IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Natalie Gunn

September 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using API, SQL and Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL and Data Visualization
  - Interactive Visual Analytics with Folium
  - Predictive Analysis with Machine Learning Models
- Summary of all results
  - Data Analysis
  - Interactive Visualizations
  - Predictive Analysis

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is due to reuse of the first stage. If a determination can be made on the successful landing of the first stage, then the cost of the launch can be determined. This information can be used by other companies to bid against SpaceX for rocket launches.
- Problems you want to find answers
  - Will SpaceX's first stage rocket launch land successfully?
  - What is the price of each launch?
  - Will SpaceX reuse the first stage?

Section 1

Methodology

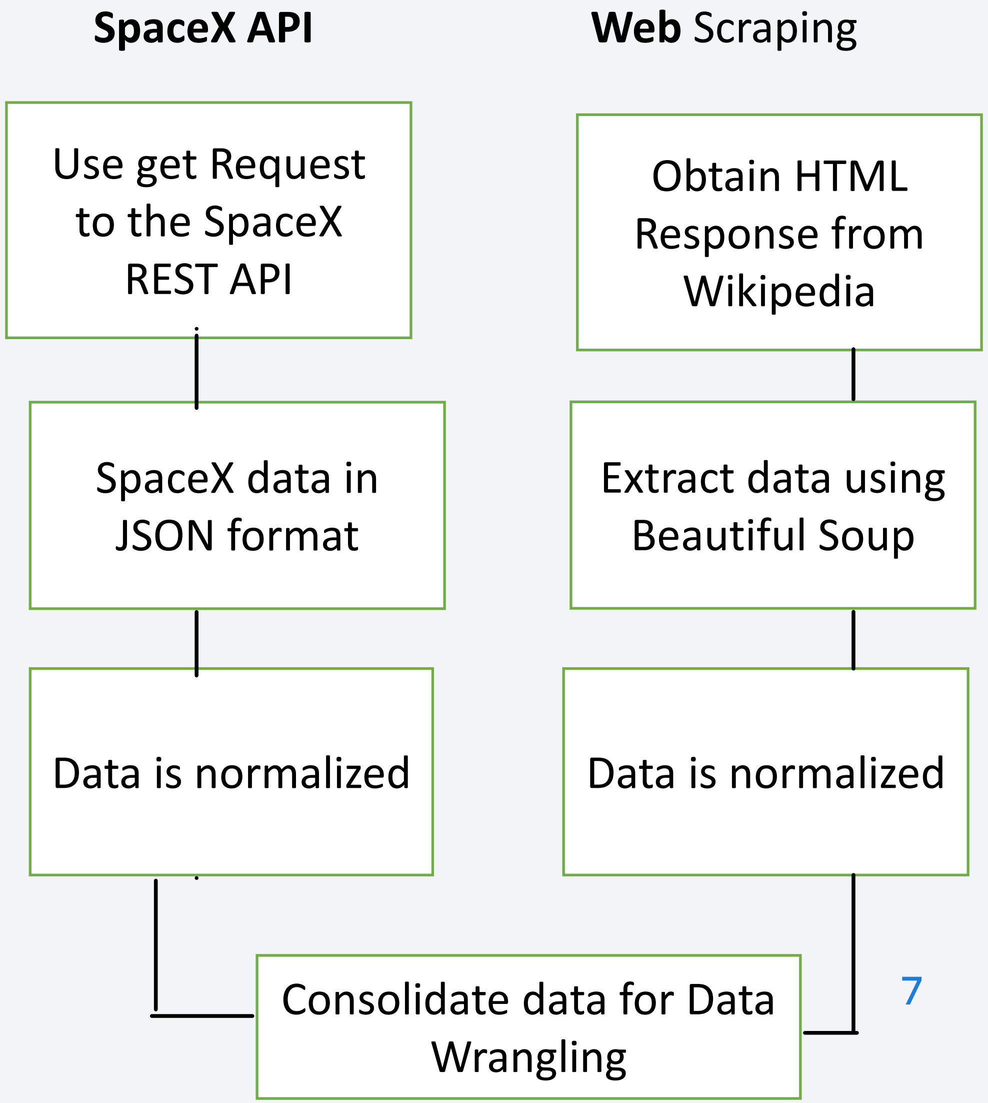
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding used for Machine Learning and cleaning data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - LR, KNN, SVM, DT models were built and evaluated

Data Collection

- Gathering of the datasets:
  - SpaceX launch data from SpaceX REST API
  - Decoded the response content as Json and convert to pandas dataframe
  - Cleaned data, checked for missing values and fill in missing values where necessary
  - Falcon 9 launch records from Web Scraping Wikipedia using BeautifulSoup
  - Extract the launch records as HTML table, parse the table and convert to pandas dataframe



SpaceX API

Use get Request to the SpaceX REST API

SpaceX data in JSON format

Data is normalized

Web Scraping

Obtain HTML Response from Wikipedia

Extract data using Beautiful Soup

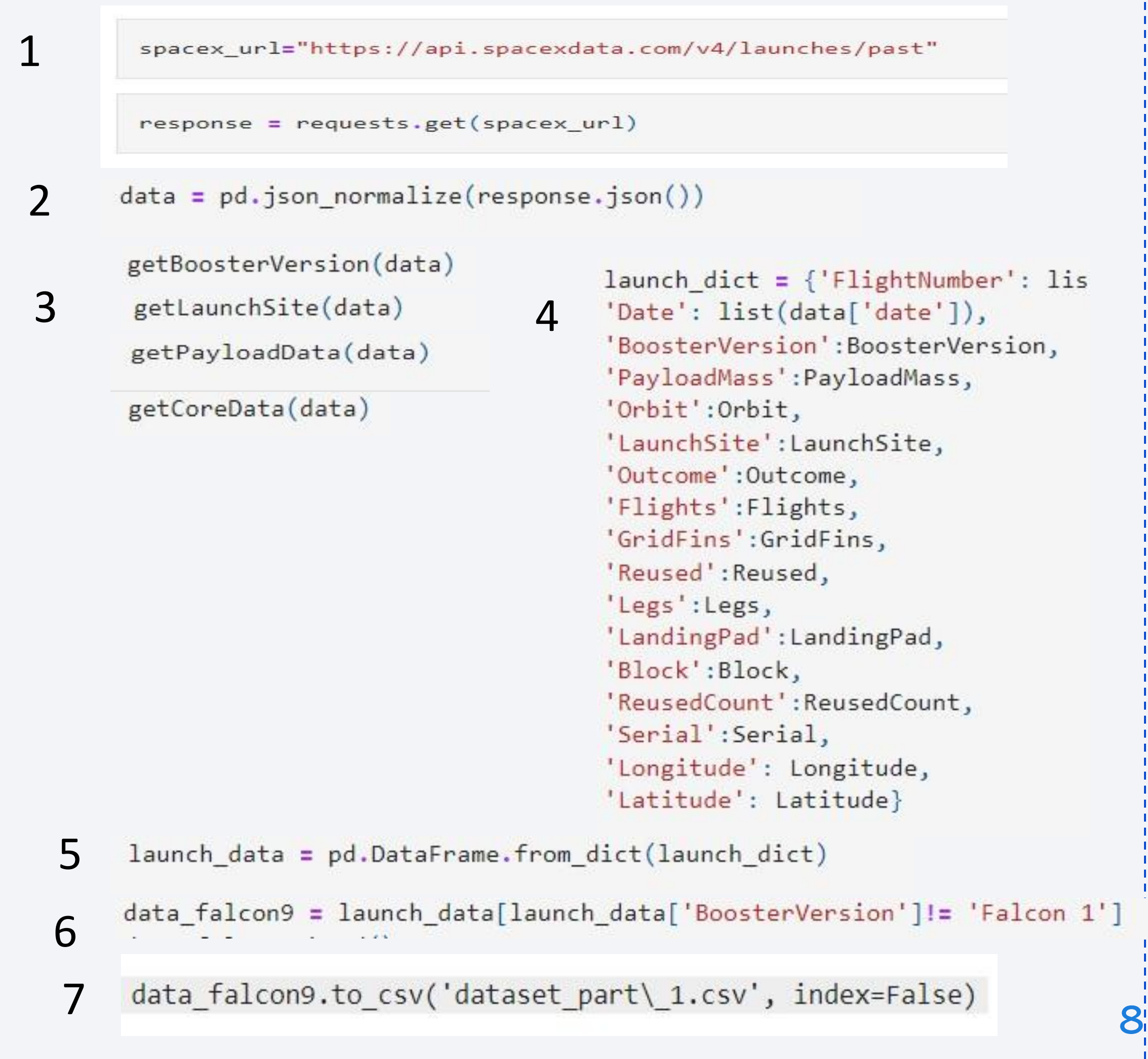
Data is normalized

Consolidate data for Data Wrangling

Data Collection – SpaceX API

- Collection of data using calls to the SpaceX REST API

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/SpaceX%20Data%20Collection%20(1).ipynb



1
```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```
```
response = requests.get(spacex_url)
```

2
```
data = pd.json_normalize(response.json())
```

3
```
getBoosterVersion(data)
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

4
```
launch_dict = {'FlightNumber': lis
'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}
```

