

# Winning Space Race with Data Science

RAPETI.DHEEKSHIT  
19 JULY 2024



# Outline

---

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

# Executive Summary

---

- Summary of Methodologies:
  - Data collection using SpaceX API and web scraping from Wikipedia.
  - Data wrangling to handle missing values and categorize data.
  - Exploratory data analysis (EDA) using visualizations and SQL queries.
  - Interactive visual analytics using Folium and Plotly Dash.
  - Predictive analysis using classification models.
- Summary of All Results:
  - Identification of key factors influencing the success of Falcon 9 landings.
  - Visualization of trends and patterns in the data.
  - Development and evaluation of predictive models to estimate landing success.

# Introduction

---

- Project background and context:
  - Aim: Predict the success of Falcon 9 first stage landings to estimate launch costs.
  - Importance: SpaceX's reusability of the first stage reduces launch costs significantly compared to competitors.
- Problems you want to find answers:
  - Can we accurately predict the success of the Falcon 9 first stage landing?
  - How can this prediction impact cost estimation and competitive bidding?



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Collected data from SpaceX API: [API URL](#).
  - Web scraping from Wikipedia: [Wikipedia URL](#).
- Perform data wrangling
  - Checked for missing values using `isnull().sum()`.
  - Handled missing values in Payload Mass with the mean value.
  - Categorized data into numerical and categorical columns.
  - Used `value_counts()` to analyze launch sites, orbits, and landing outcomes.

# Methodology

---

## Executive Summary

- Perform exploratory data analysis (EDA) using visualization and SQL.
- Perform interactive visual analytics using Folium and Plotly Dash.

# Data Collection

---

In this project, we collected data from two primary sources: SpaceX API and web scraping from Wikipedia. The following steps outline the data collection process:

## 1. Collecting Data from SpaceX API:

- Step 1: Access the SpaceX API endpoint at <https://api.spacexdata.com/v4/launches/past>.
- Step 2: Send a GET request to the endpoint to retrieve JSON data of past Falcon 9 launches.
- Step 3: Parse the JSON response to extract relevant fields such as launch dates, rocket types, launch payloads, and landing outcomes.
- Step 4: Convert the extracted data into a Pandas DataFrame for further analysis.



# Data Collection

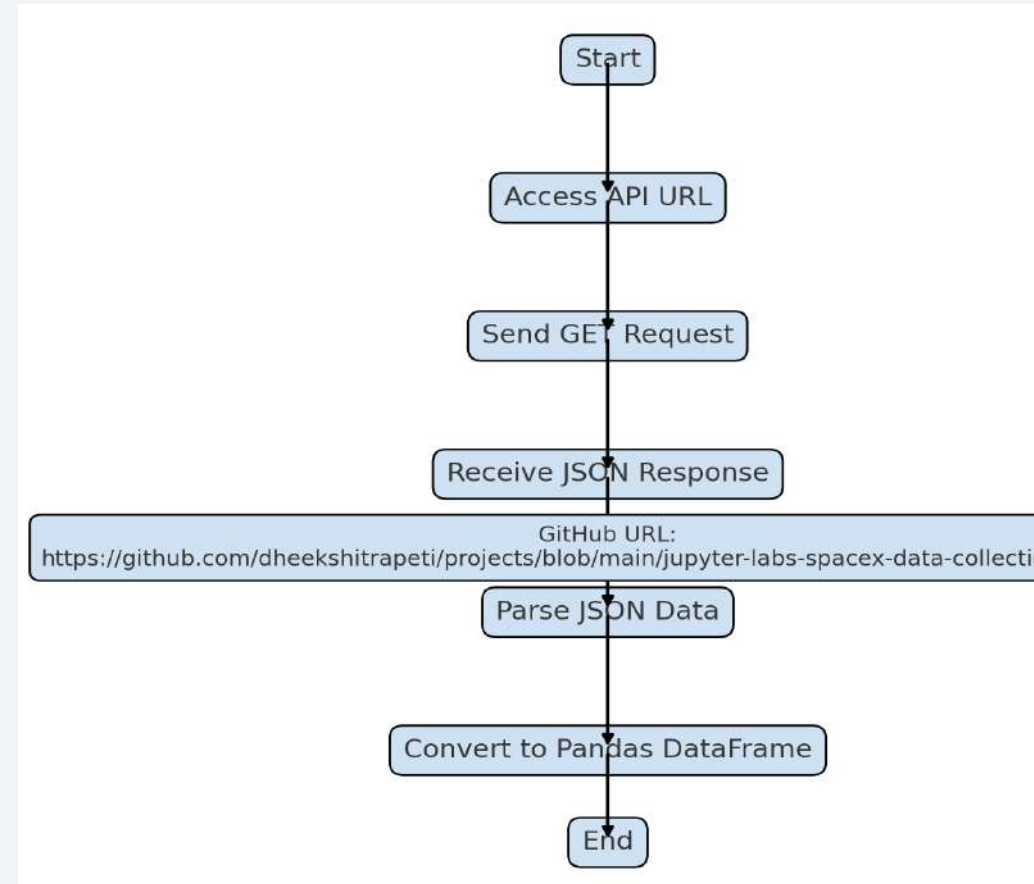
---

## 2. Collecting Data via Web Scraping from Wikipedia:

- Step 1: Navigate to the Wikipedia page:  
[https://en.wikipedia.org/wiki/List\\_of\\_Falcon\\_9\\_and\\_Falcon\\_Heavy\\_launches](https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches).
- Step 2: Identify the HTML table containing Falcon 9 launch records.
- Step 3: Use web scraping libraries (e.g., BeautifulSoup) to extract the HTML table.
- Step 4: Parse the HTML table to extract relevant data fields such as launch dates, launch sites, payload, and landing outcomes.
- Step 5: Convert the parsed table data into a Pandas DataFrame for further analysis.

