IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Akhib Umear Shaik
14-02-2023

Outline

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

Executive Summary

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

Introduction

- **Project background and context**

  This Project is about to predict if the Falcon 9 first stage rocket will land successfully. SpaceX advertises Falcon 9 rocket launches on it's Website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the saving 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 you 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.

Section 1

Methodology

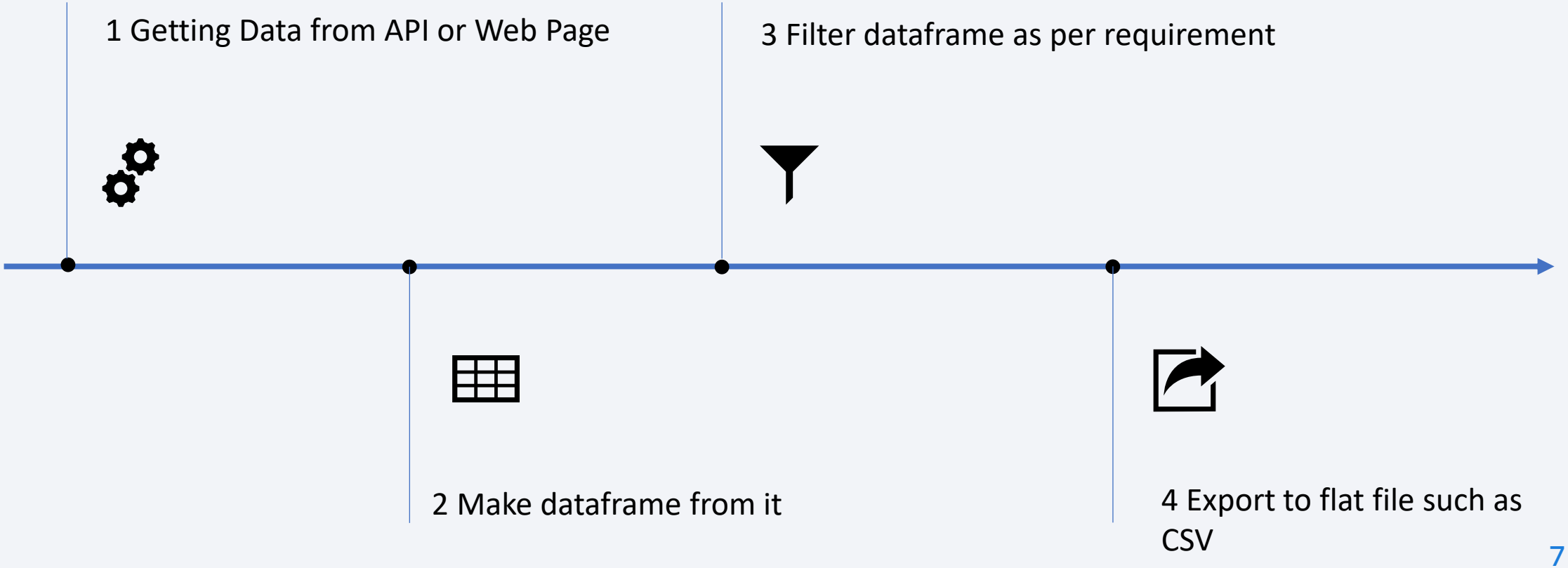
Methodology

Executive Summary

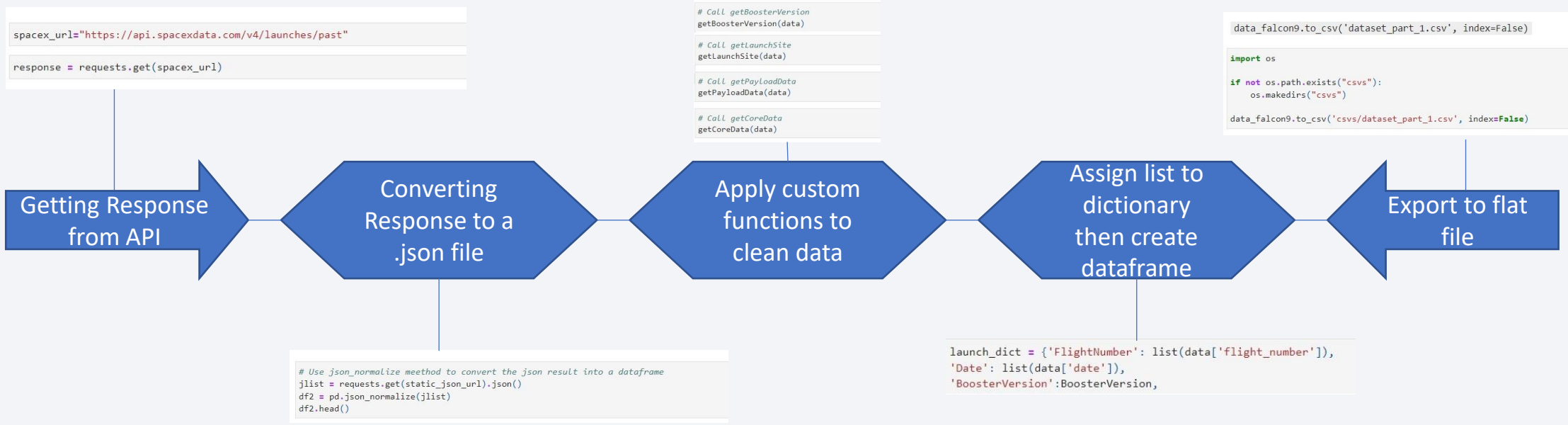
- Data collection methodology:
  - Via SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - One hot encoding data fields for machine learning and dropping irrelevant columns (Transforming data for Machine Learning)
- Perform exploratory data analysis (EDA) using visualization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

Data Collection

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 – SpaceX API




```
data_falcon9.head()
```

	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

GitHub Link

Data Collection - 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
```

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data)
```

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

```
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
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)
```

GitHub URL

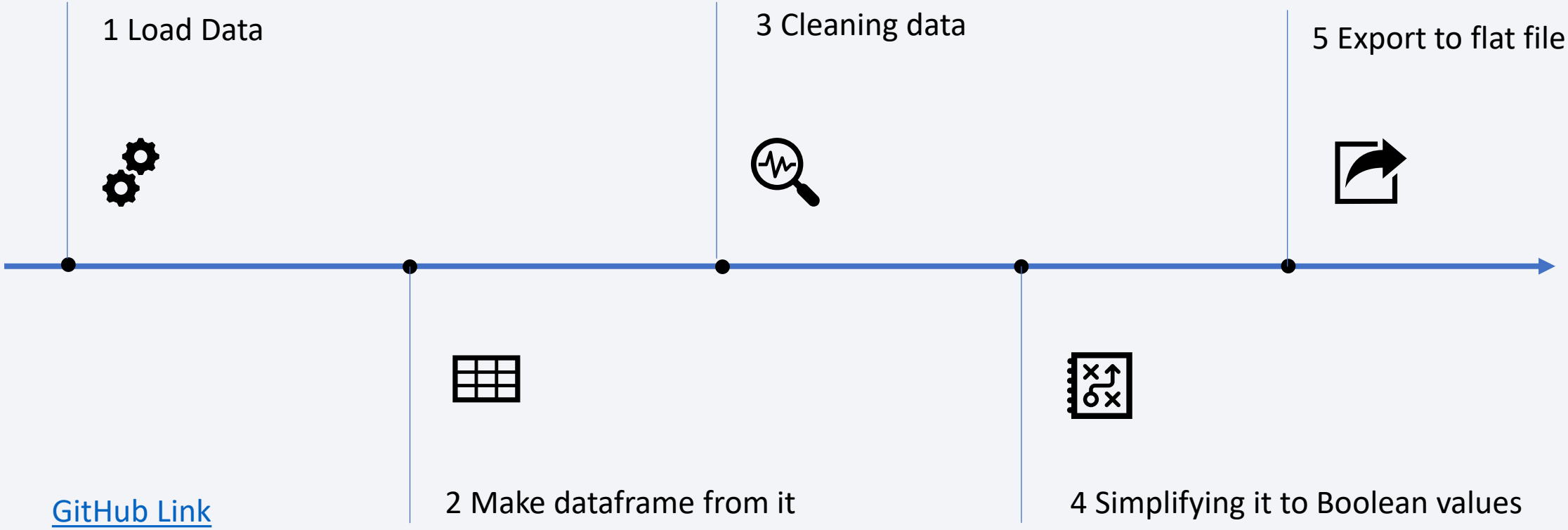
	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.000000	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 Wrangling

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

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

```
df['Class']=landing_class
```



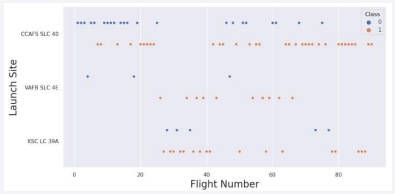
[GitHub Link]

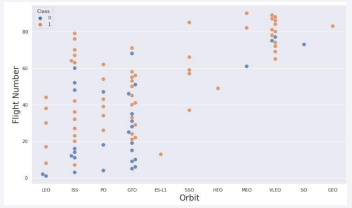
EDA with Data Visualization

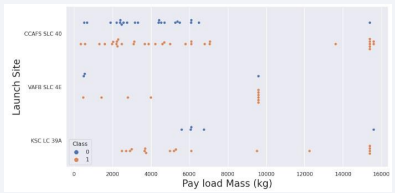
Exploratory data analysis is an approach of analysing data sets to summarize their main characteristics, using statistical Graphics and other data visualization methods.