5
```
launch_data = pd.DataFrame.from_dict(launch_dict)
```

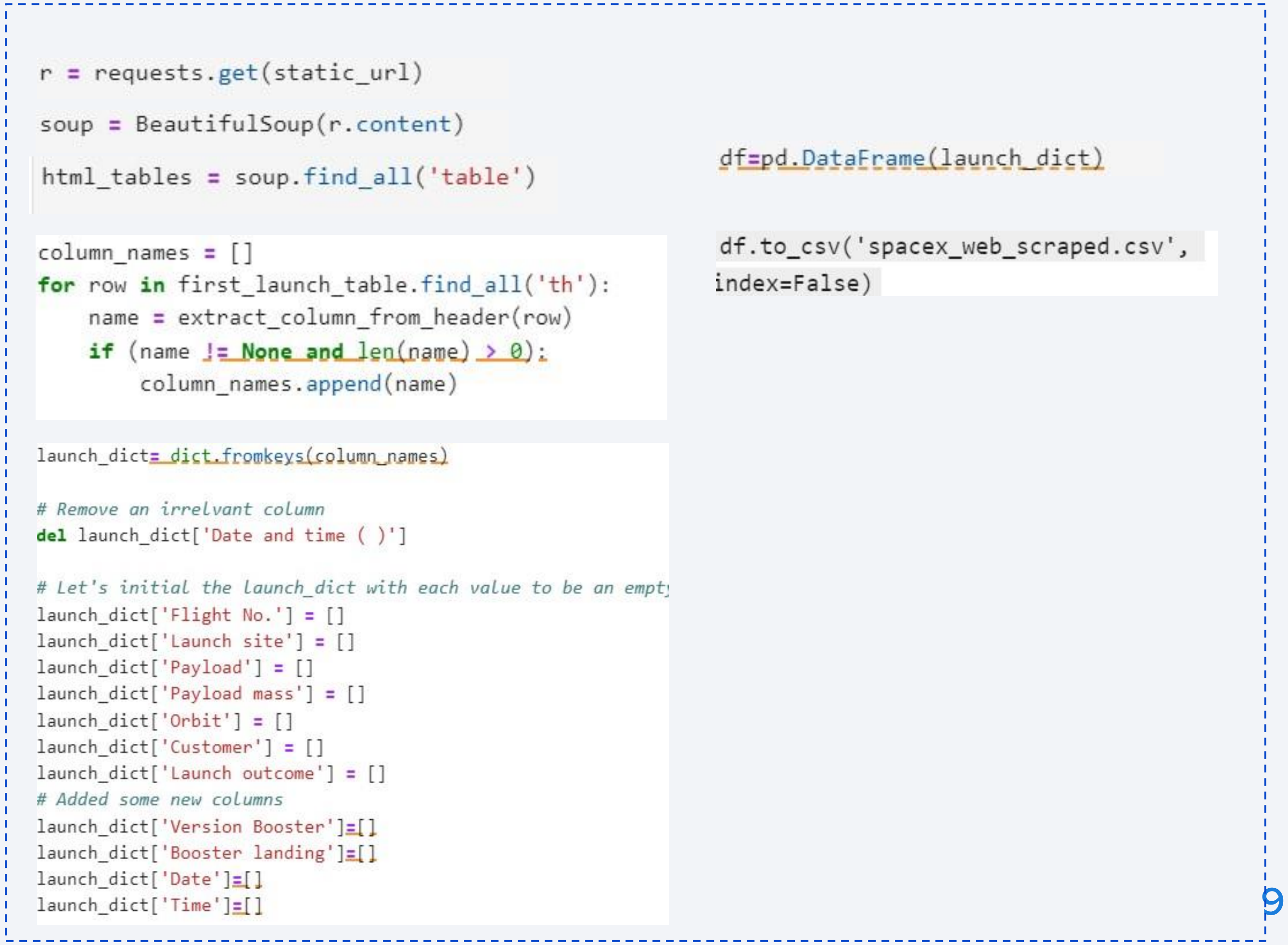
6
```
data_falcon9 = launch_data[launch_data['BoosterVersion']!= 'Falcon 1']
```

7
```
data_falcon9.to_csv('dataset_part\_1.csv', index=False)
```

Data Collection - Scraping

- Applied Web scraping techniques with BeautifulSoup
- Parsed table and converted into a Pandas dataframe

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/jupyter-labs-webscraping%20(1).ipynb



```
r = requests.get(static_url)

soup = BeautifulSoup(r.content)

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

```
column_names = []
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

```
df=pd.DataFrame(launch_dict)
```

```
df.to_csv('spacex_web_scraped.csv',
index=False)
```

Data Wrangling

- Exploratory Data Analysis
    - Calculate the number of launches at each site
    - Calculate the number and occurrence of each orbits
    - Calculate the number and occurrence of mission outcome per orbit type
    - Create landing outcome label from outcome column
    - Handle null values
    - Export the results to csv

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Visualizing the relationship between:

- flight number and launch site
- payload and launch site
- success rate of each orbit type
- flight number and orbit type
- the launch success yearly trend.

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Applied EDA with SQL to obtain insights from the data. Queries were:
    - Display the names of the unique launch sites in the space mission
    - Display 5 records where launch sites begin with the string ‘KSC’
    - Display the total payload mass carried by boosters launched by NASA (CRS)
    - Display the average payload mass carried by booster version F9 v1.1
    - List the date where the successful landing outcome in drone ship was achieved
    - List the booster names which have success in ground pad and payload mass greater than 4000 but less than 6000
    - List the total number of successful and failure mission outcomes
    - List the records which will display the month names, successful landing outcomes in ground pad, booster versions launch site for the months in year 2017
    - Ranking the count of successful landing outcomes between the 06/04/2010 and 03/20/2017 descending order

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

Objects added to the Interactive Map:

- Marked all launch sites and added map objects such as markers, circles, lines to mark the success of failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 (failure) and1 (success)
- Identified which launch sites have relatively high success rate using the color-labeled marker clusters
- Calculated the distances between launch site to its proximities

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Built interactive dashboards using Plotly dash, specifically:

- Plotted pie charts showing the total launches by certain sites
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version

https://github.com/topaz0105/IBM-DS-Capstone/blob/main/spacex_dash_app%20(1).py

Predictive Analysis (Classification)

Built a Machine Learning pipeline to predict if the first stage of the Falcon 9 lands successfully

- Preprocessing – standardize the data using numpy and pandas
- Train_test_split – split data into training and testing data
- Train the model and perform Grid Search – find the hyperparameters that allow a given algorithm to perform best
- Use the best hyperparameter values to determine the model with the best accuracy with the training data
- Test Logistic Regression, Support Vector machines, Decision Tree Classifier, and K nearest neighbors
- Output the confusion matrix
- Find the best performing classification model

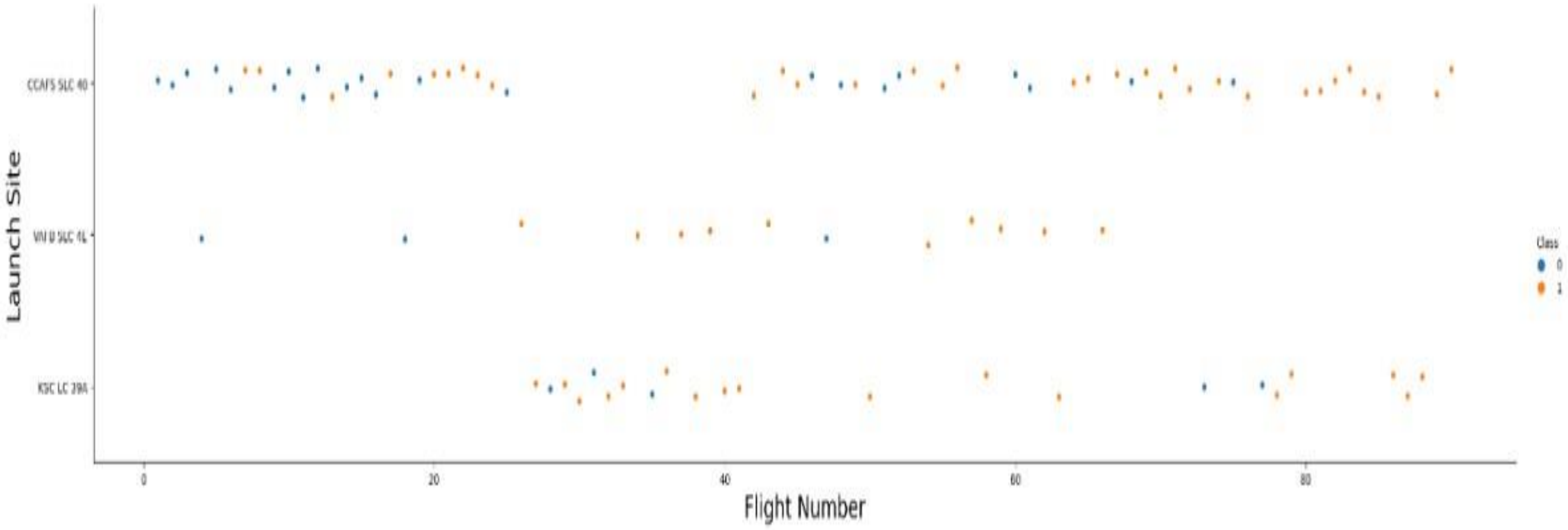
https://github.com/topaz0105/IBM-DS-Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

