IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Raúl Sanz Jodar
23/03/2024

Outline

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

Executive Summary

Summary of methodologies

- Data Collection with API and Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL and Data Visualization
- Interactive Maps with Folium and interactive Dashboard with Dash and Plotly
- Predictions with Machine Learning Algorithms

Summary of all results

- ES-L1, GEO, HEO, and SSO orbits exhibit the highest success rates.
- KSC LC-39A is the most successful launch site.
- The Decision Tree is the best machine learning model for the project.

Introduction

Project background and context

The commercial space age is here. One of the most successful company is SpaceX, thanks to its relatively inexpensive rocket launches.

SpaceX advertises Falcon 9 rocket launches on its website with a cost of $62Million, while other providers cost more than $165Million, much of the savings is because SpaceX can reuse the first stage.

Key objectives

- Investigate the correlation between mission parameters and their influence on landing outcomes.
- Determine the probability of reuse the first stage, i.e., the optimal conditions to achieve successful landings.

Section 1

Methodology

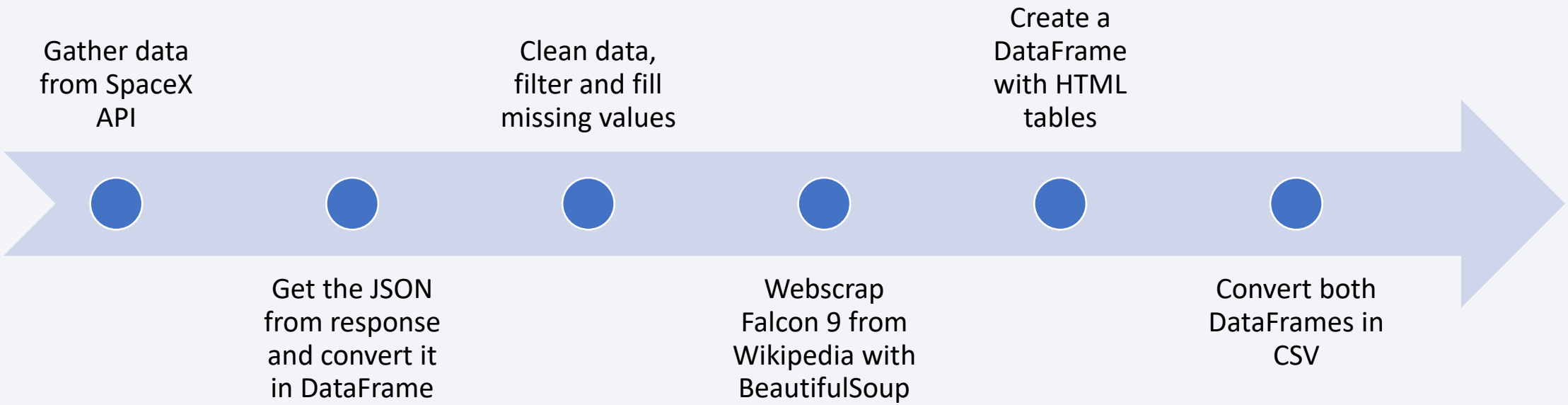
Methodology

Executive Summary

- Data collection methodology:
  - SpaceX launch data gathered from SpaceX REST API and Web Scraping Wikipedia.
- Perform data wrangling
  - Apply One-hot encoding in ‘Outcome’ to convert it into binary feature
- 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

- We have two datasets, one with SpaceX data collected directly from SpaceX REST API. The other one represents the Falcon 9 launch records and has been collected via web scraping the Wikipedia.



Data Collection – SpaceX API

Get response from API

Convert the response to DataFrame

Complete data via the API using id numbers

Clean and filter data

Deal with missing values

Export it to a CSV file

[SpaceX API calls notebook]

```
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
json_data = requests.get(static_json_url).json()
data = pd.json_normalize(json_data)
```

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

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

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

```
# Calculate the mean value of PayloadMass column
payload_mass_mean = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, payload_mass_mean, inplace=True)
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Collection - Scraping

- Get response from Website
- Create BeautifulSoup from response
- Find the launch table and column names
- Create dictionary and parse the data into it
- Convert dictionary to DataFrame
- Export it to a CSV file

[Web scraping notebook]

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
response = requests.get(static_url)  soup = BeautifulSoup(response.content, 'html.parser')
```

```
html_tables = soup.find_all('table') first_launch_table = html_tables[2]
```

```
headers = first_launch_table.find_all('th')
for header in headers:
    column_name = extract_column_from_header(header)
    if column_name != None and len(column_name) > 0:
        column_names.append(column_name)
```

```
launch_dict= dict.fromkeys(column_names)
```

Init launch_dict with empty lists and add new columns

```
del launch_dict['Date and time ( )']
```

Parse tables from soup to fill up the launch_dict

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
df.to_csv('spacex_web_scraped.csv', index=False)
```

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

Data Wrangling

- Load CSV data into DataFrame
- Identify missing values and types
- Calculate the number of launches per site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome of each orbit
- Create a landing outcome label from Outcome column
- Export it to a CSV file

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv")

df.isnull().sum()/len(df)*100   df.dtypes

df['LaunchSite'].value_counts()

df['Orbit'].value_counts()

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

bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])

landing_class = []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)

df['Class']=landing_class

df.to_csv("dataset_part_2.csv", index=False)
```

	Class
0	0
1	0
2	0
3	0
4	0
...	...
85	1
86	1
87	1
88	1
89	1

[Data wrangling notebook]

EDA with Data Visualization

The charts that have been plot are:

- Scatter plot to find relationship between features and try to predict the features that leads to successful landings:
  - FlightNumber vs. PayloadMass
  - FlightNumber vs. LaunchSite
  - PayloadMass vs. LaunchSite
  - FlightNumber vs. Orbit
  - PayloadMass vs. Orbit
- Bar chart to determine which orbits have the highest probability for successful landing:
  - Orbit vs. Success Rate (mean Class)
- Line chart to see how the trend changes over time and make predictions:
  - Date vs. Success Rate (mean Class)

EDA with Data visualization notebook

EDA with SQL

Some of the SQL queries performed to get insights of the data are:

- Display the names of the unique launch sites in the space mission
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- 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 records which will display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015.

EDA with SQL notebook

Build an Interactive Map with Folium

Folium provides a series of map objects designed to enhance data visualization. The ones that we have used are:

- Circle marker: Used for each launch site to represent the area with the label corresponding to the name of each site.
- Map markers and Icon Markers: Used to represent the success or failed launches of each site. The ones with class 0 (failure) are red while the ones with class 1 (success) are green. We have also marked landmarks nearby the launch sites to study their proximity.
- Marker cluster: Used to group nearby markers together at higher zoom levels, providing a cleaner map.
- PolyLine: Used to create a line between the launch sites and landmarks.

With all this objects we can represent and study the proximity between the launch sites and coastlines, highways, railways and cities.

Analytics with Folium notebook

Build a Dashboard with Plotly Dash

We have created an interactive dashboard to perform visual analytics on SpaceX launch data in real-time.

The plots available are:

- Pie chart with total success for all sites or a specific launch site.
- Scatter plot with the relationship between the outcome and payload mass for the different booster versions.

The interactions are:

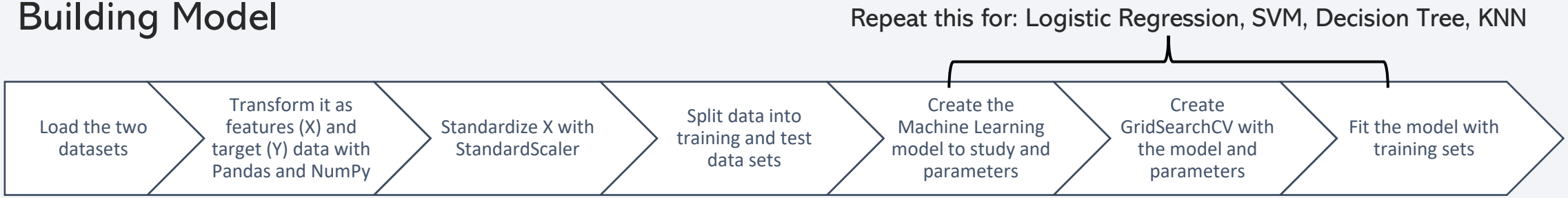
- Dropdown to select if we want to see the chart for all sites or only a specific launch site.
- Slider to specify a range of the payload mass used in the scatter plot.

With this, we can identify the site with more successful launches and highest ratio, determine the payload range with highest and lowest launch rate and the F9 booster version with the highest launch success rate.

