



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

NAKIMERA KAREN  
SATURDAY-18-06-  
2022



# Outline

---

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

# Executive Summary

---

## Summary of methodologies:

The following methodologies were used for this analysis;

- Data collection(SPACEX API and webscraping)
- Data wrangling
- EDA with Data visualization
- EDA with SQL
- INTERACTIVE MAP OF LAUNCH SITES
- DASHBOARD WITH PLOTLY ANALYSIS
- PREDICTIVE ANALYSIS

# Introduction

---

- **Project background and context:**

The project requires the success rate of future Falcon 9 launches. In this way, the company can more accurately estimate the launch costs for each rocket. Information will be collected from the SpaceX API and using web scraping on Wikipedia articles, later, all the raw data obtained will be processed in order to build accurate classification models.

- **Problems you want to find answers:**

We are predicting if the Falcon 9 will land successfully





## Section 1

# Methodology

## DATA COLLECTION:

- DATA COLLECTION FROM --  
API AND WEBSCRAPING.
- RAW DATA CHECKING
- DATA PROCESSING
- EXPORT DATA TO CSV FILE

# 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

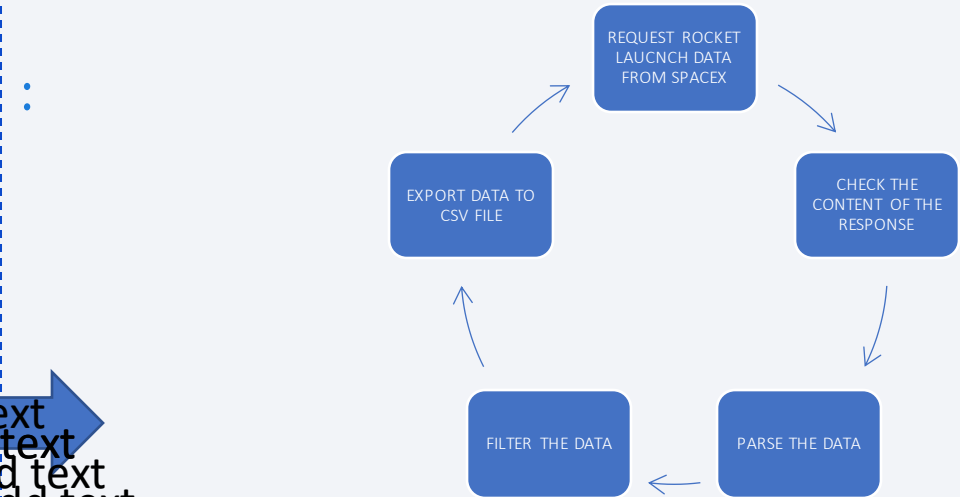
---

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

# Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

Click to add text  
Click to add text  
Click to add text  
Click to add text



```
File Edit View Insert Runtime Tools Help Δ changes saved
+ Code + Text

Import Libraries and Define Auxiliary Functions

We will import the following libraries into the lab

[1] # Requests allow us to make HTTP requests which we will use to get data from an API
import requests
# Pandas is a software library written for the Python programming language for data manipulation and analysis..
import pandas as pd
# Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of
import numpy as np
# Datetime is a library that allows us to represent dates
import datetime

# Setting this option will print all columns of a dataframe
pd.set_option("display.max_columns": None)
# Setting this option will print all of the data in a feature
pd.set_option("display.max_colwidth", None)

Below we will define a series of helper functions that will help us use the API to extract information using identification numbers in the launch
data.
From the rockets column we would like to learn the booster name.

[2] # 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['rockets']:
        response = requests.get("https://api.spacexdata.com/v4/rockets/" + str(x)).json()
        BoosterVersion.append(response['name'])

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

[3] # 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['launchpads']:
        response = requests.get("https://api.spacexdata.com/v4/launchpads/" + str(x)).json()
        Longitude.append(response['longitude'])
        Latitude.append(response['latitude'])
        LaunchSite.append(response['name'])
```



# Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts

Web scraping, was used to collect Falcon 9 historical launch records from a wiki pedia page titled list of Falcon heavy launches data .It is extracted from tables in the html code then is parsed and converted into a pandas data frame

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` `Wiki` page updated on 9th June 2021

```
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
page=requests.get(static_url)
```

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

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

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

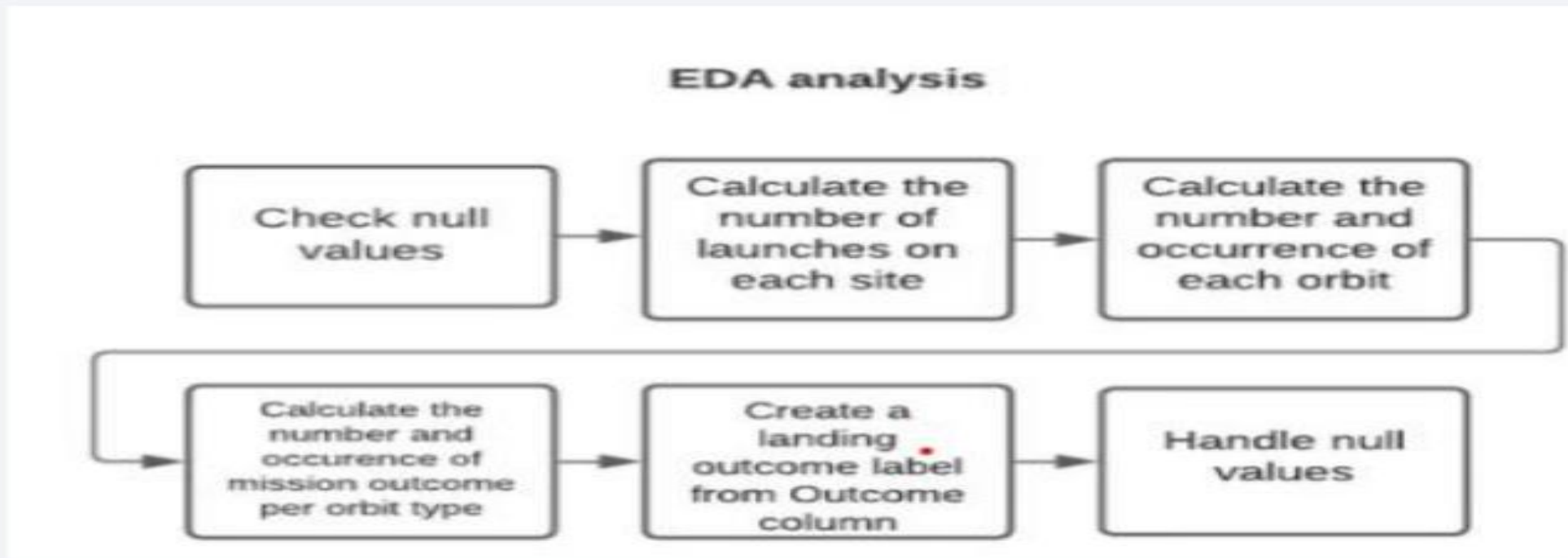
```
In [7]: # Use soup.title attribute
soup.title
```