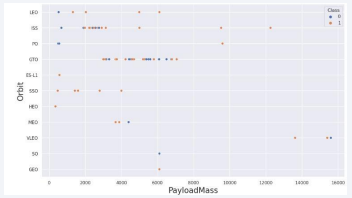
Scatter Graphs Drawn:

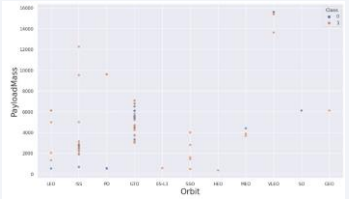
- Payload and Flight Number
- Flight Number and Launch Site
- Payload and Launch Site
- Flight Number and Orbit Type
- Payload and Orbit Type







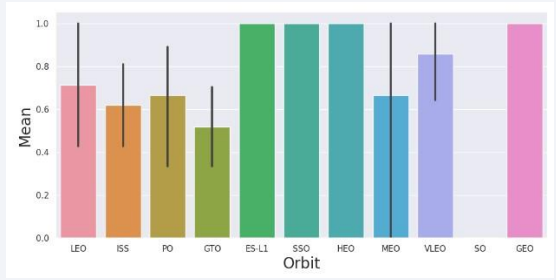




GitHub Link

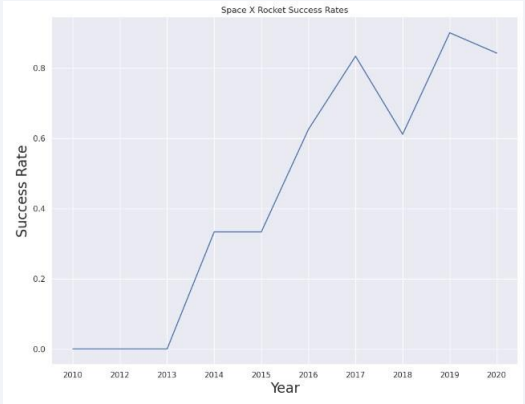
Bar Graph Drawn:

Bar graphs are easiest
To interpret a relationship
Between attributes. Via this
Bar graph we can easily
Determine which orbits
Have the highest probability
Of success.



Line Graph Drawn:

Line graphs are useful in
That they show trends
clearly and can aid in
Predictions for the future.



EDA with SQL

SQL is a crucial tool for working with relational databases and extracting insights from data. IBM Db2 for Cloud is a fully managed SQL database service that can be used by data scientists and analysts to store, manage and analyze large amounts of data.

Using SQL queries, we retrieved various information from a given dataset, including:

- Unique launch site names in the space mission.
- 5 records where launch sites start with 'CCA'.
- Total payload mass carried by booster version F9v1.1.
- Data where drone ship landing outcomes were successful.
- Names of boosters with successful ground pad landings and payload mass between 4000 and 6000.
- Total number of successful and failed mission outcomes.
- Names of booster versions that carried the maximum payload mass.
- Failed landing outcomes in drone ship, their booster versions, and launch site names for the year 2015.
- Ranking of landing outcomes (such as Failure(droneship) or Success(ground pad)) between June 4, 2010, and March 20, 2017, in descending order based on their count.

GitHub Link

Build an Interactive Map with Folium

Folium is a Python library for creating interactive leaflet maps. It allows users to create a variety of map types, including choropleth, heat map, and marker maps, with custom markers and popups. Folium supports various tile providers, including OpenStreetMap, Mapbox, and Stamen, and also enables users to add layers like GeoJSON, Image, and WMS. Folium has a simple and intuitive API, making it easy to use for both beginners and advanced users. It is built on top of the Flask framework and works well with Jupyter notebooks. Additionally, Folium has many features for customization and styling, such as changing colors, font sizes, and map themes.

Map Objects	Code	Result
Map Marker	folium.Marker(	Map object to make a mark on map.
Icon Marker	folium.Icon(	Create an icon on map.
Circle Maker	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.

GitHub Link –This is a link on GitHub that contains all the Jupyter notebooks for Folium.

Build a Dashboard with Plotly Dash

The dashboard includes three plots: a pie chart of the total success launches, a scatter plot of the payload mass versus class of success for all launch sites, and a scatter plot of the payload mass versus class of success for a selected launch site. The dashboard also includes a dropdown menu to select a launch site, a range slider to filter payload mass range, and these are used to update the scatter plots interactively. The first pie chart shows the percentage of total successful launches for all sites or a selected site. The second scatter plot displays the correlation between payload mass and the class of success (success or failure) for all sites, while the third scatter plot shows the same relationship but for a specific launch site.

I added those plots and interactions to the dashboard to visualize and explore the data in an interactive way, which can help users to gain insights, identify trends and patterns, and make data-driven decisions. Each plot or interaction serves a specific purpose, such as displaying the distribution of data, showing relationships between variables, highlighting trends over time, enabling users to filter or select specific data subsets, and so on. By using various plots and interactions, users can explore the data from different angles and answer various questions that they may have about the data.

A Python-based code for a dashboard with Plotly Dash is available on this GitHub link.

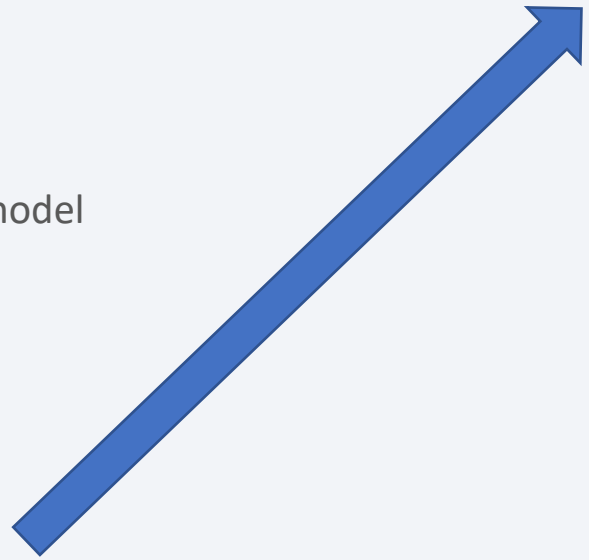
Predictive Analysis (Classification)

Building Model

- 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 out datasets into the GridSearchCV objects and train our model

Evaluating Model

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



Finding Best Performing Classification Model

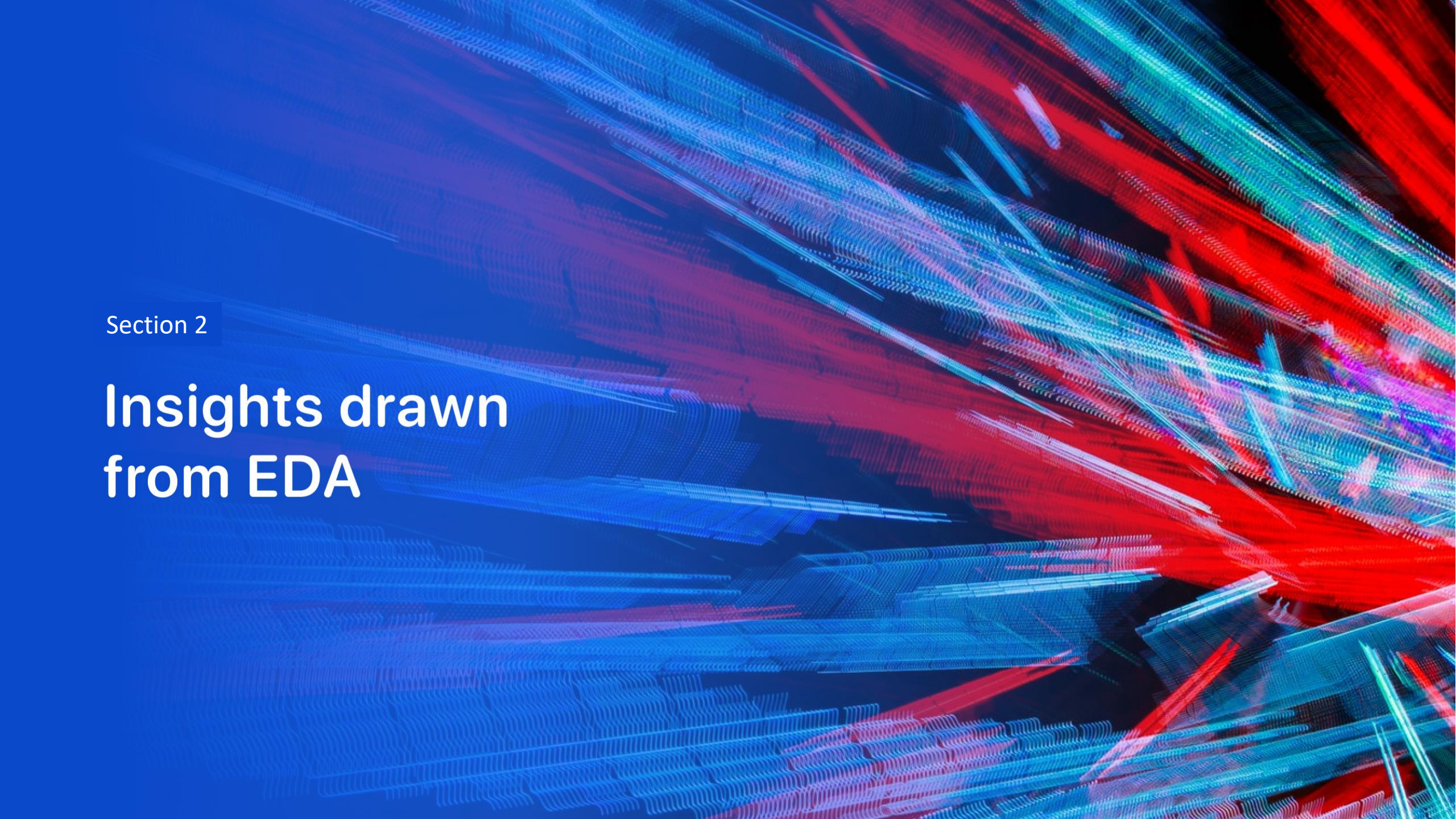
- The model with best accuracy score wins the best performing model

