IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Cormac Vautier
7/10/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection via API and HTML Scraping
  - EDA via SQL and Python
  - Data visualization via Folium Python module
  - Plotly Dashboard
  - Data Analysis via Python scikit machine learning

Introduction

- The Project explores the data set of SpaceX launch sites. We will use this predict whether the Falcon 9 launch will be successful and the associated cost with the launch.
- Classify and find the success rate of launch sites to pick the launch site to give the Falcon 9 the best odds of success. Generate associated characteristics for launch sites for use by competitors.
- Links to the code are in the slide titles.

Section 1

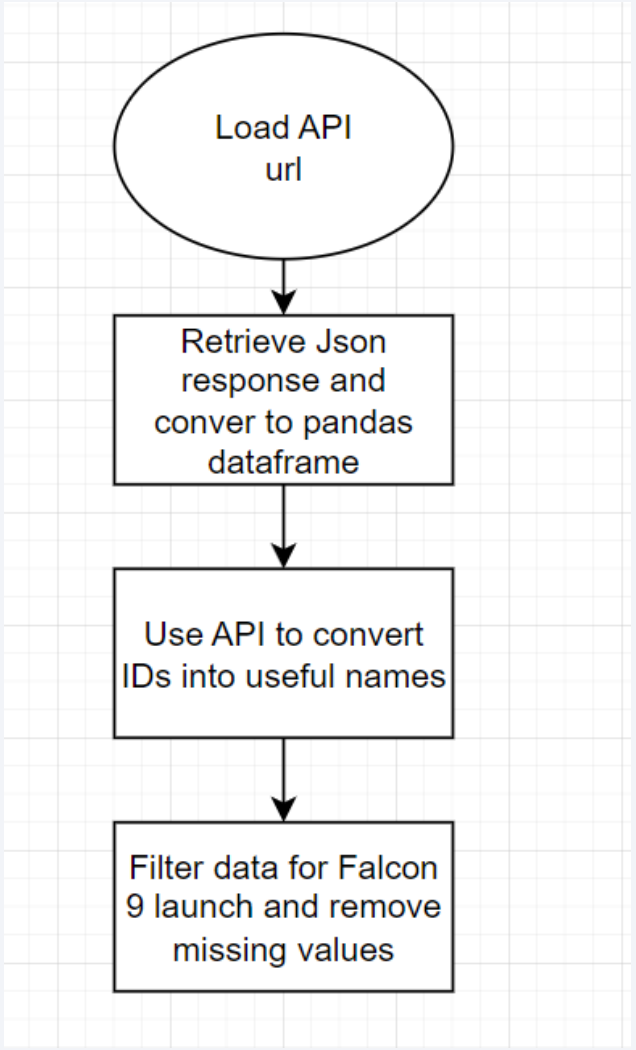
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API was used alongside beautiful soup
  - Web scraping was done on HTML of SpaceX's Wikipedia page
- Perform data wrangling
  - Drop extraneous columns
  - Use one hot encoding to classify categoric values
- 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 – SpaceX API



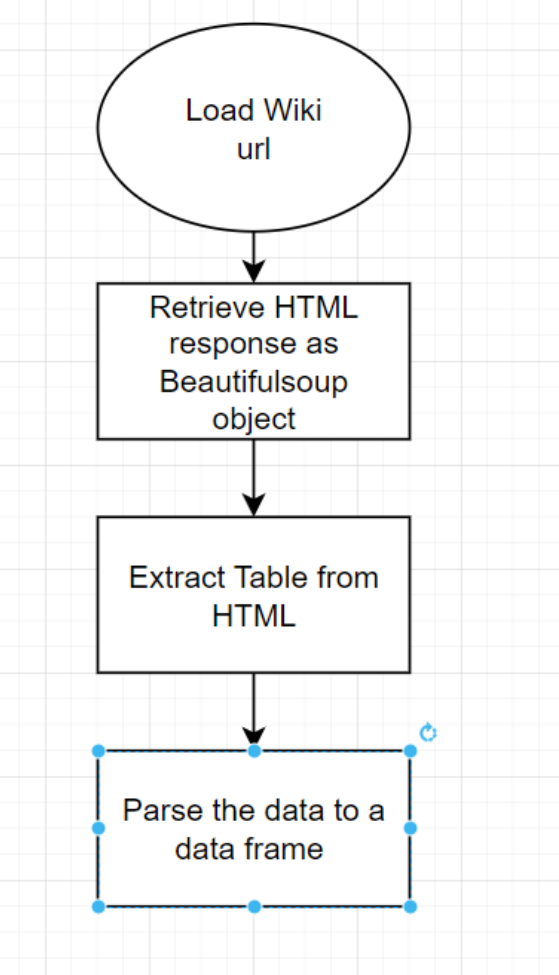
Code Snippets

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

```
# Use json_normalize meethod to convert the json result into a dataframe

response = requests.get(static_json_url)

response=response.json()  #convert result in json

#type(response)


data= pd.json_normalize(response)
```

```
data_falcon9=data[data['BoosterVersion']=='Falcon 9']

data_falcon9
```

Data Collection - Scraping



Code Snippets

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
# assign the response to a object
response=requests.get(static_url).text
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content

soup=BeautifulSoup(response)
```

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

```
# Use soup.title attribute

soup.title
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- One hot encoding was used to cover the categoric variables into numeric variables.
- A list of bad and good outcomes was used to give whether the mission was a success. This was passed to a new column OUTCOME for easy classification.

EDA with Data Visualization (Python)

- Scatter Graph
  - Flight Number vs Payload Mass
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Orbit type vs Flight Number
  - Payload vs Orbit Type
  - Orbit vs Payload Mass
- Bar Graph
  - Success rate vs Orbit
- Line Graph
  - Success rate vs Year

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS): 45596
- Display average payload mass carried by booster version F9 v1.1 : 2928.4
- List the date when the first successful landing outcome in ground pad was achieved: 2018-03-12
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass. Use a subquery
- 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
- 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.

Build an Interactive Map with Folium

- Folium map has a labelled red circle at the NASA Johnson Space Center at Houston, Texas
- Cluster markers are at each launch site with green or red markers indicator successful or unsuccessful landings.
- A line marker gives distance between the launch site and points of interest such as the sea and cities to give information for other possible launch sites.

