IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dilshad Yousuff
14-06-2022

Outline

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

Executive Summary

Summary Of Methodologies

- ➢ Data Collection
- ➢ Data Wrangling
- ➢ Exploratory Data Analysis (EDA) using SQL, Pandas & Matplotlib
- ➢ Interactive Visual Analytics and Dashboard
- ➢ Predictive Analysis

Summary Of Results

- ➢ Exploratory Data Analysis (EDA) Results
- ➢ Interactive Analysis
- ➢ Predictive Analysis (Identify the best classification model with highest accuracy)

Introduction

Project background and context

Most companies are making the space travel affordable to everyone. One way to do this is to ensure that the rocket launches are inexpensive. SpaceX has been the most successful in achieving this. While SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, the other providers have a cost in upwards of 165 million dollars. Much of the savings are due to the reusability of the first stage by SpaceX.

Problems you want to find answers

The objective of this project is to determine if the cost of launch by determining if the first stage of the rocket would land successfully. This information would be used by other companies to bid and compete with Space X for the rocket launch.

Section 1

Methodology

Methodology

Executive Summary

Data Collection Methodology:

- Data was gathered from SpaceX REST API, with different endpoints, specifically past launches (https://api.spacexdata.com/v4/launches/past)

- Web Scrapping to collect historical Falcon 9 launch data from Wikipedia (https://en.Wikipedia.org/wiki/list_of_falcon_9_and_falcon_heavy_launches)

Perform Data Wrangling

- The response from the API is a json file which is normalized into a flat table and convert into a pandas data frame.

- Using Python Beautiful Soup package to web scrape the html tables and parse the data in them to convert into the pandas data frame for visualization and analysis

Methodology

Executive Summary

Perform Exploratory Data Analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analytics using classification model

- ➢ Split the clean and normalized data into train and test sets
- ➢ Build and evaluate different classification models like –
    - ✓ Logistic Regression
    - ✓ Support Vector Machine
    - ✓ K-Nearest Neighbor
    - ✓ Decision Tree

Data Collection

- ➢ Data for the SpaceX launces is collected from the SpaceX REST API (https://api.spacexdata.com/v4/launches/past)
- ➢ The data includes information about…
  - ✓ Rocket Used
  - ✓ Payload Delivered
  - ✓ Launch Specifications
  - ✓ Landing Specifications
  - ✓ Landing Outcomes
- ➢ Other API endpoints are targeted to gather specific data like..
  - ✓ Booster Version - https://api.spacexdata.com/v4/rockets
  - ✓ Launch Site- https://api.spacexdata.com/v4/launchpads
  - ✓ Payload Data - https://api.spacexdata.com/v4/payloads
  - ✓ Core Data - https://api.spacexdata.com/v4/cores
- ➢ Web Scrapping from Wikipedia

Data Collection - SpaceX API

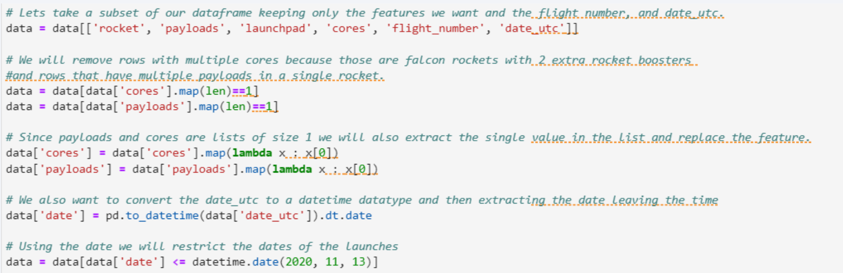
Request launch data from API and convert the response into a static json

spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
response = requests.get(static_json_url)

***static_json_url is a result of data from a URL*

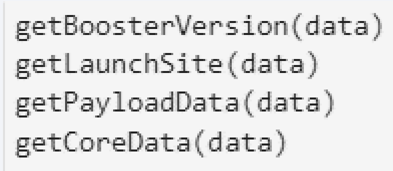
Normalize the .json file and convert into a pandas dataframe
data = pd.json_normalize(response.json())

Take a subset of data with required features and clean the dataset

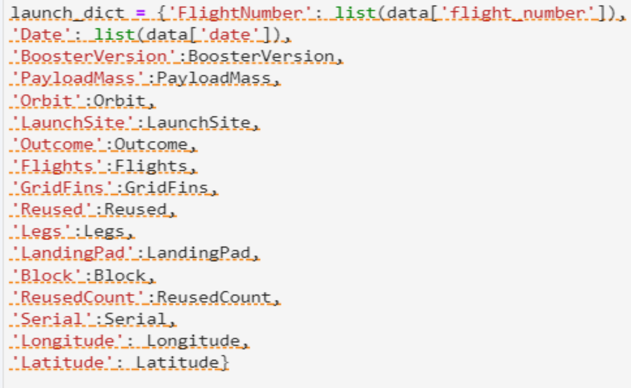


```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters
#and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Get Additional Information using functions calling the different API endpoints & combine results into a list



```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```



```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Created a Dataframe from the results and filter to include data for Falcon 9 only

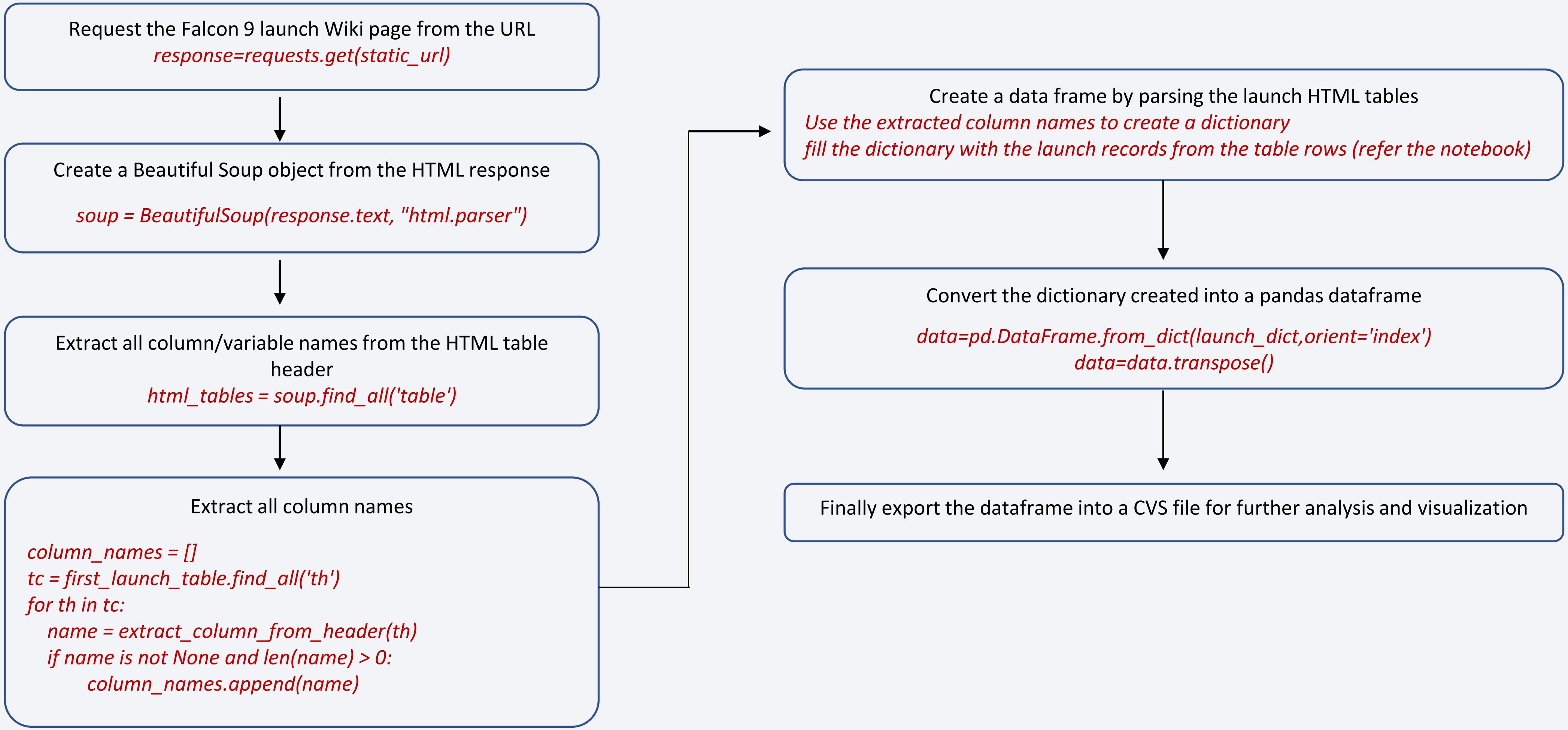
data2 = pd.DataFrame(launch_dict)
data_falcon9 = data2[data2['BoosterVersion']!='Falcon 1']

Finally perform data wrangling to identify any missing values and replace them with the mean() values

Once the data is clean, export the data into CSV format for further analysis and visualization

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20Data%20Collection.ipynb

Data Collection - Scraping



Request the Falcon 9 launch Wiki page from the URL

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

Create a Beautiful Soup object from the HTML response

```
soup = BeautifulSoup(response.text, "html.parser")
```

Extract all column/variable names from the HTML table header

```
html_tables = soup.find_all('table')
```

Extract all column names

```
column_names = []
tc = first_launch_table.find_all('th')
for th in tc:
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Create a data frame by parsing the launch HTML tables

Use the extracted column names to create a dictionary

fill the dictionary with the launch records from the table rows (refer the notebook)

