SPACEX

Data Science Capstone Project

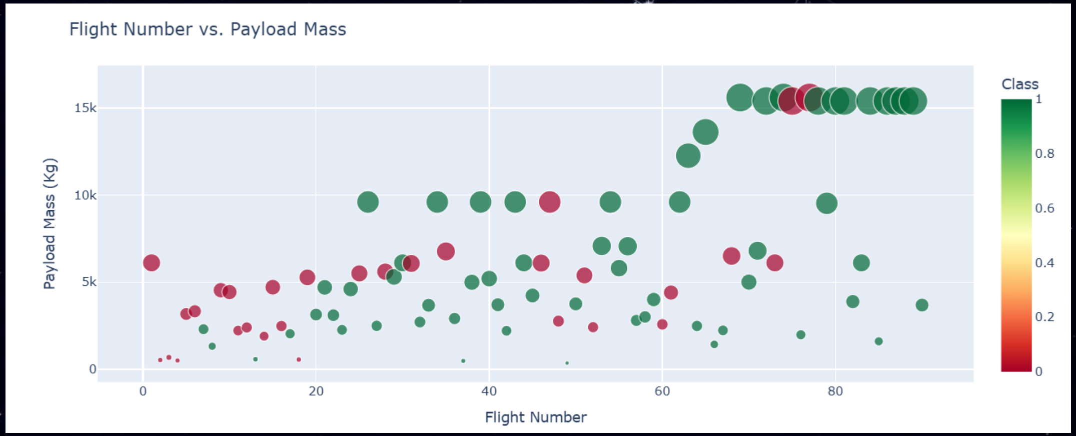
Aurthor: Duong Nhat Thanh

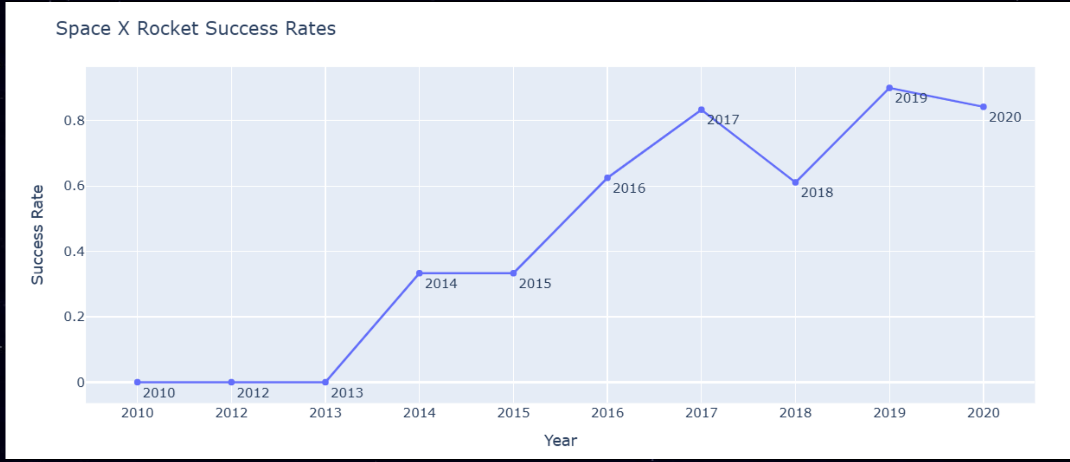
Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

- Summary of methodologies:
  - Data Collection via API, SQL and Web Scarping
  - Data Wrangling and Analysis
  - Interactive Maps and Folium
  - Predictive Analysis for each classification methods
- Summary of all results:
  - Data Analysis along with Interactive Visualizations
  - Best model for Predictive Analysis





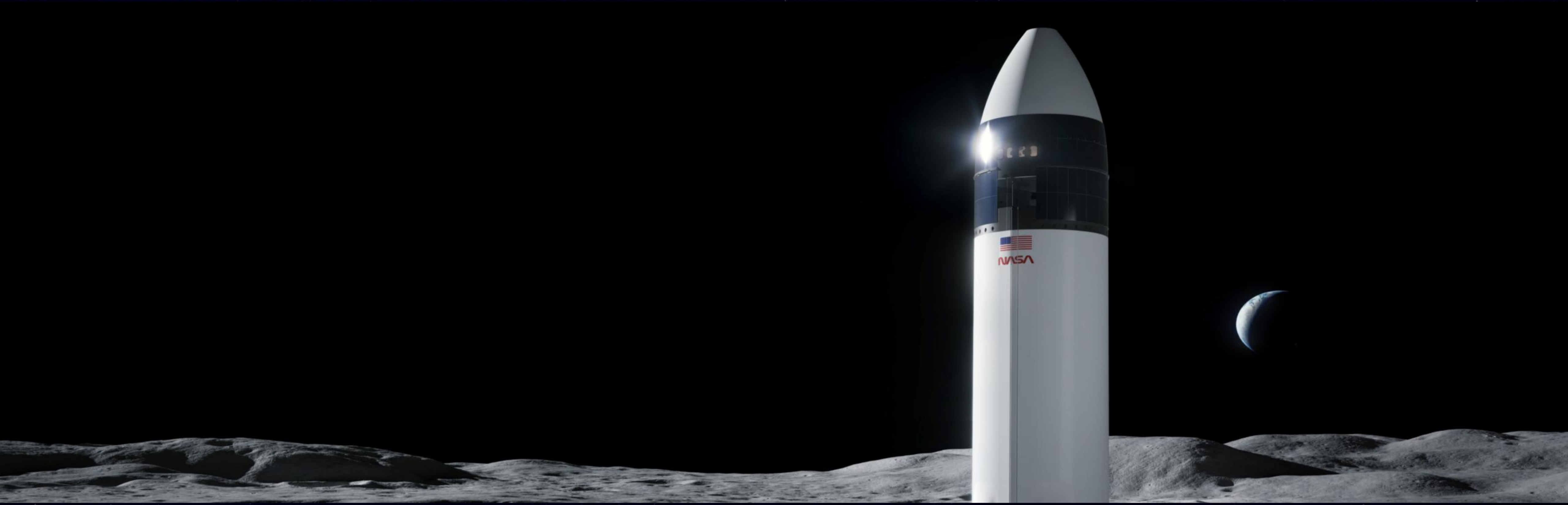
Introduction

- Project background and context:
    - Here we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land successfully. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.
- Problems we want to find answers:
    - With what factors, the rocket will land successfully?
    - The effect of each relationship of rocket variables on outcome.
    - Conditions which will aid SpaceX have to achieve the best results.

Methodology

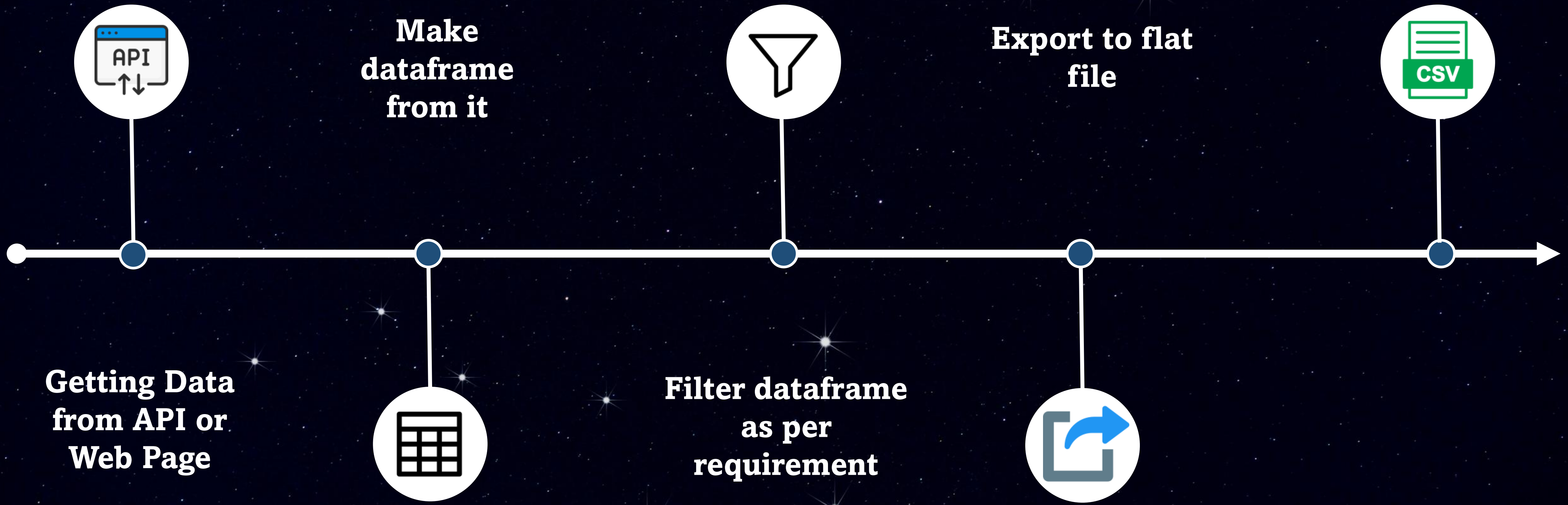
- Data collection methodology:
  - Via SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling:
  - One hot encoding data fields for machine learning and dropping irrelevant columns (Transforming data for Machine Learning)
- Perform exploratory data analysis (EDA) using visualization and SQL:
  - Scatter and bar graphs to show patterns between data
- Perform interactive visual analytics:
  - Using Folium and Plotly Dash Visualizations
- Perform predictive analysis using classification models:
  - Build and evaluate classification models

METHODOLOGY



Data Collection – Meaning & Basic steps

Data collection is the process of gathering and measuring information on targeted variables in an established system, which then enables one to answer relevant questions and evaluate outcomes.