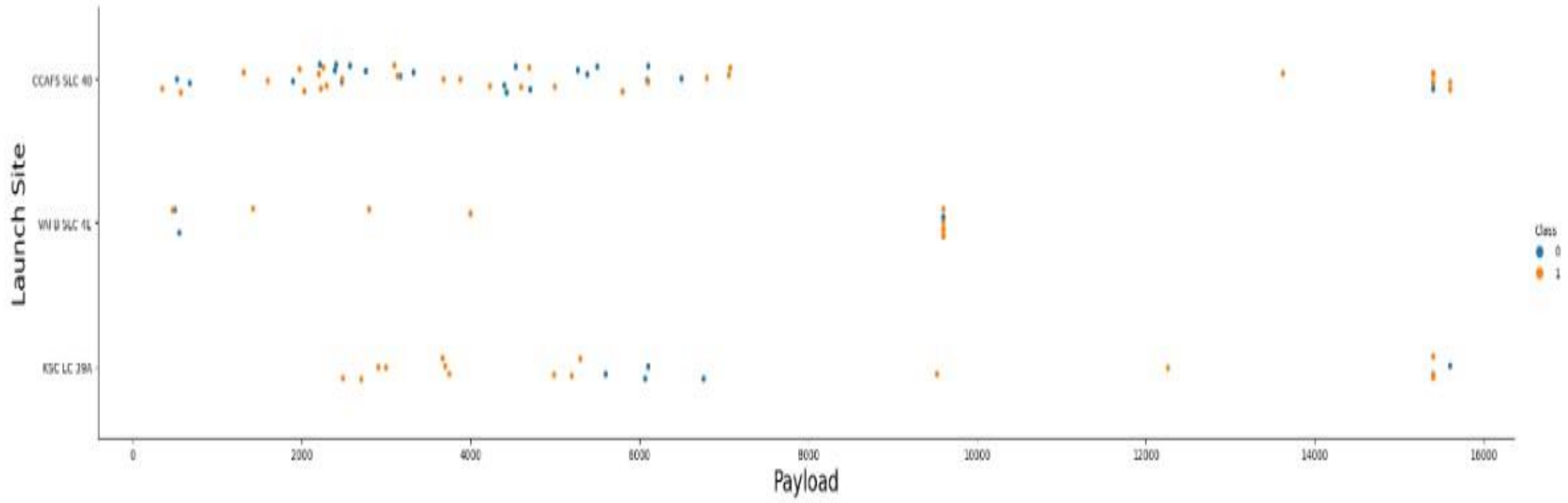
Section 2

Insights drawn from EDA



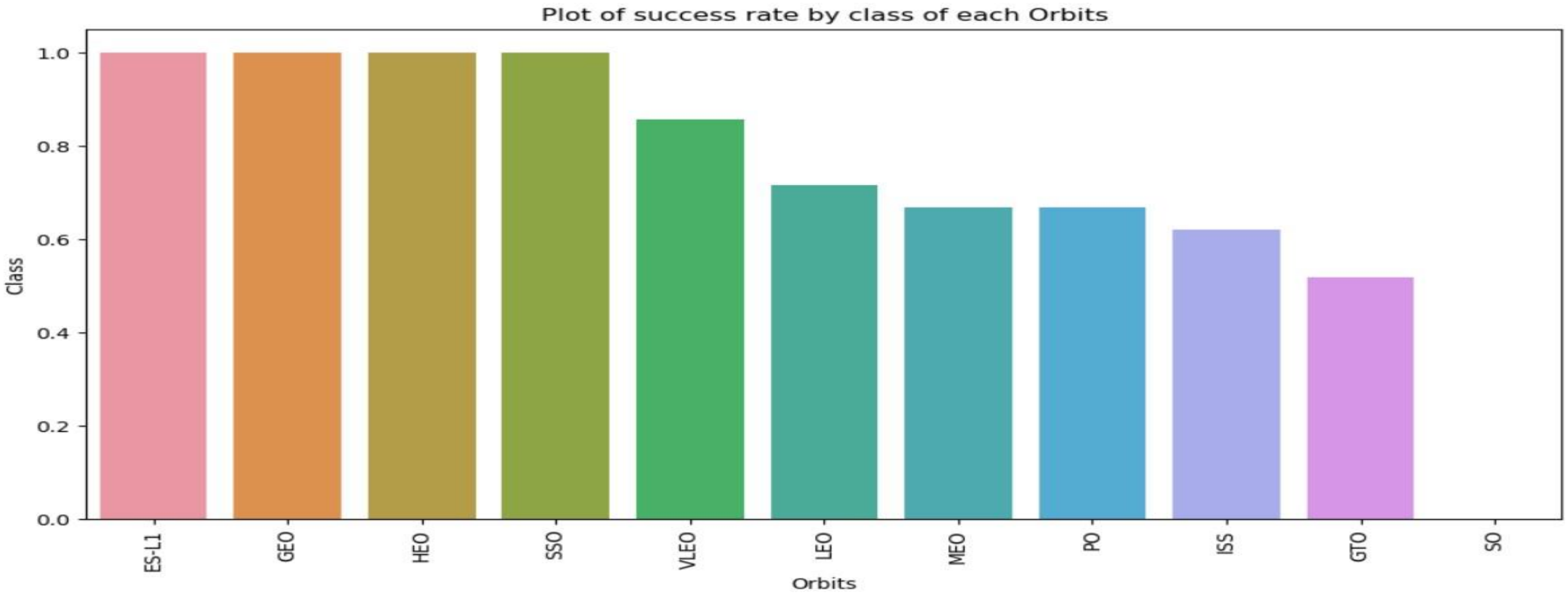
Flight Number vs. Launch Site

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



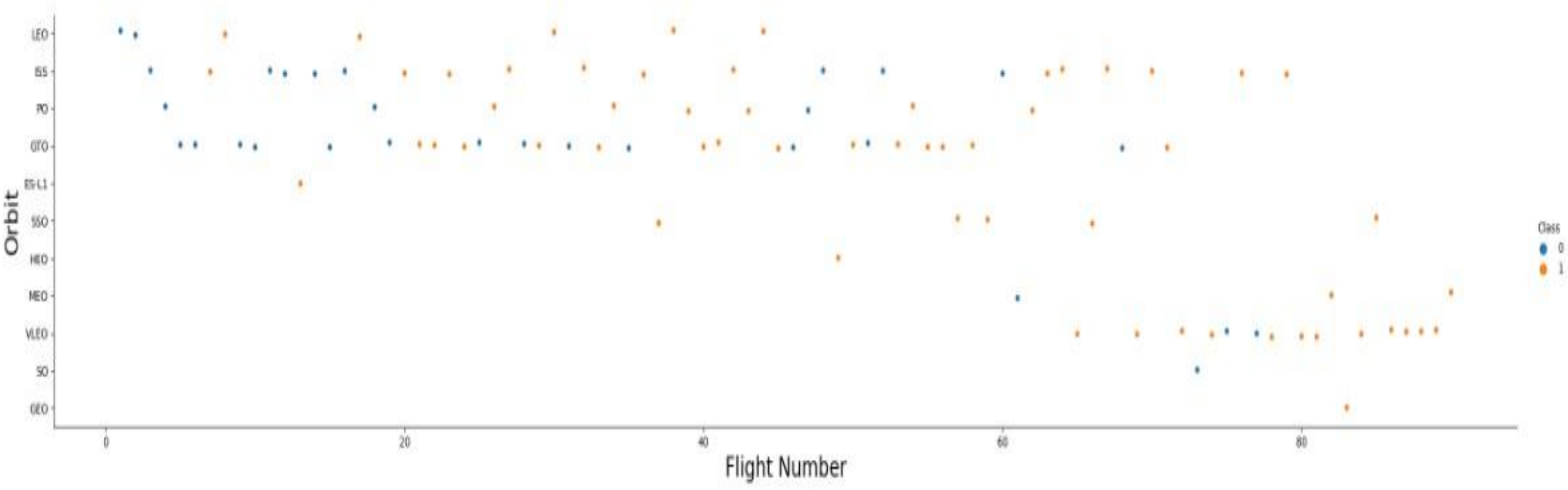
Payload vs. Launch Site

Observed greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket



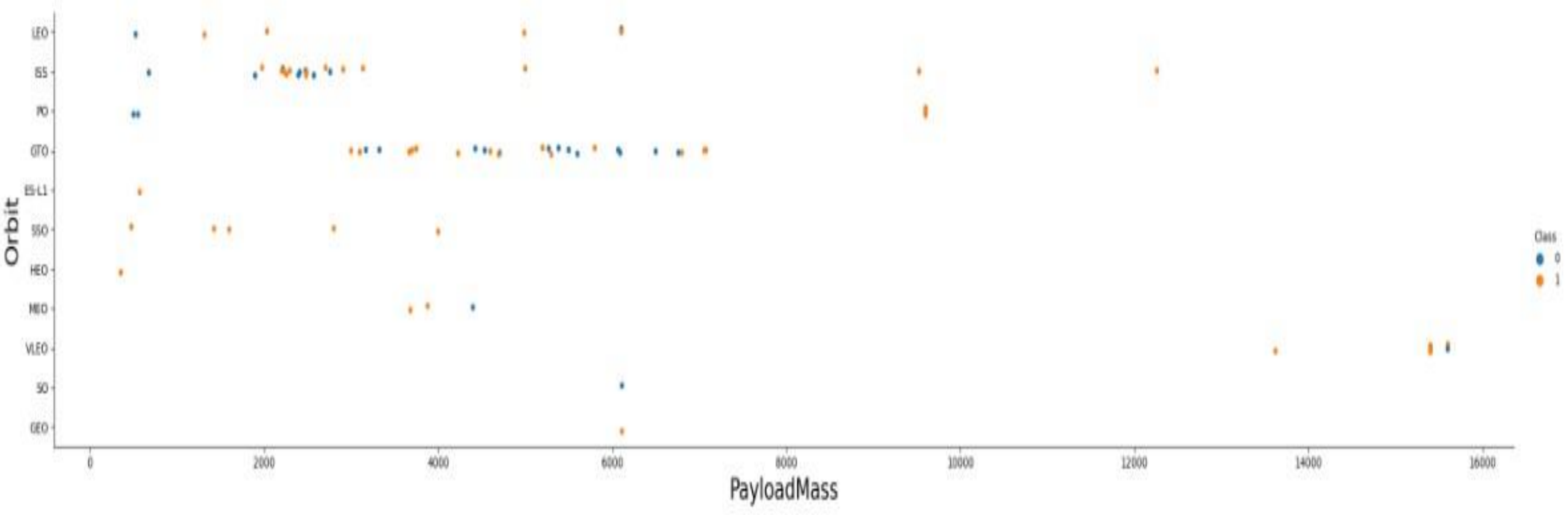
Success Rate vs. Orbit Type

Observed that ESL-1, GEO, HEO, SSO had the best success rate



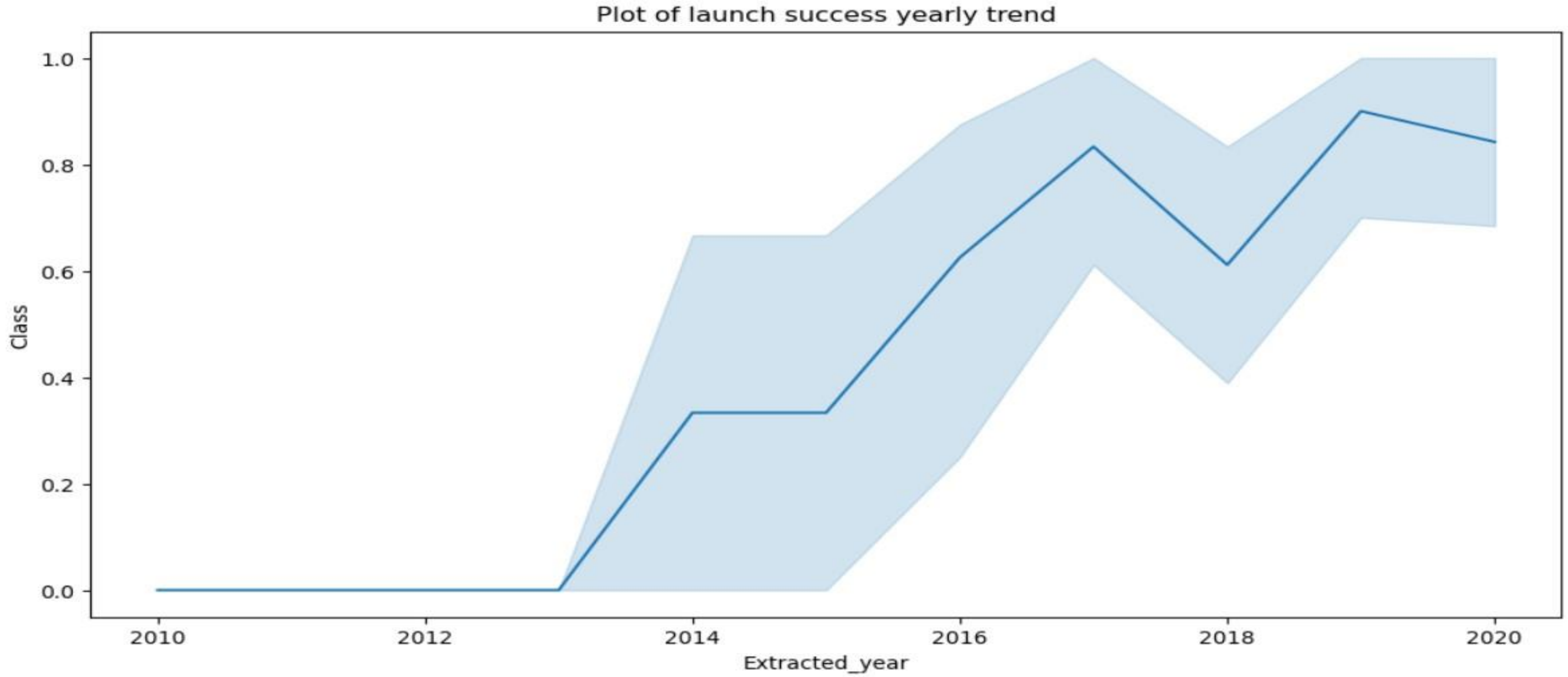
Flight Number vs. Orbit Type

Observed in the LEO orbit success is related to the number of flights but in the GTO orbit there is no relationship between flight number and the orbit



Payload vs. Orbit Type

Observed with heavy payloads that successful landings are more prevalent for PO, LEO and ISS orbits

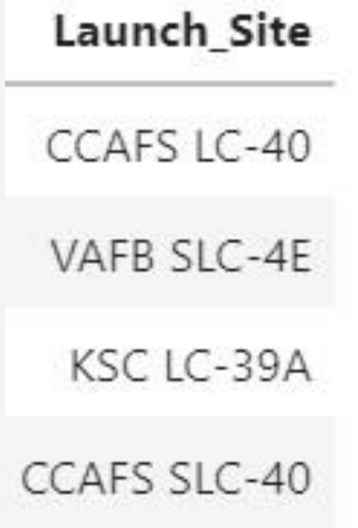


Launch Success Yearly Trend

Observed success rate since 2013 increasing until 2020

All Launch Site Names

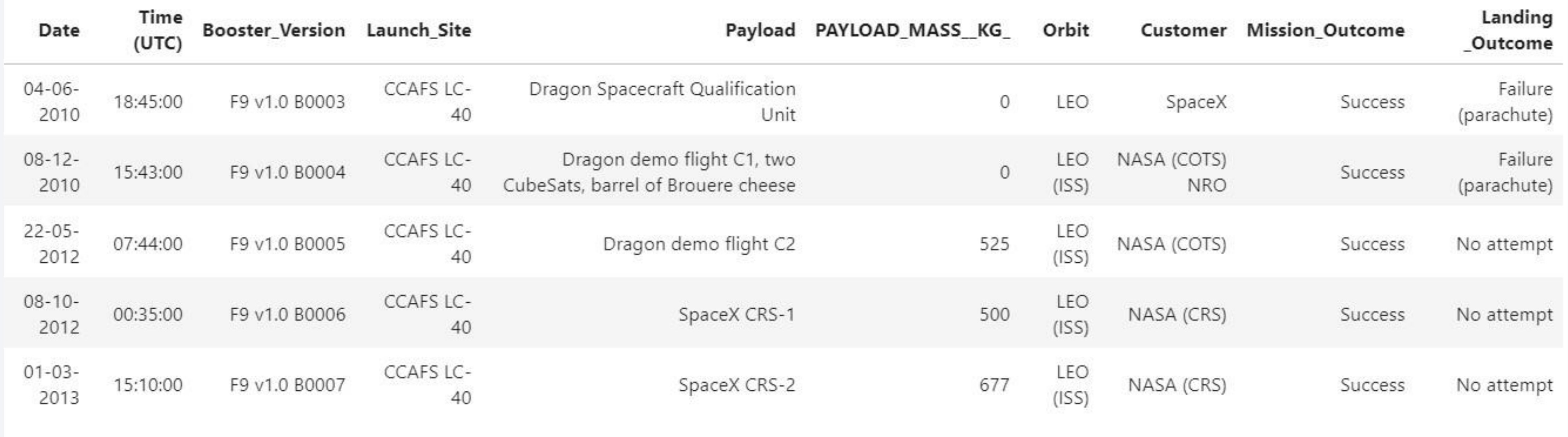
- %sql select distinct(LAUNCH_SITE) from SPACEXTBL