Convert the dictionary created into a pandas dataframe

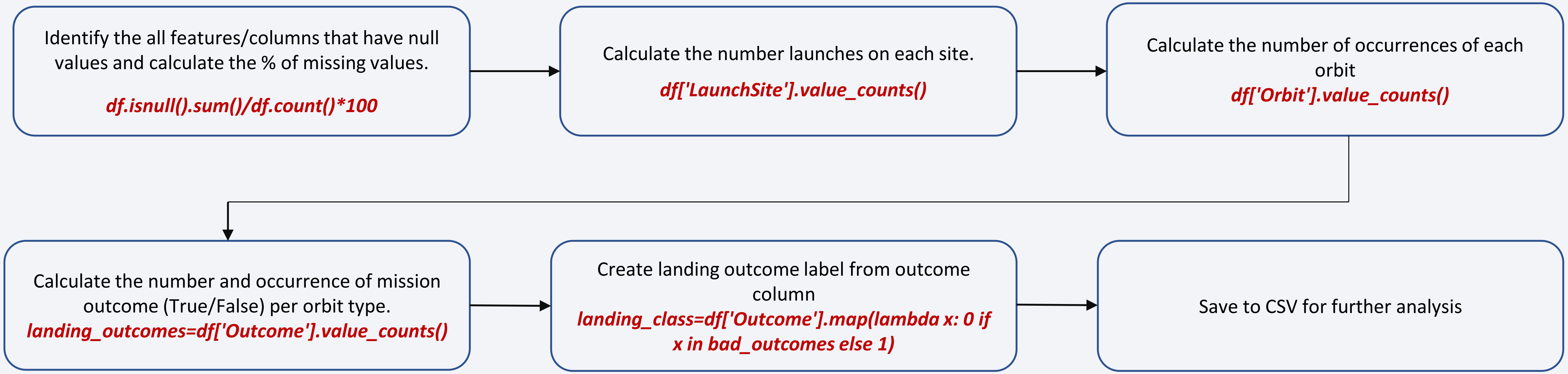
```
data=pd.DataFrame.from_dict(launch_dict,orient='index')
data=data.transpose()
```

Finally export the dataframe into a CVS file for further analysis and visualization

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20WebScrapping.ipynb

Data Wrangling

In this project, the data wrangling is done to convert the outcomes of the space missions into training labels with 1 as a Successful landing and 0 for unsuccessful landing.



Identify the all features/columns that have null values and calculate the % of missing values.

df.isnull().sum()/df.count()*100

Calculate the number launches on each site.

df['LaunchSite'].value_counts()

Calculate the number of occurrences of each orbit

df['Orbit'].value_counts()

Calculate the number and occurrence of mission outcome (True/False) per orbit type.

landing_outcomes=df['Outcome'].value_counts()

Create landing outcome label from outcome column

landing_class=df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)

Save to CSV for further analysis

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20Data%20Wrangling.ipynb

EDA with Data Visualization

Scatter Plots to visualize relationship between different feature:

- Flight Number & Launch Site:
- Payload & Launch Site
- Flight Number & Orbit Type
- Payload & Orbit Type

Bar Chart to plot the success rate of each Orbit Type

- Orbit & Mean of success rate

Line Chart to plot the trend of the average success rate

- Success rate trend over the years 2010 +

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20EDA%20with%20Pandas%20%26%20Matplotlib.ipynb

EDA with SQL

SQL Queries used for Exploratory Data Analysis:

1. Display the names of the unique launch sites in the space mission (*using the DISTINCT command*)
2. Display 5 records where the launch site had a specific text in the name (*using LIKE in where clause and LIMIT 5*)
3. Display the total payload mass carried by boosters launched by NASA (CRS) (*using SUM function in select statement*)
4. Display average payload mass carried by booster version F9 v1.1 (*using AVG function in select statement*)
5. List the date when the first successful landing outcome in ground pad was achieved (*using the MIN function in select statement*)
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7. List the total number of successful and failure mission outcomes (*using COUNT function and GROUP BY*)
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery (*using a SELECT statement to get the max payload as a sub query*)
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 (*using an AND function in the where clause*)
10. Rank the landing outcomes between given date ranges, based on the count in descending order (*using the COUNT, GROUP BY & ORDER BY*)

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

One objective of the project is to identify the optimal location for a launch site.

- Mark the existing launch sites on a map
- Add success/failure launches per site to the map
- Calculate the distance between the launch site and its close proximities like coast, railway station etc.

To mark all the above, we use…

- **Folium Circles**, to mark the launch sites
- **Folium Markers / Marker Cluster,** to mark the launch outcomes (0/1) for each site
- **Mouse Position,** to easily get the coordinates of any points of interest
- **PolyLine**, to draw a line marking the distance from the launch site to the point of interest

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Space%20X%20-%20Data%20Visualization%20with%20Folium.ipynb

GitHub Link (Screenshots): https://github.com/yousuffd/Data-Science---Projects-Coursera-/tree/master/Screenshots/Folium

Build a Dashboard with Plotly Dash

Dashboard has following features

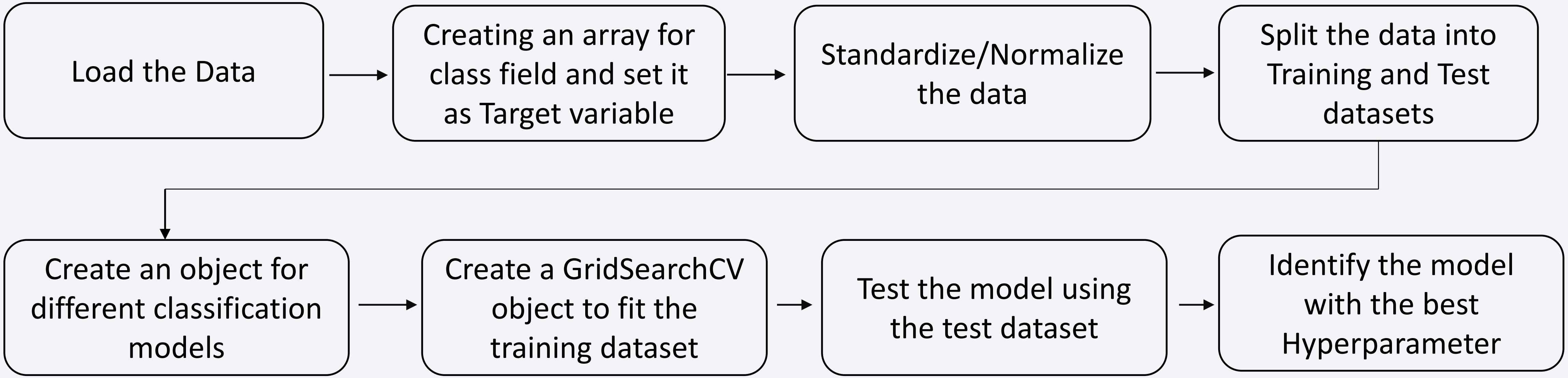
- ➢ Pie Chart showing the success rate for each of the launch sites
- ➢ Dropdown list to select the launch sites and view the detailed success rate
- ➢ Callback function to render the success rate pie chart based on the value selected in dropdown
- ➢ Scatter Plot to show the correlation of Payload to the mission outcome
- ➢ Range Slider to select the Payload
- ➢ Callback function to render the scatter plot based on the Payload Selection

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/SpaceX%20Dash%20App.py

GitHub Link (Screenshots): https://github.com/yousuffd/Data-Science---Projects-Coursera-/tree/master/Screenshots/Folium

Predictive Analysis (Classification)

The objective of the Predictive Analysis is to build and test a best performing classification model. Models built and tested include – Logistic Regression, K-Nearest Neighbor, Decision Tress & Support Vector Machine



Load the Data

Creating an array for class field and set it as Target variable

Standardize/Normalize the data

Split the data into Training and Test datasets

Create an object for different classification models

Create a GridSearchCV object to fit the training dataset

Test the model using the test dataset

Identify the model with the best Hyperparameter

GitHub Link: https://github.com/yousuffd/Data-Science---Projects-Coursera-/blob/master/Capstone%20Project%20-%20Machine%20Learning%20Prediction.ipynb

Results

Exploratory data analysis results

- Landing outcome on drone ships has been the most successful as opposed to the others
- Drone ships have only had two failed landing outcomes
- Most mission outcomes (over 98%) have been successful
- The first successful landing was towards the end of the year 2015
- F9 Boosters are most successful with a maximum payload
- Average payload on the F9 boosters is about 2534 Kg
- The correlation between the LEO orbit and the
- Heavy payload launches cannot be done from all the launch sites
- The success rate of the launches seems to have increased over the year

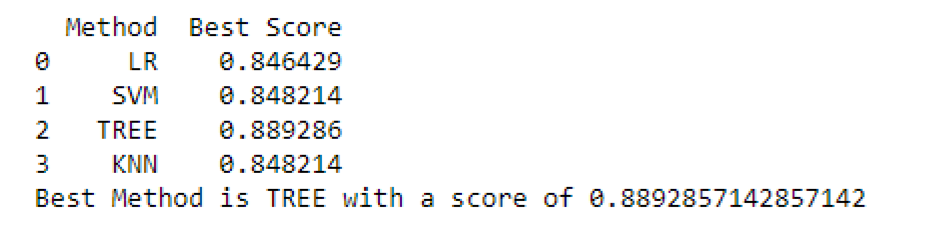
Results

Interactive analytics demo in screenshots

- ➢ Coastal areas are preferred locations to set up the launch sites, specifically areas with a proximity to the basic infrastructure like the railway station
- ➢ Most launches are done at the east coast launch sites (CCFAS SLC 40), owing to the proximity to the equator

Predictive analysis results

- ➢ Of all the classification models built and tested, Decision Tree was found to be the best model with a best score of 88.92%