Data Collection – Via SpaceX API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"

response = requests.get(spacex_url)
```

Getting Response from API → Converting Response to a .json file → Apply custom functions to clean data → Assign list to dictionary then create dataframe → Filter dataframe and export to flat file

```
jlist = requests.get(static_json_url).json()
df2 = pd.json_normalize(jlist)
df2.head()
```

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

```
launch_list = {'FlightNumber': list(data_falcon9['flight_number']),
               'Date': list(data_falcon9['date']),
               'BoosterVersion': BoosterVersion,
```

```
data_falcon9.drop(data_falcon9[data_falcon9['BoosterVersion'] != 'Falcon 9'].index, inpcae = True)
data_falcon9.loc[:, 'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0
5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0
6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0
7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0
8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0

Data Collection – Via Web Scraping

- Getting Response from HTML
- Creating BeautifulSoup Object
- Finding tables
- Getting column names
- Creation of dictionary and appending data to keys
- Converting dictionary to dataframe
- Dataframe to .CSV

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

data = requests.get(static_url).text
```

```
soup = BeautifulSoup(data, 'html5lib')
```

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

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

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

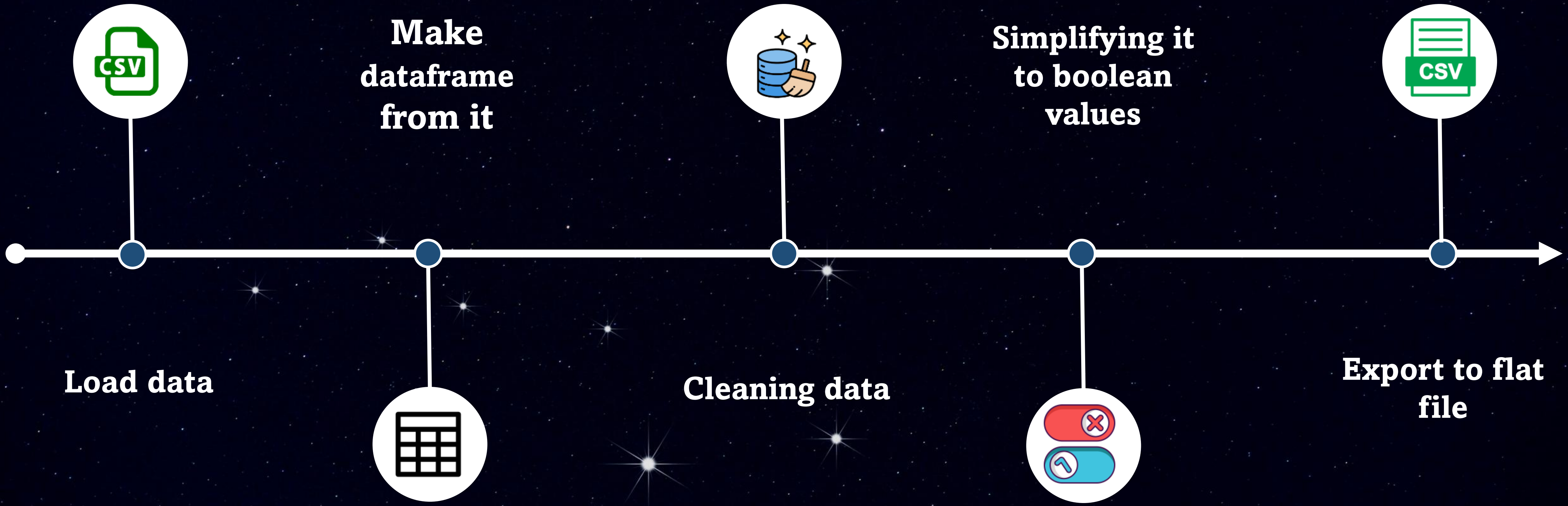
	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.07B0003.18	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.07B0004.18	Failure	8 December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

Data Wrangling – Meaning & Basic steps

Data wrangling is the process of cleaning and unifying messy and complex data sets for easy access and analysis.

Here we mainly convert those outcomes into Training Labels with 1 means the booster successfully landed 0 means it was unsuccessful.

```
df['Class'] = df['Outcome'].apply(lambda landing_class: 0 if landing_class in bad_outcomes else 1)
```



Data Wrangling

Calculate number of launches at each site

Calculate number and occurrence of each orbit

Calculate number and occurrence of mission outcome per orbit type

Create landing outcome label from Outcome column

Export dataset as .CSV

```
df['LaunchSite'].value_counts()

LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: count, dtype: int64
```

```
df['Orbit'].value_counts()

Orbit
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
HEO       1
ES-L1     1
SO        1
GEO       1
Name: count, dtype: int64
```

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

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

Outcome
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: count, dtype: int64
```

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0

EDA with SQL

SQL is an indispensable tool for Data Scientists and analysts as most of the real-world data is stored in databases. It's not only the standard language for Relational Database operations, but also an incredibly powerful tool for analyzing data and drawing useful insights from it. Here we store the data extracted using API to SQLite, which is a locally managed SQL Database.

```
import csv, sqlite3, pandas as pd

con = sqlite3.connect("my_data1.db")
cur = con.cursor()
%sql sqlite:///my_data1.db
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spac
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

We performed SQL queries to gather information from given dataset:

* Displaying the names of the unique launch sites in the space mission
* Display 5 records where launch sites begin with the string 'CCA'
* Displaying the total payload mass carried by boosters launched by NASA (CRS)
* Displaying average payload mass carried by booster version F9 v1.1
* Listing the date where the successful landing outcome in drone ship was achieved
* Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
* Listing the total number of successful and failure mission outcomes
* Listing the names of the booster_versions which have carried the maximum payload mass
* Listing the failed landing_outcomes in drone ship, their booster versions, and launch site names for the year 2015
* Ranking 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 makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. We use the latitude and longitude coordinates for each launch site and added a Circle Marker around each launch site with a label of the name of the launch site. It is also easy to visualize the number of success and failure for each launch site with Green and Red markers on the map.

Map Objects	Code	Results
Map Marker	folium.Marker(	Map object to make a mark on map
Icon Marker	folium.Icon(	Create an icon on map
Circle Marker	folium.Circle(	Create a circle where Marker is being placed
PolyLine	folium.PolyLine(	Create a line between points
Marker Cluster Object	MarkerCluster()	This is a good way to simplify a map containing many markers having the same coordinate
AntPath	folium.plugins.AntPath(	Create an animated line between points

Build an Interactive Map with Plotly Dash

Pie Chart showing the total success for all sites or by certain launch site

- *Percentage of success in relation to launch site*

Scatter Graph showing the correlation between Payload and Success for all sites or by certain launch site

- *It shows the relationship between Success rate and Booster Version Category.*

Map Objects	Code	Result
Dash and its components	import dash import dash_html_components as html import dash_core_components as dcc from dash.dependencies import Input, Output	Plotly stewards Python's leading data viz and UI libraries. With Dash Open Source, Dash apps run on your local laptop or server. The Dash Core Component library contains a set of higher-level components like sliders, graphs, dropdowns, tables, and more. Dash provides all of the available HTML tags as user-friendly Python classes.
Pandas	import pandas as pd	Fetching values from CSV and creating a dataframe
Plotly	import plotly.express as px	Plot the graphs with interactive plotly library
PolyLine	folium.PolyLine(	Create a line between points
Dropdown	dcc.Dropdown(	Create a dropdown for launch sites
Rangeslider	dcc.RangeSlider(	Create a rangeslider for Payload Mass range selection
Pie Chart	px.pie(	Creating the Pie graph for Success percentage display
Scatter Chart	px.scatter(	Creating the Scatter graph for correlation display

Predictive Analysis (Classification)

- Load our feature engineered data into dataframe
- Transform it into NumPy arrays
- Standardize and transform data
- Split data into training and test data sets
- Check how many test samples has been created
- List down machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the Grid
- SearchCV objects and train our model

```
y = data['Class'].to_numpy()
transform = preprocessing.StandardScaler()
X = transform.fit(X).transform(X)
X_train, X_test, Y_train, Y_test = train_test_split(X, y,
test_size=0.2, random_state=2)
Y_test.shape
```

