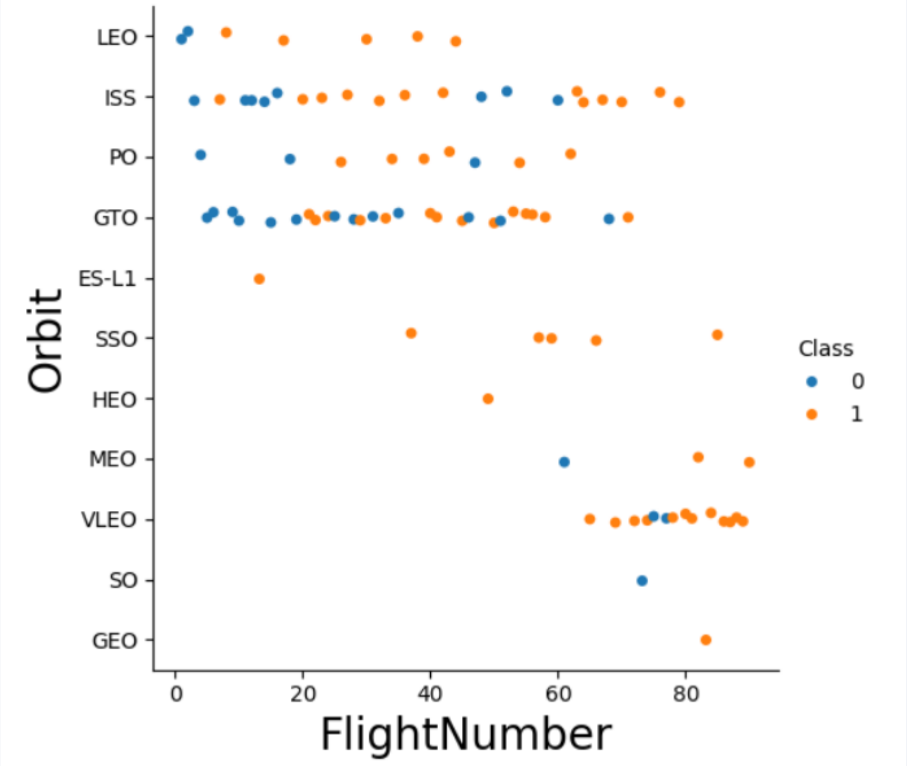
Success Rate vs. Orbit Type

- bar chart for the success rate of each orbit type
- The plot illustrates that ES-L1, GEO, HEO, SSO, and VLEO exhibited the highest success rates.



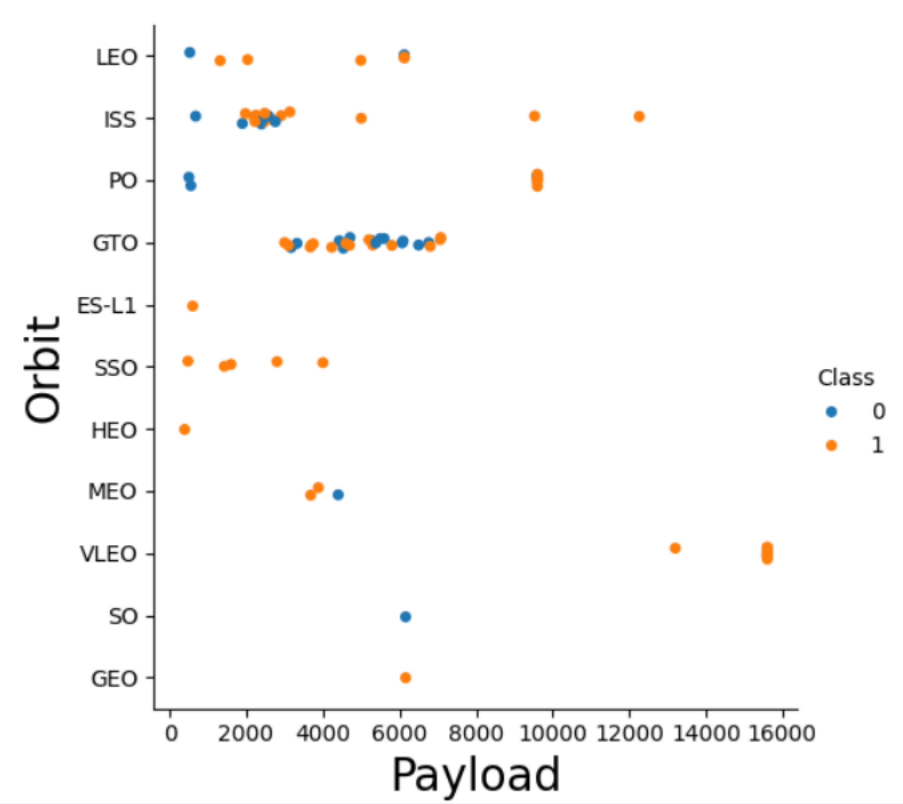
Flight Number vs. Orbit Type

- scatter point of Flight number vs. Orbit type
- The chart depicts the relationship between Flight Number and Orbit type. It is evident that in the LEO orbit, success is correlated with the number of flights, while in the GTO orbit, there is no discernible relationship between flight number and orbit.



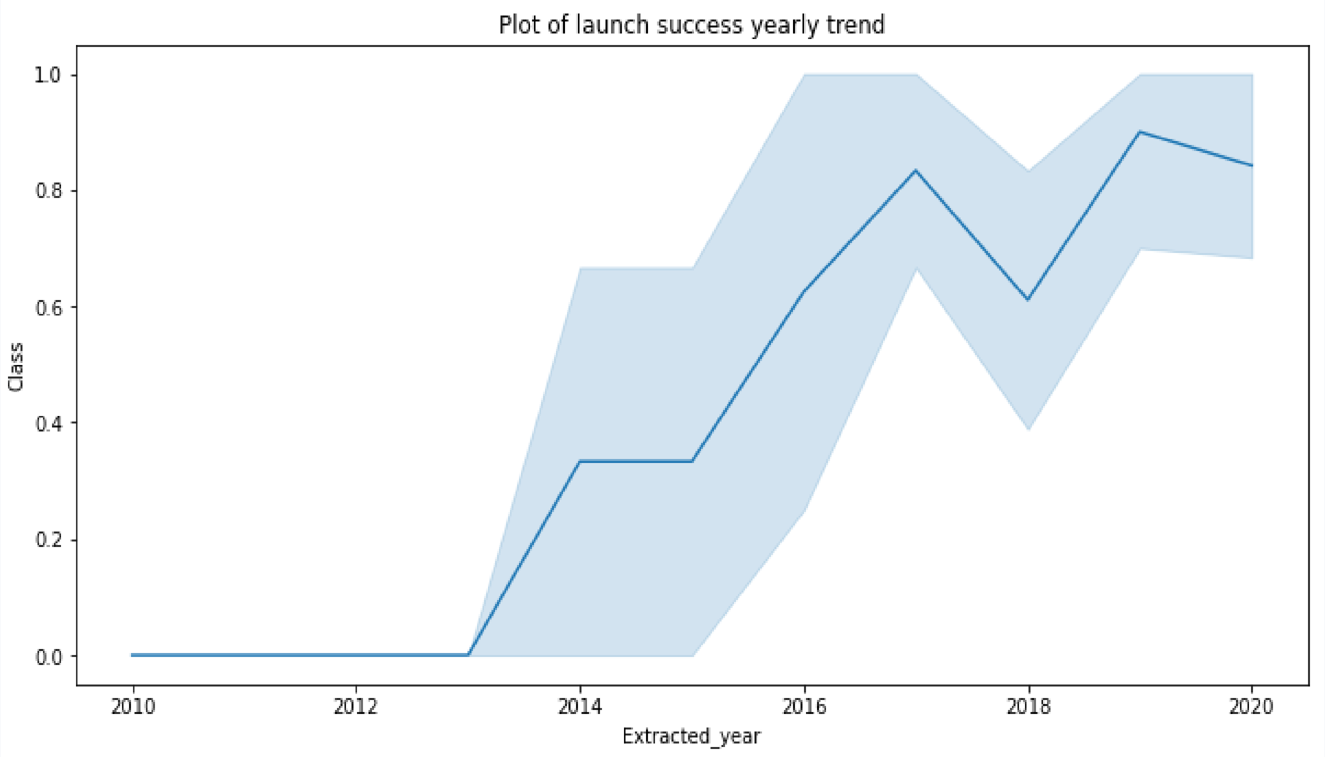
Payload vs. Orbit Type

- scatter point of payload vs. orbit type
- It is noticeable that heavier payloads correspond to a higher rate of successful landings, particularly for PO, LEO, and ISS orbits.



