IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Bryan David Vasquez Paz
November 27th 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Collection data with the SpaceX API and cleaning using Python with Pandas.
  - EDA with visualization and SQL queries.
  - Geospatial Analysis of the launch sites with Folium.
  - Success rate evaluation with Dashboards
  - Classification Machine learning model with Scikit-Learn
- Summary of all results
  - Correlation and characteristics of different variables in the data set like payload mass average, first success landing in drone ship, best orbit types and launch sites. See conclusion and insight graphics.
  - Selection of a Decision Tree model like the best model to predict the success or unsuccess of a launch.

Introduction

- Project background and context

The lead of this space race is the company SpaceX, which has demonstrated that sending spacecraft to the international Space Station creates a system to provide satellite internet access and send manned missions to Space. One reason for its position is the ability of its famous rocket (Falcon 9) to be reused in the first stage which reduces the cost of each launch from 165 million dollars to 62 each.

SpaceX's Falcon 9 Can recover the first stage. Sometimes the first stage does not land. Sometimes it will crash as shown in this clip. Other times, Space X will sacrifice the first stage due to the mission parameters like payload, orbit, and customer.

- Problems you want to find answers

Using public information we want to find:

1. What are the principal variables in a success launch?
2. Instead of using rocket science, Can we use machine learning models to predict if SpaceX will reuse the first stage in each launch?

Section 1

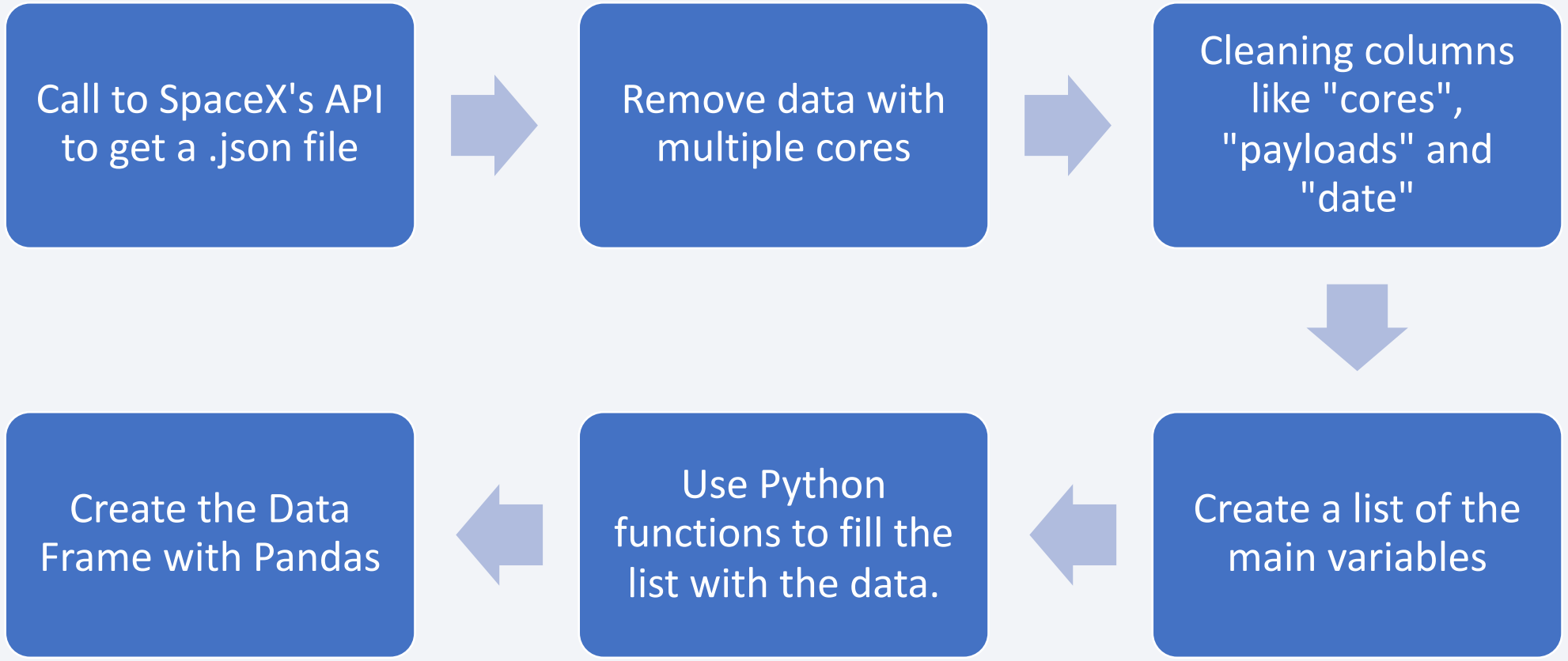
Methodology

Methodology

Executive Summary

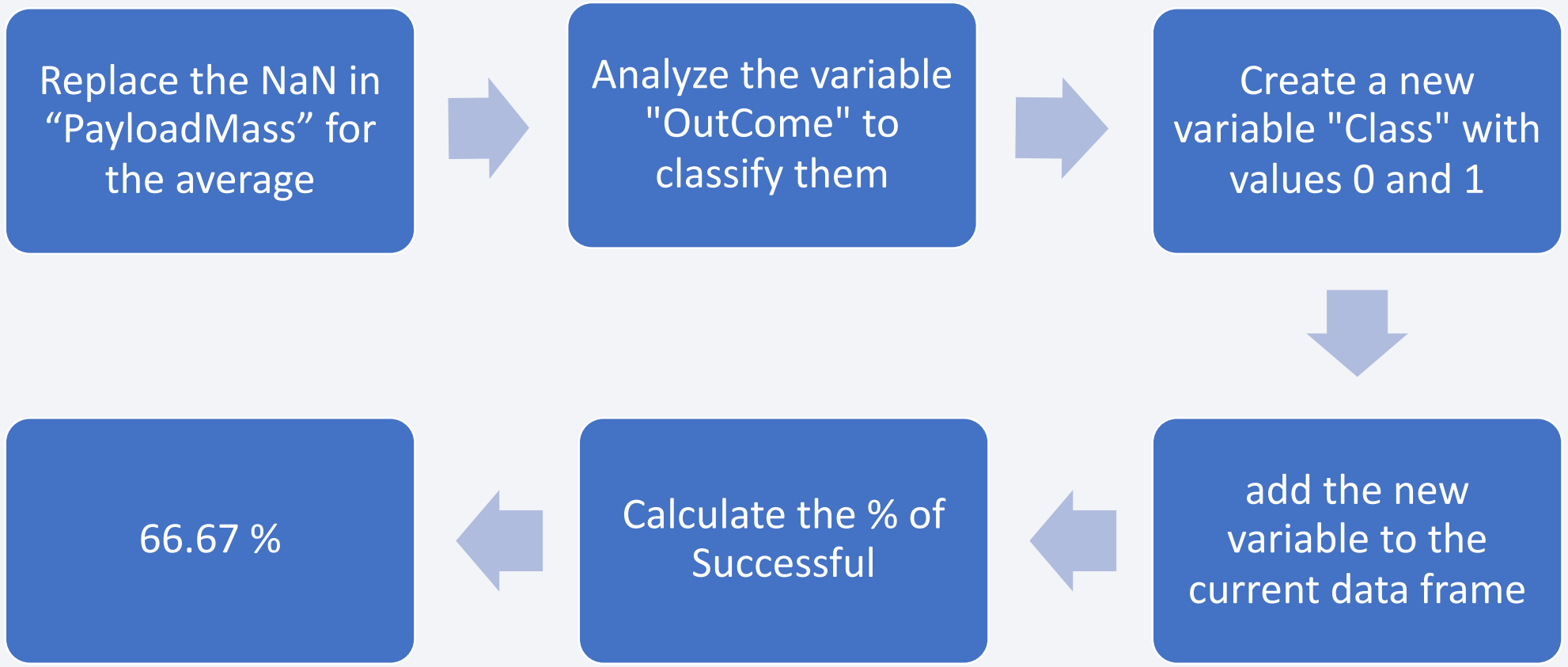
- Data collection methodology:
  - Data was collected through Calls to the SpaceX API and Specific Python function for this case.
- Perform data wrangling
  - Data was proceeded and cleaned with pandas, and creating a new class to show the success of each launch.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - We build, tuned, and validated different classification models with the libraire Scikit-learn

Data Collection – SpaceX API



https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 1-Data_Colection_API

Data Wrangling



https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 2-Data_Wrangling

EDA with Data Visualization

- “Launch Site” Relationships differentiated by success and unsuccess launch:
  - Flight Number and Launch Site
  - Payload and Launch Site
- “Orbit type” Relationships:
  - Orbit type vs Success rate
  - Flight Number and Orbit type
  - Pay load mass and Orbit type
- Launch Success yearly trend

https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 4-EDA_With_Visualization

EDA with SQL

- Find the names of the Launch sites in the space mission.
- Total Payload mass carried by boosters launched by NASA (CRS).
- Average Payload mass carried by booster version F9 v1.1.
- Date of the first success landing.
- Names of the boosters which have success landing in drone ship.
- Total of successful and failure mission outcomes.
- Booster with the higher payload mass.
- Landing Outcome Rank

Build an Interactive Map with Folium

- Markers with the names of each launch site.
- Markers with the success/failed launches for each site on the map.
- Lines comparing different positions respect to the launch site.

With the name of each launch, we now know each site's exact position. Also, with success/failed markers, we can differentiate on the map which launch site has a better rate and analyze its geo position.

In the end, we can see the comparison of distances between different locations on the map.

https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 5-Interective_Analitics_Folium

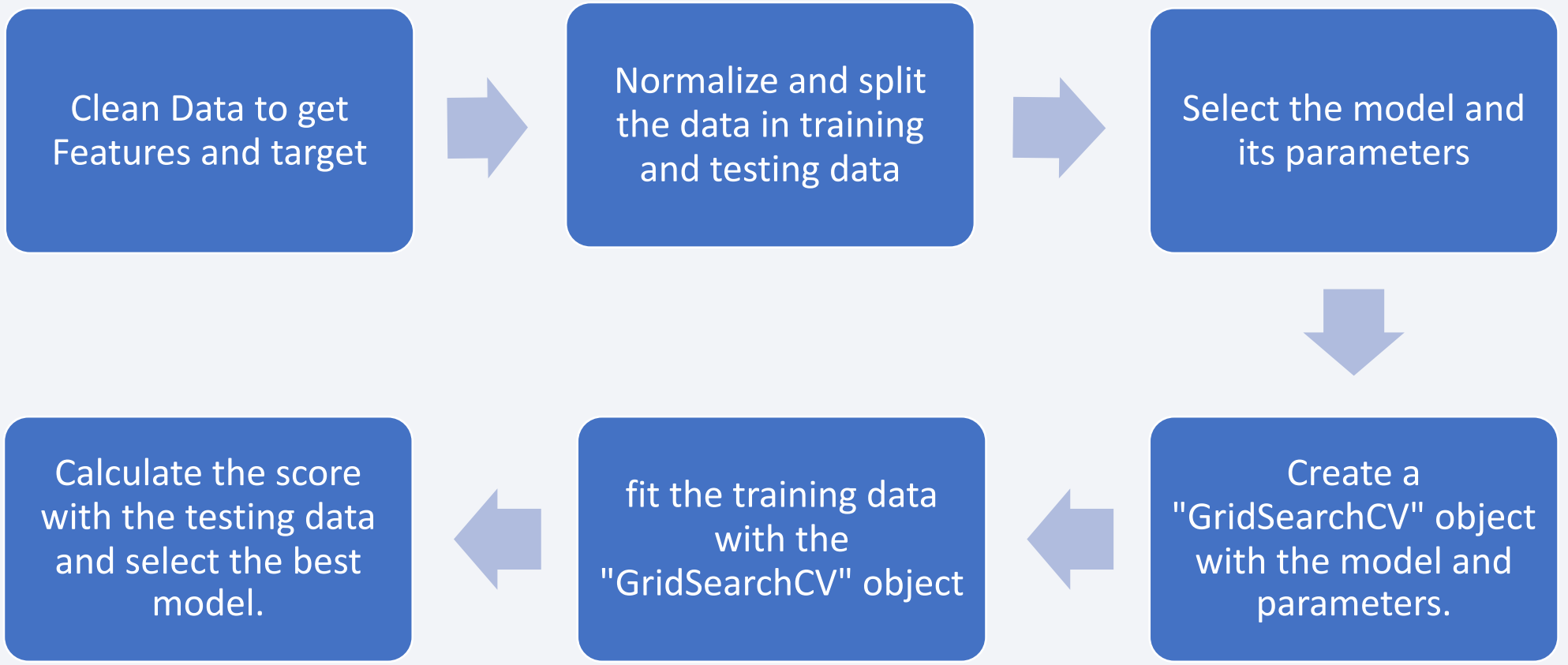
Build a Dashboard with Plotly Dash

- With All sites
  - We can see a pie chart comparing the total success rate for each launch site and a scatter plot showing the correlation between the Payload mass and Success rate for all sites classified by the booster version.
- With each site
  - We can see a pie chart with the total success rate for a specific launch site and we can see the same scatter plot but with just the information of the specific launch site.

