Outline

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

Executive Summary

- We have explored data available about Space X´s launches to build machine learning models that predict if Space X plan to reuse the First Stage on the next launches.
- Predicting Space X plans is key to Space Y, as the reuse of the First Stage allows a launch at a cost of USD 62 million, whilst not reusing it the cost remains around USD 165 million.
- Data indicates that:
    - Success rate on landing Falcon 9 first stage has significantly increased over the years
    - Most of the successful landing and reuse of FS have occurred on KSC 39A site
    - About 75% of CCAFS LC40 launches did not end with FS being landed
    - Booster version v 1.1 in general did not land to be reused
- Logistic Regression, Decision Tree, SVM and KNN methods had a similar performance with an accuracy of 83.3%

Introduction

- Space Y wishes to enter on the Space Race that is dominated by Space X.
- One of the main Space X's competitive advantages is their capacity to land and reuse the First Stage of their rockets. This technology allows Space X to operate at a USD 62 million cost per launch as compared to USD 165 million all other companies' cost.
- For different reasons, Space X not always reuse their Falcon 9 First Stage and we wish to be able to predict when this will happen, using data available about the several Space X launches.
- The proposed approach is to build machine learning models, based on public information, that predicts when Space X will not reuse the First Stage allowing Space Y to define its pricing strategy for the next bids.
- On this project we tested KNN, Decision Tree, SVM and Logistic Regression models.
- This presentation summarizes the methodology, results and insights obtained from an exploration of data provided by Space X and the support provided by IBM Coursera team.

Methodology

Executive Summary

- Data collection methodology:
- Perform data wrangling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- SpaceX launch data is available at url= https://api.spacexdata.com/v4/launches/past
- Data was obtained through SpaceX REST API, using the Python´s requests library:

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

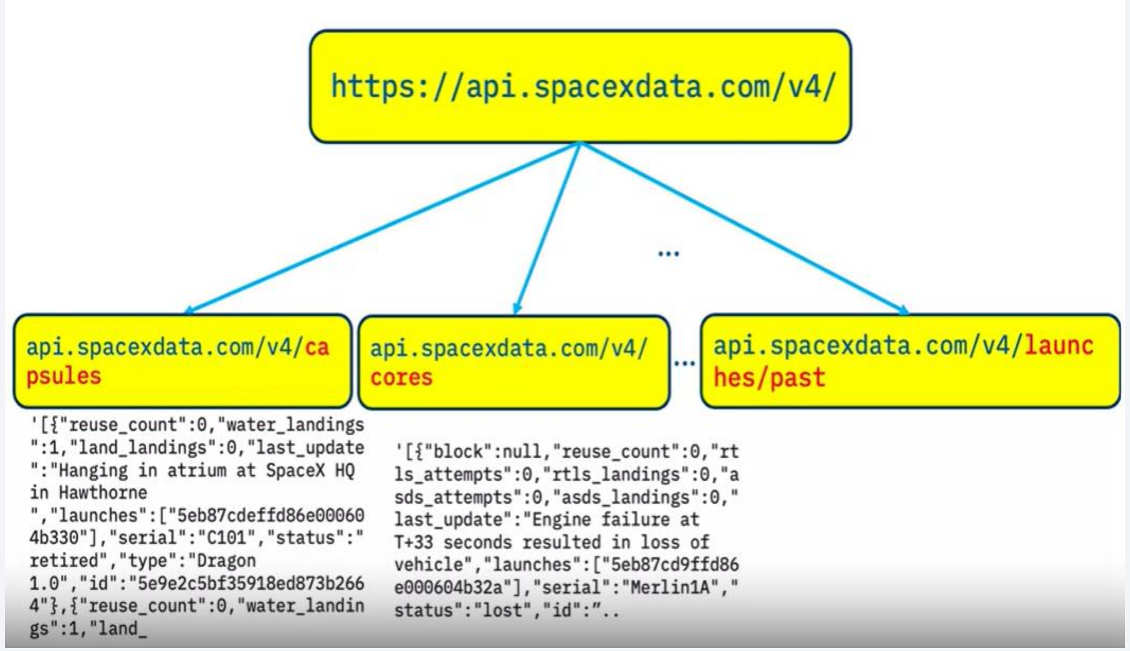
- The SpaceX REST API returned JSON objects that were converted into tables using:

```
data= pd.json_normalize(response.json())
```

- Additional info was collected by web scraping related Wiki pages, using the BeautifulSoup package
- The complex work was done by IBM Data Science team that compiled launches' data containing rocket used, payload delivered, launch specifications, landing specifications, and landing outcome and saved on a Lab Test directory.

