

# **IBM Data Science Capstone Project**

---

**Dilan Çelik**

**10th May 2022**



# Outline

---

Executive Summary

Introduction

Methodology

Results

Conclusions

Appendix

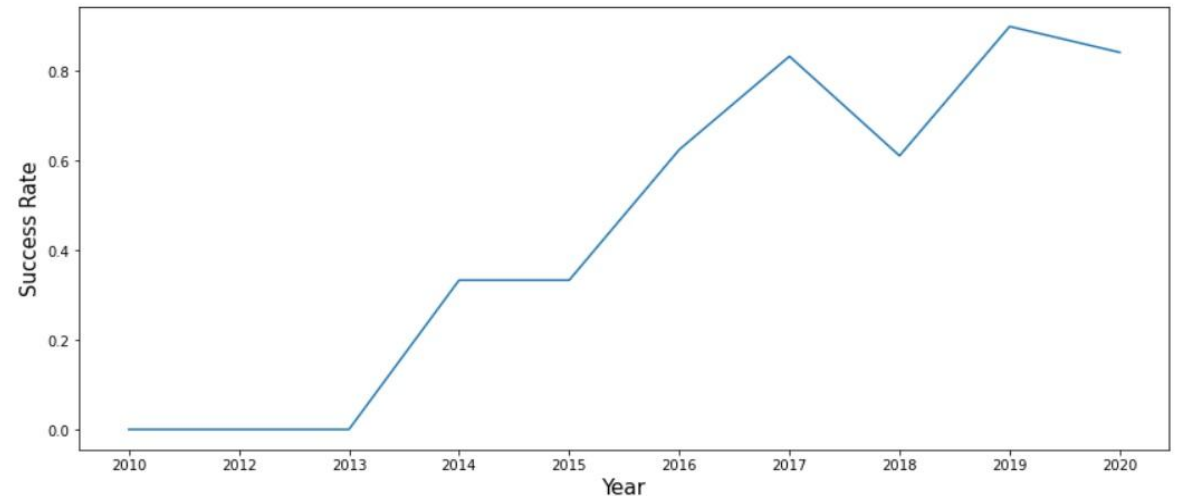
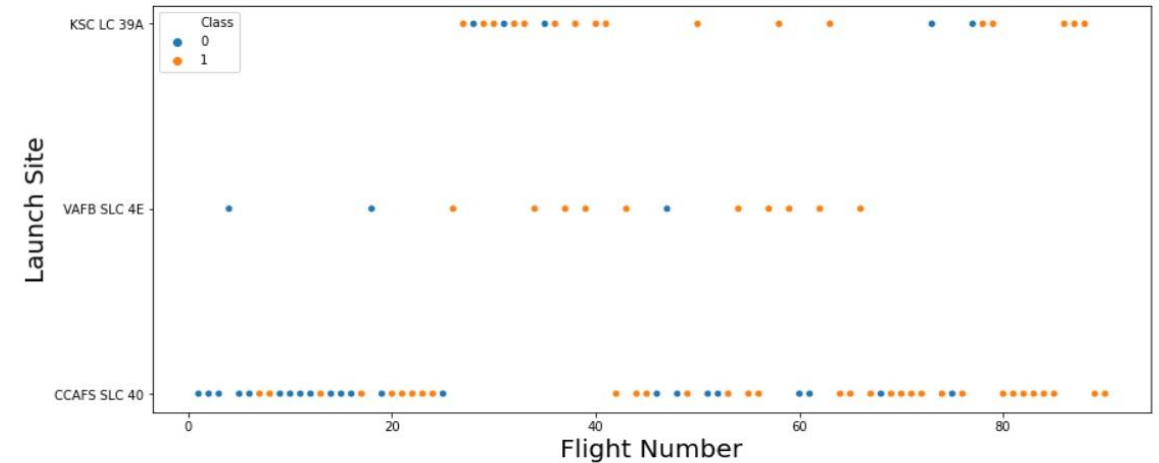
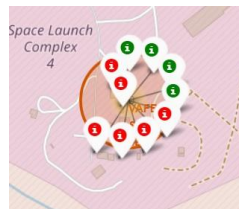
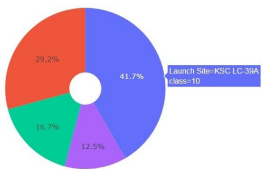
# Executive Summary

## Summary of the methodologies:

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

## Summary of all results:

- Data Analysis along with Interactive Visualizations
- Best Model for Predictive Analysis



# Introduction

---

## **Project background and context:**

In this project, 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, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

## **Problems we want to find answers:**

- By what factors will the rocket land successfully?
- The relationship between each variables and how it affects the results.
- The best conditions which will help to increase the successful landing.



Section 1

# Methodology

# Methodology

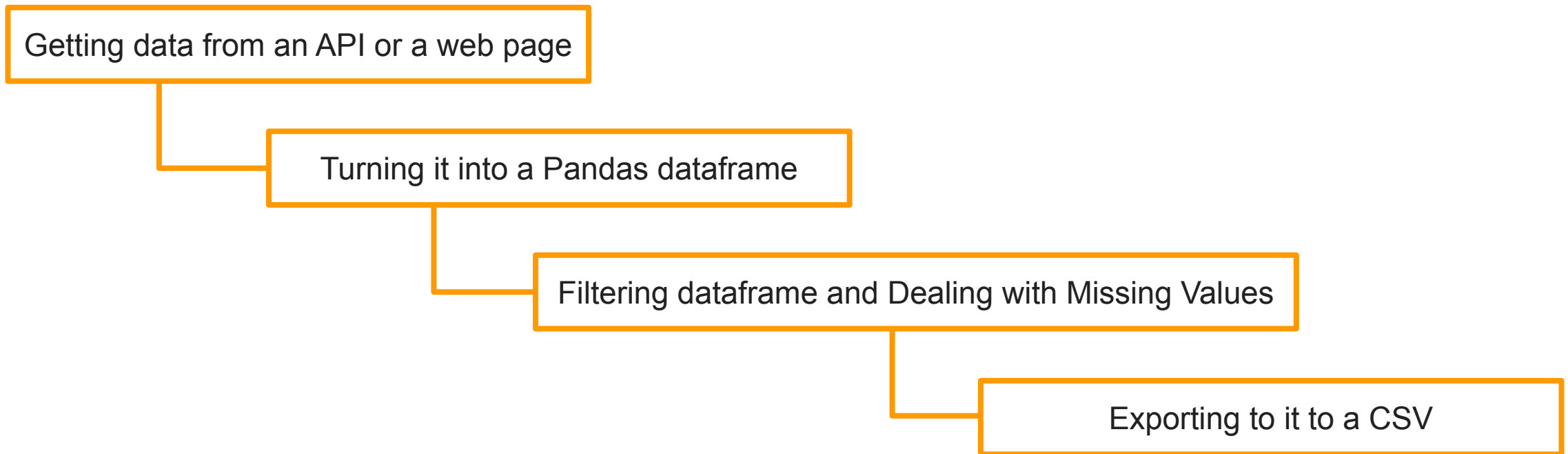
---

- Data collection methodology:
  - SpaceX API and web scraping from Wikipedia
- Perform data wrangling
  - One-hot encoding was used for categorical features
- 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

---

Data collection is the process of collecting, measuring and analyzing different types of information data from a variety of relevant sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities. As mentioned before, the SpaceX API and web scraping methods were used for data collection.



# Data Collection - SpaceX API

- Get response from API
- Convert the response to a dataframe
- Apply custom function
- Clean and filter data
- Deal with the missing values(with mean)
- Export it to a CSV file

[GitHub](#)

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False
...											

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

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

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

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

```
data_falcon9 = df.loc[df['BoosterVersion']!="Falcon 1"]  
data_falcon9.loc[:, 'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
```

```
mean = data_falcon9['PayloadMass'].mean()  
data_falcon9['PayloadMass'].fillna(mean, inplace=True)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```



# Data Collection - Scraping

Getting response from Web page

Creating a BeautifulSoup object

Finding tables and getting the column names

Creating a dictionary and appending data to keys

Converting dictionary to a dataframe

Export it to a CSV file

[GitHub](#)

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

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

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

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

```
first_launch_table = html_tables[2]
```

```
column_names = []

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

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

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

	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	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	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success	F9 v1.0B0007.1	No attempt	1 March 2013	15:10

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

# Data Wrangling

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

- 1- Load Data
- 2- Make a dataframe from it
- 3- Cleaning data
- 4- Calculate number and occurrence of mission outcome for per orbit type
- 5- Create landing outcome label from Outcome column
- 6- Export it to a CSV file

[GitHub](#)

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS032
```

```
landing_outcomes = df["Outcome"].value_counts()  
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