In each graphic you can change the range of the pay load mass.

https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 6-Dashboard_Plotly_Dashk

Predictive Analysis (Classification)



https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git

7-Predictive_Analysis(Classification)

Results EDA_SQL

- With the EDA_SQL I found that exist four different launch sites:
  CAAFS LC-40, CCAFS SLC-40, KSC LC-39A, VAFB SLC-4E
- The total payload carried by boosters launched by NASA (CRS) is 45596 Kg and the average carried by booster Falcon 9 v1,1 is 340 Kg.
- The unique booster with success landing on a drone ship is the F9 FT B10(22, 26, 21.2, 31.2).
- The success rate in mission outcomes is approximately 99.0%.
- The booster version which has carried the maximum payload mass is the F9 B5.

https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 3-EDA_SQL

Results EDA

- When the flight number increases is clear that the success rate improves and at the same time the payload mass.

- The launch site CCAFS SLC 40 has the biggest number of unsuccessful cases compared to the other launch sites.

- Around 10000 Kg to higher values of payload mass the success rate improves a lot and the launch VAFB SLC 4E doesn’t have launches with payload mass bigger than 10000 kg
- The mission Orbit types with the best success rates are ES-L1, GEO, HEO, SSO and the lowest success rate is the orbit type GTO.

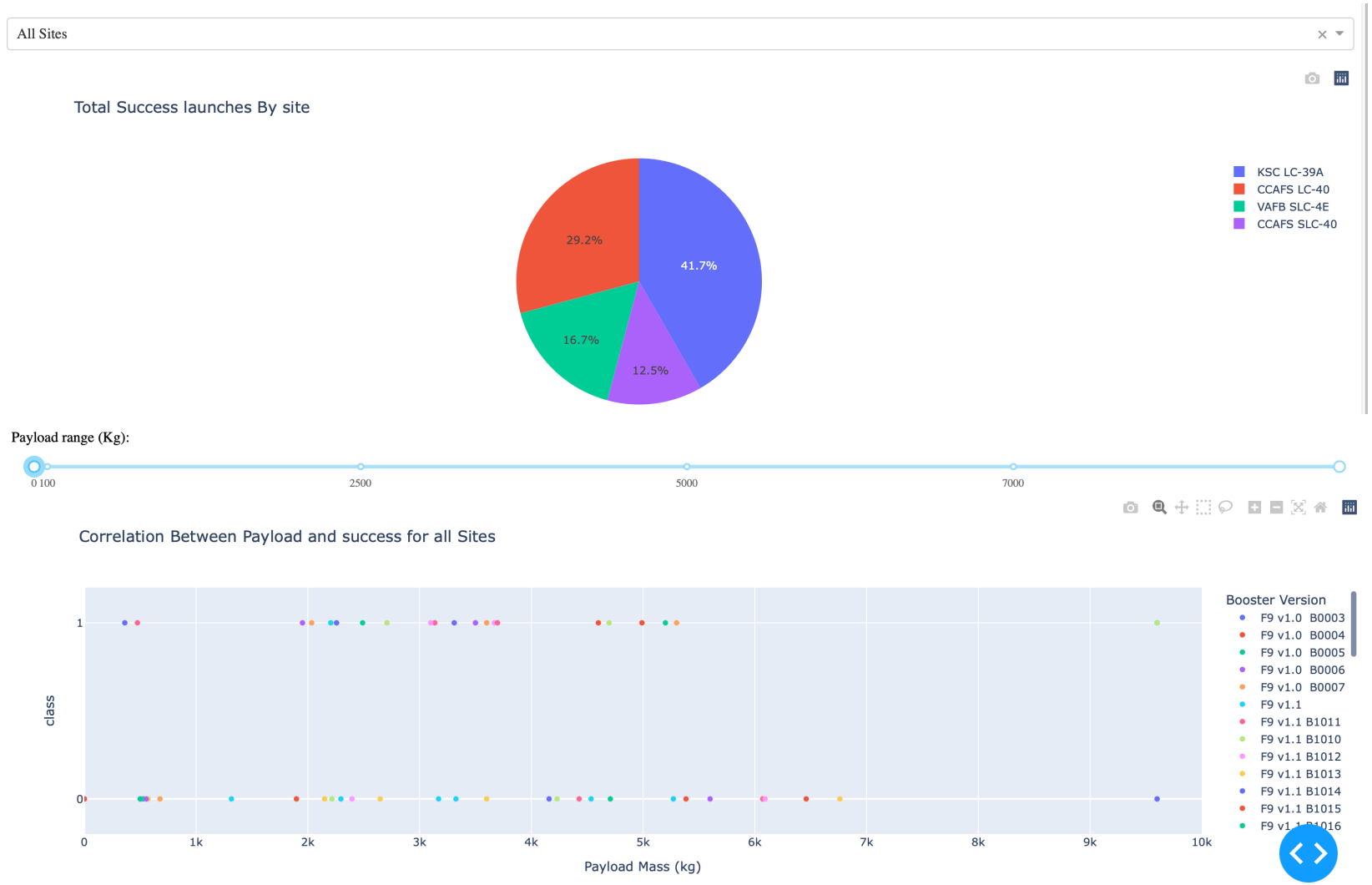
- There is no relationship between the flight number and payload mass with the variable Orbit Type GTO.

- Clearly, 2013 started the first successful lading outcomes.

https://github.com/Bryan-V-Dev/Data-Scienc-Capstone-spaceX.git 4-EDA_With_Visualization