Data Collection – SpaceX API

Space X files were organized as below:



To have a consolidated table the following steps were observed:

1. Booster name obtained from Rocket column
2. Launch site, long. and lat. obtained from Launchpad
3. Payload mass and orbit obtained from Payload
4. Outcome, Type of landing number of flights obtained from Cores
5. For the course all the info above was saved on a static objects and in : 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/**API_call_spacex_api.json'**

[IBM-Data-Science-Capstone-Project/jupyter-labs-spacex-data-collection-api.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)]

Data Collection - Scraping

To extract the info from the Falcon 9 Wiki page, the following steps were observed:

1. Used *requests.get(url).text*, to store the content of the webpage into a text object ('data')
2. With `BeautifulSoup` parsed the text from 'data' into a structured `BeautifulSoup` object ('soup')
3. With html identifiers in soup, found the tables and in the tables the columns names and respective contents
4. Columns contents were first stored in a dictionary type and then converted into a pandas dataframe type.
5. The dataframe could be then saved as a .csv file
6. For the course all the html page was sourced from a static

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

IBM-Data-Science-Capstone-Project/jupyter-labs-webscraping.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)

Data Wrangling

Data were processed from: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv

The discovery and preparation of the data observed the following steps:

1. Identify and calculate the percentage of the missing values in each attribute using: df.isnull().sum()/len(df)*100
2. Identify which columns are numerical and categorical, using: df.dtypes
3. Calculate the number of launches on each site, using: df.value_counts('LaunchSite')
4. Calculate the number and occurrence of each orbit, using: df.value_counts('Orbit')
5. Determine the number of landing_outcomes: using df.value_counts('Outcome')
6. Create a list with (0) if the landing outcome was failed or (1) if the landing outcome was successful, using: ~df['Outcome'].isin(bad_outcomes)

IBM-Data-Science-Capstone-Project/labs-jupyter-spacex-Data wrangling.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)

EDA with SQL

The following queries were made on the file: https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv:

1. Names of the unique launch sites in the space mission
2. 5 records where launch sites begin with the string 'CCA'
3. The total payload mass carried by boosters launched by NASA
4. Average payload mass carried by booster version F9 v1.1
5. Date when the first successful landing outcome in ground pad was achieved]
6. The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

EDA with SQL (cont.)

7. List the total number of successful and failure mission outcomes

8. Names of the booster versions which have carried the maximum payload mass.

9. List the records with month names, failure landing outcomes in drone ship, booster versions, launch site for the months in year 2015.

10. Rank of the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20)

IBM-Data-Science-Capstone-Project/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)

EDA with Data Visualization

Charts generated at this phase:

1. Scatter with FlightNumber x Payload
2. FlightNumber vs LaunchSite
3. launch sites and their payload mass
4. relationship between success rate and orbit type.
5. relationship between FlightNumber and Orbit type
6. relationship between Payload and Orbit type
7. to get the average launch success trend.

This phase aimed to obtain some preliminary insights about how each important variable would affect the success rate,

IBM-Data-Science-Capstone-Project/edadataviz.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)

Build an Interactive Map with Folium

- Map objects added to a folium map with coordinates close to the launch sites:
    - Markers, Circles, Lines, Text, Pop up text, markers clusters
- Those objects were added using the following commands:
    - circle = folium.Circle(long_lat_coord, radius=1000, color='#d35400', fill=True).add_child(folium.Popup(row['Launch Site']))
    - marker = folium.map.Marker(Coordinates, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0),
    - html='<div style="'])
        - icon=folium.Icon(color='white', icon_color=row['marker_color']
    - lines=folium.PolyLine(locations=coordinates, weight=1)