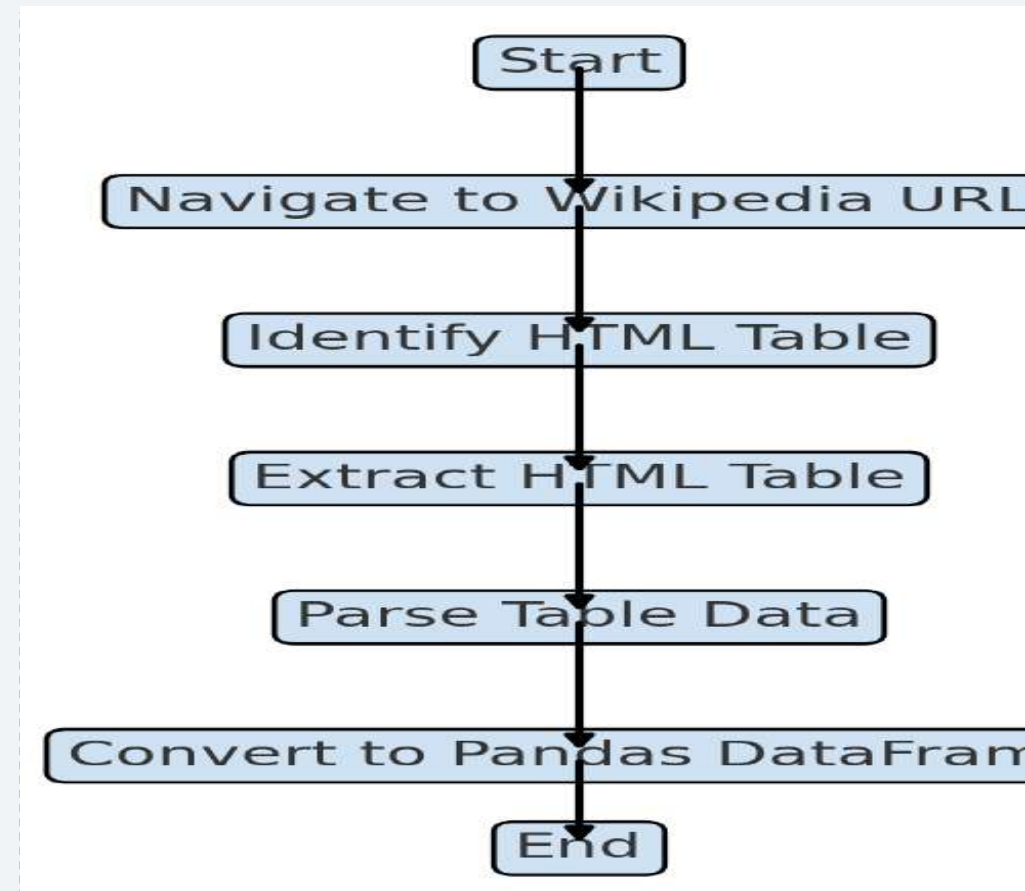
# Data Collection – SpaceX API

- Access API URL
- Send GET Request
- Receive JSON Response
- Parse JSON Data
- Convert to Pandas DataFrame
- **GitHub URL Link:**  
<https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-spacex-data-collection-api.ipynb>



# Data Collection - Scraping

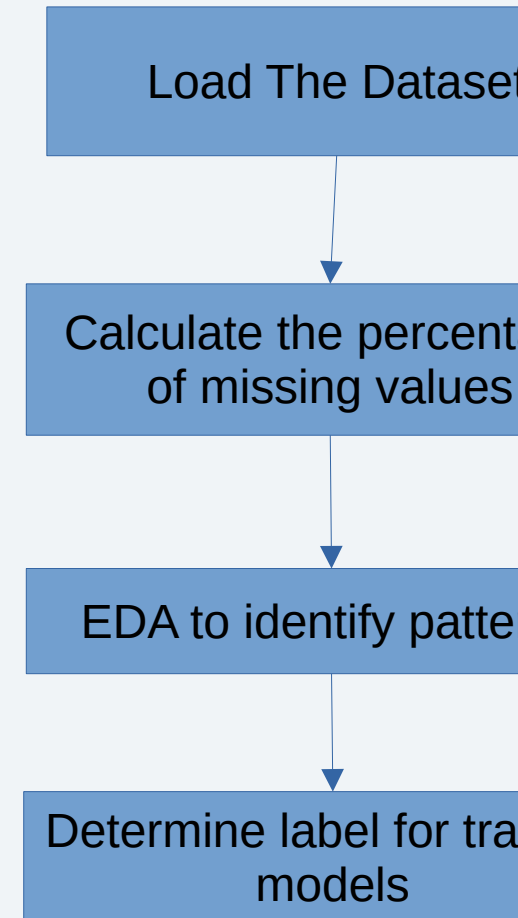
- Navigate to Wikipedia URL
- Identify HTML Table
- Extract HTML Table
- Parse Table Data
- Convert to Pandas DataFrame
- GitHub URL:  
<https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-webscraping.ipynb>



# Data Wrangling

---

- Load the dataset into the project file.
- Identify and calculate the percentage of missing values in each attribute. And handle them.
- Exploratory Data Analysis (EDA) to find some patterns in the data.
- Determine what would be the label for training supervised models.
- GitHub Reference Link:  
<https://github.com/dheekshitrapeti/projects/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb>



# EDA with Data Visualization

---

- Plotted scatter plot between the FlightNumber and PayloadMass and identified that as FlightNumber increases, the first stage is more likely to land successfully.
- Plotted scatter plot between FlightNumber and LaunchSite and identified that there are more launches from the CCAFS SLC 40 site compared to the other two sites.
- Plotted scatter plot between PayloadMass and LaunchSite and identified that the VAFB-SLC launches are no rockets launched for heavy payload mass (greater than 10000).
- Plotted a bar graph to show the success rate of each orbit type and came to the conclusion that the success rate of ES-L1, GEO and SSO are the same and equal to 1.0. And the SO orbit has no success.
- Plotted scatter plot between FlightNumber and Orbit type and identified that in the LEO orbit, the SuccessRate appears related to the number of flights.
- Finally visualized the launch success yearly trend and observed that the success rate since 2013 kept increasing till 2017.
- GitHub Reference link: <https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-eda-dataviz.ipynb>

# EDA with SQL

---

- CCAFS LC-40, VAFB SLC-4E, KSC LC-39A, CCAFS SLC-40 are the four launch sites in the space mission.
- The total payload mass carried by boosters launched by NASA (CRS) is 45596 kgs.
- Average payload mass carried by booster version F9 v1.1 is 2928.4 kgs.
- There are 13 dates in total where the succesful landing outcome in drone ship was acheived.
- NROL-76, Boeing X-37B OTV-5, Zuma are the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000.
- The total number of successful and failure mission outcomes are Failure (in flight)-1, Success-98, Success-1, Success (payload status unclear)-1.
- There are 11 booster\_versions which have carried the maximum payload mass.
- 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 are Failure (drone ship)-5, Success (ground pad)-3.
- GitHub Reference Link: [https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-eda-sql-edx\\_sqlite.ipynb](https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-eda-sql-edx_sqlite.ipynb)

# Build an Interactive Map with Folium

---

- Markers are used to locate a particular place in a map. Circles are used to show a certain region on the maps 100km etc and Lines are used to show connection between to places on the graph.
- Using Folium I plotted all the four launch sites in space mission using markers and labelled them with there respective names.
- Later on a add a circle to denote the success and failure rate of each launch site in space mission using differ colors and symbols.
- And finally I add lines connecting the launch sites with nearby coastline, railway station, and Highway along v the distance between them as a label.
- The launch success rate may depend on the location and proximities of a launch site, i.e., the initial position c rocket trajectories.
- Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.
- GitHub Reference Link:  
[https://github.com/dheekshitrapeti/projects/blob/main/lab\\_jupyter\\_launch\\_site\\_location.jupyterlite.ipynb](https://github.com/dheekshitrapeti/projects/blob/main/lab_jupyter_launch_site_location.jupyterlite.ipynb)

# Build a Dashboard with Plotly Dash

---

- Created a dropdown list to enable Launch Site selection. And the Default select value is all sites.
- Added a pie chart to show the total successful launches count for all sites. And If a specific launch site was selected, show the Success vs. Failed counts for the site.
- Added a slider to select payload range which varies from 0 to 10000.
- Added a scatter chart to show the correlation between payload and launch success as per our requirements.
- Finally I created a Dashboard using Plotly Dash to show the relationship between the payload and launch success of each individual launch site or for all four launch site.
- GitHub Reference Link: [https://github.com/dheekshitrapeti/projects/blob/main/spacex\\_dash\\_app.py](https://github.com/dheekshitrapeti/projects/blob/main/spacex_dash_app.py)



