



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

<Greg Books>  
<5/31/25>



# 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 offers Falcon 9 rocket launches at a cost of \$62 million, whereas other providers charge upwards of \$165 million per launch. A major contributor to this cost advantage is SpaceX's ability to reuse the rocket's first stage.
- Accurately predicting the success of first-stage landings is crucial for estimating launch costs. Such insights could empower competing companies to make informed bids against SpaceX in the commercial launch market.

## **Problems we want to answer**

- Which factors most influence the success of a rocket's first-stage landing?
- How do these factors interact to impact the landing outcome?
- What operational conditions are necessary to ensure consistent and successful landings?

Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

---

## **API Access**

Launch data was retrieved by sending GET requests to the SpaceX API. The response content was decoded in JSON format using the `.json()` function and then converted into a pandas DataFrame using `json normalize()`.

## **Data Cleaning**

The resulting dataset was cleaned by checking for and handling missing values appropriately.

## **Web Scraping**

Additional launch records were obtained by scraping Wikipedia using BeautifulSoup. The Falcon 9 launch table was extracted from the HTML content, parsed, and converted into a pandas DataFrame for further 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
- GitHub URL-  
<https://github.com/GBooks1/Course-Capstone/blob/main/jupyter-labs-webscraping.ipynb>

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

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

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

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

```
response.status_code
```

```
200
```

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

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

Using the dataframe `data` print the first 5 rows

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

	static_fire_date_utc	static_fire_date_unix	tbd	net	window	\
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	False	0.0	
1	None	NaN	False	False	0.0	
2	None	NaN	False	False	0.0	
3	2008-09-20T00:00:00.000Z	1.221869e+09	False	False	0.0	
4	None	NaN	False	False	0.0	

	rocket	success	\
0	5e9d0d95eda69955f709d1eb	False	
1	5e9d0d95eda69955f709d1eb	False	
2	5e9d0d95eda69955f709d1eb	False	
3	5e9d0d95eda69955f709d1eb	True	
4	5e9d0d95eda69955f709d1eb	True	



# Data Collection - Scraping

- Web scraping was applied using BeautifulSoup to extract Falcon 9 launch records from Wikipedia, with the launch table parsed and converted into a pandas DataFrame for analysis.
- GitHub  
URL: <https://github.com/GBooks1/Coursera-Capstone/blob/main/jupyter-labs-webscraping.ipynb>

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

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

```
# Use requests.get() method with the provided static_url  
r = requests.get(static_url)  
# Assign the response to a object  
data = r.text
```

Create a BeautifulSoup object from the HTML response

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

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

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

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

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about BeautifulSoup, please check the external towards the end of this lab

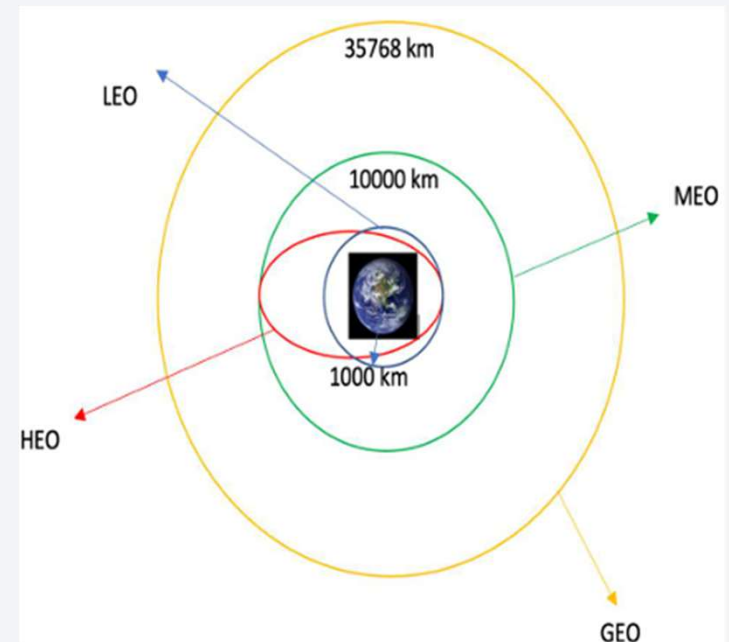
```
# Use the find_all function in the BeautifulSoup object, with element type 'table'.  
# Assign the result to a list called 'html_tables'  
html_tables = soup.find_all('table')
```

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

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

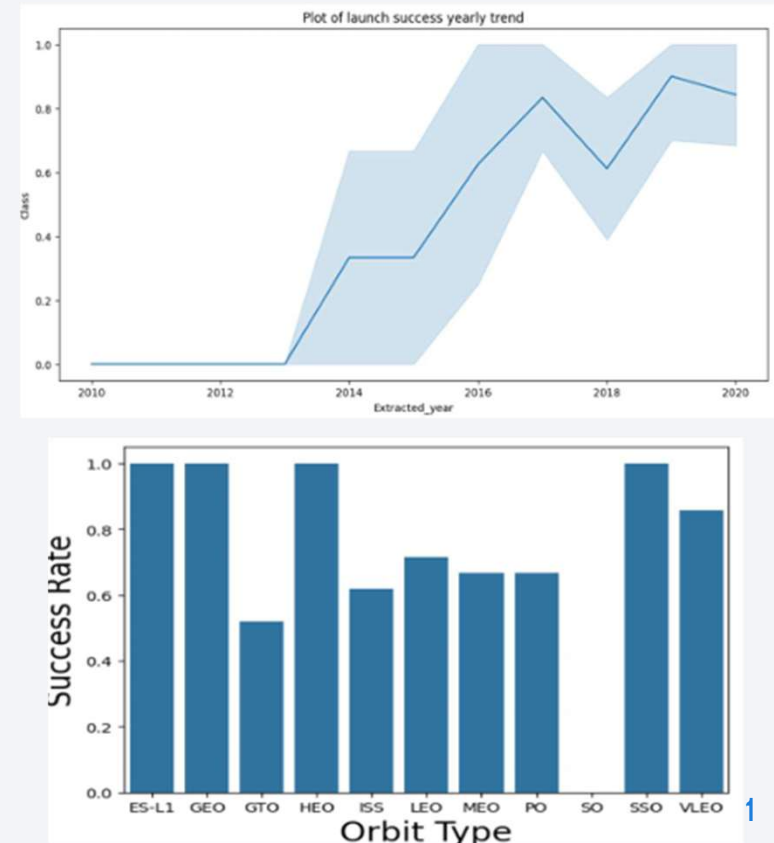
# Data Wrangling

- Data were collected using GET requests from the SpaceX API and web scraping from Wikipedia with BeautifulSoup. The raw data were then converted into pandas DataFrames, cleaned for missing values and inconsistencies, and merged into a single dataset for analysis.
- Data Collection (API + Web Scraping)
- Data Conversion (JSON + HTML DataFrames)
- Data Cleaning (Missing values, type correction)
- Data Integration (Merge, align keys)
- Feature Engineering (Derived variables)
- GitHubURL:  
[https://github.com/GBooks1/Coursera-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling%20\(1\).ipynb](https://github.com/GBooks1/Coursera-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb)