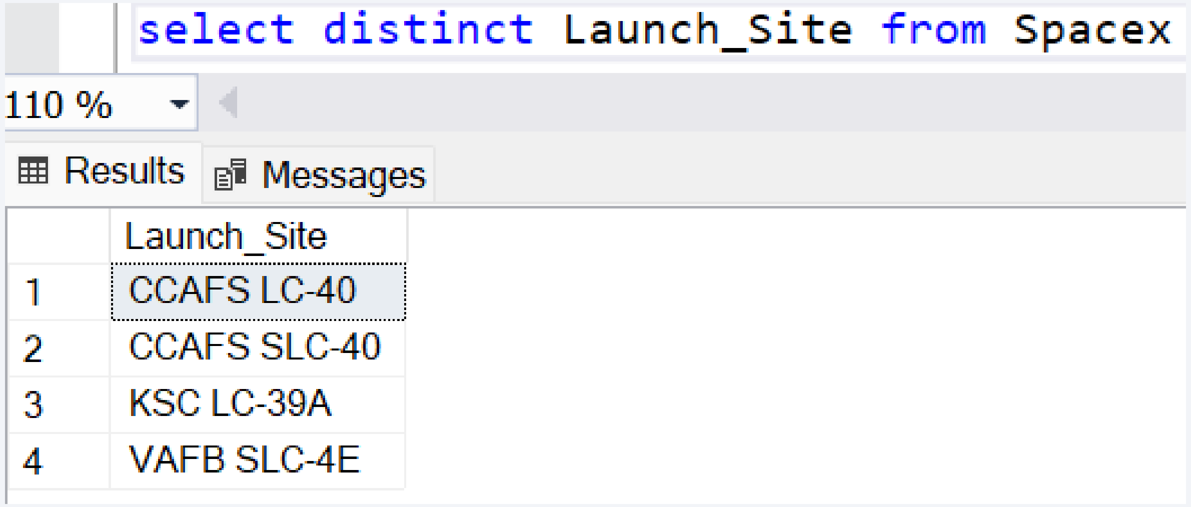
Launch Success Yearly Trend

- line chart of yearly average success rate
- The plot indicates that the success rate has been consistently increasing from 2013 to 2020.



All Launch Site Names

- names of the unique launch sites
- We utilized the DISTINCT keyword to display only the unique launch sites from the SpaceX data.

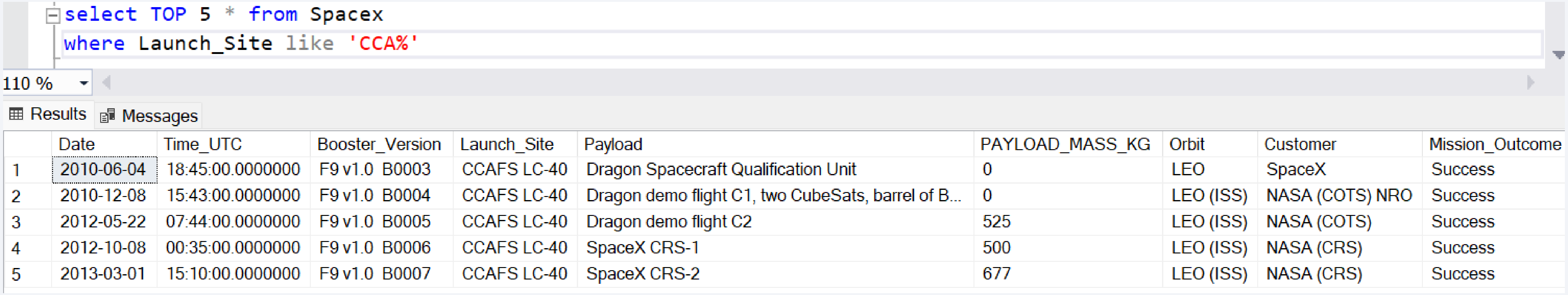


```
select distinct Launch_Site from Spacex
```

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

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

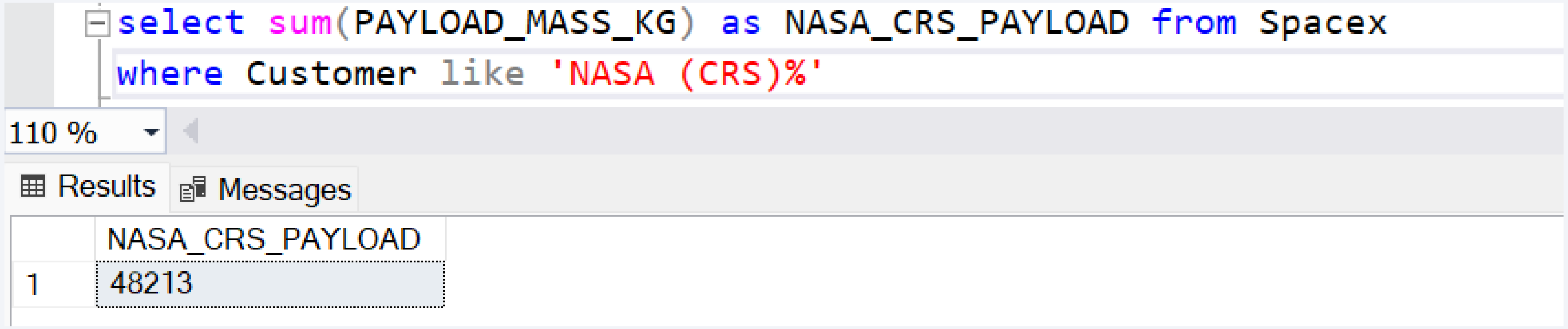


```
select TOP 5 * from Spacex
where Launch_Site like 'CCA%'
```

	Date	Time_UTC	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG	Orbit	Customer	Mission_Outcome
1	2010-06-04	18:45:00.0000000	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
2	2010-12-08	15:43:00.0000000	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of B...	0	LEO (ISS)	NASA (COTS) NRO	Success
3	2012-05-22	07:44:00.0000000	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
4	2012-10-08	00:35:00.0000000	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
5	2013-03-01	15:10:00.0000000	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

- Calculate the total payload carried by boosters from NASA.
- This SQL query retrieves the total payload mass (in kilograms) for SpaceX launches with customers whose names start with "NASA (CRS)". It uses the SUM() function to calculate the sum of the payload masses and filters the data based on the specified customer condition.

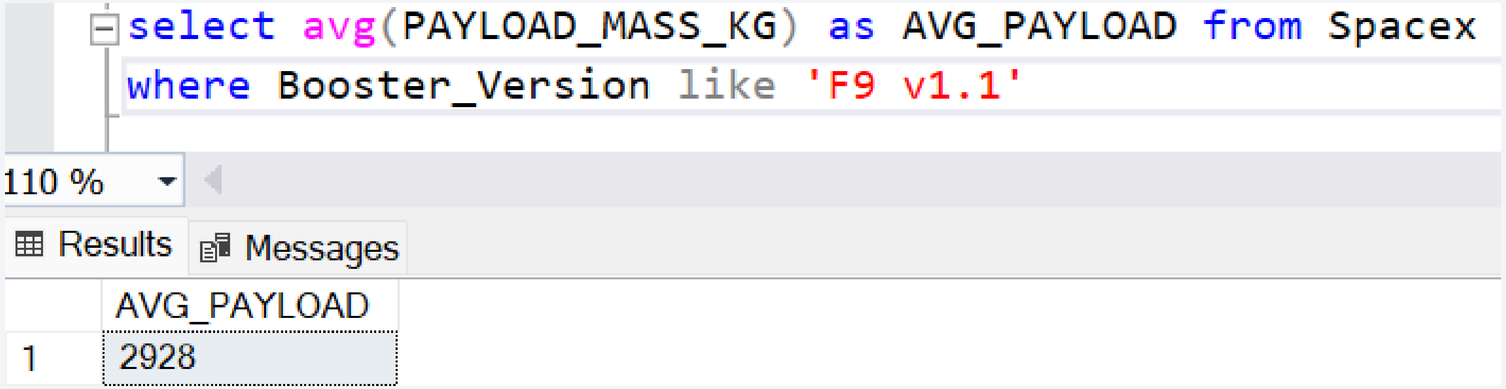


```
select sum(PAYLOAD_MASS_KG) as NASA_CRS_PAYLOAD from Spacex
where Customer like 'NASA (CRS)%'
```

	NASA_CRS_PAYLOAD
1	48213

Average Payload Mass by F9 v1.1

- average payload mass carried by booster version F9 v1.1
- This SQL query calculates the average payload mass (in kilograms) for SpaceX launches where the Booster Version is 'F9 v1.1'. It uses the AVG() function to compute the average payload mass and filters the data based on the specified booster version condition.