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

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

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

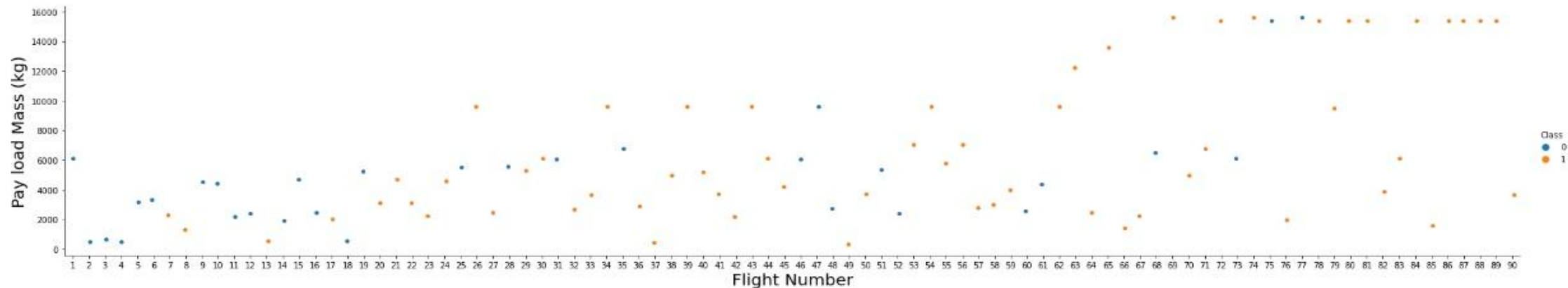
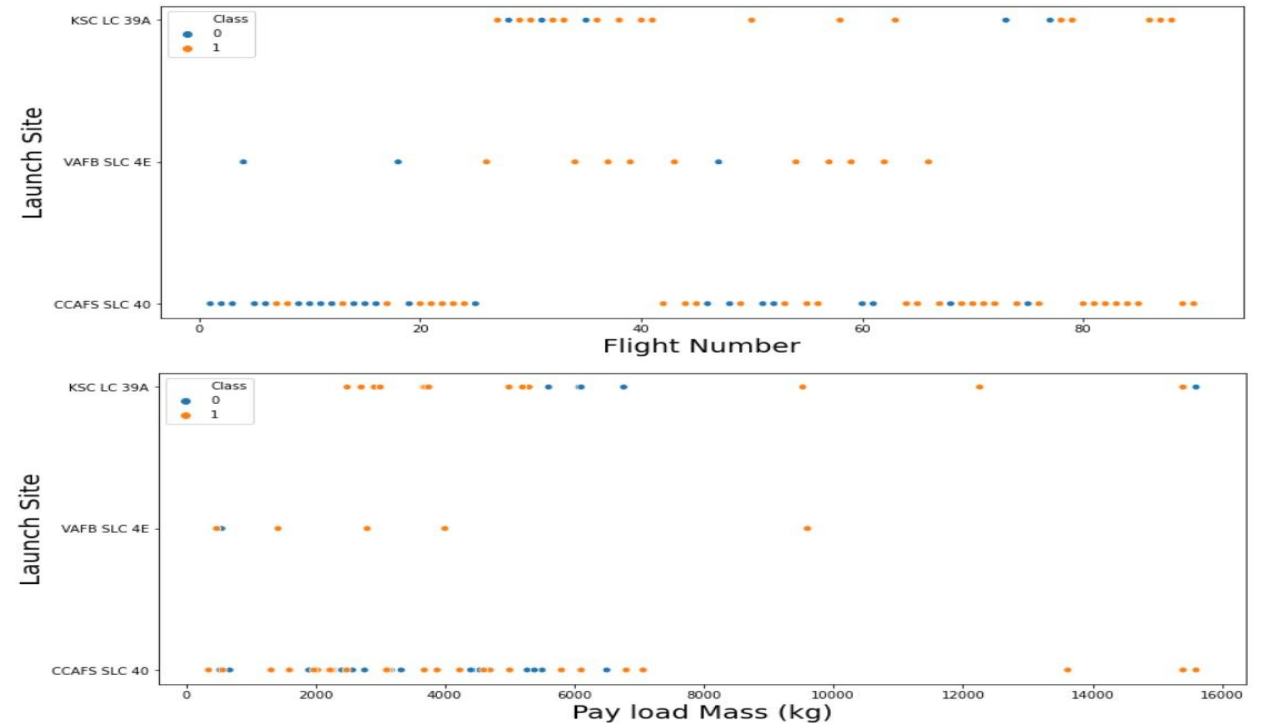
# EDA with Data Visualization

We first started with the scatter graph to find relationship between features;

- Payload and Flight Number
- Flight Number and Launch Site
- Payload and Launch Site
- Payload and Orbit Type
- Flight Number and Orbit Type

With the scatter plot, we can easily observe the dependency of features with each other. And it gives us a sight to predict of which factors affect the successful landing.

[GitHub](#)



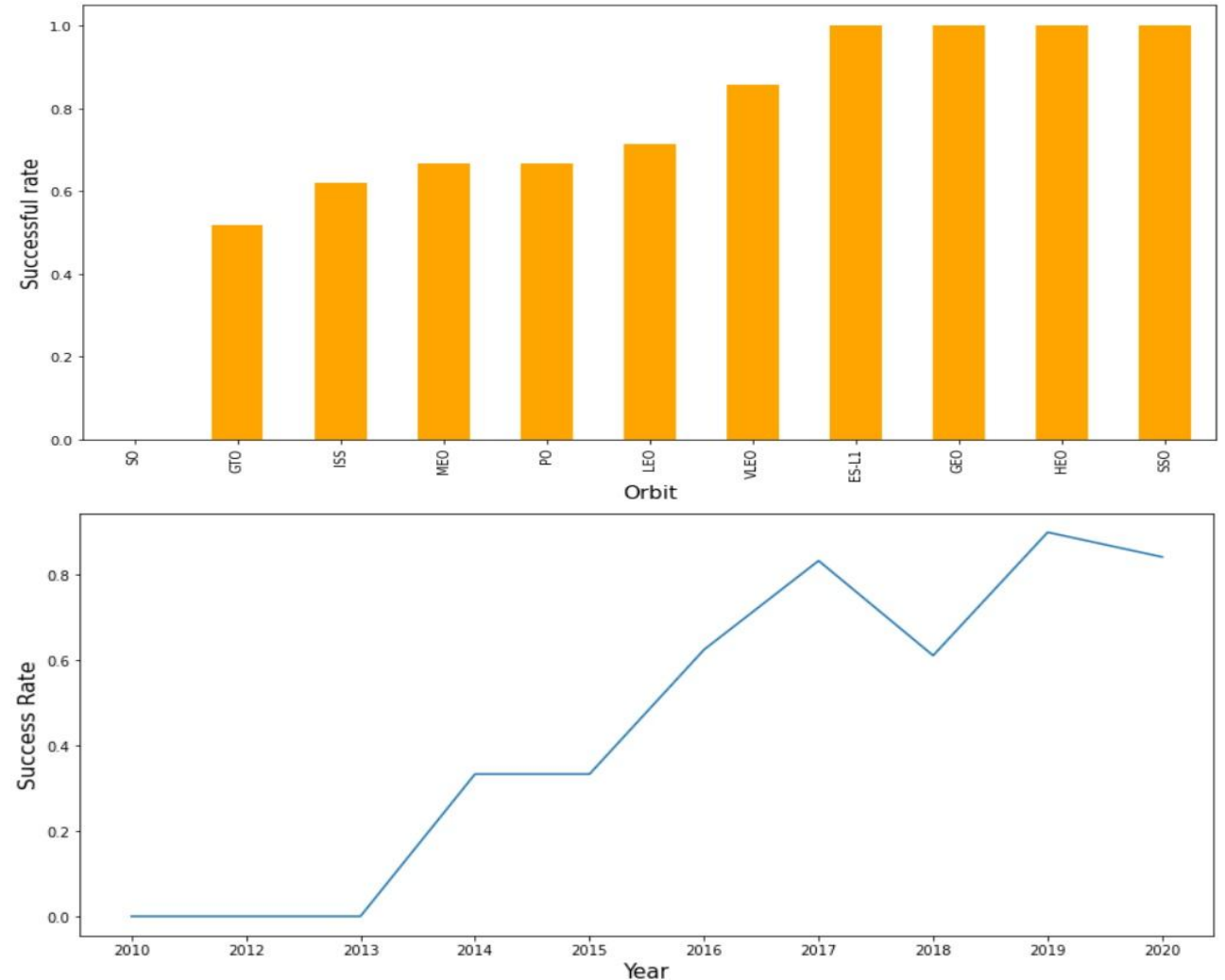
# EDA with Data Visualization

After the scatter plot, we used the bar graph and line plot for better understanding about data.

The bar graph is the easiest way to interpret the relationship between the features. We used the bar graph to observe the successful rates for each orbit. It helps us to determine which orbits have the highest probability for successful landing.

We used the line graph to see how the trends changes over time. It will help us to make predictions for the future.

[GitHub](#)



# EDA with SQL

---

**We performed the SQL queries to gather information from given dataset:**

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster\_versions which have carried the maximum payload mass. Use a subquery
- List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order



# Built an Interactive Map with Folium

Folium makes it easy to visualize data on an interactive leaflet map. We used the latitude and the longitude coordinates for each launch site with added the circle markers around the each launch site with the label of the name of the launch site.

We used the Red marker for failure landings (class 0) and Green marker for successful landings (class 1) on the map in Marker Cluster.

Map Marker	<code>folium.map.Marker()</code>	Map object to make a mark on map
Circle Marker	<code>folium.Circle()</code>	Create a circle where marker is being placed
Icon Marker	<code>folium.Icon()</code>	Create an icon on map
Marker Cluster	<code>MarkerCluster()</code>	It simplify a map containing many markers having the same coordinate.
PolyLine	<code>folium.PolyLine()</code>	Create a line between points

With those objects we can analyze the how close the launch sites with coastlines, highways, or cities.