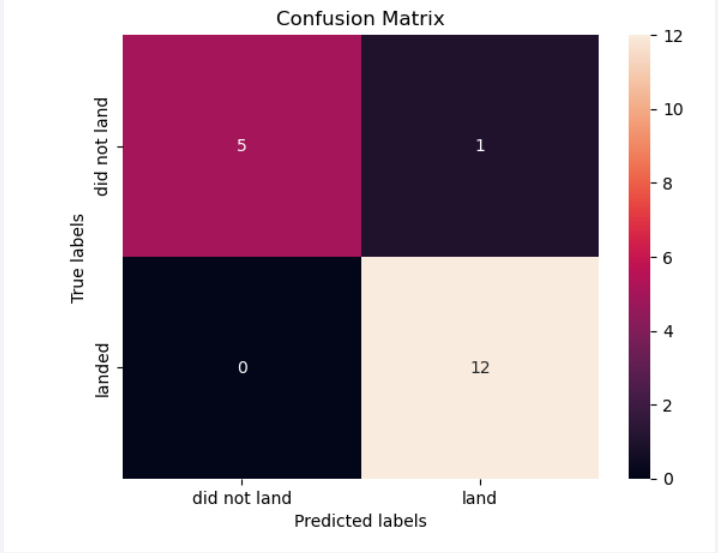
Build a Dashboard with Plotly Dash

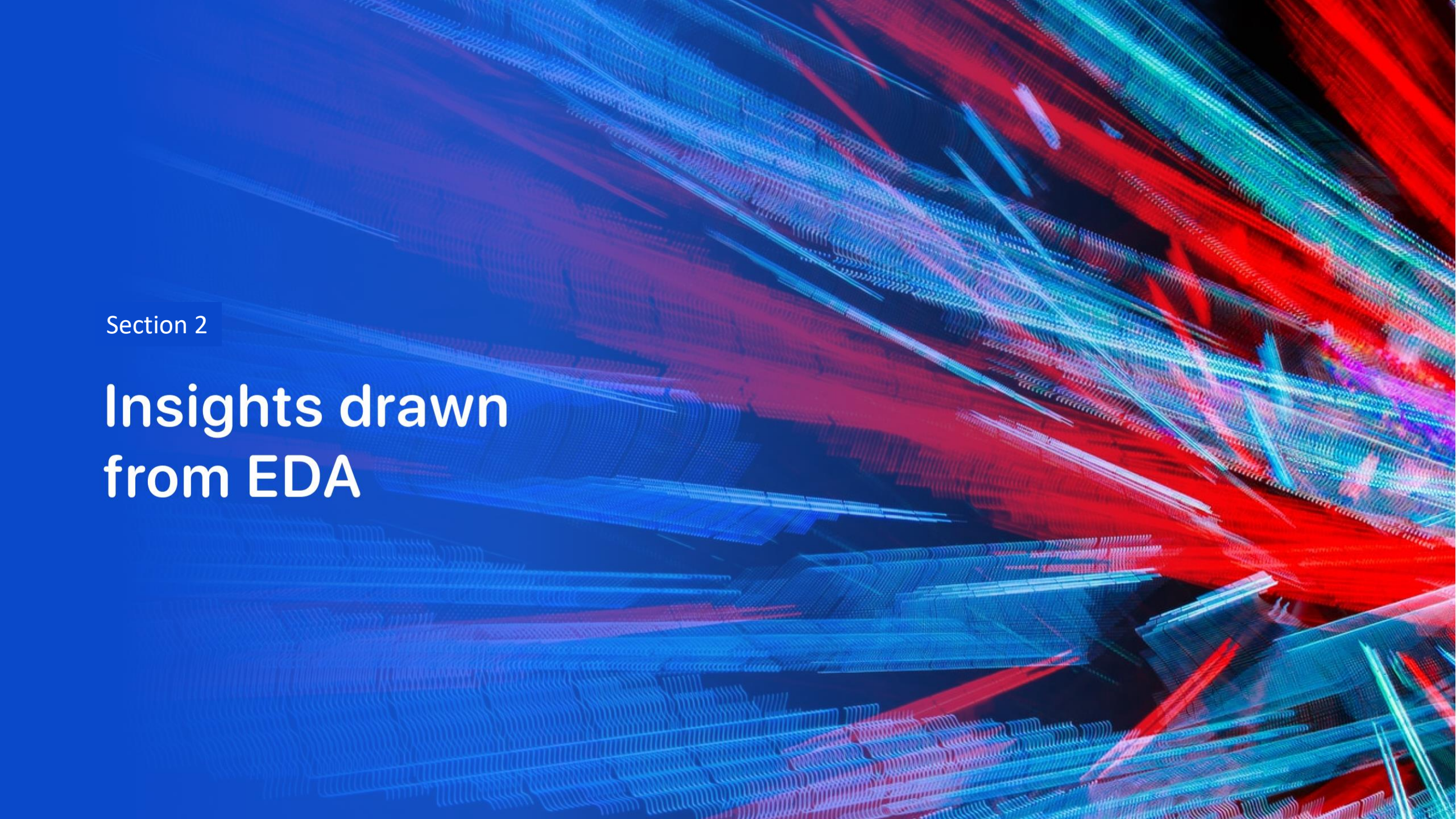
- The dashboard has a dropdown menu selecting each of the launch sites or all sites.
- Pie chart to show the success rate of launches from that site.
- Range slider to select a range of payload masses
- Scatter chart to show Successes vs Payload Mass

Predictive Analysis (Classification)

- Data was prepared by normalizing the data and splitting it into training and test data sets.
- 4 Models were prepared, SVM, Logistic Regression, K nearest neighbors, decision tree.