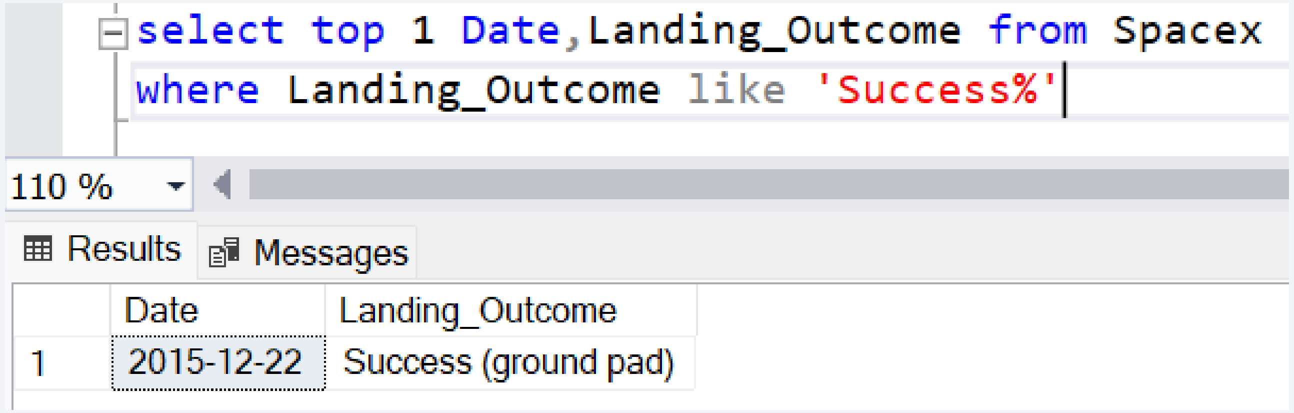


```sql
select avg(PAYLOAD_MASS_KG) as AVG_PAYLOAD from Spacex
where Booster_Version like 'F9 v1.1'
```

	AVG_PAYLOAD
1	2928

First Successful Ground Landing Date

- dates of the first successful landing outcome on ground pad
- This SQL query retrieves the top (most recent) record from the SpaceX dataset where the Landing Outcome is labeled as 'Success%' (indicating a successful landing
- The result will show the date and landing outcome of the most recent successful landing in the dataset.

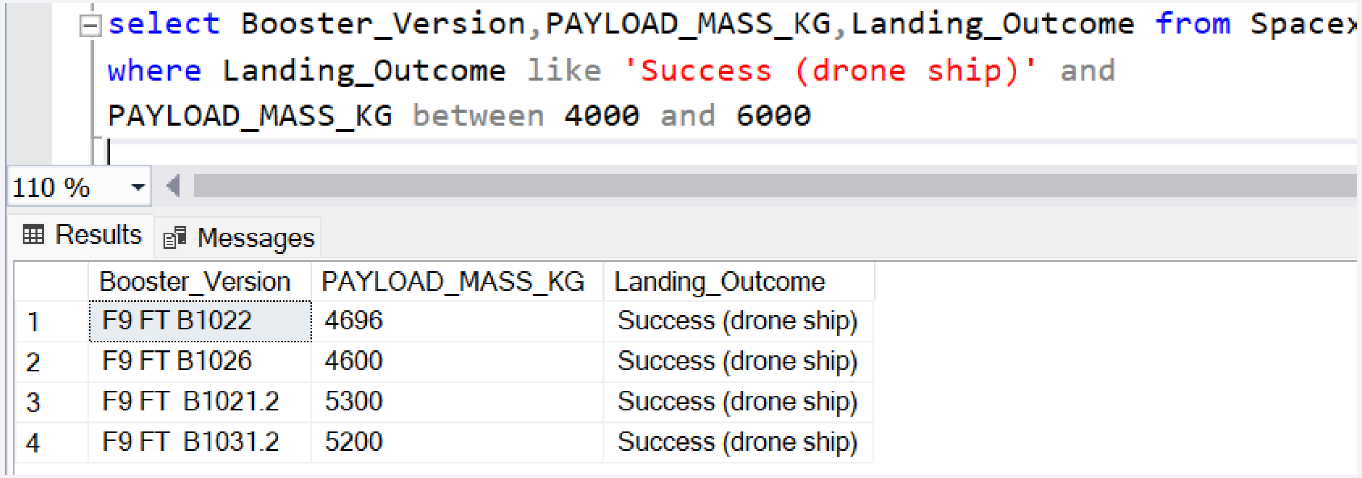


```sql
select top 1 Date,Landing_Outcome from Spacex
where Landing_Outcome like 'Success%'
```

	Date	Landing_Outcome
1	2015-12-22	Success (ground pad)

Successful Drone Ship Landing with Payload between 4000 and 6000

- names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000
- This SQL query retrieves records from the SpaceX dataset where the Landing Outcome is labeled as 'Success (drone ship)' and the PAYLOAD_MASS_KG falls within the range of 4000 to 6000 using “BETWEEN” operator.

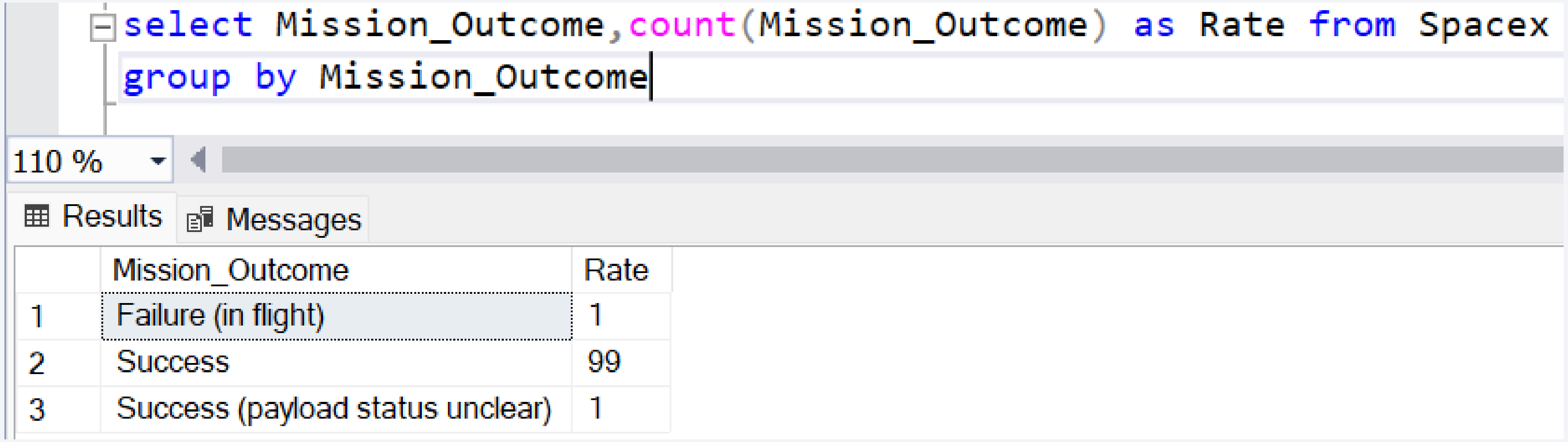


```sql
select Booster_Version,PAYLOAD_MASS_KG,Landing_Outcome from Spacex
where Landing_Outcome like 'Success (drone ship)' and
PAYLOAD_MASS_KG between 4000 and 6000
```

	Booster_Version	PAYLOAD_MASS_KG	Landing_Outcome
1	F9 FT B1022	4696	Success (drone ship)
2	F9 FT B1026	4600	Success (drone ship)
3	F9 FT B1021.2	5300	Success (drone ship)
4	F9 FT B1031.2	5200	Success (drone ship)

Total Number of Successful and Failure Mission Outcomes

- total number of successful and failure mission outcomes
- This SQL query counts the occurrences of each unique value in the 'Mission_Outcome' column in the SpaceX dataset. The COUNT function is used to count the occurrences, and the result is grouped by the 'Mission_Outcome'.