# Predictive Analysis (Classification)

---

- Loaded the dataset into the project file using pandas.
- Divided the dataset into four parts namely X\_\_train, X\_\_test, Y\_\_train, Y\_\_test where X is all the attributes in the dataset except class attribute and Y is the class attribute.
- Created a logistic regression object then create a GridSearchCV object logreg\_\_cv with cv = 10 to find the best parameters from dictionary parameters.
- Created a support vector machine object then create a GridSearchCV object svm\_\_cv with cv = 10.
- Created a decision tree classifier object then create a GridSearchCV object tree\_\_cv with cv = 10.
- Created a k nearest neighbors object then create a GridSearchCV object knn\_\_cv with cv = 10.
- Finally calculated the accuracy and score of each model.
- GitHub Reference Link:

[https://github.com/dheekshitrapeti/projects/blob/main/SpaceX\\_Machine\\_Learning\\_Prediction\\_Part\\_5.jupyterlite.ipynb](https://github.com/dheekshitrapeti/projects/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)

# Results

- For Logistic regression the best parameters from the dictionary parameters is "lbfgs". It has an accuracy : **0.7785714285714286** and score of 0.9444444444444444.
- For Support Vector Machines(SVM) the best parameter is "sigmoid". It has an accuracy of **0.7910714285714286** and score of 0.9444444444444444.
- For Decision Trees the best parameters 'criterion': 'gini', 'max\_depth': 2, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'splitter': 'best' with an accuracy : **0.8767857142857143**.
- For K nearest neighbours(KNN) the best parameters are 'algorithm': 'auto', 'n\_neighbors': 5, 'p': 1 with an accuracy : **0.8053571428571429**.
- Therefore Logistic Regression: Accuracy = 0.9444 Support Vector Machine (SVM): Accuracy = 0.9444 Decision Tree Classifier: Accuracy = 0.8889 K Nearest Neighbors (KNN): Accuracy = 0.9444 All three models (Logistic Regression, SVM, and KNN) achieved the same accuracy of 94.44% on the test set. However, if we consider the complexity and interpretability, Logistic Regression might be preferred due to its simplicity and ease of interpretation of coefficients. SVM and KNN, while achieving the same accuracy, might be considered if the decision boundary is non-linear or when the data has complex interactions that benefit from a more flexible model.

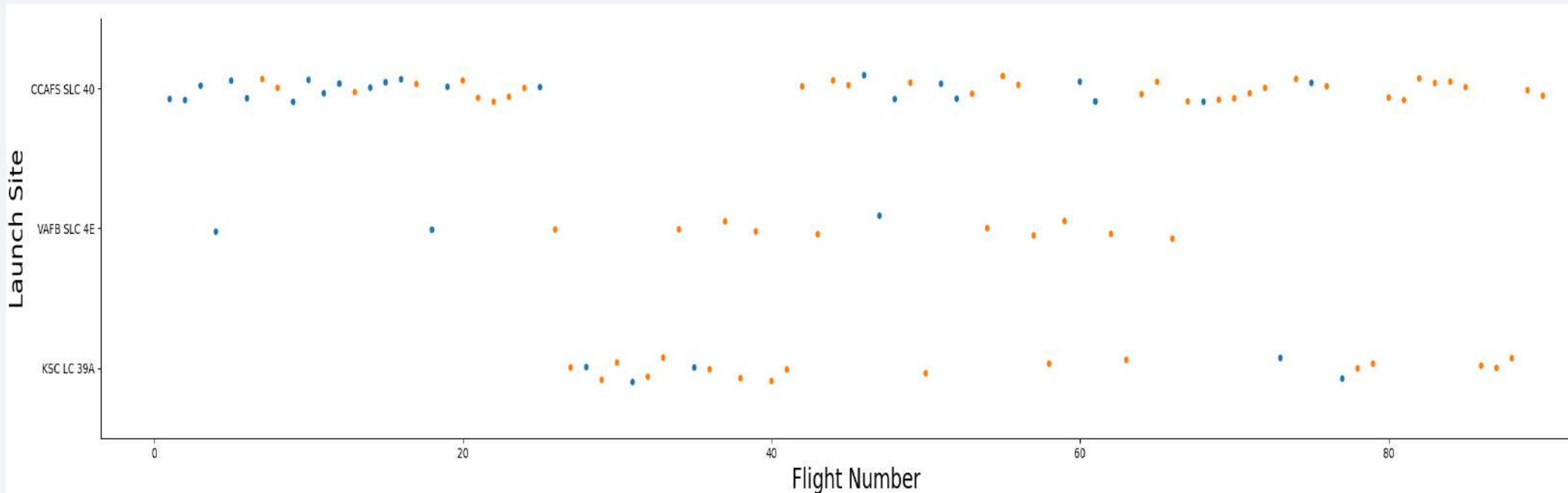
The background of the slide is a complex, abstract composition. It features a solid blue field on the left side, which transitions into a series of diagonal, overlapping bands of red and cyan. These bands are composed of fine, parallel lines that create a sense of depth and movement. A faint, grid-like pattern is visible across the entire image, particularly in the blue and cyan areas, suggesting a digital or data-driven theme.

Section 2

# Insights drawn from EDA



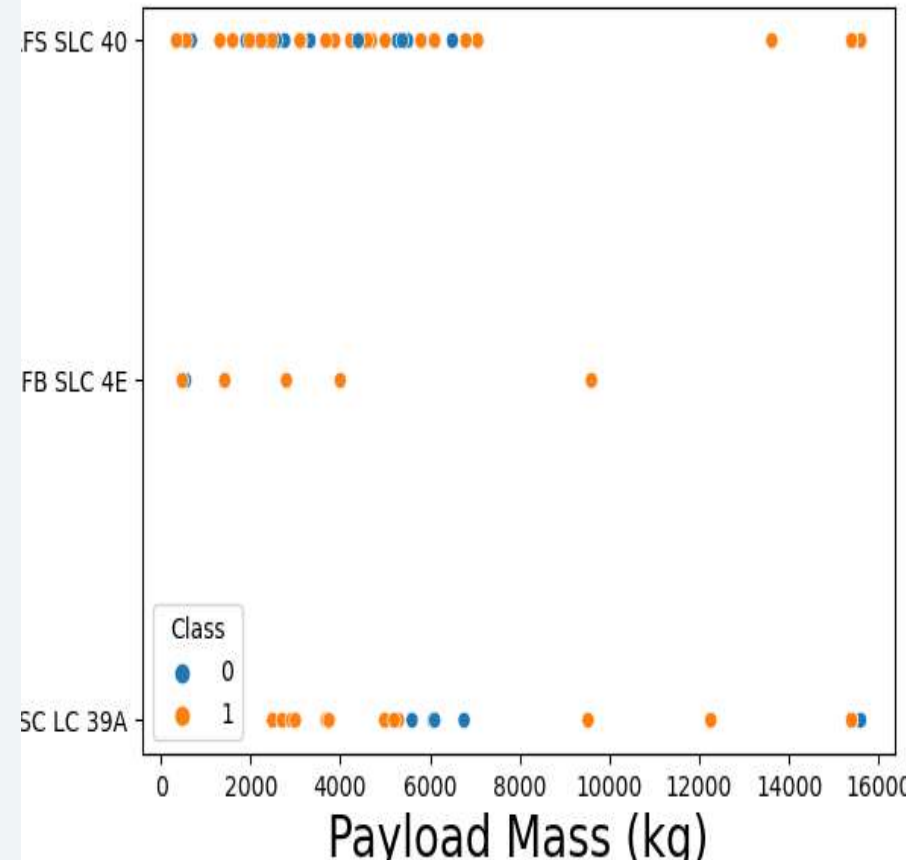
# Flight Number vs. Launch Site



- Used the function `catplot` to plot `FlightNumber` vs `LaunchSite`, set the parameter `x` parameter to `FlightNumber`, set the parameter `y` to `Launch Site` and set the parameter `hue` to `'class'`.
- We observed that there are more launches from the CCAFS SLC 40 site compared to the other two sites. The KSC LC 39A site has a mix of successful (Class 1) and unsuccessful (Class 0) launches spread across its flight numbers.
- VAFB SLC 4E has relatively fewer launches, with a noticeable mix of success and failure outcomes.