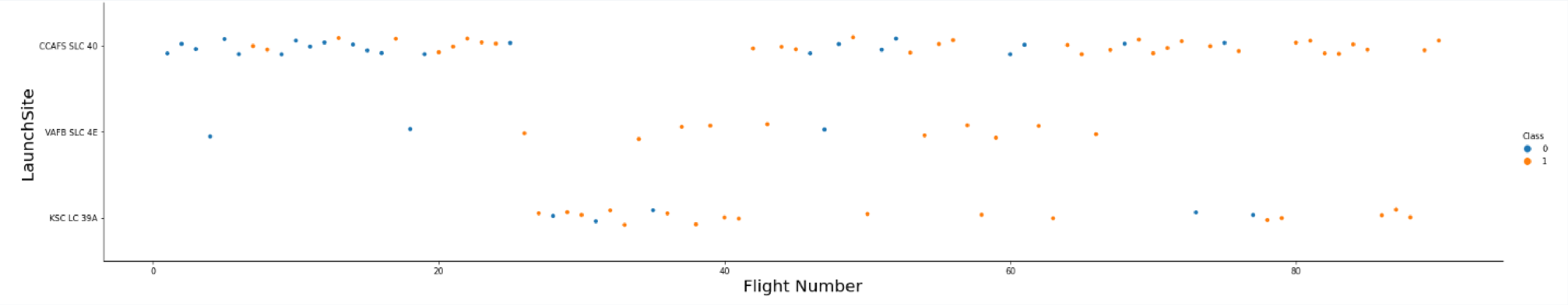


```
  Method  Best Score
0     LR    0.846429
1    SVM    0.848214
2   TREE    0.889286
3    KNN    0.848214
Best Method is TREE with a score of 0.8892857142857142
```

Section 2

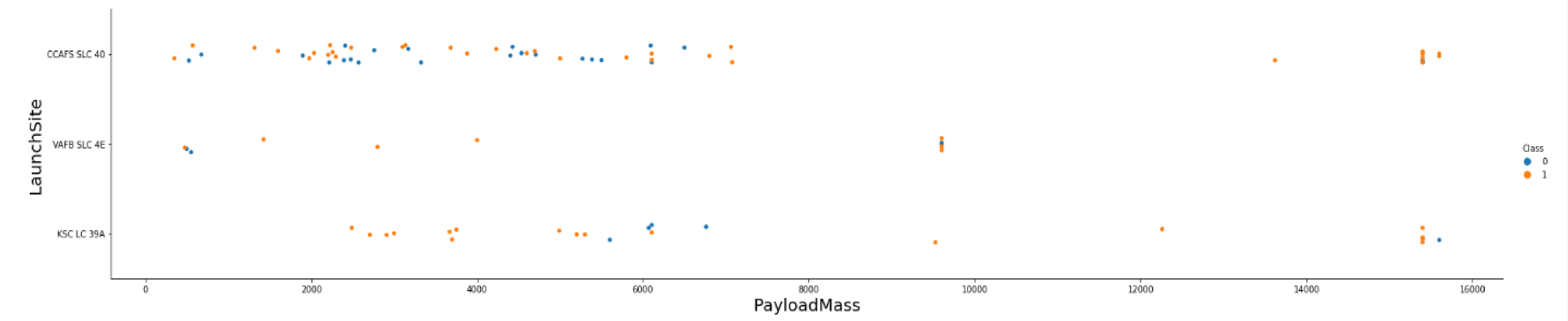
Insights drawn from EDA

Flight Number vs. Launch Site



✓ The success rate of the launches from the launch site CCAF5 SLC 40 has been higher compared to other launch sites

Payload vs. Launch Site

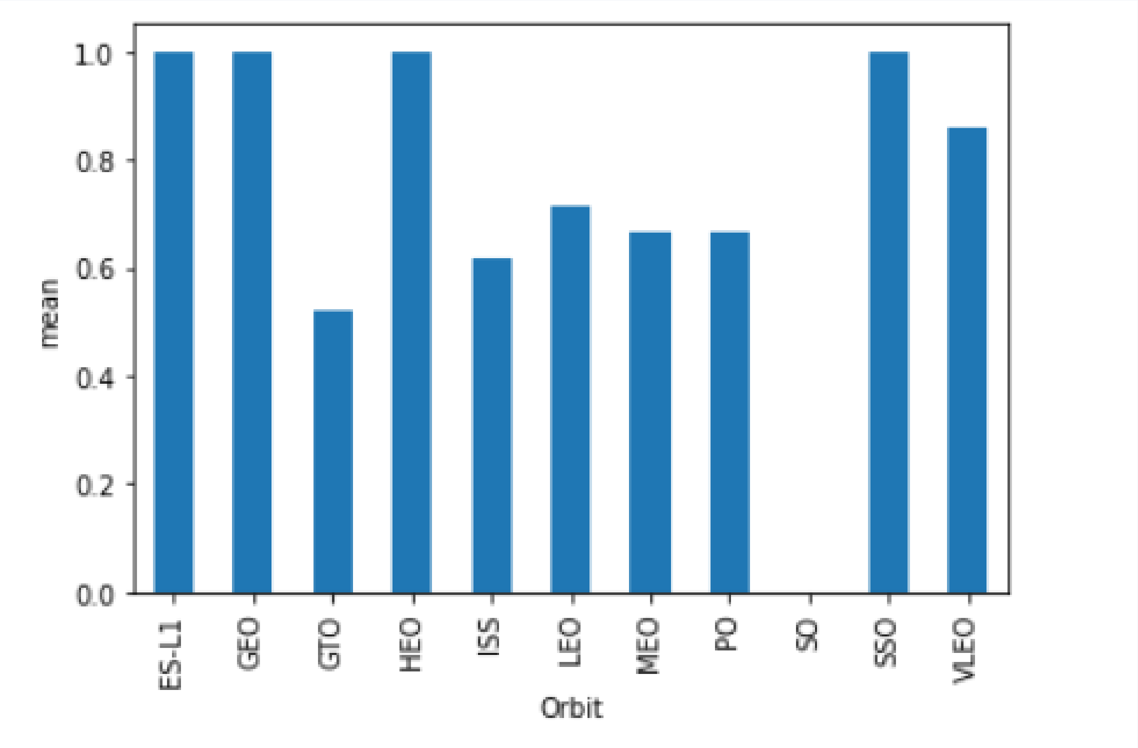


✓ Launches with a smaller payloads have a better success rate

✓ Payloads of over 10000 Kg are launched mainly from either CCAFS SLC 40 or KSC LC 39A

✓ VAFB SLC 4E location is usually not used for heavy payload launches

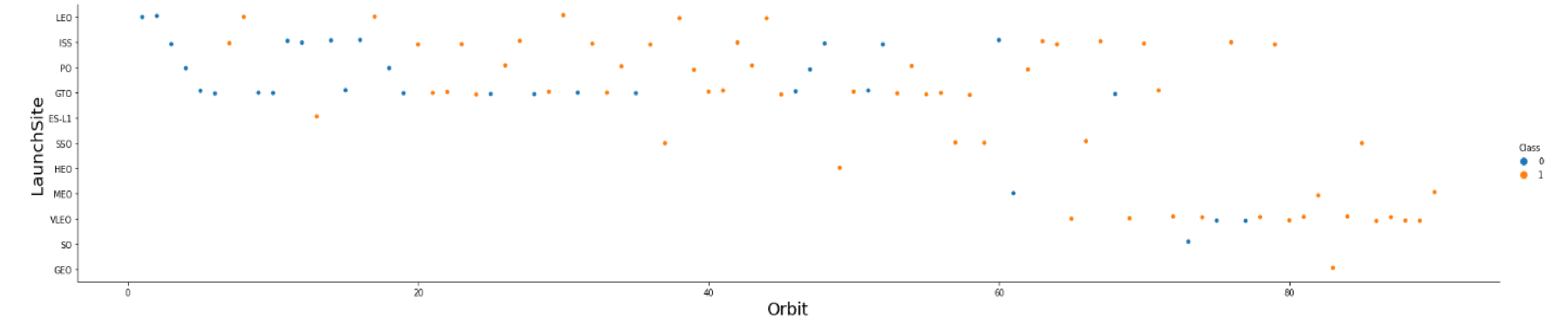
Success Rate vs. Orbit Type



It is observed that the launches to the following orbits have a higher average success rate compared to the others.

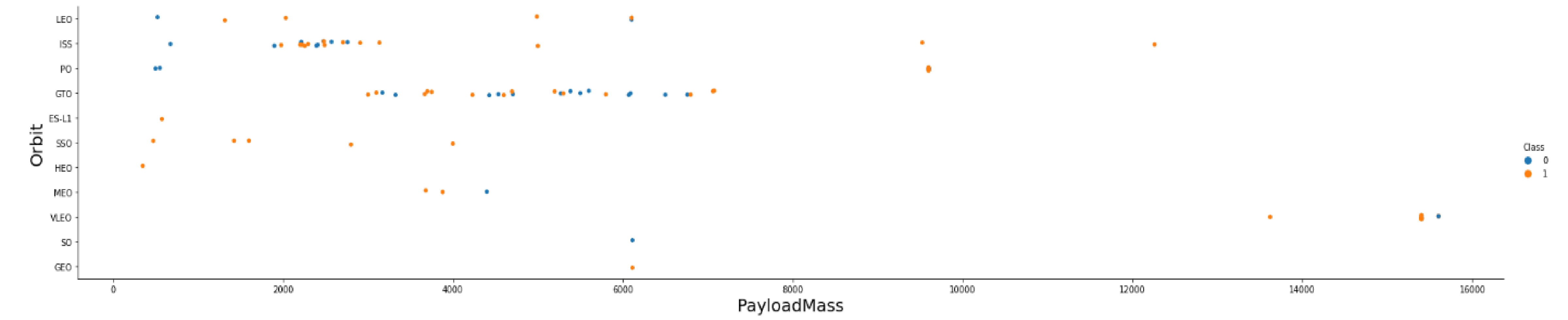
- ✓ ES-L1
- ✓ GEO
- ✓ HEO
- ✓ SSO

Flight Number vs. Orbit Type



- ✓ The launches have appeared to stabilize across all the obits
- ✓ The frequency of launches into the VLEO has increased