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

# Data Wrangling

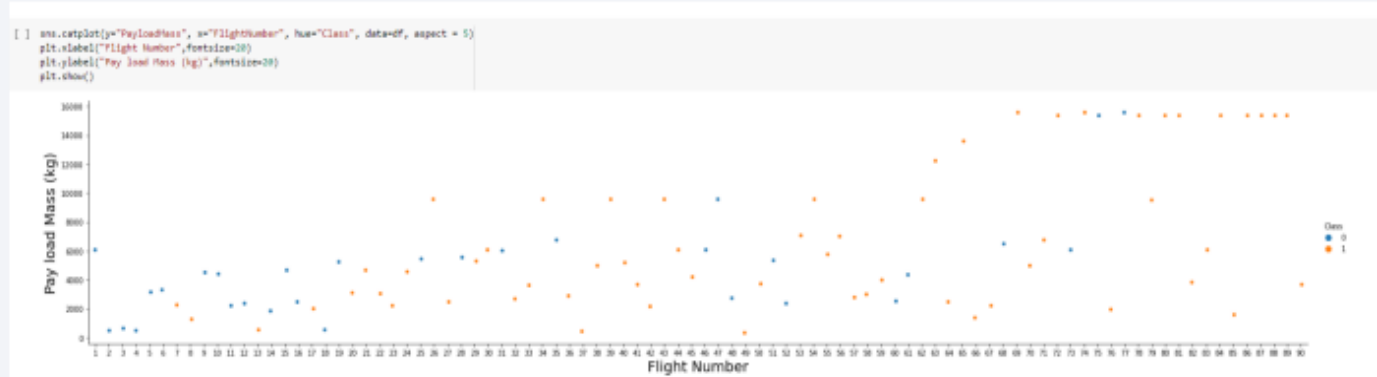
---

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts



# EDA with Data Visualization

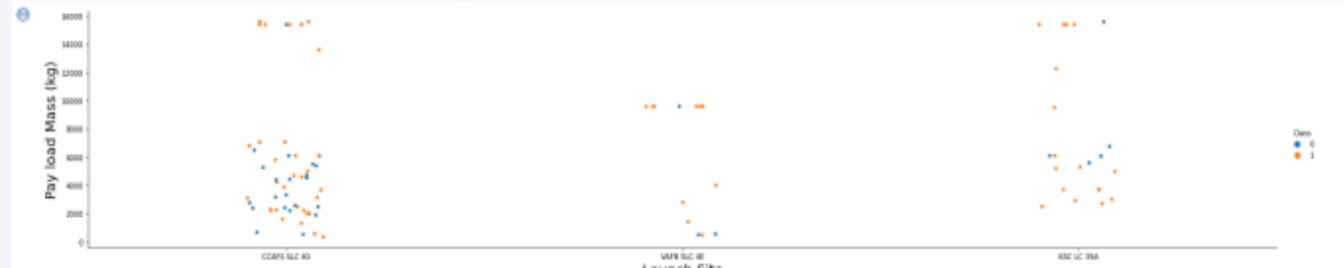
- Summarize what charts were plotted and why you used those charts



TASK 2: Visualize the relationship between Payload and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Launch Site", fontsize=20)
plt.ylabel("Pay Load Mass (kg)", fontsize=20)
plt.show()
```

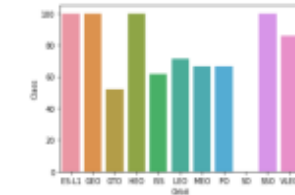


```
# FOMT use groupby method on Orbit column and get the mean of Class
temp = df.groupby(["Orbit"]).mean().reset_index()
temp2 = temp[["Orbit", "Class"]]
temp2["Class"] = temp2["Class"]*100
sns.kdeplot(x = "Orbit", y = "Class", data = temp2)
```



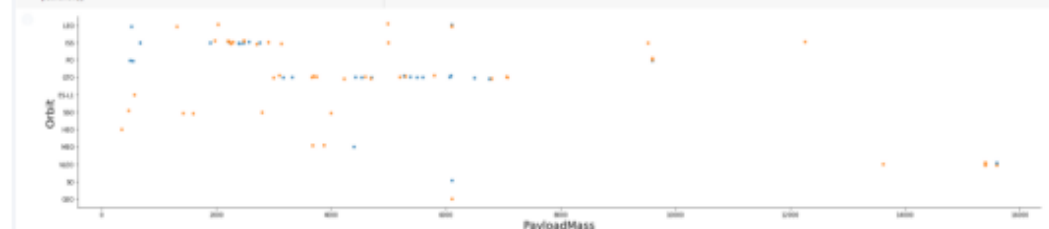
```
(ipython-input-48-788f588a37):4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)