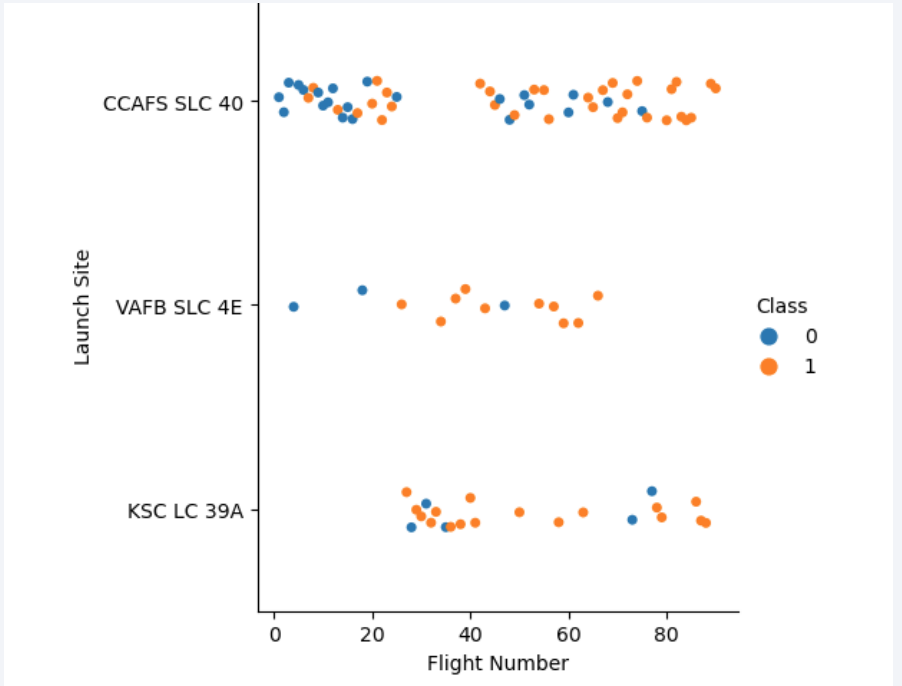
Confusion matrices were plotted for all models, with the decision tree method proving the best as shown to the right. R values further proved the accuracy of the decision tree method.




Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

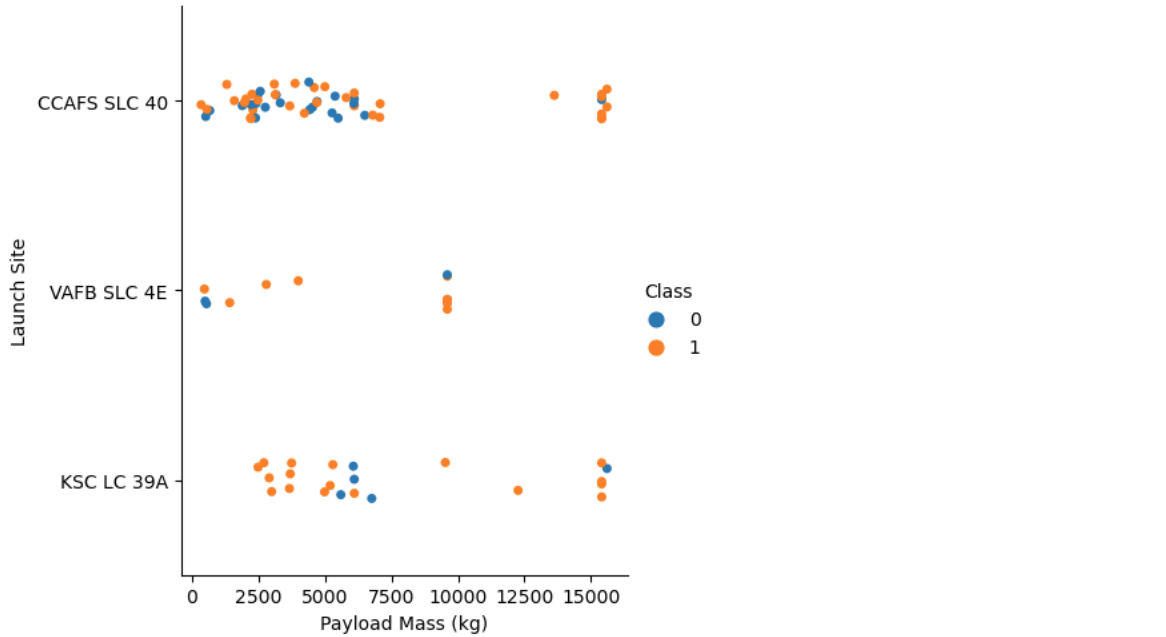
- The class gives launch success or failure with 1 being a successful launch and 0 being a failure. We can see that KSC LC 39A has the highest proportion of successful launches




```
]:
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class valu
sns.catplot(x="FlightNumber",y="LaunchSite",hue="Class",data=df)
plt.xlabel('Flight Number')
plt.ylabel('Launch Site')
plt.show()
```

Payload vs. Launch Site

- CCAFS SLC 40 has the majority of low Payload launches.

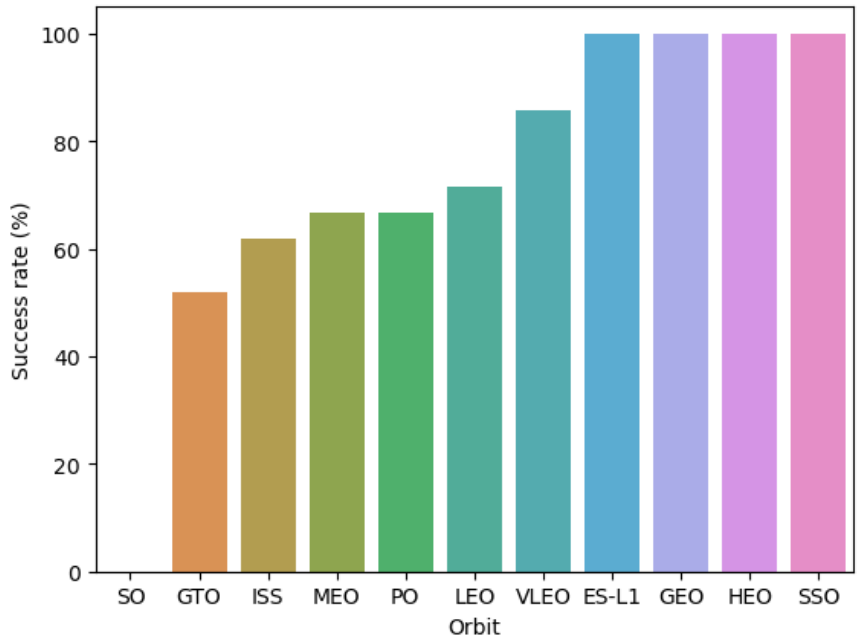


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

```
]:
# 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
sns.catplot(x="PayloadMass",y="LaunchSite",hue="Class",data=df)
plt.xlabel('Payload Mass (kg)')
plt.ylabel('Launch Site')
plt.show()
```

Success Rate vs. Orbit Type

- SO has no successes whilst ES-L1, GEO, HEO, SSO has only successful launches.