Plotly Dash lab

Predictive Analysis (Classification)

Building Model



Evaluating each Model

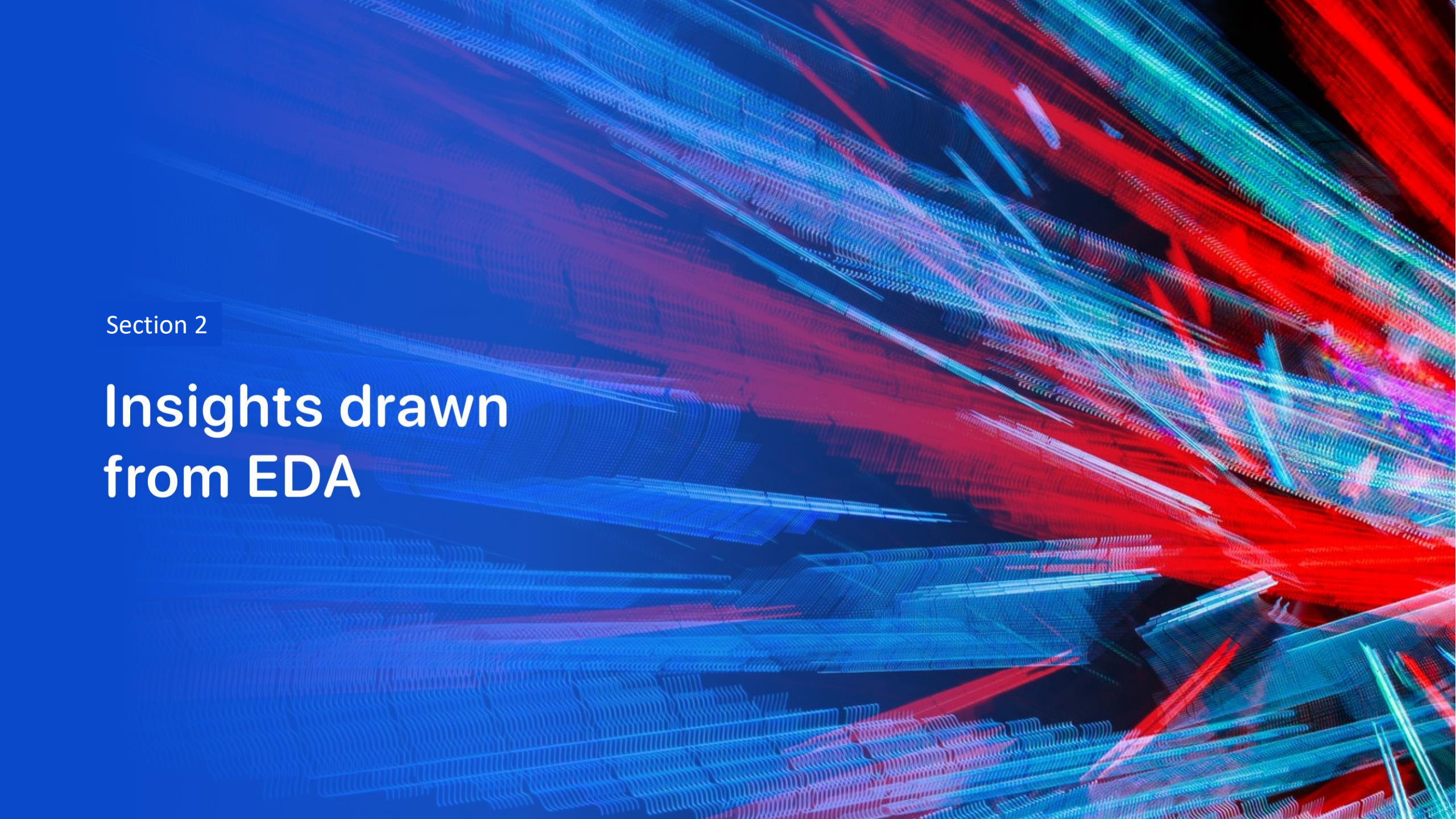
Finding best Model

The model with the higher accuracy (best score) will be the best model to make predictions.

[Machine Learning notebook]

Results

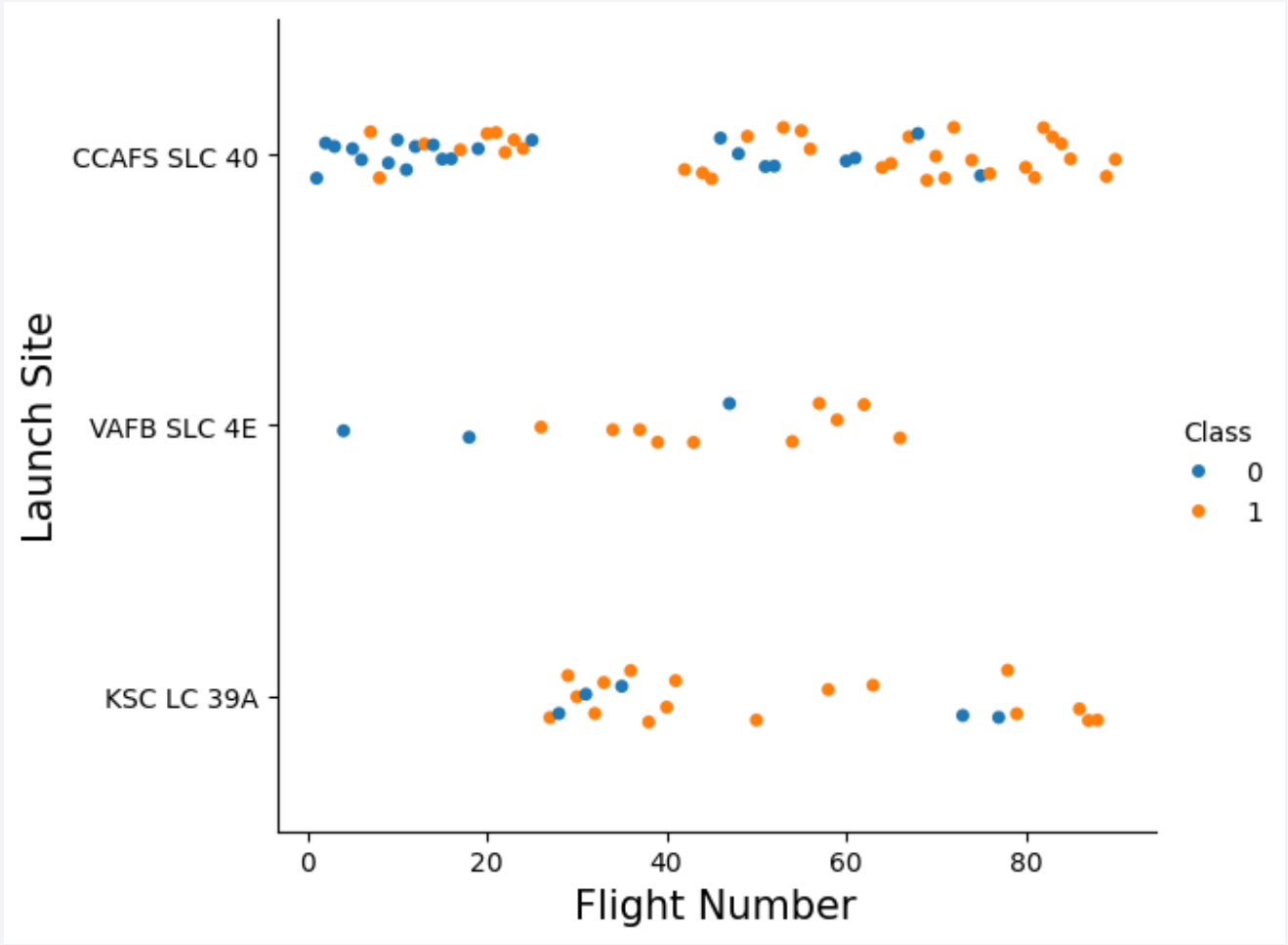
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

- This scatter plot shows the FlightNumber in x and the Launch Site in y, with the hue representing the class (success or fail).