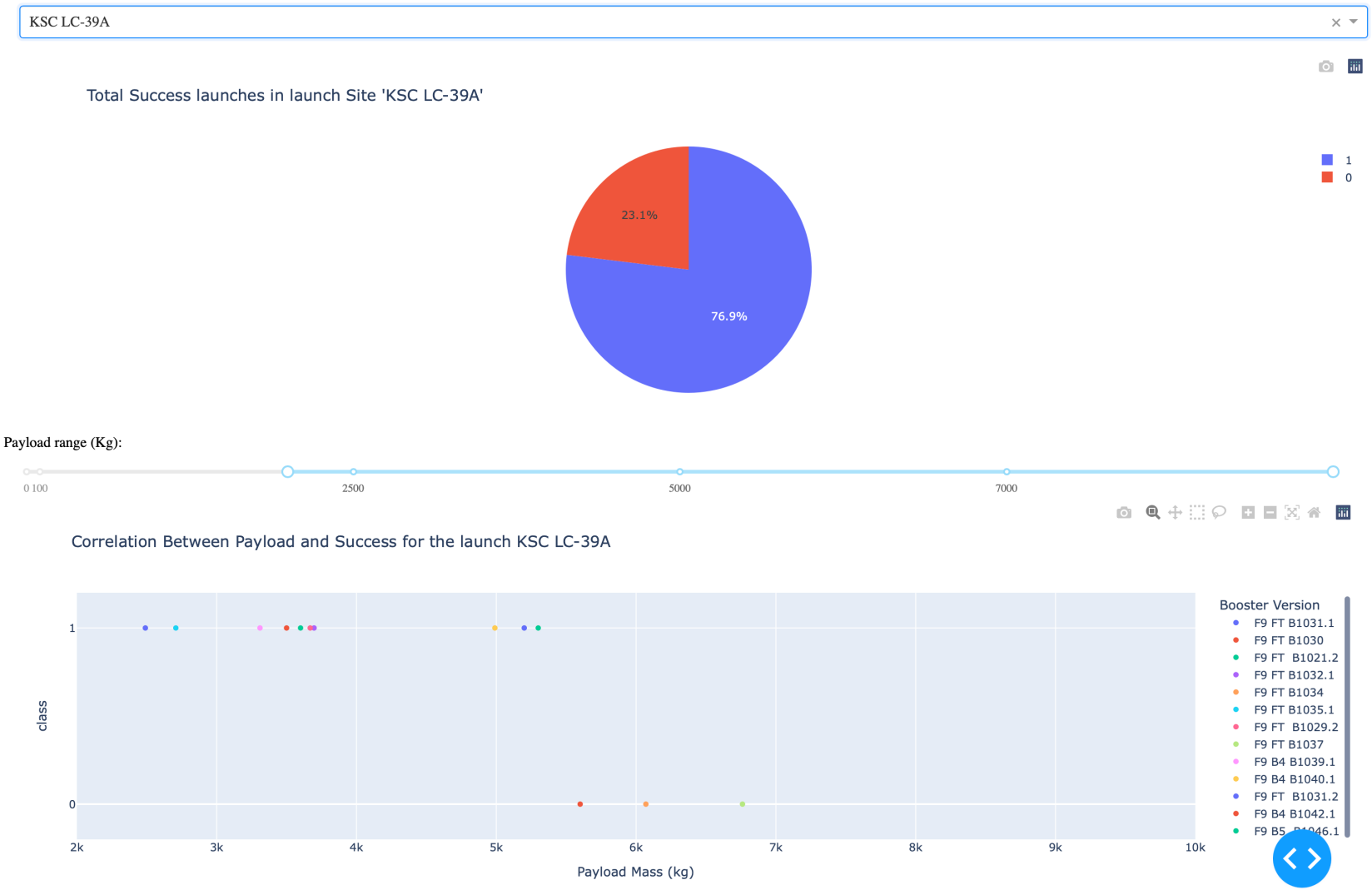
Results Interactive demo

- All sites



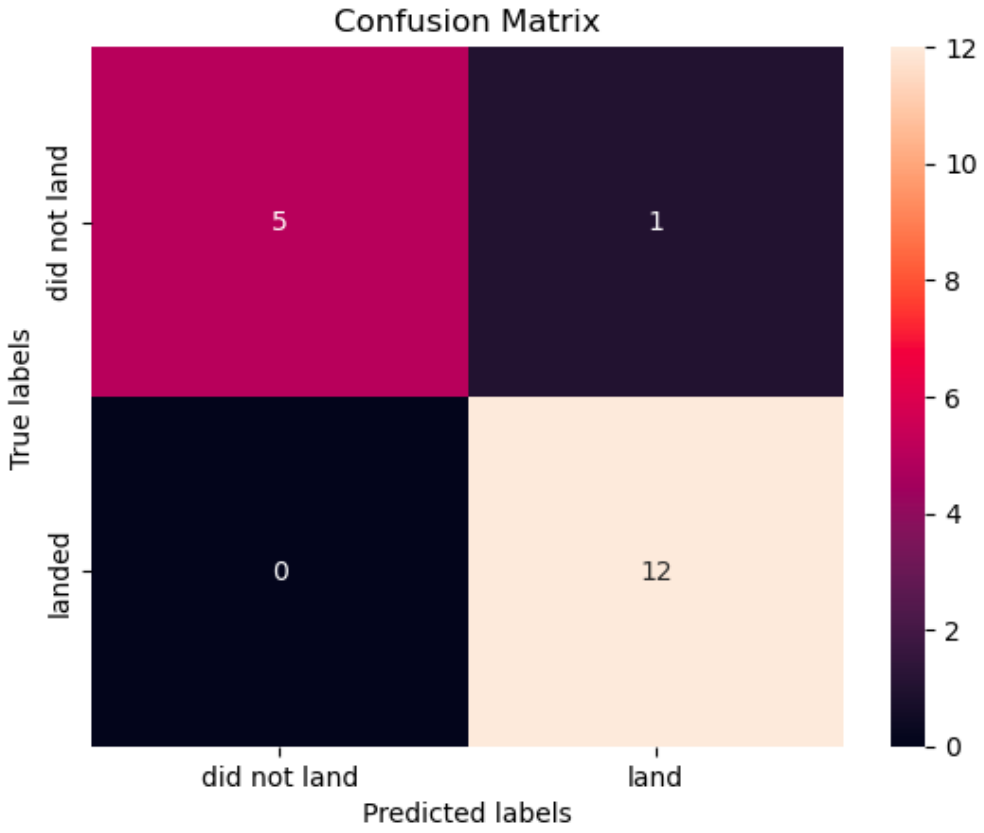
Results Interactive demo

- Each Site



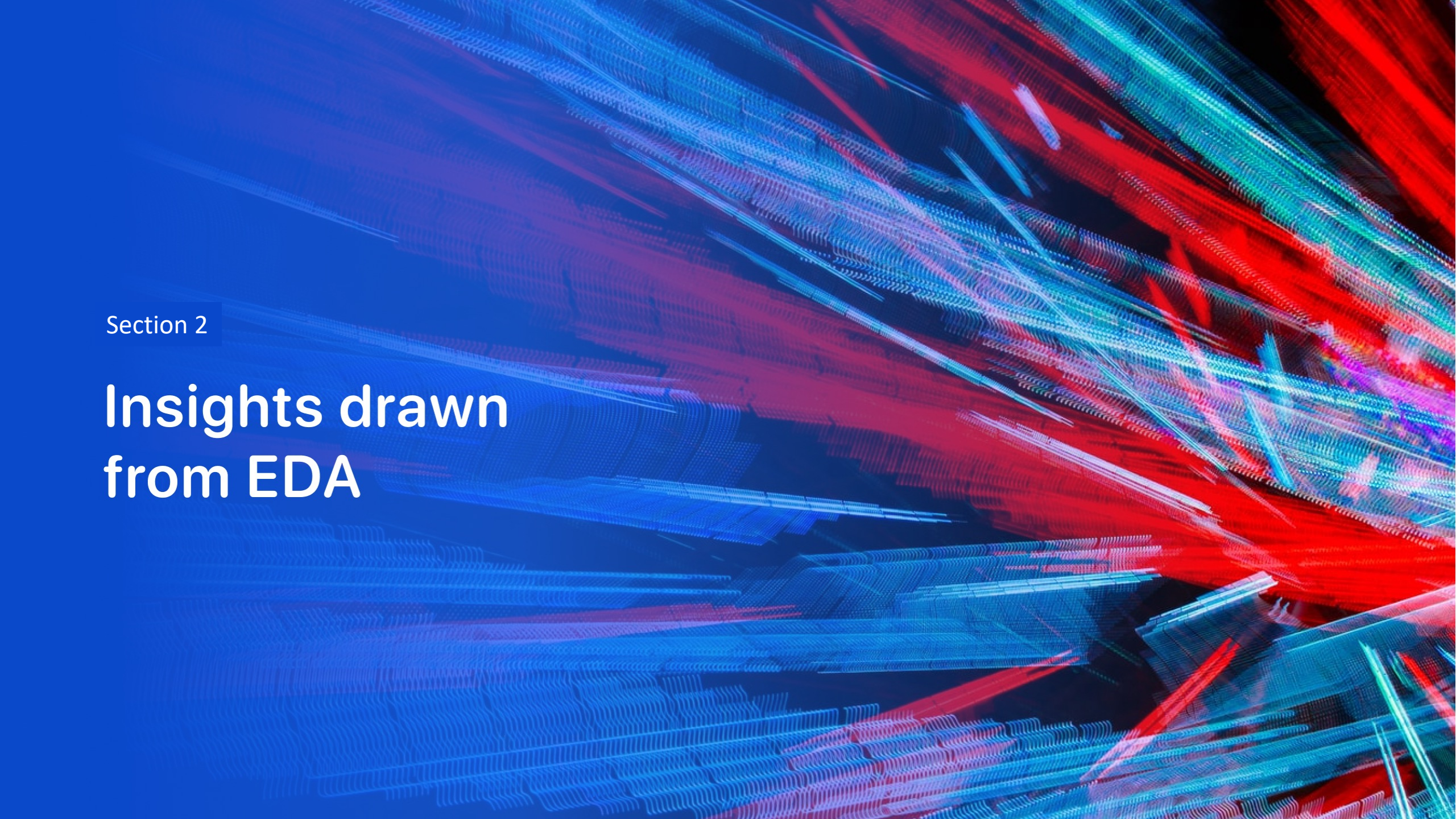
Results

The best Confusion matrix for a Decision Tree model

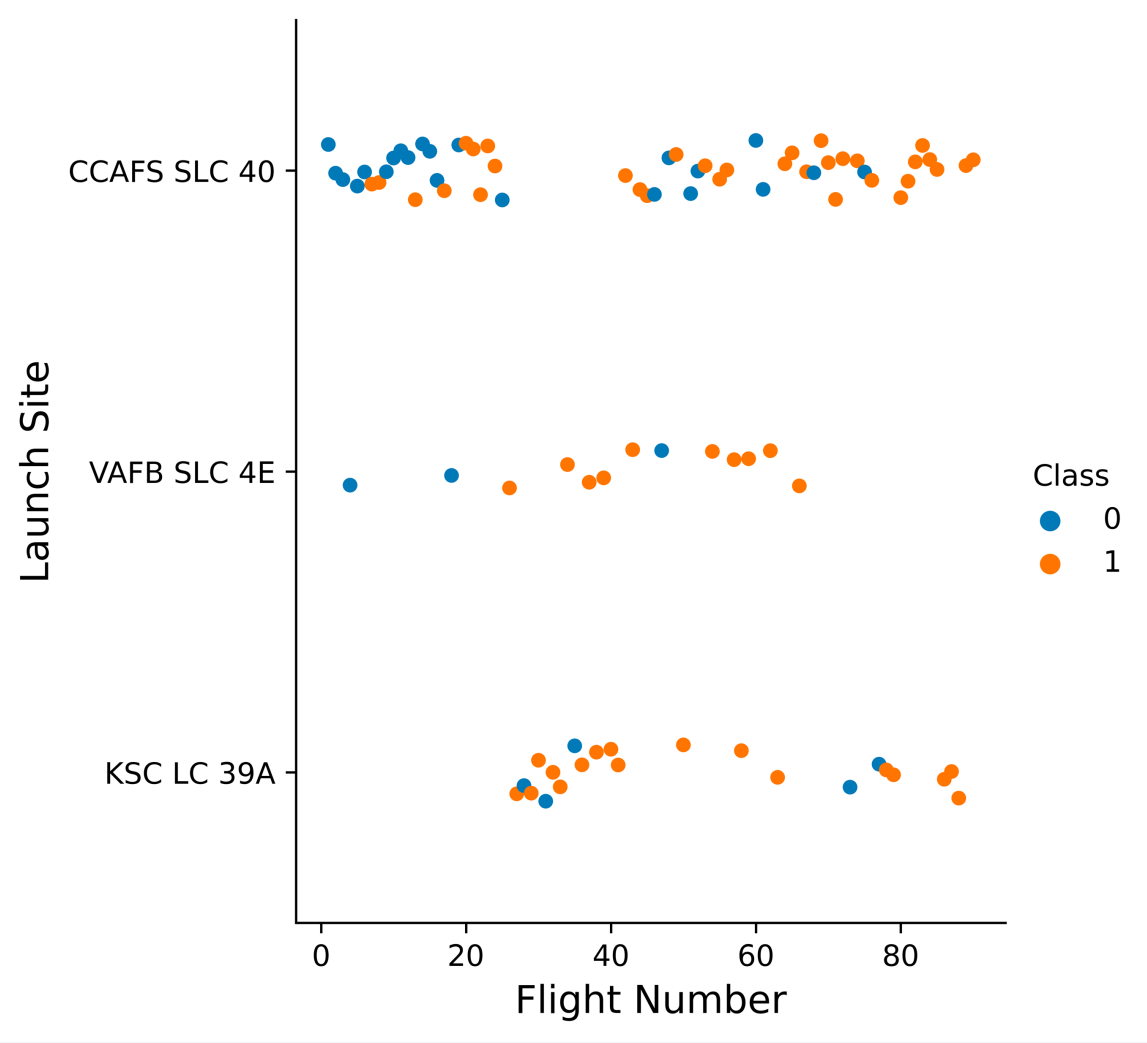


Models and scores with the test data.

Model	Score (%)
Logistic Regression	83.3
SMV	83.3
Decision Tree	94.4
KNN	83.3

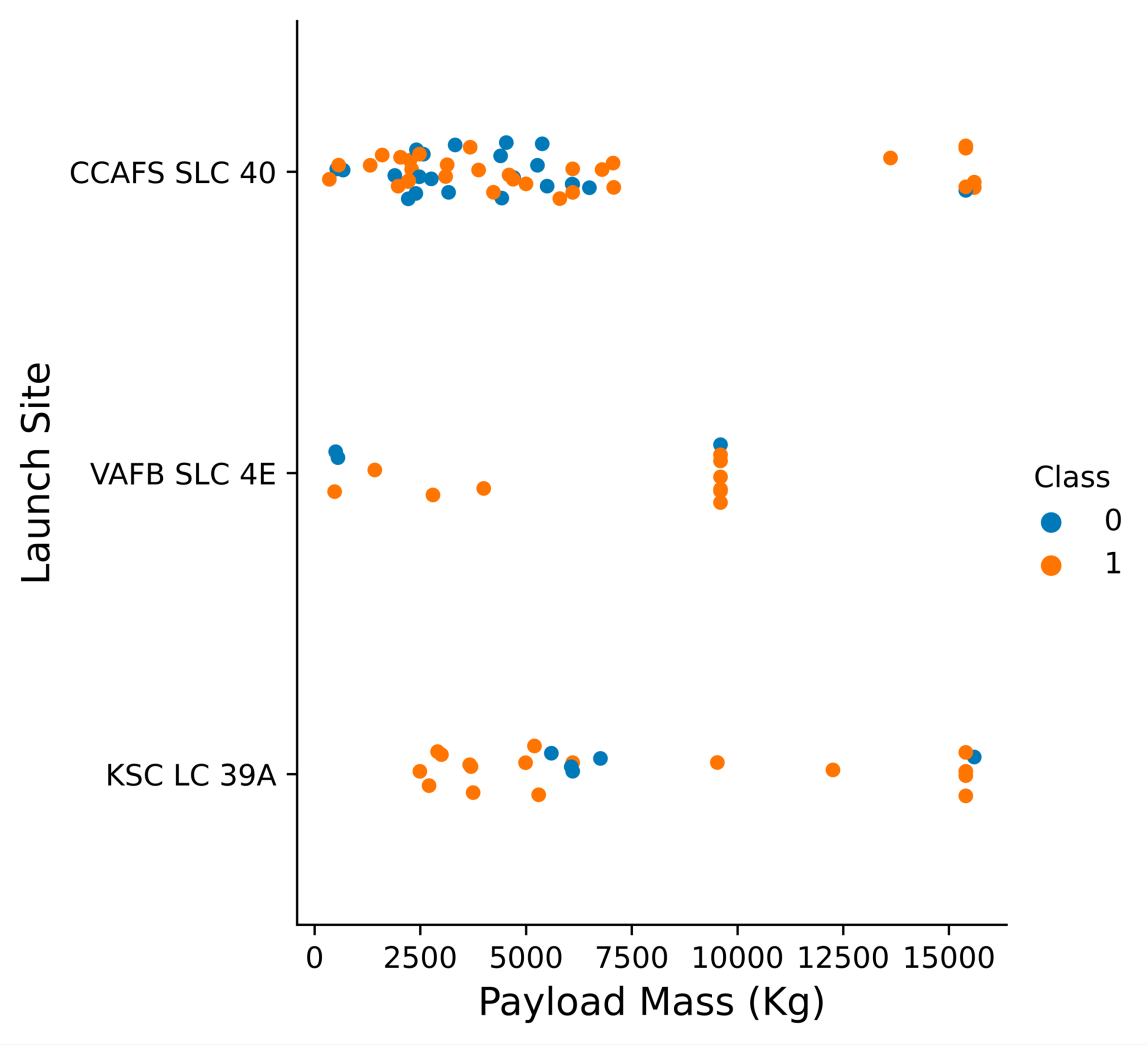

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



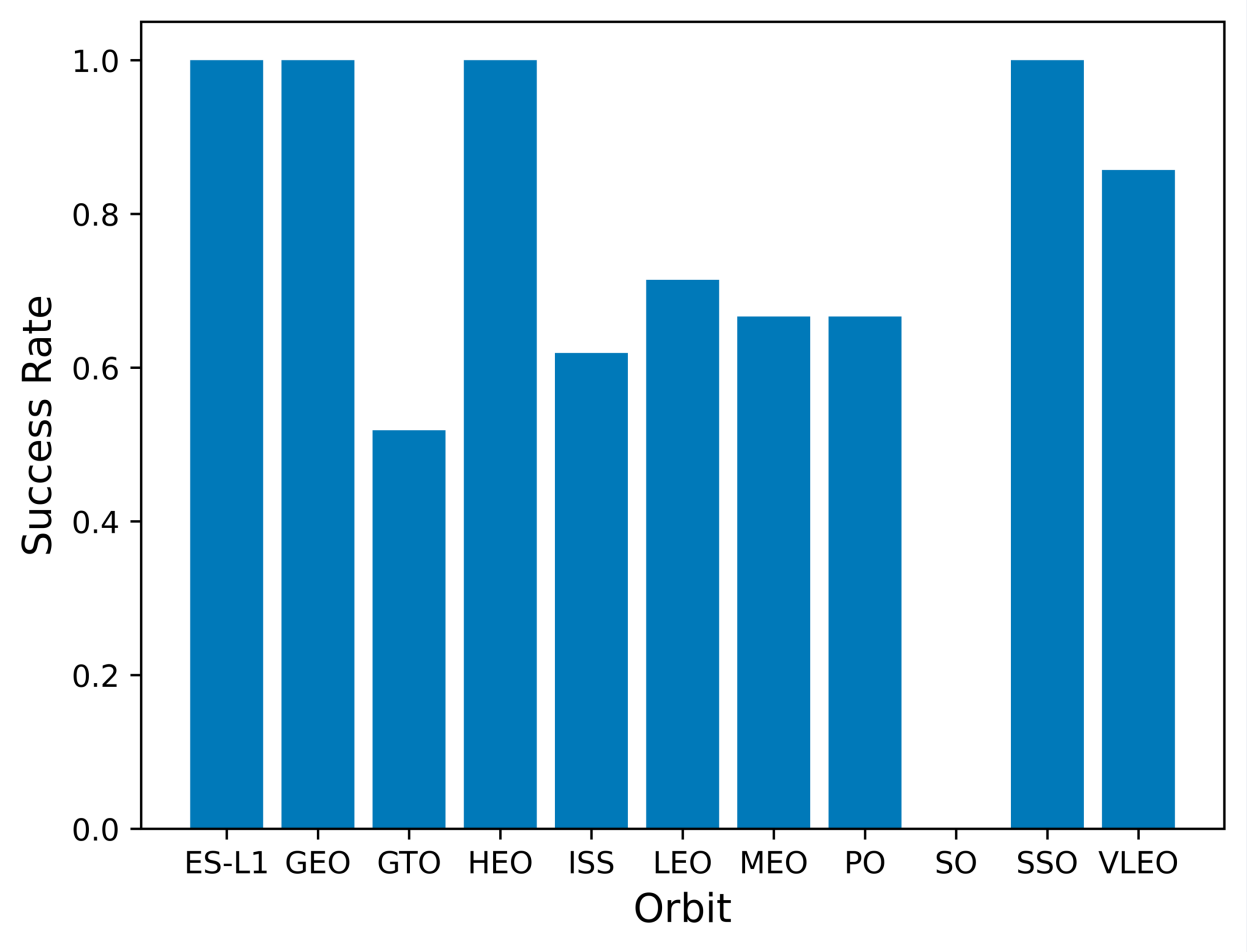
We are able to see that when we increment the number of flights well increases the success rate, but the most important feature in this graphic is the launch site "CCAFS SLC 40" has the biggest number of unsuccessful cases compared to the other launch sites.