```
algorithms = {'KNN': knn_cv.best_score_,
              'Decision Tree': tree_cv.best_score_,
              'Logistic Regression': logreg_cv.best_score_}
best_algorithm = max(algorithms, key=lambda x:
algorithms[x])
```

- The model with best accuracy score wins the best performing model

Building Model → Evaluating Model → Finding Best Performing Classification Model

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

- Check accuracy for each model
- Get best hyperparameters for each type of algorithms
- Plot Confusion Matrix

Result

Exploratory data analysis results

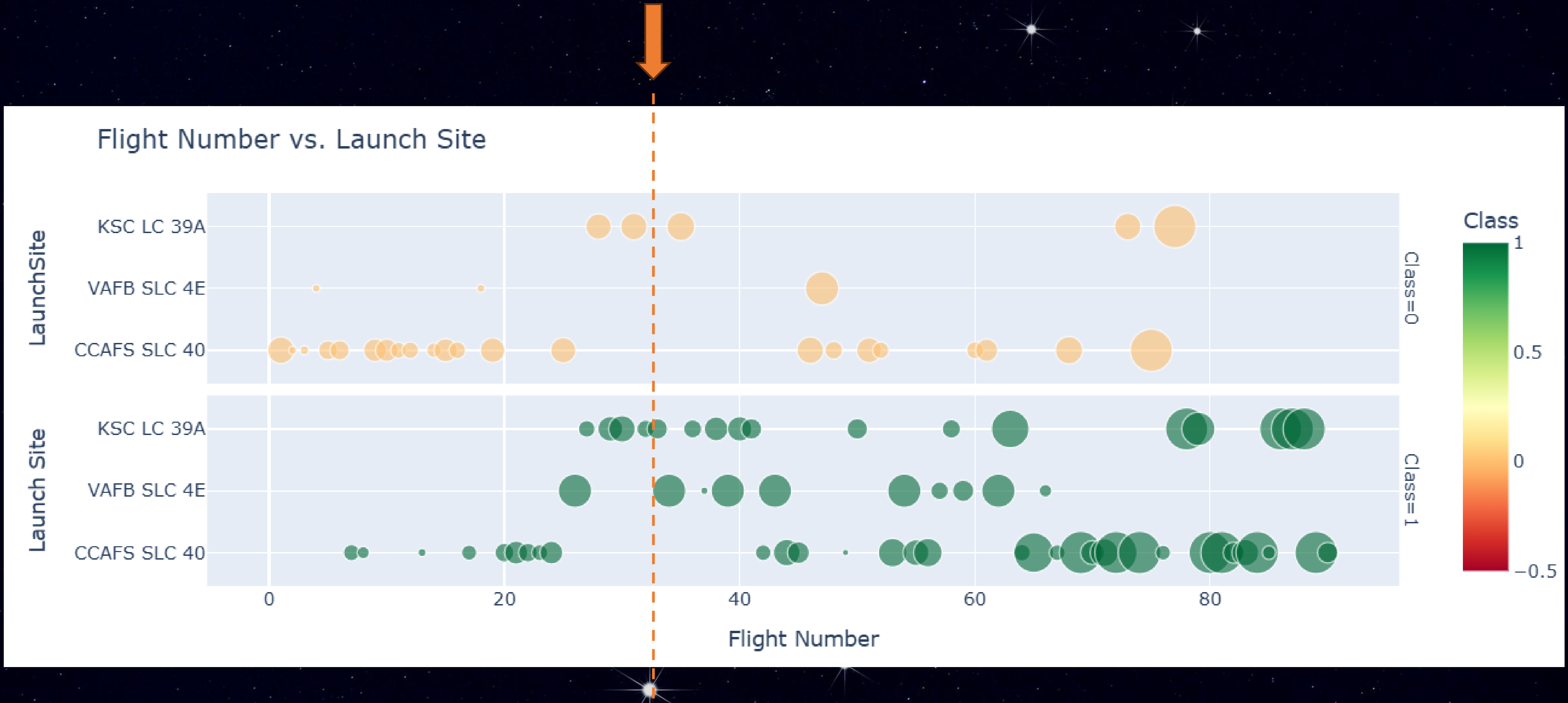
Interactive analytics demo in screenshots

Predictive analysis result

EDA with Visualization

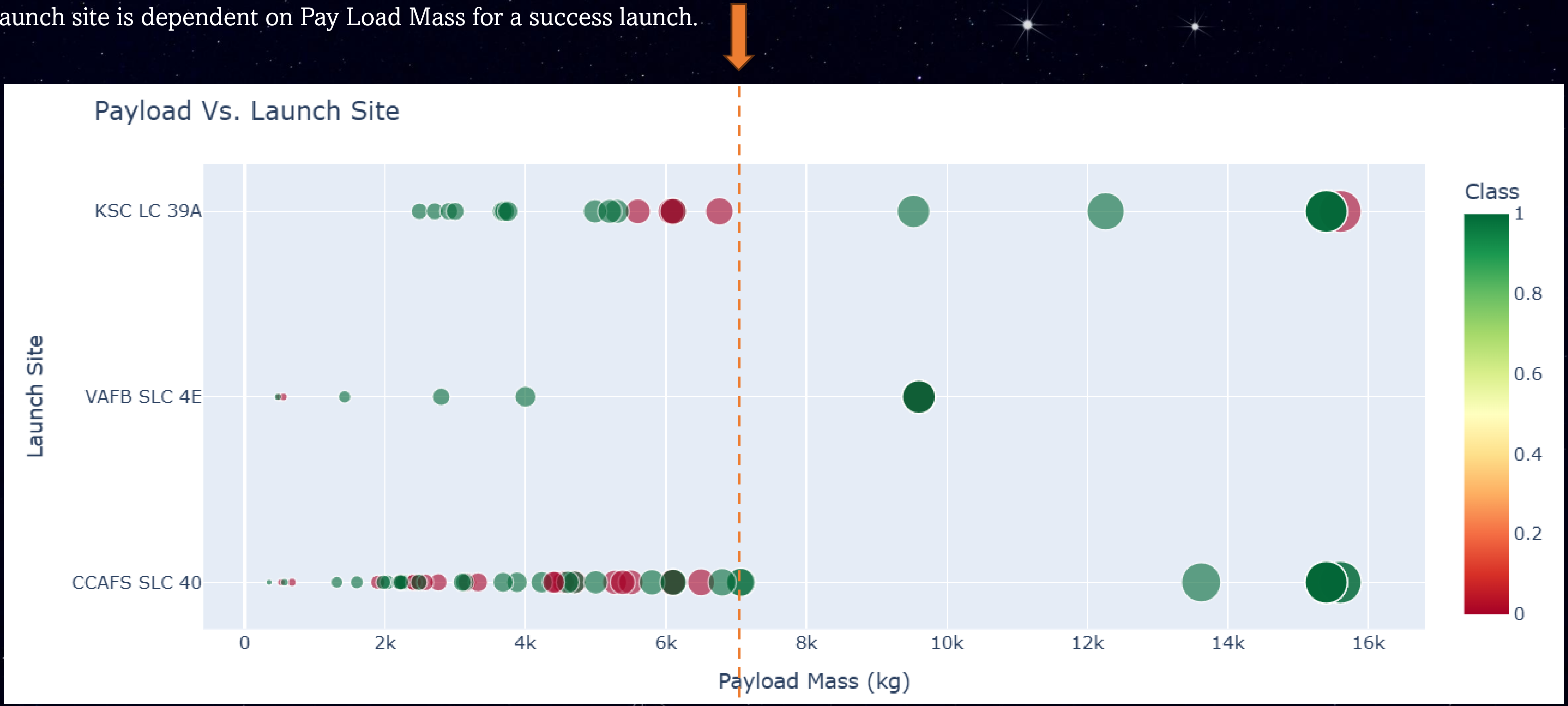
Flight Number vs. Launch Site

With higher flight numbers (greater than 30) the success rate for the Rocket is increasing.



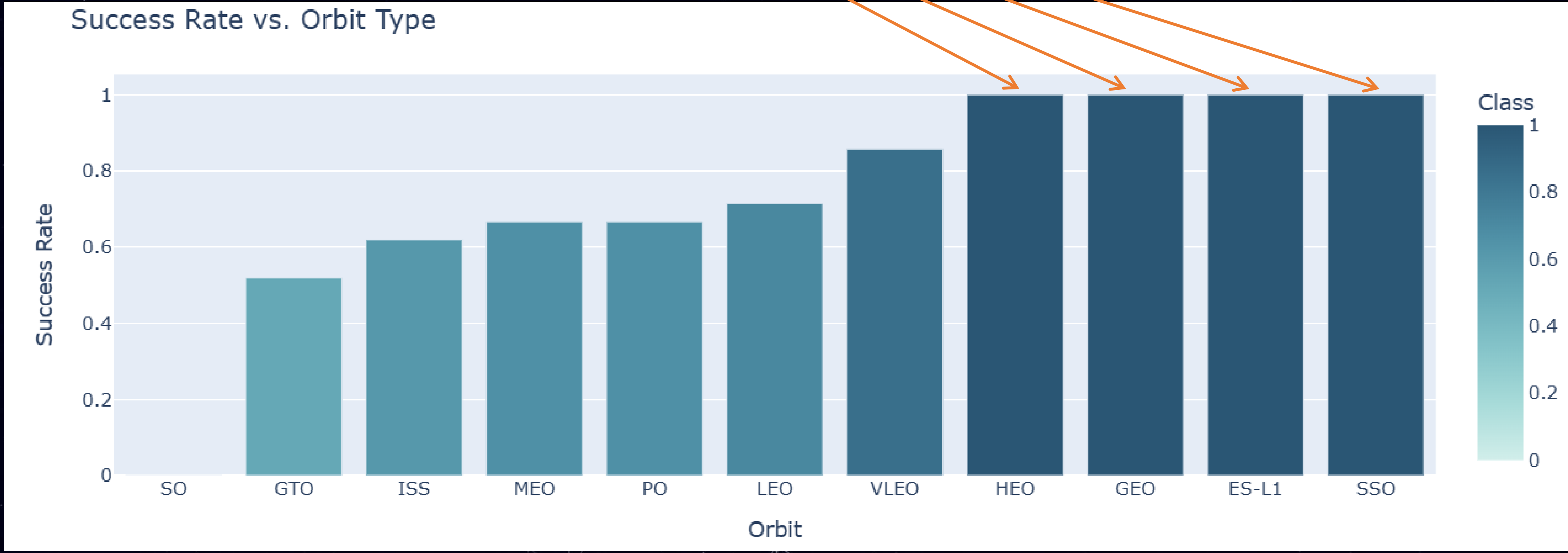
Payload vs. Launch Site