Analyze the ploted bar chart try to find which orbits have high sucess rate.

ES-L1,GEO,HEO,SSO have high success rates

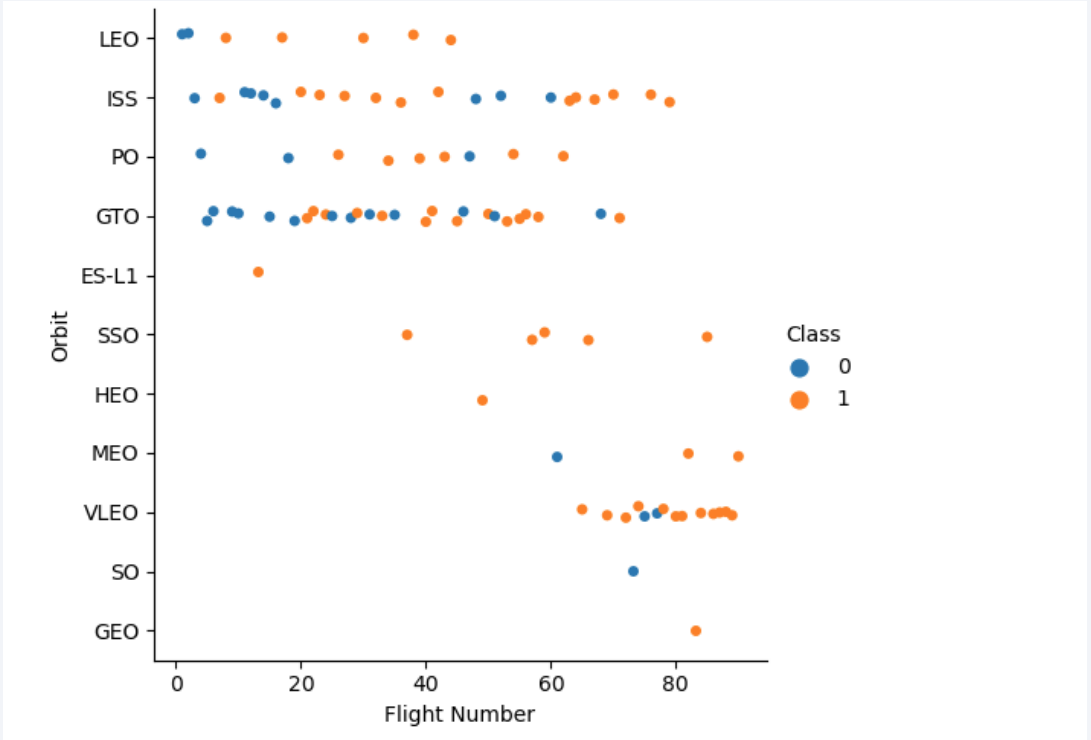
[18]:

```
# HINT use groupby method on Orbit column and get the mean of Class column
orb=df.groupby('Orbit')['Class'].mean().reset_index().sort_values(by='Class')
orb['Class']*=100
sns.barplot(data=orb,x='Orbit',y='Class')
plt.xlabel('Orbit')
plt.ylabel('Success rate (%)')
plt.show()
```

Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

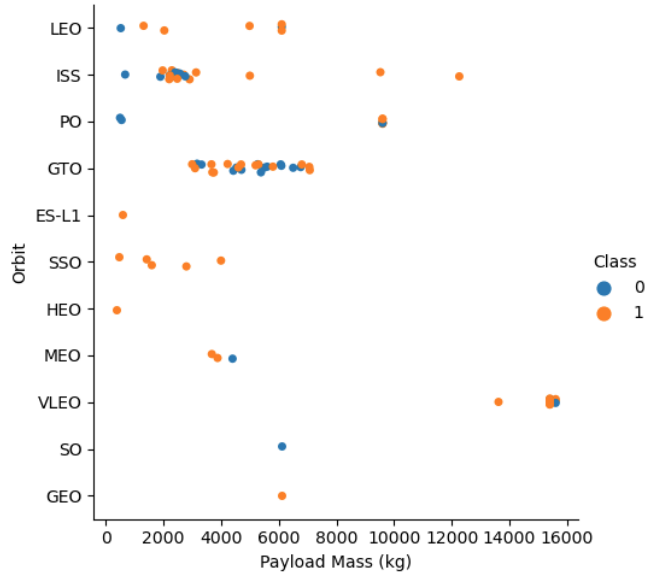
- Show the screenshot of the scatter plot with explanations



In [20]:

```
# 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(x="FlightNumber",y="Orbit",hue="Class",data=df)
plt.xlabel('Flight Number')
plt.ylabel('Orbit')
plt.show()
```

Payload vs. Orbit Type



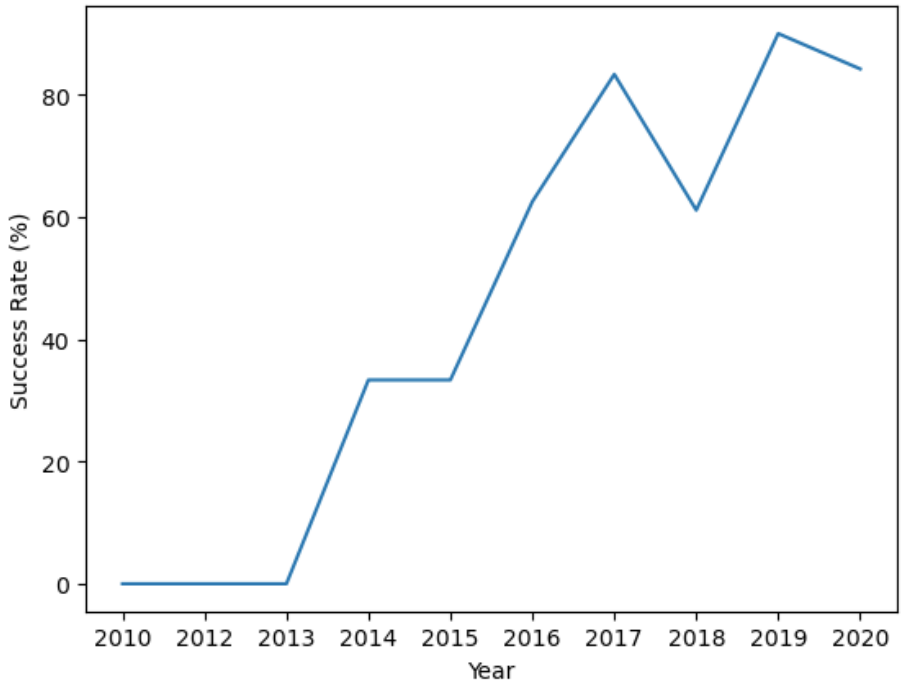
With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