    - map.add_child(circle)
    - map.add_child(marker)
    - map.add_child(lines)

[IBM-Data-Science-Capstone-Project/lab_jupyter_launch_site_location (1).ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)]

Build a Dashboard with Plotly Dash

- Pie Charts and Scatter plots about success rate on first stage landing were added to a dashboard.
- Options were provided to look for the overall launches on all launch sites or to select the details of launches in each one of the sites.
- Plots were added to seek for more details on whether first stage successful landings could be different with payload and booster version

[IBM-Data-Science-Capstone-Project/spacex_dash_app (3).py at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)]

Predictive Analysis (Classification)

- Data was extracted from: 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_3.csv, normalized and transformed (X = preprocessing.StandardScaler().fit(X).transform(X))
- Data was split betwee train and test sets.
- models were trained and hyperparameters were selected using GridSearchCV, to find the best estimator parameters
- The accuracy on the test data was calculated using the method score
- Results were visualized with confusion matrix

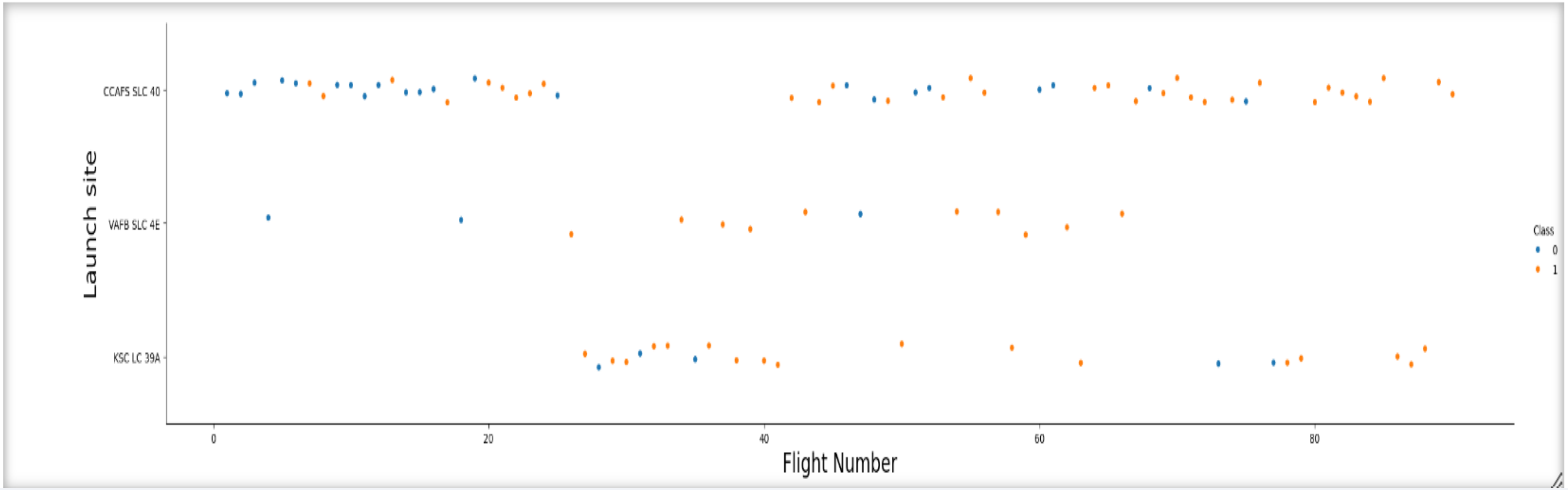
IBM-Data-Science-Capstone-Project/SpaceX_Machine Learning Prediction_Part_5.ipynb at main · hsgeral/IBM-Data-Science-Capstone-Project (github.com)