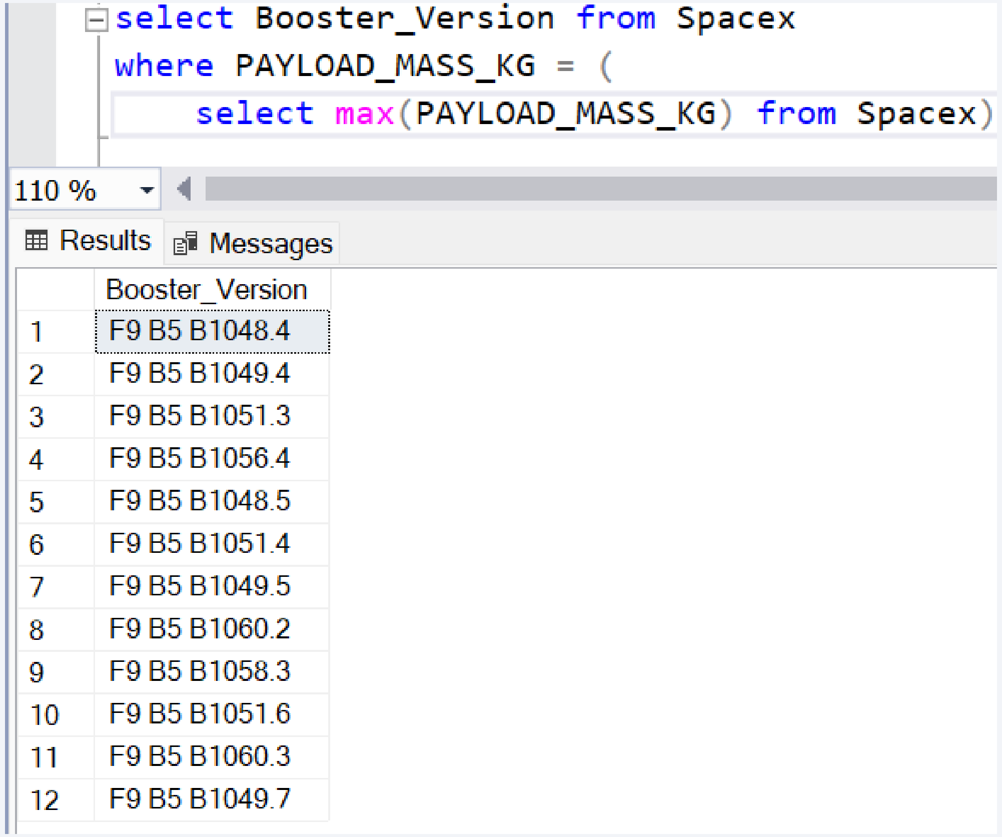


```
select Mission_Outcome,count(Mission_Outcome) as Rate from Spacex
group by Mission_Outcome
```

	Mission_Outcome	Rate
1	Failure (in flight)	1
2	Success	99
3	Success (payload status unclear)	1

Boosters Carried Maximum Payload

- names of the booster which have carried the maximum payload mass
- This SQL query retrieves the 'Booster_Version' from the SpaceX dataset for the record where the 'PAYLOAD_MASS_KG' is equal to the maximum payload mass in the entire dataset.
- The inner subquery calculates the maximum payload mass using the MAX() function, and the outer query retrieves the corresponding 'Booster_Version' for the record with that maximum payload mass.

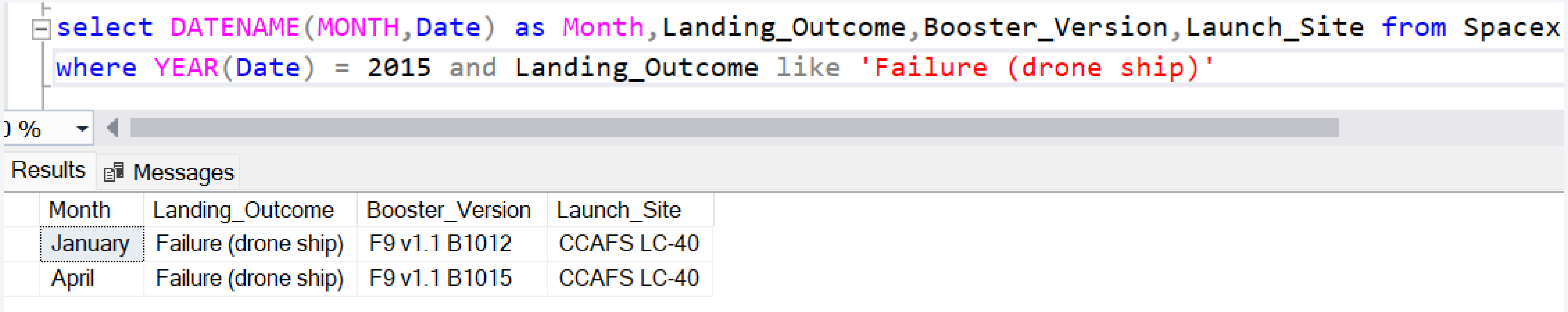


```
select Booster_Version from Spacex
where PAYLOAD_MASS_KG = (
    select max(PAYLOAD_MASS_KG) from Spacex)
```

	Booster_Version
1	F9 B5 B1048.4
2	F9 B5 B1049.4
3	F9 B5 B1051.3
4	F9 B5 B1056.4
5	F9 B5 B1048.5
6	F9 B5 B1051.4
7	F9 B5 B1049.5
8	F9 B5 B1060.2
9	F9 B5 B1058.3
10	F9 B5 B1051.6
11	F9 B5 B1060.3
12	F9 B5 B1049.7

2015 Launch Records

- failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- We used a combinations of the **WHERE** clause, **LIKE**, **AND**, and **BETWEEN** conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

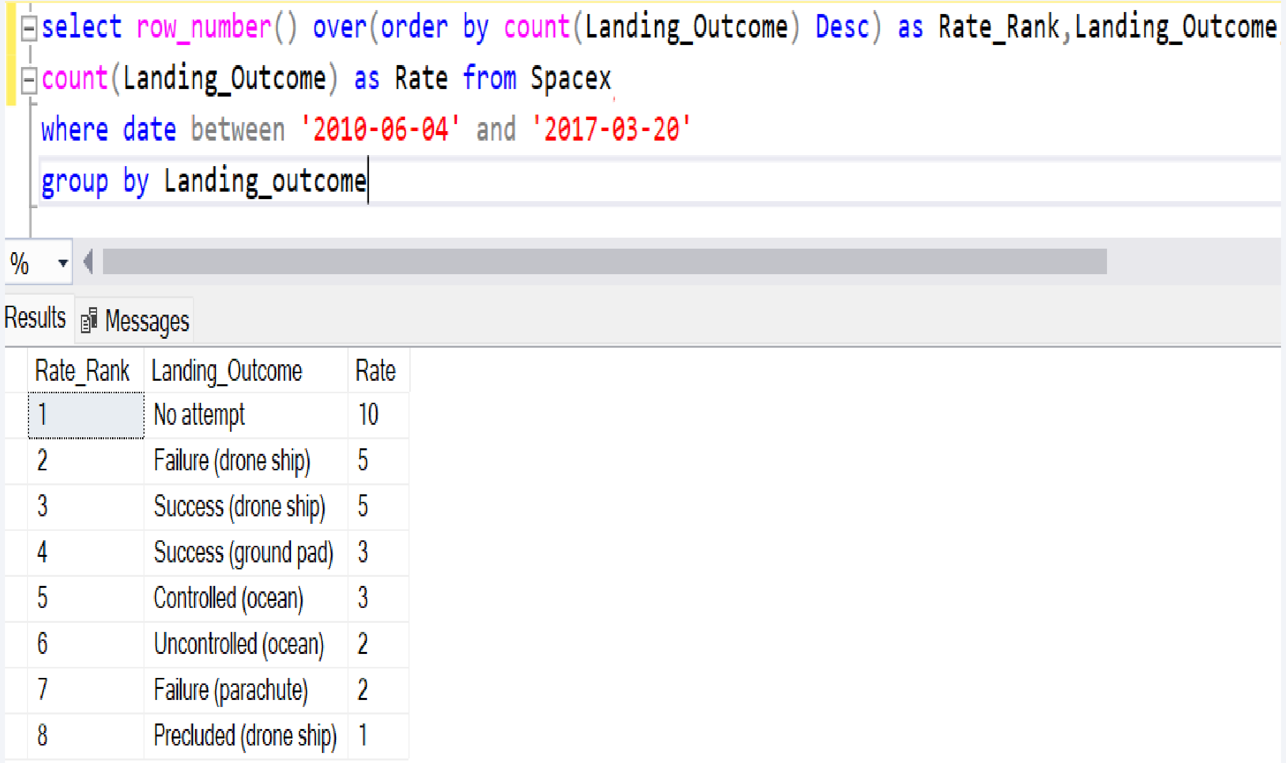


```
select DATENAME(MONTH,Date) as Month,Landing_Outcome,Booster_Version,Launch_Site from Spacex
where YEAR(Date) = 2015 and Landing_Outcome like 'Failure (drone ship)'
```

Month	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

- We selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.
- We applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.