In [21]:

```
# 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(x="PayloadMass",y="Orbit",hue="Class",data=df)
plt.xlabel('Payload Mass (kg)')
plt.ylabel('Orbit')
plt.show()
```

Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

```
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
yearly_data=df.groupby('Year')['Class'].mean().reset_index()
yearly_data['Class']*=100



sns.lineplot(data=yearly_data,x='Year',y='Class')
plt.xlabel('Year')
plt.ylabel('Success Rate (%)')
plt.show()
```

All Launch Site Names

In [13]:
```
%sql select DISTINCT(LAUNCH_SITE) FROM SPACEXTBL
```

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

Out[13]:

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

Launch Site Names Begin with 'CCA'

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

In [14]:
```
%sql select LAUNCH_SITE FROM SPACEXTBL WHERE LAUNCH_SITE like "%CCA%" LIMIT 5
```

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

Out[14]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

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

In [18]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER = "NASA (CRS)"
```

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

Out[18]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

In [19]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

Out[19]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

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

Hint:Use min function

In [21]:
```
%sql select min(DATE) FROM SPACEXTBL WHERE Landing_Outcome ='Success'
```

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

Out[21]:

min(DATE)
2018-03-12

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [25]:
```
%sql select booster_version from SPACEXTBL where Landing_Outcome = 'Success (drone ship)' and payload_mass__kg_ between 400
```

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

Out[25]:

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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [35]:
```
%sql select COUNT(Mission_Outcome),Mission_Outcome FROM SPACEXTBL GROUP BY Mission_Outcome