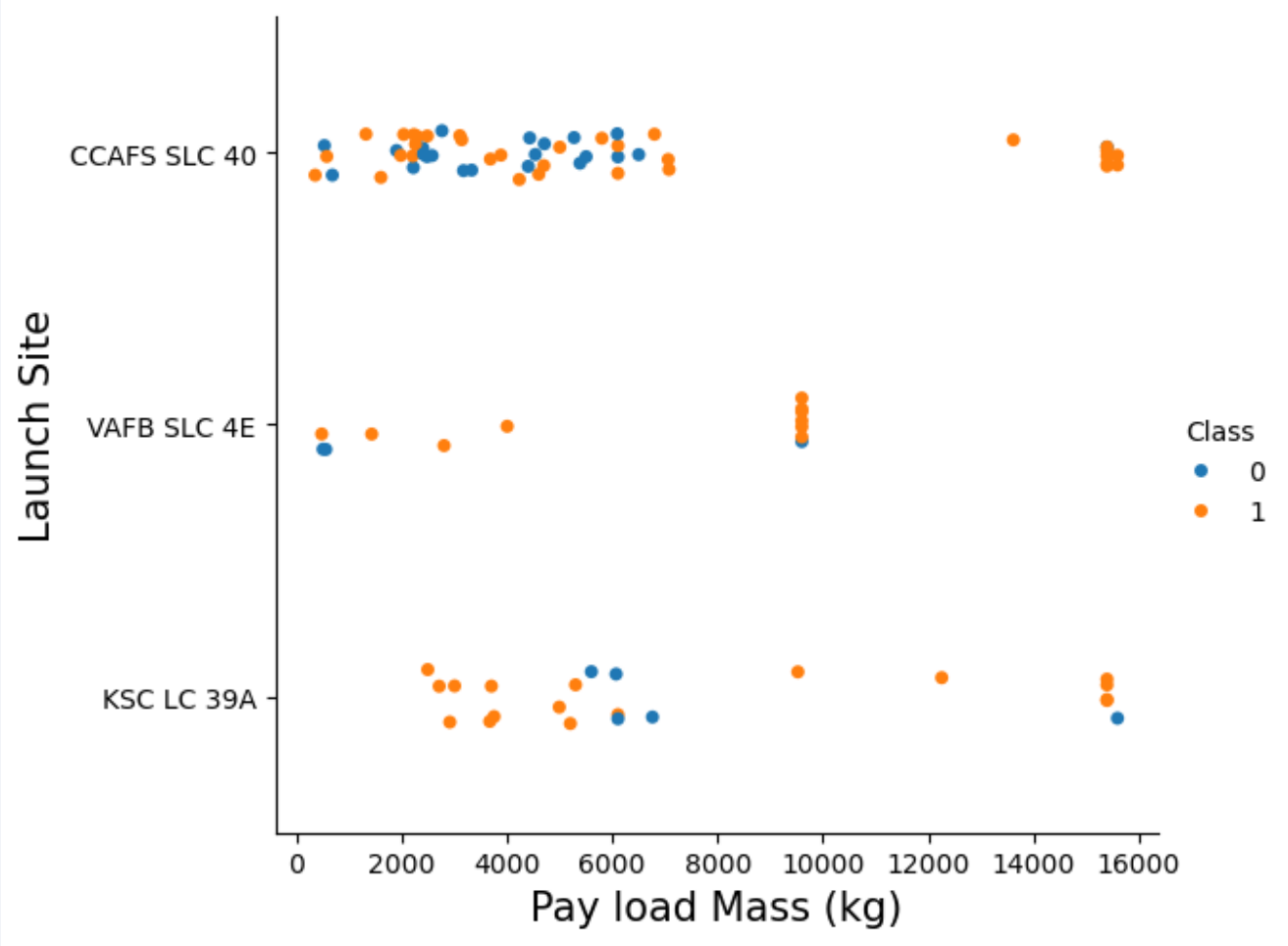
- Indicates that the success rate incresease up to around the 30 flight number.



Payload vs. Launch Site

- This scatter plot shows the Payload Mass in x and the Launch Site in y, with the hue representing the class (success or fail).

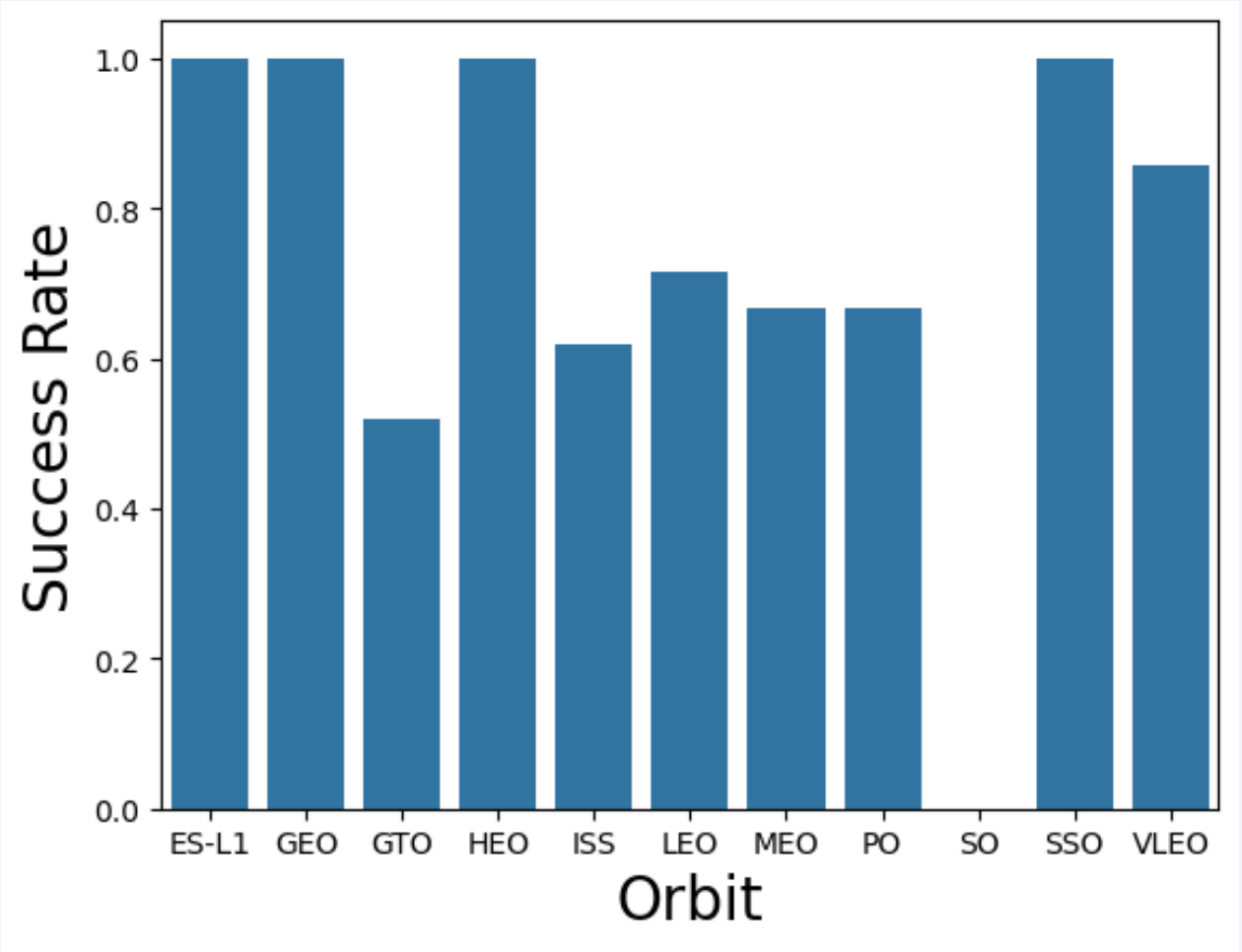
- The only relationship between launch sites and their payload mass is that for the VAFB SLC 4E, there are no rockets launched for heavy payload (greater tan 10000).



Success Rate vs. Orbit Type

- This bar plot shows the Orbit in x and the Success Rate for each orbit in y, which is the mean of class (success or fail).

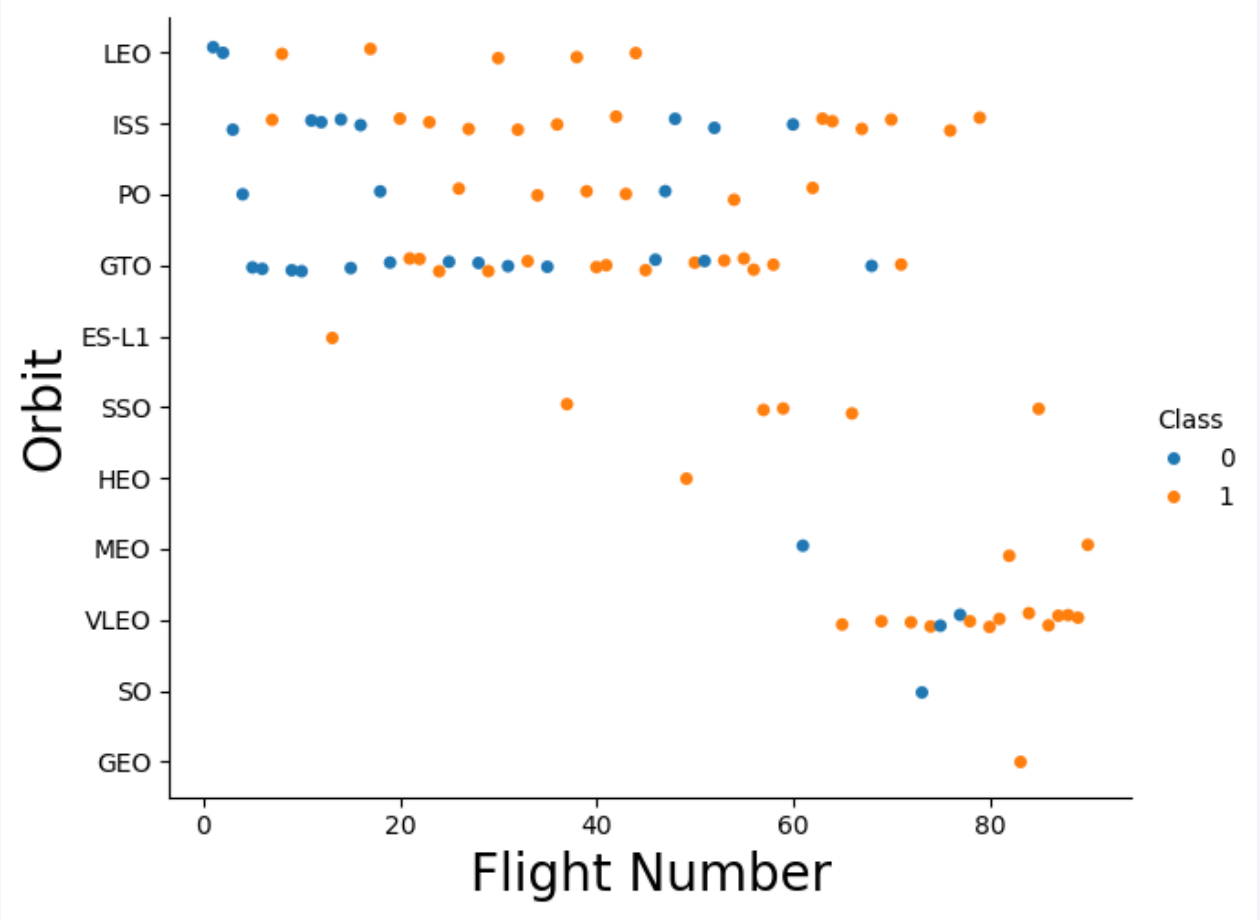
- We can see that the ES-L1, GEO, HEO and SSO orbits have highest success rate while SO has no successful landings.