Best Model

GitHub Link

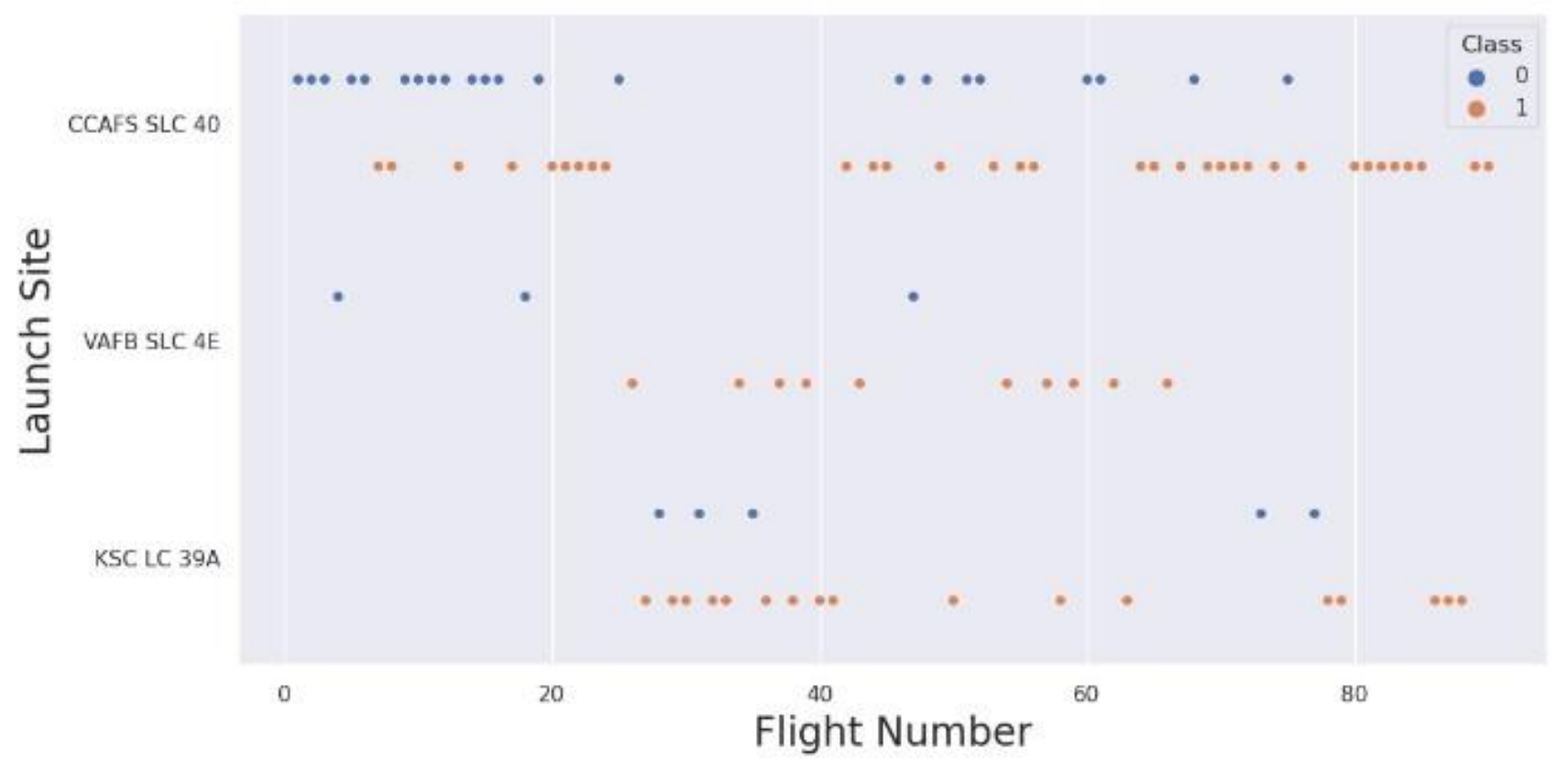
Results

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


Section 2
Insights drawn from EDA

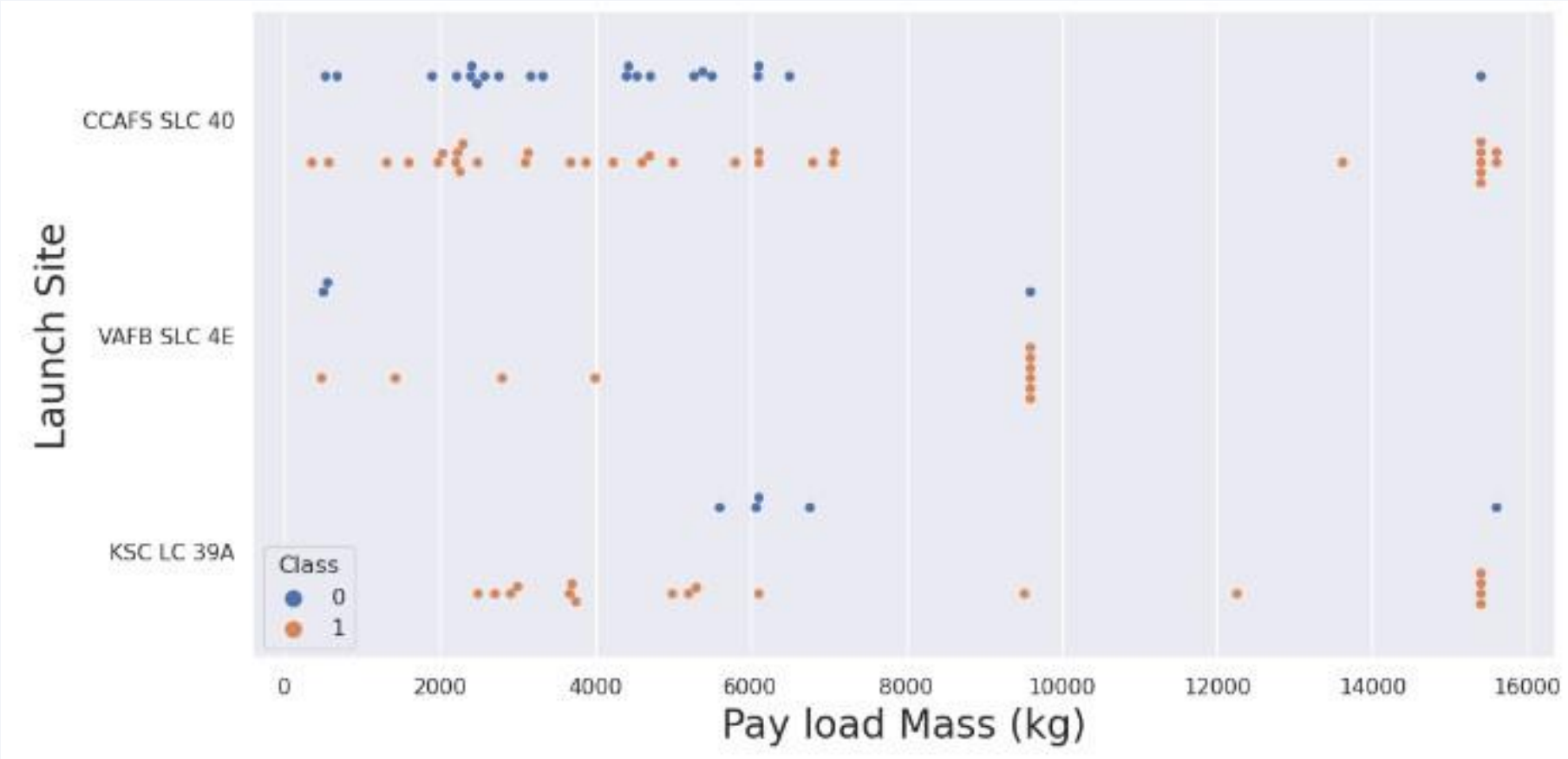
Flight Number vs. Launch Site

The Rocket's performance is improving with each flight, particularly with higher flight numbers. Recent data shows that the success rate for the Rocket increases significantly as flight numbers exceed 30. This indicates that the Rocket is becoming more reliable with increased experience and testing. The improved performance is a positive development, providing greater confidence in the Rocket's ability to carry out successful missions in the future. As the Rocket continues to be tested and refined, it is likely that its success rate will continue to climb, further enhancing its reputation as a dependable launch vehicle