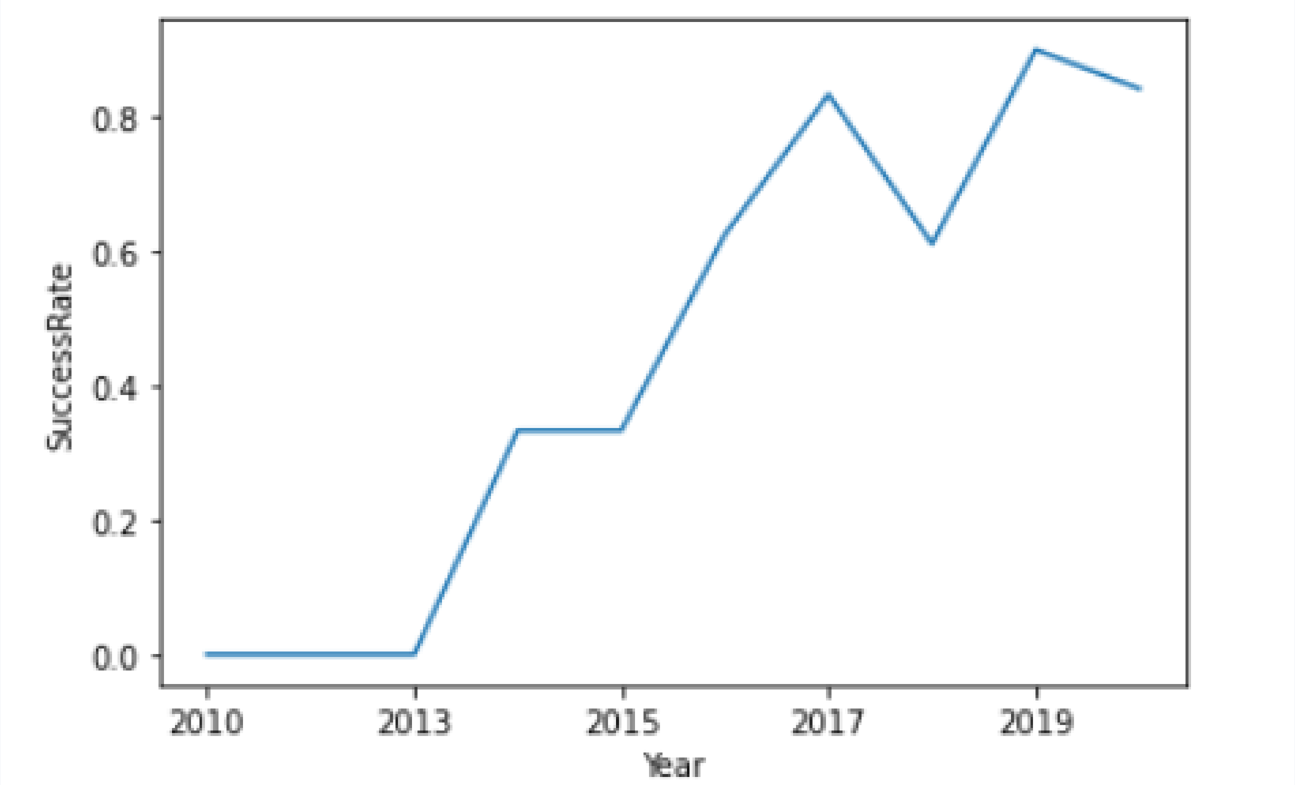
Payload vs. Orbit Type



✓ The correlation between the payload (when it is between 2000 & 4000 Kg), is highest when the launch is for the orbit ISS

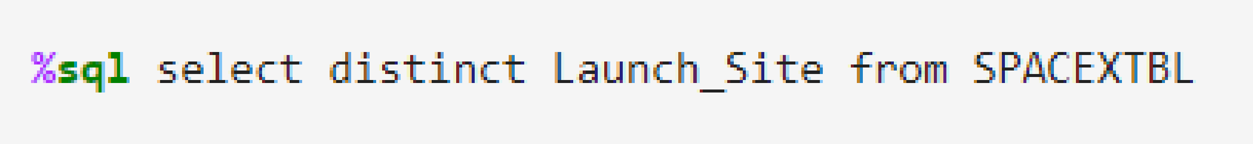
✓ The correlation between the payload (when it is between 3000 & 9000 Kg), is highest when the launch is for the orbit GTO

Launch Success Yearly Trend

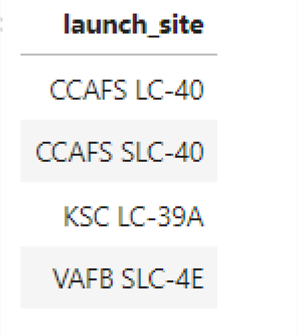


- ✓ The success has constantly increased until 2018 which is when it a dip compared to the prior year.

All Launch Site Names



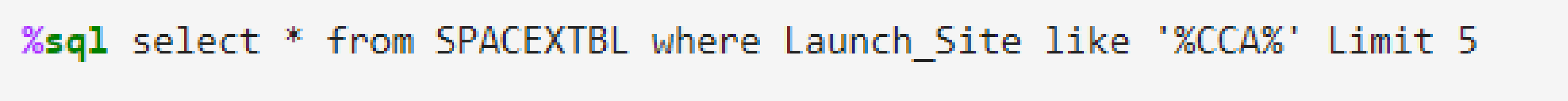
```
%sql select distinct Launch_Site from SPACEXTBL
```



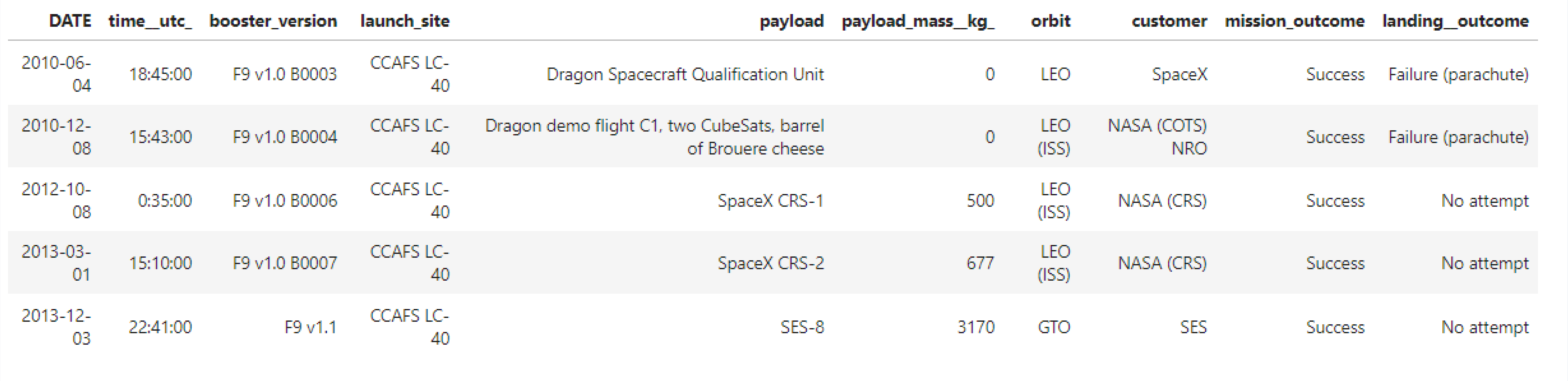
launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Query pulls all distinct/unique values in the Launch Site column of the SpaceX data (Table name: SPACEXTBL)

Launch Site Names Begin with 'CCA'



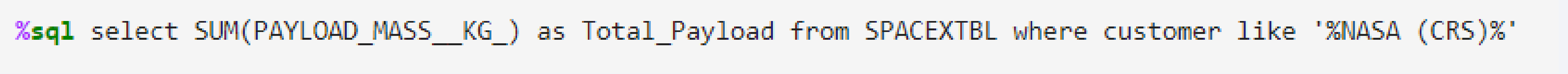
```
%sql select * from SPACEXTBL where Launch_Site like '%CCA%' Limit 5
```



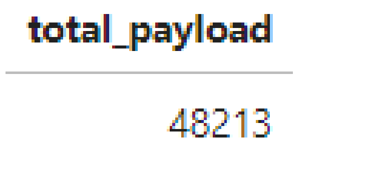
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-10-08	0:35:00	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
2013-12-03	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Query pulls the data from the SPACEXTBL, where the Launch Site consists/contains ‘CCA’ in its name. This also limits the number of records pulled to just 5

Total Payload Mass



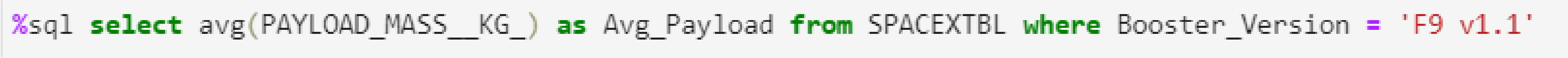
```
%sql select SUM(PAYLOAD_MASS__KG_) as Total_Payload from SPACEXTBL where customer like '%NASA (CRS)%'
```



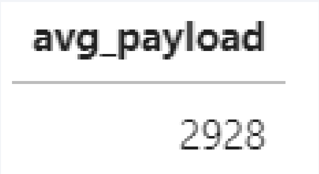
total_payload
48213

Uses the SUM function to calculate the total payload mass for the customer NASA (CRS) and names the field as total_payload

Average Payload Mass by F9 v1.1



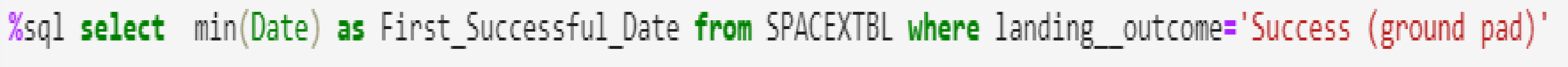
```
%sql select avg(PAYLOAD_MASS__KG_) as Avg_Payload from SPACEXTBL where Booster_Version = 'F9 v1.1'
```