Similarly, we can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and 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
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass", fontsize=20)
plt.ylabel("Orbit", fontsize=20)
plt.show()
```



# EDA with SQL

---

- Using bullet point format, summarize the SQL queries you performed
- Names of the unique launch sites in the space mission.
- 5 records where launch begins with the string 'CCA' .
- Total payload mass carried by booster launched by NASA (CRS)
- Average payload mass carried booster version F9v1.1.
- Date when the first successful landing outcome in the grouped pad was achieved.

# Build an Interactive Map with Folium

---

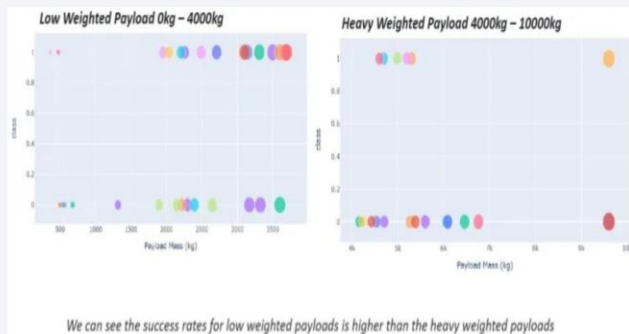
- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose



# Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions

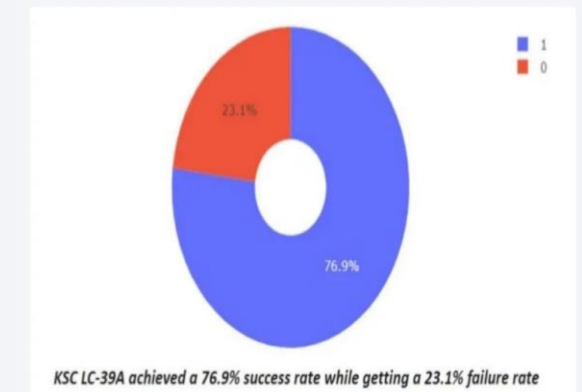
## Payload vs launch outcome



## Total success launches by all sites



## Success rate by sites



# Predictive Analysis (Classification)

---

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

# Results

---

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



The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of blue and red, creating a sense of motion or data flow. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant. The overall effect is high-tech and digital.

Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

---

- Show a scatter plot of Flight Number vs. Launch Site
- Show the screenshot of the scatter plot with explanations



# Payload vs. Launch Site

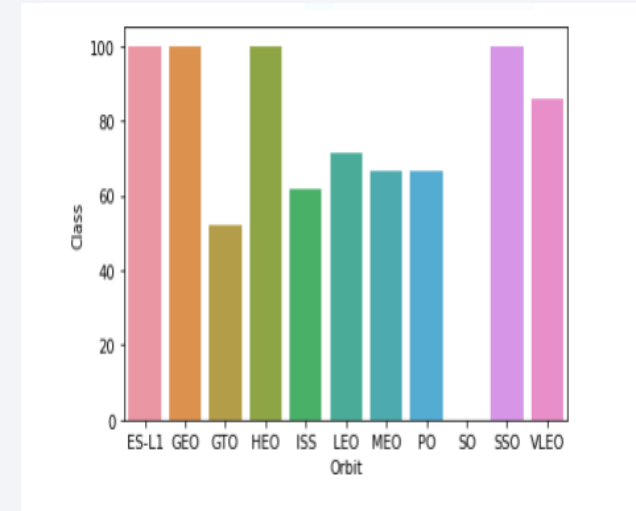
---

- Show a scatter plot of Payload vs. Launch Site
- Show the screenshot of the scatter plot with explanations

# Success Rate vs. Orbit Type

---

- Show a bar chart for the success rate of each orbit type
- Show the screenshot of the scatter plot with explanations



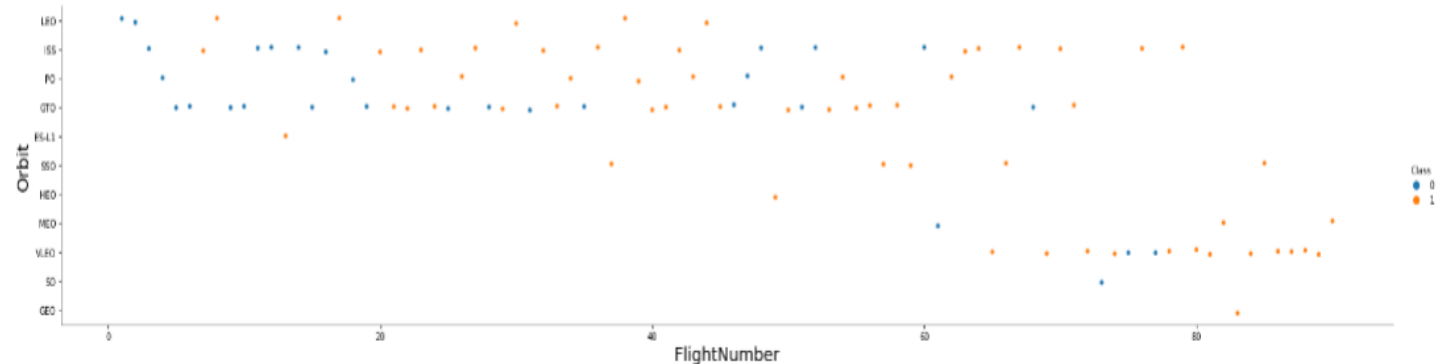
- The orbit of ES-L1 , GEO , HEO ,SSO are among the highest success rate

# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type
- Show the screenshot of the scatter plot with explanations

In [ ]:

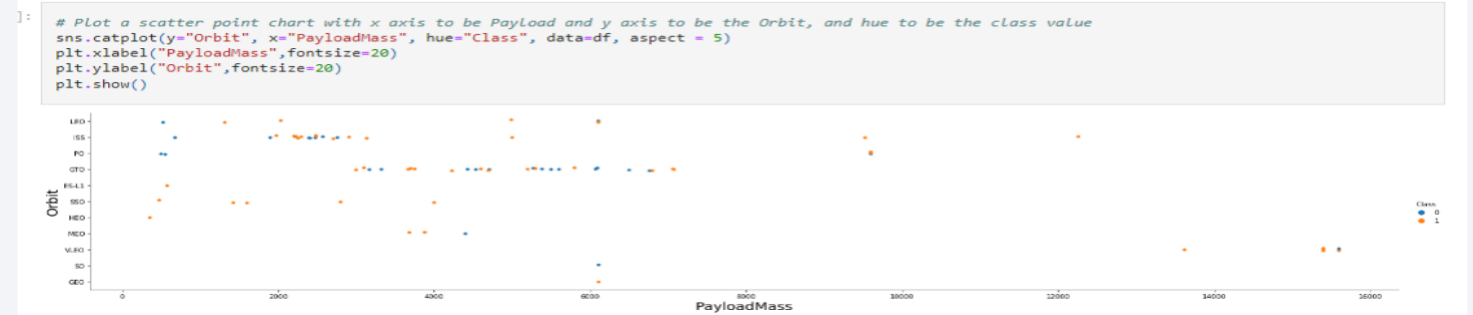
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



- The trend can be observed of shifting to VLEO launches in next years.

# Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations

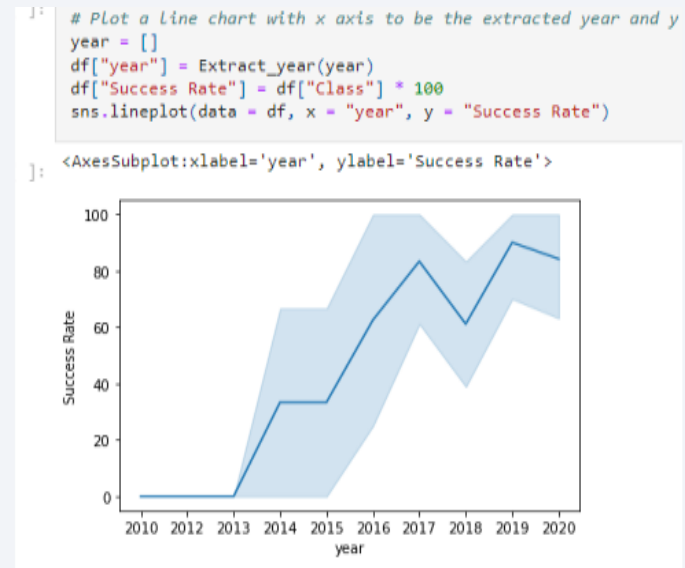


- There is a strong relation between ISS and Payload at a range around 2000

# Launch Success Yearly Trend

---

- Show a line chart of yearly average success rate
- Show the screenshot of the scatter plot with explanations



Launch rate have increase significantly since 2013



# All Launch Site Names

---

- Find the names of the unique launch sites
- Present your query result with a short explanation here