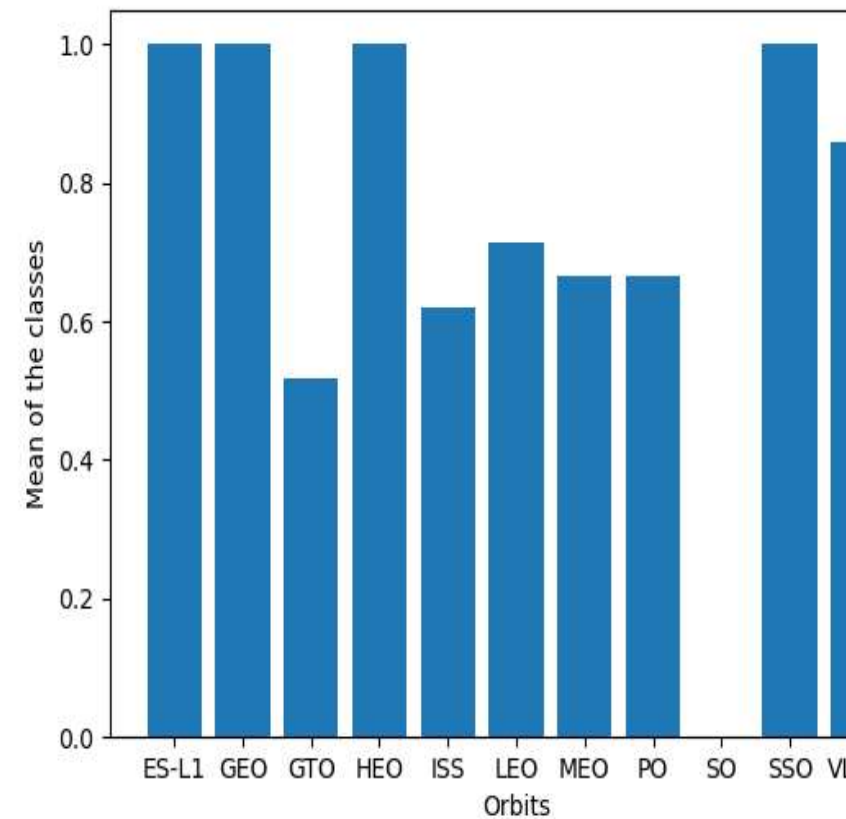
# Payload vs. Launch Site

- Used the function catplot to plot PayloadMass vs LaunchSite, set the parameter x parameter to PayloadMass ,set the y to Launch Site and set the parameter hue to 'class'.
- We observed Payload Vs. Launch Site scatter point chart and find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).



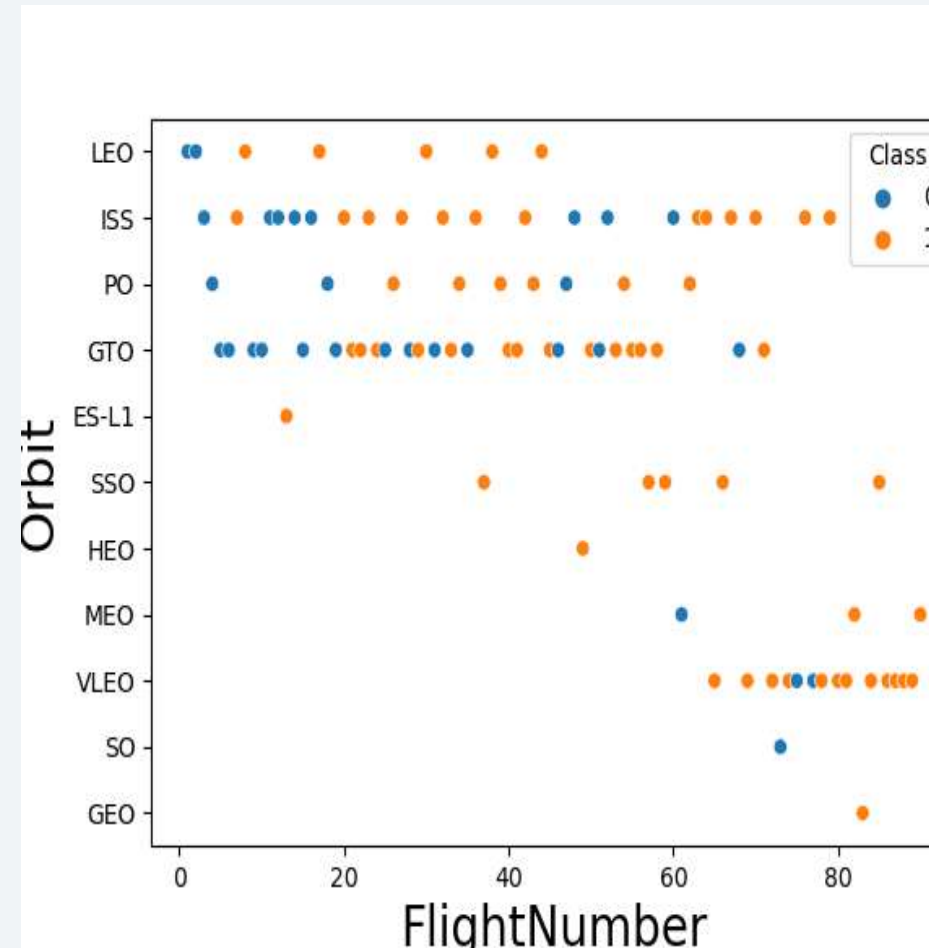
# Success Rate vs. Orbit Type

- We created a bar chart for the success rate of each orbit by calculating the mean of the 'Class' column.
- From the beside graph it is clear that the success rate of ES-L1, GEO and SSO are same and equal to 1.0. And the SO orbit has no success.