# EDA with Data Visualization

- The data was explored through a series of visualizations, including: the relationship between flight number and launch site; the correlation between payload and launch site; the success rate by orbit type; the interaction between flight number and orbit type; and the yearly trend in launch success.
- GitHub  
URL:[https://github.com/GBooks1/Coursera-Capstone/blob/main/jupyter-labs-eda-sql-coursera\\_sqlite.ipynb](https://github.com/GBooks1/Coursera-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqlite.ipynb)



## EDA with SQL

---

- Loaded the SpaceX dataset into a PostgreSQL database directly from within the Jupyter Notebook environment.
- Performed exploratory data analysis (EDA) using SQL to derive insights from the dataset.
- Executed queries to identify:
  - Unique launch site names involved in the space missions.
  - Total payload mass transported by boosters launched under NASA (CRS) missions.
  - Average payload mass carried by the F9 v1.1 booster version.
  - Total number of missions categorized as successful or failed.
  - Failed drone ship landing outcomes, including their associated booster versions and launch site names. •
- GitHub URL: [https://github.com/GBooks1/Coursera-Capstone/blob/main/jupyter-labs-eda-sql-coursera\\_sqlite.ipynb](https://github.com/GBooks1/Coursera-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqlite.ipynb)

# Build an Interactive Map with Folium

---

## Map Objects Added to Folium

- Launch Site Marker Pinpointed the exact coordinates of the launch site for spatial reference
- Proximity Markers Identified nearby features such as cities, highways, railways, and coastlines
- Distance Lines (PolyLine) Connected the launch site to each nearby feature to visualize direct distances
- Optional Circles (Buffer Zones) Represented radius-based proximity to assess spatial influence or safety zones

## Purpose of Map Elements

- Visualize spatial relationships between launch sites and surrounding infrastructure
- Assess proximity to key features for logistical and safety considerations

**GitHub URL:** [https://github.com/GBooks1/Coursera-Capstone/blob/main/lab\\_jupyter\\_launch\\_site\\_location.ipynb](https://github.com/GBooks1/Coursera-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb)

# Predictive Analysis (Classification)

---

- Data Preparation
  - Extracted target (Class), dropped from features, standardized features with StandardScaler
- Train/Test Split
  - 80/20 split, random\_state=2, 18 samples in test set
- Model Selection & Hyperparameter Tuning
- Model Evaluation
  - KNN performed the best with a score of 0.8482142857142858
  - All models showed low false negatives according to the confusion matrixes
  - GitHub [URL:https://github.com/GBooks1/Coursera-Capstone/blob/main/SpaceX\\_Machine%20Learning%20Prediction\\_Part\\_5%20\(1\).ipynb](https://github.com/GBooks1/Coursera-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb)

# Results

---

Exploratory data analysis result.

- The larger the flight amount at a launch site, the greater the success rate at a launch site

Interactive analytics demo in screenshots

- Visualize spatial relationships between launch sites and surrounding infrastructure

Predictive analysis results

- The k nearest neighbors classifier is the best machine learning algorithm for this task



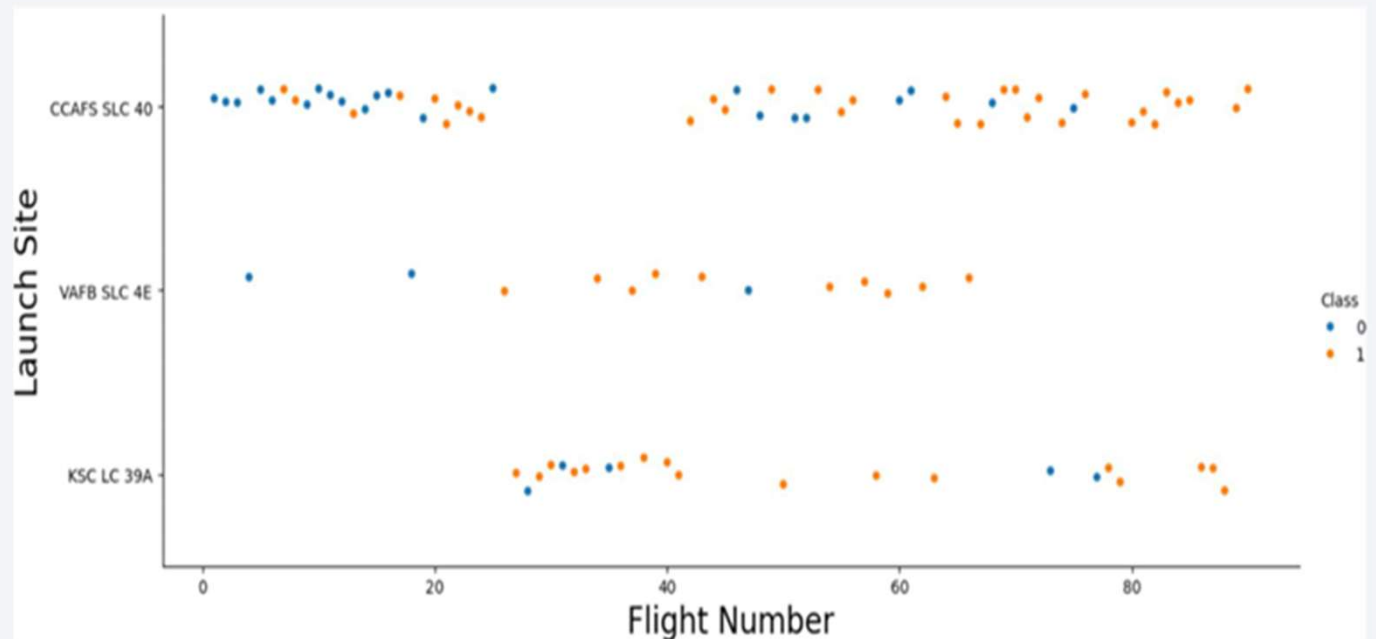
The background of the slide is a dynamic, abstract composition of numerous thin, overlapping lines and streaks. These lines are primarily in shades of blue and red, with some green and purple accents, creating a sense of motion and depth. The lines are most concentrated on the right side of the image, where they appear to radiate or flow towards the left, leaving a more solid blue area on the far left where the text is located.