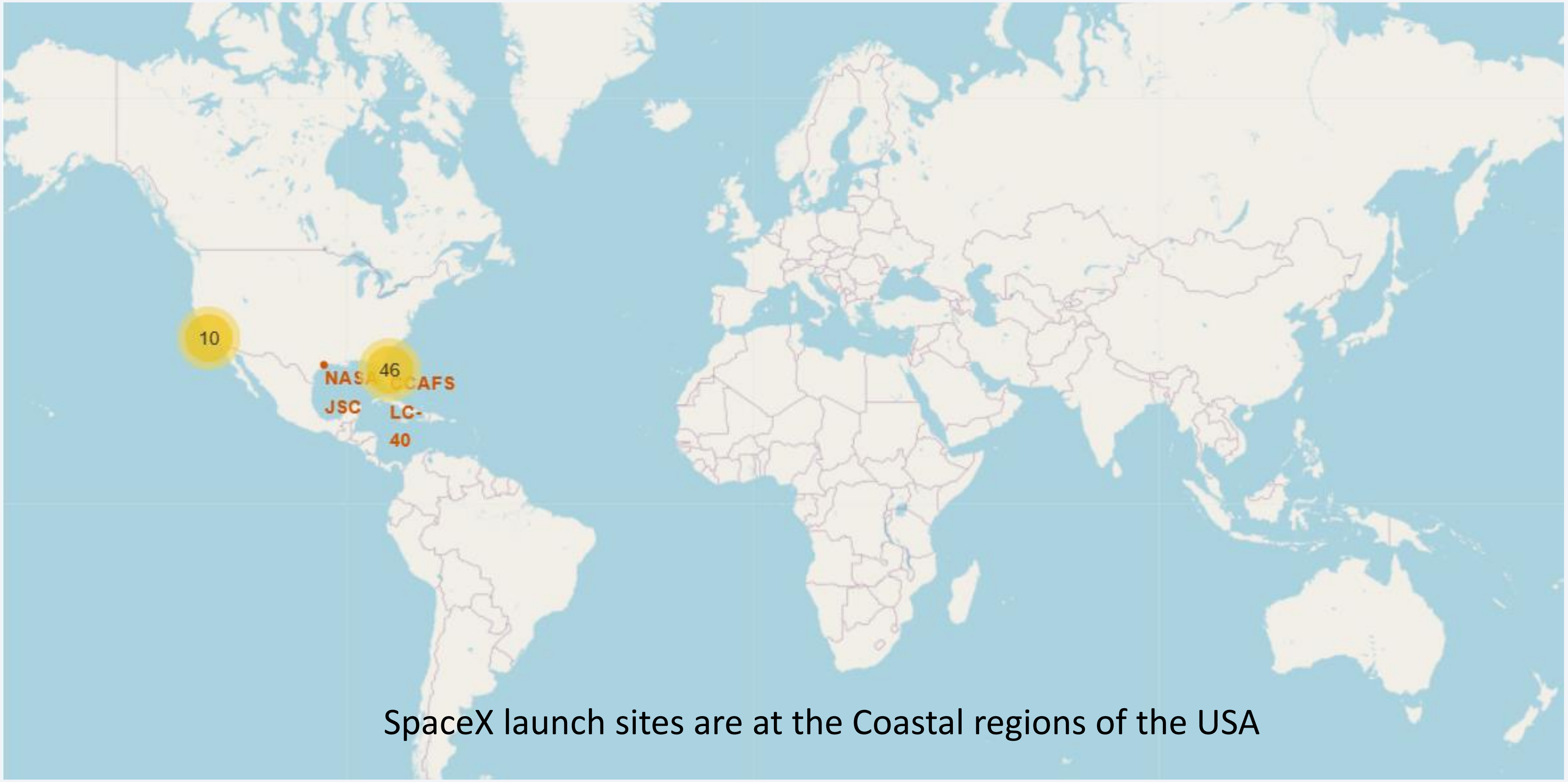
```
select row_number() over(order by count(Landing_Outcome) Desc) as Rate_Rank,Landing_Outcome
count(Landing_Outcome) as Rate from Spacex
where date between '2010-06-04' and '2017-03-20'
group by Landing_outcome
```

Rate_Rank	Landing_Outcome	Rate
1	No attempt	10
2	Failure (drone ship)	5
3	Success (drone ship)	5
4	Success (ground pad)	3
5	Controlled (ocean)	3
6	Uncontrolled (ocean)	2
7	Failure (parachute)	2
8	Precluded (drone ship)	1

Section 3

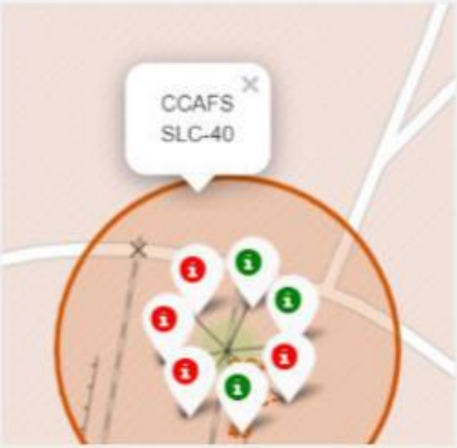
Launch Sites Proximities Analysis

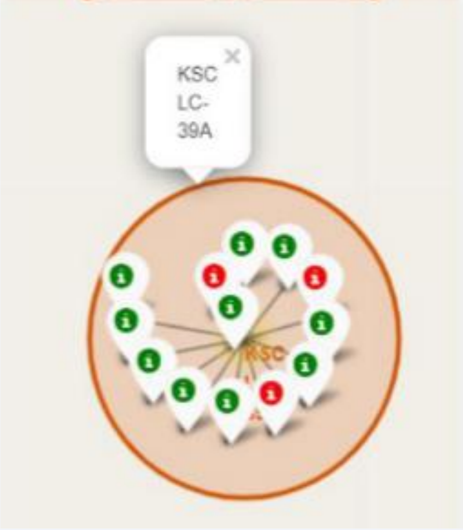
all launch sites location markers on a global map

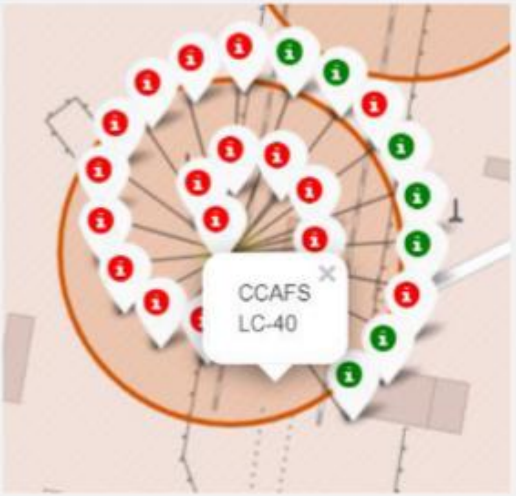


SpaceX launch sites are at the Coastal regions of the USA

Markers showing launch sites with color labels



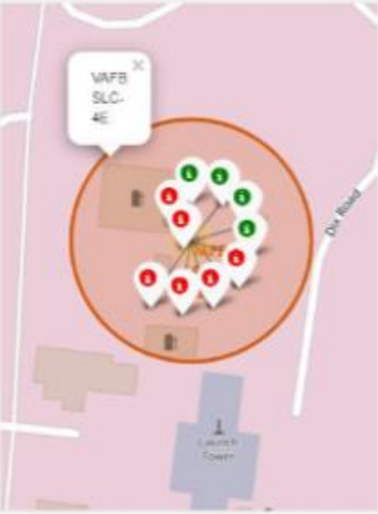




7
SLC-
40
26
LC-
40
Centaur Road

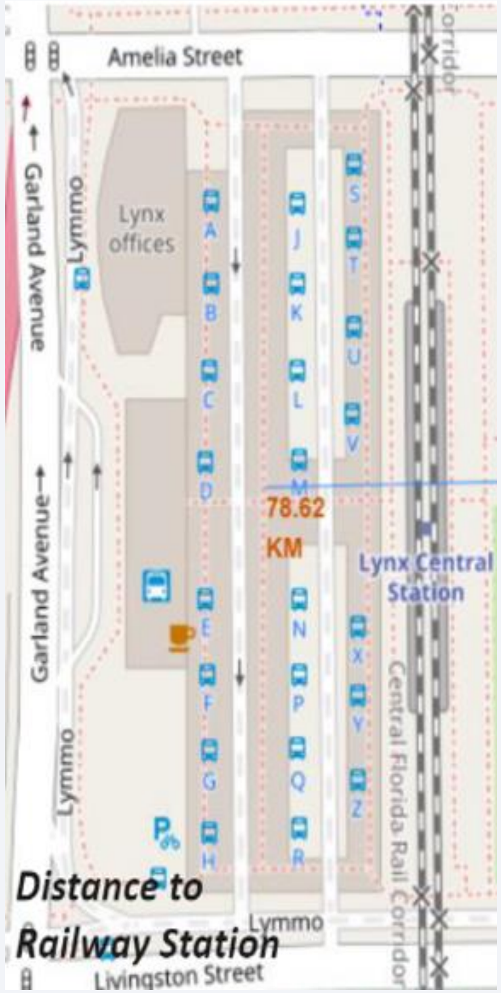
Florida Launch Sites

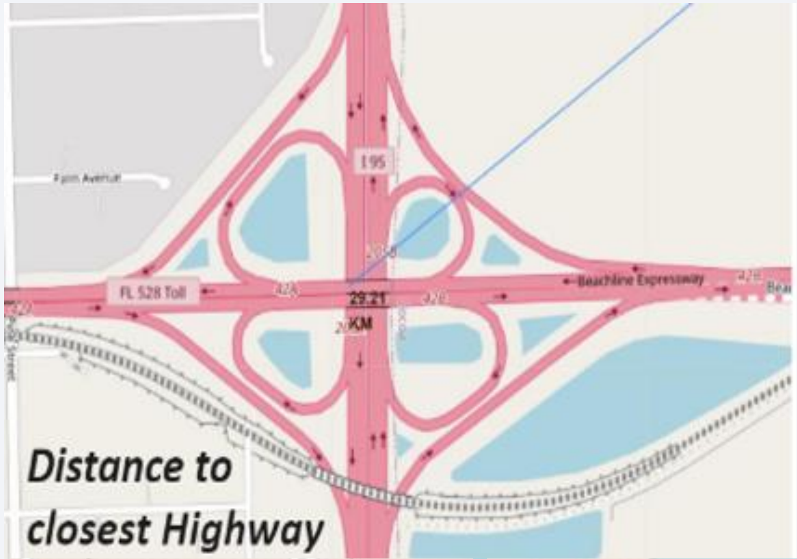
Green Marker shows successful Launches and Red Marker shows Failures