Results

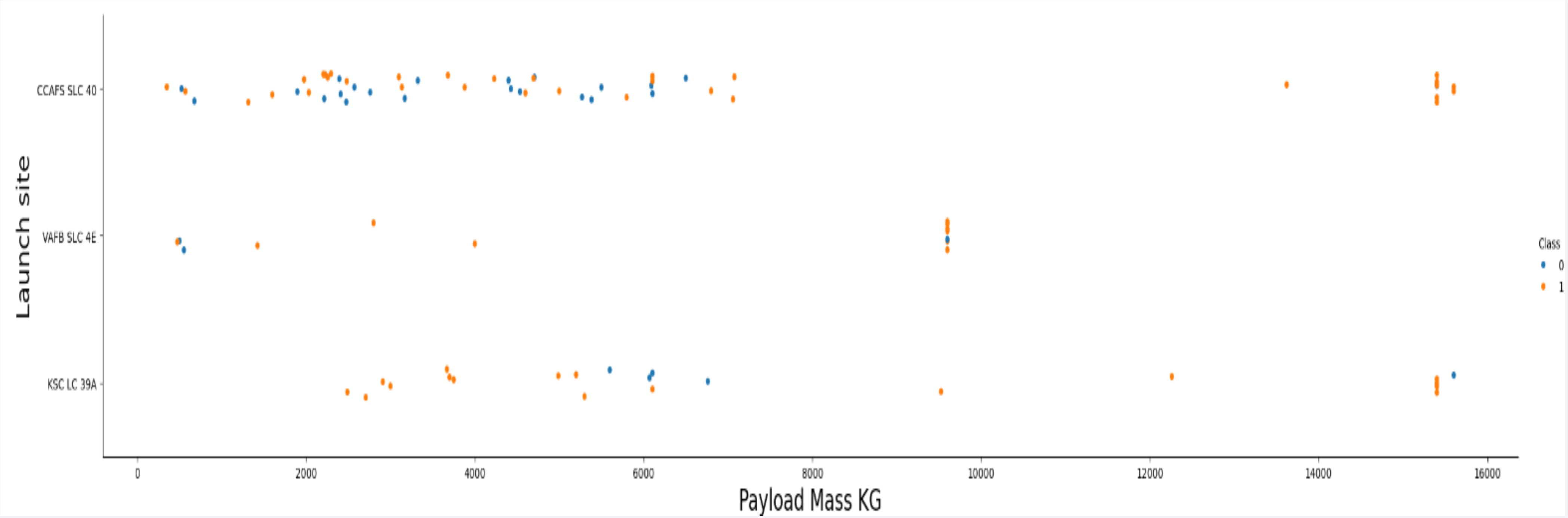
In exploring the dataset, we noticed some interesting points:

- Success rate on landing Falcon 9 first stage has significantly increased over the years
- Different launch sites have different success rates:
  - Most of the successful landing and reuse of FS have occurred on KSC 39A site
  - About 75% of CCAFS LC40 launches did not end with FS being landed
- Booster version v 1.1 in general did not land to be reused
- Majority of payloads seems to be in the range of 1000 to 8000KG
- Most common Orbits (GTO, ISS, LEO, PO) have a success landing rate <100% others have a 100% rate
- As there are several parameters, using a machine learning capable of processing all data seems much more effective than trying to analyzing them manually.