Flight Number vs. Orbit Type

- This scatter plot shows the Flight Number in x and the Orbit in y, with the hue representing the class (success or fail).

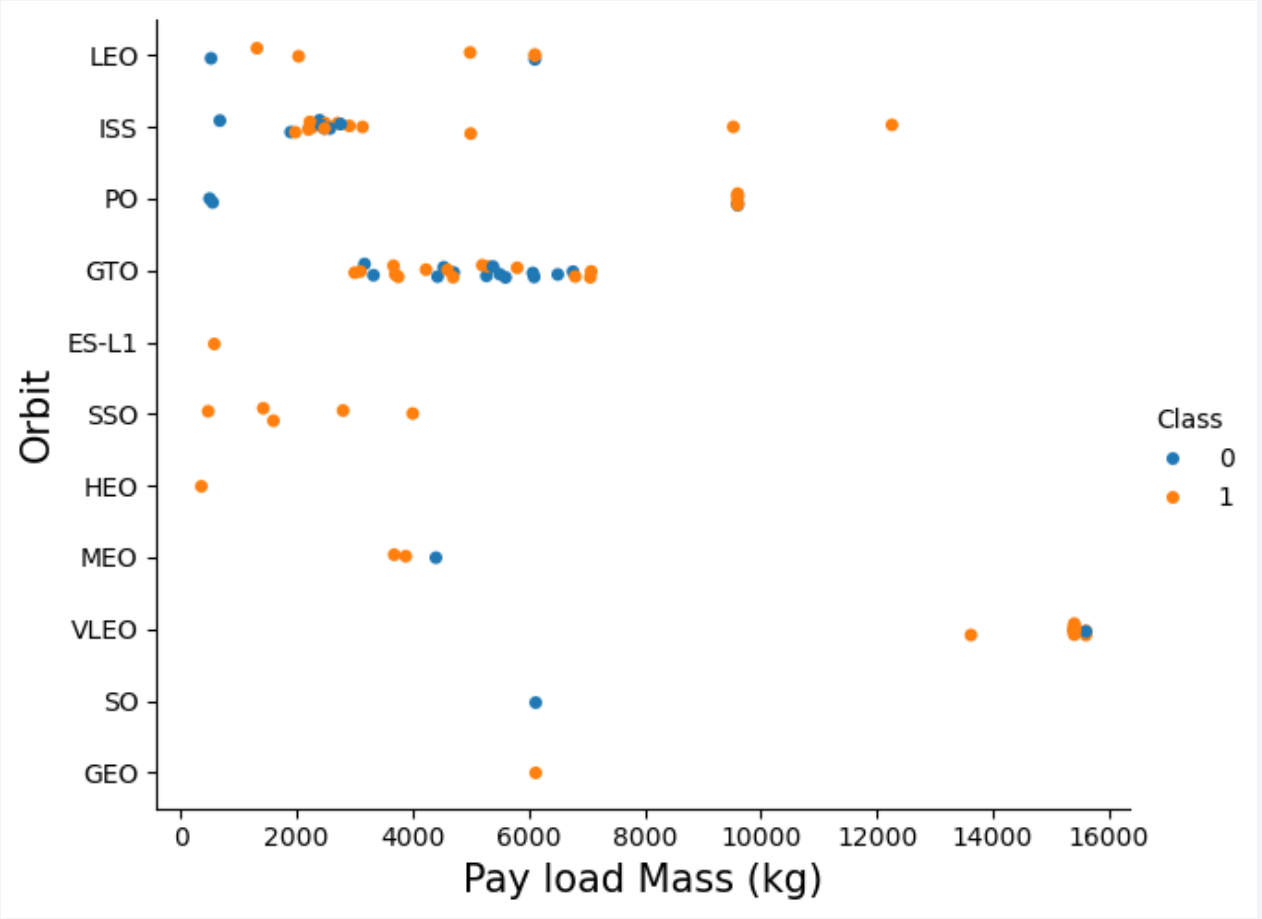
- We can see that mostly, the greater the fligth number, the greater the success rate, like with LEO. However, for example, there seems to be no relationship for GTO.



Payload vs. Orbit Type

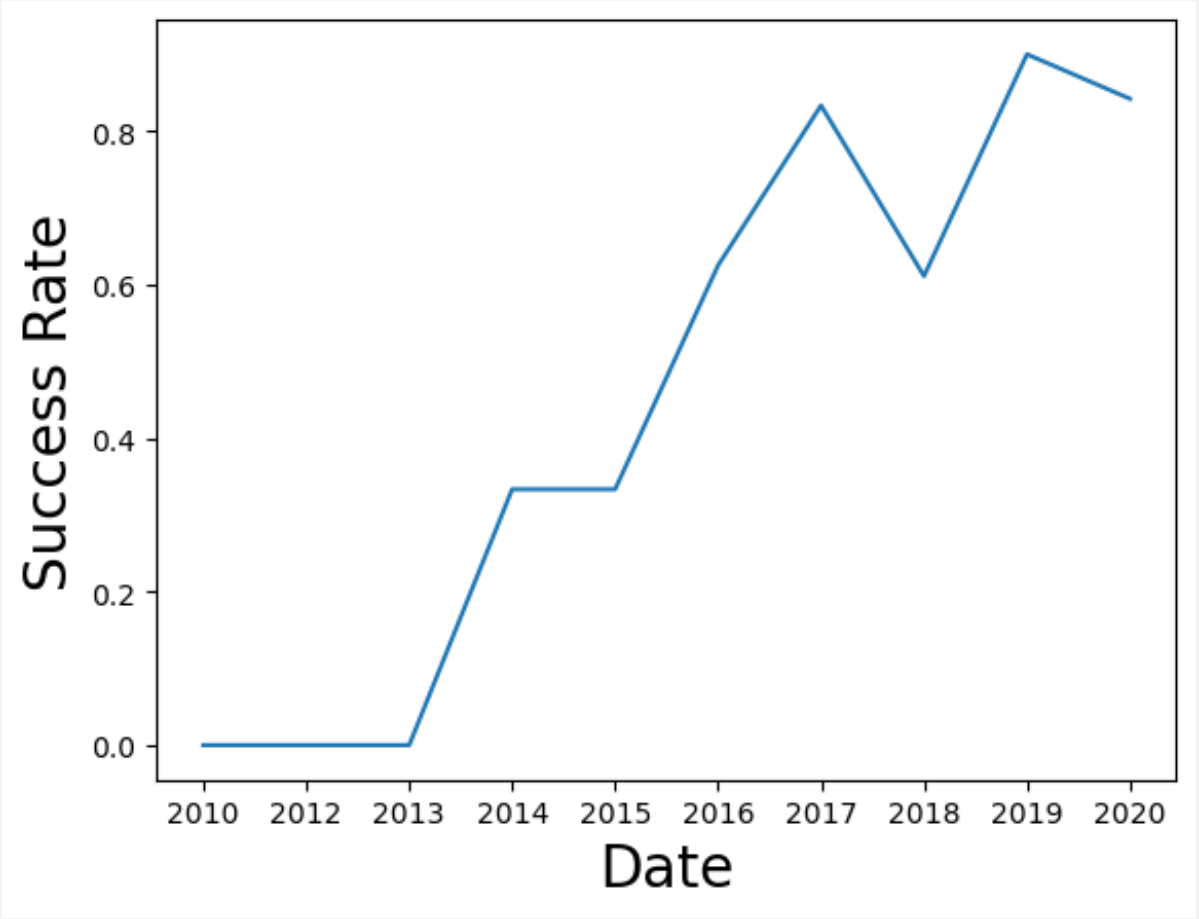
- This scatter plot shows the Payload mass in x and the Orbit in y, with the hue representing the class (success or fail).

- With heavy payloads, the successful landing or positive landing rate is higher for Polar, LEO and ISS. However, for GTO we can't discern this tren as both positive landing rates and negative outcomes are present.



Launch Success Yearly Trend

- This line chart shows the Date in x and the Success Rate in y, which is the mean of class (success or fail).
- The chart clearly shows that the success rate since 2013 kept increasing till 2020.