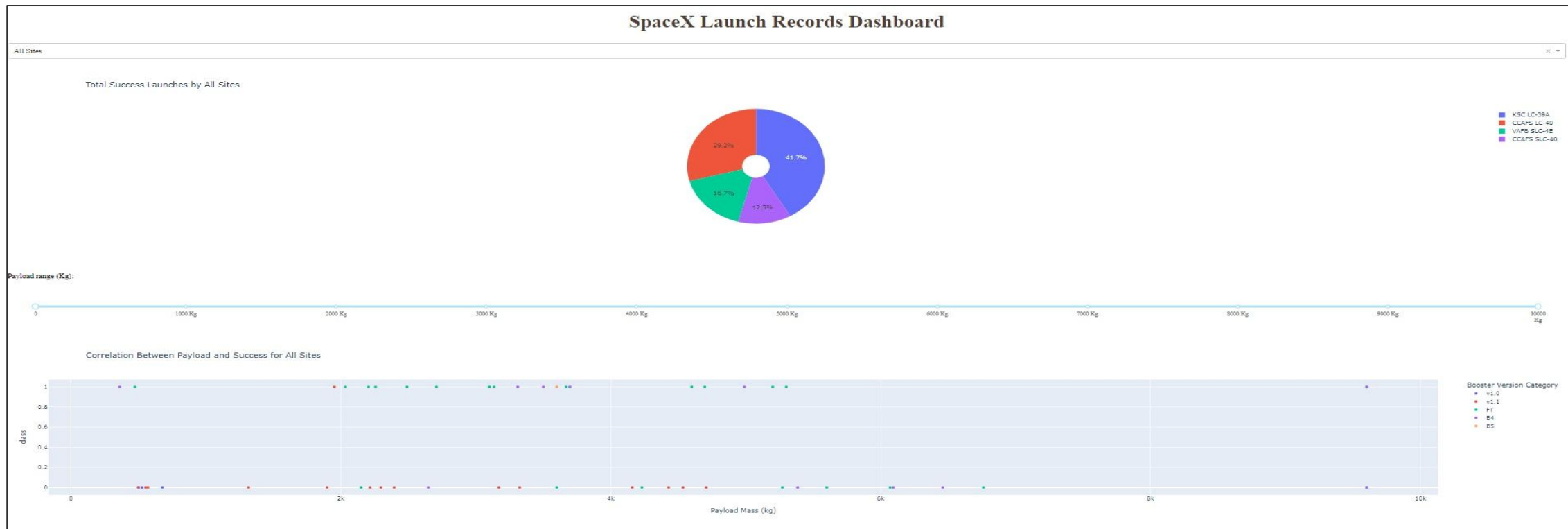
# Built a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly Dash, and added a pie chart and a scatter plot which providing the users to interact with the data as they wish.

The pie chart shows total success for all sites or by a certain launch site.

The scatter plot shows the relationship between the Outcome and Payload Mass(kg) for the different booster version.

[GitHub](#)



# Predictive Analysis (Classification)

---

## Building Model

- Load the data.
- Transform it as features (X) and target (y) data with Pandas and NumPy modules.
- Standardize data with StandardScaler() method.
- Split data into training sets and test sets
- Create the object of Machine Learning Algorithm which will be used
- Set the parameters and use the GridSearchCV method to find the best parameters
- Fit the training set into the model

## Evaluating Model

- Calculate the accuracy on the test data for each model
- Get the best parameters for each model
- Plot the Confusion Matrix

## Finding the Best Model

- The model which has the best accuracy score will be the Best Model to make predictions.

[GitHub](#)

# Results

---

Exploratory data analysis results

Interactive analytics demo in screenshots

Predictive analysis results



The background of the slide is an abstract composition. It features a solid blue area on the left side, which transitions into a dynamic pattern of diagonal streaks in shades of blue, red, and cyan on the right. These streaks have a textured, almost woven appearance, suggesting a digital or data-driven theme. A faint, light blue grid pattern is also visible, particularly in the lower-left quadrant, adding to the technical aesthetic.

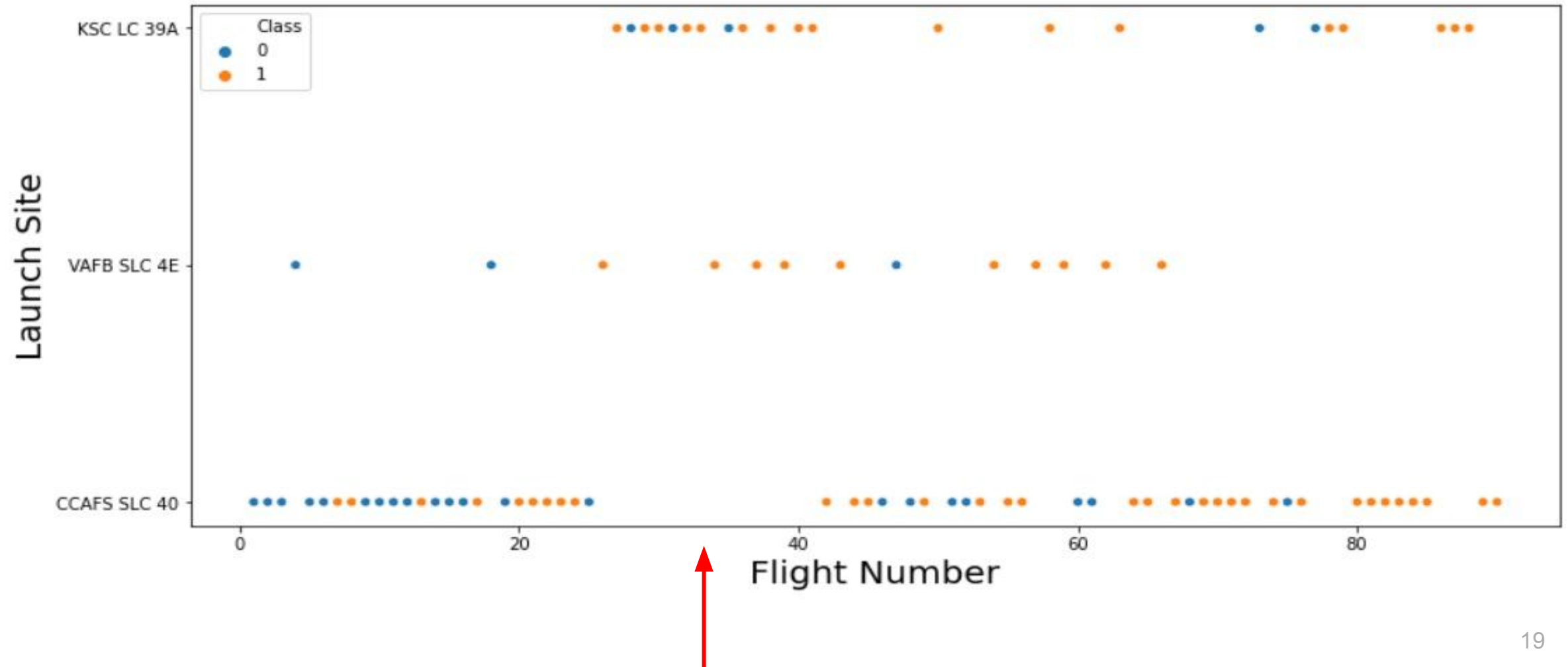
Section 2

# Insights drawn from EDA



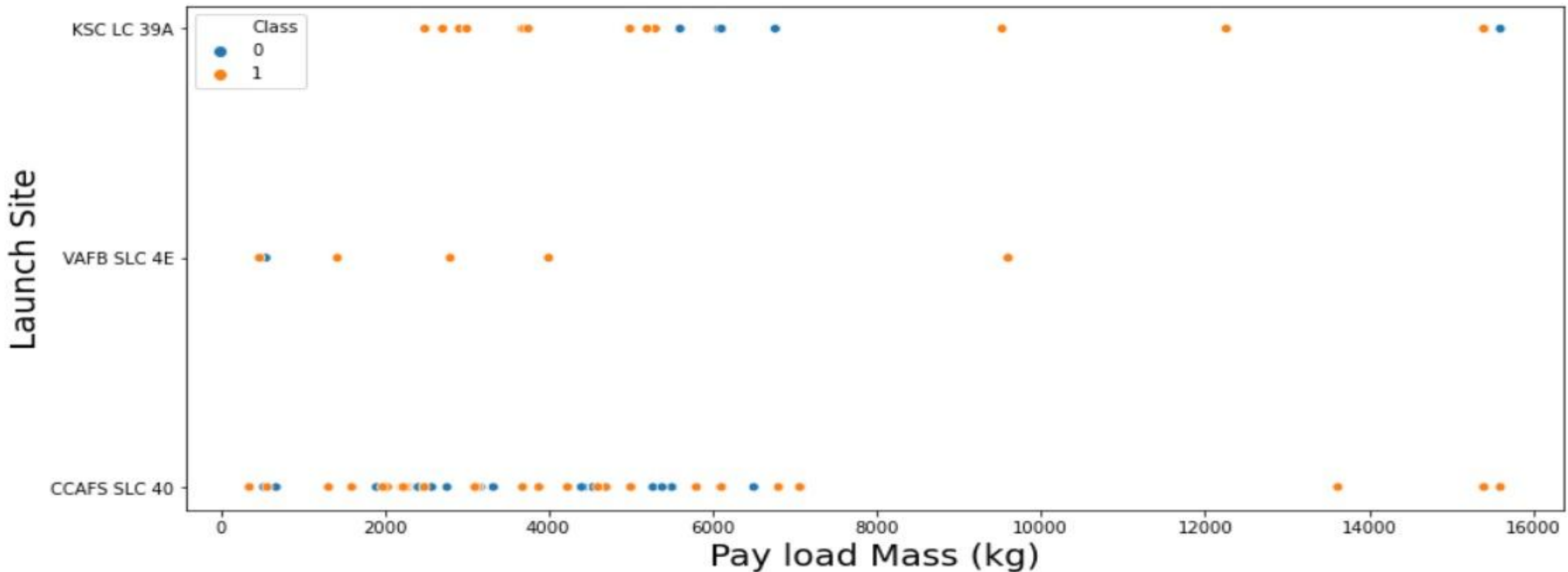
# Flight Number vs. Launch Site

This scatter plot shows that with higher flight numbers the success rate is increasing.