The greater the payload mass (greater than 7000 Kg) higher the success rate for the Rocket. But there's no clear pattern to take a decision, if the launch site is dependent on Pay Load Mass for a success launch.



Success Rate vs. Orbit Type

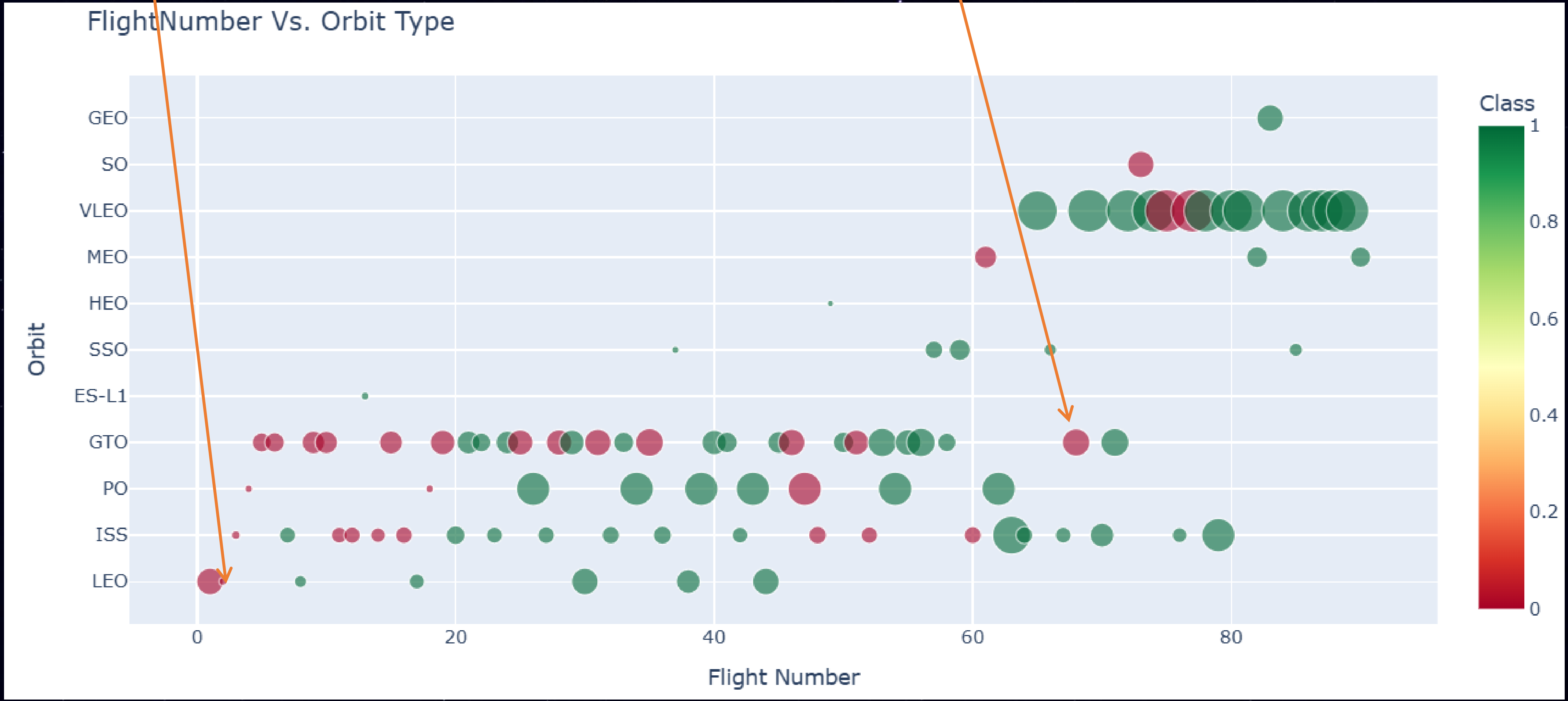
ES-L1, GEO, HEO, SSO has highest Sucess rates.



Flight Number vs. Orbit Type

We see that for **LEO** orbit the success increases with the number of flights

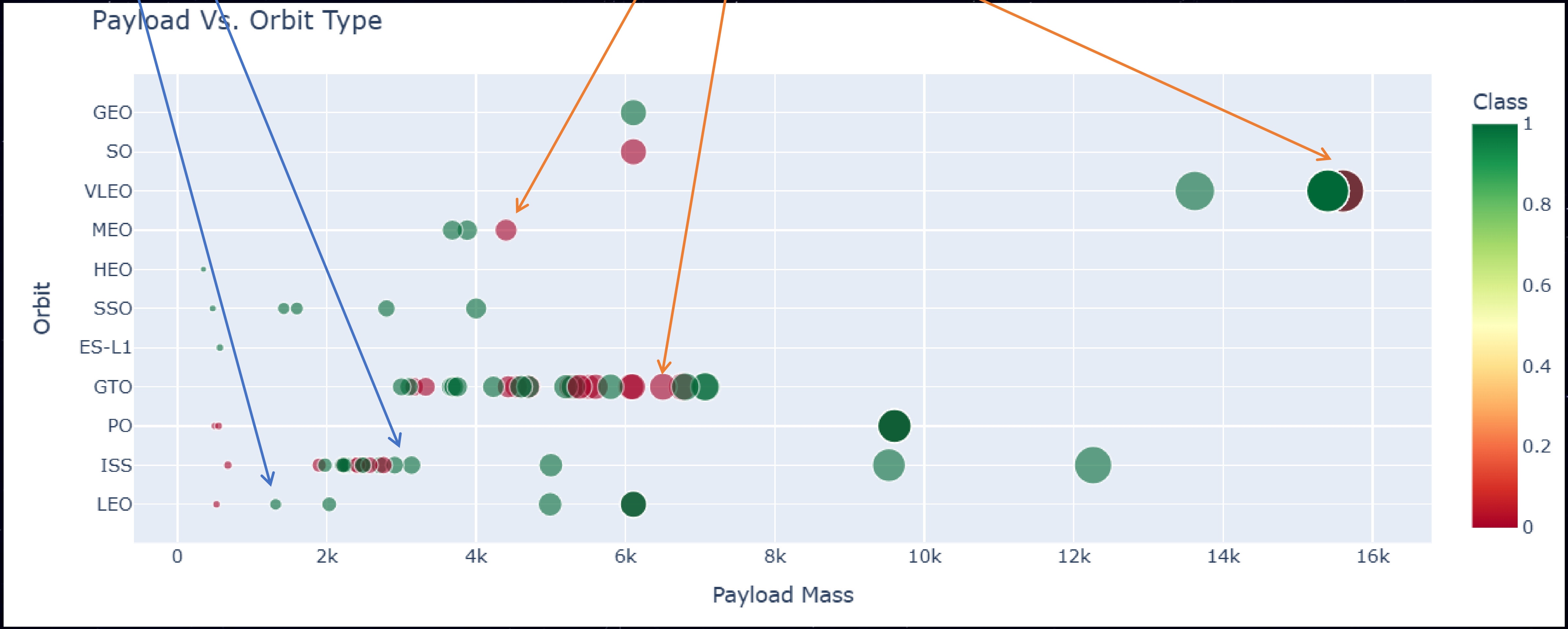
On the other hand, there seems to be no relationship between flight number and the **GTO** orbit.



Payload vs. Orbit Type

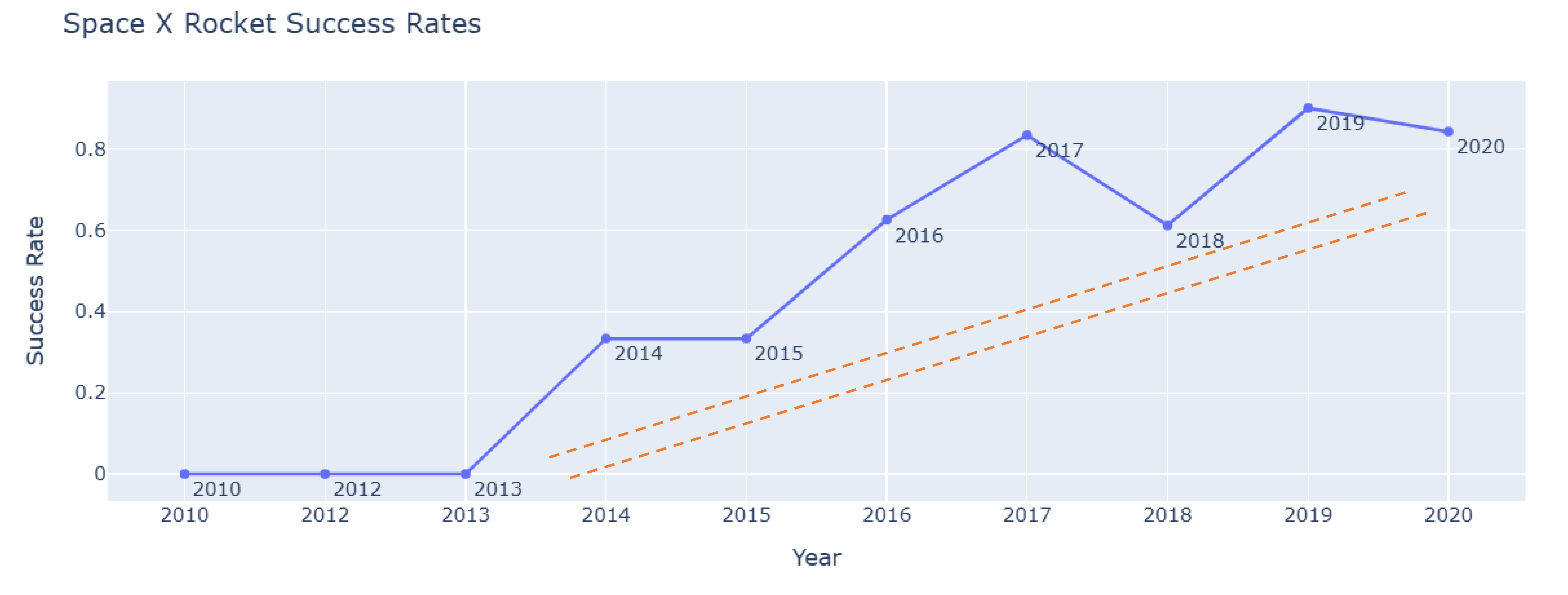
We observe that heavy payloads have a negative influence on **MEO**, **GTO**, **VLEO** orbits

Positive on **LEO**, **ISS** orbits

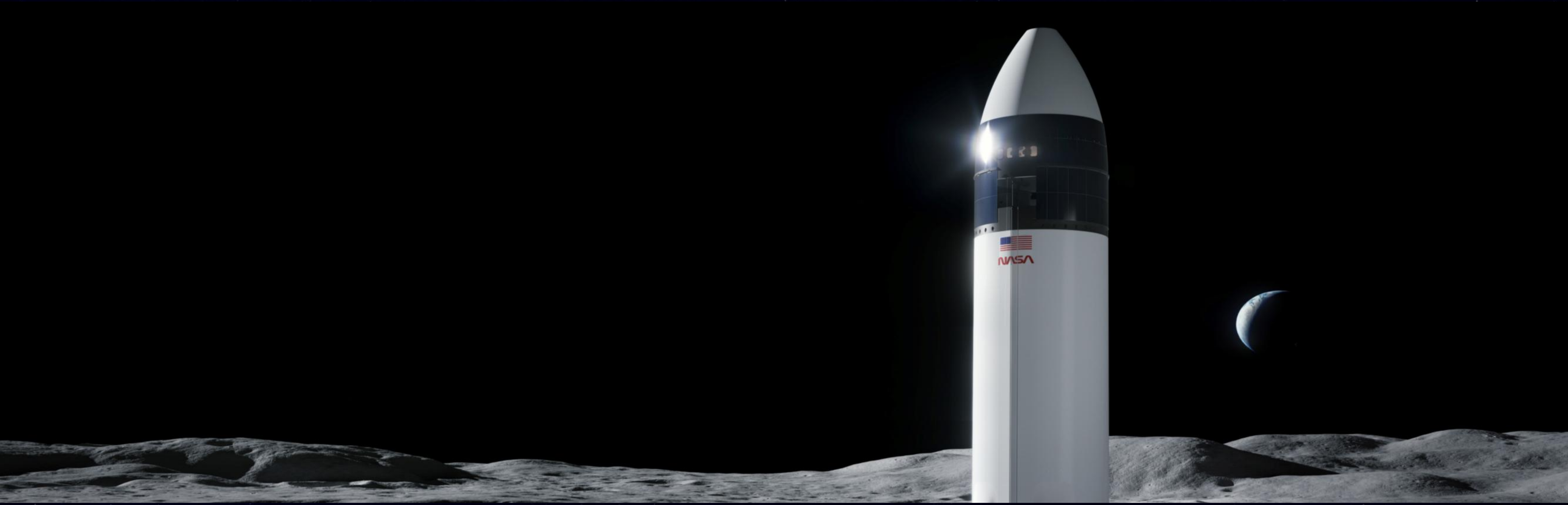


Launch Success Yearly Trend

We can observe that the success rate since 2013 kept increasing relatively though there is slight dip after 2019.



EDA with SQL



All Launch Site Names

SQL Query