All Launch Site Names

SQL Query:

```
%sql SELECT DISTINCT(Launch_Site) FROM SPACEXTABLE
```

Description and output:

We used the DISTINCT function to find the all-unique Launch Site names from SPACEXTABLE

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

Launch Site Names Begin with 'CCA'

SQL Query:

```
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

Description and output:

We used the LIKE operator to filter the Launch Site which begin with ‘CCA’ and LIMIT operator to get only the first 5 records

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
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

Total Payload Mass

SQL Query:

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as 'Total payload mass (NASA (CRS))' FROM SPACEXTABLE WHERE Customer = 'NASA (CRS)'
```

Description and output:

We used the SUM function to calculate the total Payload Mass carried by boosters from the Customer NASA(CRS). We have also renamed the result column for better understanding.

Total payload mass (NASA (CRS))
45596

Average Payload Mass by F9 v1.1

SQL Query:

```
%sql SELECT ROUND(AVG(PAYLOAD_MASS__KG_), 2) as 'Total payload mass (Booster Version F9 v1.1)' FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 V1.1%'
```

Description and output:

We used the AVG function to calculate the mean Payload Mass carried by Booster Version F9 V1.1. We have also used the ROUND function with 2 for better presentation and get only 2 decimals.

Total payload mass (Booster Version F9 v1.1)
2534.67

First Successful Ground Landing Date

SQL Query:

```
%sql SELECT MIN(DATE) as 'First successful landing in ground pad' FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (ground pad)'
```

Description and output:

We used the MIN function to calculate the first successful Landing Outcome on ground pad. We have also renamed the result column for better understanding.

First successful landing in ground pad
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL Query:

```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

Description and output:

We used the WHERE operator to get the Successful drone ship landings, then combined with AND operator so we get the Payload Mass BETWEEN 4000 and 6000 Kg, finally show the corresponding boosters.

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 Query:

```
%%sql
SELECT
    COUNT(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 END) AS 'Successful Missions',
    COUNT(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 END) AS 'Failure Missions'
FROM SPACEXTABLE
```

Description and output:

We used the CASE clause within the COUNT functions to calculate the total number of successful and failure Mission Outcomes, since their format is not unique. We have also renamed the result column for better understanding.

Successful Missions	Failure Missions
100	1

Boosters Carried Maximum Payload

SQL Query:

```
%sql SELECT Booster_version as 'Boosters which carried max Payload' FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

Description and output:

We used the MAX function in the sub-query to get the maximum Payload Mass. Then we SELECT the Booster version which have carried this Payload. We have also renamed the result column for better understanding.

Boosters which carried max Payload
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

SQL Query:

```
%%sql
SELECT
    CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
    END AS 'Month', Landing_Outcome, Booster_version, Launch_site
FROM SPACEXTABLE
WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 0, 5) = '2015'
```

Description and output:

We used the WHERE operator to get the Failure drone ship landings, then combined with AND operator so we get the Date specifically filtered with substr(Date, 0, 5) to check if the year is 2015. Then, in the SELECT we use CASE with substr(Date, 6,2) to only print the Month name.

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

SQL Query:

```
%sql SELECT Landing_Outcome, COUNT(*) AS Outcome_Count FROM SPACEXTABLE WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY Outcome_Count DESC
```

Description and output:

We used the COUNT function to calculate total Landing Outcomes BETWEEN the date 2020-06-04 and 2017-03-20 GROUP BY Landing Outcome and ORDER BY the total outcome calculated in descending order.

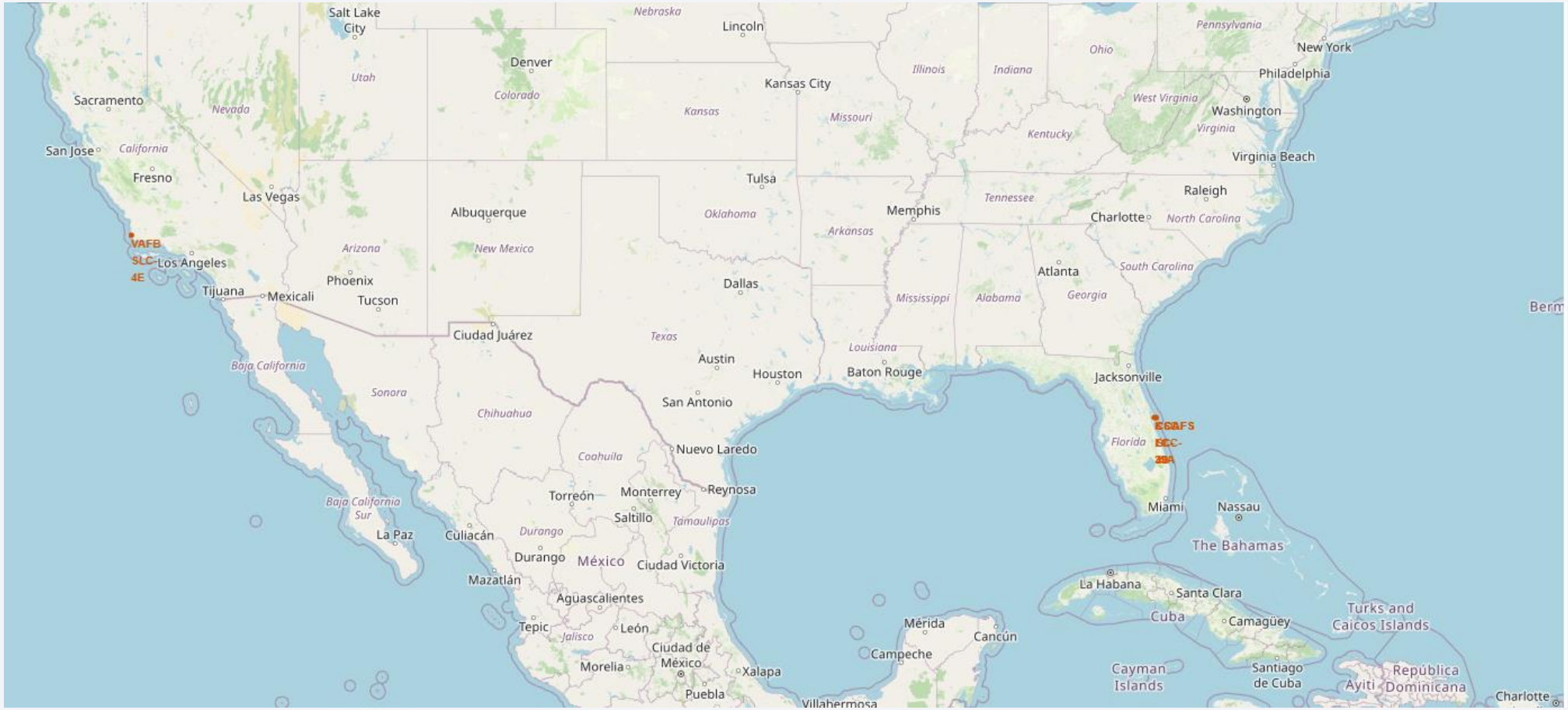
Landing_Outcome	Outcome_Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

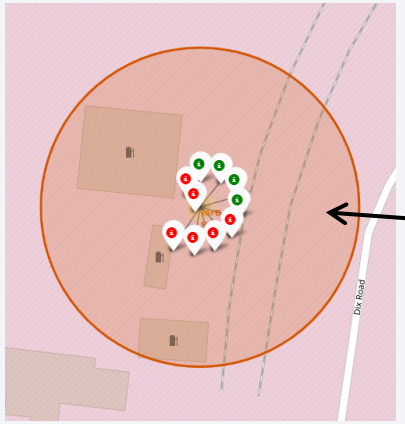
All Launch Sites on Folium Map

We can observe that all launch sites in the United States of America are very close to the coast and the close they can to the Equator line.