Payload vs. Launch Site



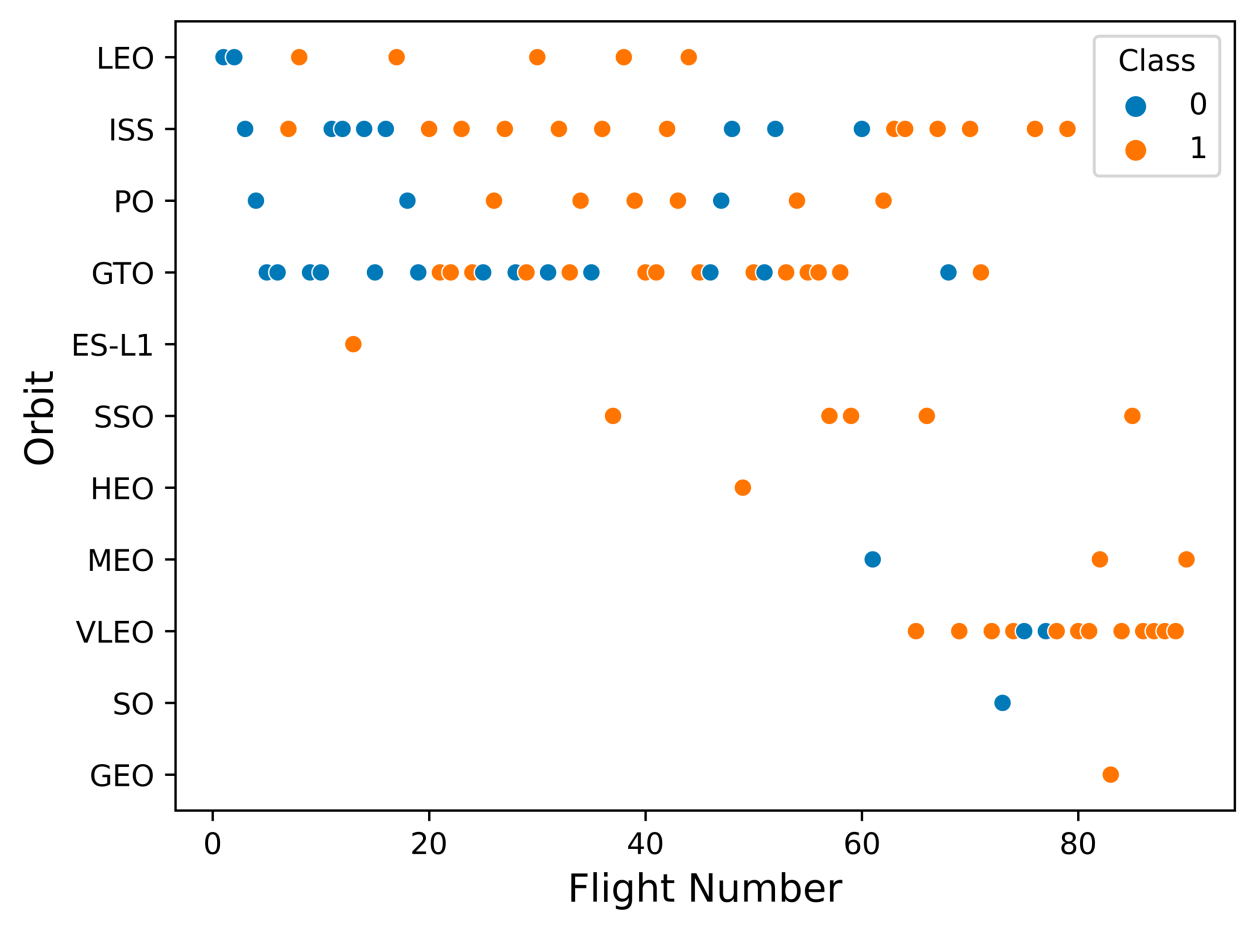
Now if we observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

Success Rate vs. Orbit Type



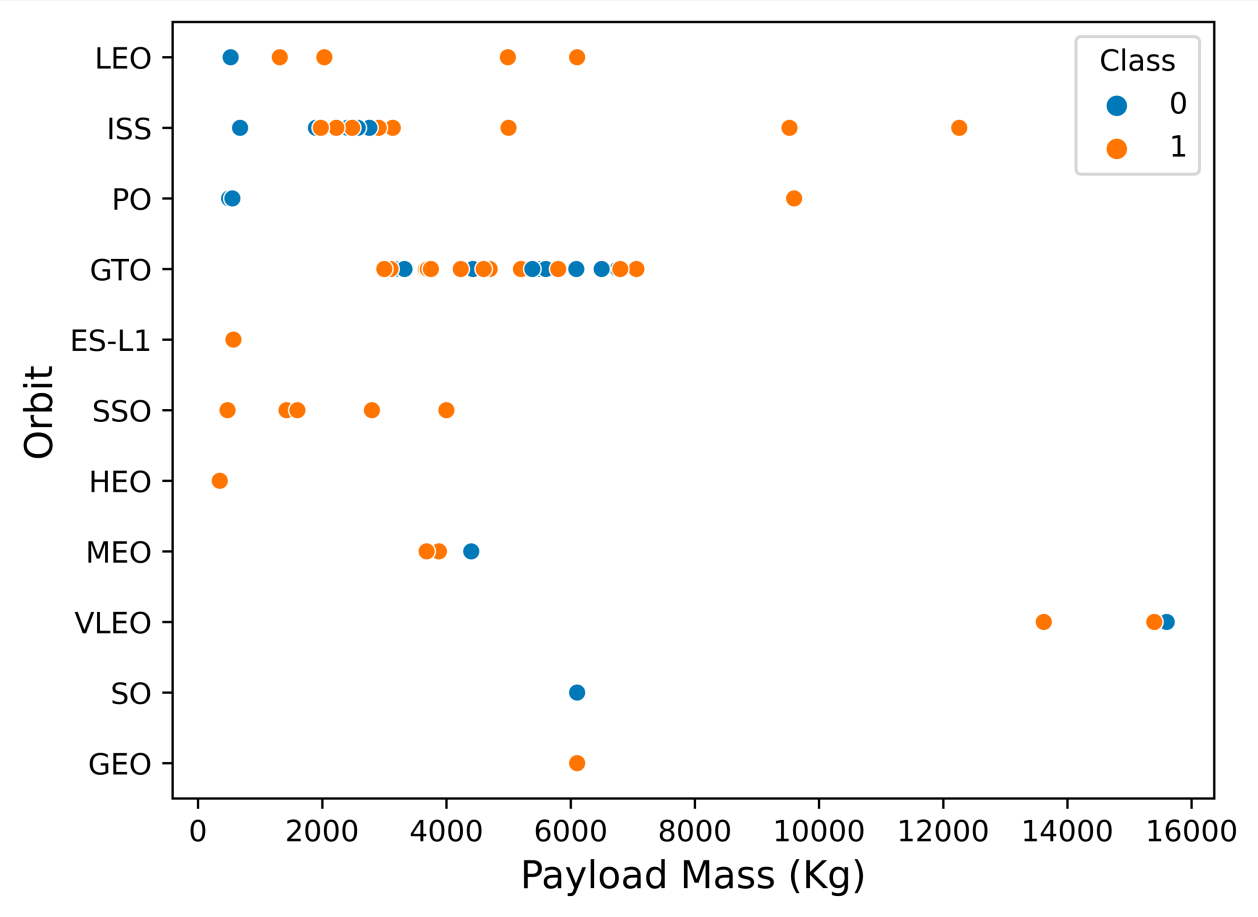
There are some Orbit types with high success rates like ES-L1, GEO and SSO, but the most interesting is the mission with the orbit "SO" and Orbit type "GTO" has a low rate.

Flight Number vs. Orbit Type



We should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

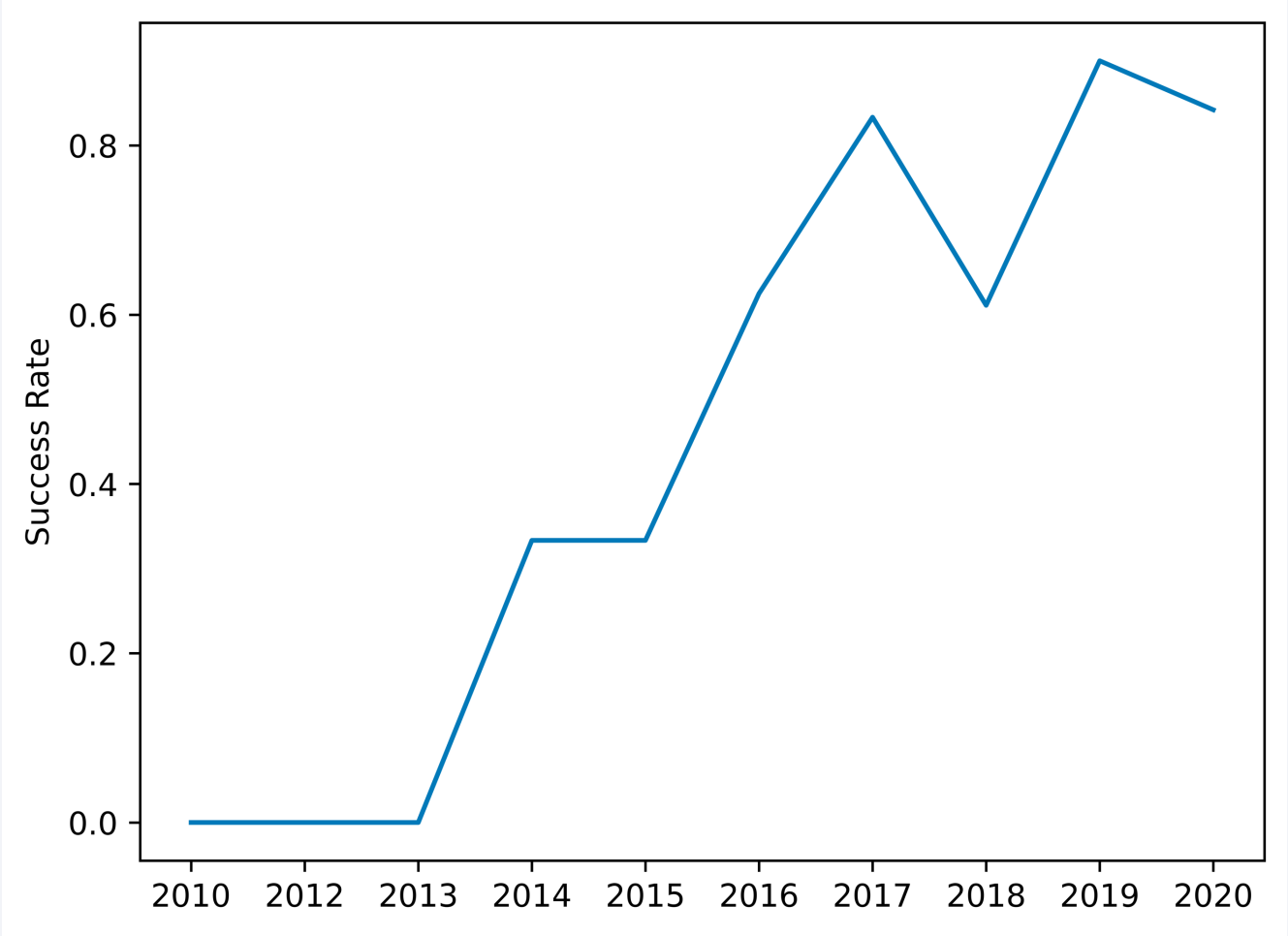
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.