# Payload vs. Launch Site

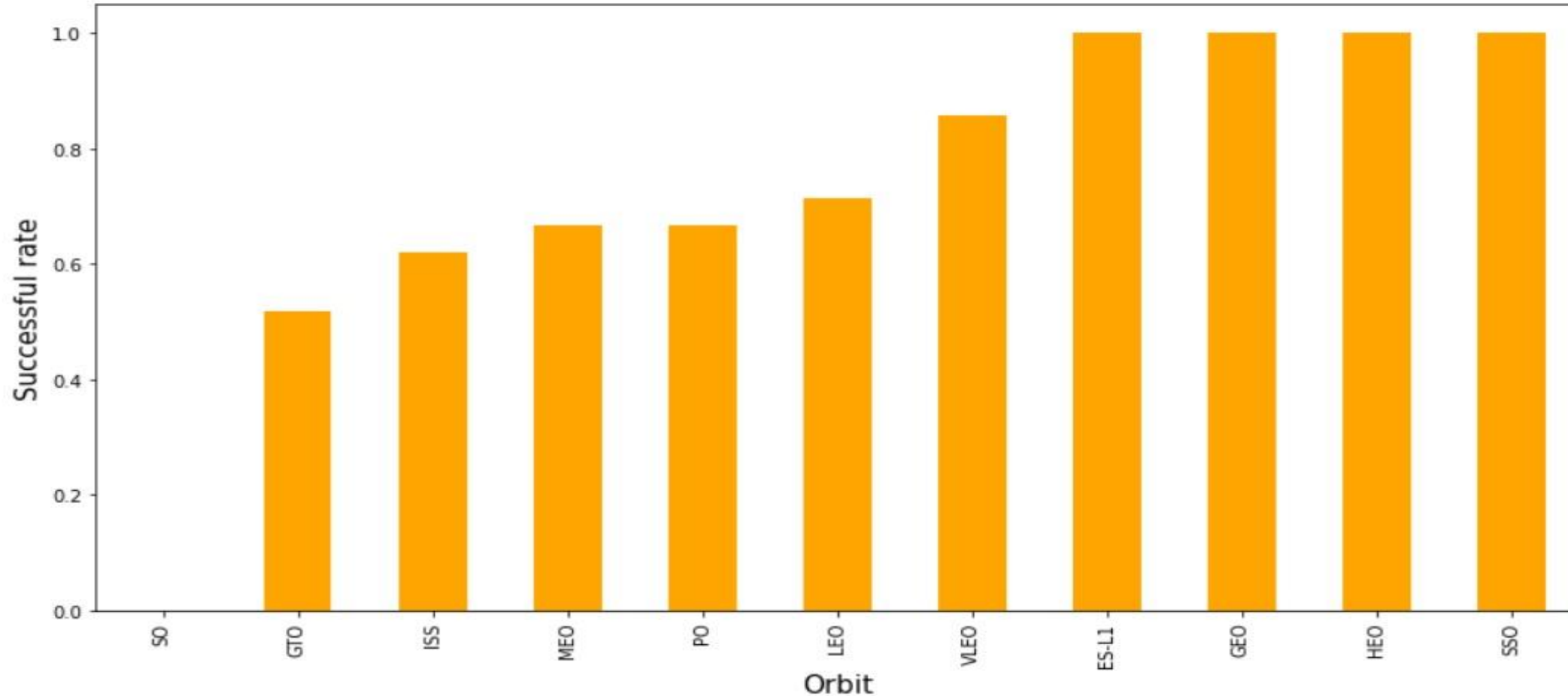
This scatter plot shows that there is no clear pattern between Launch site and Payload Mass for successful landing but we can observe that the VAFB-SLC launchsite there are no rockets launched for heavy payload mass(greater than 10000).



# Success Rate vs. Orbit Type

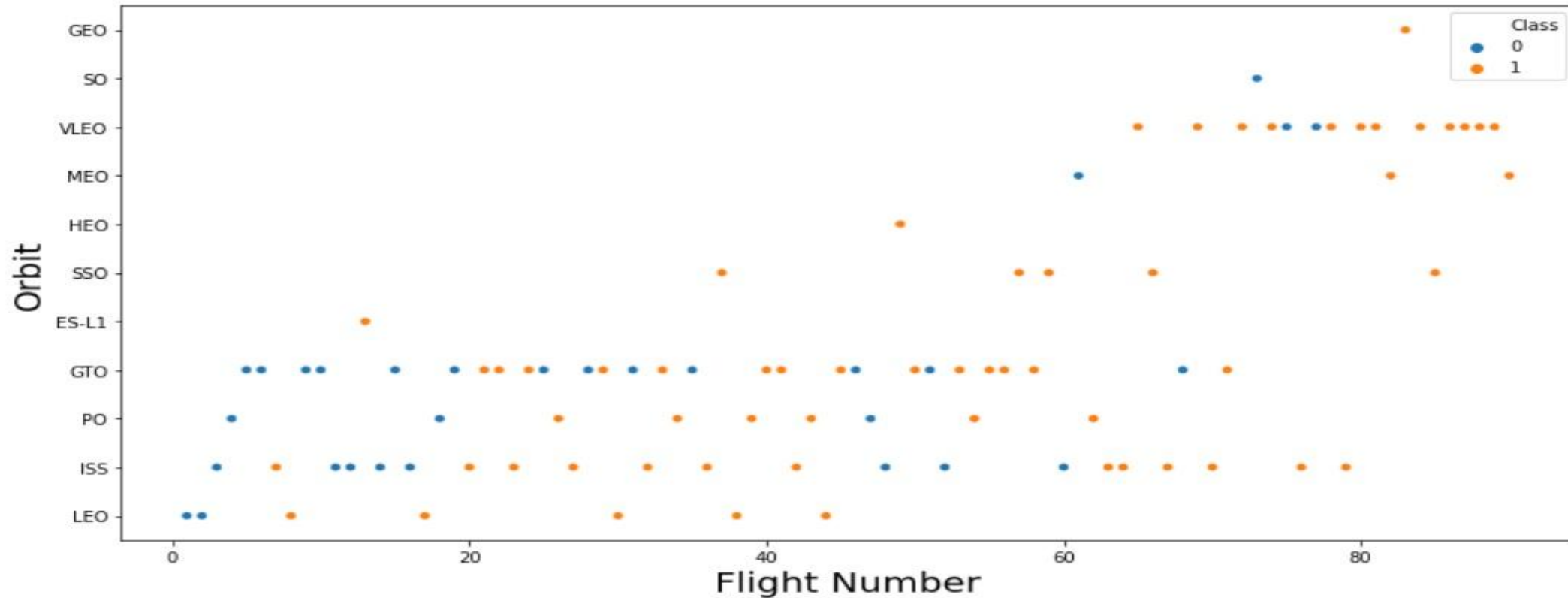
---

With this bar plot, we can see that ES-L1, GEO, HEO and SSO have highest success rate and SO has no successful landing.



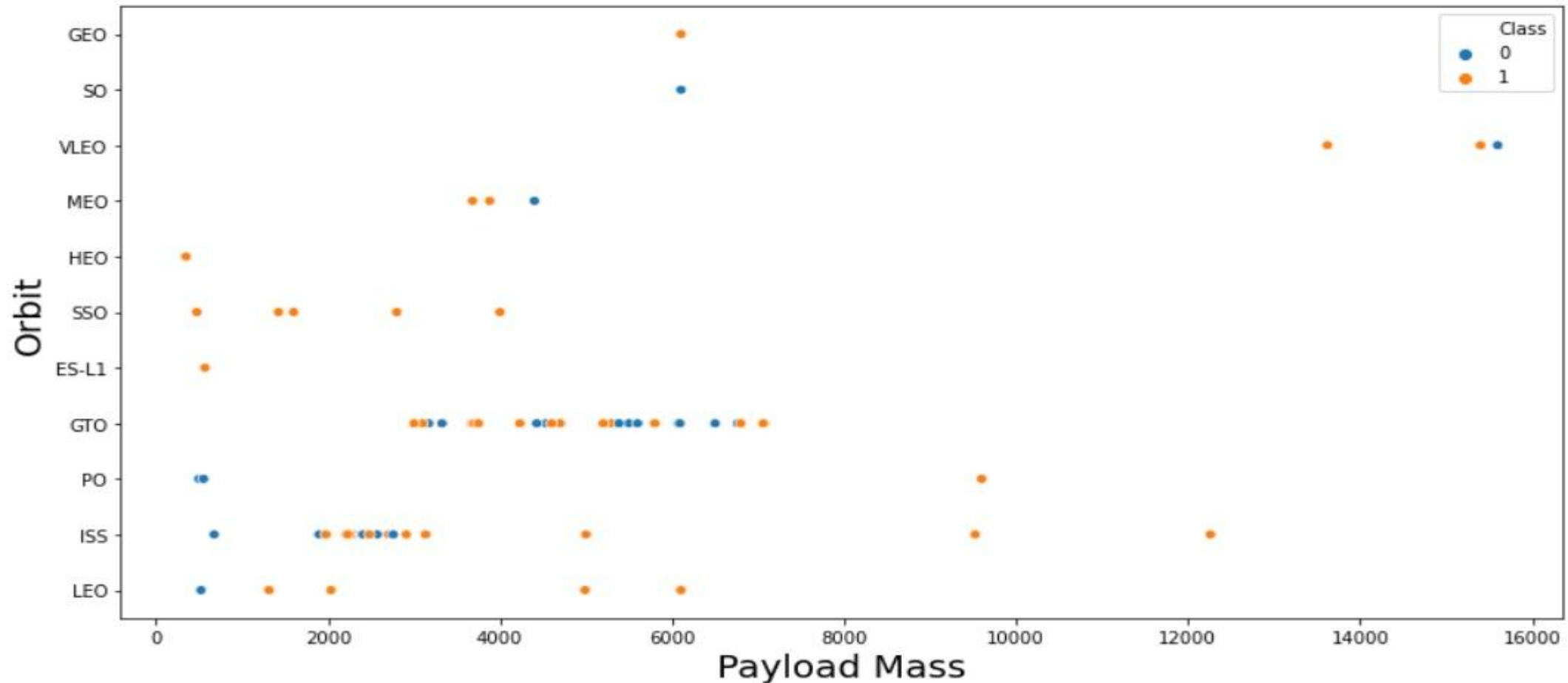
# Flight Number vs. Orbit Type

This scatter plot shows that the larger flight number on each orbits, greater the success rate, such as LEO, except for GTO which indicates no relationship between these features.