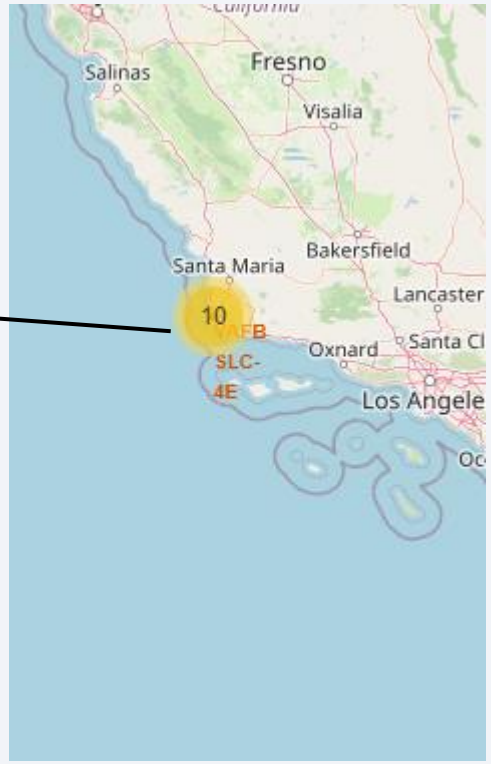


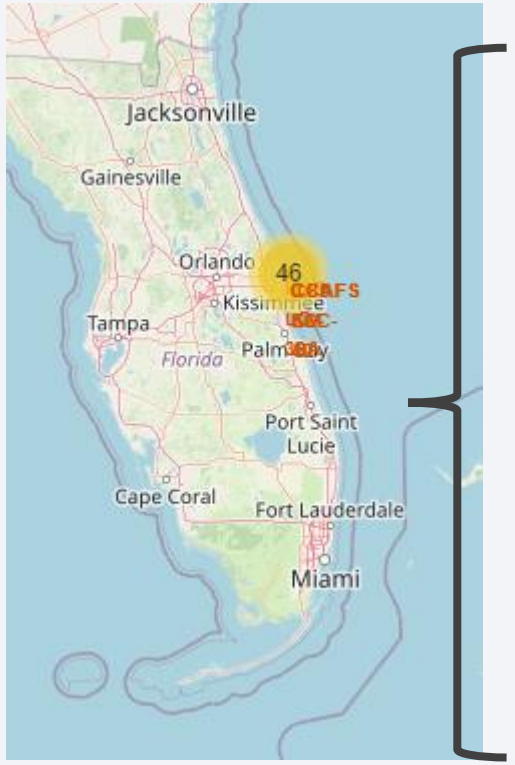
Markers for launch outcomes

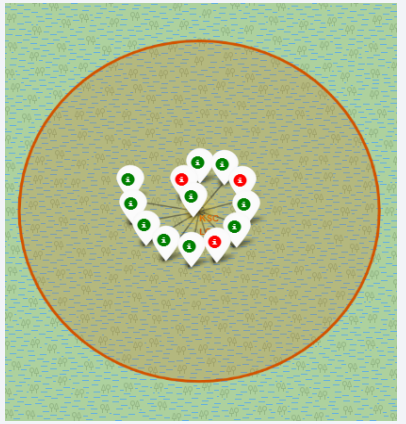
The launch outcomes are represented in marker clusters for each launch site. We can zoom in and see specific markers, green for successful ones and red for failed ones. With this, the launch site with more probability of success is KSC LC-39A



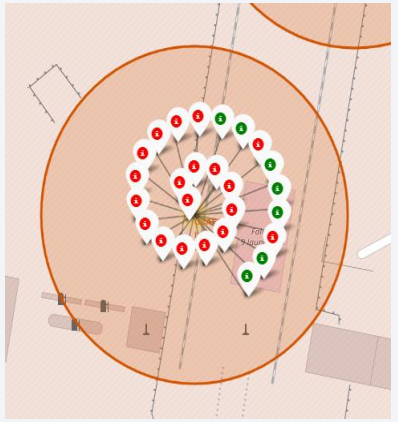
VAFB SLC-4E



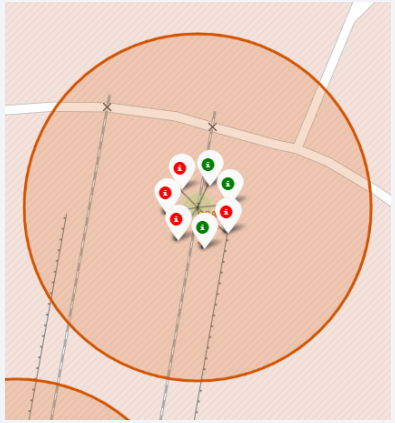




KSC LC-39A



CCAFS LC-40



CCAFS SLC-40

Distance between launch sites to landmarks

We calculated the distances between launch sites to railway, highway, coastline and cities.

- Are launch sites near railways?

Yes (e.g., 1.22 Km)

- Are launch sites near highways?

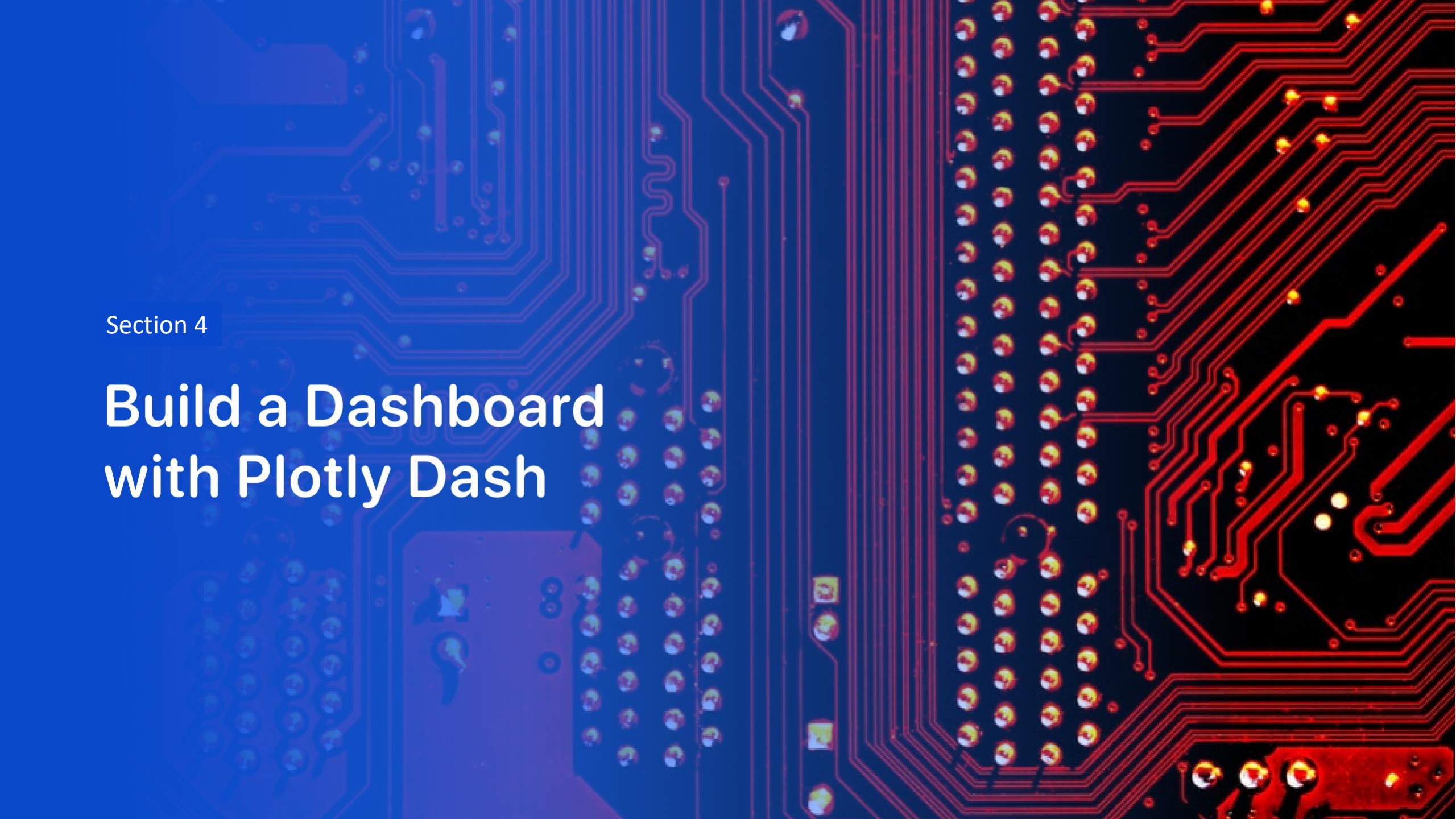
Yes (e.g., 0,98 Km)

- Are launch sites near coastline?

Yes (e.g., 0,85 Km)