However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



You can observe that the success rate since 2013 kept increasing until 2017 (stable in 2014) and after 2015 it started increasing in success rates bibber than 50%.

All Launch Site Names

```
%%sql

SELECT Launch_Site FROM SPACEXTABLE
GROUP BY Launch_Site;
```

Python

Launch Sites Names

Launch Site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

We select the column Launch_Site from the data frame and group it by its categorical data(Launch site names)

Launch Site Names Begin with 'CCA'

```
%%sql

SELECT * FROM SPACEXTABLE
    WHERE Launch_Site like "CCA%"
    LIMIT 5;
```

Python

Date	Time (UTC)	Booster_Version	Launch_Site	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
6/4/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	0	LEO	SpaceX	Success	Failure (parachute)
12/8/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	525	LEO (ISS)	NASA (COTS)	Success	No attempt
10/8/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	500	LEO (ISS)	NASA (CRS)	Success	No attempt
3/1/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%%sql

SELECT SUM(PAYLOAD_MASS__KG_) as "Total payload kg" FROM SPACEXTABLE
    WHERE "Customer" = "NASA (CRS)";
```

Python

Total payload (Kg)
45596

Select the variable Payload mass, apply the function sum() and specify just for the customer "Nasa (CRS)"

Average Payload Mass by F9 v1.1

```
%%sql

SELECT AVG(PAYLOAD_MASS__KG_) as "Average payload kg" FROM SPACEXTABLE
    WHERE "Booster_Version" like "F9 v1.0%";
```

Python

Average payload (Kg)
340.4

Select the variable Payload mass, apply the function AVG() and specify just for the "Booster_Version" looks like "F9 v1.0"

First Successful Ground Landing Date

```
%%sql

SELECT "date", "landing_outcome" FROM SPACEXTABLE
    WHERE "Landing_outcome" like "%Success%"
    ORDER BY strftime("Date", '%Y-%m-%d') DESC
    LIMIT 1;
```

Python

Date	Landing_Outcome
22/12/2015	Success (ground pad)

Select the variables "date" and "landing_outcome", and specify that "landing_outcome" has the word "Success", order by date and limit to one output.

Successful Drone Ship Landing with Payload between 4000 and 6000

```sql
%%sql

SELECT "Booster_Version", "Landing_Outcome", "PAYLOAD_MASS__KG_" FROM SPACEXTABLE
    WHERE "Landing_Outcome" like "%Success (drone ship)%"
    AND ("PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000);
```

Python

Select the variables "Booster_Version", "landing_outcome", and "Payload_Mass_KG", and then specify that "landing_outcome" has the word "Success (drone ship)" and "Payload_Mass_KG" is between (4000-6000)Kg.

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

```
%%sql

SELECT COUNT(*) as "Total", "Mission_Outcome" FROM SPACEXTABLE
    GROUP BY "Mission_Outcome" like "%Success%";
```

✓ 0.0s Python

Total	Mission_Outcome
1	Failure (in flight)
100	Success

With the function, Count() count each row, select the variable "Mission_Outcome", and then specify that "Mission_Outcome" looks like the word "Success".

Boosters Carried Maximum Payload

```
%%sql

SELECT "Booster_Version", "PAYLOAD_MASS__KG_", "landing_outcome" FROM SPACEXTABLE
    WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTABLE)
    GROUP BY "Booster_Version";
```

✓ 0.0s Python

Select "Booster_Version", "Payload_Mass", and "Landing_Outcomes", and then with a subquery calculate the maximum payload mass, and then group by "Booster_Version".

Boosters Carried Maximum Payload

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

2015 Launch Records

```
%%sql

SELECT "Date", "Landing_Outcome", "Booster_Version", "Launch_site" FROM SPACEXTABLE
    WHERE ("Landing_Outcome" LIKE "%Failure (drone ship)%")
    AND "Date" like "%2015%";
```

Python

Select the variables "Date", "Landing_Outcome", "Booster_Version" and "Launch_Site", and then specify that "Landing_Outcome" has the phrase "Failure (dron ship)", and the date has the word "2015".

Date	Landing_Outcome	Booster_Version	Launch_Site
1/10/2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
14/04/2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