```
%sql select DISTINCT LAUNCH_SITE from SPACEXDATASET
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612
Done.
 launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E
```

# Launch Site Names Begin with 'CCA'

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

```
] %sql select * from SPACEXDATASET where launch_site like 'CCA%' limit 5
```

```
* ibm_db_sa://nx527972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
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 | 07:44:00 | F9 v1.0 B0005   | CCAFS LC-40 | Dragon demo flight C2   | 525             | LEO (ISS) | NASA (COTS)     | Success         | No attempt          |
| 2012-10-08 | 00:35:00 | 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          |

# Total Payload Mass

---

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

```
%sql select sum(payload_mass_kg_) as sum from SPACEXDATASET where customer like 'NASA (CRS)'
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB  
Done.
```

```
SUM
```

```
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

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

```
] %sql select avg(payload_mass__kg_) as Average from SPACEXDATASET where booster_version like 'F9 v1.1%'
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done,
]: average
2534
```

# 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

```
%sql select min(date) as Date from SPACEXDATASET where mission_outcome like 'Success'
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB  
Done.
```

```
DATE
```

```
2010-06-04
```



## 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

```
%sql select booster_version from SPACEXDATASET where (mission_outcome like 'Success')  
AND (payload_mass__kg_ BETWEEN 4000 AND 6000) AND (landing__outcome like 'Success (drone ship)')
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB  
Done.
```

```
: booster_version
```

```
F9 FT B1022
```

```
F9 FT B1026
```

```
F9 FT B1021.2
```

```
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

List the total number of successful and failure mission outcomes

```
] : %sql SELECT mission_outcome, count(*) as Count FROM SPACEXDATASET GROUP by mission_outcome ORDER BY mission_outcome
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB  
Done.
```

```
] :
```

| mission_outcome                  | COUNT |
|----------------------------------|-------|
| Failure (in flight)              | 1     |
| Success                          | 99    |
| Success (payload status unclear) | 1     |

# 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

```
maxm = %sql select max(payload_mass__kg_) from SPACEXDATASET
maxv = maxm[0][0]
%sql select booster_version from SPACEXDATASET where
payload_mass__kg_=(select max(payload_mass__kg_) from SPACEXDATASET)
```

Rectangular Snip

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
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

---

- 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

List the records which will display the month names, failure landing\_outcomes in drone ship ,booster versions, launch\_site for the months in year 2015

```
%sql select MONTHNAME(DATE) as Month, landing__outcome, booster_version, launch_site
from SPACEXDATASET where DATE like '2015%' AND landing__outcome like 'Failure (drone ship)'
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

| 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

---

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- Present your query result with a short explanation here

Rectangular Snip

```
%sql select landing__outcome, count(*) as count from SPACEXDATASET
where Date >= '2010-06-04' AND Date <= '2017-03-20'
GROUP by landing__outcome ORDER BY count Desc
```

```
* ibm_db_sa://nxs27972:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

| landing__outcome       | COUNT |
|------------------------|-------|
| No attempt             | 10    |
| Failure (drone ship)   | 5     |
| Success (drone ship)   | 5     |
| Controlled (ocean)     | 3     |
| Success (ground pad)   | 3     |
| Failure (parachute)    | 2     |
| Uncontrolled (ocean)   | 2     |
| Precluded (drone ship) | 1     |