```
SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTABLE
```

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

Description

Using the word DISTINCT in the query we pull unique values for Launch_Site column from table SPACEX.

Launch Site Names begin with 'CCA'

SQL Query

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

Description

Using keyword 'LIMIT 5' in the query we fetch 5 records from table spacex and with condition LIKE keyword with wildcard - 'CCA%'. The percentage in the end suggests that the Launch_Site name must start with CCA.

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

SQL Query

```
SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTABLE WHERE CUSTOMER = 'NASA (CRS)';
```

Description

Using the function SUM calculates the total in the column PAYLOAD_MASS_KG_ and WHERE clause filters the data to fetch Customer's by name "NASA (CRS)".

Total Payload Mass by NASA (CRS)
45596

Average Payload Mass by F9 v1.1

SQL Query

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS "Average Payload Mass by Booster Version F9 v1.1" FROM SPACEXTABLE \
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Description

Using the function AVG works out the average in the column PAYLOAD_MASS_KG_The WHERE clause filters the dataset to only perform calculations on Booster_version "F9 v1.1".

Average Payload Mass by Booster Version F9 v1.1
2928.4

First Successful Ground Landing Date

SQL Query

```
%sql SELECT MIN(DATE) AS "First Succesful Landing Outcome in Ground Pad" FROM SPACEXTABLE \
WHERE Landing_Outcome = 'Success (ground pad)';
```

Description

Using the function MIN works out the minimum date in the column Date and WHERE clause filters the data to only perform calculations on Landing_Outcome with values "Success (ground pad)".

First Succesful Landing Outcome 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_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

Description

Selecting only Booster_Version,

WHERE clause filters the dataset to Landing_Outcome = Success (drone ship)

AND clause specifies additional filter conditions

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 Query

```
%sql SELECT sum(case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end) AS "Successful Mission", \
    sum(case when MISSION_OUTCOME LIKE '%Failure%' then 1 else 0 end) AS "Failure Mission" \
FROM SPACEXTABLE;
```

Description

Selecting multiple count is a complex query. I have used case clause within sub query for getting both success and failure counts in same query.

Case when MISSION_OUTCOME LIKE '%Success%' then 1 else 0 end" returns a Boolean value which we sum to get the result needed.

Successful Mission	Failure Mission
100	1

Total number of Successful and Failure Mission Outcomes

SQL Query

```
%sql SELECT DISTINCT BOOSTER_VERSION AS "Booster Versions which carried the Maximum Payload Mass" FROM SPACEXTABLE \
WHERE PAYLOAD_MASS__KG_ =(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE);
```

Description

Using the function MAX works out the maximum payload in the column PAYLOAD_MASS__KG_ in sub query.

WHERE clause filters Booster Version which had that maximum payload.

Booster Versions which carried the Maximum Payload Mass
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 strftime('%m', DATE) AS Month, BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTABLE \
WHERE strftime('%Y', DATE) = '2015' AND Landing_Outcome = 'Failure (drone ship)';
```

Description

We need to 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.

Via year function we extract the year and future where clause 'Failure (drone ship)' fetches our required values.

Also, am using {fn MONTHNAME(DATE)} to get the Month name.

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

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

SQL Query

```
%sql SELECT Landing_Outcome as "Landing_Outcome", COUNT(Landing_Outcome) AS "Total Count" FROM SPACEXTABLE \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' \
GROUP BY  Landing_Outcome \
ORDER BY COUNT(Landing_Outcome) DESC ;
```

Description

Selecting only LANDING__OUTCOME,WHERE clause filters the data with DATE BETWEEN '2010-06-04' AND '2017-03-20’

Grouping by LANDING__OUTCOMEOrder by COUNT(LANDING__OUTCOME) in Descending Order.