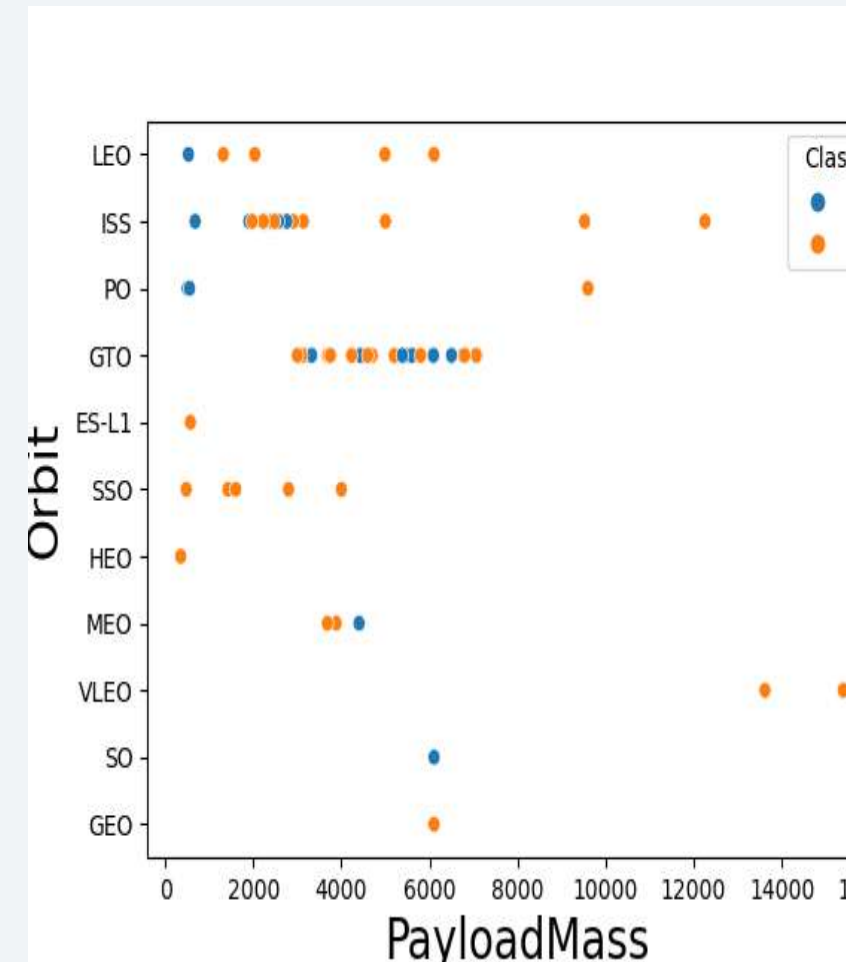
# Flight Number vs. Orbit Type

- Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value.
- We should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



# Payload vs. Orbit Type

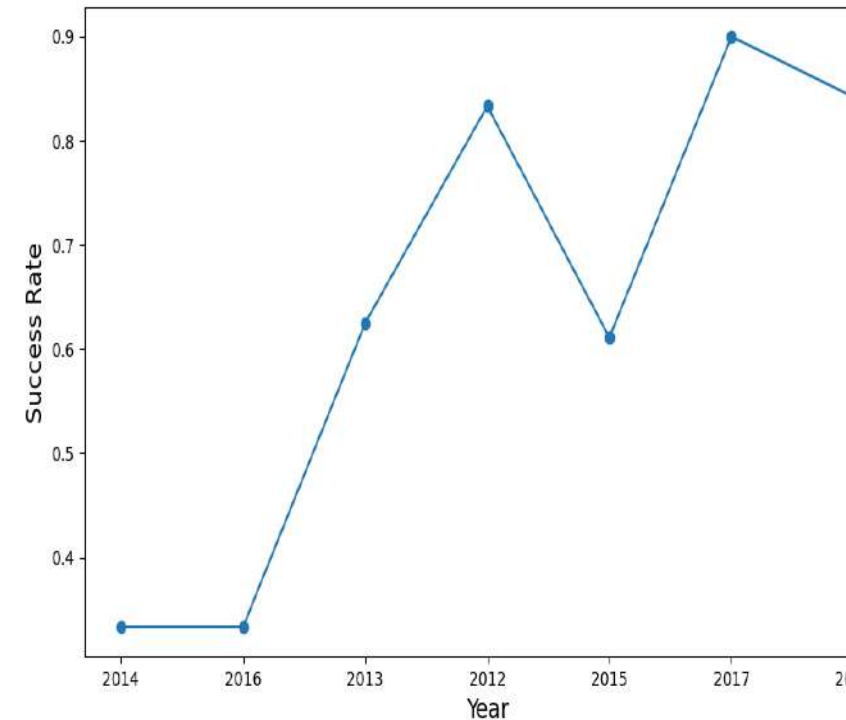
- Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value.
- We observed that with heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccesful mission) are both there here.





# Launch Success Yearly Trend

- Plotted a line chart with x axis to be Year and y axis to be average success rate, to get the average launch success trend.
- Calculate the success rate per year using the below formula:
$$\text{success\_rate\_per\_year} = \frac{\text{successful\_launches\_per\_year}}{\text{total\_launches\_per\_year}}$$
- We observed that the success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing.



# All Launch Site Names

---

**Query :** SELECT DISTINCT Launch\_Site FROM SPACEXTBL

**Result:** Launch\_Site

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

**Explanation:** We selected distinct launch sites from the SPACEXTBL using above mentioned query.

# Launch Site Names Begin with 'KSC'

---

**Query:** SELECT Launch\_\_Site from SPACEXTBL WHERE Launch\_\_Site LIKE '%KSC%' LIMIT 5

**Result:** Launch\_\_Site

0 KSC LC-39A

1 KSC LC-39A

2 KSC LC-39A

3 KSC LC-39A

4 KSC LC-39A

**Explanation:** Selected the launch sites that begins with the letters "KSC" using the above mentioned query.

# Total Payload Mass

---

**Query:** SELECT SUM(PAYLOAD\_\_MASS\_\_KG\_\_) from SPACEXTBL WHERE  
Customer== 'NASA (CRS)'

**Result:**SUM(PAYLOAD\_\_MASS\_\_KG\_\_)

0	45596
---	-------

**Explanation:**Used the sum aggregate function to calculate the sum of PayLoadMass using above query.

# Average Payload Mass by F9 v1.1

---

Query: SELECT AVG(PAYLOAD\_\_MASS\_\_KG\_\_) from SPACEXTBL WHERE  
Booster\_\_Version== 'F9 v1.1'

Result: AVG(PAYLOAD\_\_MASS\_\_KG\_\_)

0	2928.4
---	--------

Explanation: We calculated the average using the avg aggregate function.

# First Successful Ground Landing Date

---

**Query:** SELECT Date from SPACEXTBL WHERE Landing\_\_Outcome== 'Success (drone ship)'

**Result:** Date

0	2016-04-08	7	2017-06-25
1	2016-05-06	8	2017-08-24
2	2016-05-27	9	2017-10-09
3	2016-08-14	10	2017-10-11
4	2017-01-14	11	2017-10-30
5	2017-03-30	12	2018-04-18
6	2017-06-23	13	2018-05-11

**Explanation:** Selected the dates which has Landing Outcome as Success(drone Ship).

# Successful Drone Ship Landing with Payload between 4000 and 6000

---

**Query:** SELECT Payload from SPACEXTBL WHERE Landing\_Outcome== 'Success (ground pad)'  
AND PAYLOAD\_MASS\_\_\_KG\_ BETWEEN 4000 AND 6000