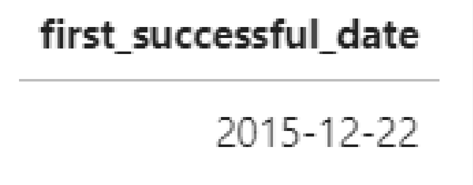
avg_payload
2928

Uses the AVG function and calculates the average payload mass for the booster version = F9 v1.1

First Successful Ground Landing Date



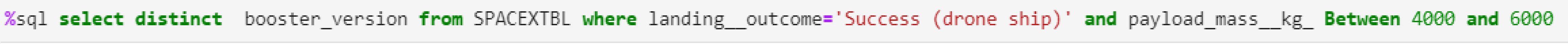
```
%sql select  min(Date) as First_Successful_Date from SPACEXTBL where landing__outcome='Success (ground pad)'
```



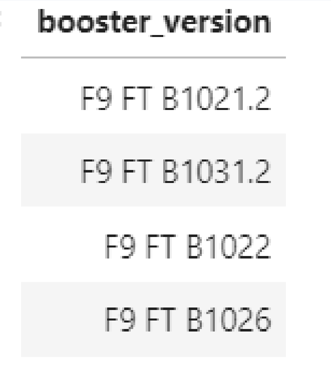
first_successful_date
2015-12-22

Query uses the MIN function to find the first date on which the first successful landing was done on the ground pad

Successful Drone Ship Landing with Payload between 4000 and 6000



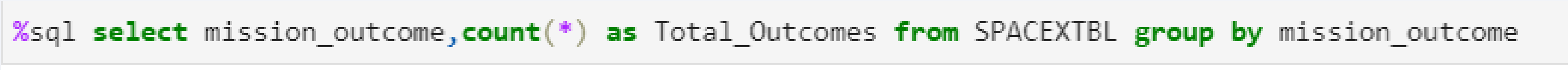
```
%sql select distinct  booster_version from SPACEXTBL where landing__outcome='Success (drone ship)' and payload_mass__kg_ Between 4000 and 6000
```



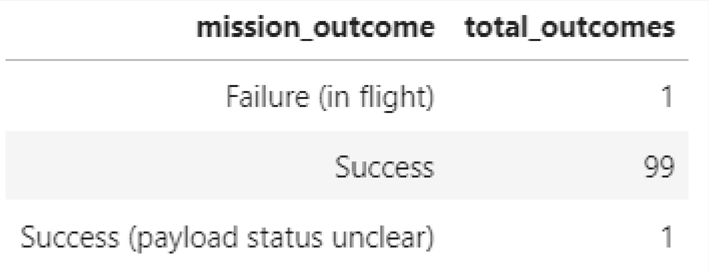
booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

Query pull the distinct booster versions, which had a successful landing on drone ship and had a payload mass between the range of 4000 Kg to 6000 Kg

Total Number of Successful and Failure Mission Outcomes



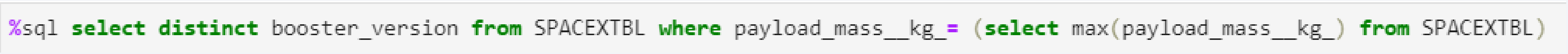
```
%sql select mission_outcome,count(*) as Total_Outcomes from SPACEXTBL group by mission_outcome
```



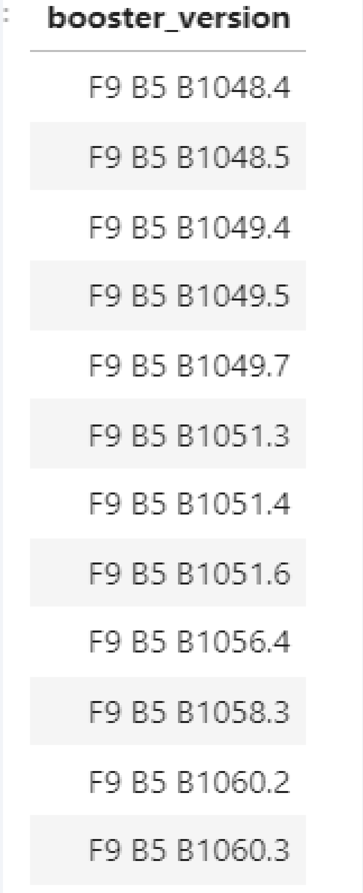
mission_outcome	total_outcomes
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Query pull the list of mission outcomes and the number count per outcome. This used the Count function and groups by the mission outcome field

Boosters Carried Maximum Payload



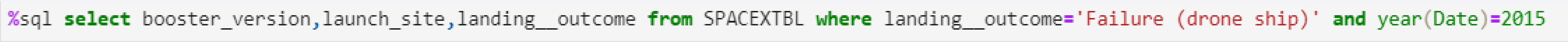
```
%sql select distinct booster_version from SPACEXTBL where payload_mass__kg_= (select max(payload_mass__kg_) from SPACEXTBL)
```



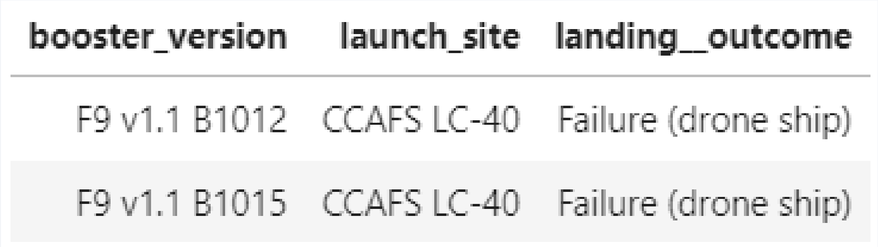
booster_version
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

Query pulls a list of distinct booster version where the total payload is equal to the maximum payload. A sub query using MAX function is used as a filter in the where clause of the main query

2015 Launch Records



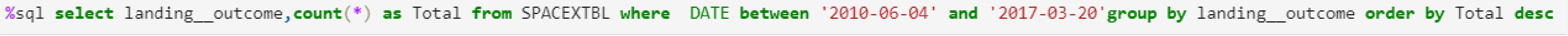
```
%sql select booster_version,launch_site,landing__outcome from SPACEXTBL where landing__outcome='Failure (drone ship)' and year(Date)=2015
```



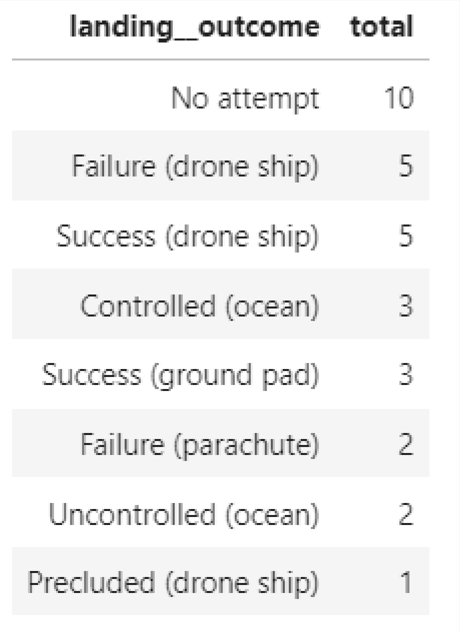
booster_version	launch_site	landing__outcome
F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Query pulls data from specific fields/columns from the SPACEXTBL, where the landing outcome was failure and in the year 2015. Query uses AND criteria in the where clause, which pull the data only when both the criterions (landing Outcome and Year) match

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



```
%sql select landing__outcome,count(*) as Total from SPACEXTBL where  DATE between '2010-06-04' and '2017-03-20'group by landing__outcome order by Total desc
```



landing__outcome	total
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

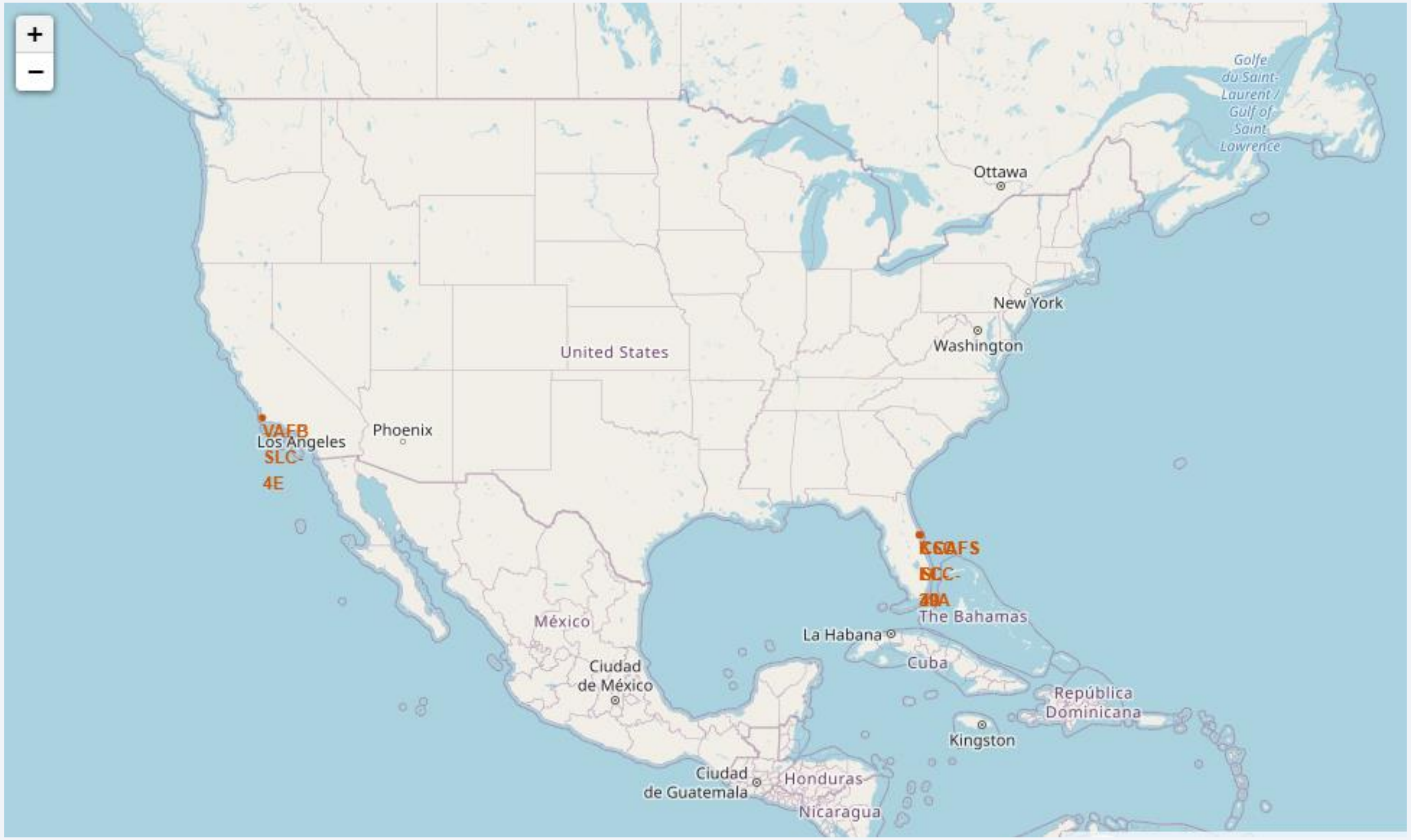
Query pulls the total count for each landing outcome (using GROUP BY) for the given date range and then sorts the result in descending order with the landing outcome with a max count at the top