Landing_Outcome	Total 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

Rank Success Count between 2010-06-04 and 2017-03-20

SQL Query

```
%sql SELECT COUNT(Landing_Outcome) AS "Rank success count between 2010-06-04 and 2017-03-20" FROM SPACEXTABLE \
WHERE Landing_Outcome LIKE '%Success%' AND DATE > '2010-06-04' AND DATE < '2017-03-20' ;
```

Description

COUNT counts records in column LANDING__OUTCOME

WHERE filters data with '%Success%'

AND DATE > '2010-06-04'

AND DATE < '2017-03-20'

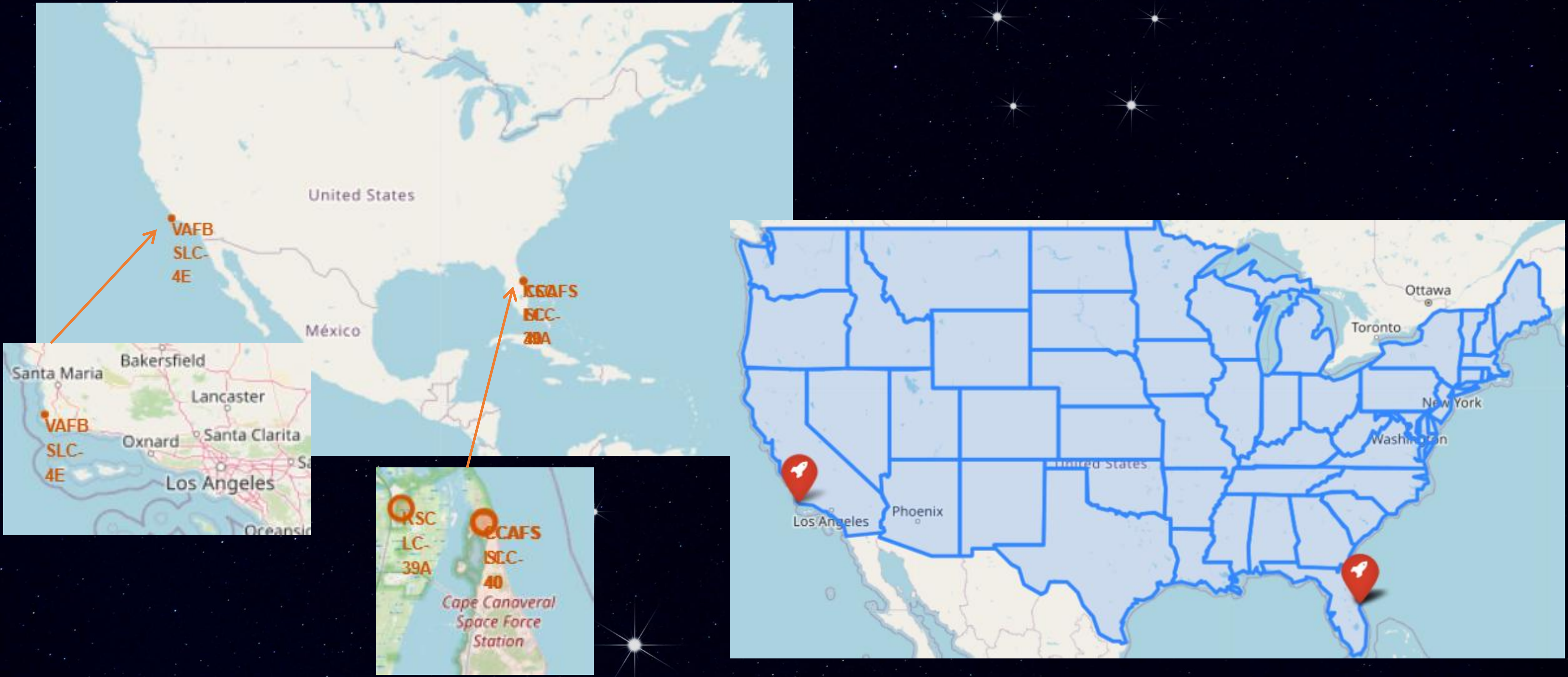
Rank success count between 2010-06-04 and 2017-03-20
8

Interactive map

with Folium

All Launch Sites on Folium Map

We can see that the SpaceX launch sites are near to the United States of America coasts i.e., Florida and California Regions.

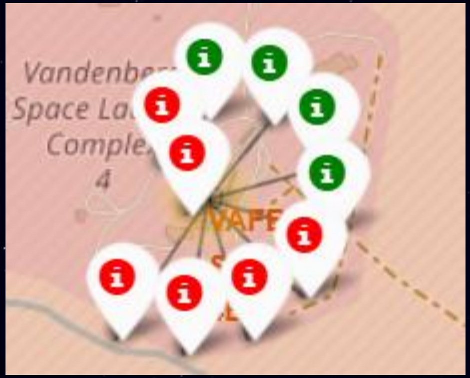


Color Labeled Launch Records

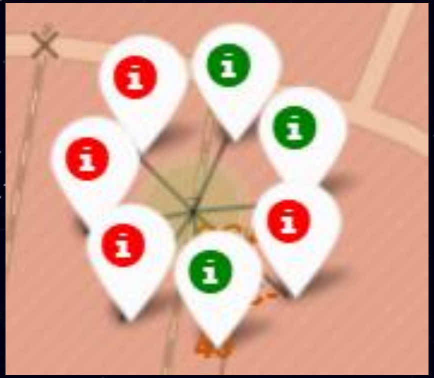
Green Marker shows successful launches and **Red Marker** shows failures.

From these screenshots **its easily understandable that KSC LC-39A has the maximum probability of success.**

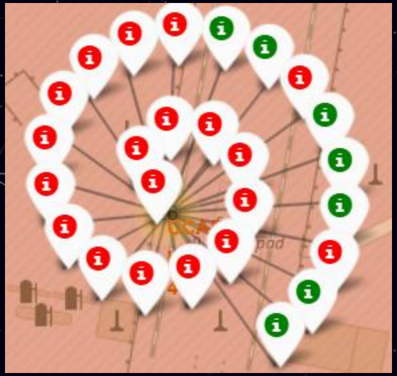
VAFB SLC-4E



CCAFS SLC-40



CCAFS LC-40

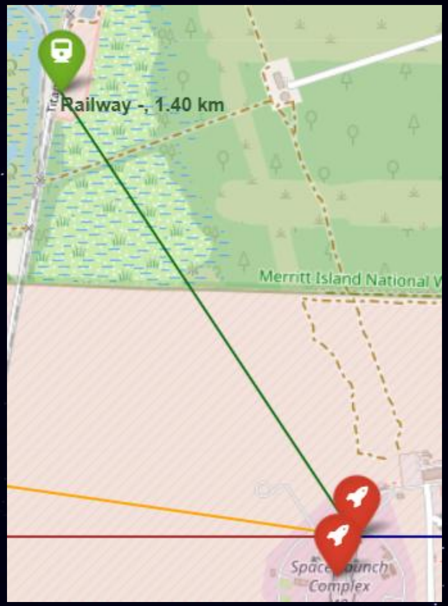


KSC LC-39A

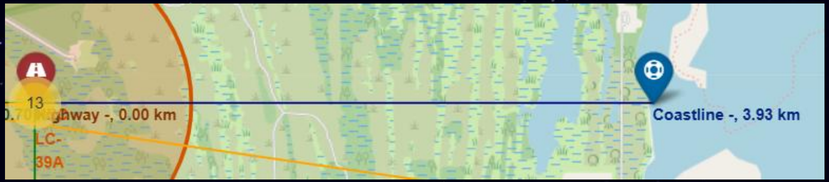
Launch Site Distances from Equator & Railways

Distance from Equator is greater than 3000 Km for all sites.

Distance for all launch sites from railway tracks are greater than .7 Km for all sites. So, launch sites are not so far away from railway tracks.



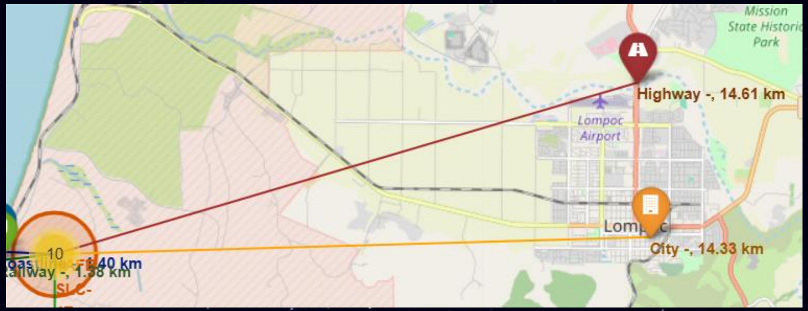
Launch Site Distances from Coastlines & Cities

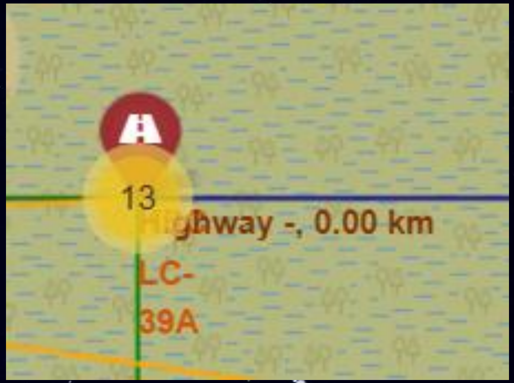


Distance for all launch sites from coastline is less than 4 Km.