**Result:** Payload

0	NROL-76
1	Boeing X-37B OTV-5
2	Zuma

**Explanation:** Selected the Payload where the PayloadMass\_kg is in between 4000 and 6000.

# Total Number of Successful and Failure Mission Outcome

---

**Query:** SELECT Mission\_\_Outcome, COUNT(Mission\_\_Outcome) FROM SPACEXTBL GROUP BY Mission\_\_Outcome

**Result:** Mission\_\_Outcome COUNT(Mission\_\_Outcome)

0	Failure (in flight)	1
1	Success	98
2	Success	1
3	Success (payload status unclear)	1

**Explanation:** Used the count function to calculate the count of success and failure outcomes.



# Boosters Carried Maximum Payload

---

**Query:**SELECT Payload from SPACEXTBL WHERE PAYLOAD\_\_MASS\_\_KG\_\_ =(SELECT MAX(PAYLOAD\_\_MASS\_\_KG\_\_) FROM SPACEXTBL)

**Result:**Payload

- 0 Starlink 1 v1.0, SpaceX CRS-19
- 1 Starlink 2 v1.0, Crew Dragon in-flight abort t...
- 2 Starlink 3 v1.0, Starlink 4 v1.0
- 3 Starlink 4 v1.0, SpaceX CRS-20
- 4 Starlink 5 v1.0, Starlink 6 v1.0 etc.

**Explanation:**Selected the Payload which are carrying maximum payloadmass. In total we get 11 payloads satisfying above conditions.

# 2015 Launch Records

---

<b>Result:</b>	Month	Booster_Version	Launch_Site	Landing_Outcome
0	February	F9 FT B1031.1	KSC LC-39A	Success (ground pad)
1	May	F9 FT B1032.1	KSC LC-39A	Success (ground pad)
2	June	F9 FT B1035.1	KSC LC-39A	Success (ground pad)
3	August	F9 B4 B1039.1	KSC LC-39A	Success (ground pad)
4	September	F9 B4 B1040.1	KSC LC-39A	Success (ground pad)
5	December	F9 FT B1035.2	CCAFS SLC-40	Success (ground pad)

**Explanation:**Displayed the month names, succesful landing\_outcomes in ground pad ,booster versions, launch\_site for the months in year 2017

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

---

## Result:

	Landing__Outcome	Outcome__Count
0	Failure (drone ship)	5
1	Success (ground pad)	3

**Explanation:** Calculated 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 Reference Link:

[https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-eda-sql-edx\\_sqlite.ipynb](https://github.com/dheekshitrapeti/projects/blob/main/jupyter-labs-eda-sql-edx_sqlite.ipynb)

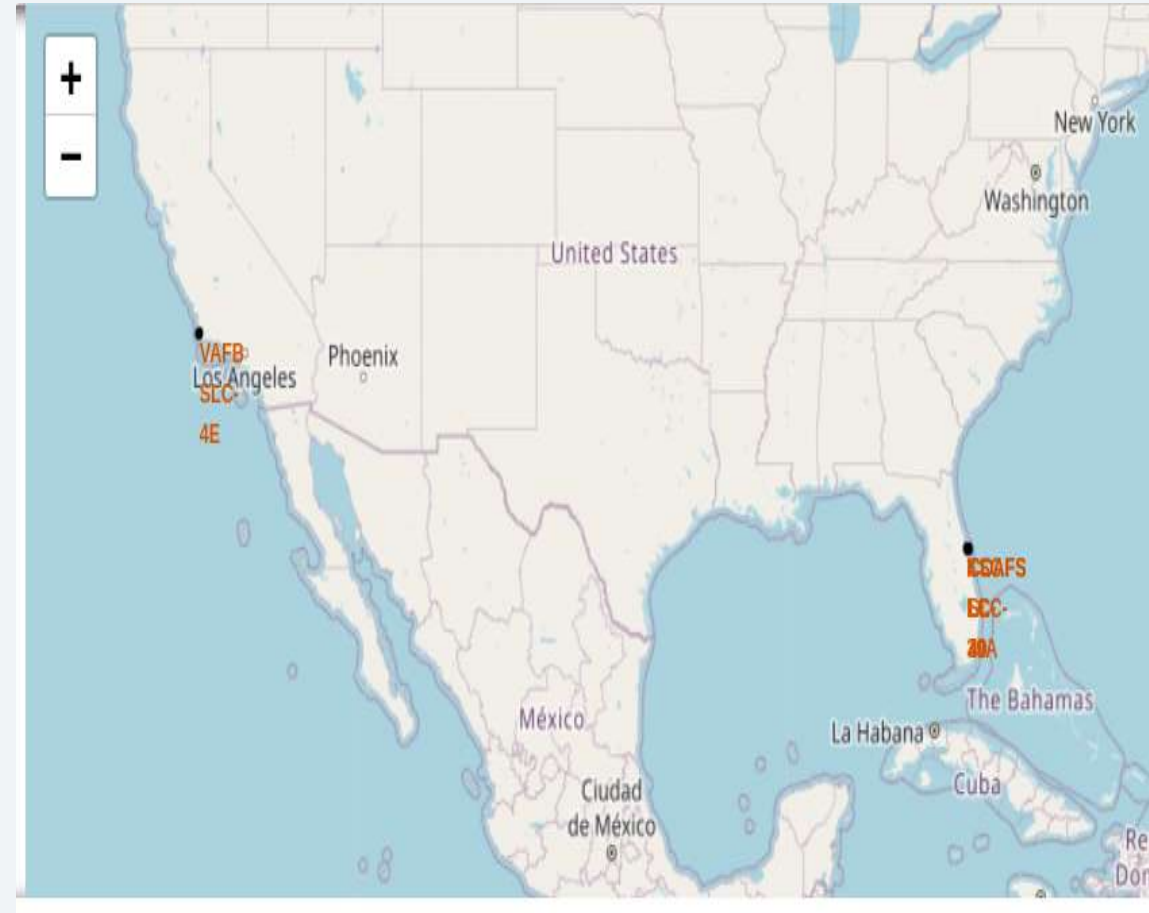
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The image is a composite of a dark blue sky and a view of the Earth's surface, which is covered in a dense network of city lights and clouds. The lights are concentrated in the lower right portion of the image, while the upper left shows a clear blue sky.

Section 3

# Launch Sites Proximities Analysis

# Location of launch sites on the world map

- Created a folium map and located the four launch sites on the map using the latitude and longitude of the launch sites
- .
- We also labeled the launch sites with their names using orange color.
- As we can see on the on beside image there are four launch sites are located and labeled with their respective names in orange colour.
- By doing so we can easily identify the launch sites and there merits and demerits.



# Mark the success/failed launches for launch sites

- Identified the no of success and failed launches of each launch site.
- Created MarkerCluster object to mark the no of success and failed launches of each launch site.
- From the image we can easily understand that that number which is in red color circle is failed launches where as green indicate successfull launches.
- We can get a clear clarity of the success and failed launches of each site.





# Distances between a launch site to its proximities

- Calculated the distance between the nearest proximities using distance between two points formula.
- Created a folium.PolyLine object to draw a line from the launch site to its nearest proximities and labeled the distance between them using yellow color
- We can identify which mode of transportation is useful for the map so that it happen in low cost.