Section 2

# Insights drawn from EDA

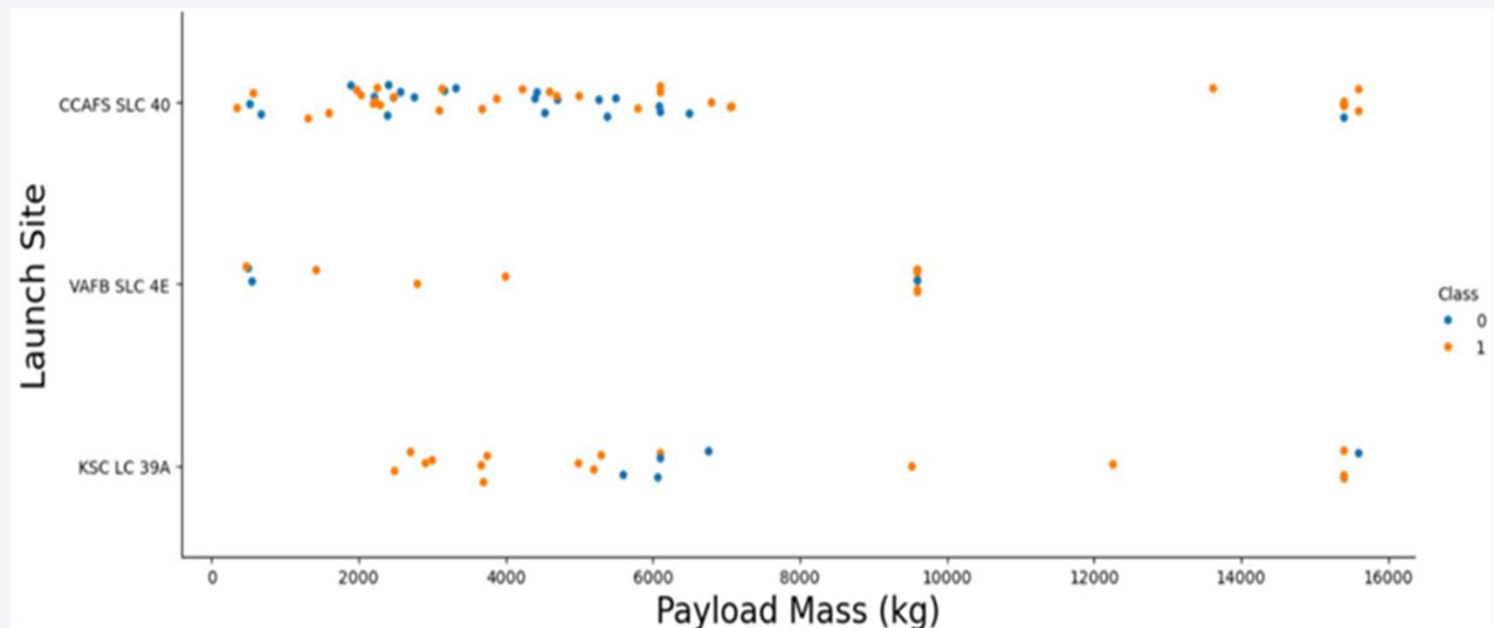
# Flight Number vs. Launch Site

The scatter plot shows that for each launch site, a higher number of flights is associated with a higher success rate, indicating that increased launch experience at a site contributes to improved reliability



# Payload vs. Launch Site

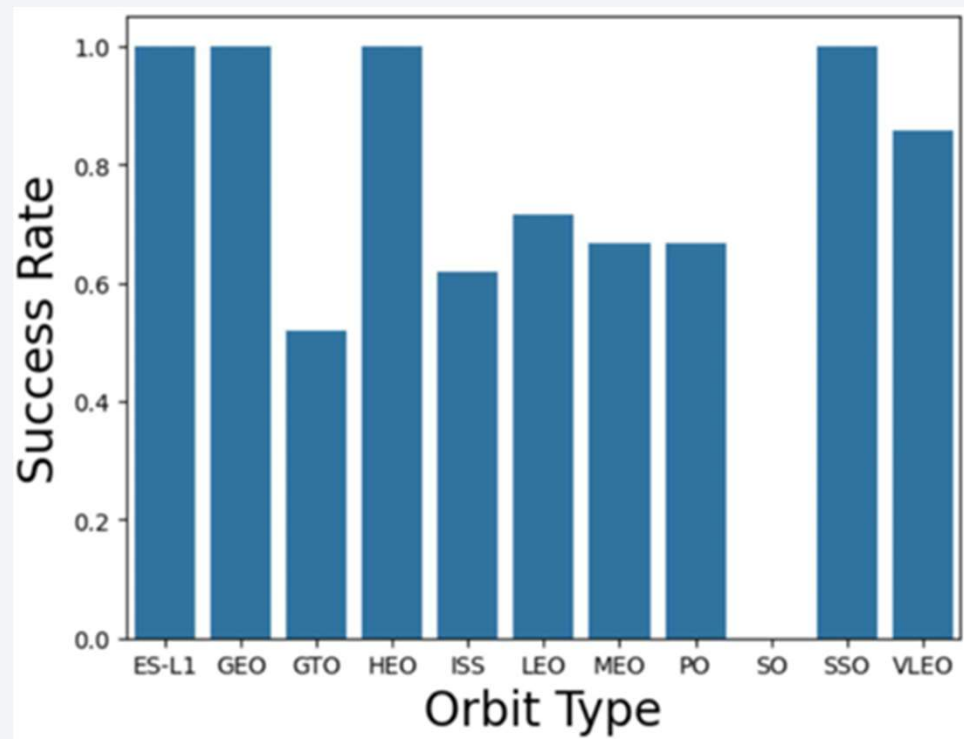
For the VAFB-SLC launchsite there are no rockets launched for heavy payload mass (greater than 10000).