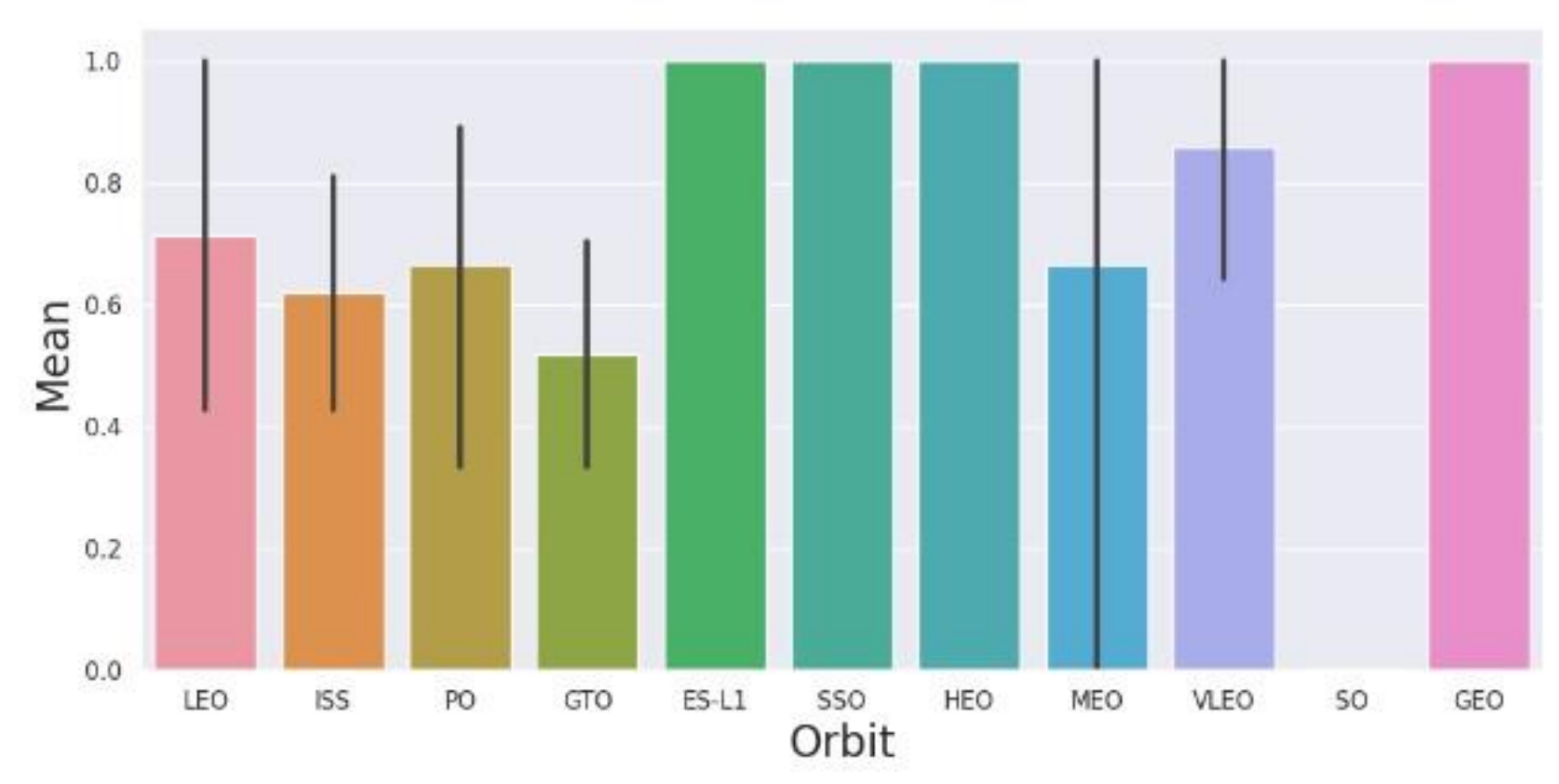
Payload vs. Launch Site

The success rate of the Rocket increases as the payload mass exceeds 7000 Kg, but there is no clear correlation to determine if the launch site is dependent on payload mass for a successful launch.



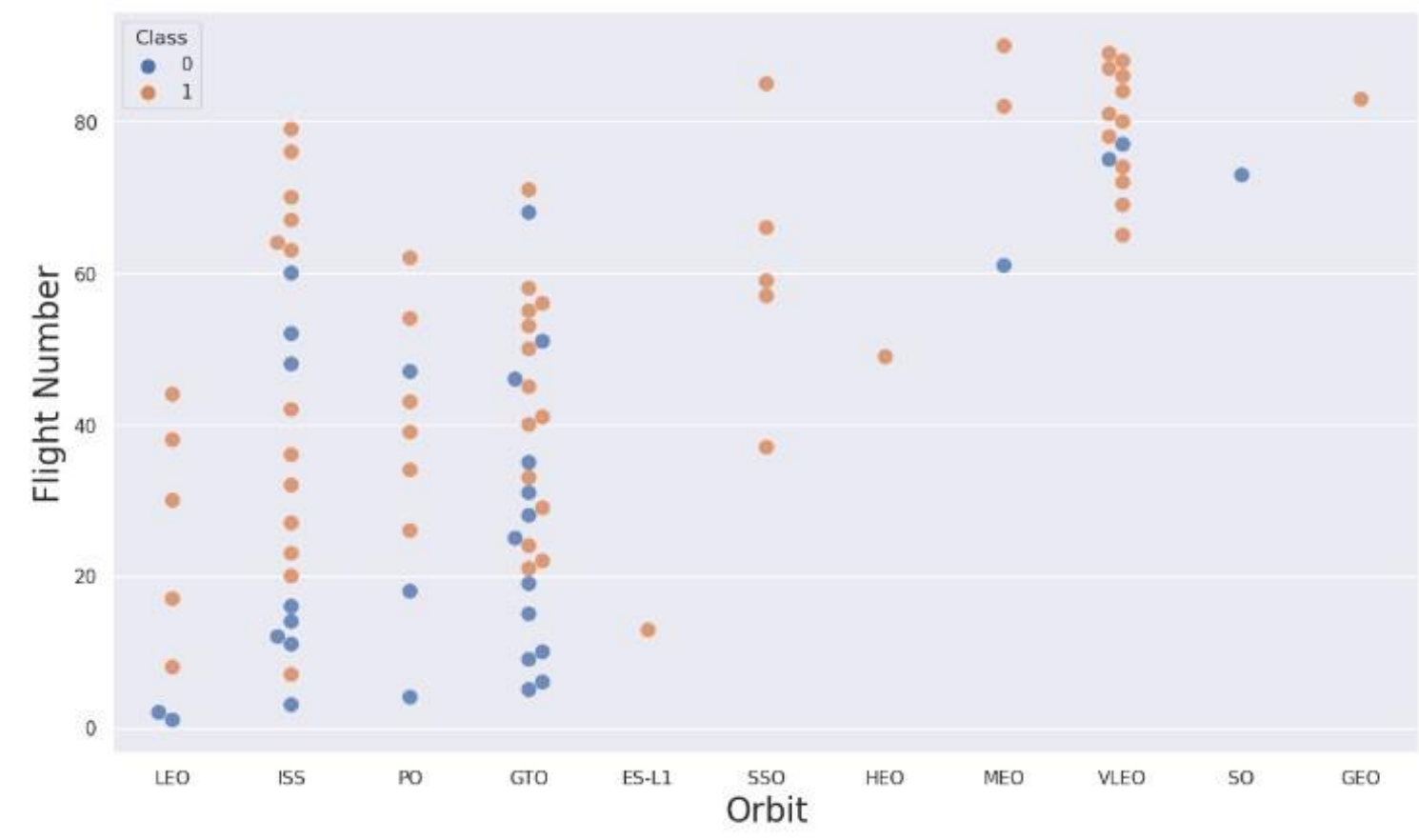
Success Rate vs. Orbit Type

ES-L1, GEO, SSO has highest Success rates.
The graph below represents the mean, which is essentially the same as the success rate.