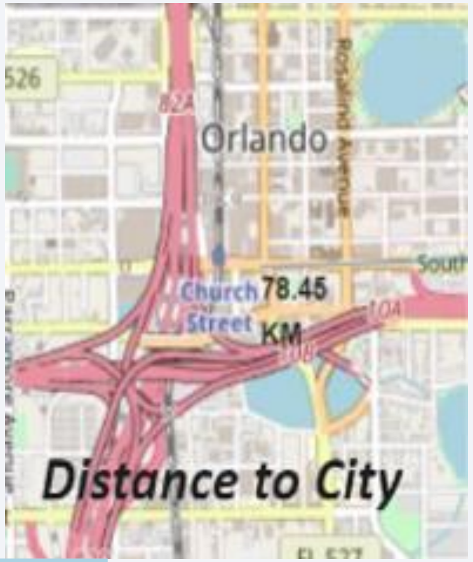


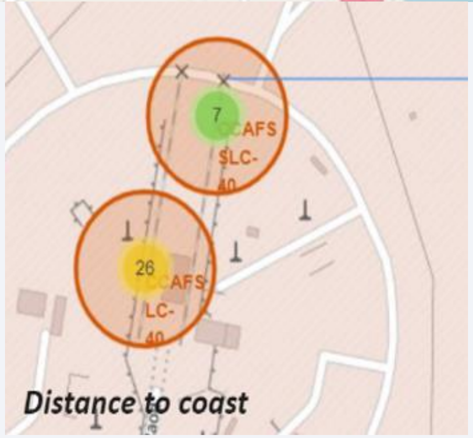
California Launch Site

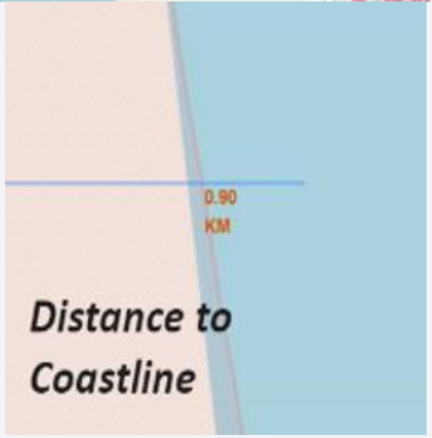
Launch Site distance to landmarks





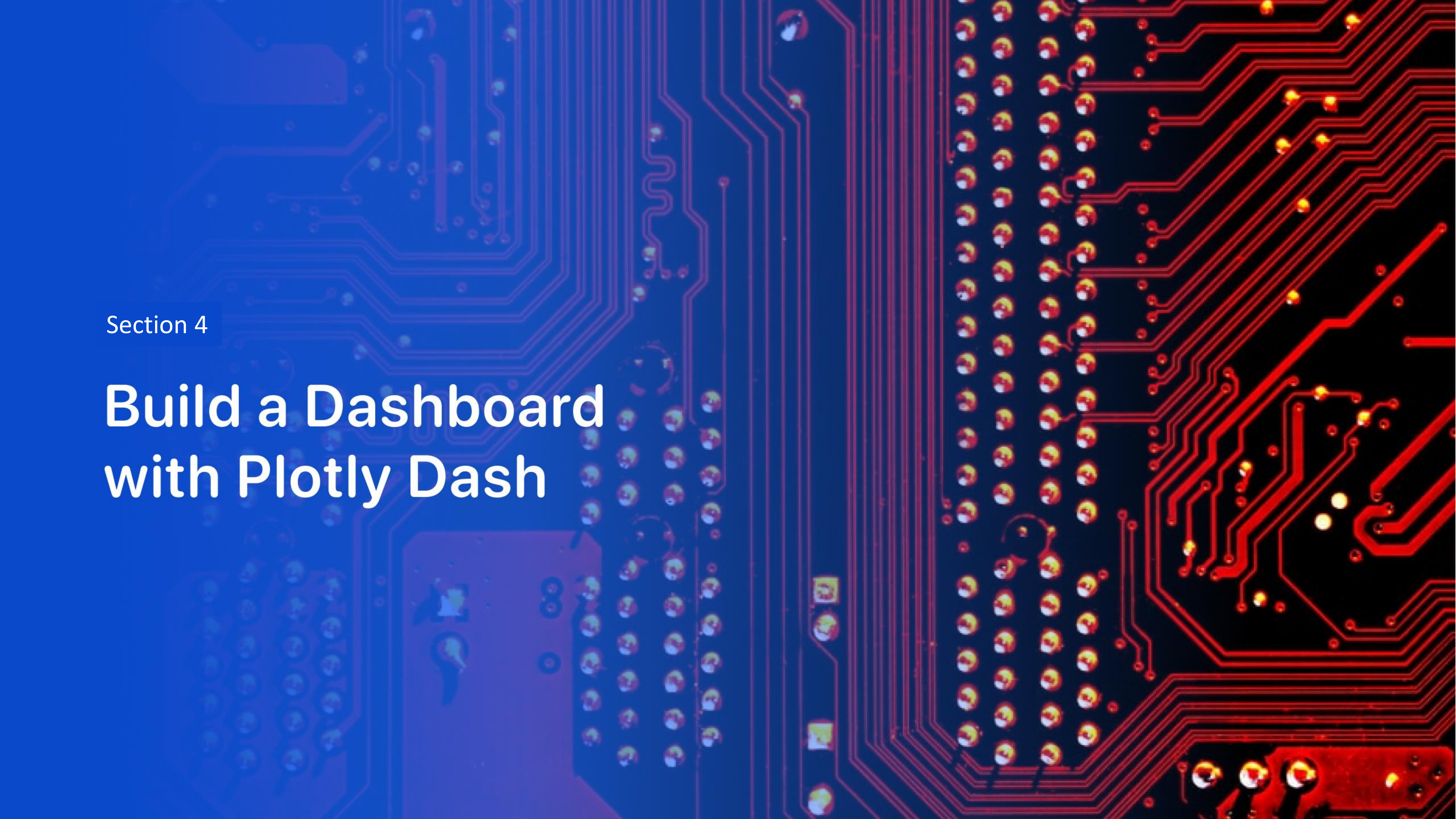




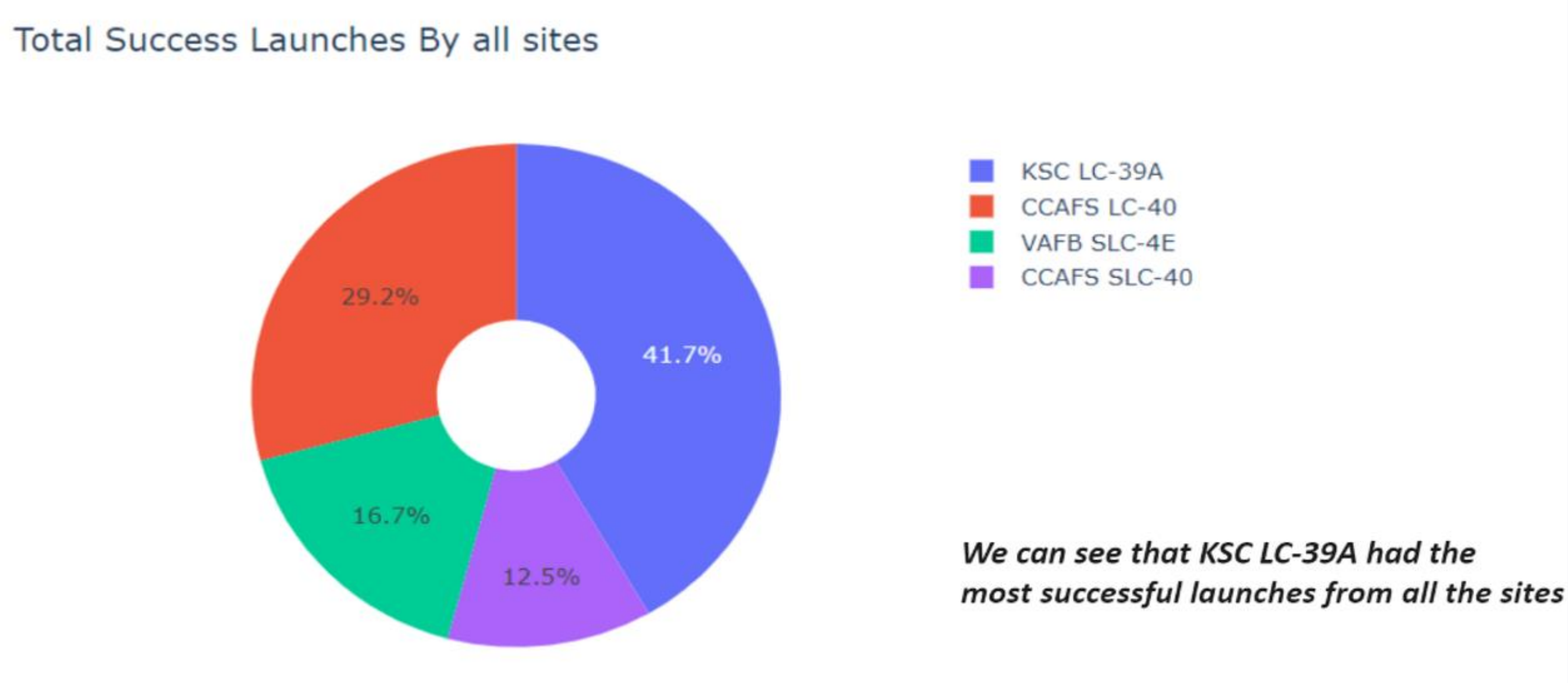


Section 4

Build a Dashboard with Plotly Dash