# Success Rate vs. Orbit Type

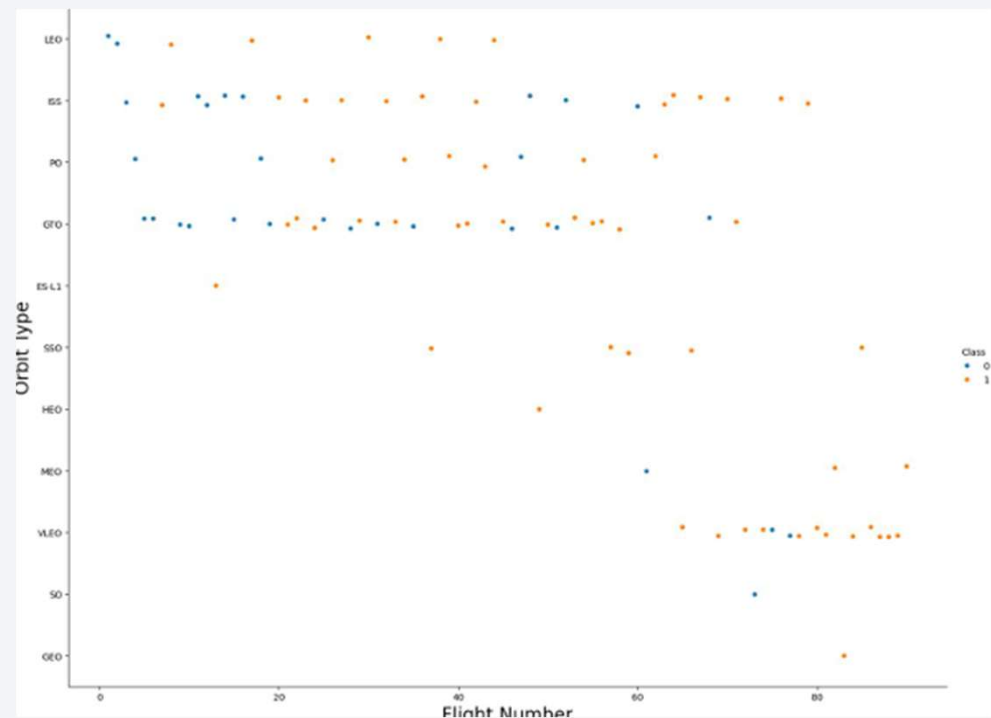
---

- The ES-L1, GEO, HEO, and SSO have the highest success rate.
- The SO has the lowest success rate.



# Flight Number vs. Orbit Type

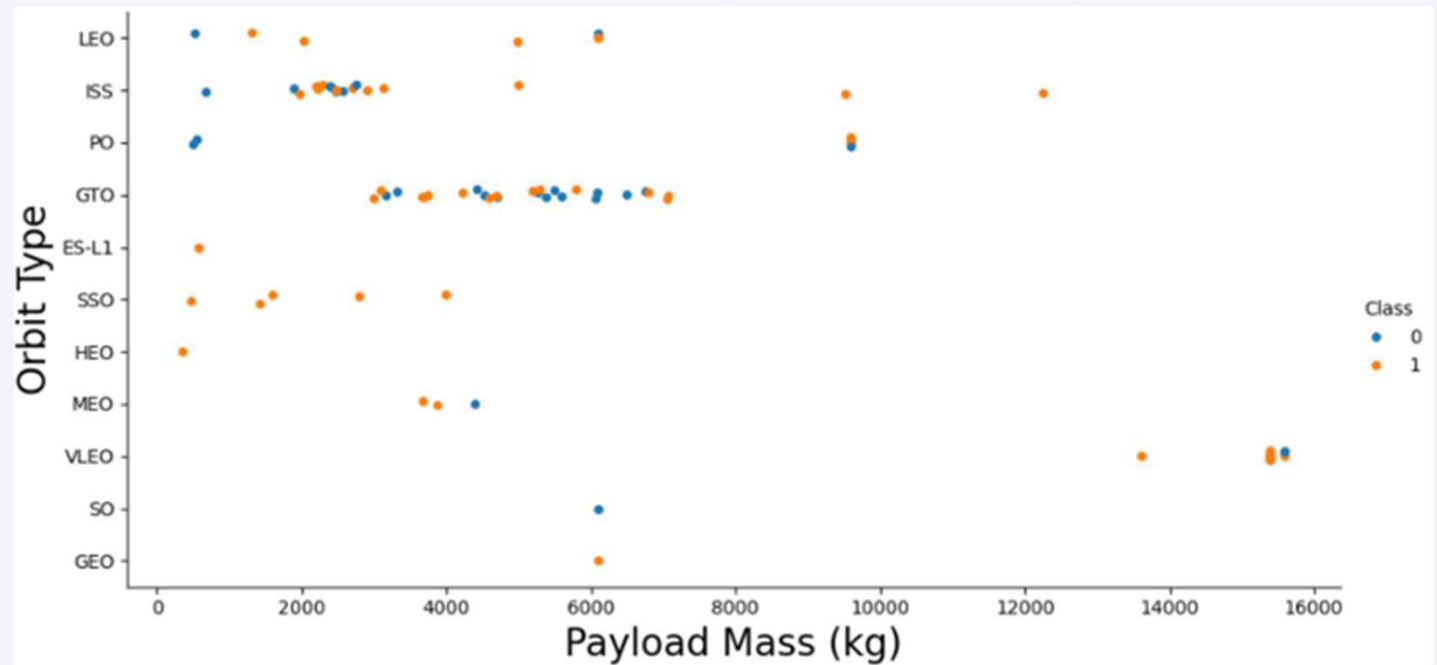
- In the LEO orbit, success seems to be related to the number of flights.
- Conversely, in the GTO orbit, there appears to be no relationship between flight number and success.