# Payload vs. Orbit Type

This scatter plot shows that the heavier payloads have a positive influence on ISS, LEO and PO. On the other hand, it has negative influence on the MEO and VLEO. We can not see any relationship between the GTO and the payload mass.

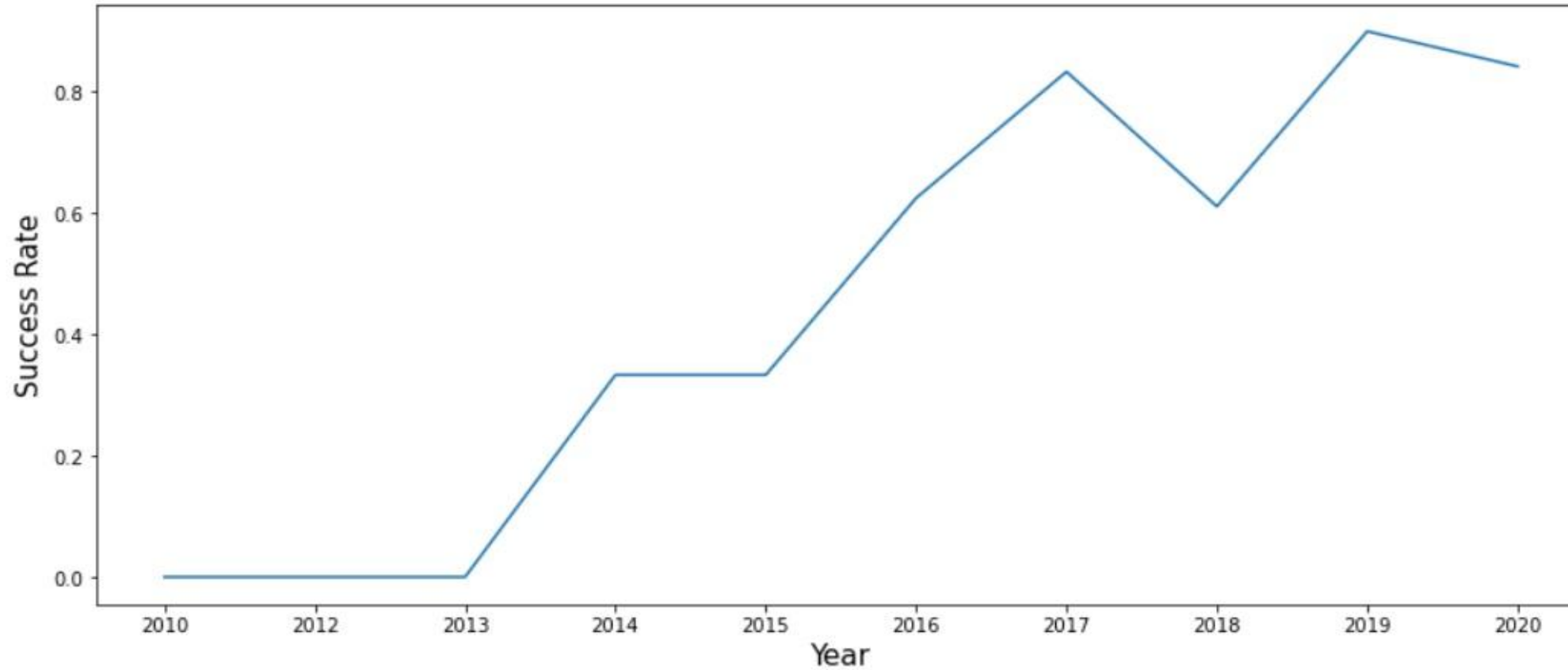




# Launch Success Yearly Trend

---

This chart shows that the successful rate is increasing relatively from the year 2013 until the 2020.



# All Launch Site Names

---

## SQL Query:

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

## Description:

We used the DISTINCT function to find the all unique launch site names from Spacextbl with a launch\_site column name.

### **launch\_sites**

CCAFS LC-40

CCAFS SLC-40

KSC LC-39A

VAFB SLC-4E

# Launch Site Names Begin with 'CCA'

## SQL Query:

```
%%sql  
  
SELECT * FROM SPACEXTBL  
WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

## Description:

We used the LIKE operator to find the records which begin with 'CCA' and LIMIT 5 operator to get 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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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 by NASA(CRS)" FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)';
```

## Description:

We used the SUM function to calculate the total payload carried by boosters from the 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 SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

## Description:

We used the AVG function to calculate the average payload carried by booster version F9 v1.1

Average Payload Mass by Booster Version F9 v1.1

2928



# First Successful Ground Landing Date

---

## SQL Query:

```
%%sql
SELECT MIN(DATE) AS "The first successful landing date" FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

## Description:

We used the MIN function to calculate the first successful landing outcome on ground pad

**The first successful landing date**

2015-12-22

# Successful Drone Ship Landing with Payload Between 4000 and 6000

## SQL Query:

```
%%sql
SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

## Description:

We used the multiple where selection to find the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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 SPACEXTBL;
```

## Description:

We used the CASE clause within the SUM functions to calculate the total number of successful and failure mission outcomes.

Successful Mission	Failure Mission
100	1

# Boosters Carried Maximum Payload

## SQL Query:

```
%%sql
SELECT BOOSTER_VERSION AS "Booster Versions which carried the maximum payload mass" FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

## Description:

We used the sub-query with MAX function to find the names of the booster which have carried the maximum payload mass.

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 BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE year(DATE) = '2015' AND LANDING__OUTCOME = 'Failure (drone ship)';
```

## Description:

We used the year function to list the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
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 SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING__OUTCOME
ORDER BY COUNT(LANDING__OUTCOME) DESC;
```

## Description:

We used the COUNT function to calculate number of the failed landing\_outcomes between the date 2020-06-04 and 2017-03-20 group by landing\_outcome and ranked them in descending order.

Landing Outcome	Total Count
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

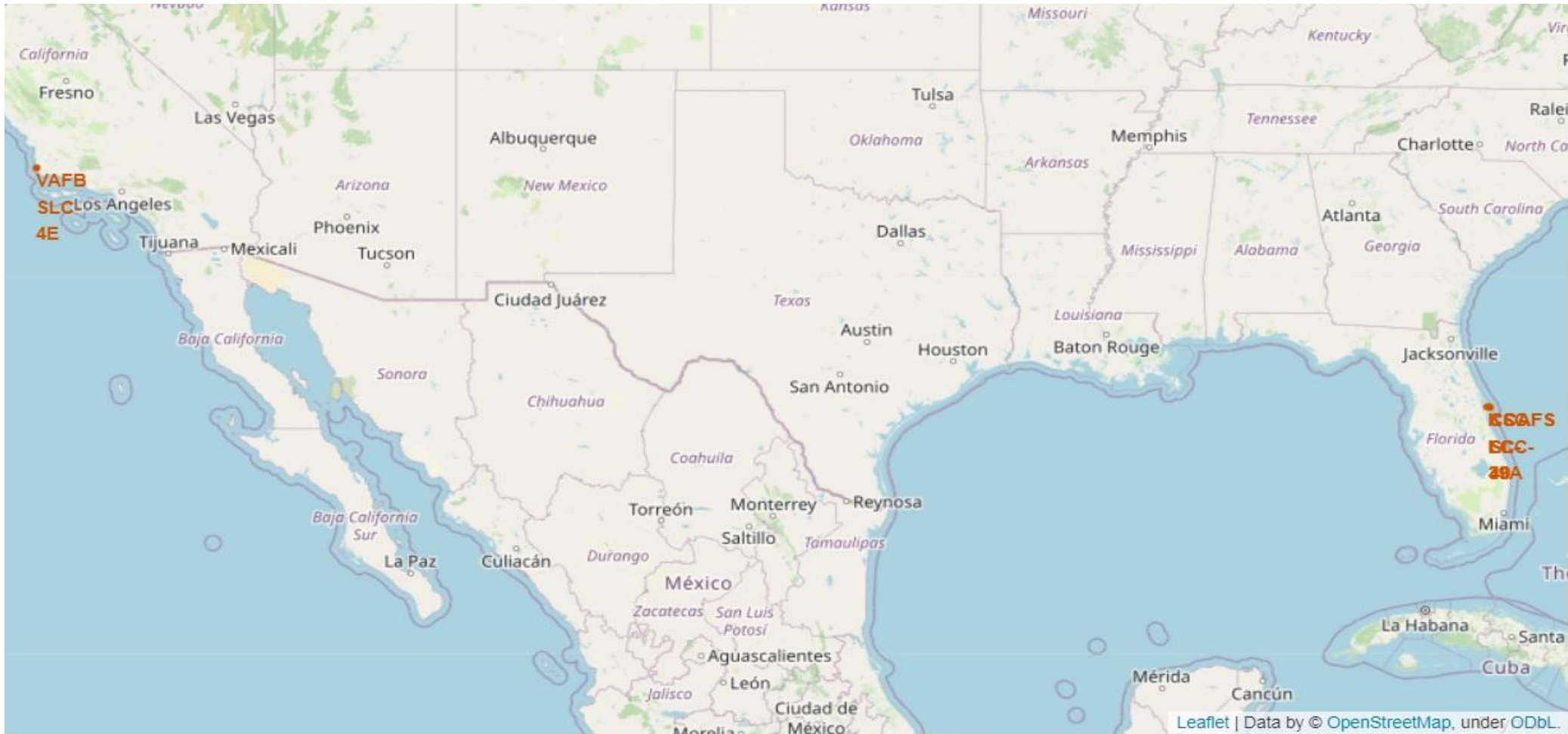
# Launch Sites Proximities Analysis





# All Launch Sites on Folium Map

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



# All Launch Sites on Folium Map

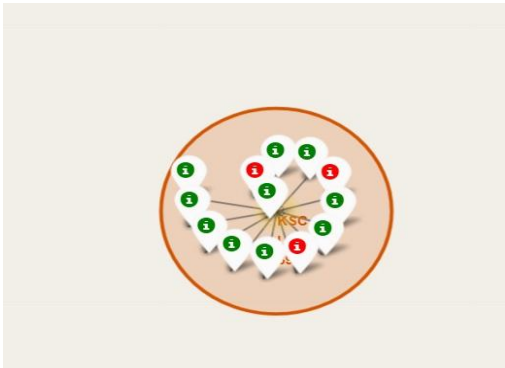
Green markers show the successful launches and Red markers show the failures. From these maps we can see that the KSC LC-39-A maximum probability of success