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

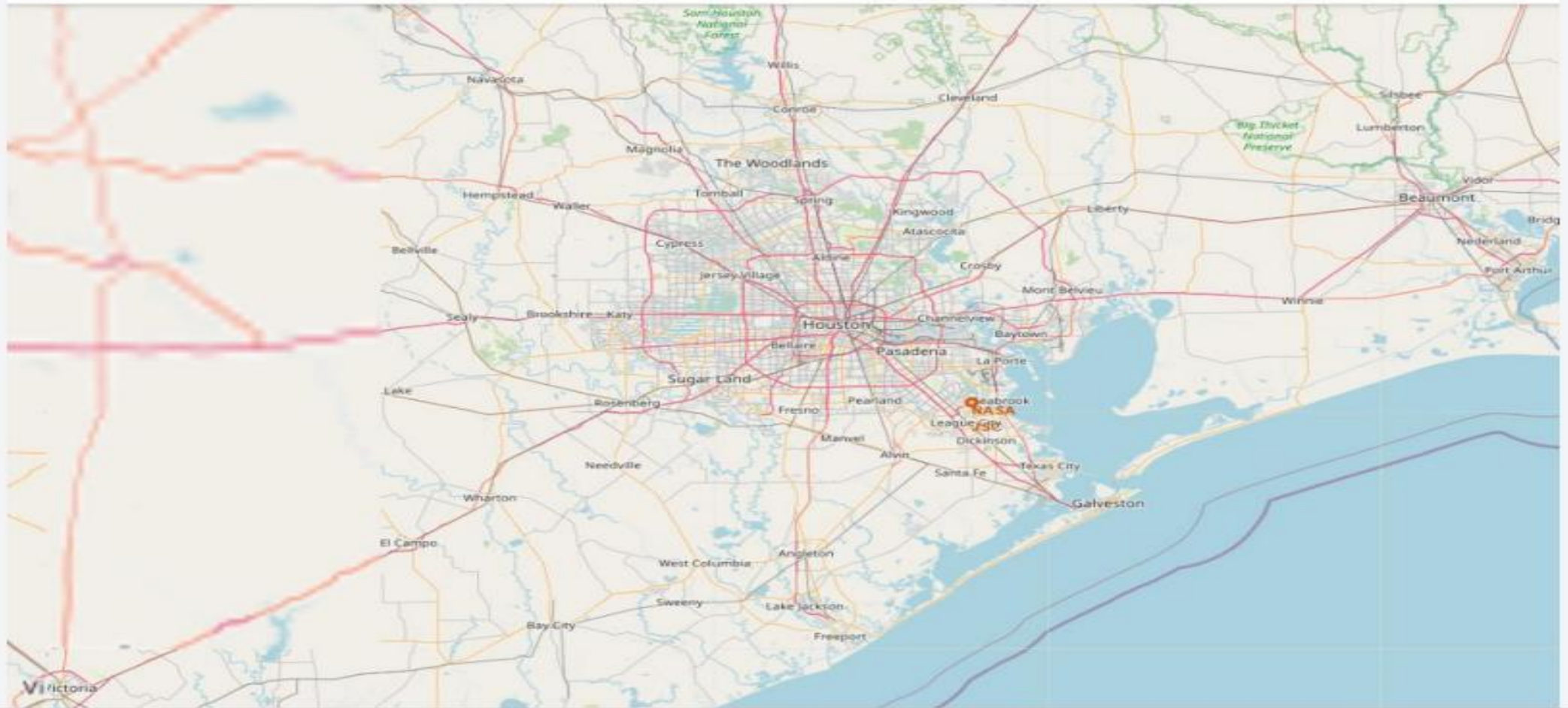
# Launch Sites Proximities Analysis



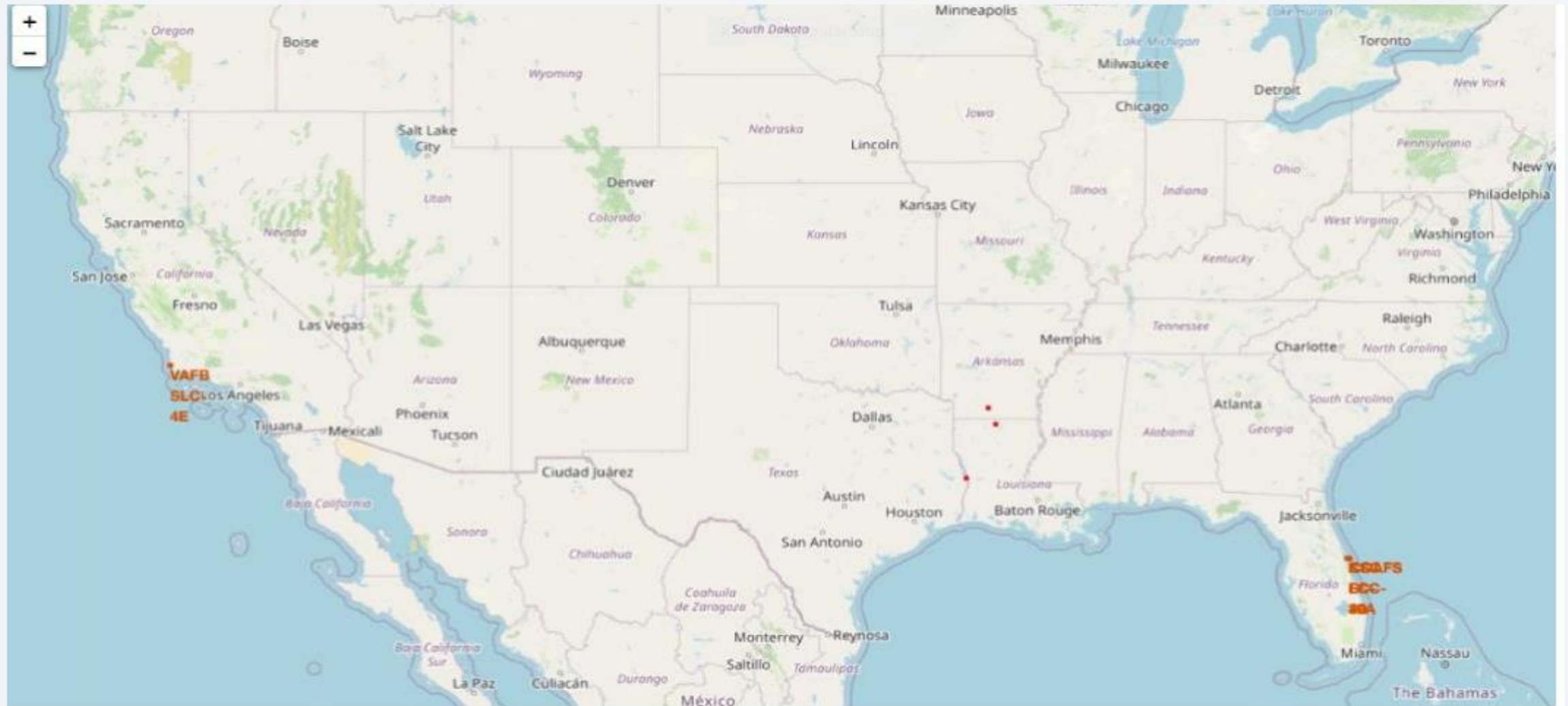
# <Folium Map Screenshot 1>

folium.ipynb ☆

help Last saved at 17:43

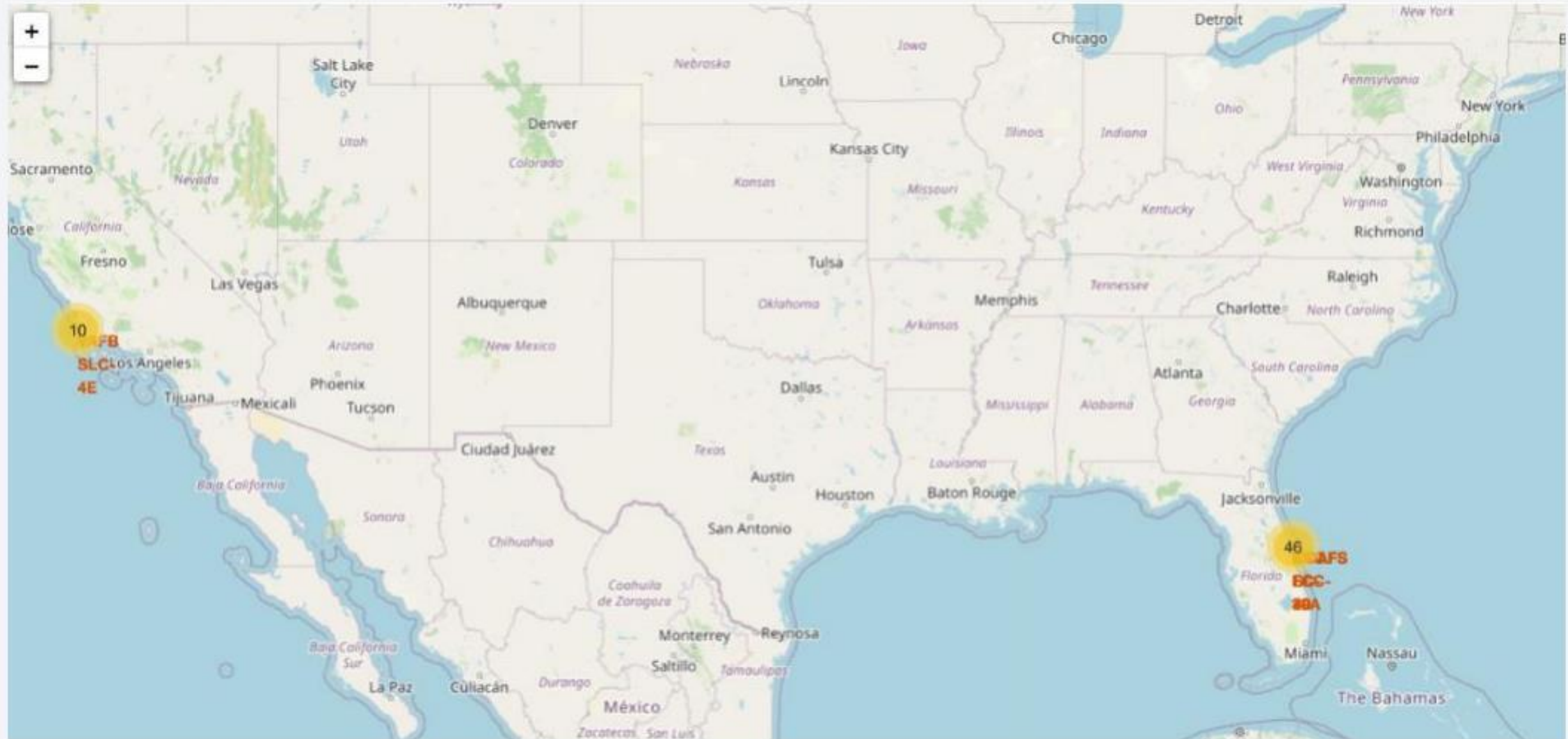


# <Folium Map Screenshot 2>





# <Folium Map Screenshot 3>





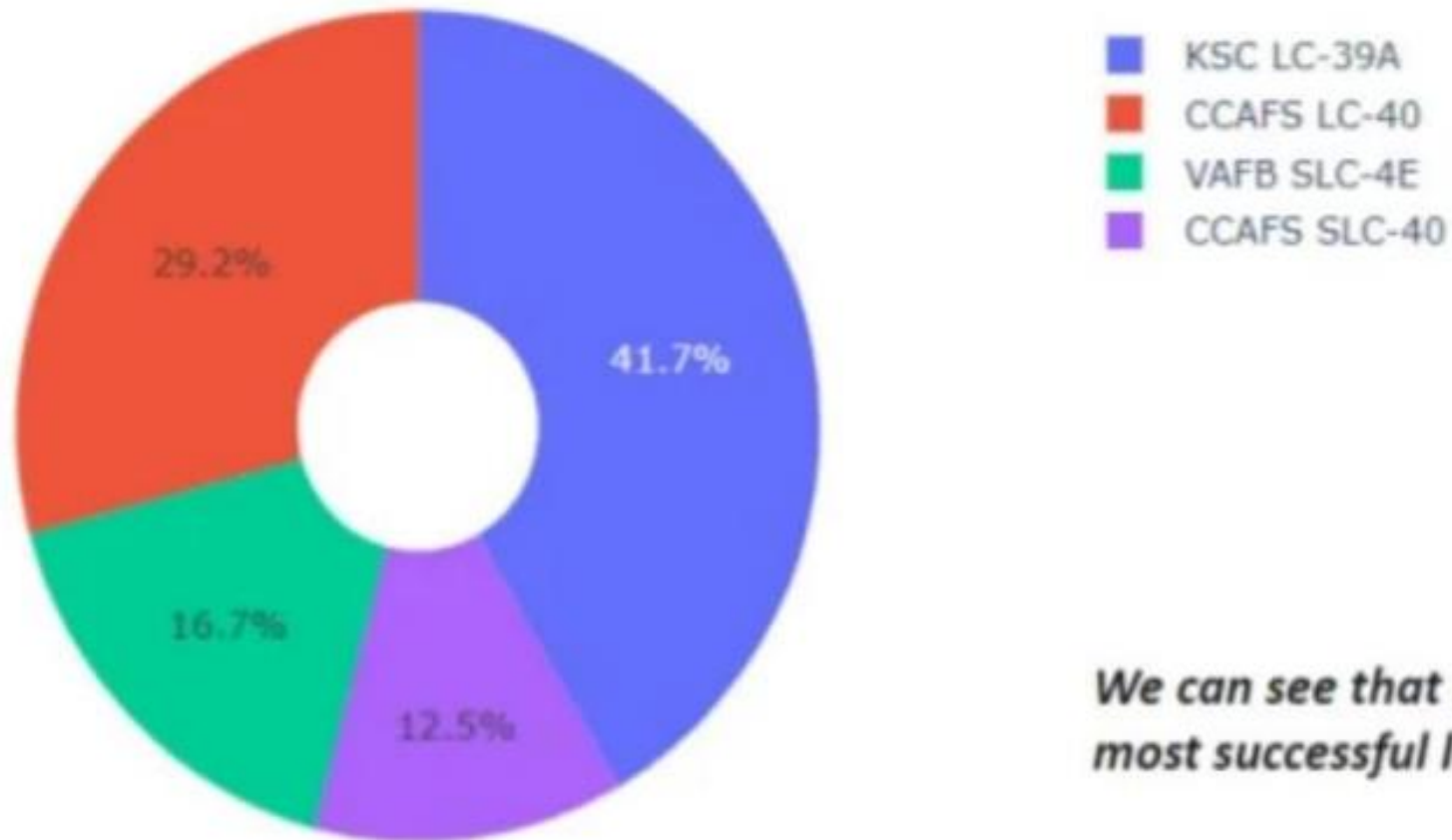
Section 4

# Build a Dashboard with Plotly Dash



## <Dashboard Screenshot 1>

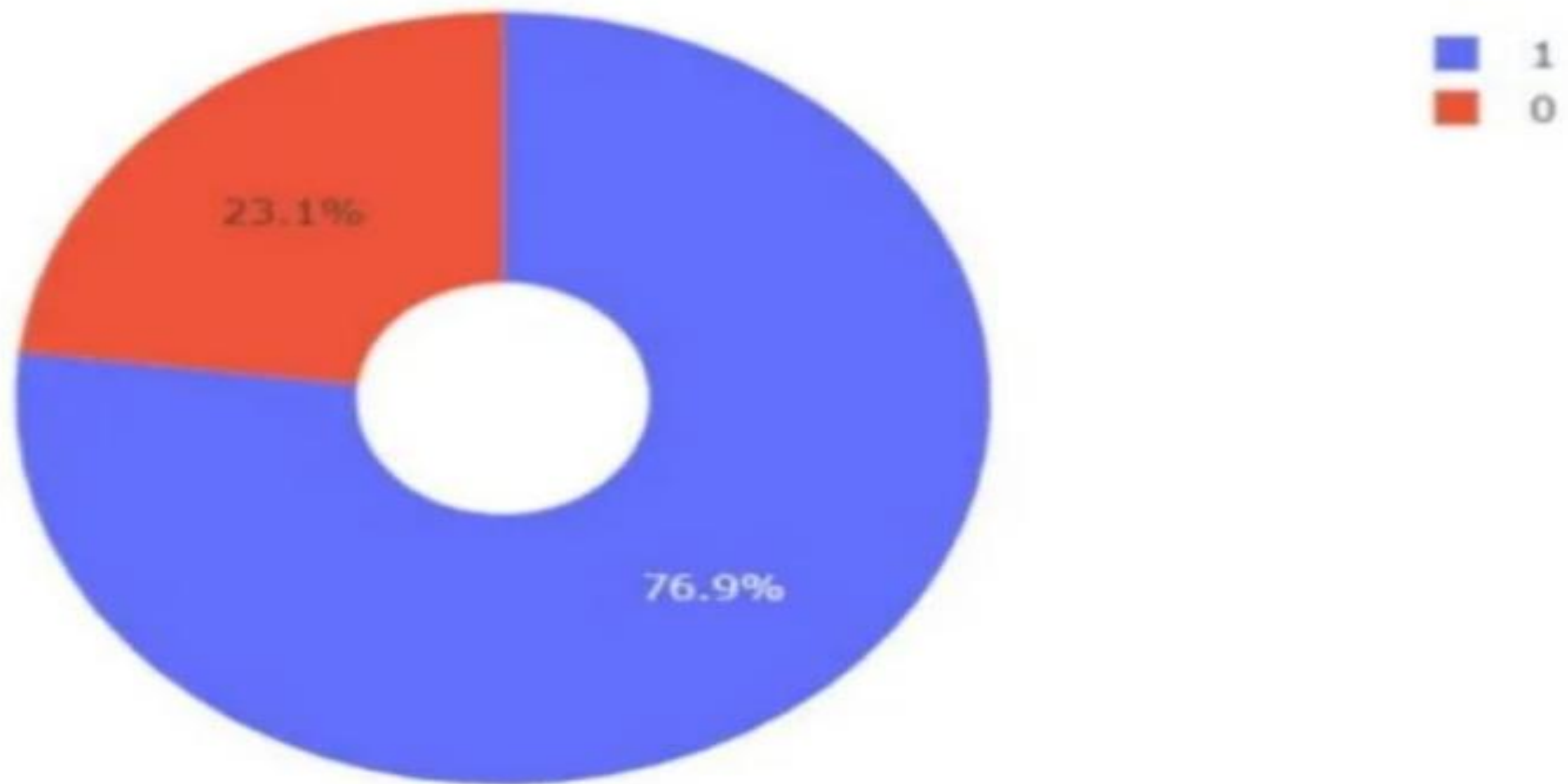