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

Launch Site Names Begin with 'CCA'

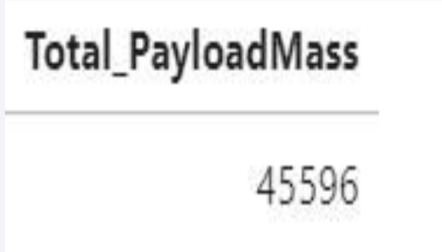
- %sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

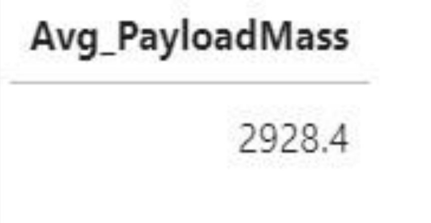
- %sql select SUM(PAYLOAD_MASS__KG_) AS Total_PayloadMass from SPACEXTBL where CUSTOMER like 'NASA (CRS)'



Total_PayloadMass
45596

Average Payload Mass by F9 v1.1

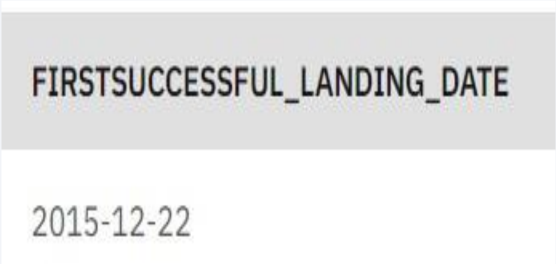
- %sql select AVG(PAYLOAD_MASS__KG_) AS Avg_PayloadMass from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'



Avg_PayloadMass
2928.4

First Successful Ground Landing Date

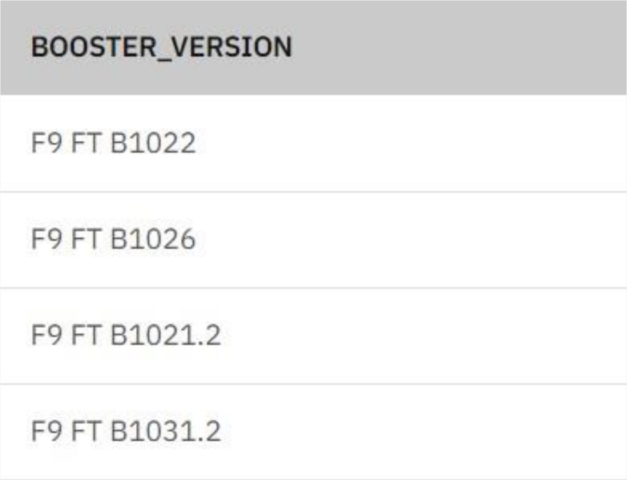
- %sql select MIN(DATE) AS FirstSuccessful_landing_date from SPACEXTBL where "Landing__Outcome" LIKE 'Success (ground pad)'



FIRSTSUCCESSFUL_LANDING_DATE
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

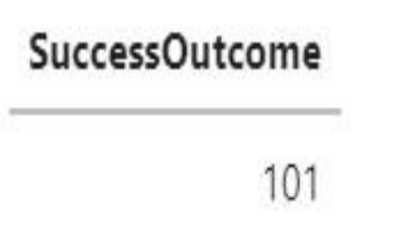
- %sql select BOOSTER_VERSION from SPACEXTBL where "Landing__Outcome" = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS__KG_ < 6000