# Payload vs. Orbit Type

- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

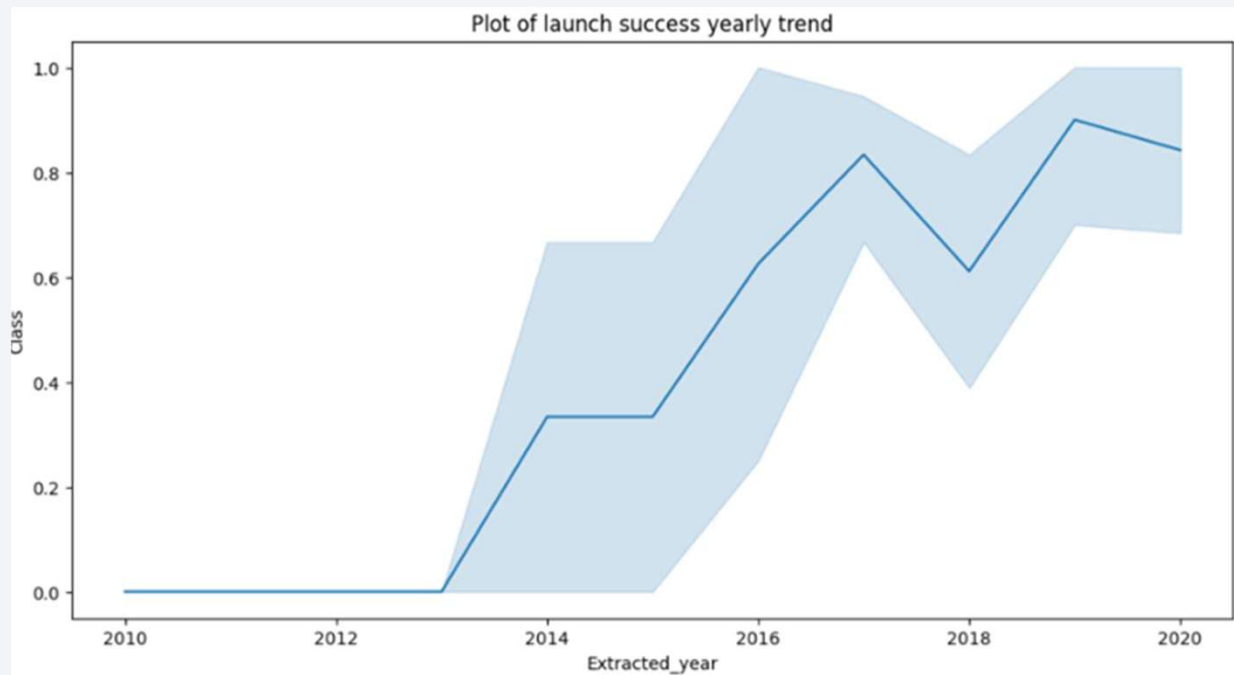
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present.





# Launch Success Yearly Trend

---



- The success rate since 2013 keeps increasing through 2020



# All Launch Site Names

---

- Find the names of the unique launch sites
- Present your query result with a short explanation here
- Used the key word DISTINCT to show only unique launch sites from the SpaceX data.

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

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with 'CCA'
- Present your query result with a short explanation here

```
SELECT *  
FROM SpaceX  
WHERE LaunchSite LIKE 'CCA%'  
LIMIT 5
```

index	Date	Time_(UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome	
0	0	None	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	1	None	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2	None	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	3	None	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	4	None	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

# Total Payload Mass

---

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

```
SELECT  
SUM(PayloadMassKG)  
AS Total_PayloadMass  
FROM SpaceX  
WHERE Customer LIKE  
'NASA (CRS)'
```

sum(PAYLOAD_MASS_KG_)	
0	45596

## Average Payload Mass by F9 v1.1

---

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

```
SELECT  
AVG(PayloadMassKG) AS  
Avg_PayloadMass FROM  
SpaceX WHERE  
BoosterVersion = 'F9 v1.1'
```

avg(PAYLOAD_MASS_KG_)	
0	2928.4

# First Successful Ground Landing Date

---

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

```
SELECT MIN(Date) AS  
FirstSuccessfull_landing_date FROM  
SpaceX WHERE LandingOutcome LIKE  
'Success (ground pad)'
```

firstsuccessfull_landing_date	
0	2015-12-22

## 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
- Present your query result with a short explanation here

```
SELECT BoosterVersion
FROM SpaceX
WHERE LandingOutcome = 'Success (drone ship)'
AND PayloadMassKG > 4000
AND PayloadMassKG < 6000 S
```

Booster_Version	
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

## Total Number of Successful and Failure Mission Outcomes

---

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

```
SELECT COUNT(MissionOutcome) AS  
SuccessOutcome FROM SpaceX WHERE  
MissionOutcome LIKE 'Success%'  
SELECT COUNT(MissionOutcome) AS  
FailureOutcome FROM SpaceX WHERE  
MissionOutcome LIKE 'Failure%'
```

	Mission_Outcome	count(*)
0	Failure	1
1	Success	100



## Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

```
SELECT BoosterVersion, PayloadMassKG FROM  
SpaceX WHERE PayloadMassKG = ( SELECT  
MAX(PayloadMassKG) FROM SpaceX ) ORDER BY  
BoosterVersion
```

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

## 2015 Launch Records

---

- List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Present your query result with a short explanation here

```
SELECT BoosterVersion, LaunchSite,  
LandingOutcome FROM SpaceX WHERE  
LandingOutcome LIKE 'Failure (drone ship)' AND  
Date BETWEEN '2015-01-01' AND '2015-12-31'
```

	boosterversion	launchsite	landingoutcome
0	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
1	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

---

- Present your query result with a short explanation here

```
SELECT LandingOutcome,  
COUNT(LandingOutcome)  
FROM SpaceX  
WHERE DATE BETWEEN '2010-06-04' AND  
'2017-03-20'  
GROUP BY LandingOutcome  
ORDER BY COUNT(LandingOutcome) DESC
```

	landingoutcome	count
0	No attempt	10
1	Success (drone ship)	6
2	Failure (drone ship)	5
3	Success (ground pad)	5
4	Controlled (ocean)	3
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1
7	Failure (parachute)	1

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a solid blue rectangle on the left and a satellite photograph of Earth on the right. The Earth is shown from a high altitude, with the horizon line curving across the middle. The night side of the Earth is visible, with numerous bright yellow and orange lights from cities and towns scattered across the dark landmasses. The atmosphere is visible as a thin blue layer along the horizon.