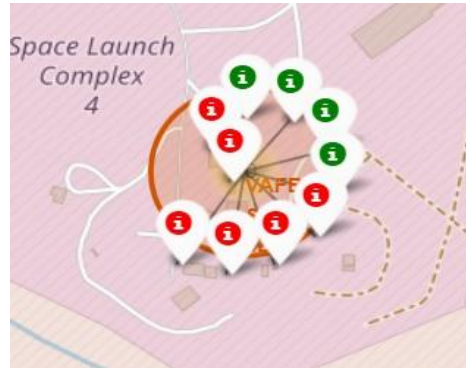
CCAFS LC-40



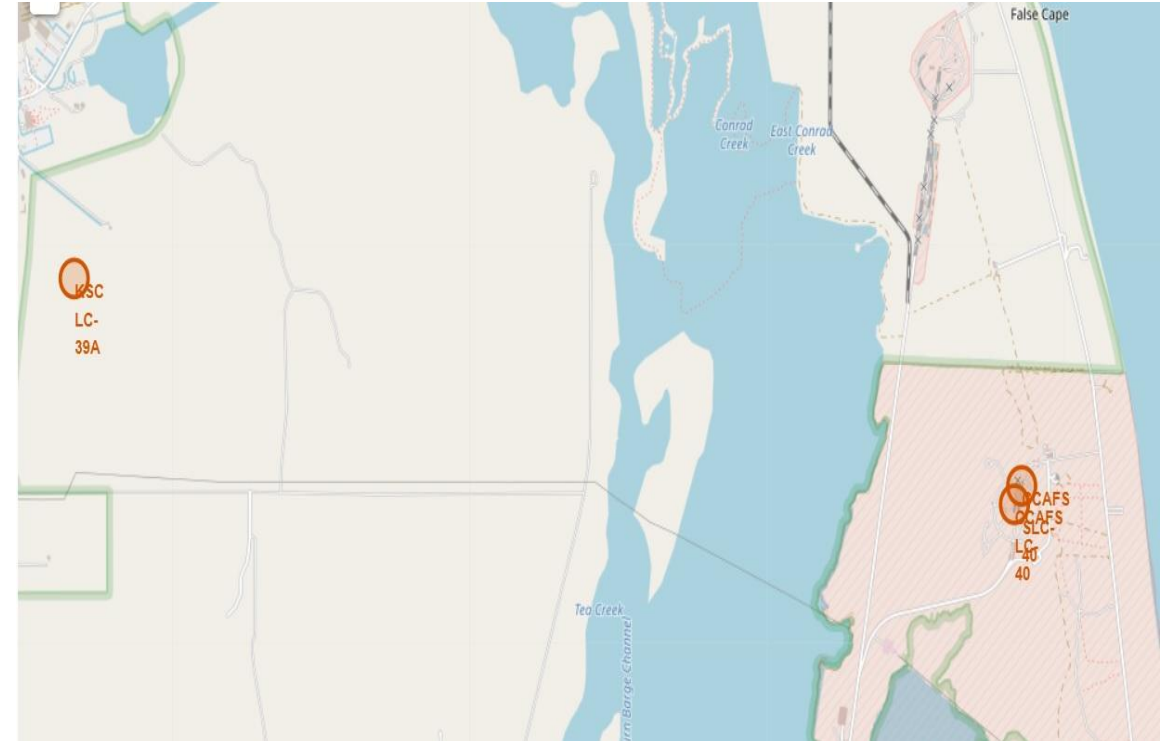
CCAFS SLC-40



KSC LC-39A

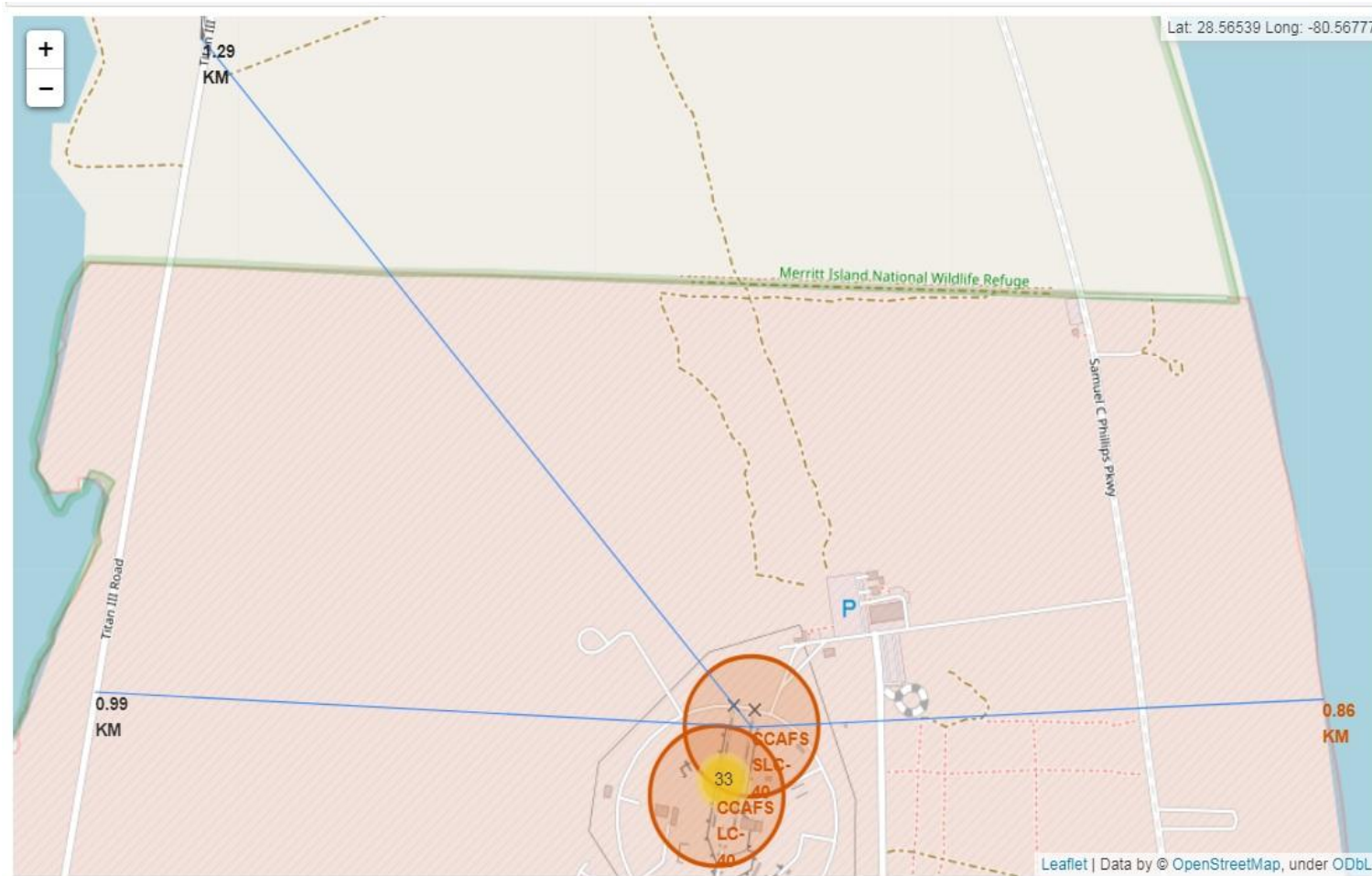


VAFB SLC-4E



# The distances between launch sites to landmarks

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



With these informations we can answer these questions:

- Are launch sites in close proximity to railways?

**Yes**

- Are launch sites in close proximity to highways?

**No**

- Are launch sites in close proximity to coastline?