Total Success Launches By all sites



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

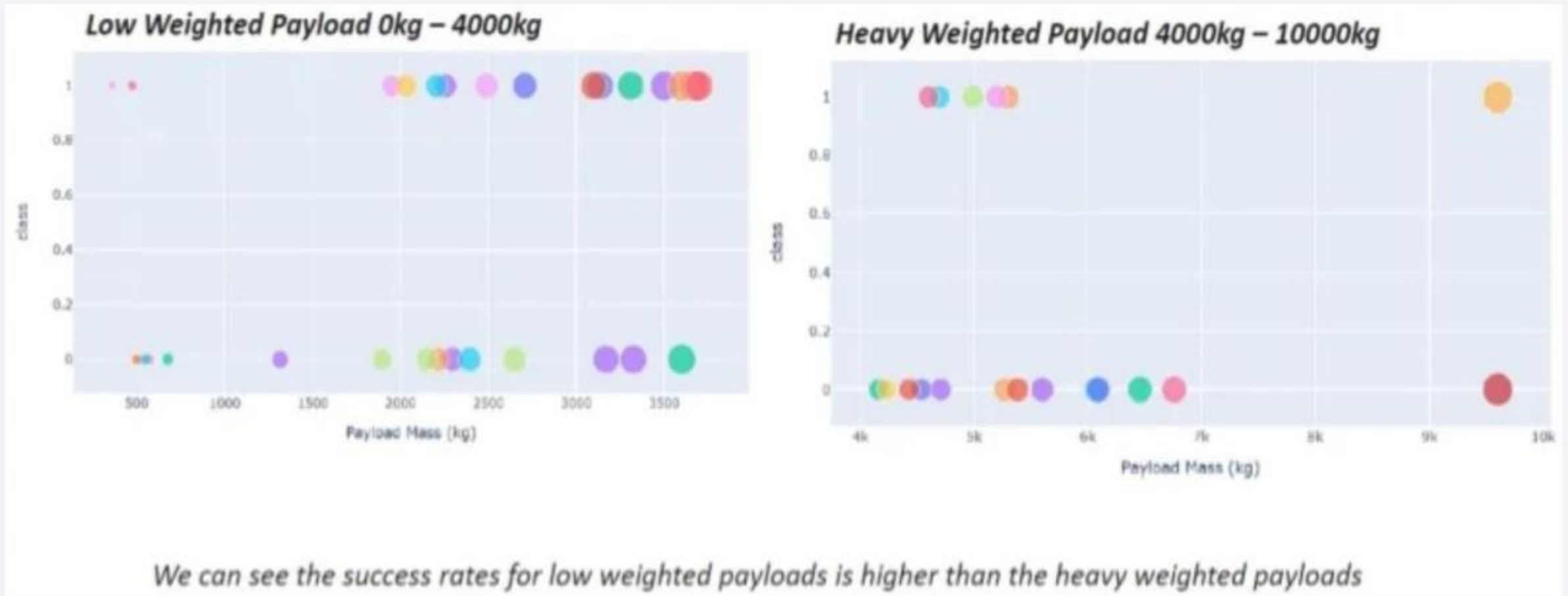
## <Dashboard Screenshot 2>

---



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

## <Dashboard Screenshot 3>





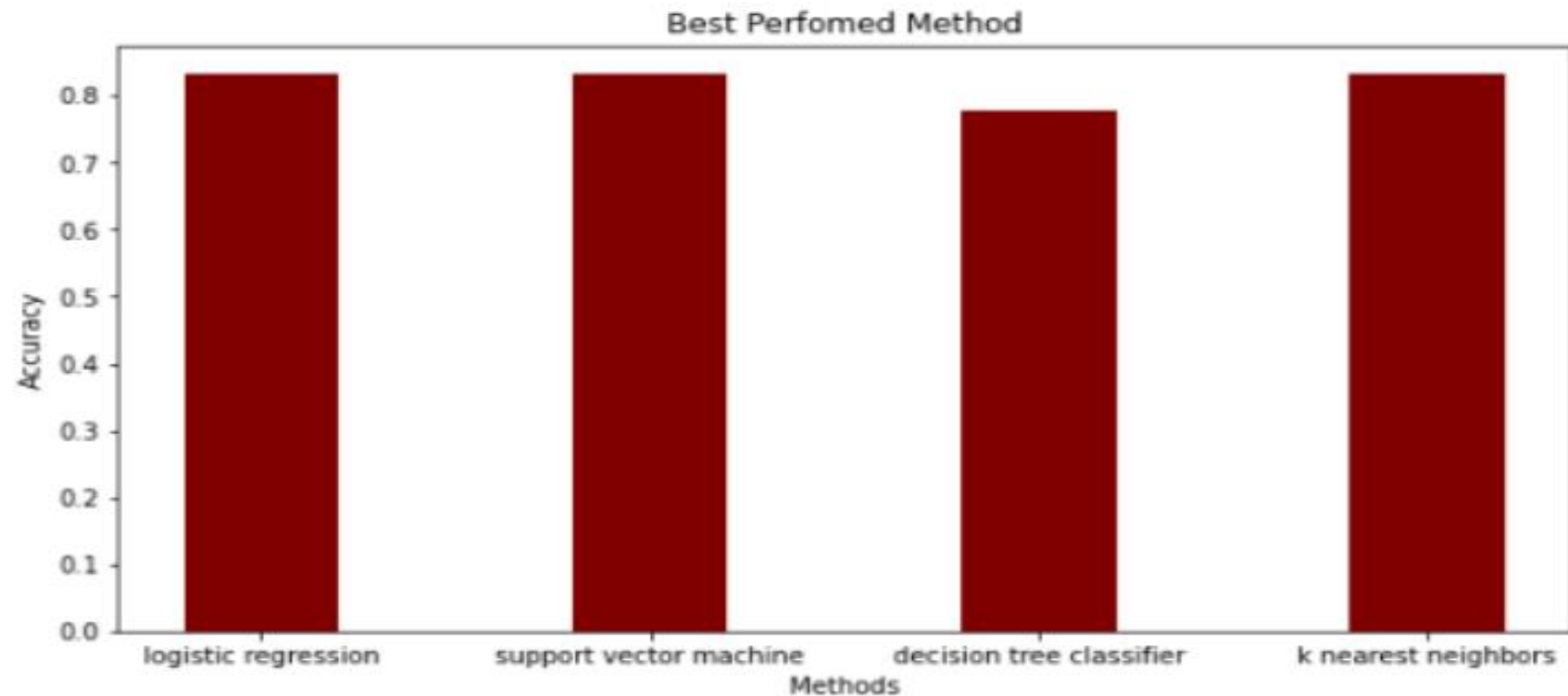
Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

---

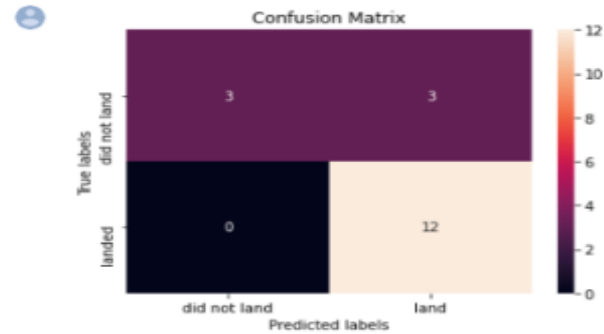
```
plt.xlabel("Methods")  
plt.ylabel("Accuracy")  
plt.title("Best Perfomed Method")  
plt.show()
```



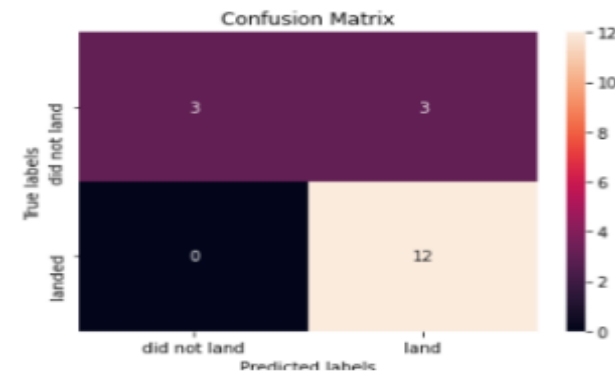