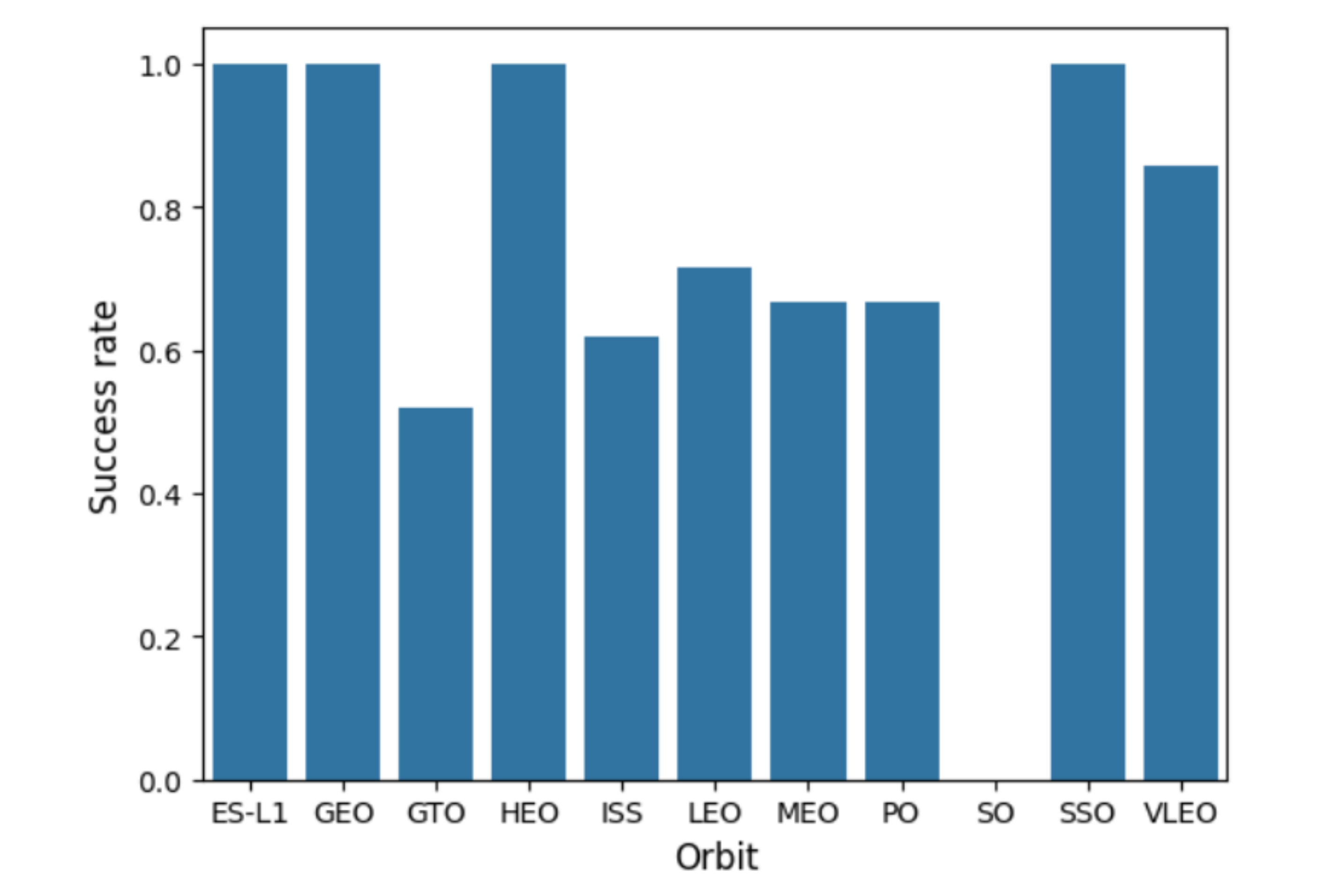
Flight Number vs. Launch Site



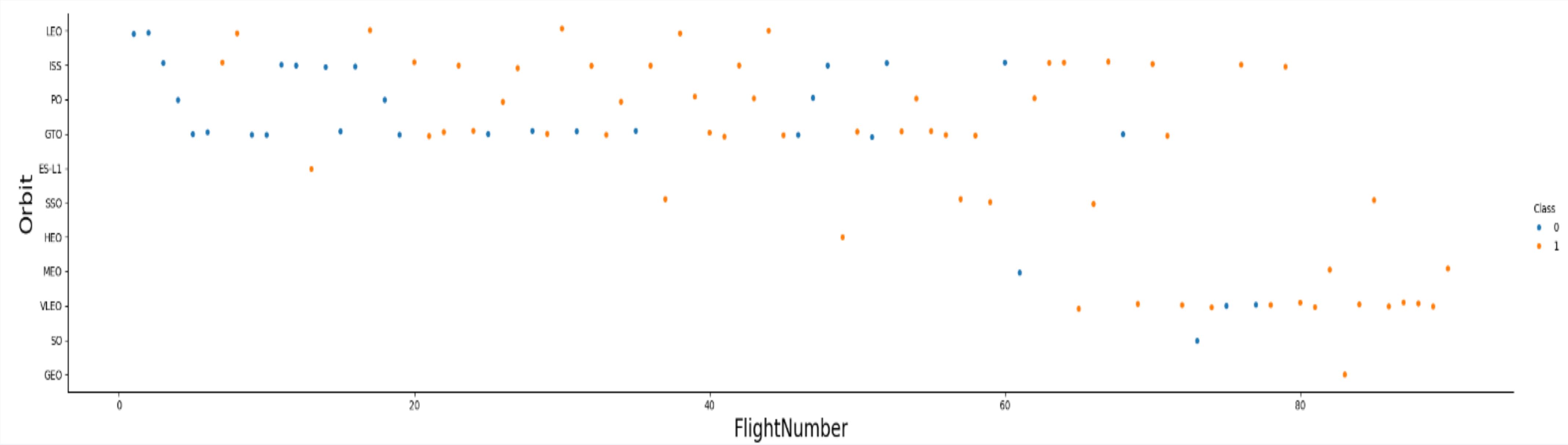