```

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

Out[35]:

COUNT(Mission_Outcome)	Mission_Outcome
1	Failure (in flight)
98	Success
1	Success
1	Success (payload status unclear)

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [38]:
```
%sql select DISTINCT(Booster_version), Payload_Mass__kg_ FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (select MAX(PAYLOAD_MASS_
```

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

Out[38]:

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2015 Launch Records

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

In [50]:
```
%sql select booster_version, launch_site, strftime('%m',DATE) as "Month" from SPACEXTBL where landing_outcome = 'Failure (d
```

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

Out[50]:

Booster_Version	Launch_Site	Month
F9 v1.1 B1012	CCAFS LC-40	10
F9 v1.1 B1015	CCAFS LC-40	04

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.

In [55]:
```
%sql select count(Landing_Outcome), Landing_Outcome from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' group b
```

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

Out[55]:

count(Landing_Outcome)	Landing_Outcome
10	No attempt
5	Success (ground pad)
5	Success (drone ship)
5	Failure (drone ship)
3	Controlled (ocean)
2	Uncontrolled (ocean)
1	Precluded (drone ship)
1	Failure (parachute)

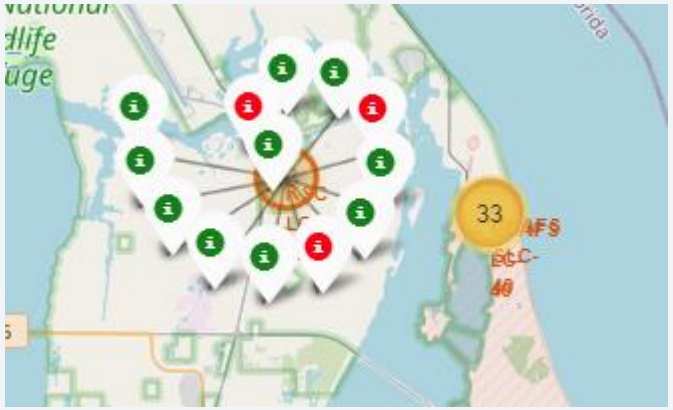
Section 3

Launch Sites Proximities Analysis

Folium map – All Launch sites

There are two sets if launch sites, one on the west coast near Los Angeles California and the other on the east coast near Orlando Florida

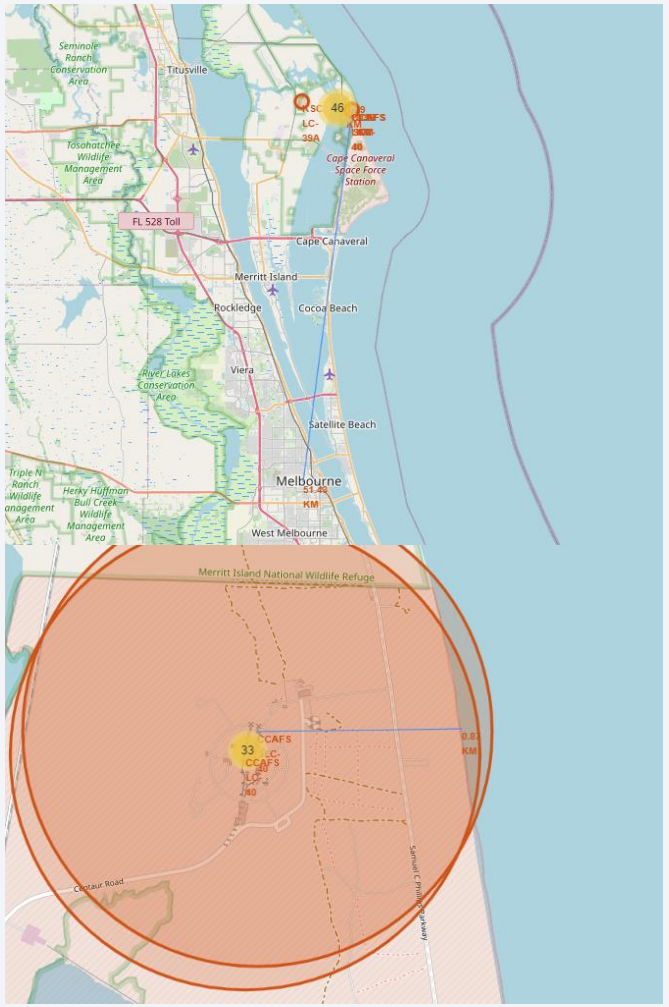
Folium map – Success/Failed Launch sites



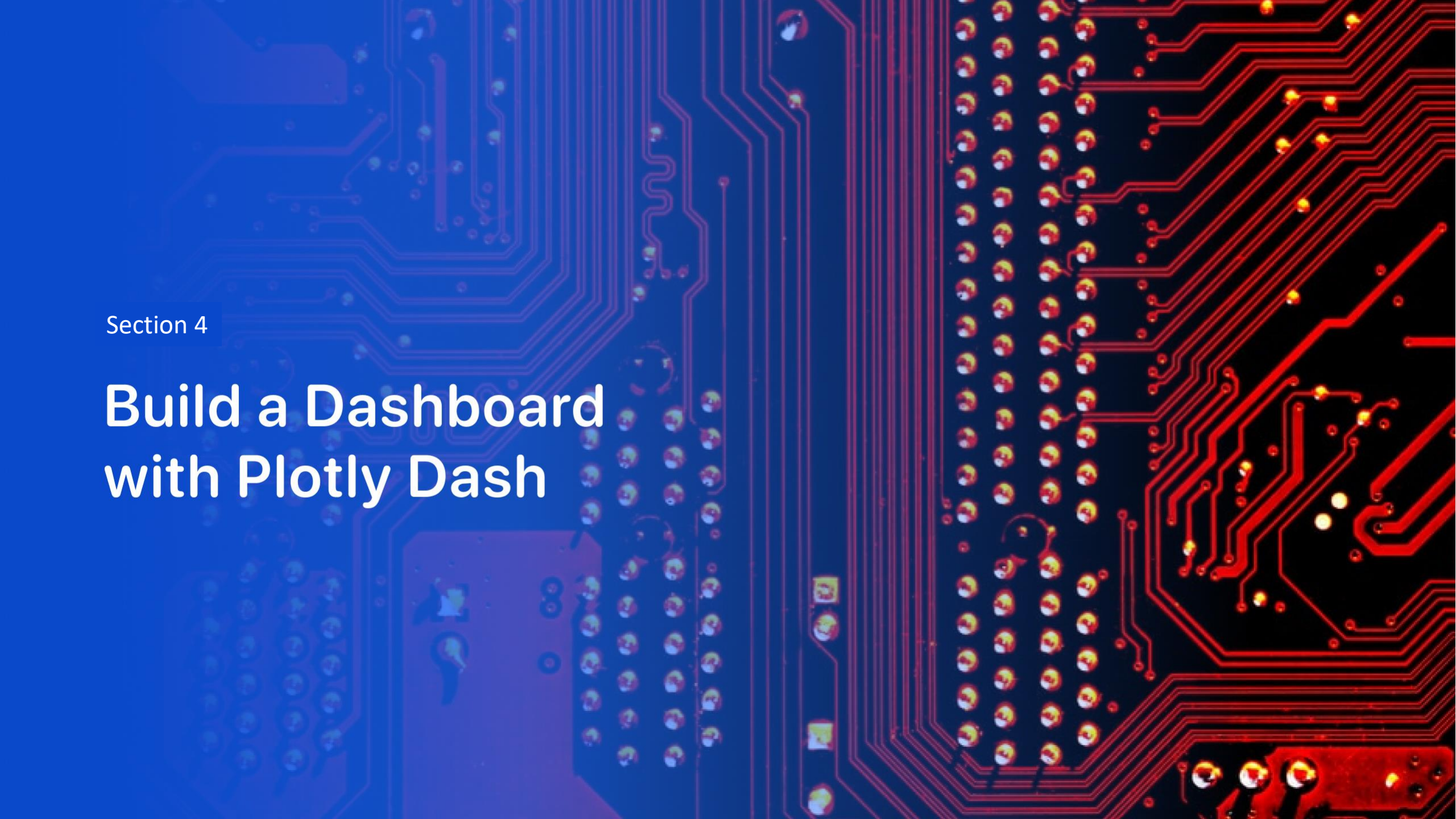
10 Sites are near Los Angeles, 46 near Orlando. These sites can be selected from a cluster to determine success (green) from failure (red).

Folium map – Distances to POI

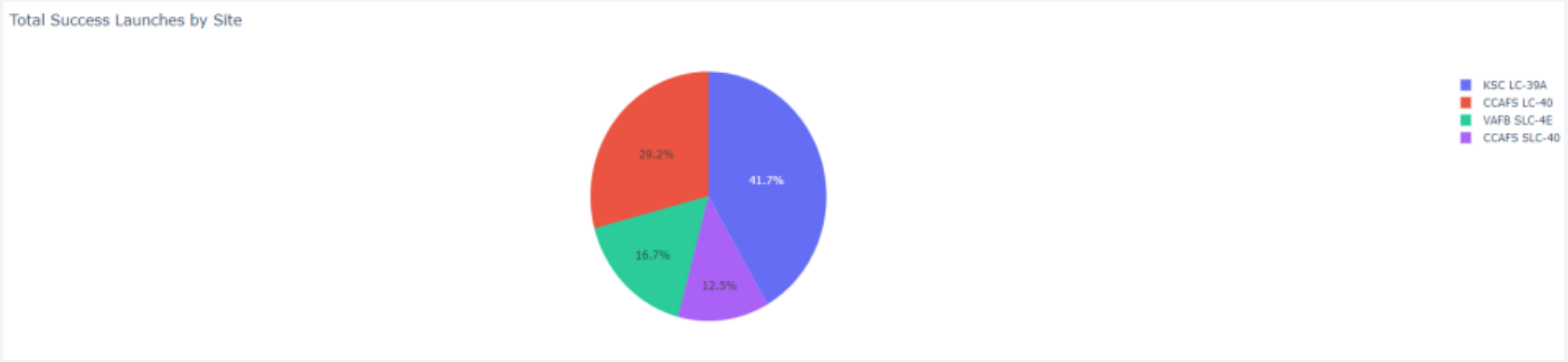
- Distances from major points of interest such a highways, railroads, cities and the sea are marked on the map with a blue line.