Notice that the No Attempt has the highest count and hence needs to be taken into consideration

Section 3

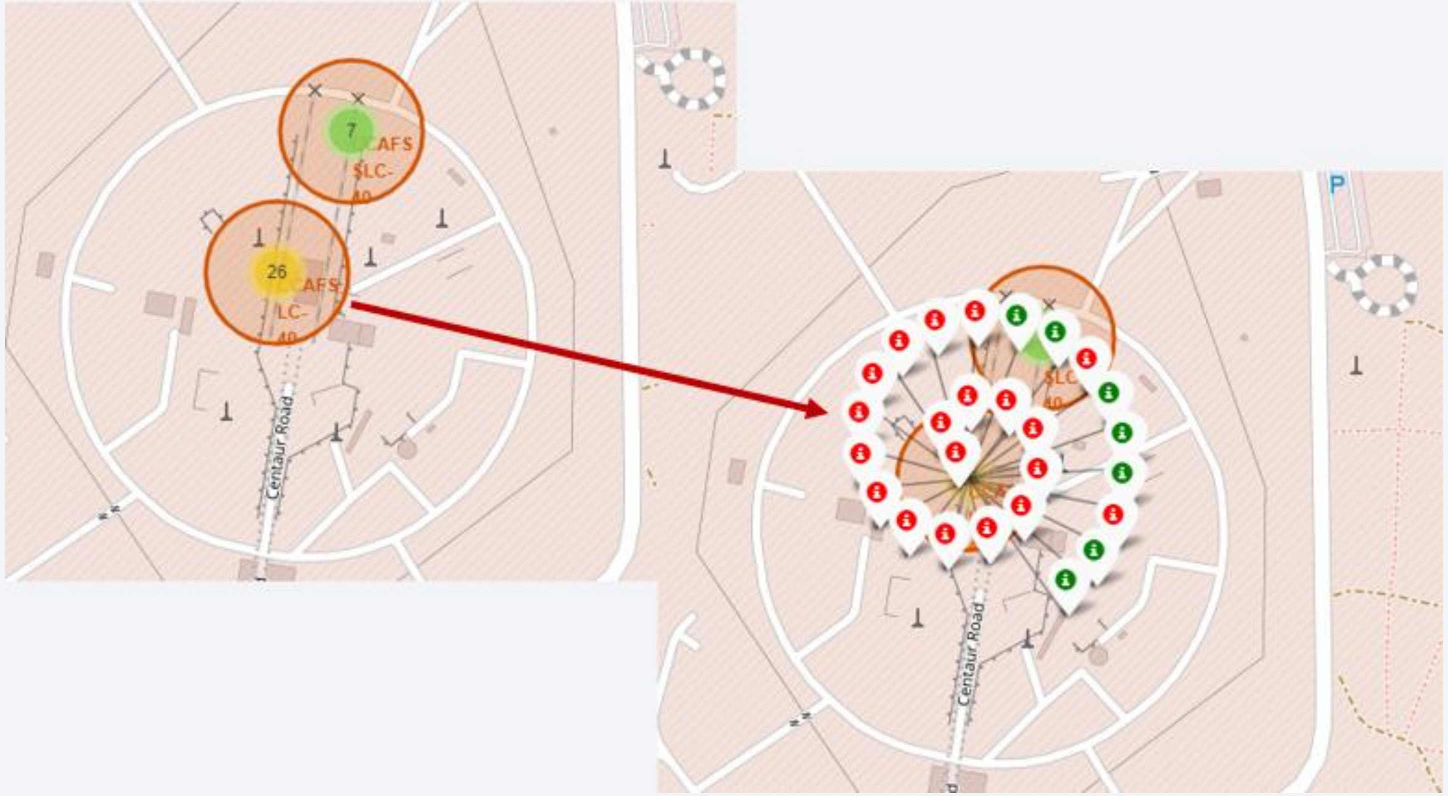
Launch Sites Proximities Analysis

Launch Sites - Location



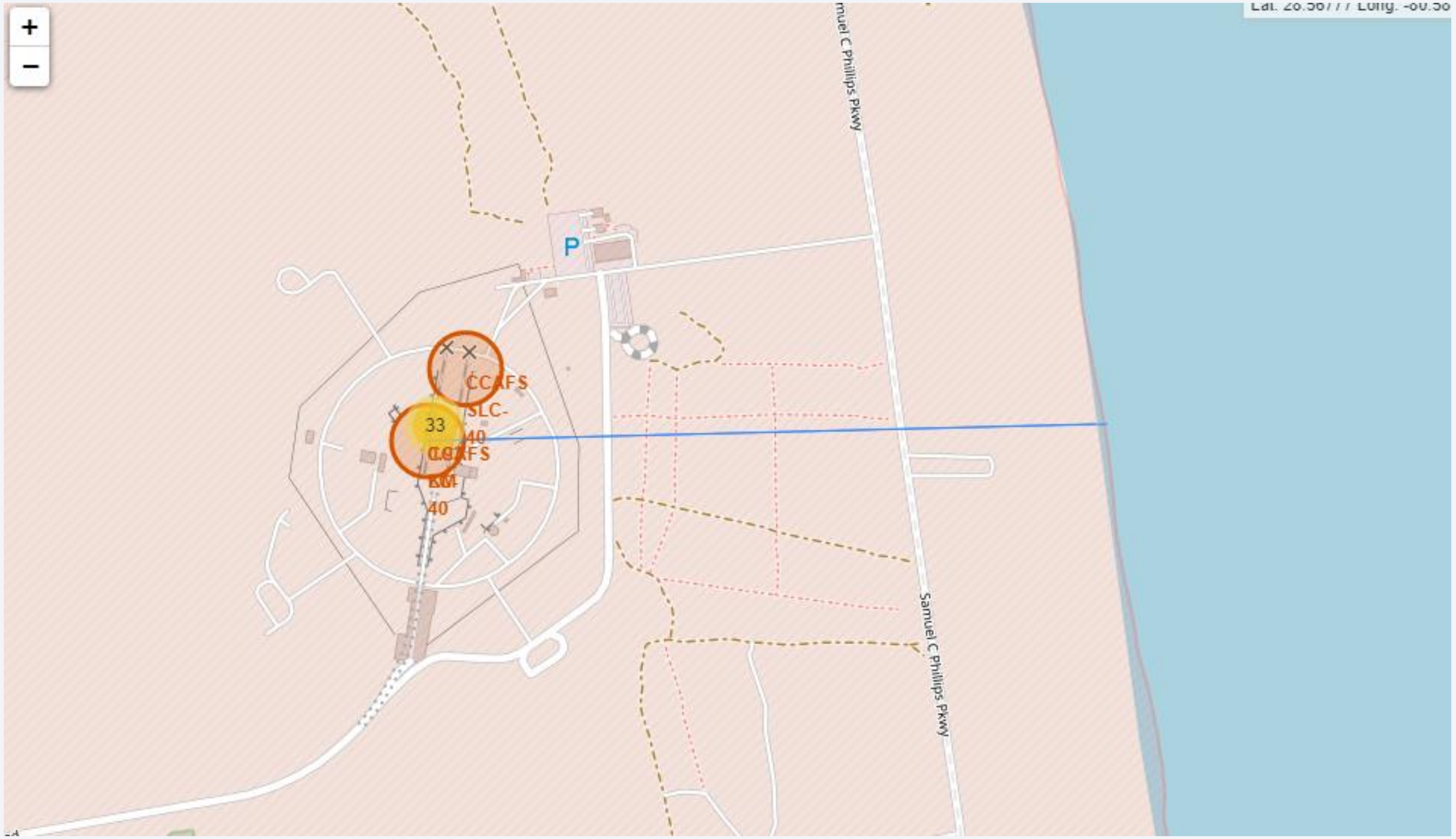
All launch sites are in a proximity to the coastal area and the equator