Section 3

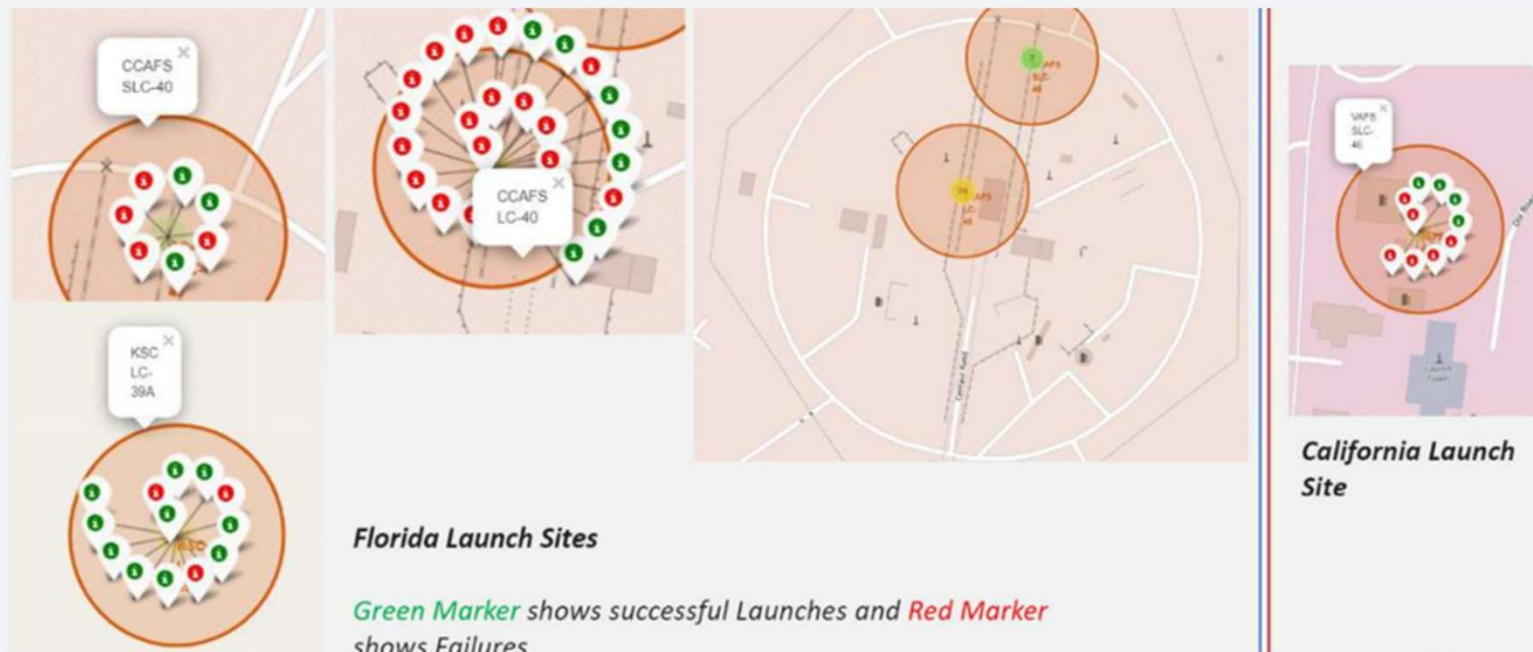
# Launch Sites Proximities Analysis

# All launch sites global map markers

---

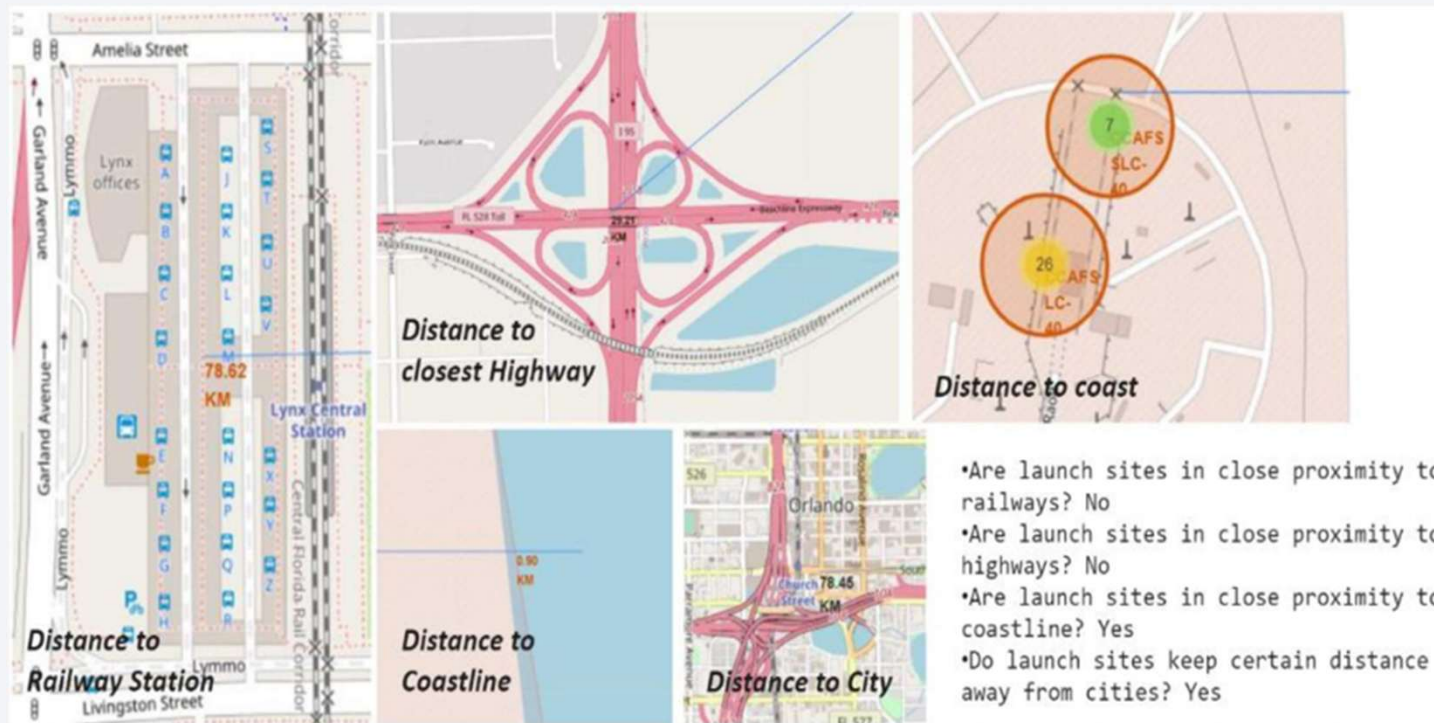


## Markers showing launch sites with color labels





# Launch