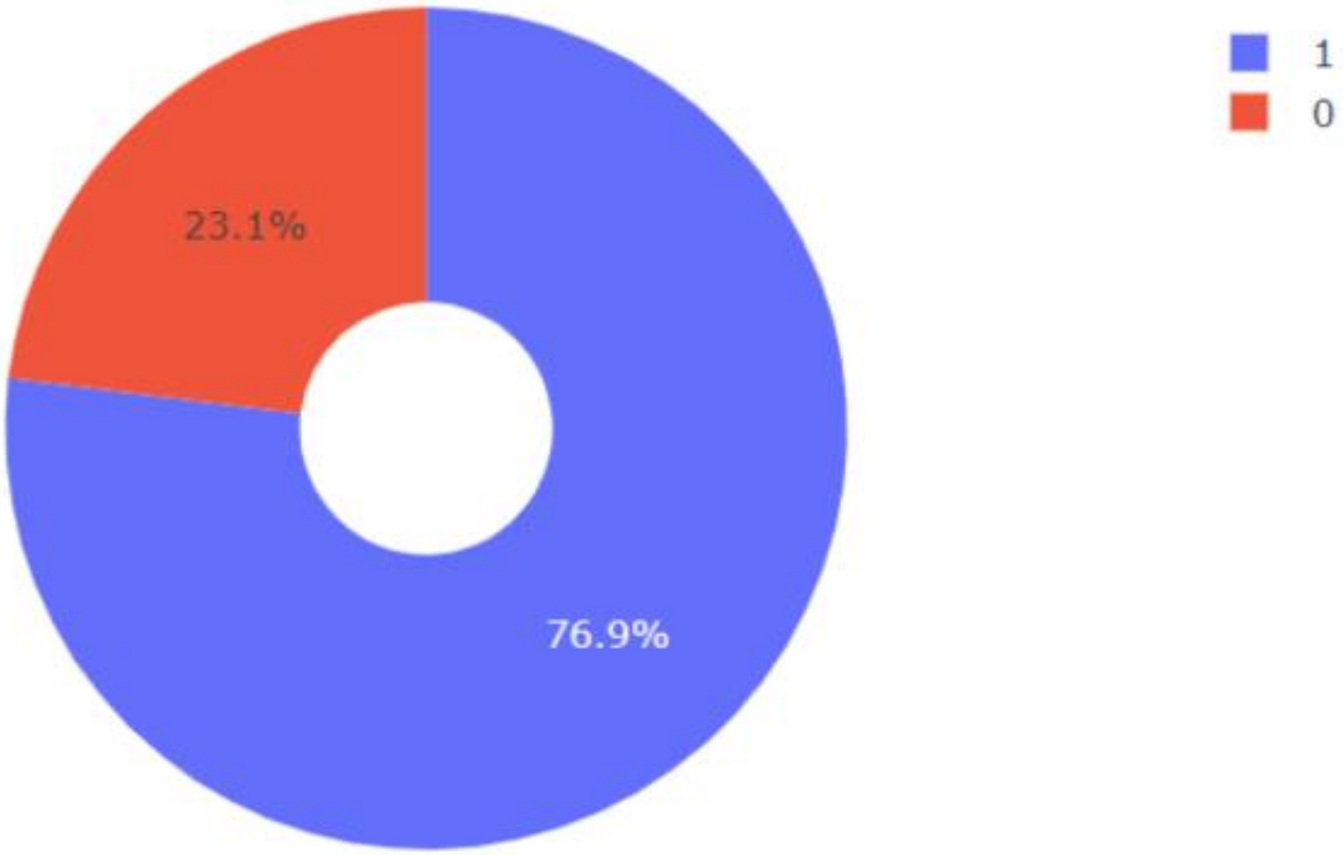


Pie chart showing the success percentage achieved by each launch site



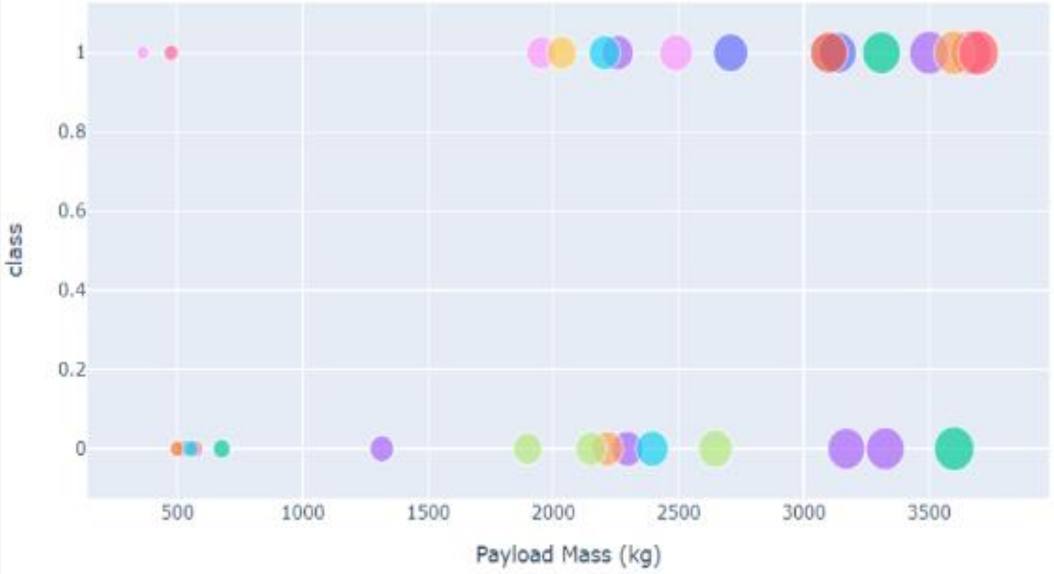
We can see that KSC LC-39A had the most successful launches from all the sites

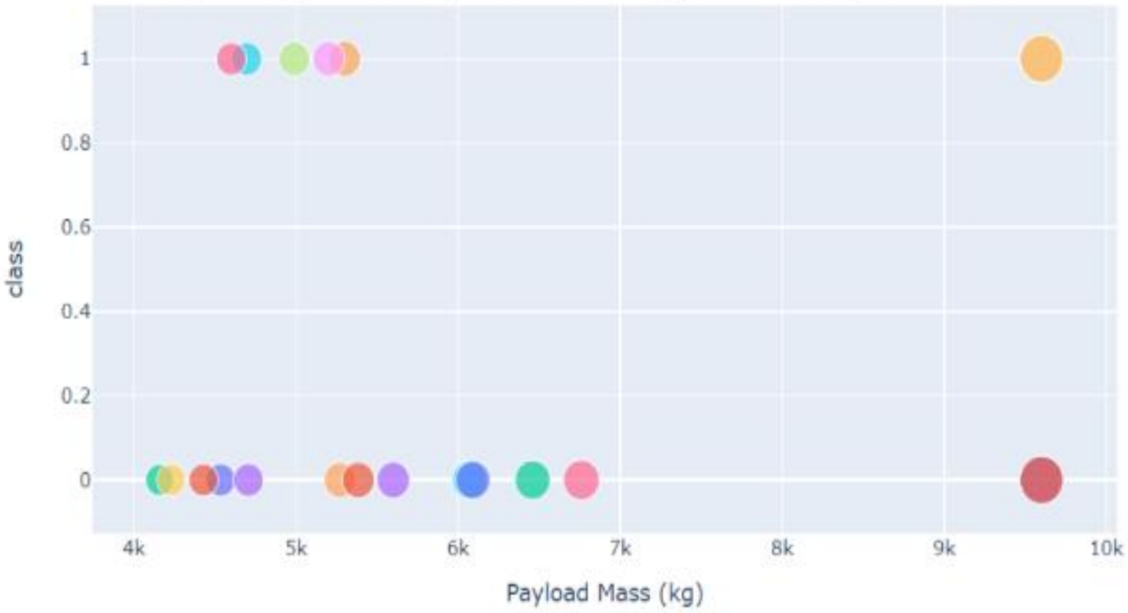
Pie chart showing the Launch site with the highest launch success ratio