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

Total Number of Successful and Failure Mission Outcomes

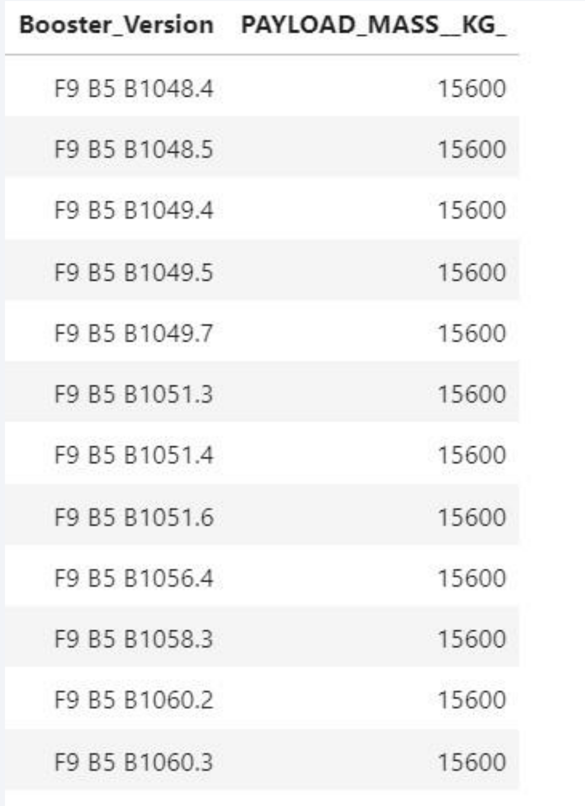
- %sql select count(MISSION_OUTCOME) AS SuccessOutcome from SPACEXTBL where "Mission_Outcome" LIKE 'Success%' or "Mission_Outcome" = 'Failure (in flight)'



SuccessOutcome
101

Boosters Carried Maximum Payload

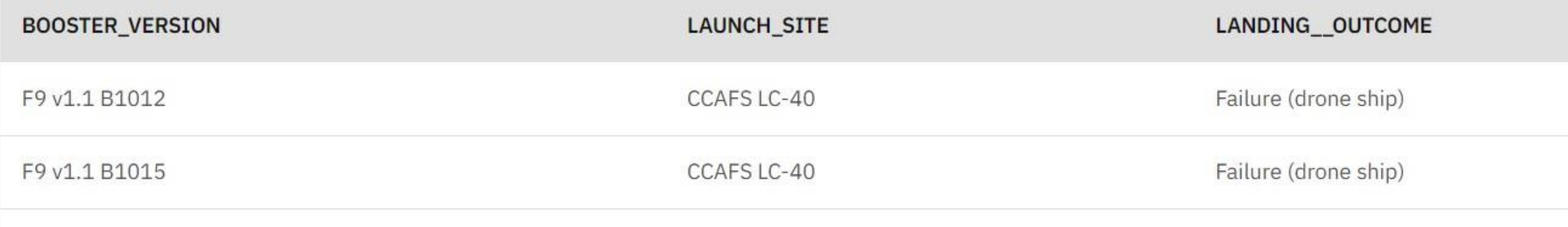
- %sql select BOOSTER_VERSION, Payload_Mass__KG_ from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL) ORDER BY Booster_Version



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

2015 Launch Records

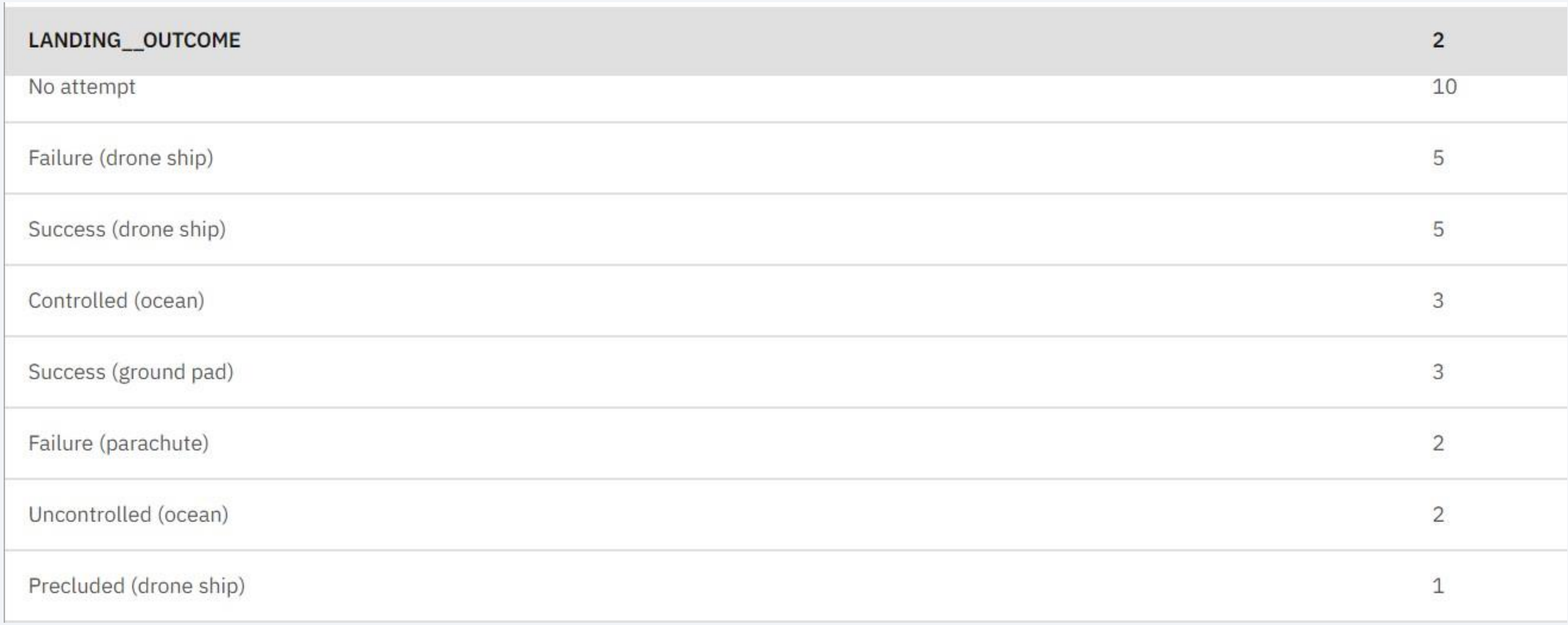
- %sql select Booster_Version, Launch_Site, "Landing__Outcome" from SPACEXTBL where "Landing__Outcome" like 'Success%' and (DATE between '2015-01-01' and '2015-12-31') order by date desc



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)

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

- %sql select "Landing__Outcome", COUNT("Landing__Outcome") from SPACEXTBL where DATE between '2010-06-04' and '2017-03-20' GROUP BY "Landing__Outcome" ORDER BY COUNT("LANDING__OUTCOME") DESC