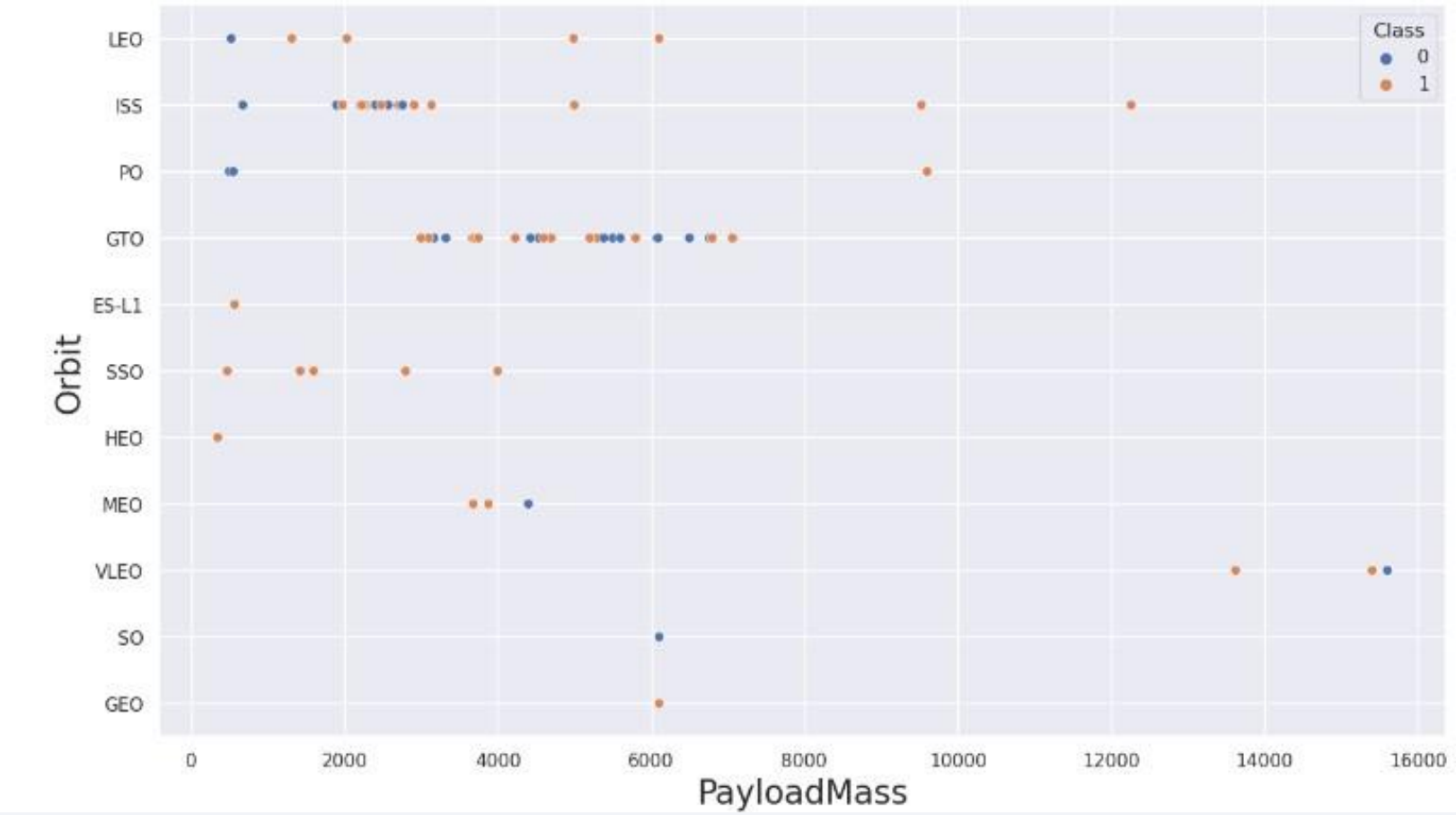
Flight Number vs. Orbit Type

The success rate for LEO orbit improves as the number of flights increase, while the number of flights appears to have no correlation with the success of the GTO orbit.



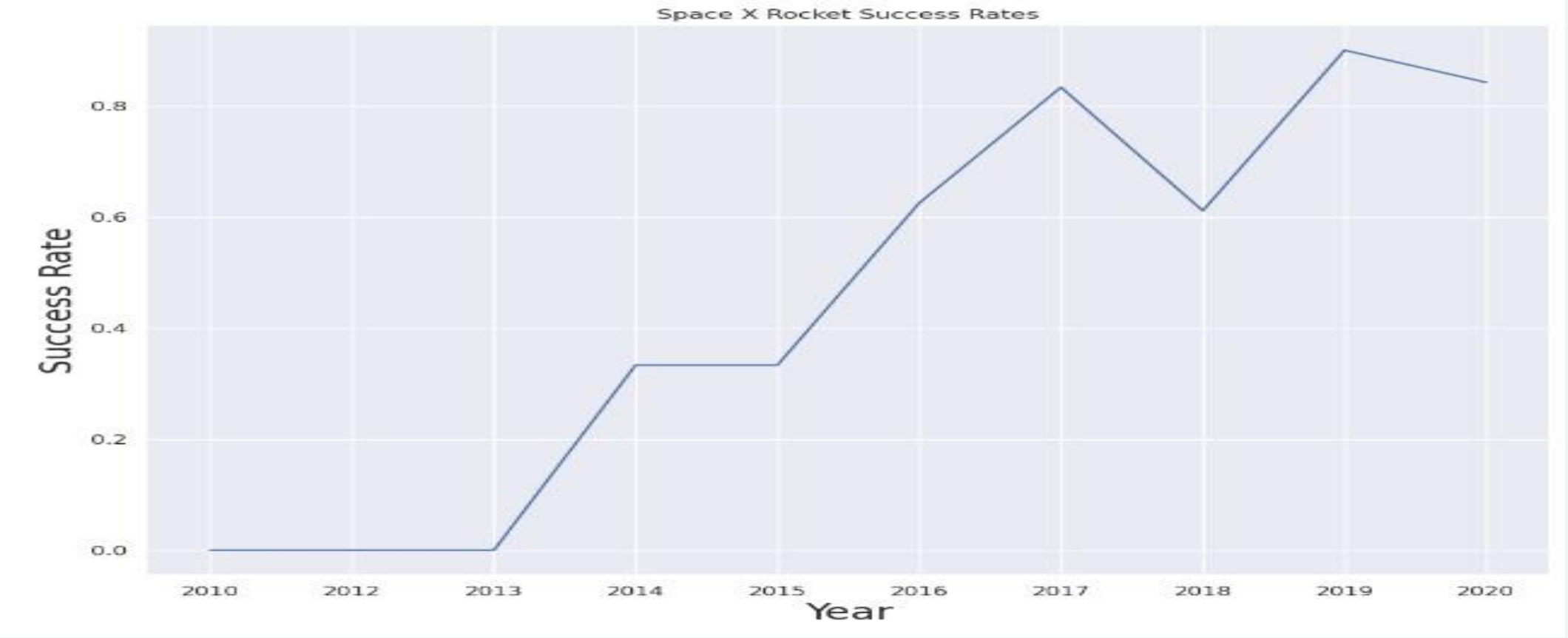
Payload vs. Orbit Type

We observe that heavy payloads have a negative influence on MEO, GTO, VLEO orbits
Positive in 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.



All Launch Site Names

Display the names of the unique launch sites in the space mission

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

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

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'

Display 5 records where launch sites begin with the string 'CCA'

```
%sql SELECT * FROM SPACEXTBL 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 wild Card = ‘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	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

Display the total payload mass carried by boosters launched by NASA (CRS)

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS "Total Payload Mass by NASA (CRS)" FROM SPACEXTBL 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

Display average payload mass carried by booster version F9 v1.1

```
%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

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

First Successful Ground Landing Date

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

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

Description

Using the function MIN works out the minimum data 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

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL 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 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

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

Boosters Carried Maximum Payload

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

```
%sql SELECT DISTINCT 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

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

Booster Versions which carried the Maximum Payload Mass
F9 B5 B1048.4
F9 B5 B1048.5
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1049.7
F9 B5 B1051.3
F9 B5 B1051.4
F9 B5 B1051.6
F9 B5 B1056.4
F9 B5 B1058.3
F9 B5 B1060.2
F9 B5 B1060.3

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT {fn MONTHNAME(DATE)} as "Month", BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL WHERE year(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 cluse ‘Failure (drone ship)’ fetches our required values.

Month	booster_version	launch_site
January	F9 v1.1 B1012	CCAFS LC-40
April	F9 v1.1 B1015	CCAFS LC-40

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

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