```
%%sql

SELECT "Date","Landing_Outcome" FROM SPACEXTABLE
WHERE "DATE" LIKE "%2010%" OR "DATE" LIKE "%2011%" OR "DATE" LIKE "%2012%" OR "DATE" LIKE "%2013%"
    OR "DATE" LIKE "%2014%" OR "DATE" LIKE "%2015%" OR "DATE" LIKE "%2016%" OR "DATE" LIKE "%2017%"
;
```

Python

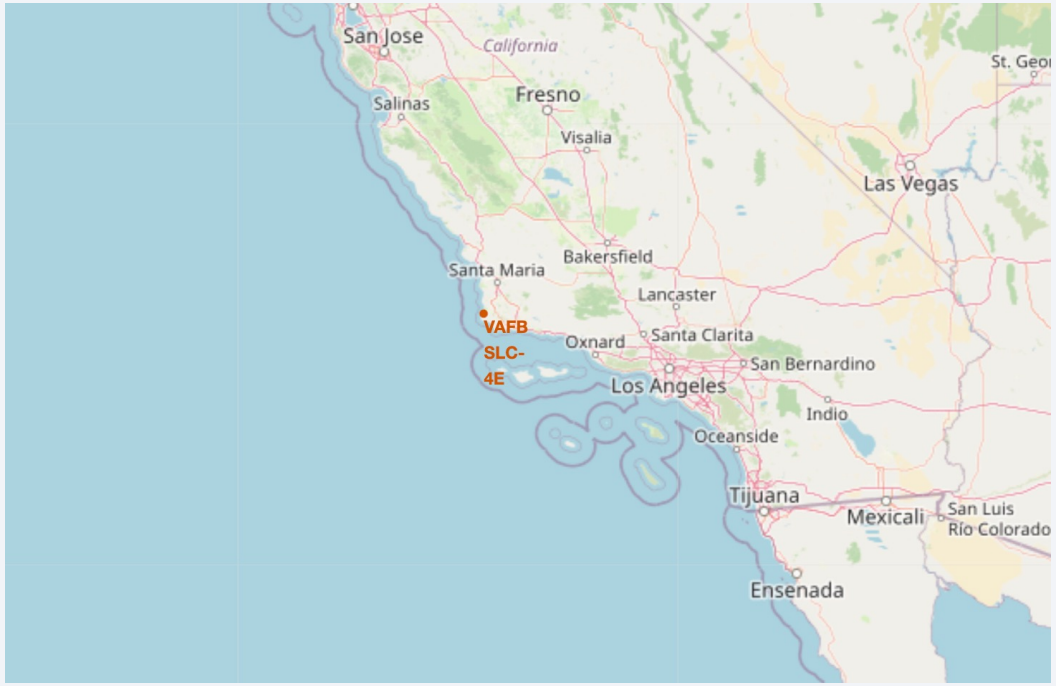
Select the variables "date" and "Landing_Outcome" and sort them year by year because "sql lite" in Python doesn't support date() functions.

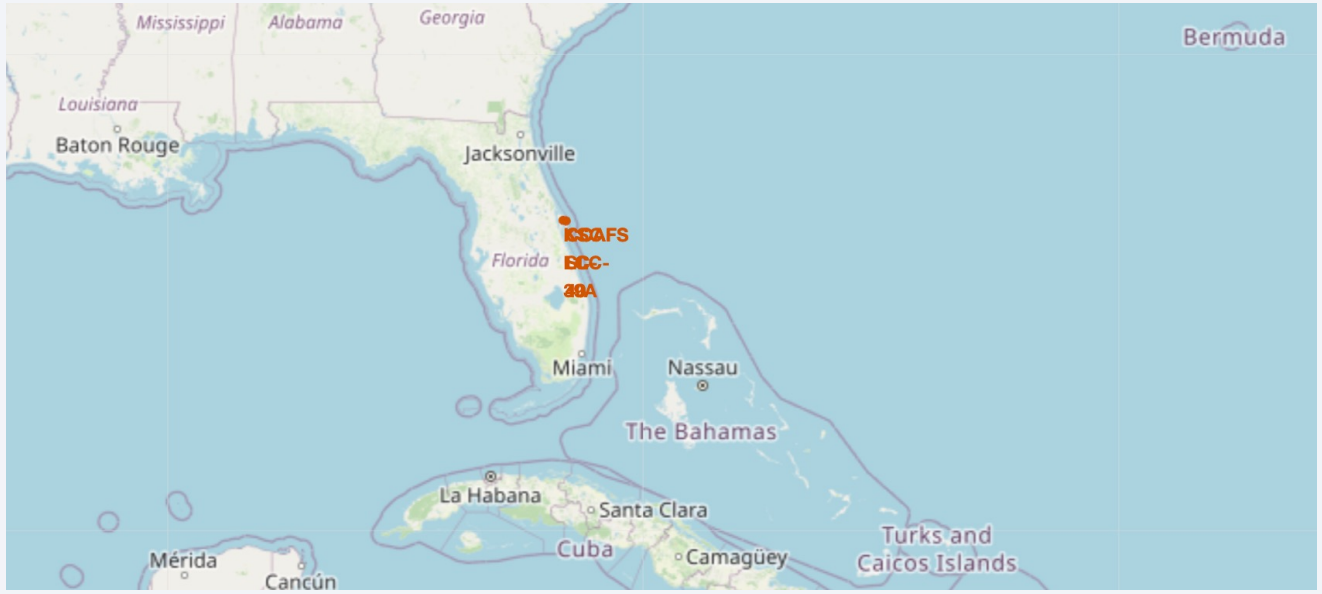
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

Date	Landing_Outcome	Date	Landing_Outcome	Date	Landing_Outcome
6/4/2010	Failure (parachute)	14/04/2015	Failure (drone ship)	30/03/2017	Success (drone ship)
12/8/2010	Failure (parachute)	27/04/2015	No attempt	5/1/2017	Success (ground pad)
22/05/2012	No attempt	28/06/2015	Precluded (drone ship)	15/05/2017	No attempt
10/8/2012	No attempt	22/12/2015	Success (ground pad)	6/3/2017	Success (ground pad)
3/1/2013	No attempt	17/01/2016	Failure (drone ship)	23/06/2017	Success (drone ship)
29/09/2013	Uncontrolled (ocean)	3/4/2016	Failure (drone ship)	25/06/2017	Success (drone ship)
12/3/2013	No attempt	4/8/2016	Success (drone ship)	7/5/2017	No attempt
1/6/2014	No attempt	5/6/2016	Success (drone ship)	14/08/2017	Success (ground pad)
18/04/2014	Controlled (ocean)	27/05/2016	Success (drone ship)	24/08/2017	Success (drone ship)
14/07/2014	Controlled (ocean)	15/06/2016	Failure (drone ship)	9/7/2017	Success (ground pad)
8/5/2014	No attempt	18/07/2016	Success (ground pad)	10/9/2017	Success (drone ship)
9/7/2014	No attempt	14/08/2016	Success (drone ship)	10/11/2017	Success (drone ship)