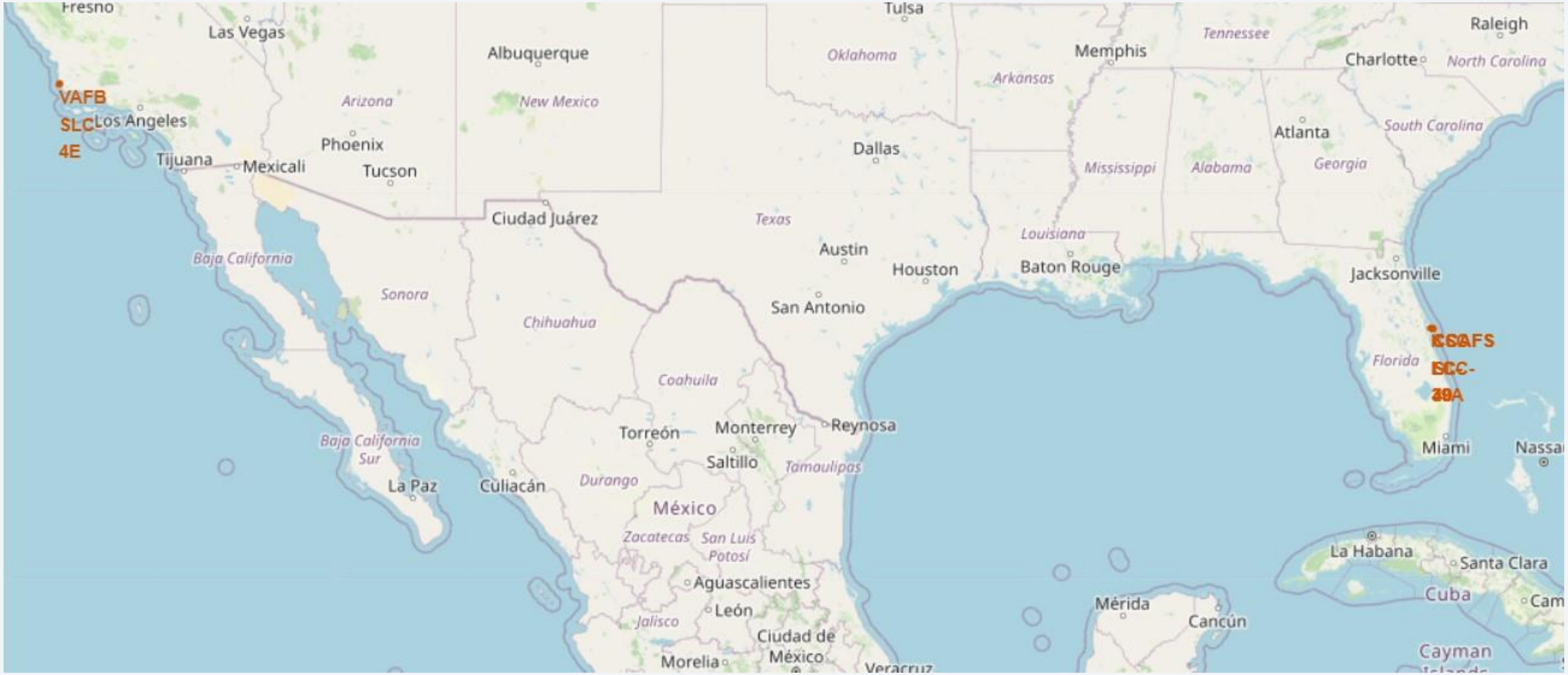


LANDING__OUTCOME	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

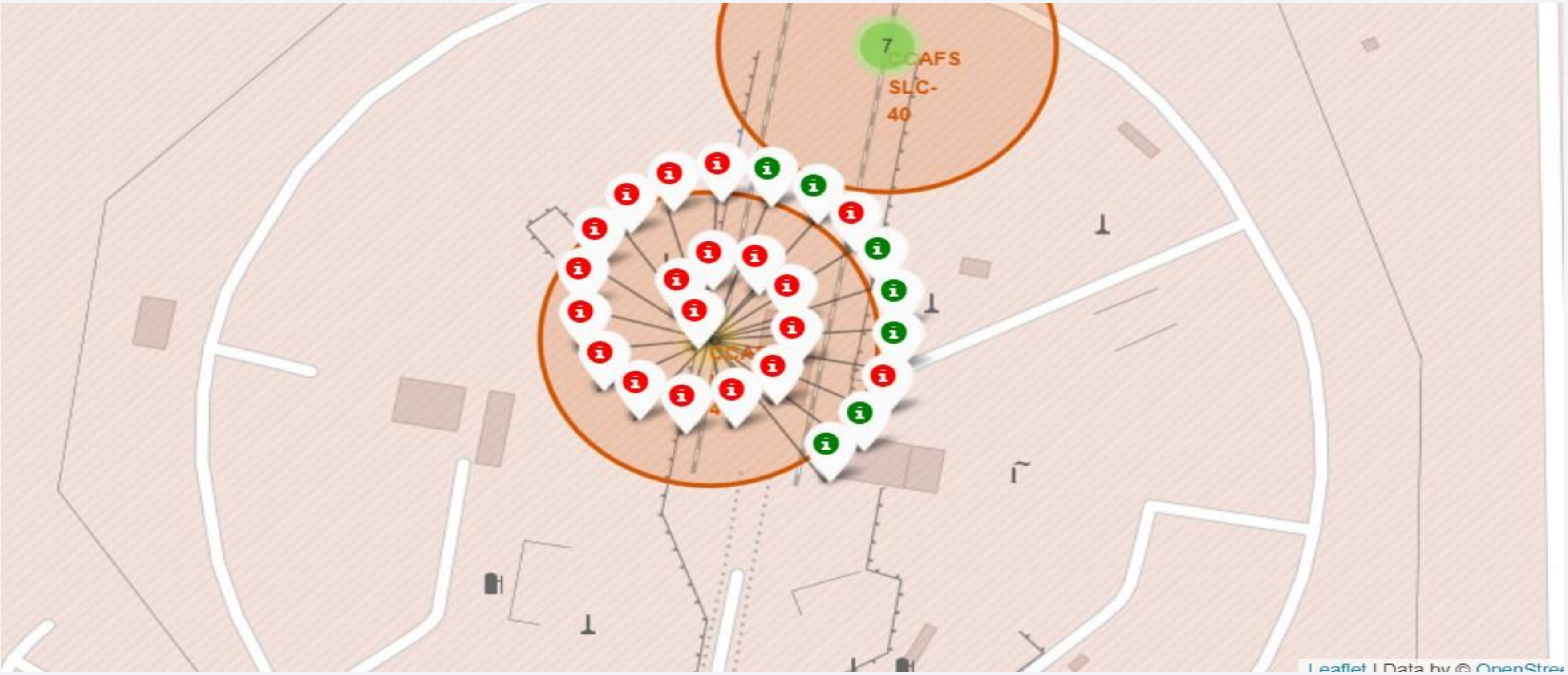
Launch Sites Proximities Analysis

All Launch Sites' Locations on Global Map



Many of the launch site locations are in the United States near the coast and in areas that have a warmer climate.

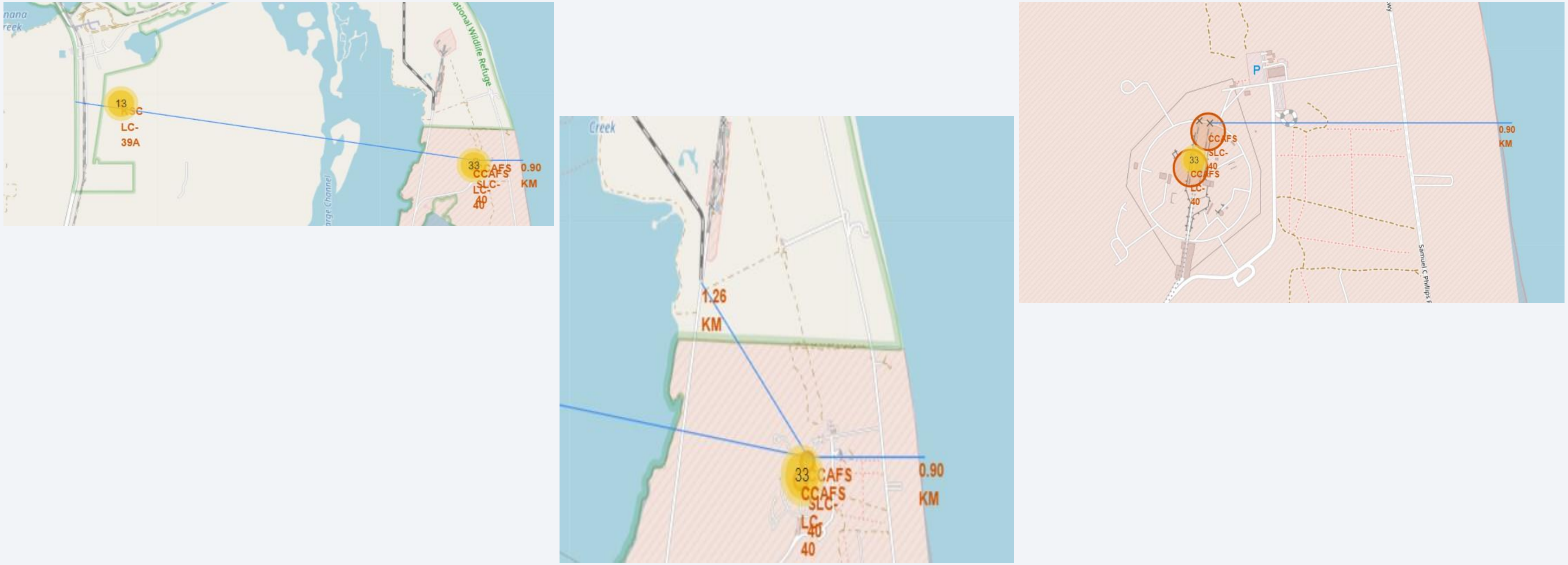
Success/Failure of Launch Outcomes



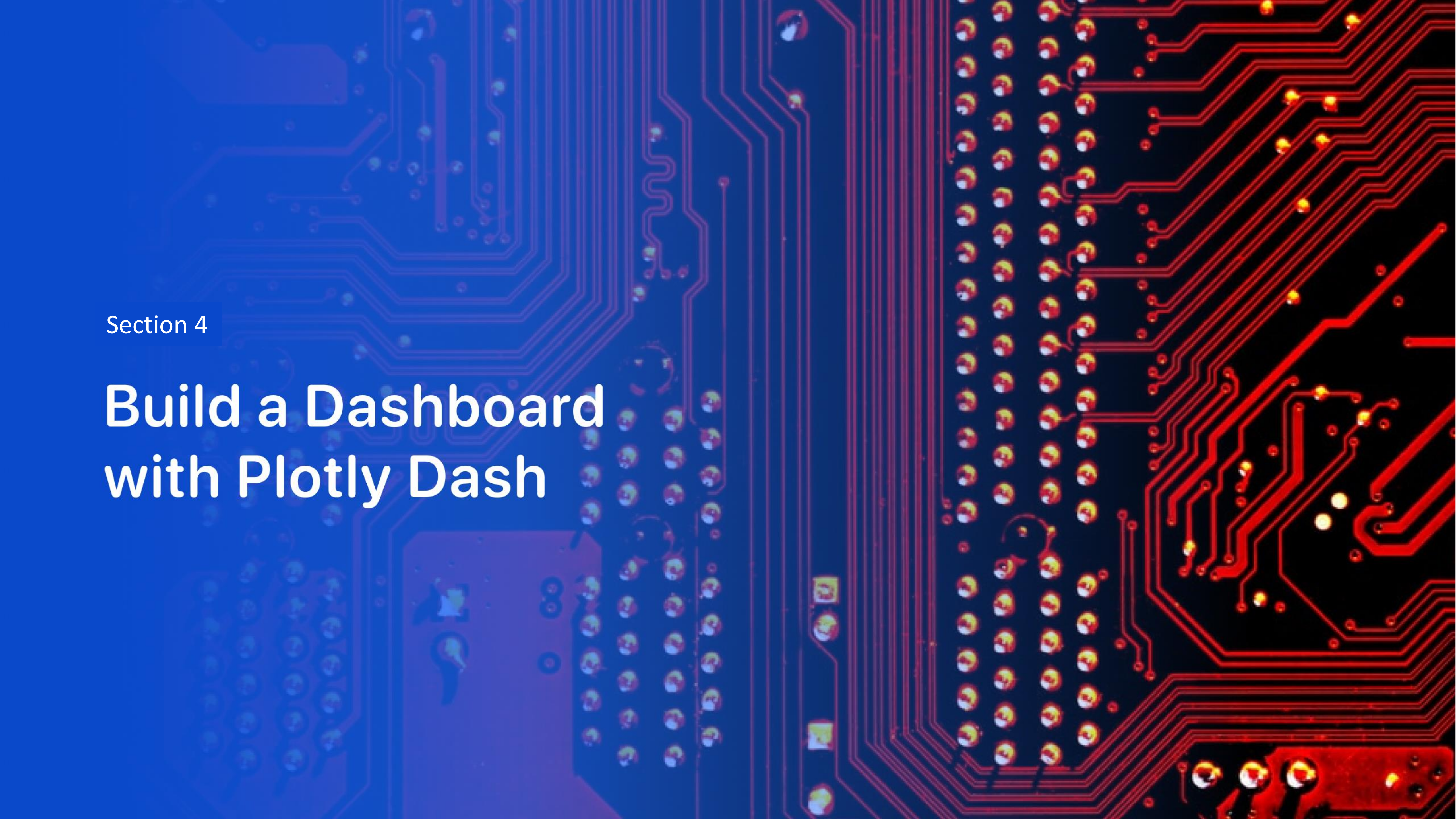
From the color-coded markers it is easy to identify which launch sites have relatively high success rates.