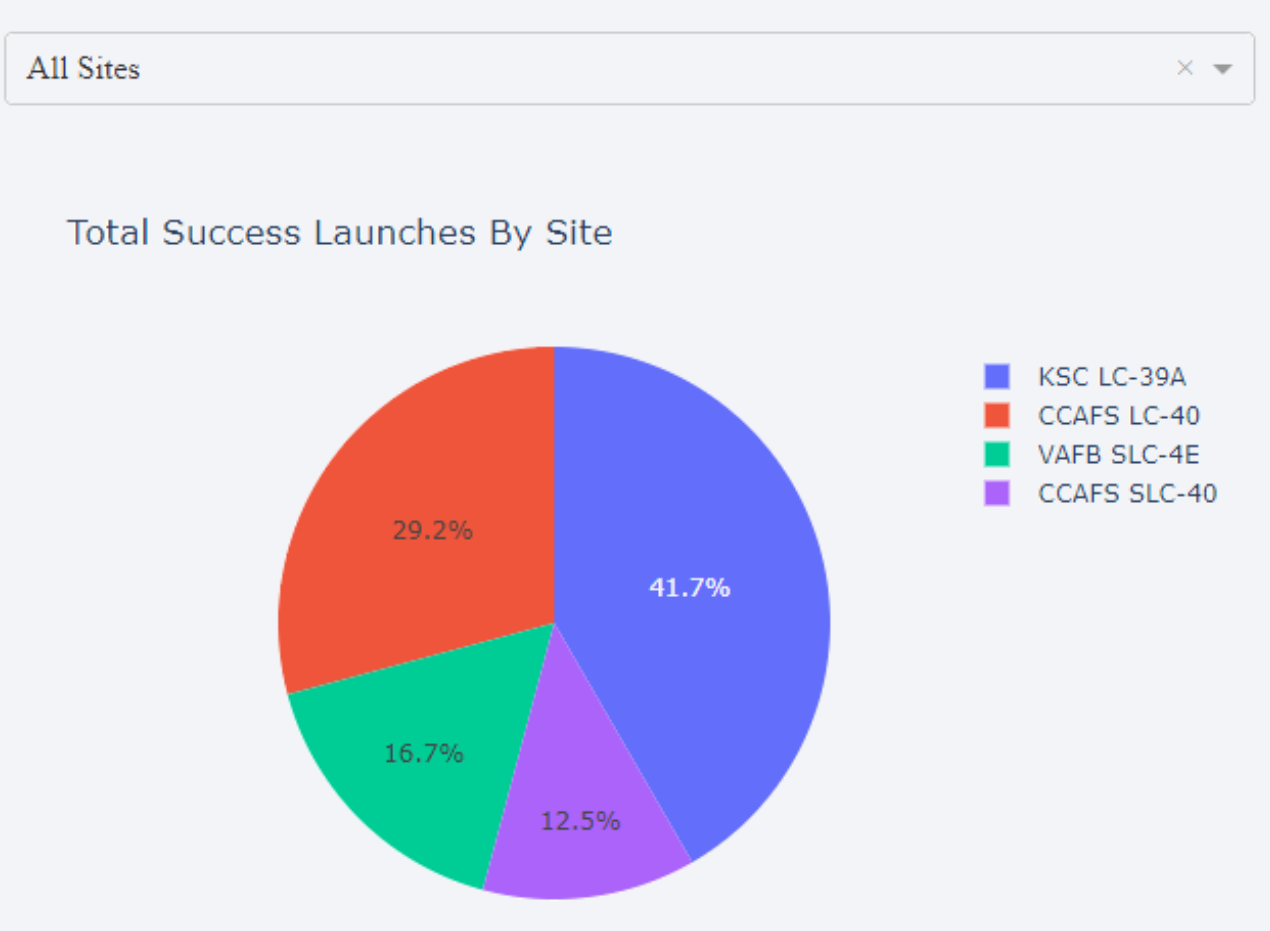
- Do launch sites keep certain distance away from cities?

Yes (e.g., 19,72 Km)


Section 4
Build a Dashboard with Plotly Dash

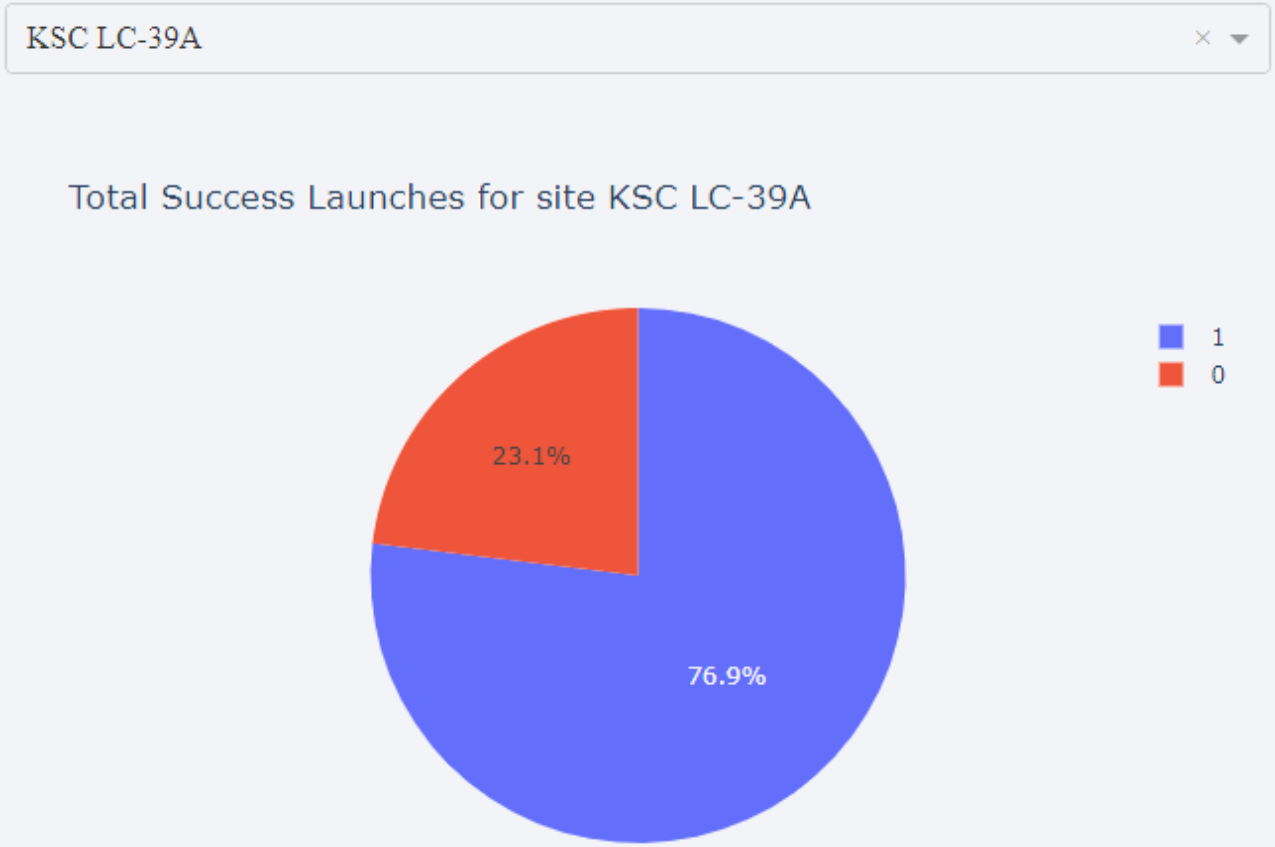
Launch success count of All Sites

We can easily observe that KSC LC-39A has the most successful launches rate from all the sites.

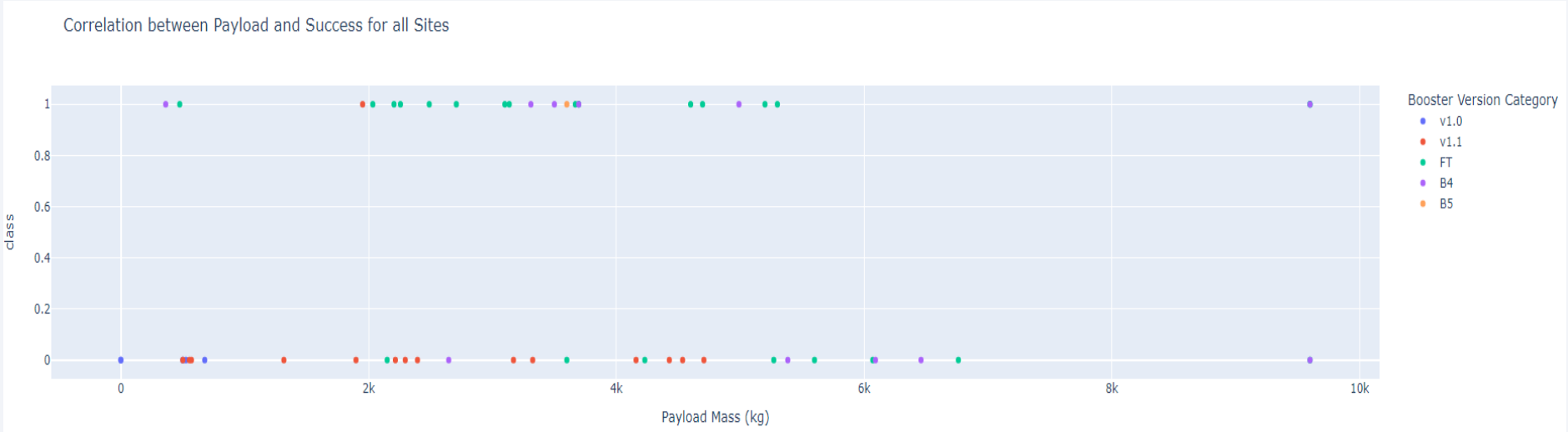


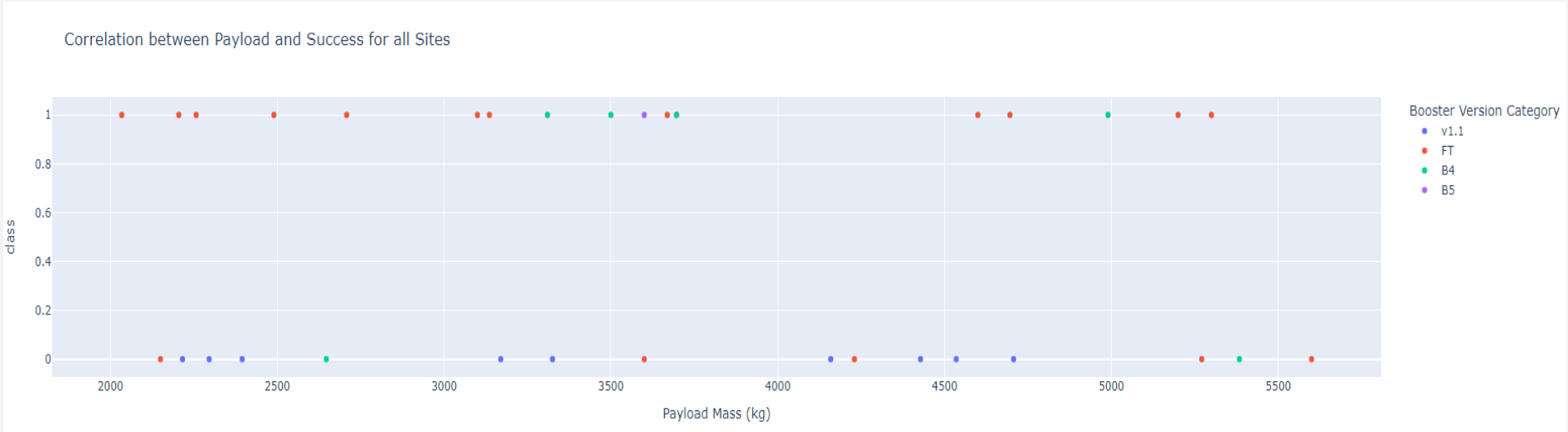
Launch Site with highest launch success ratio