```
In [23]: distance_highway = calculate_distance(launch_site_lat, launch_site_lon, closest_highway[0], closest_highway[1])
         print('distance_highway =',distance_highway, ' km')
         distance_railroad = calculate_distance(launch_site_lat, launch_site_lon, closest_railroad[0], closest_railroad[1])
         print('distance_railroad =',distance_railroad, ' km')
         distance_city = calculate_distance(launch_site_lat, launch_site_lon, closest_city[0], closest_city[1])
         print('distance_city =',distance_city, ' km')

         distance_highway = 0.5834927849513031  km
         distance_railroad = 1.2851321381588416  km
         distance_city = 51.43339806036189  km
```

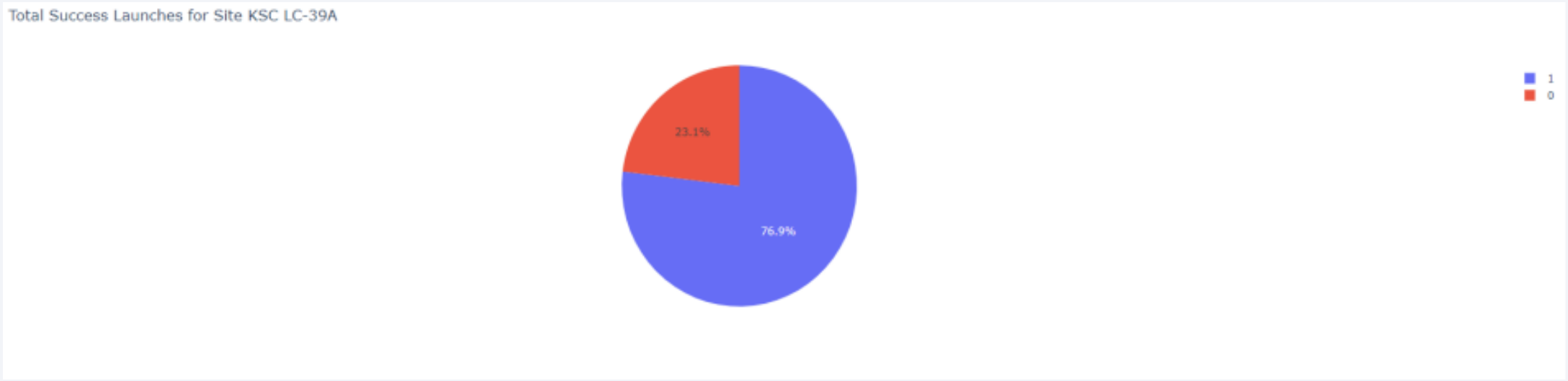

Section 4
Build a Dashboard with Plotly Dash

Plotly Dashboard- Total Success by Site



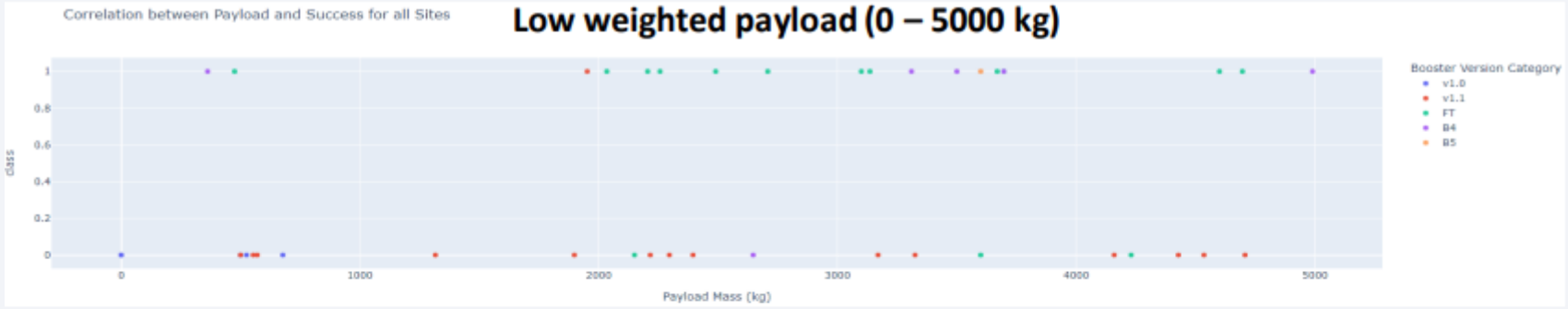
We can see that KSC LC-39A has the highest success rate of all launch sites.

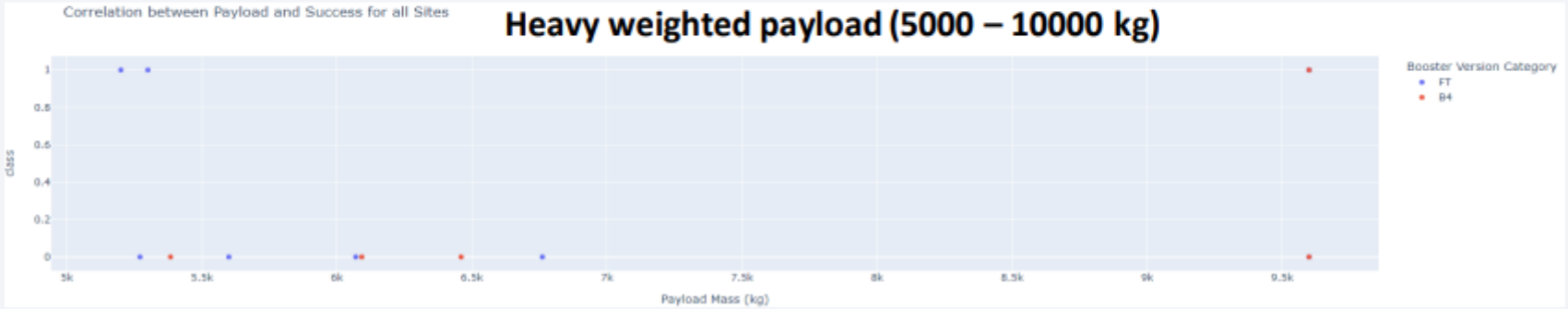
Plotly Dashboard- Total Success for KSC LC-39A



KSC LC-39A has the highest success rate so it was chosen. 1 signifies a successful launch whereas 0 signifies a failure.

Plotly Dashboard- Payload slider





Lower weighted payload give a higher success rate than heavier payloads.