Green Marker shows successful launches and Red Marker shows Failures

Launch Location Distance to Landmarks



Launch sites are in close proximity to the coastline but keep a far amount of distance from highways, railways and cities

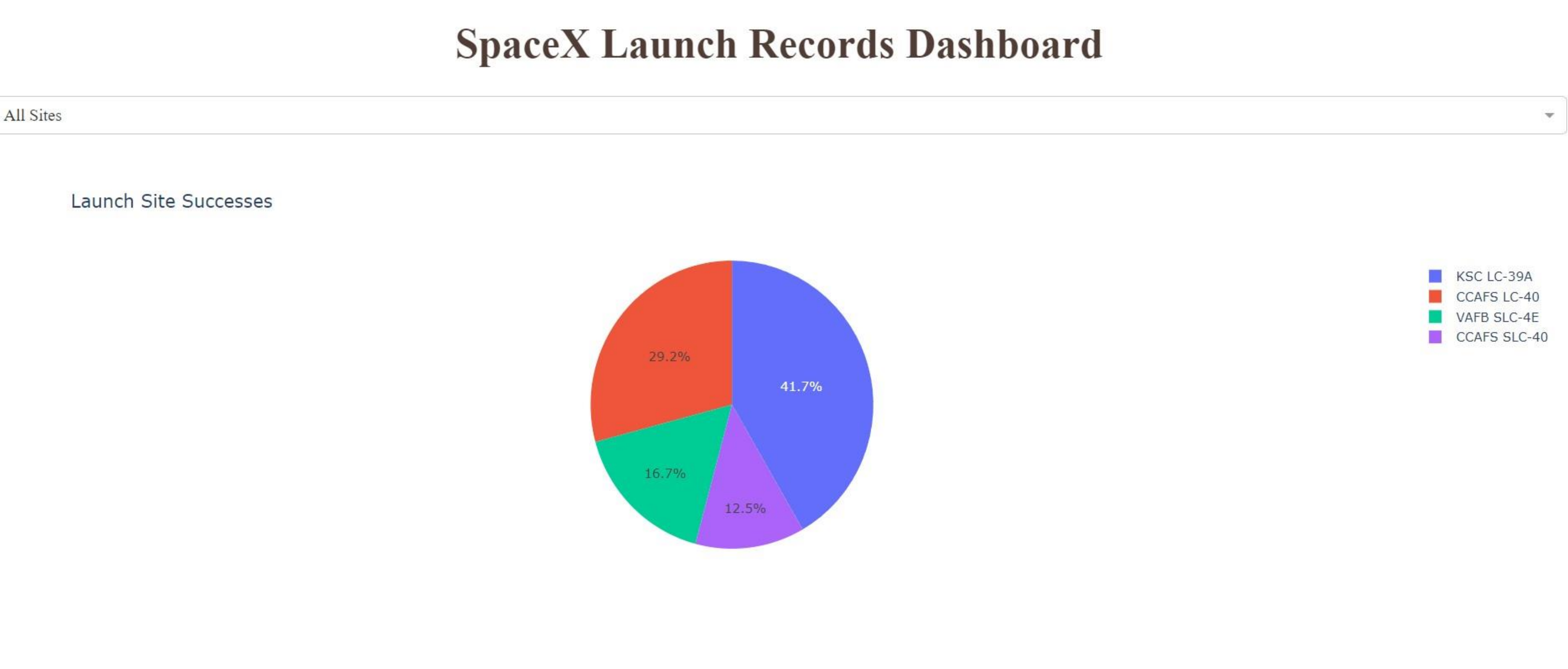


Section 4

Build a Dashboard with Plotly Dash

Total Success Percentage for all Launch Sites

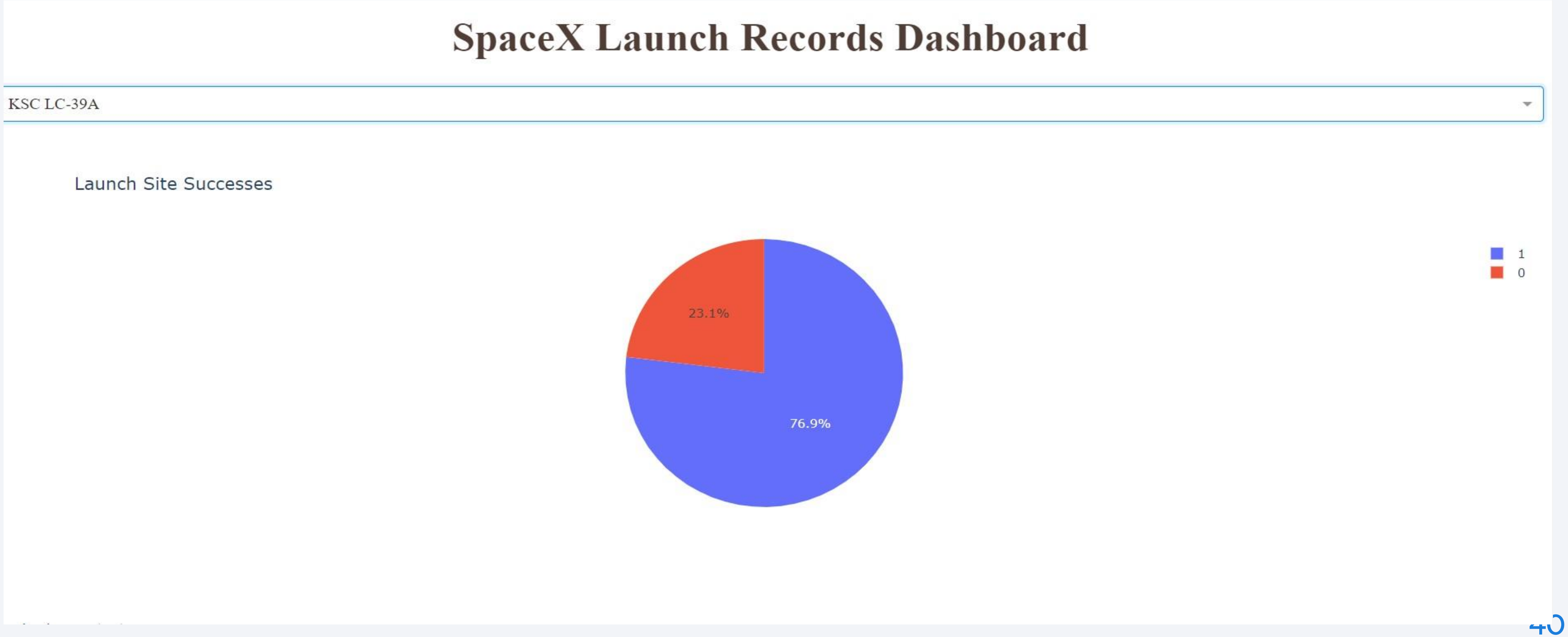
Site KSC LC-39A had the most successful launches from all the sites