**Yes**

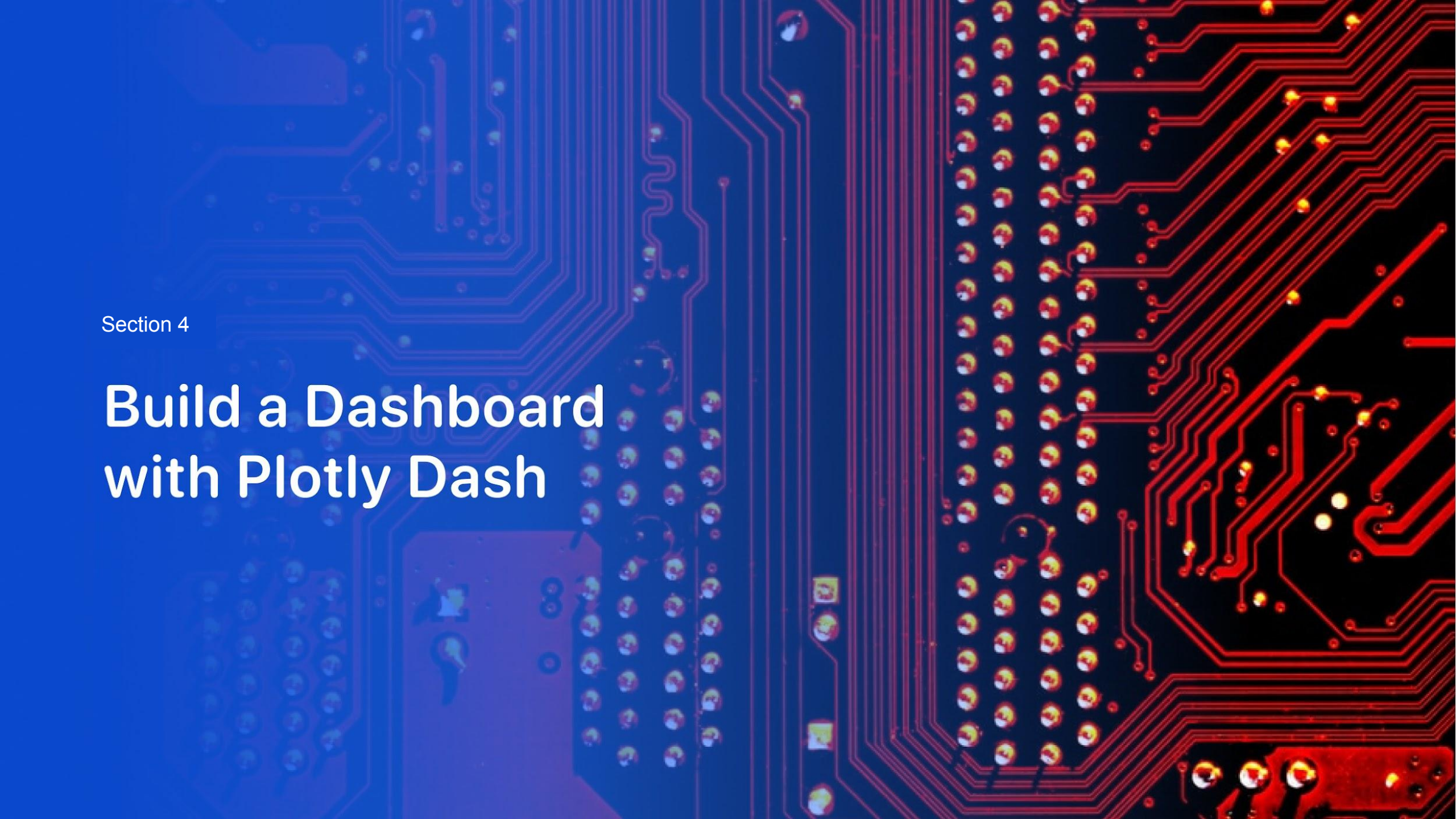
- Do launch sites keep certain distance away from cities?

**Yes**



Section 4

# Build a Dashboard with Plotly Dash



# Launch Success Percentages by All Sites

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

## SpaceX Launch Records Dashboard

All Sites



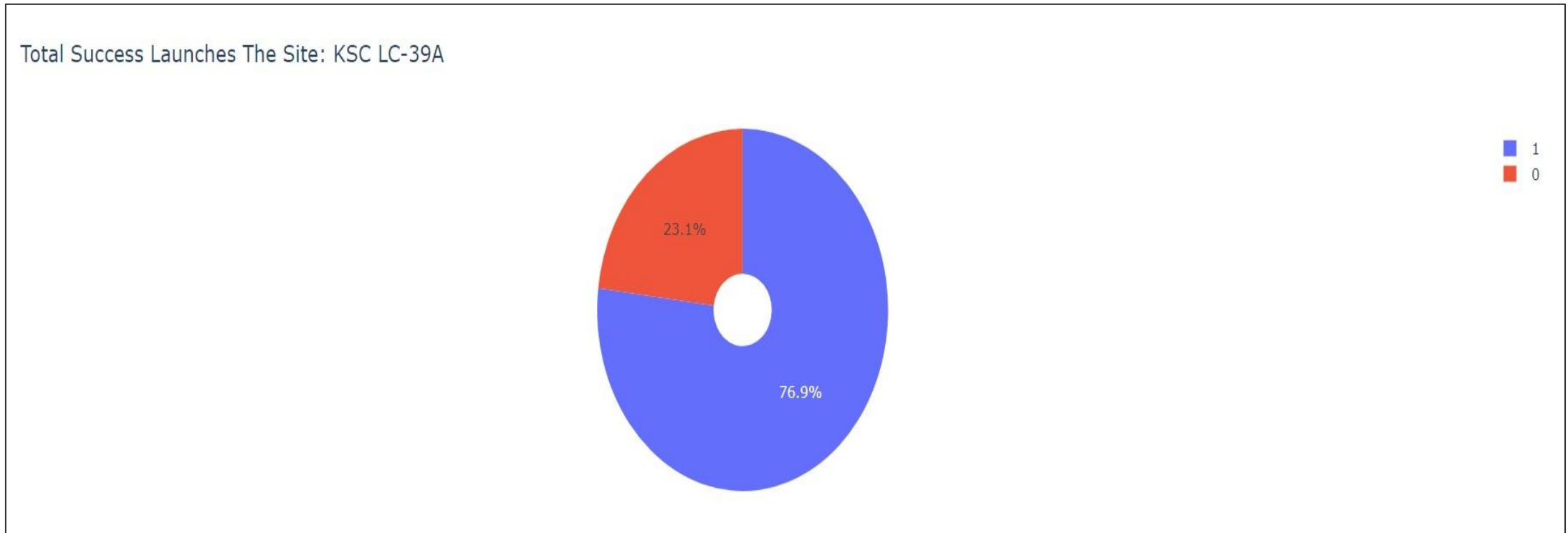
Total Success Launches by All Sites



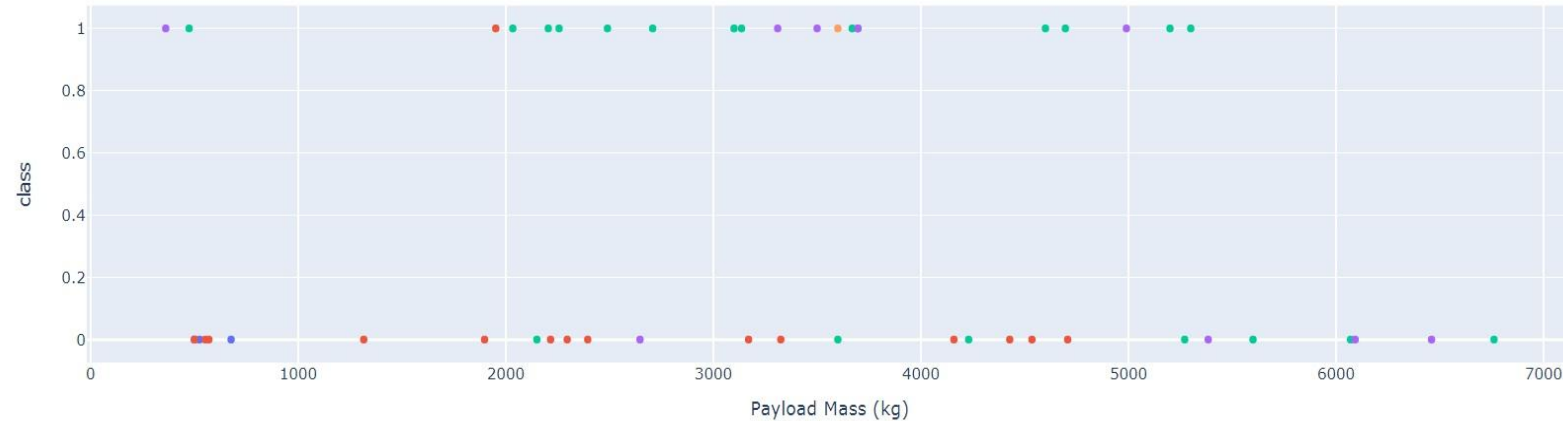
# The Highest Launch Success: KSC LC-39A

---

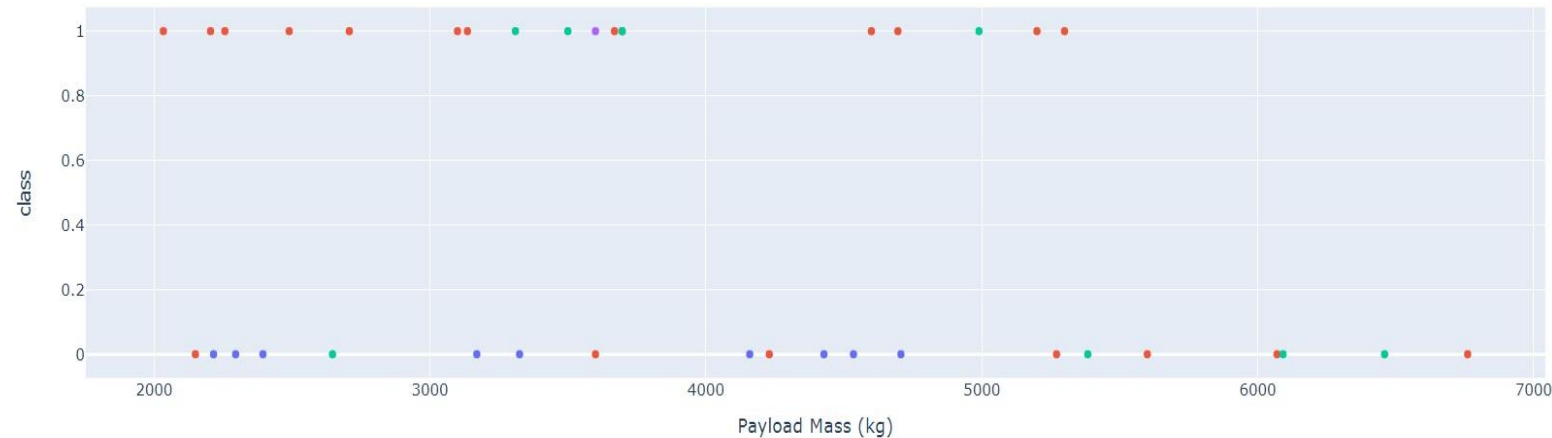
KSC LC-39A has a 76.9% success rate but getting a 23.1% failure rate.