Distance for all launch sites from cities is greater than 14 Km for all sites. So, launch sites are far away from cities.

Launch Site Distances from Highways





Distance for all launch sites from cities is greater than 14 Km for all sites. So, launch sites are far away from cities.

Conclusion:

- Are all launch sites in proximity to the Equator line?

No (4000 Km > distance > 3000 Km)

- Are launch sites in close proximity to railways?

Yes (2 Km > distance > .5 Km)

- Are launch sites in close proximity to highways?

No (15 Km > distance > 5 Km)

- Are launch sites in close proximity to coastline?

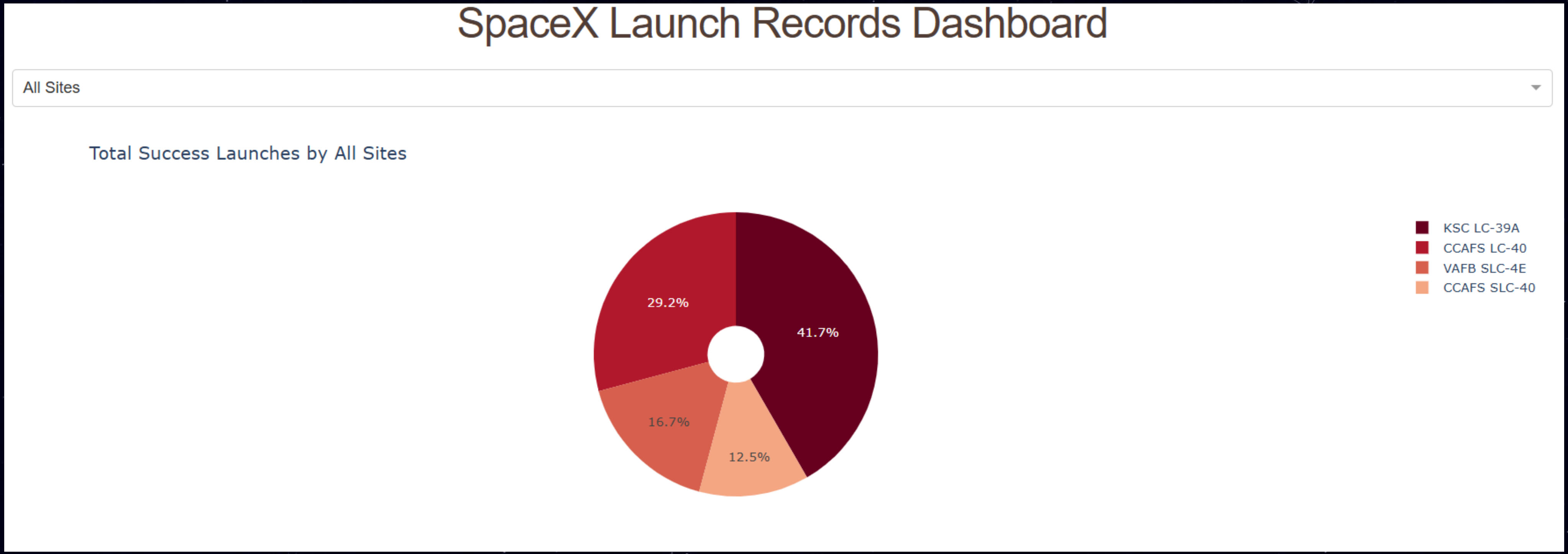
Yes (5 Km > distance > .5 Km)

- Do launch sites keep certain distance away from cities?

Yes (15 Km > distance > 80 Km)

Build a Dashboard with Plotly Dash

Launch Success Count for All Sites

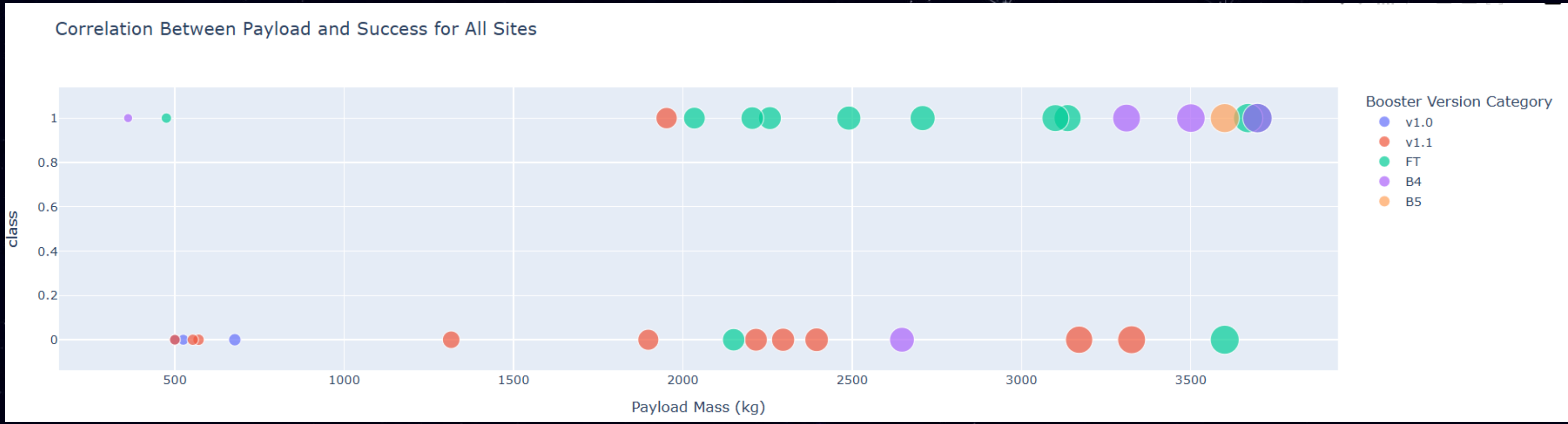


We can see that KSC LC-39A had the most successful launches from all the sites.

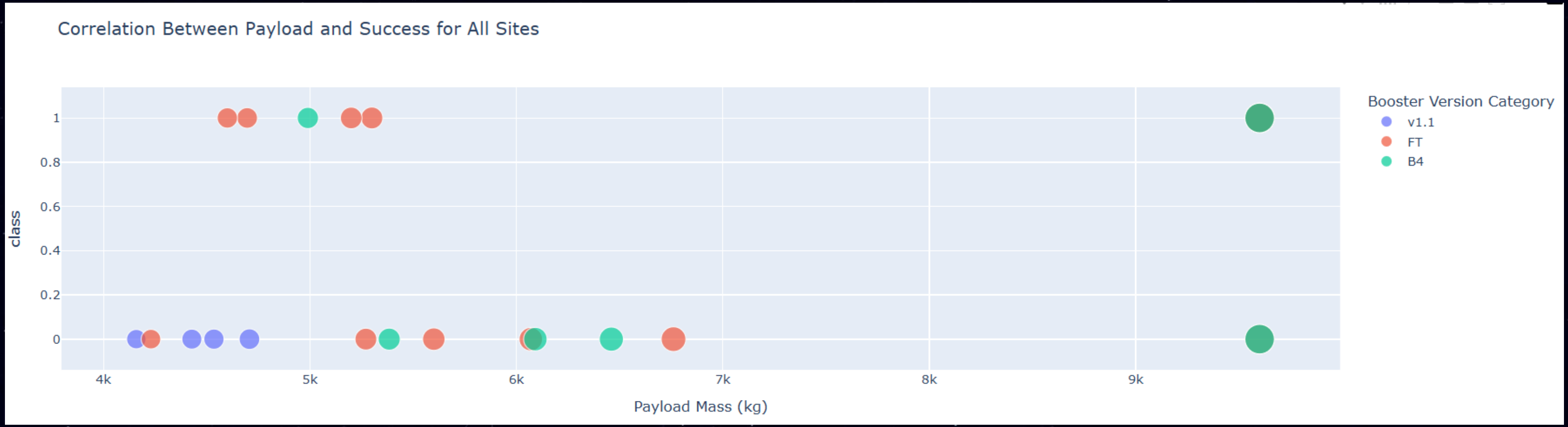
Payload vs. Launch Outcome Scatter Plot for All Sites

We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Low Weighted Payload 0kg – 4000kg