```
%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 ;
```

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

Description

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

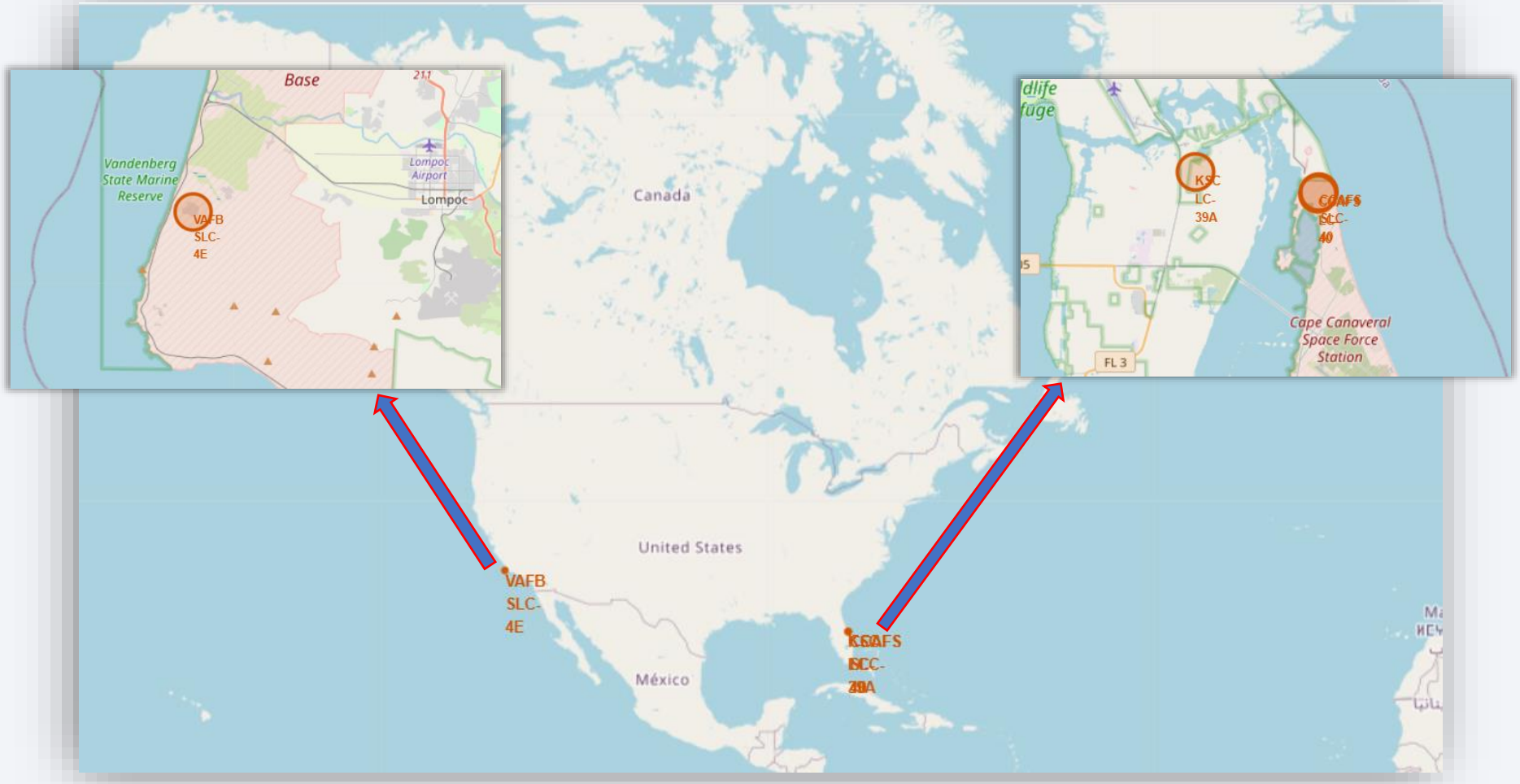
Grouping by LANDING__OUTCOME
Order by COUNT(LANDING__OUTCOME)IN Descending Order.

Section 3

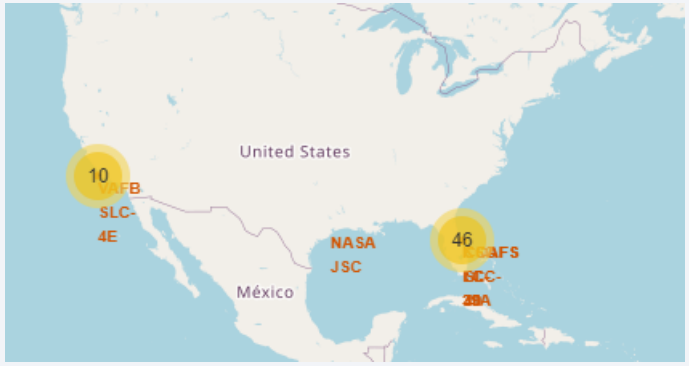
Launch Sites Proximities Analysis

Total Launch Sites by SpaceX on Folium Map

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



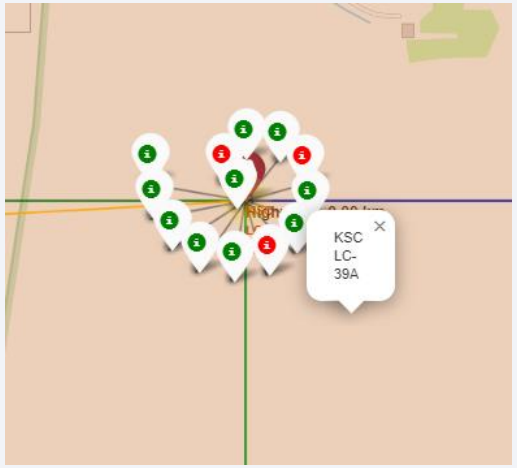
Showing color Labeled Launch Records



Green Marker Shows successful launches and
Red Marker shows failures.

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

VAFB SLC -4E



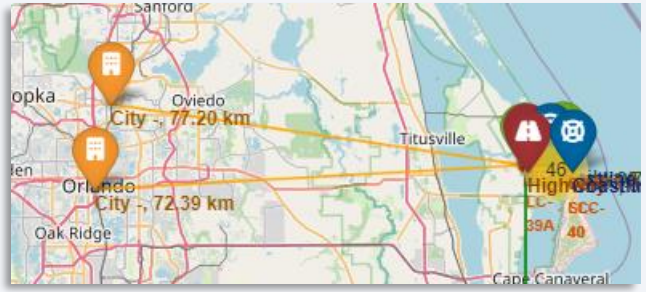
KSC LC-39A

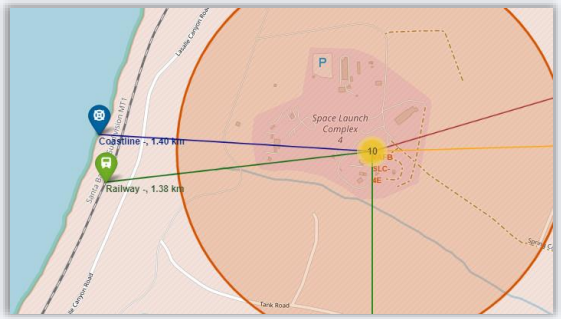
CCAFS LC-40

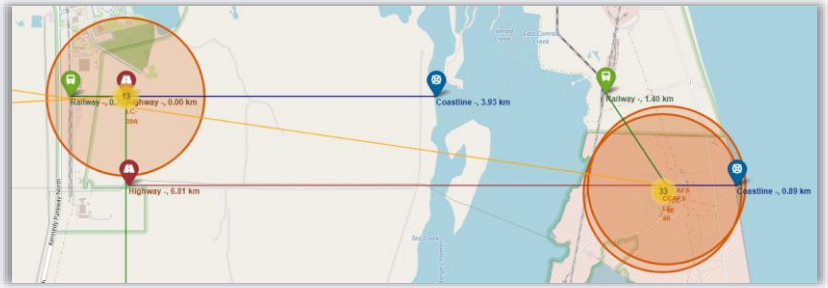
CCAFS SLC-40

Proximity Analysis of Selected Launch Site on Folium Map