We can identify from the specific pie chart of KSC LC-39A that has a 76.9% rate of successful launches



Payload vs. Launch Outcome for All Sites





With this dashboard we can observe these insights:

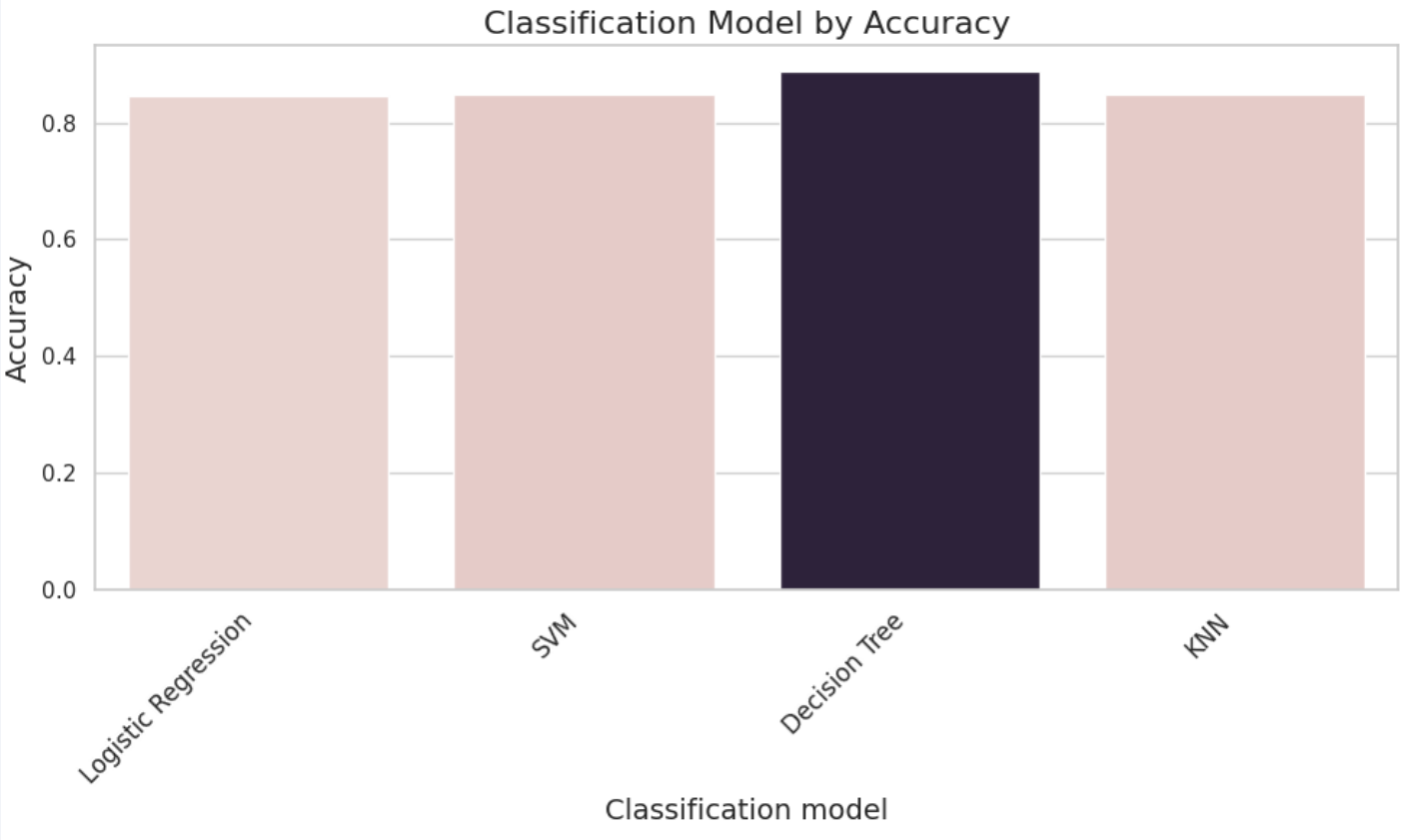
- Payload range with highest launch success: 2000-6000 Kg
- Booster version with highest launch success: FT

Section 5

Predictive Analysis (Classification)

Classification Accuracy

We can identify in this bar chart and DataFrame that “Decision Tree” is the classification model with the highest accuracy.



	Accuracy
Decision Tree	0.887500
KNN	0.848214
SVM	0.848214
Logistic Regression	0.846429

Confusion Matrix

- Confusion matrix helps to determine the performance of a classification model by determining accuracy, precision, recall, specificity and F1 Score.
- All classification models produced identical confusion matrices.

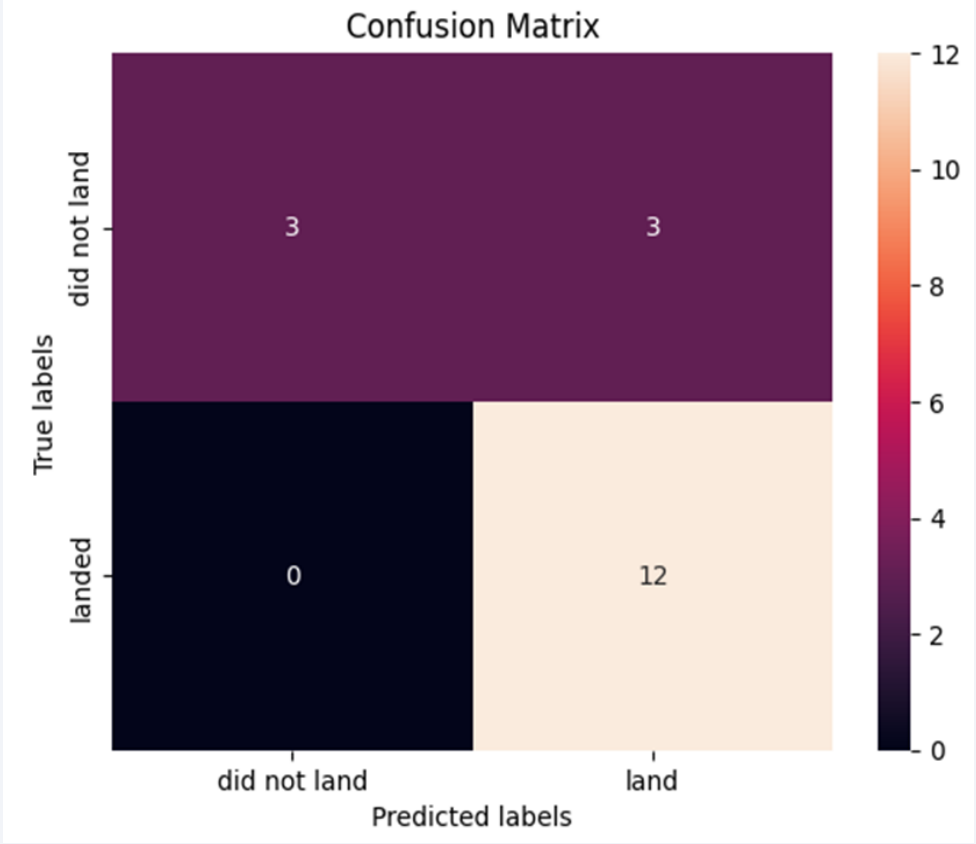
$$Precision = \frac{TP}{TP + FP} = \frac{3}{3 + 3} = 0{,}5$$

$$Recall = \frac{TP}{TP + FN} = \frac{3}{3 + 0} = 1$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = \frac{3 + 12}{3 + 12 + 3 + 0} = 0{,}83$$

$$Specifity = \frac{TN}{TN + FP} = \frac{12}{12 + 3} = 0{,}8$$

$$F1\ Score = 2 * \frac{Precision * Recazll}{Precision + Recall} = 2 * \frac{0{,}5 * 1}{0{,}5 + 1} = 0{,}67$$



Conclusions

- The success rate have been increasing by the time since 2013.
- The ES-L1, GEO, HEO and SSO orbits have the highest success rates.
- KSC LC-39A is the most successful launch site.
- The Payload range between 2000Kg and 6000Kg has the highest launch success.
- F9 FT booster version has the highest launch success.
- The Decision Tree is the best classification model for this problem.

Thank you!