Section 5

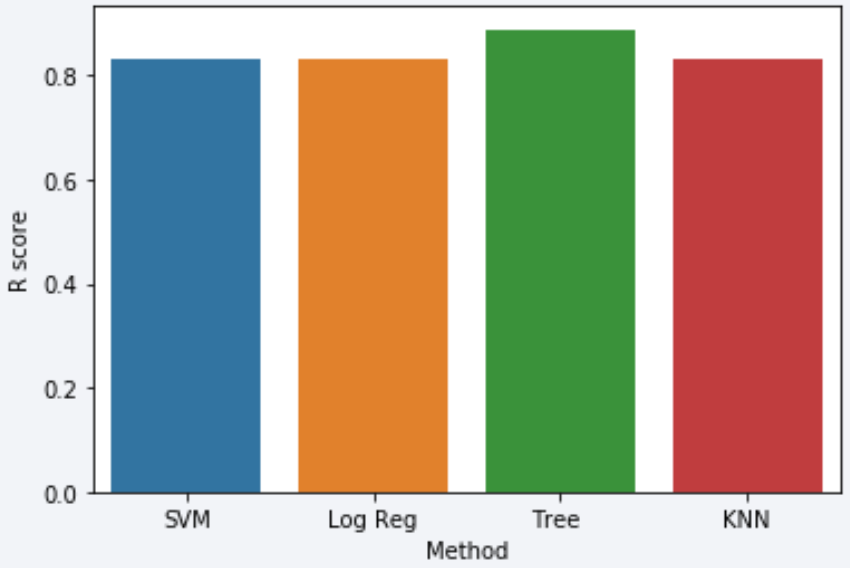
Predictive Analysis (Classification)

Classification Accuracy

Find the method performs best:

In [93]:
```
print('SVM method                 : ', svm_cv_score)
print('Logistic Regression method:' , logreg_score)
print('Tree method                : ', tree_cv_score)
print('KNN method                 : ', knn_cv_score)
print('')
print('Decision tree method works the best.')
```

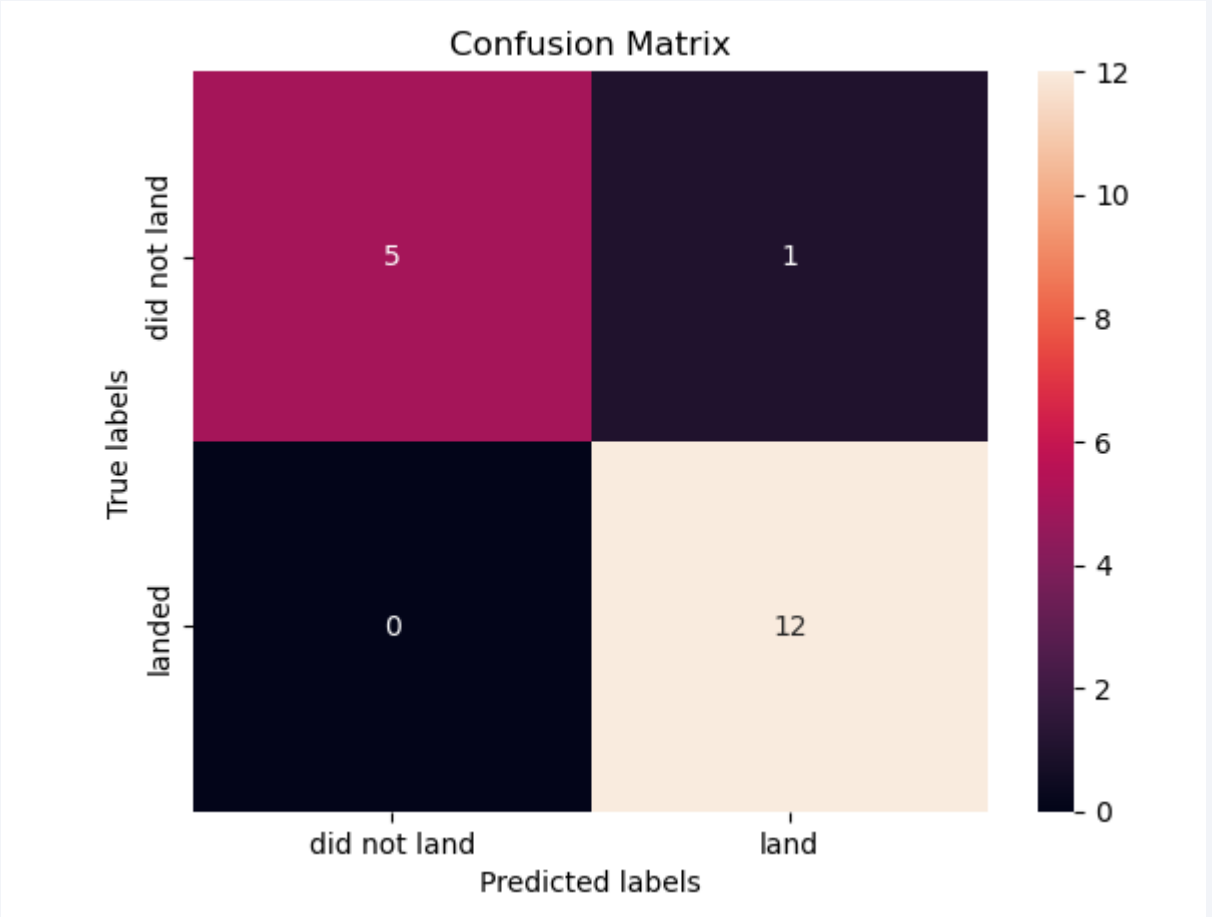
```
SVM method                 :  0.8333333333333334
Logistic Regression method: 0.8333333333333334
Tree method                :  0.9444444444444444
KNN method                 :  0.8333333333333334

Decision tree method works the best.
```



We can see the decision tree method is slightly more accurate than the other methods.

Confusion Matrix

The Decision Tree method predicted all the launches that did land correctly, and predicted 83% of launches that didn't land correctly.



Conclusions

- When launching rockets, the KSC LC-39A launch site should be chosen
- Lighter rockets have higher success rates than Heavier rockets
- A Decision tree algorithm should be used to model rocket outcomes
- Launch sites are all far from city centers but close to the sea and to highways

Thank you!