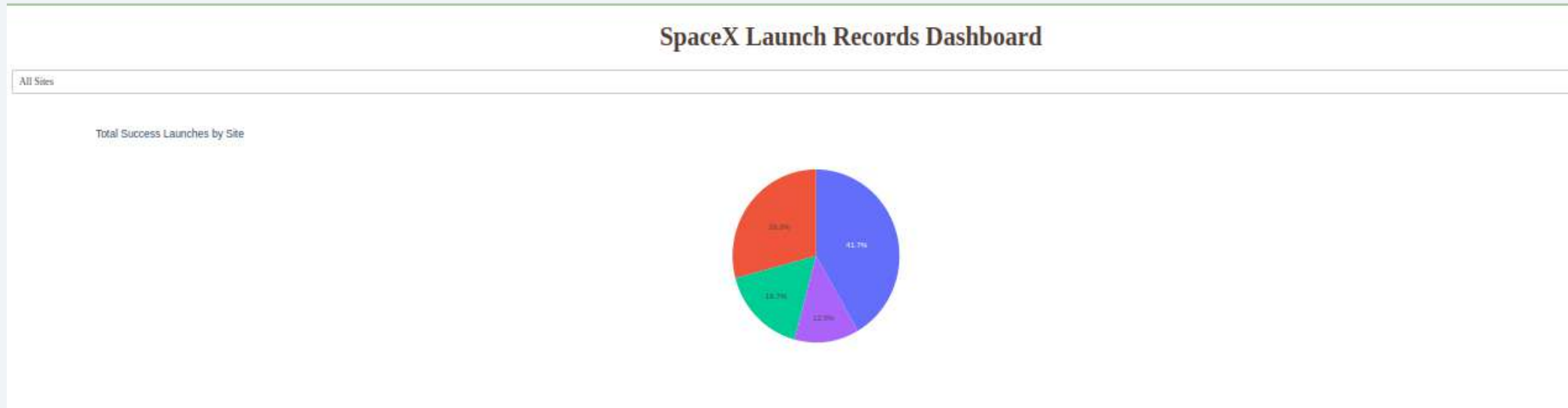


Section 4

# Build a Dashboard with Plotly Dash

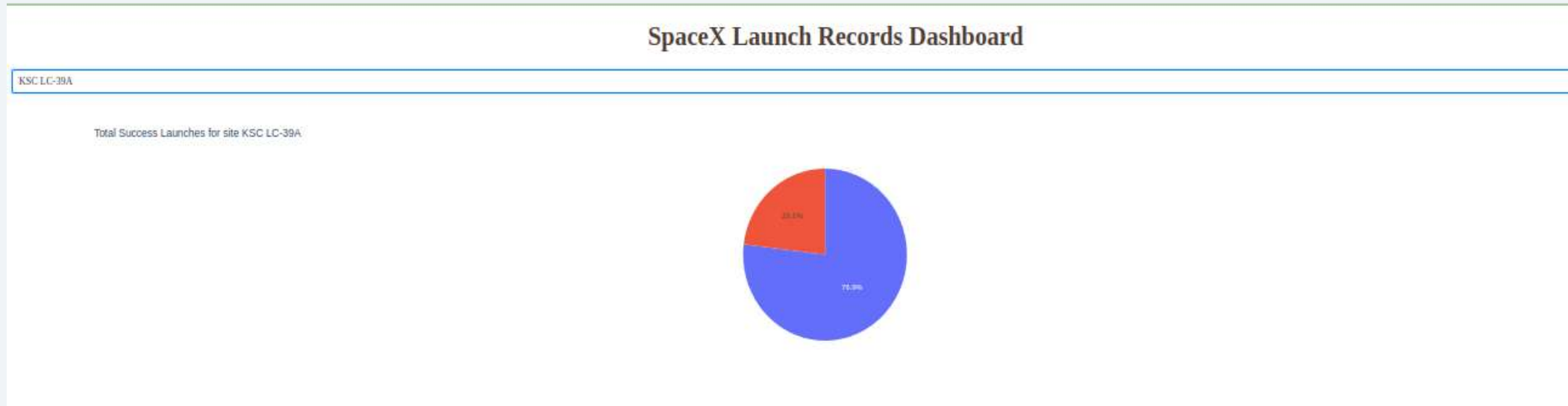


# Dashboard to show launch success count for all sites



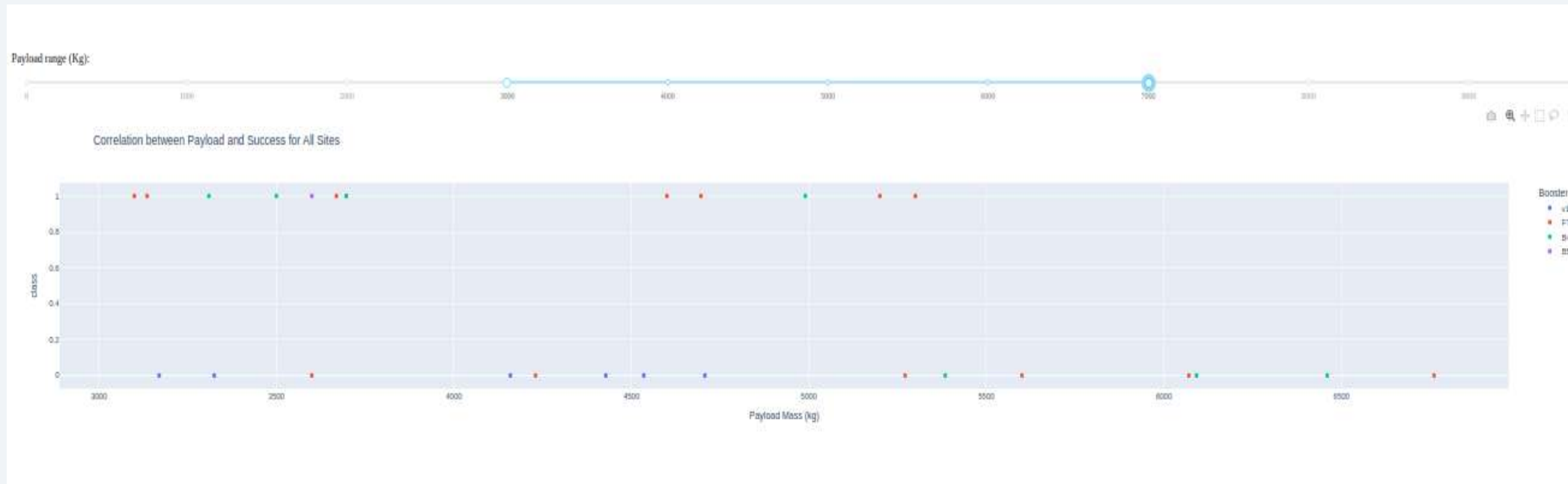
- From the above pie chart it is clear that the KSC LC-39A has highest success launches when compared to other launch sites.
- From the pie chart the KSC LC-39A holds 41.7% of overall share of success of all launch sites which is labeled with blue colour in the pie chart.
- In second place CCAFS LC-40 holds 29.2% in overall share of success of all launch sites which is in red colour.

# Dashboard to show the launch site with highest success rate



- From the above pie chart it was clear that the KSC LC-39A has 76.9% success ratio which is in blue color.
- From the pie chart we can also say that the failure ratio of KSC LC-39A is about 23.1% which is colored in red color.

# Dashboard to display Payload vs Launch Outcome



- From the above scatter plot we identify the relationship between the Payload and Launch Outcomes of all sites.
- We can also identify the relationship between Payload and success ratio of each site individual also.