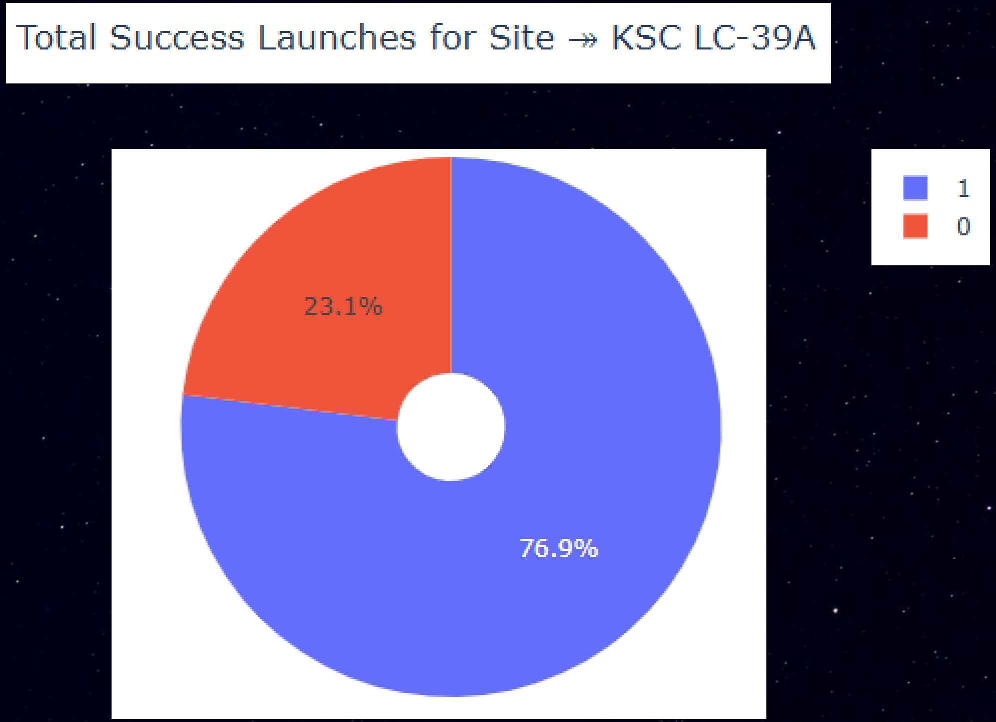


Low Weighted Payload 4000kg – 10000kg



Payload vs. Launch Outcome Scatter Plot for All Sites

We can see the success rates for low weighted payloads is higher than the heavy weighted payloads



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

After visual analysis using the dashboard, we are able to obtain some insights to answer these questions:

- Which site has the highest launch success rate?

KSC LC – 39A

- Which payload range(s) has the highest launch success rate?

2000 Kg – 10000 Kg

- Which payload range(s) has the lowest launch success rate?

0 Kg – 1000 Kg

- Which F9 Booster version (v1.0, v1.1, FT, B4, B5, etc.) has the highest launch success rate?

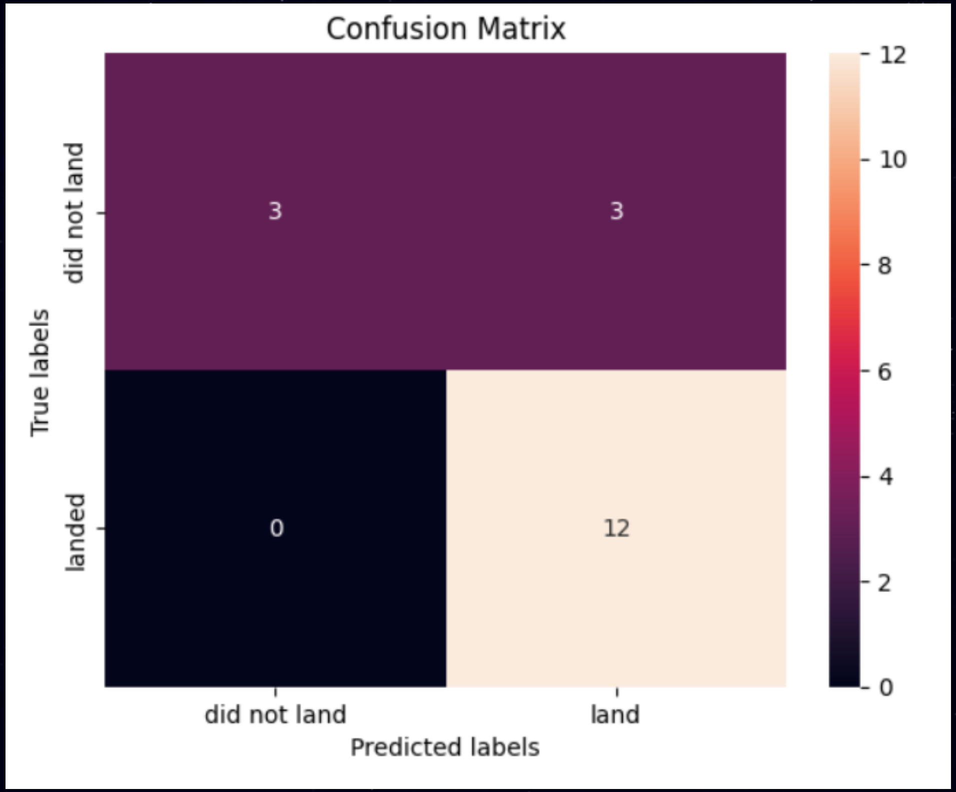
FT

Predictive Analysis

(Classification)

Confusion Matrix

Out here for all models unfortunately, we have same confusion matrix.



Accuracy: (TP+TN)/Total = (12+3)/18 = 0.83333

Misclassification Rate: (FP+FN)/Total = (3+0)/18 = 0.1667

True Positive Rate: TP/Actual Yes = 12/12 = 1

False Positive Rate: FP/Actual No = 3/6 = 0.5

True Negative Rate: TN/Actual No = 3/6 = 0.5

Precision: TP/Predicted Yes = 12/15 = 0.8

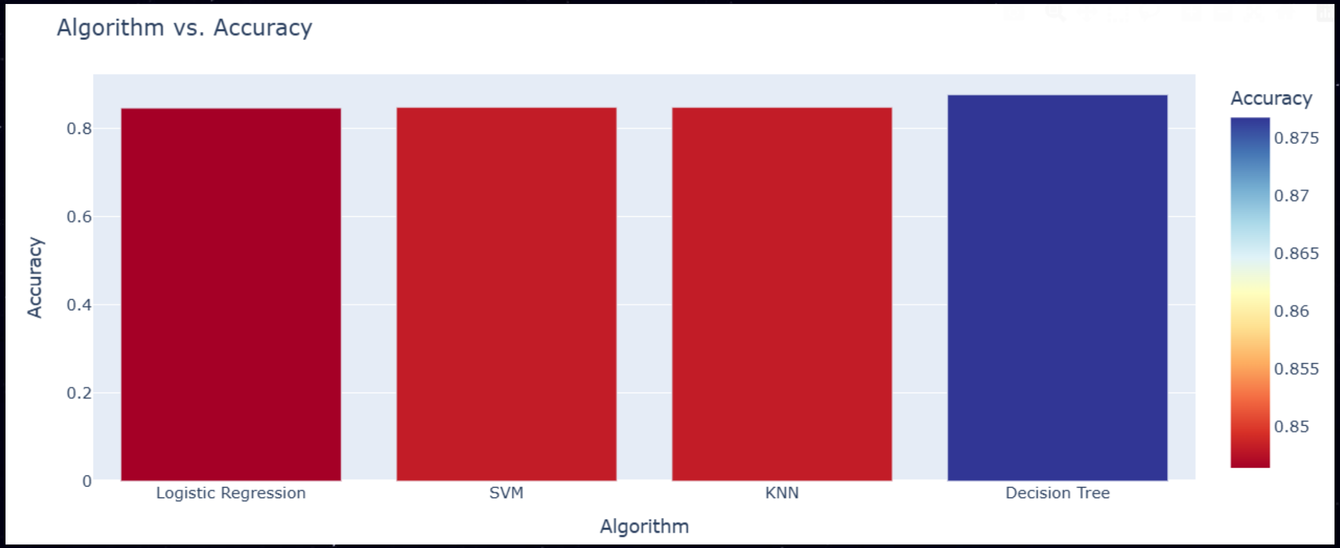
Prevalence: Actual yes/Total = 12/18 = 0.6667

Classification Accuracy

As you can see our accuracy is extremely close, but we do have a clear winner which performs best - "Decision Tree" with a score of 0.90178.

Algorithm	Accuracy	Accuracy on Test Data	Tuned HyperParameter
Logistic Regression	0.846429	0.833334	{'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
SVM	0.848214	0.833334	{'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
KNN	0.848214	0.833334	{'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}
Decision Tree	0.901786	0.833334	{'criterion': 'gini', 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'best'}

We trained four different models which each had an 83% accuracy rate.



Conclusion

1. Orbits ES-L1, GEO, HEO, SSO has highest Success rates

2. Success rates for SpaceX launches has been increasing relatively with time and it looks like soon they will reach the required target

3. KSC LC-39A had the most successful launches but increasing payload mass seems to have negative impact on success

4. Decision Tree Classifier Algorithm is the best for Machine Learning Model for provided dataset

THANKS