21/09/2014	Uncontrolled (ocean)	14/01/2017	Success (drone ship)	30/10/2017	Success (drone ship)
1/10/2015	Failure (drone ship)	19/02/2017	Success (ground pad)	15/12/2017	Success (ground pad)
2/11/2015	Controlled (ocean)	16/03/2017	No attempt	23/12/2017	Controlled (ocean)
3/2/2015	No attempt				


Section 3
Launch Sites
Proximities Analysis

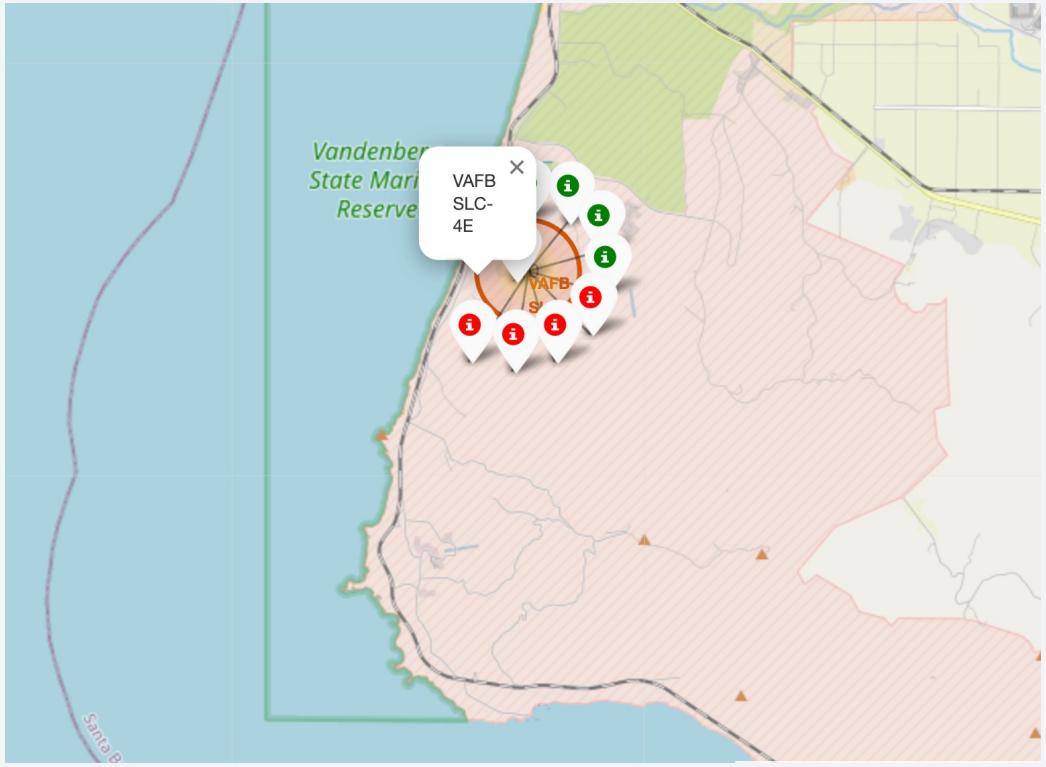
Launch site coordinates

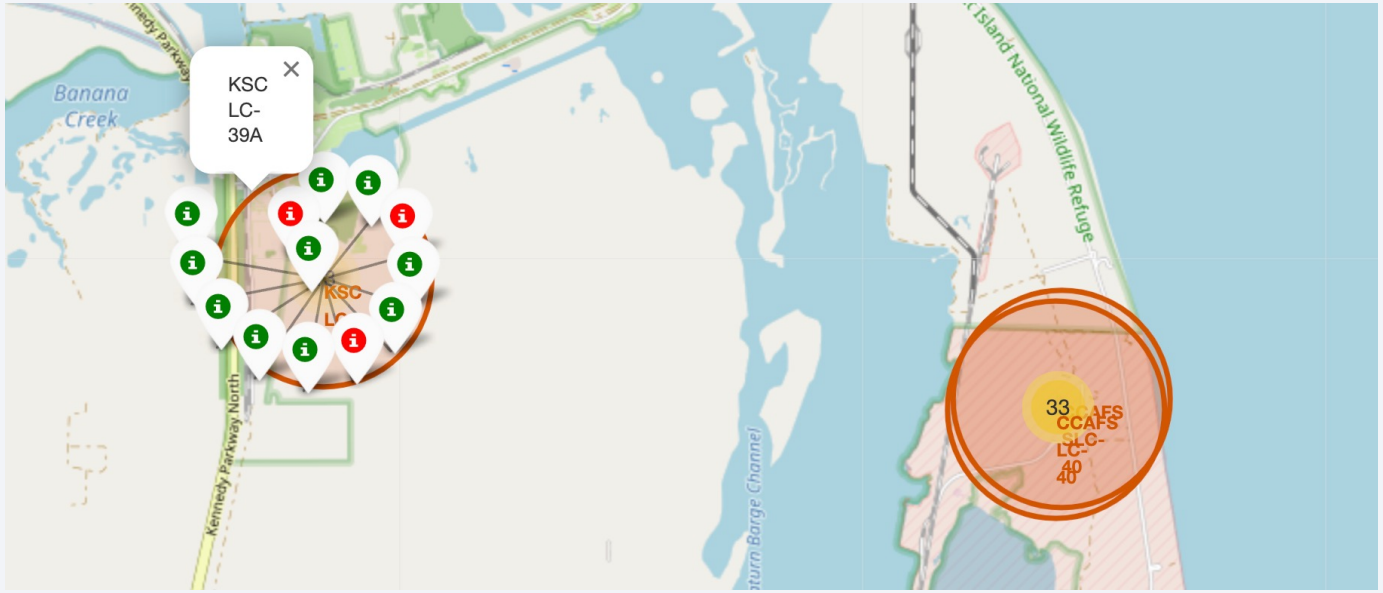




The launches are in proximity to the tropic cancer lines, and the launch sites are considerably close to the coast areas.

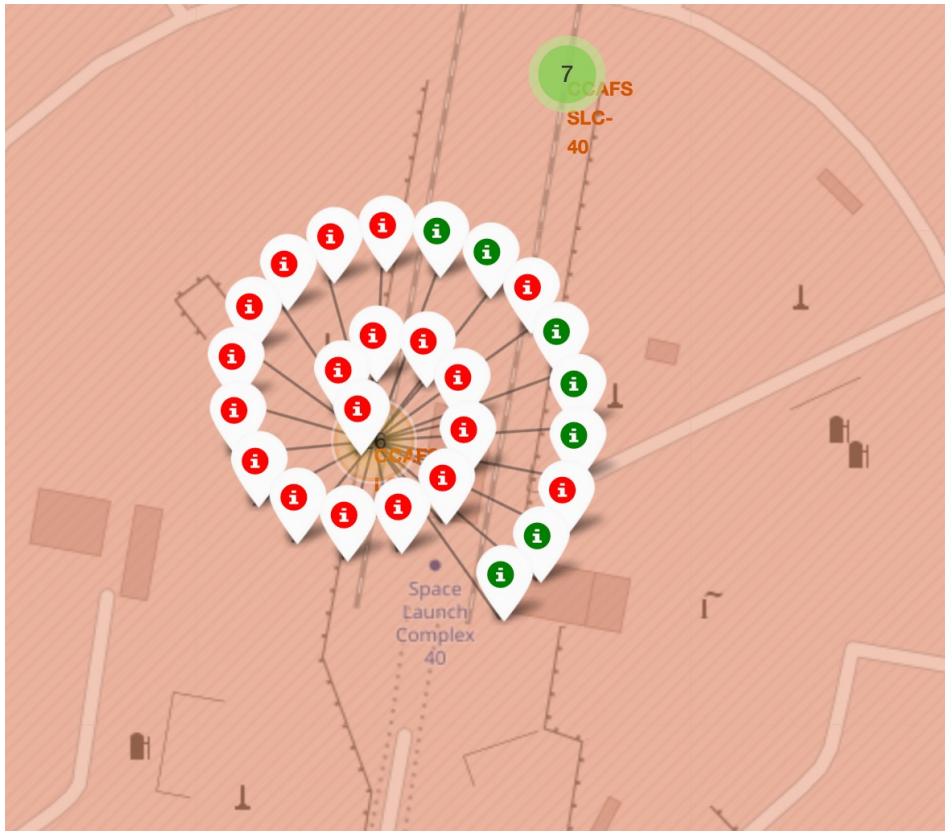
Graphic Success rate in each Launch site

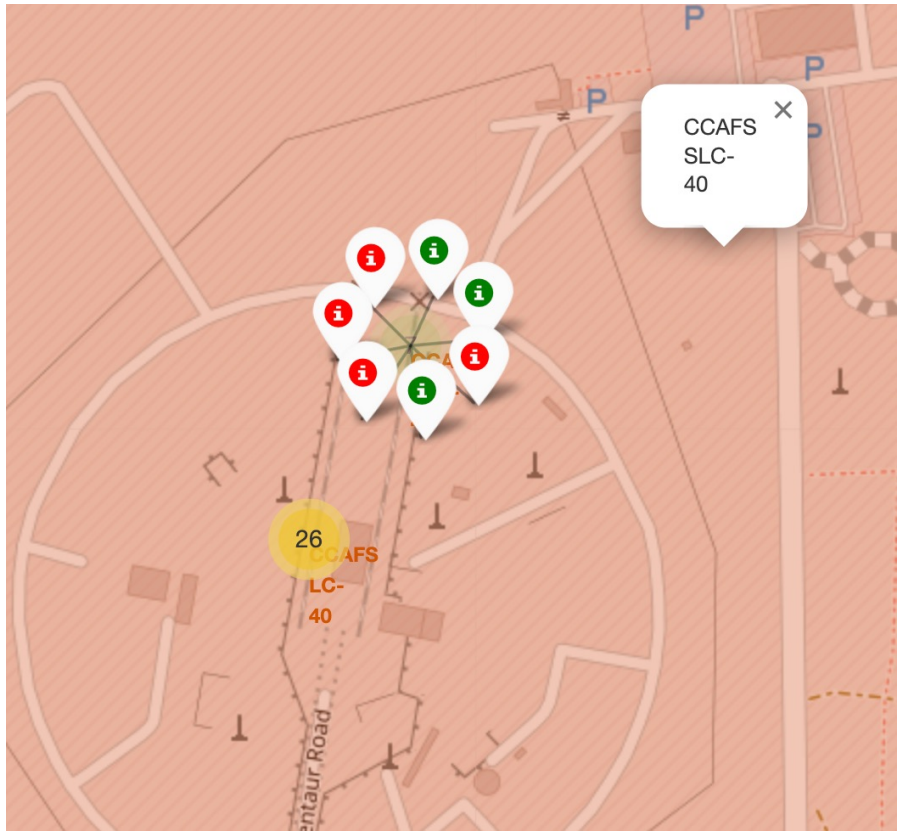




Highest success rate is the launch site KSC LC-39A

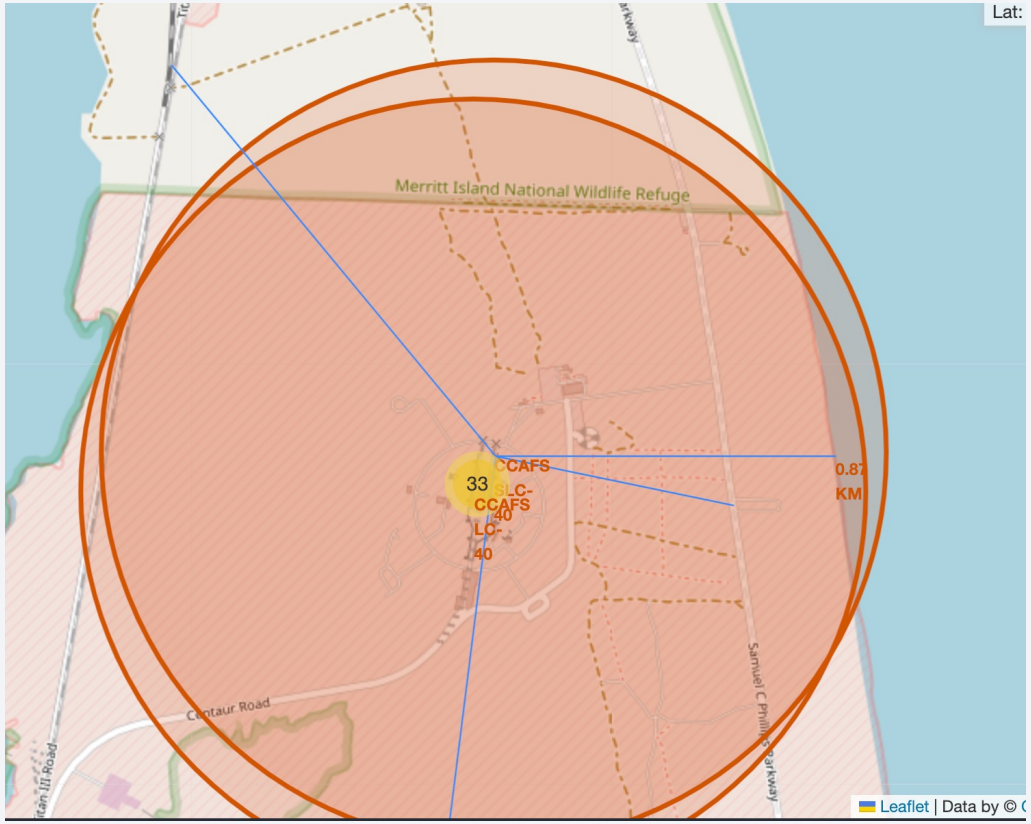
Graphic Success rate in each Launch site

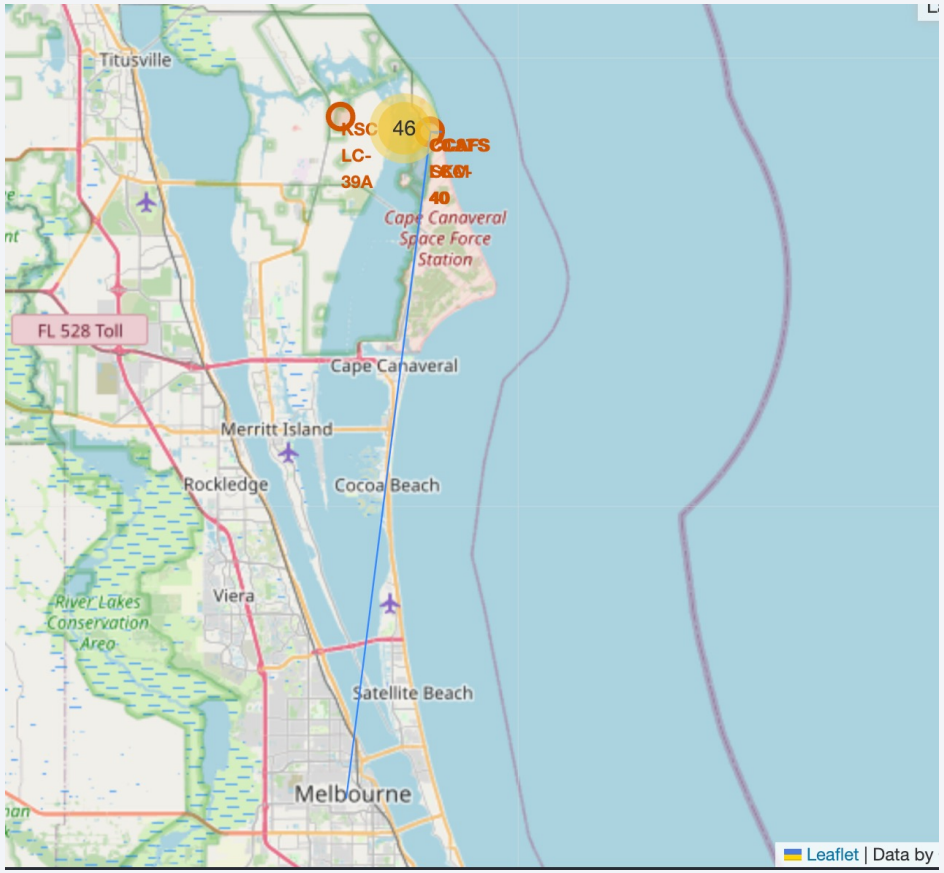




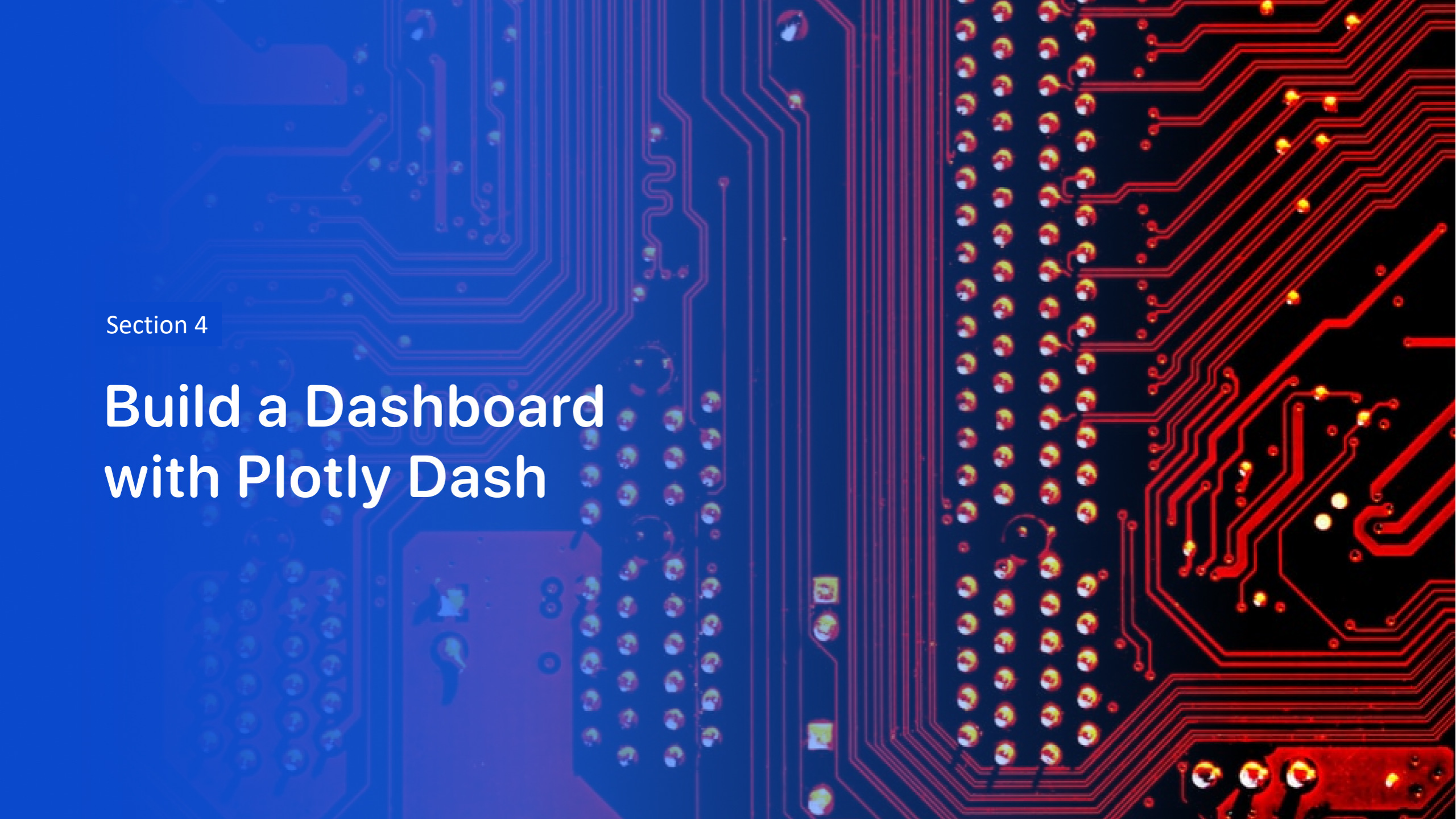
Lowest success rate is the launch site CCAFS LC-40.

Around the Launch sites

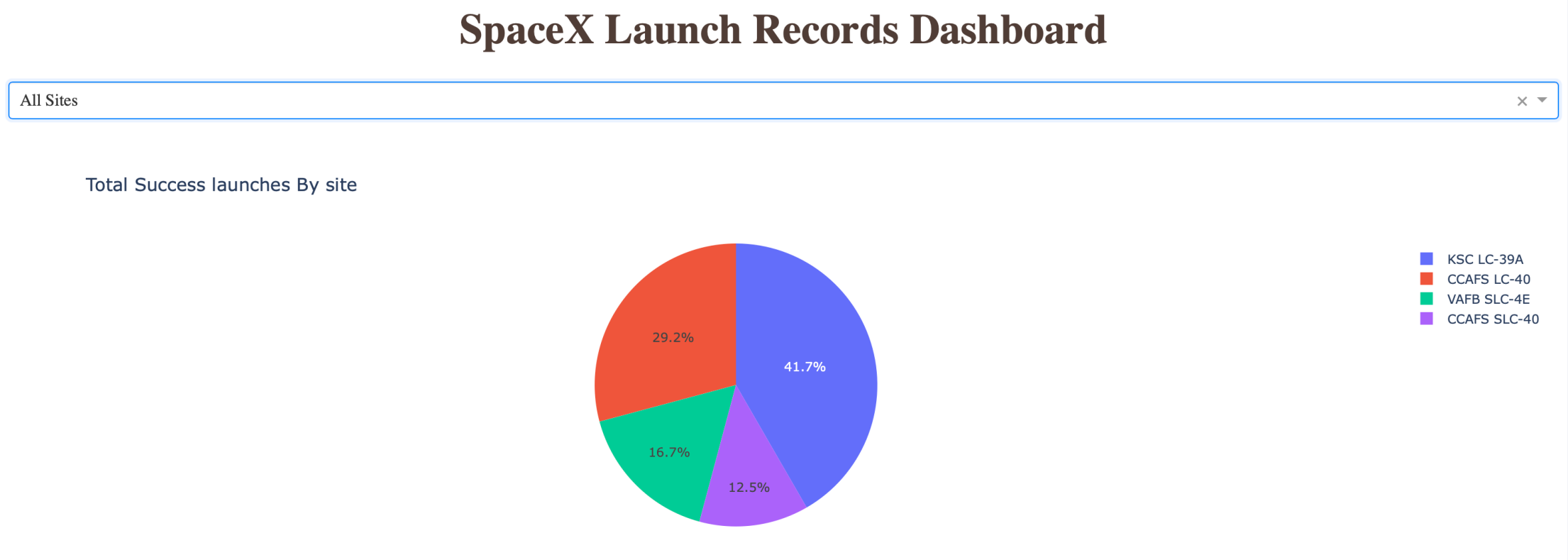




The launch sites are close to the railways, highways and coastlines, but they keep a certain distance away from cities.

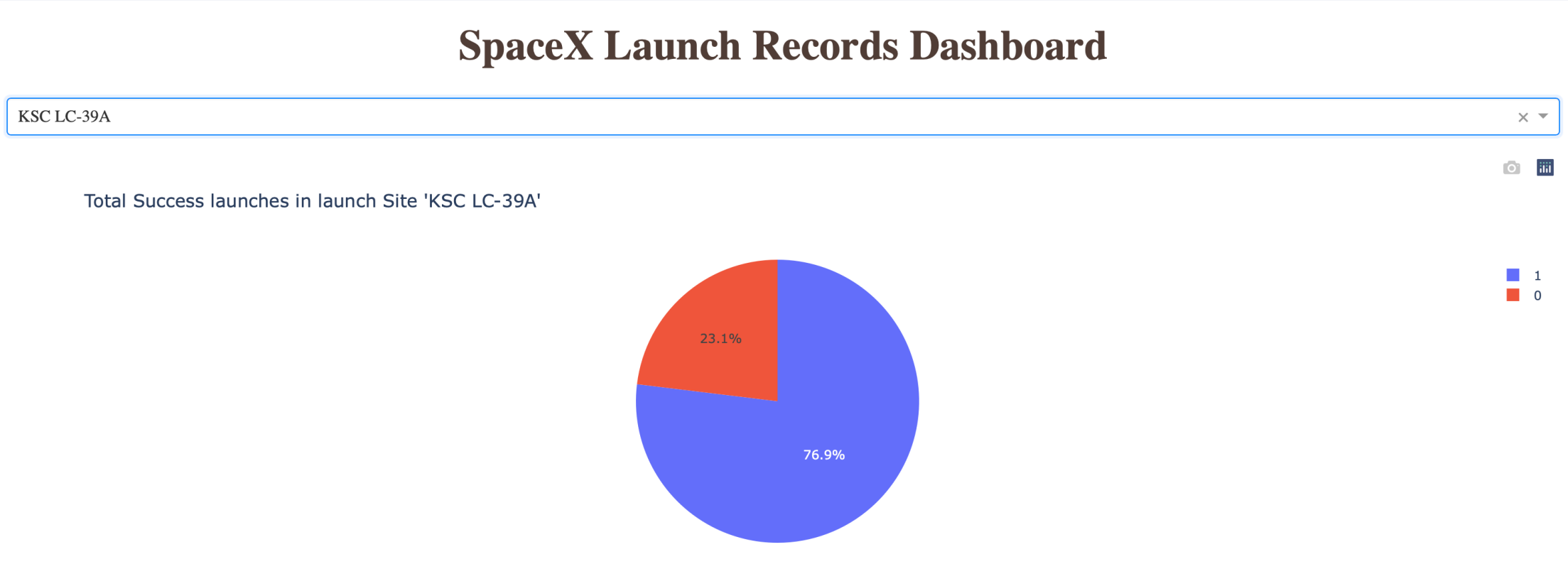

Section 4
Build a Dashboard with Plotly Dash

All sites success rate



We can see the Total Success launches By Site and the highest success rate is for the launch site "KSC LC-39A"

The best Launch Site

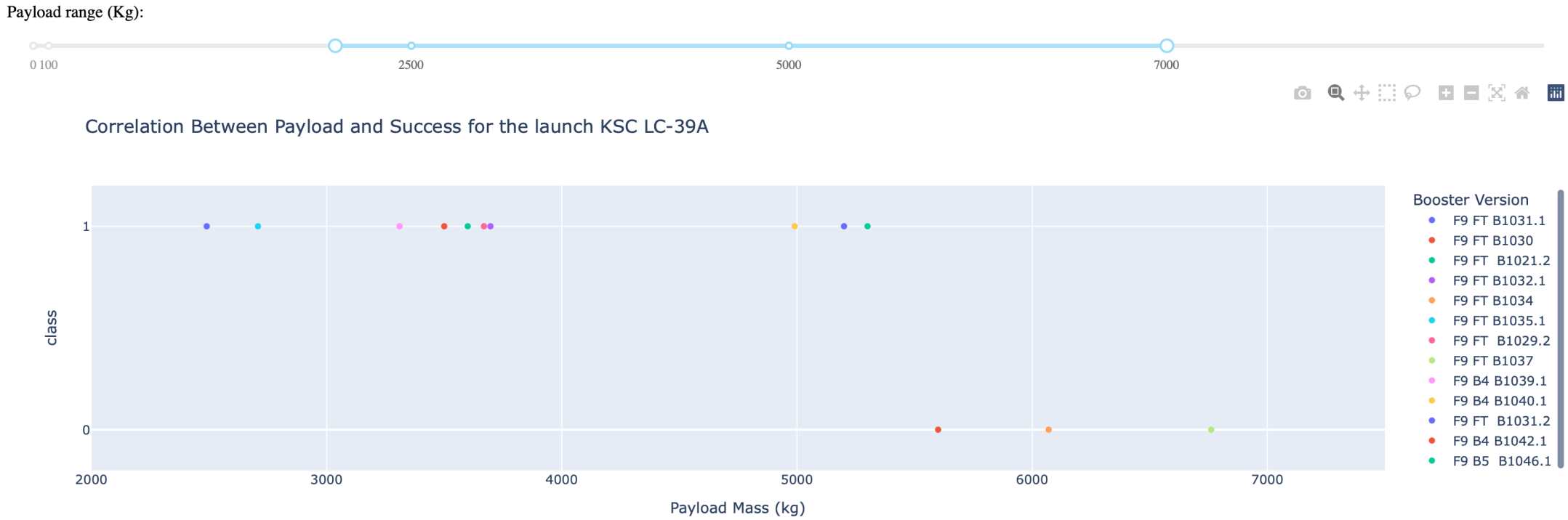


We are able to see that the launch "KSC LC-39A" has a 76.9% success rate in its launches.

Payload mass and success rate correlation