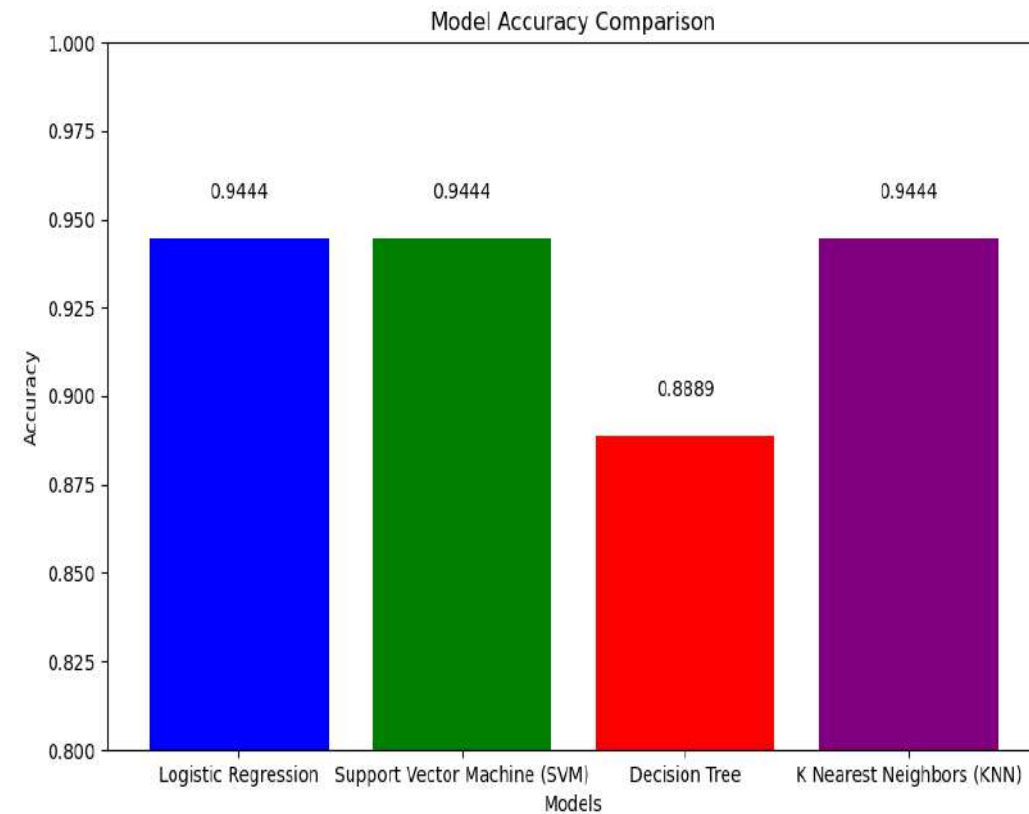
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

From the bar graph it is clear that except Decision Tree all other 3 algorithm have same accuracy i.e 0.9444.

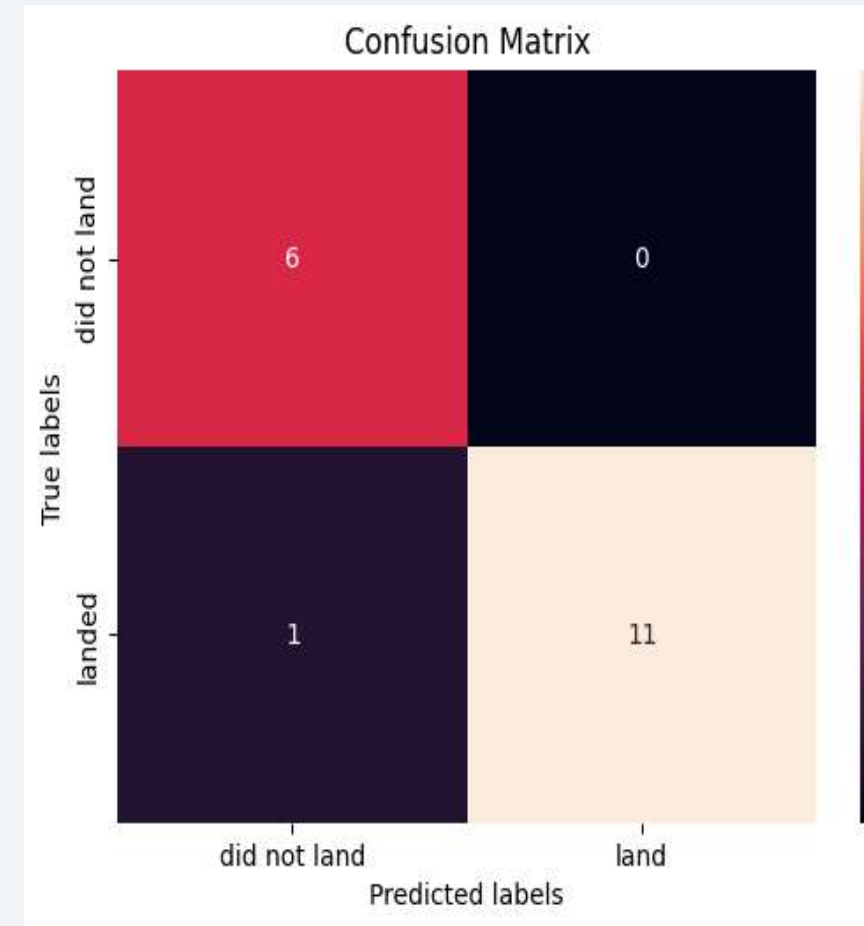
SO we adopt any of the three algorithms to Predict the success or failure of falcon9 first stage mission.



# Confusion Matrix

## Confusion Matrix Breakdown:

- True Positives (TP): The model correctly predicted the positive class (landed) 11 times.
- True Negatives (TN): The model correctly predicted the negative class (did not land) 6 times.
- False Positives (FP): The model incorrectly predicted the positive class (landed) 0 times.
- False Negatives (FN): The model incorrectly predicted the negative class (did not land) 1 time
- Therefore The confusion matrix shows a well-performing model with a strong ability to predict both the positive and negative classes accurately.



# Conclusions

---

## Summary:


- The model has high accuracy (94.4%), precision (100%), and specificity (100%).
- The recall is also quite high at 91.7%, indicating the model is good at identifying positive cases (landed).
- The F1 Score of 95.7% suggests a balanced performance between precision and recall.
- All three models (Logistic Regression, SVM, and KNN) achieved the same accuracy of 94.44% on the test set. However, if we consider the complexity and interpretability, Logistic Regression might be preferred due to its simplicity and ease of interpretation of coefficients. SVM and KNN, while achieving the same accuracy, might be considered if the decision boundary is non-linear or when the data has complex interactions that benefit from a more flexible model.
- GitHub Reference Link:  
[https://github.com/dheekshitrapeti/projects/blob/main/SpaceX\\_Machine\\_Learning\\_Prediction\\_Part\\_5.jupyterlite.ipynb](https://github.com/dheekshitrapeti/projects/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb)

# Appendix

---

You can go through all my work through the below mentioned link

GitHub Reference Link:<https://github.com/dheekshitrapeti/projects/tree/main>.

Thank you for going through the ppt and spending your valuable time on this content. Thank you once again from my hearts 



Thank you!