SpaceX Launch Records Dashboard

Launch Site with Highest Launch Success Ratio

KSC LC-39A had a 76.9% success rate and only a 23.1% failure rate



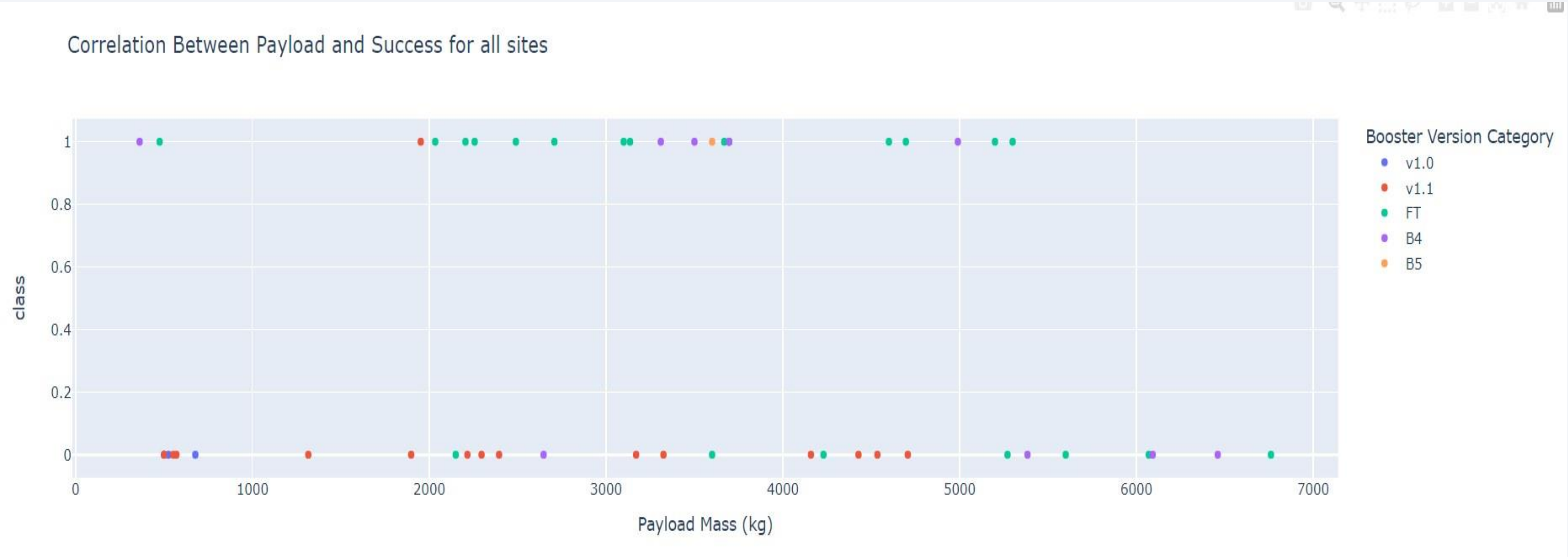
SpaceX Launch Records Dashboard

KSC LC-39A

Launch Site Successes

Correlation between Payload and Success for all sites

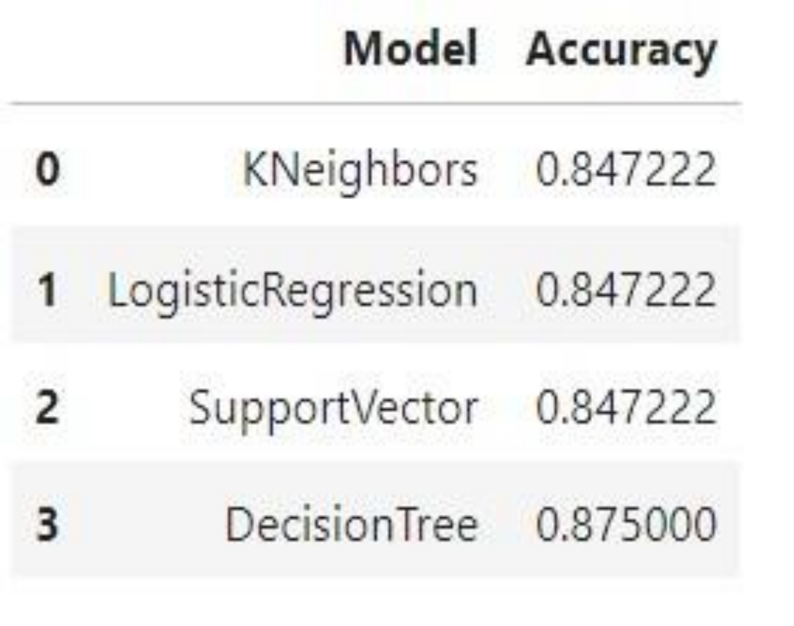
Observed the lower weighted payloads had a higher success rate than heaver weighted payloads



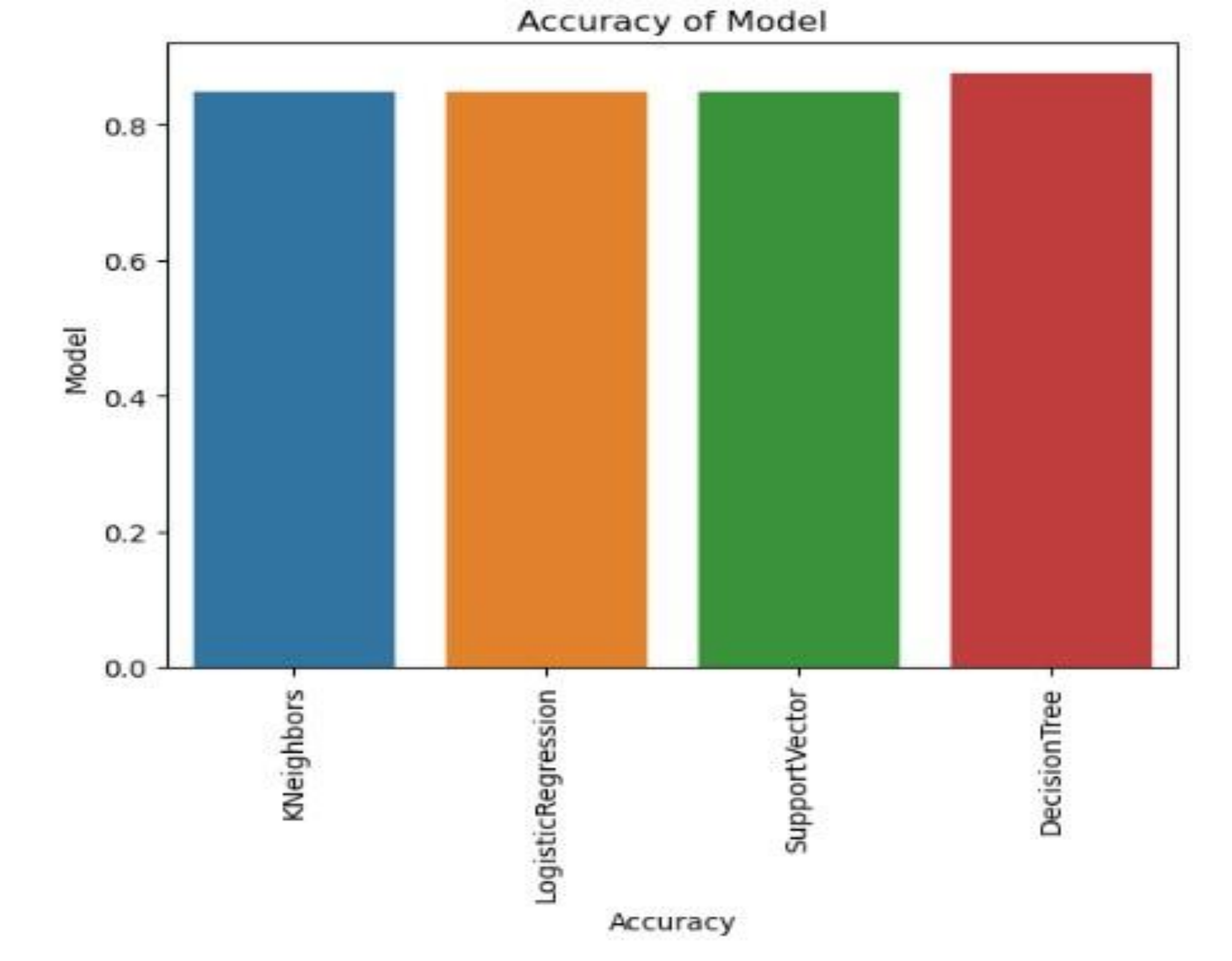
Section 5

Predictive Analysis (Classification)

Classification Accuracy



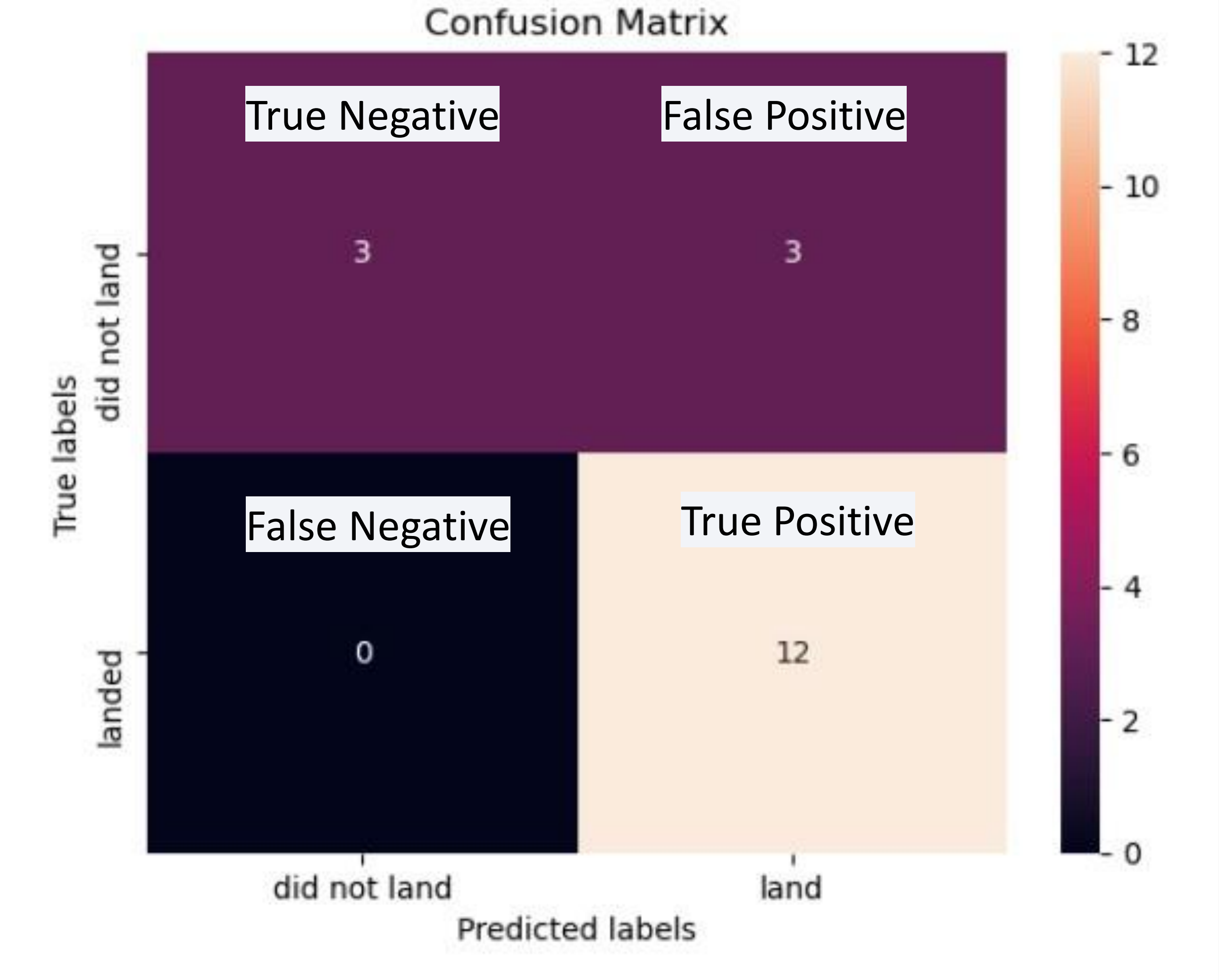
	Model	Accuracy
0	KNeighbors	0.847222
1	LogisticRegression	0.847222
2	SupportVector	0.847222
3	DecisionTree	0.875000



The Best Model is the Decision Tree classifier with 87.5%.

Confusion Matrix

The Confusion Matrix for the Decision Tree classifier can distinguish between the different classes. Major problem is false positives. The classifier may indicate an unsuccessful landing as a successful landing.



	did not land	land
did not land	True Negative 3	False Positive 3
landed	False Negative 0	True Positive 12

Conclusions

- Larger the flight amount at a launch site, the greater the success rate at a launch site
- Launch success rate started to increase in 2013 until 2020
- Orbits ES-L1, GEO, HEO, SSO had the most success rate
- KSC LC-39A had the most successful launches of any sites
- Low weighted payloads perform better than heavier payloads
- The Decision Tree classifier is the best machine learning algorithm

Thank you!