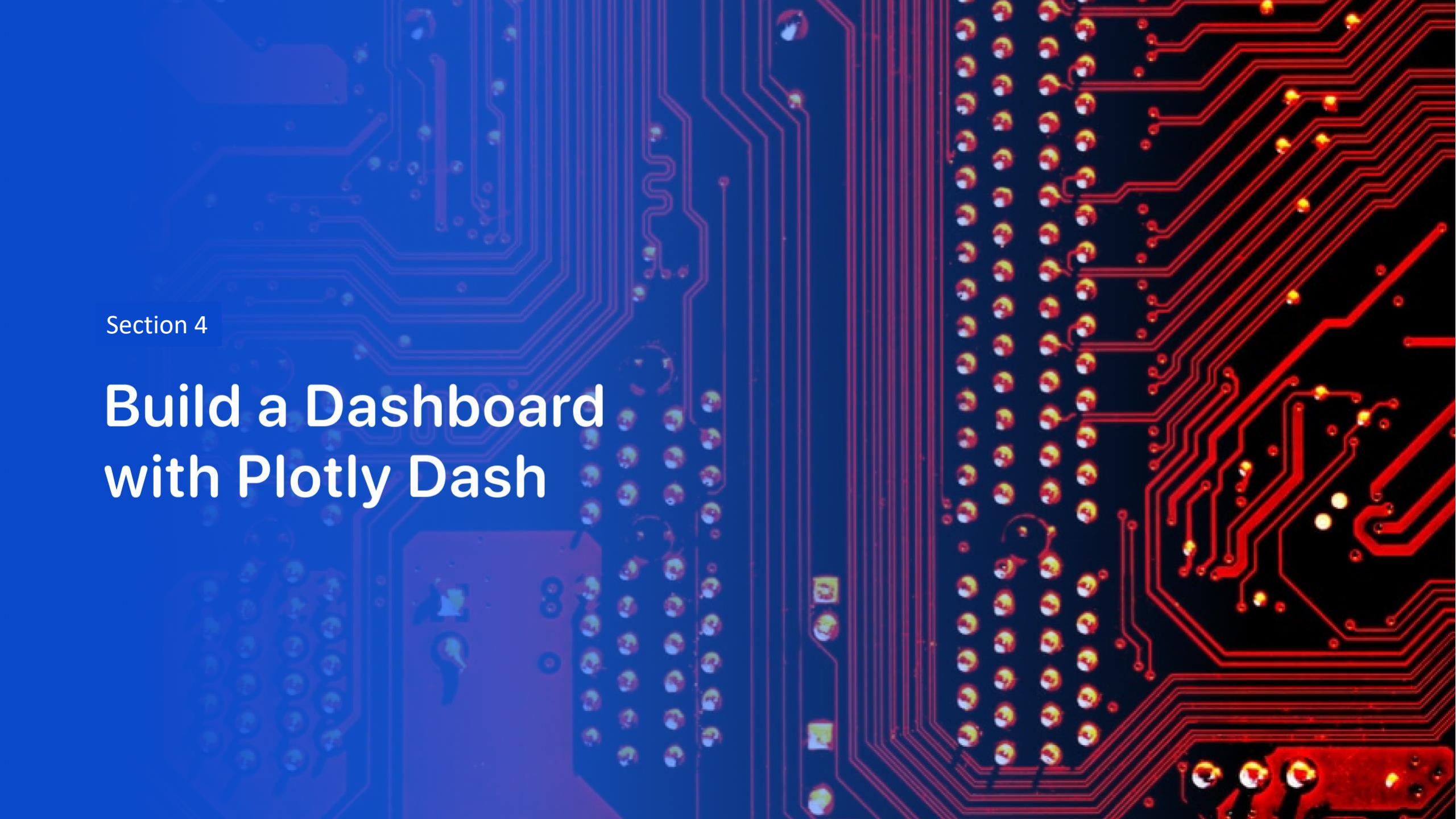
Distance from Equator
Is greater than 3000 Km
For all sites,





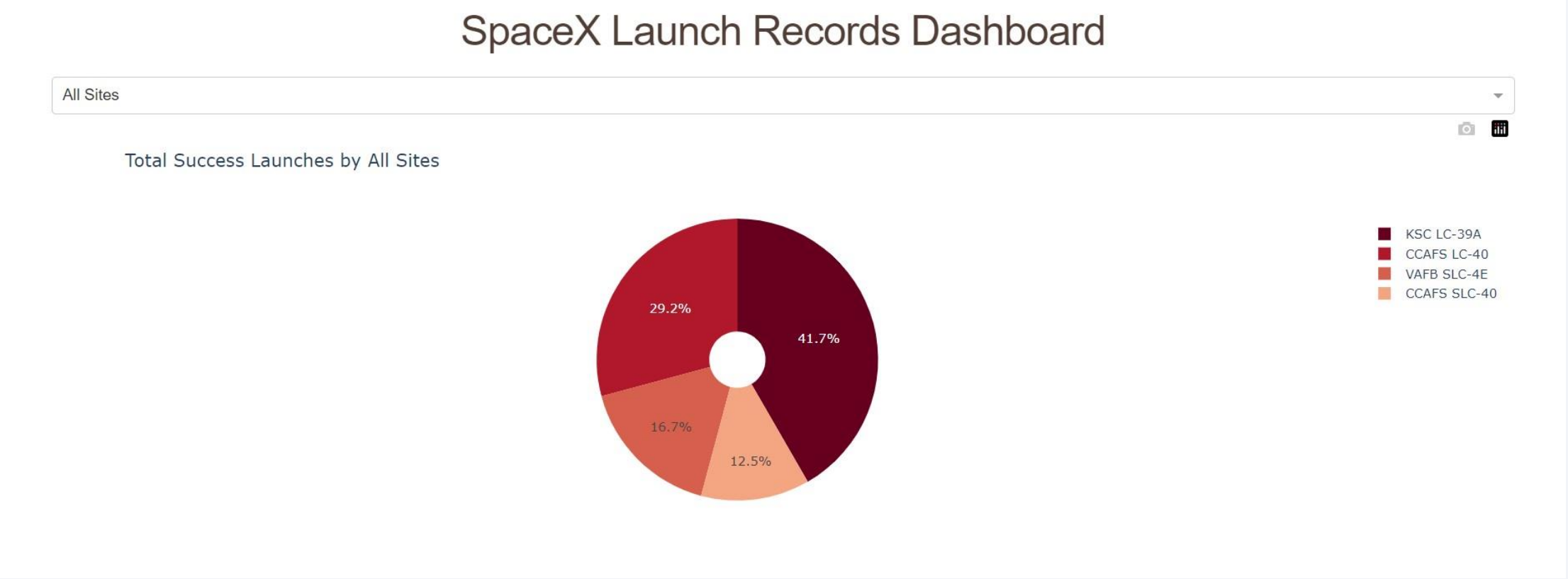


- Launch sites are not so far away from railway tracks
- Launch sites are far away from cities
- Launch sites ate relatively far away from highways


Section 4
Build a Dashboard with Plotly Dash

Launch Success Count for All Sites

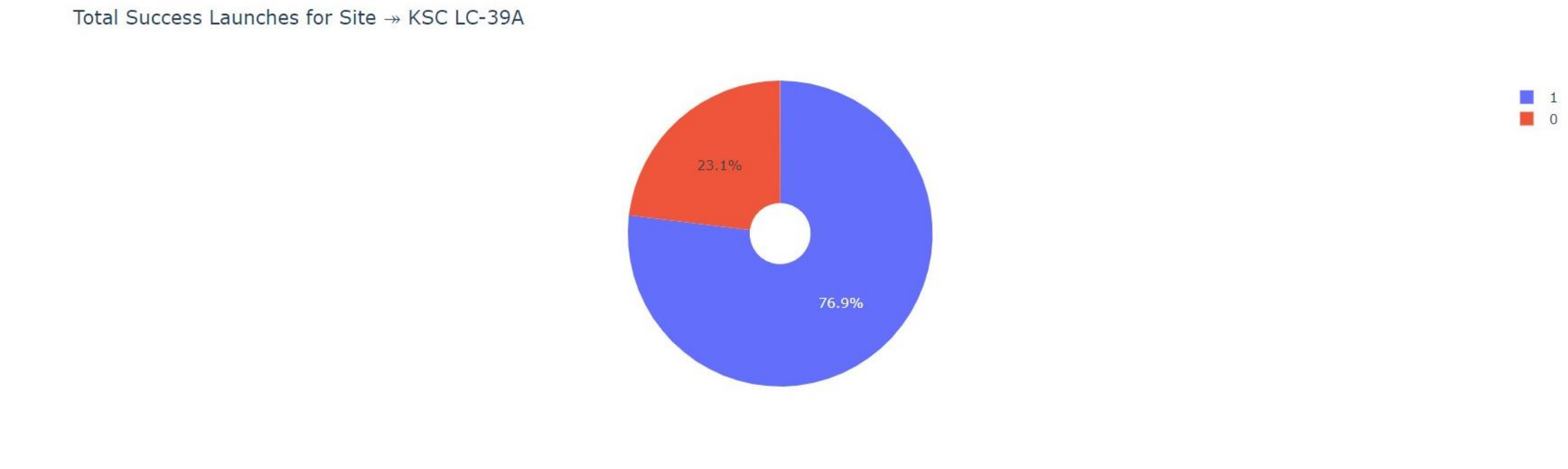
We can see that KSC LC-39A had the most successful launches from all the sites.



Launch Site with Highest Launch Success Ratio

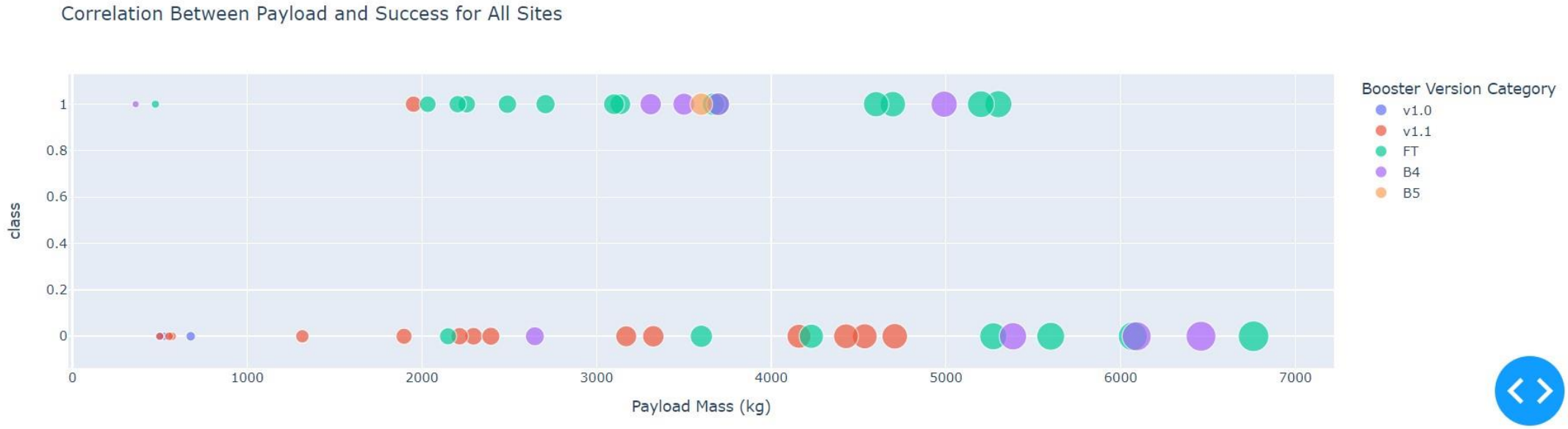
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate.

- **KSC LC-39A** has the highest launch success rate
- **2000 Kg – 10000 Kg** has the highest launch success rate
- **FT** has the highest launch success rate



Payload vs Success for all sites Correlation

Correlation Between Payload and Success for All Sites