Launch Sites - Landing Outcomes

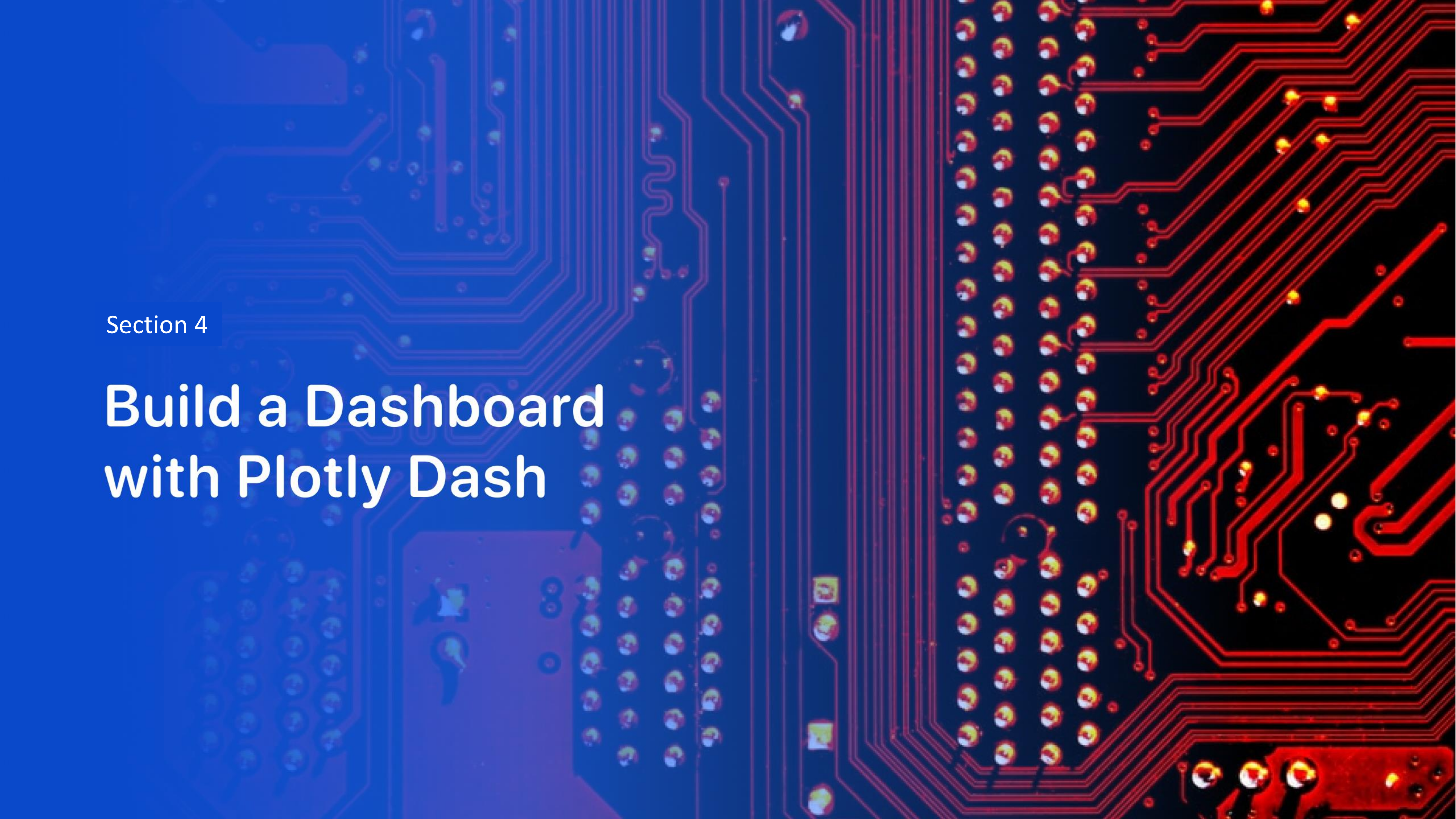


- Figure indicates the location for the CCAFS SLC 40 launch site with the marker showing the successful and failed launches.
- The successful launches are marked with a “Green” marker while those failed are marked with a “Red” marker

Launch Sites – Proximity to Point of Interest



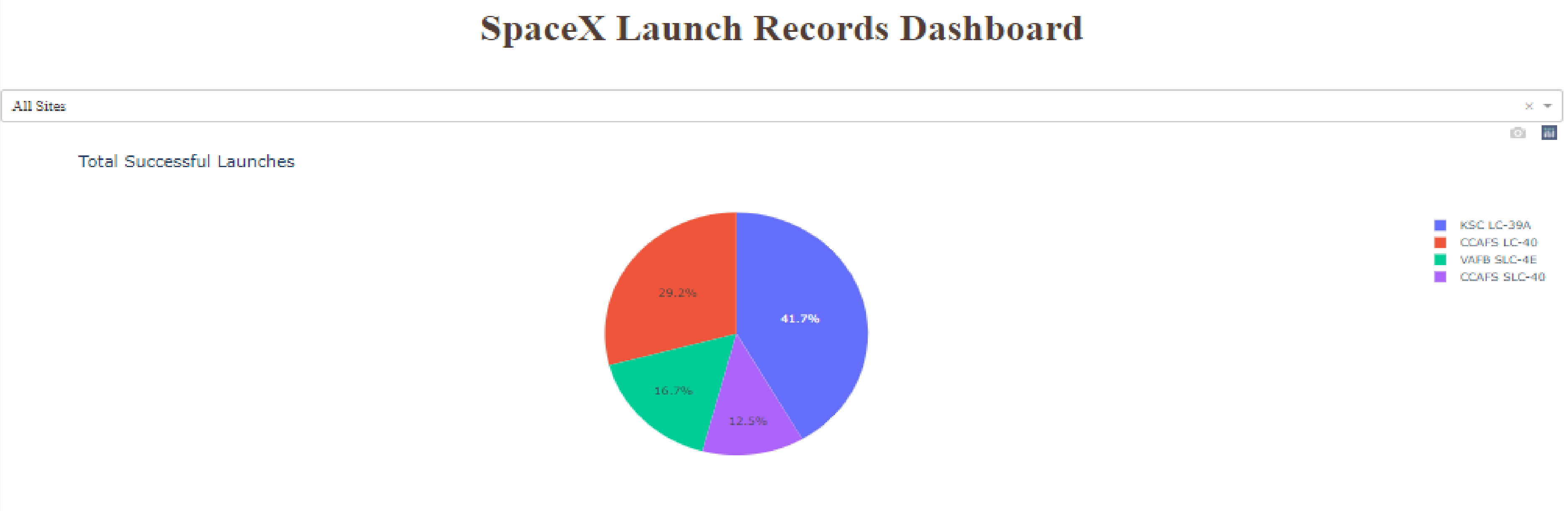
- ✓ The view has location marked for the CCAFS SLC 40 launch site indicated by the yellow circle
- ✓ The blue line marks the distance between the launch site in this case CCAFS SLC 40 and the closest coastline.