Site distance to landmarks





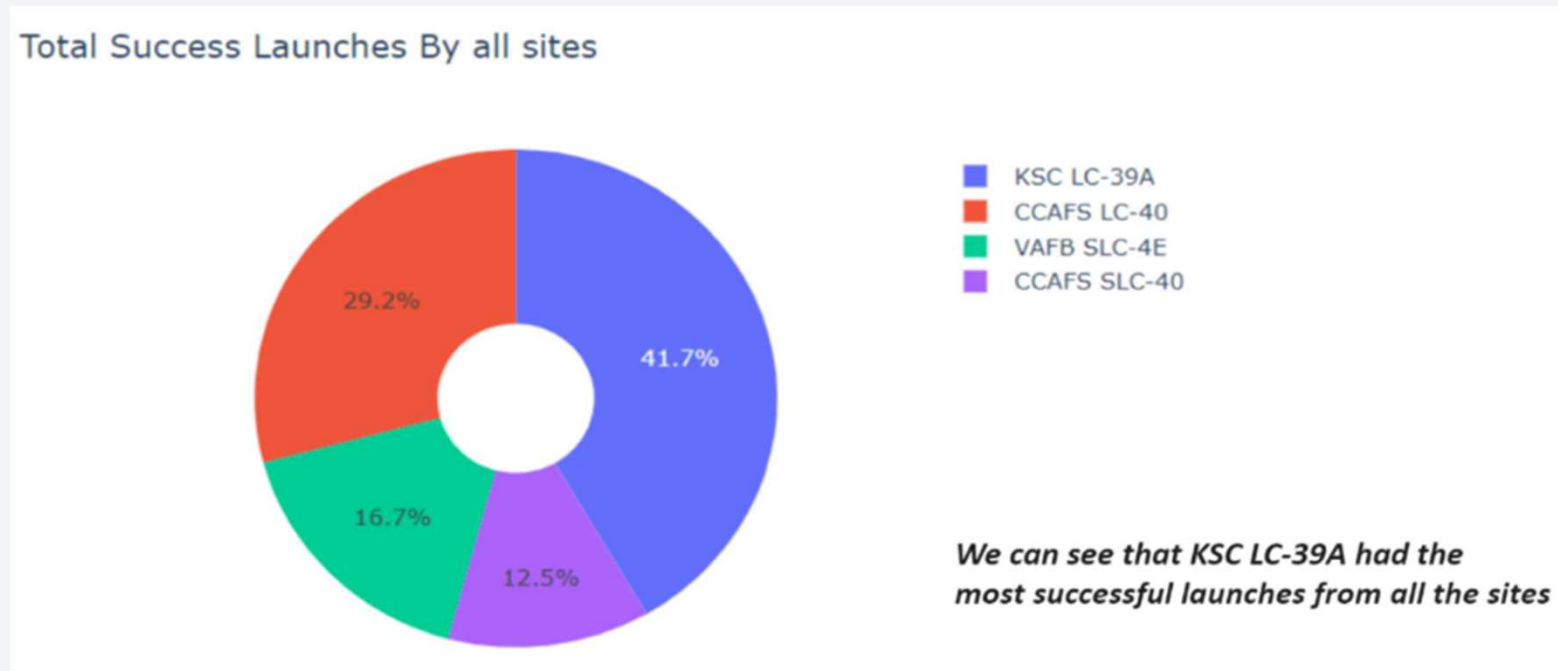


Section 4

# Build a Dashboard with Plotly Dash

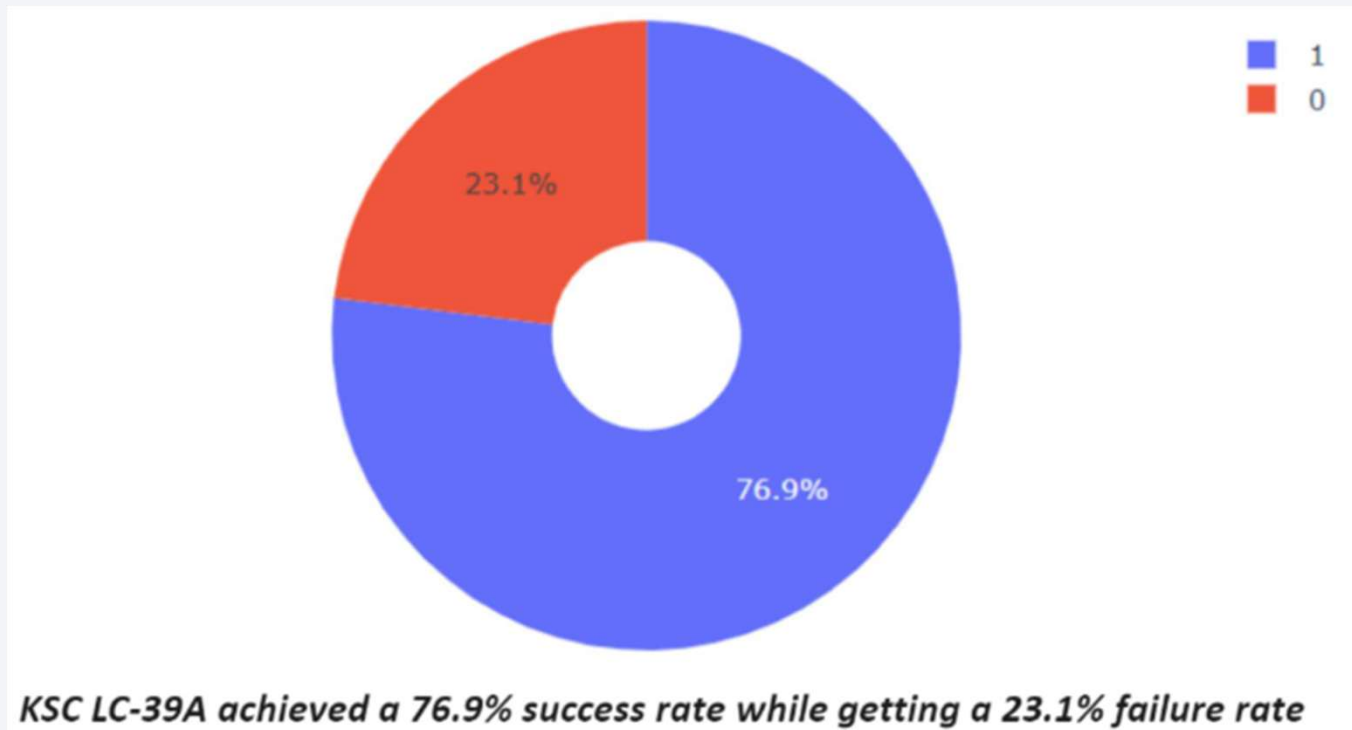
## Pie chart showing the success percentage achieved by each launch site