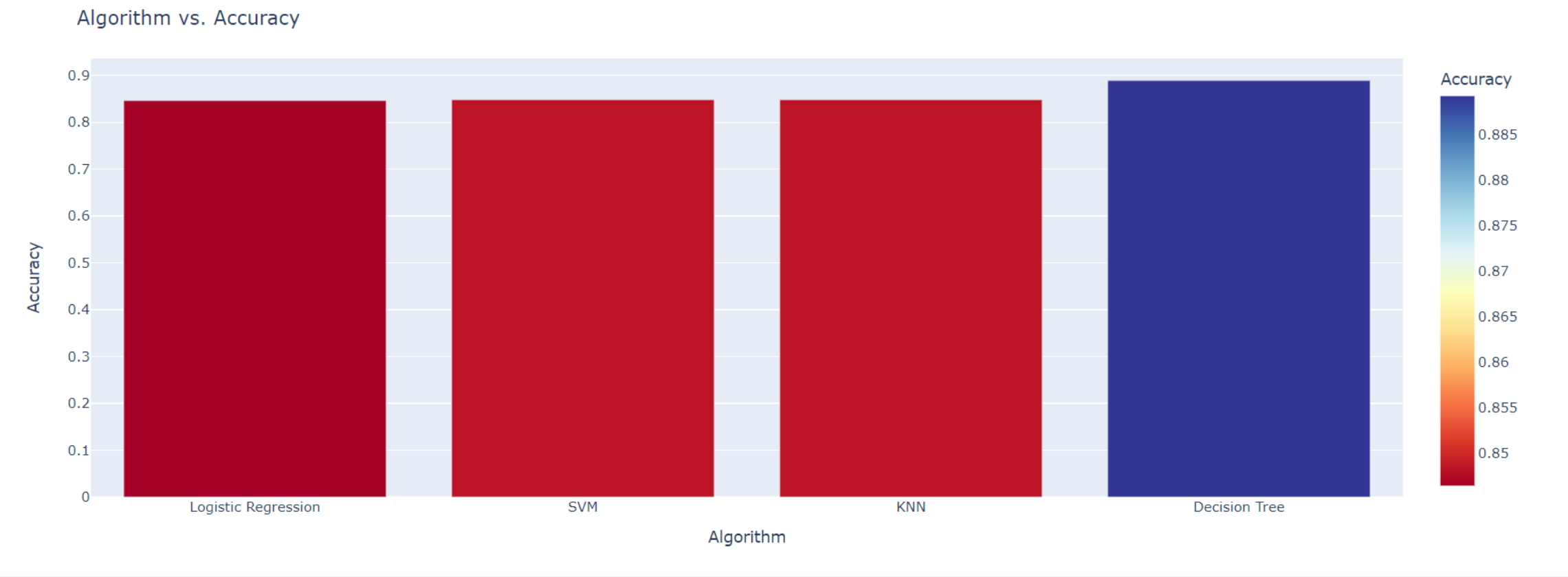


Section 5

Predictive Analysis (Classification)

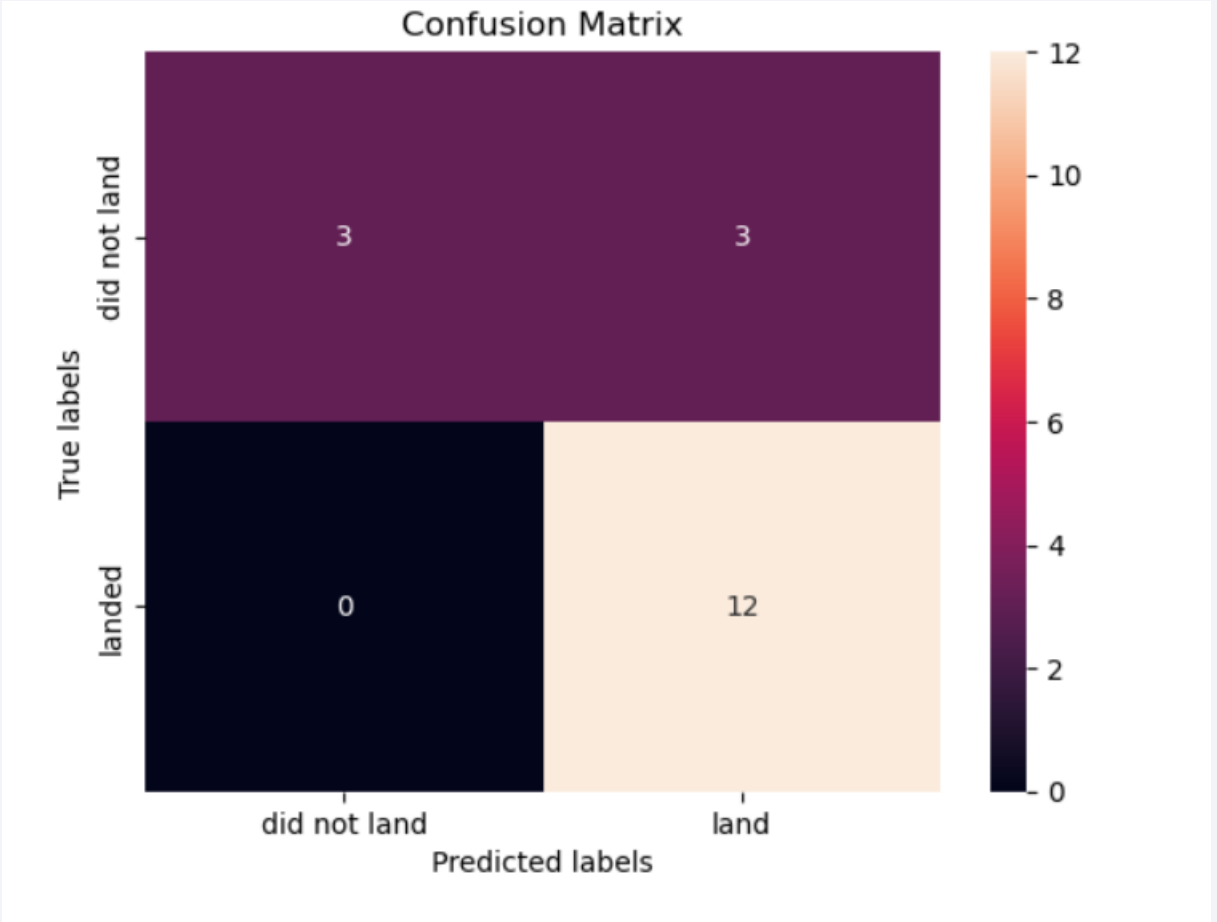
Classification Accuracy

As you can see our accuracy is extremely close, but we do have clear winner which performs best – “**Decision Tree**” With a score of 0.889286.



Confusion Matrix

For all the Models such as **SVM**, **KNN**, **Logistic Regression**, **Decision Tree** we have a common Confusion Matrix with same **True Negative**, **False Negative**, **False Positive**, **True Positive**



Conclusions

1. The highest success rates are found in orbits ES-L1, GEO, HEO, and SSO

2. SpaceX's launch success rates have been improving over the past few years, and soon they will reach their target.

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

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

Appendix

You can access my entire project via Jupyter notebooks on GitHub by clicking here

Thank you!