# The Payload vs. Launch Outcome Scatter Plot



**Payload 0 kg - 7000 kg**



**Payload 2000 kg - 8000 kg**

After using the dashboard, we can gain some insights about our data. With this dashboard we can observe these:

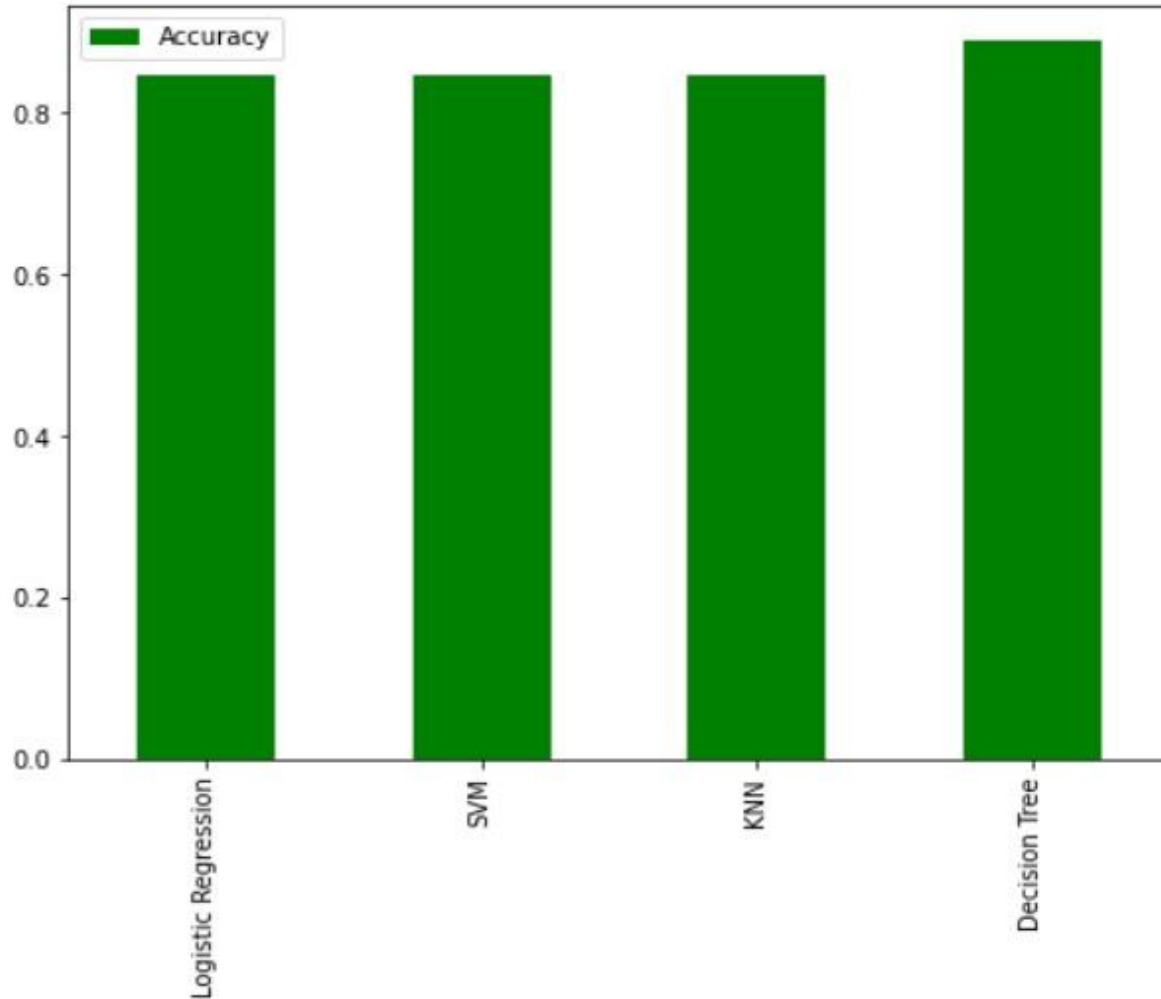
- The KSC LC-39A has the highest launch success rate
- The payload ranges between 2000kg - 10000kg has the highest launch success rate
- The FT booster version has the highest launch success rate



Section 5

# Predictive Analysis (Classification)

# Classification Accuracy



As you can see, the accuracy rates are really close to each other, but the **“Decision Tree”** model has the highest accuracy.

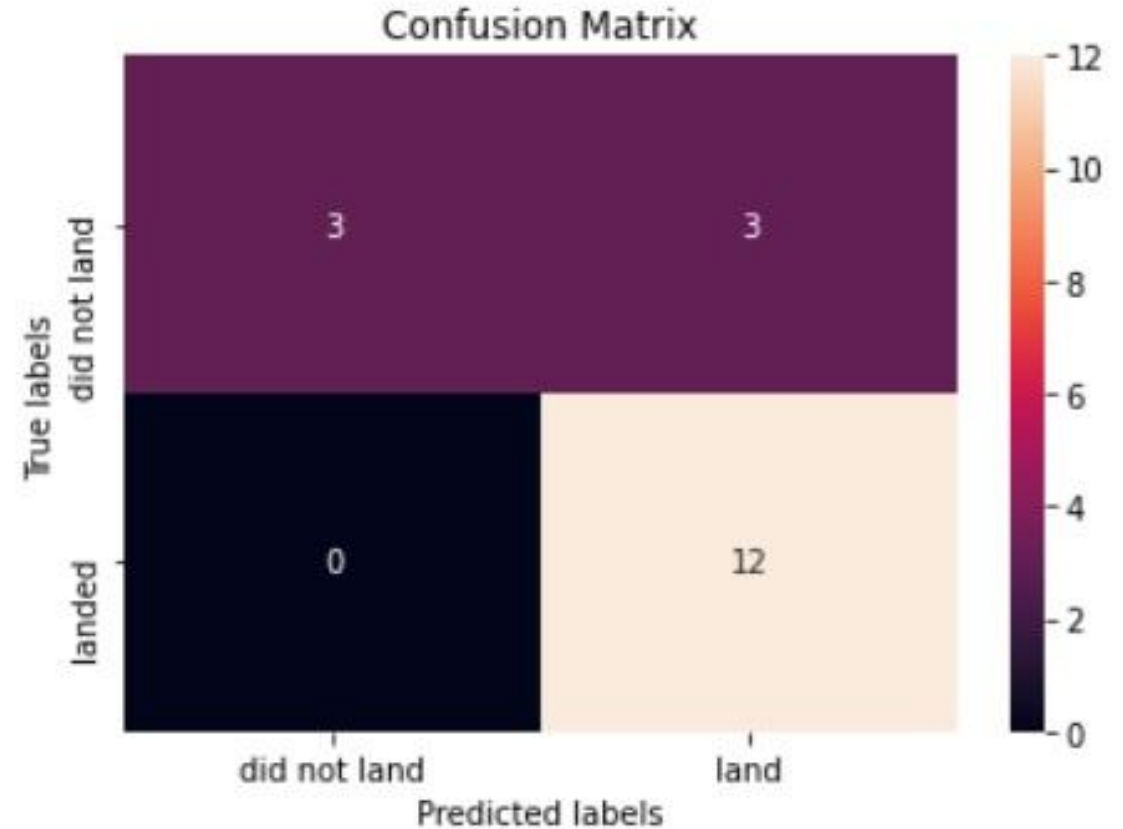
Accuracy	
Logistic Regression	0.846429
SVM	0.848214
Decision Tree	0.876786
KNN	0.848214

# Confusion Matrix

A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

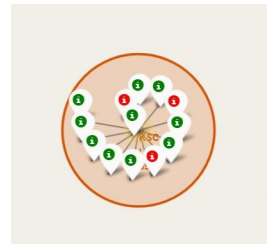
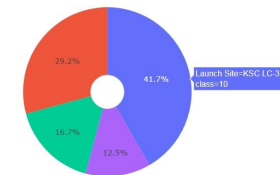
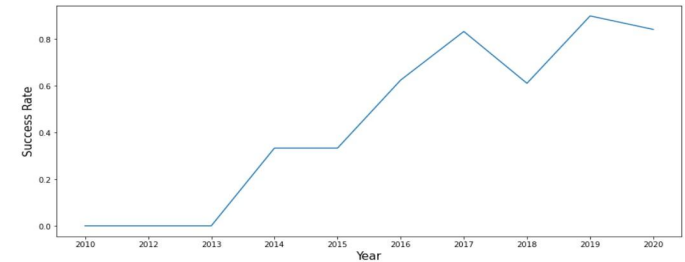
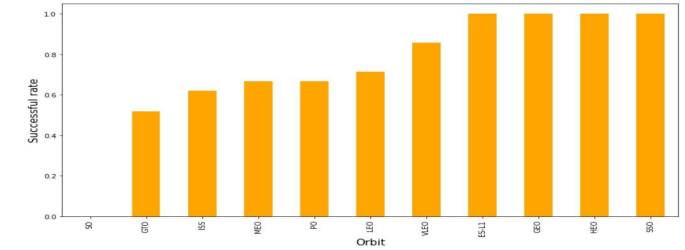
Unfortunately for all models, we found the same confusion matrix.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN



# Conclusions

- The orbits ES-L1, GEO, HEO, SSO have the highest success rates.
- The success rates have been increasing by the time.
- KSC LC-39A has the most successful launches but increasing payload mass is affecting it negatively.
- The Decision Tree Classification is the best algorithm for this dataset.



	Accuracy
Logistic Regression	0.846429
SVM	0.848214
Decision Tree	0.876786
KNN	0.848214





Thank you..