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


Low Weighted Payload 0kg – 4000kg


Heavy Weighted Payload 4000kg – 10000kg

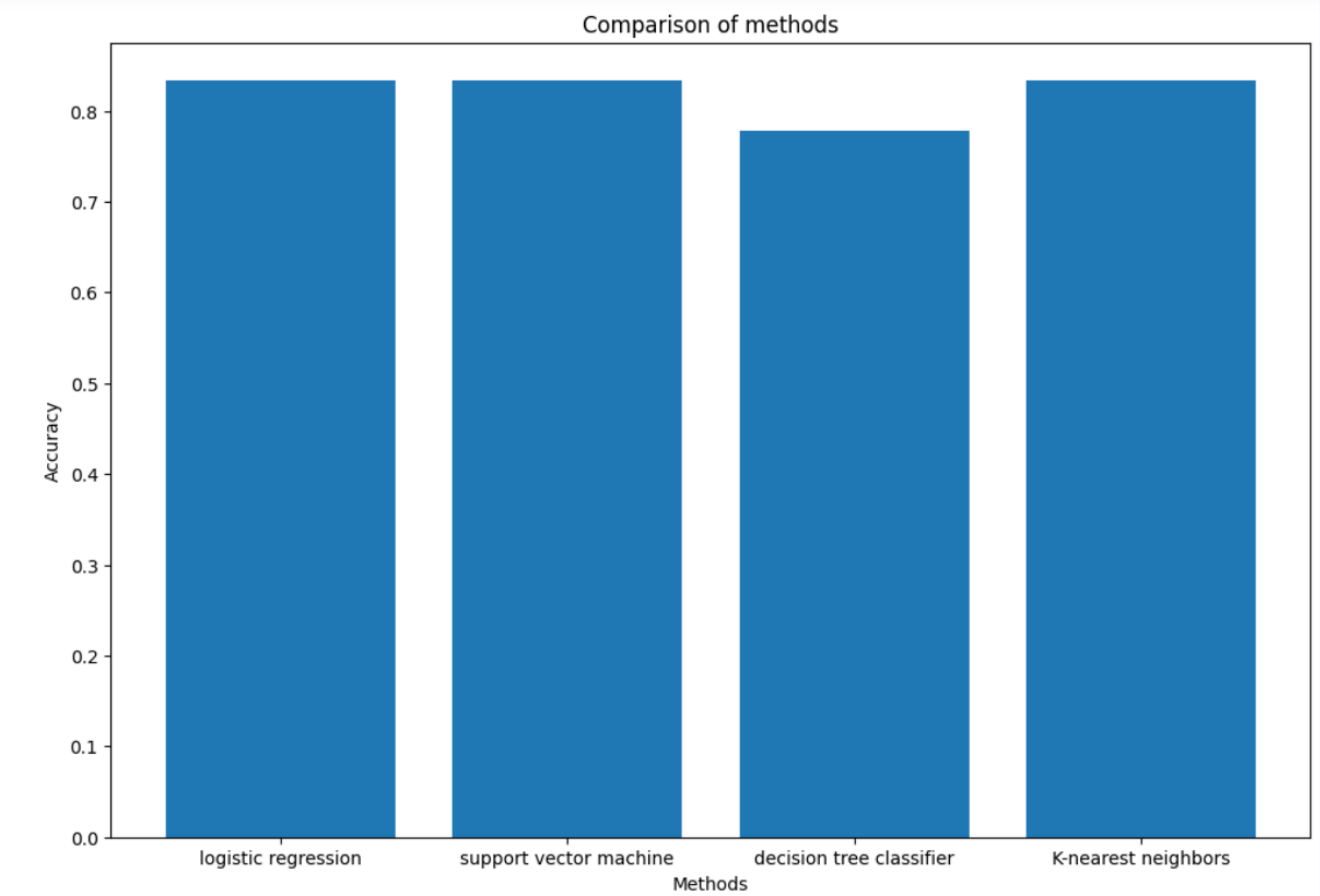
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

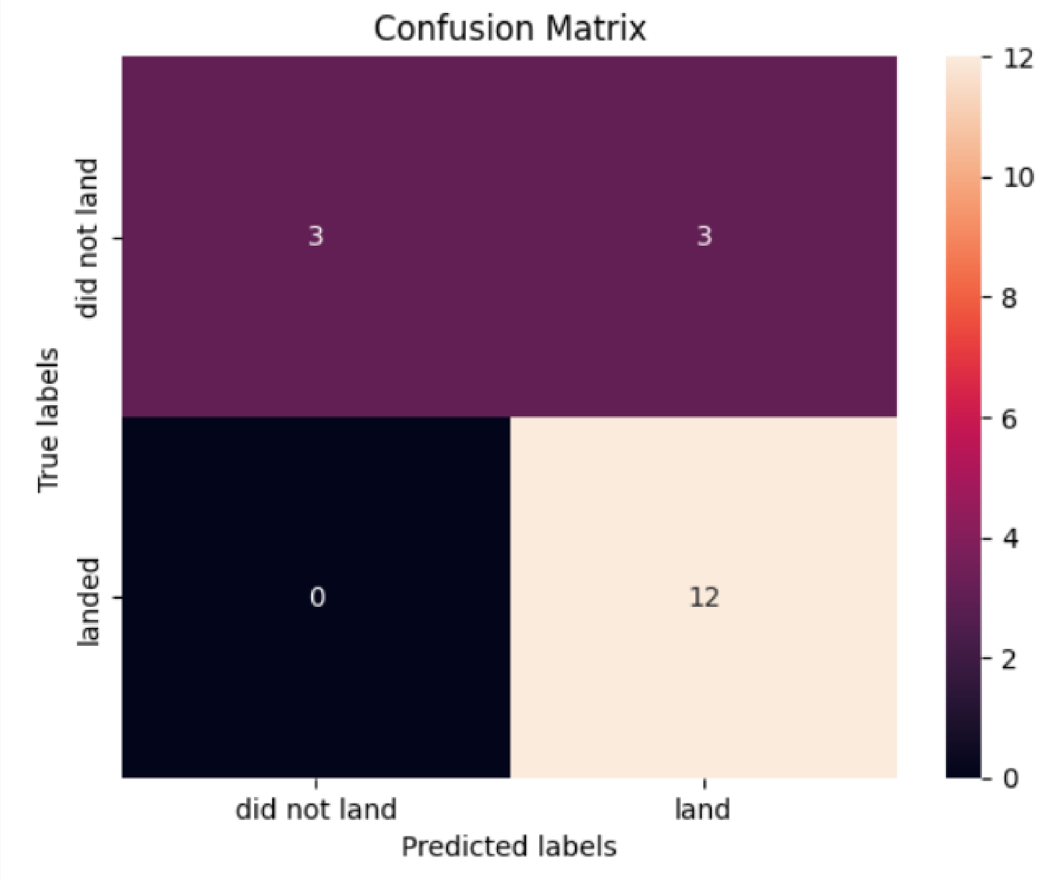
Predictive Analysis (Classification)

Classification Accuracy

- Visualize the built model accuracy for all built classification models, in a bar chart
- The best method can be any of the three in logistic regression, SVM or KNN.



Confusion Matrix



- A confusion matrix summarizes the performance of a classification algorithm
- All the confusion matrices were identical
- The fact that there are false positives (Type 1 error) is not good
- Confusion Matrix Outputs:
  - 12 True positive
  - 3 True negative
  - 3 False positive
  - 0 False Negative

Conclusions

- Model Performance: The models performed similarly on the test set with the decision tree model slightly outperforming
- Equator: Most of the launch sites are near the equator for an additional natural boost - due to the rotational speed of earth - which helps save the cost of putting in extra fuel and boosters
- Coast: All the launch sites are close to the coast
- Launch Success: Increases over time
- KSC LC-39A: Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg
- Orbits: ES-L1, GEO, HEO, and SSO have a 100% success rate
- Payload Mass: Across all launch sites, the higher the payload mass (kg), the higher the success rate

Appendix

- Whole Capstone project relevant datasets, ipynb files, SQL files, text documents are available in the below Github Repo.
- https://github.com/RaghavendarV/IBM-Capstone/tree/main

Thank you!