# Confusion Matrix

We can plot the confusion matrix

```
yhat=svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```

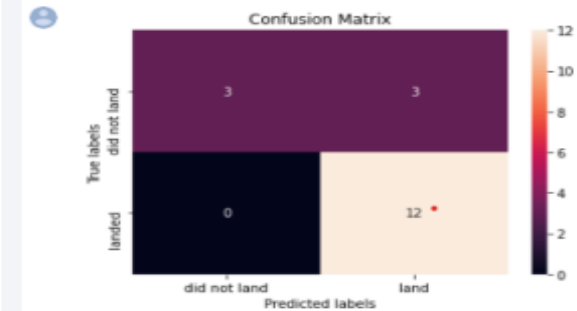


```
yhat=logreg_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```

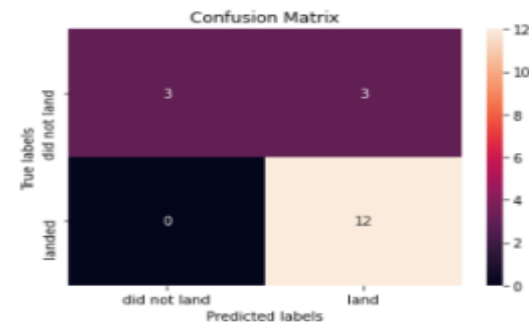


We can plot the confusion matrix

```
yhat = knn_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```



```
[ ] yhat = svm_cv.predict(X_test)  
plot_confusion_matrix(Y_test,yhat)
```





# Conclusions

---

- Point 1- Low weight payload performs better than the heavier payload
- Point 2-The success rate for spacex launches is directly proportional to time in years they will eventually the launches
- Point 3-Orbit GEO,HEO,ESL 1 has the best success rate
- Point 4-KSALC 39A had the most successful launches from all the sites
- The SVM and LR are the best models in terms of prediction and accuracy for the provided dataset
- ...

# 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

```
Df_pivot=df_grp.  
Pivot(index='drive_wheels',  
columns='body_style')
```

Thank you!