---



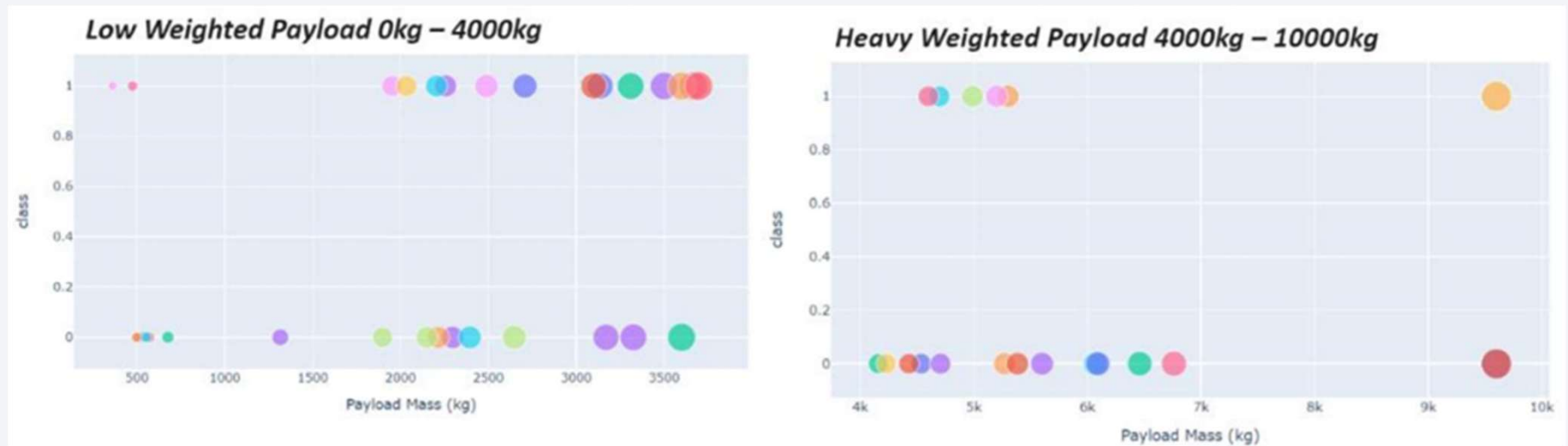
## Pie chart showing the Launch site with the highest launch success ratio

---



## Scatter plot of Payload vs Launch Outcome for all sites

---



The success rate for light weighted payloads is higher than the success rate for heavy weighted payloads.

The background of the slide is a composite image. The left side is a solid blue field. The right side features a perspective view of a tunnel with curved, ribbed walls. A bright light source at the end of the tunnel creates a strong lens flare and illuminates the interior. The overall effect is one of depth and forward motion.

Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

---

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy
- Best model is KNN with a score of 0.8482142857142858
- Best params is : {'algorithm': 'auto', 'n\_neighbors': 10, 'p': 1}

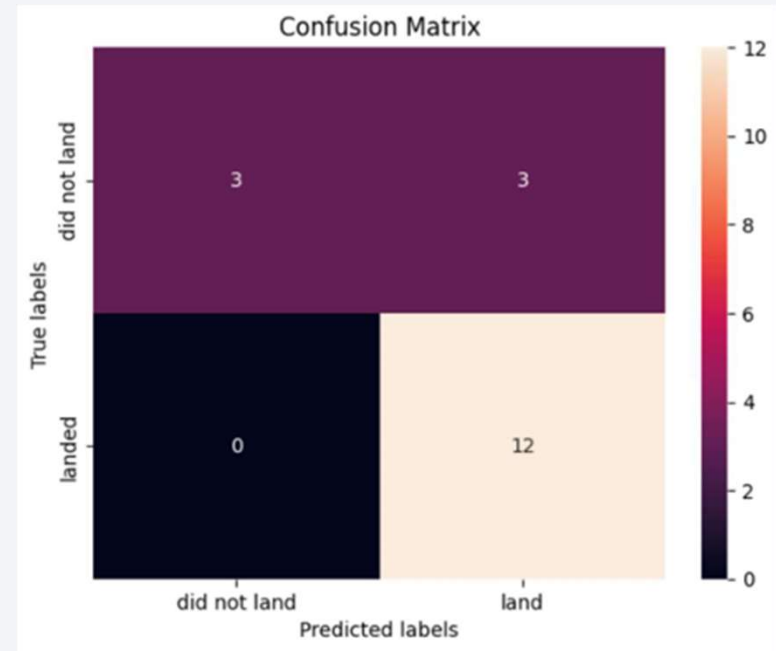
Model	Validation Data Accuracy	Test Accuracy
Logistic Regression	0.846	0.833
SVM	0.848	0.833
Decision Tree	0.836	0.778
KNN	0.848	0.833

# Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

The classifier can distinguish between the different classes.

- The major problem is the false positives e.g., unsuccessful landing marked as successful landing by the classifier.



# Conclusions

---

- 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 k nearest neighbors classifier is the best machine learning algorithm for this task.



# Appendix

---

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!