As we can see, there is no clear correlation between Payload mass and success rate when we see all the launch sites together.

Payload mass and success rate correlation

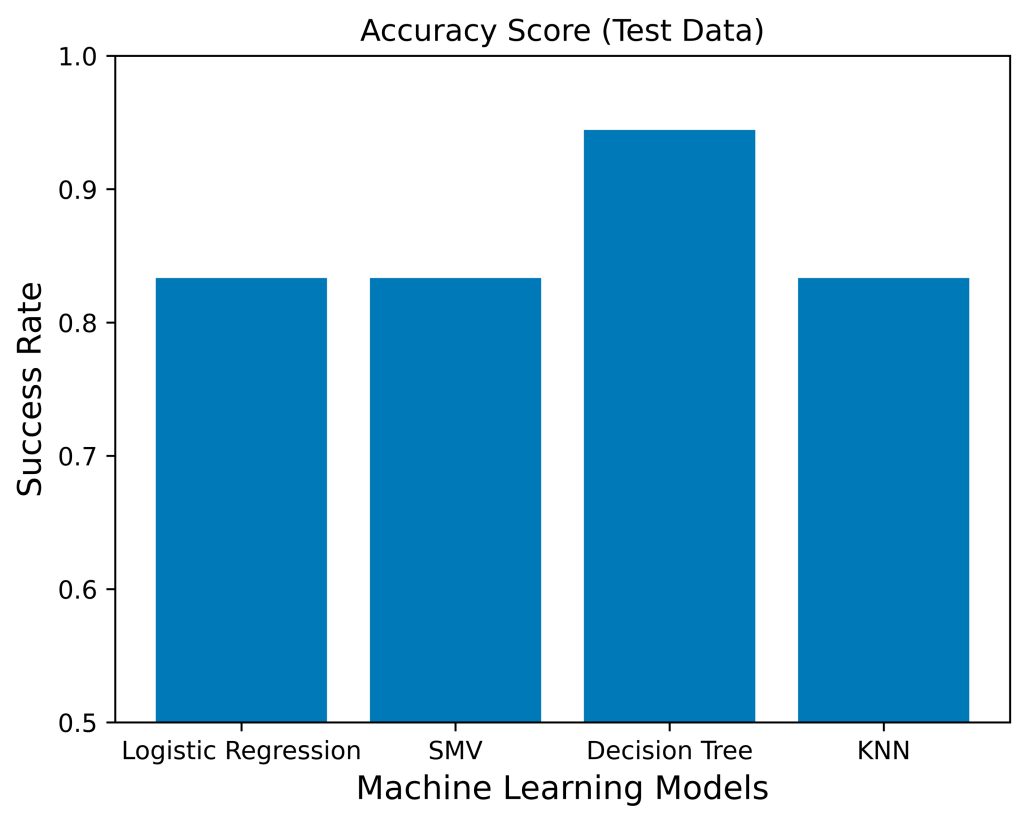


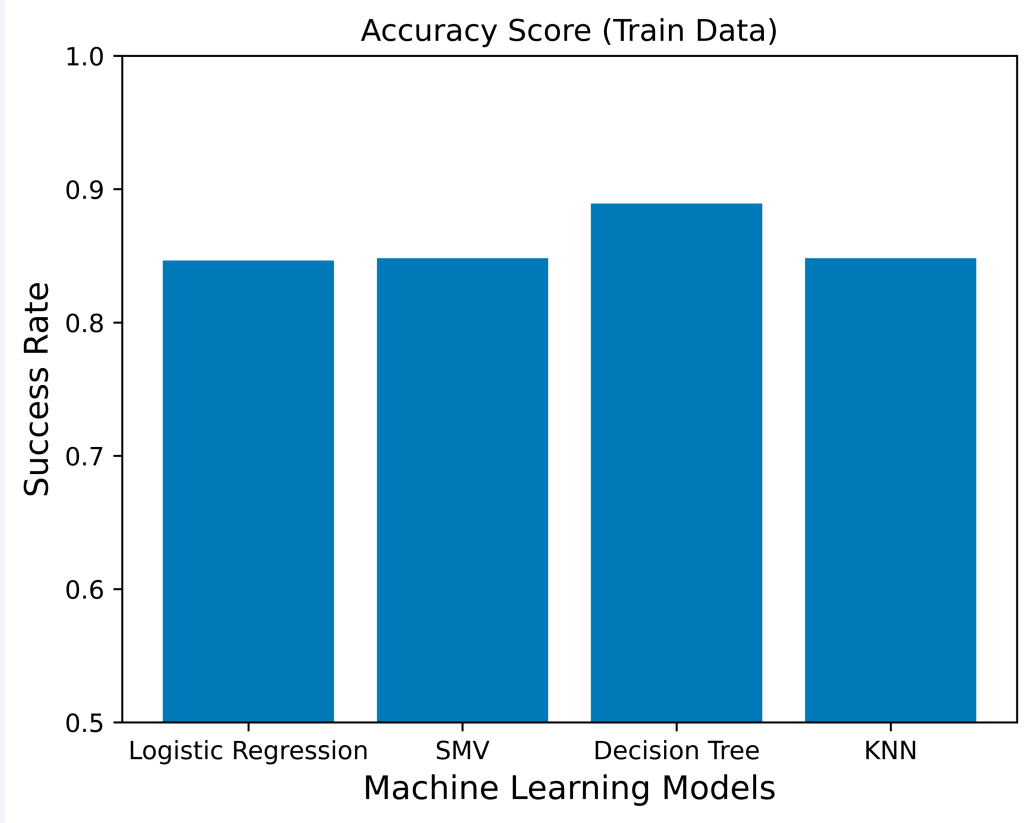
Now if we see just the launch site with the best success rate, we can see that values of payload mas bigger than 5500Kg the launches were unsuccessful independent of the Booster version.

Section 5

Predictive Analysis (Classification)

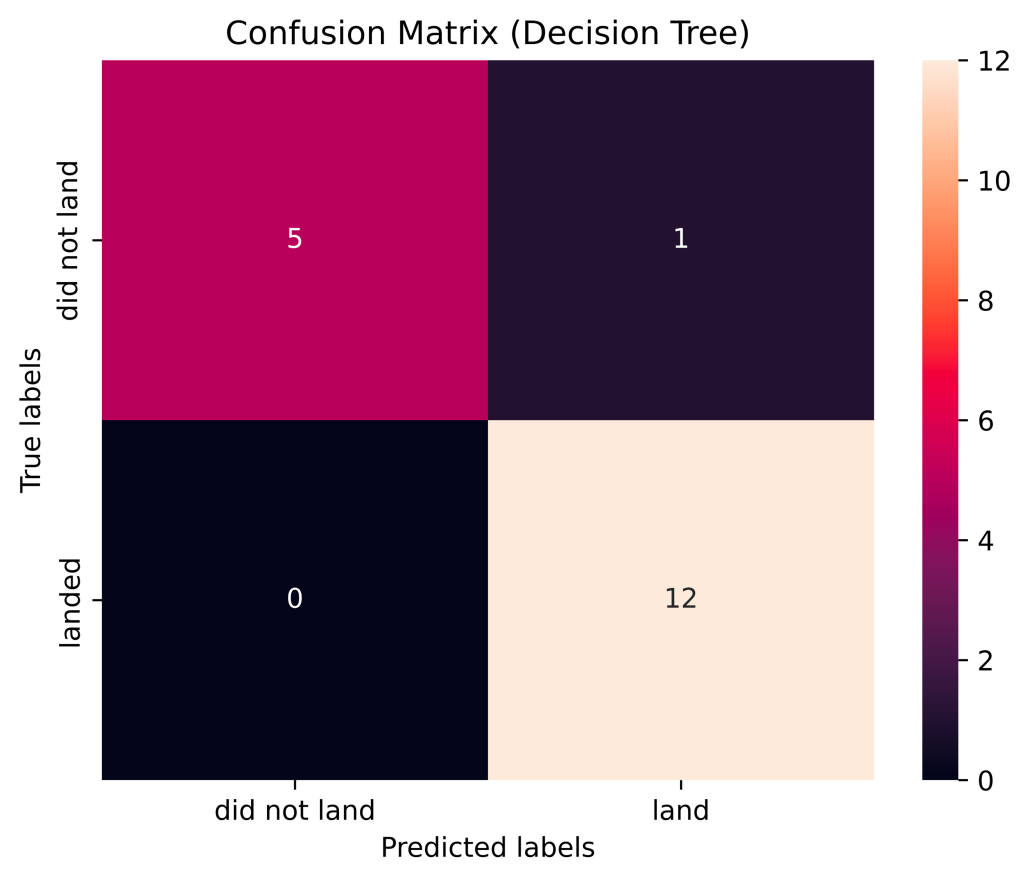
Classification Accuracy

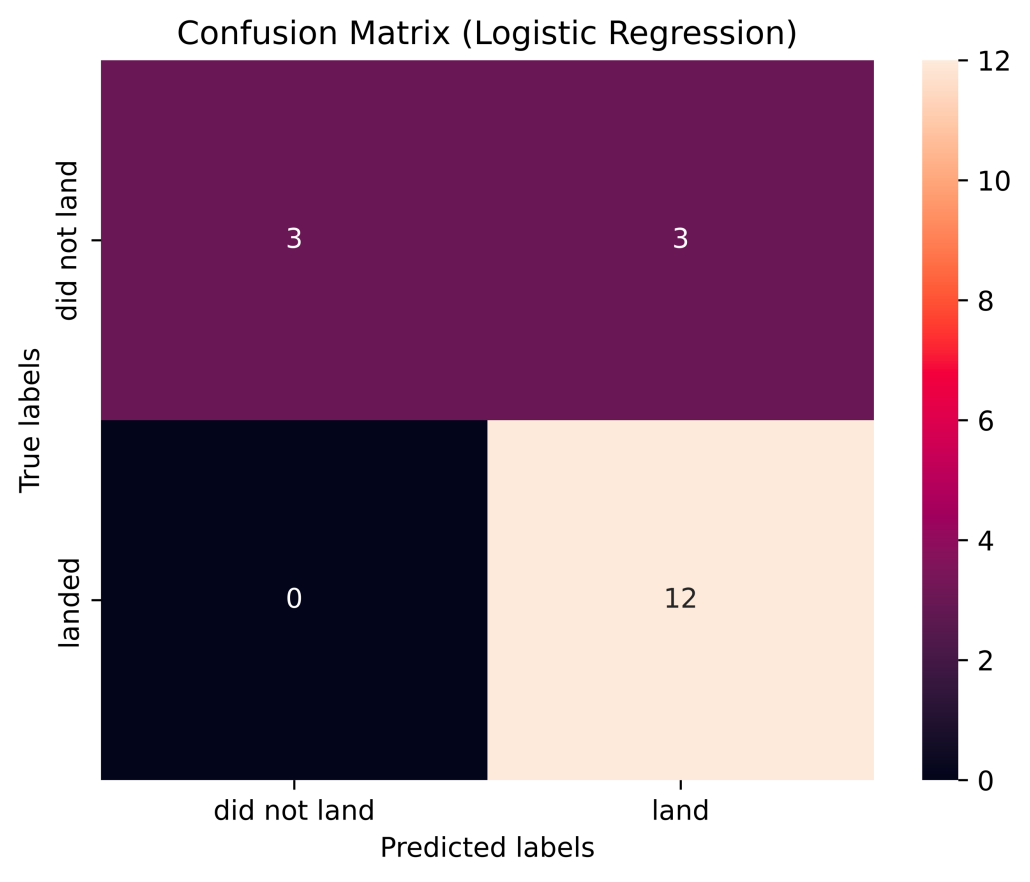




We can see that the best model with the best score is the model Decision Tree.

Confusion Matrix





The Confusion Matrix for the Decision Tree and the Logistic Regression has no fake negative. Still, the Decision Tree model presents an improvement with respect to the fake positives in this study.

Conclusions

- The best launch site is "KSC LC 39A" with a success rate of 76.9%
- The best orbit types are GEO, HEO and SSO
- 2013 started the successful missions but 2015 was the first landing success outcome on the ground pad.
- Falcon payload mass capacity is around 340.4 Kg
- 99% of all missions were successful
- Max carried payload was 15600Kg with a success rate of 83.%
- All the launch sites are close to the coastlines but within a certain distance of the cities.
- All machine learning models can classify successful launches but with problems with Fake positives.
- The best machine learning model to classify success and unsuccess launches is the Decision Tree model with a 94% accuracy score.

Thank you!