Section 4

Build a Dashboard with Plotly Dash

Launch Sites – Successful Rates



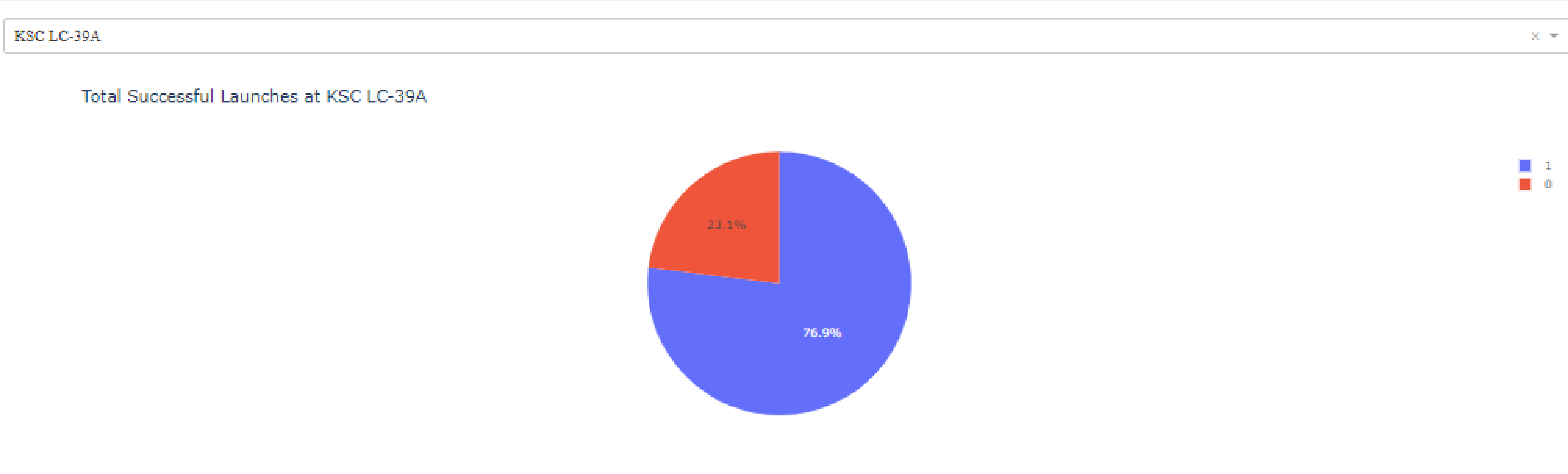
SpaceX Launch Records Dashboard

All Sites

✓ The Pie chart shows the Success rate for ach of the four launch sites

✓ The KSC LC – 39A launch site has the best success rate with 41.7%

Best Launch Site, KSC LC – 39A

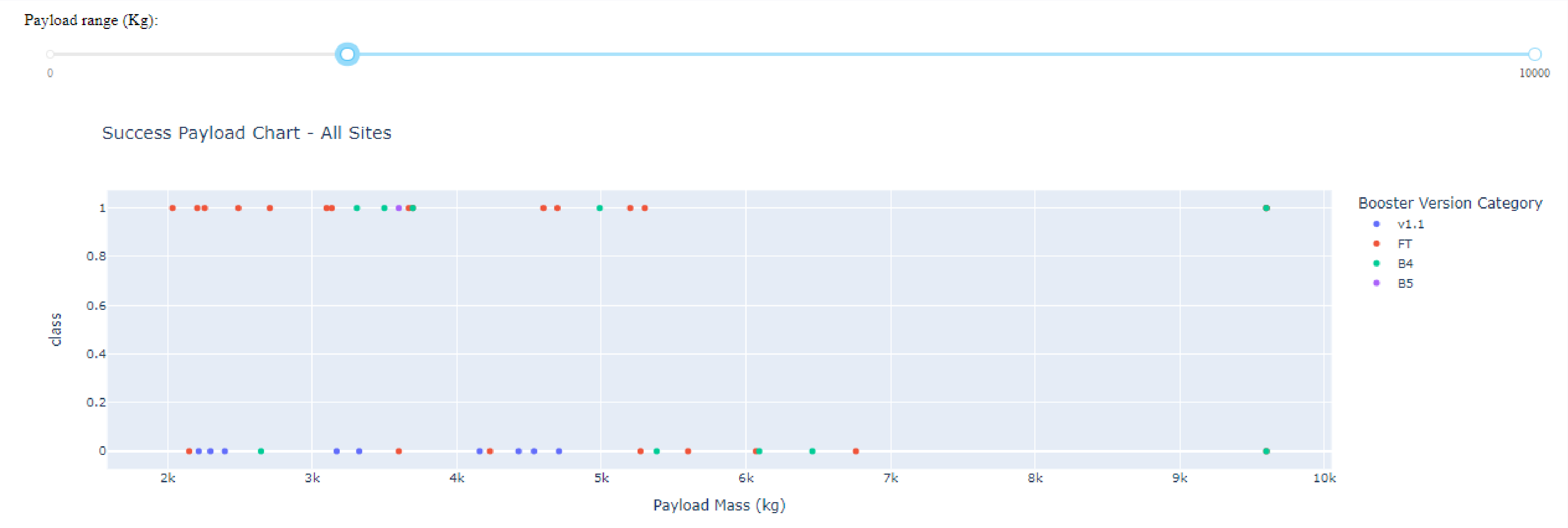


KSC LC-39A

Total Successful Launches at KSC LC-39A

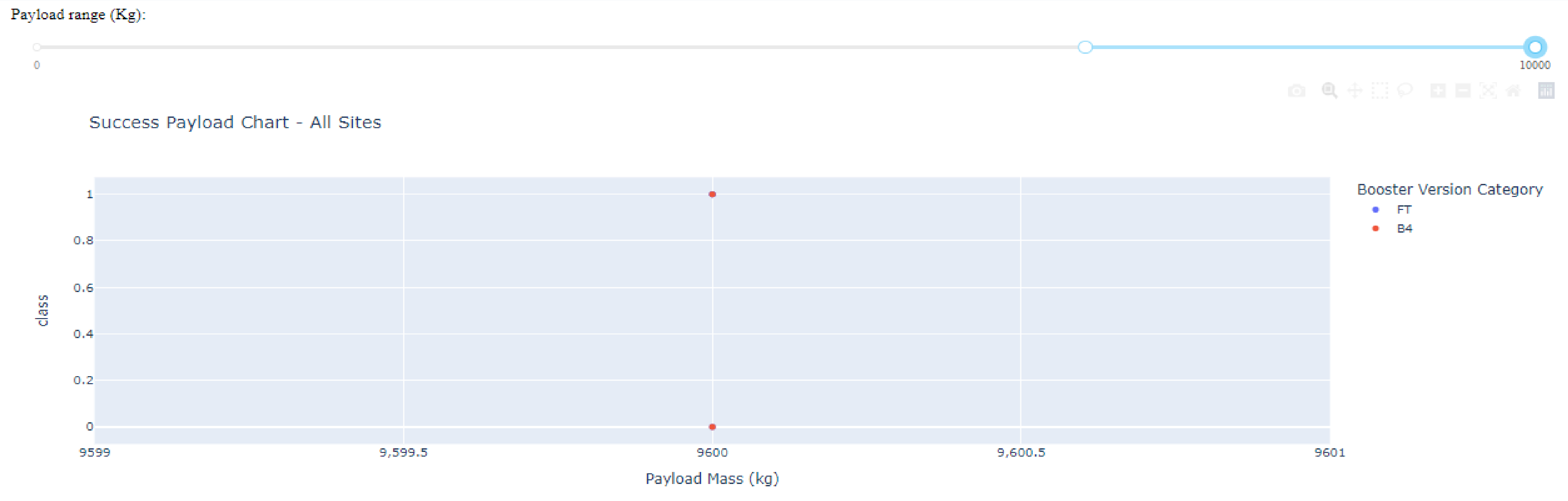
- ✓ The Pie chart shows the successful vs failed launch %s for the site that had the highest success rate (KSC LC – 39A) among all sites
- ✓ 76.9% of the launches are successful at this site

Best Launch Site, KSC LC – 39A



- ✓ The launches with payloads of less than 6000 Kg have a higher success rate
- ✓ The Booster Version FT is most successful when the payload is less than 6000 Kg

Best Launch Site, KSC LC – 39A



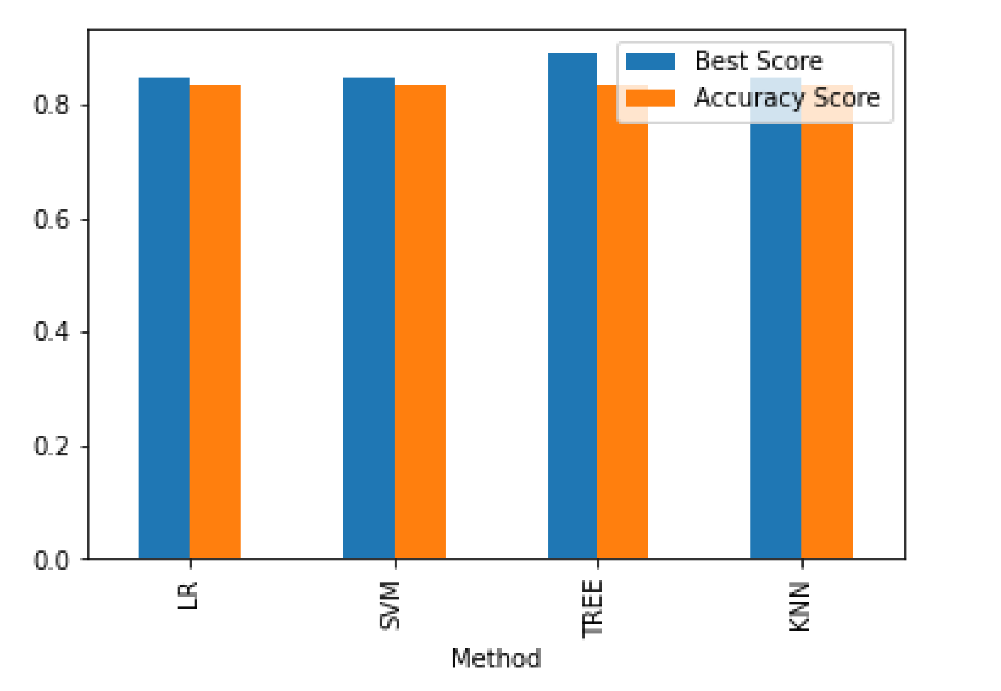
Payload range (Kg):

✓ There is not enough data available to analyze the landing outcomes for booster versions with a payload of >7000 Kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

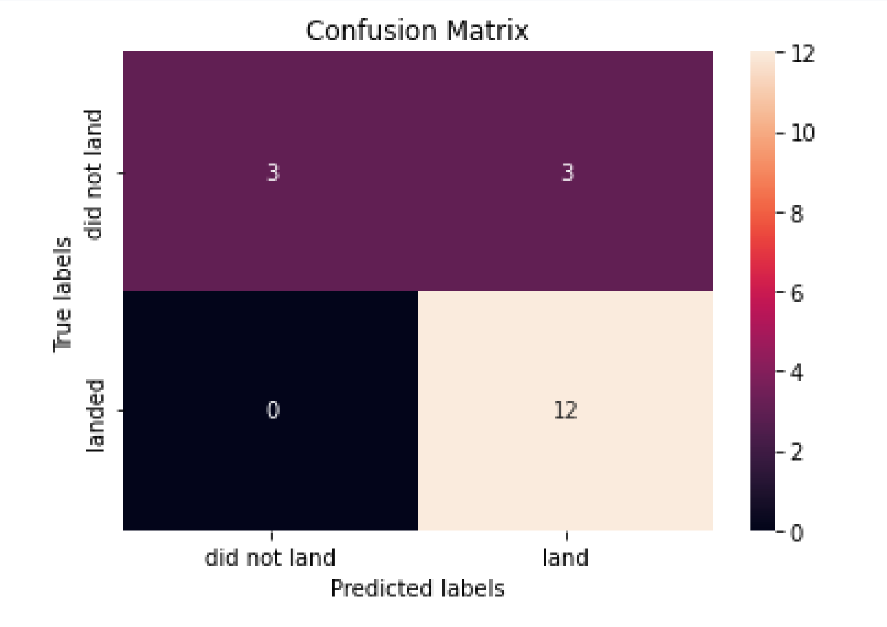


Classification models tested

- ✓ LR (Logistic Regression)
- ✓ SVM (Support Vector Machine)
- ✓ TREE (Decision Tree)
- ✓ KNN (K-Nearest Neighbor)

Decision Tree was the best model with an accuracy of 88.92%

Confusion Matrix



Confusion Matrix of the Decision Tree model shows majority of landing outcome to be successful (True Positive) versus very few in the failures (True Negative)

Conclusions

- Most preferred locations to set up the launching sites are the coastal areas specifically with proximity to the equator would be better
- Among the launch sites considered, KSC LC-39A was the site with highest success rate
- Most of the launches done with drone ship as the landing pad, are success as compared to say the ground landing pad
- Launches done to certain orbits like GEO, ES L1, HEO and SSO have had maximum successful outcomes
- The lower the payload, the higher would be the success rate
- Of all the classification models consider, Decision Tree was the best to predict the landing outcome and hence the cost of the launch

Thank you!