Payload vs. Launch Site



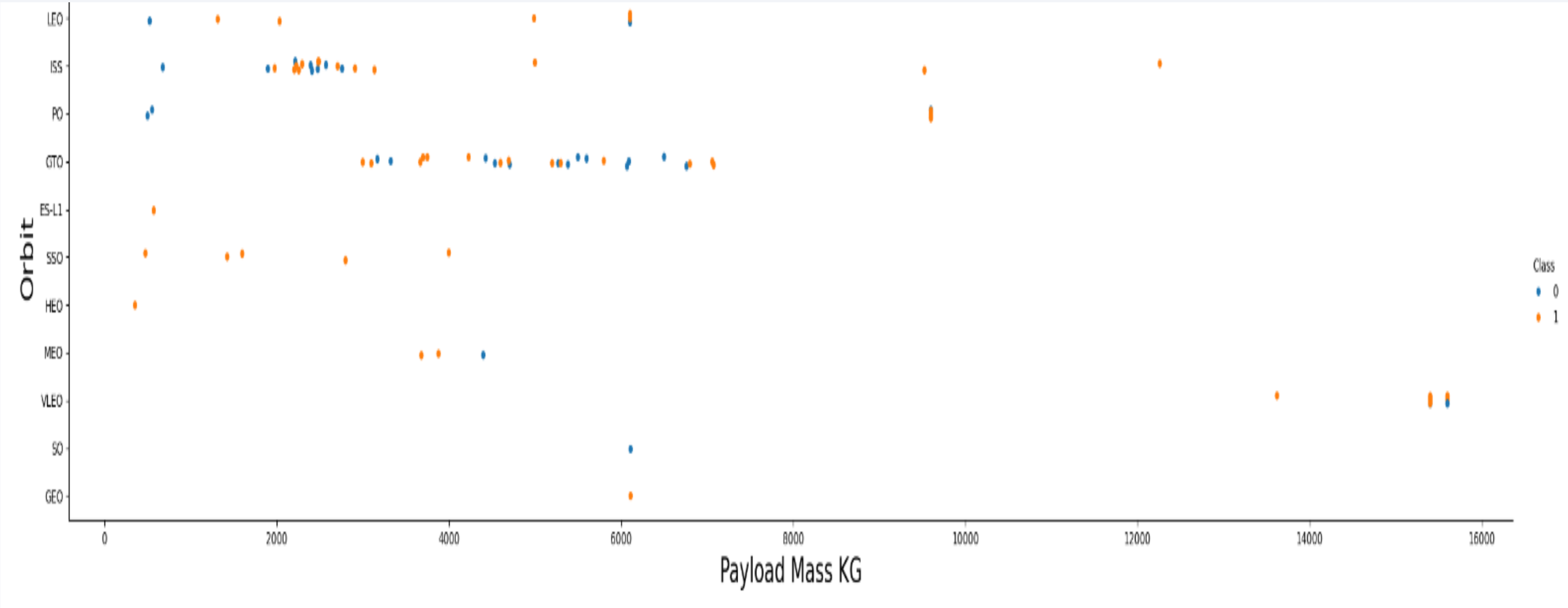
Success Rate vs. Orbit Type



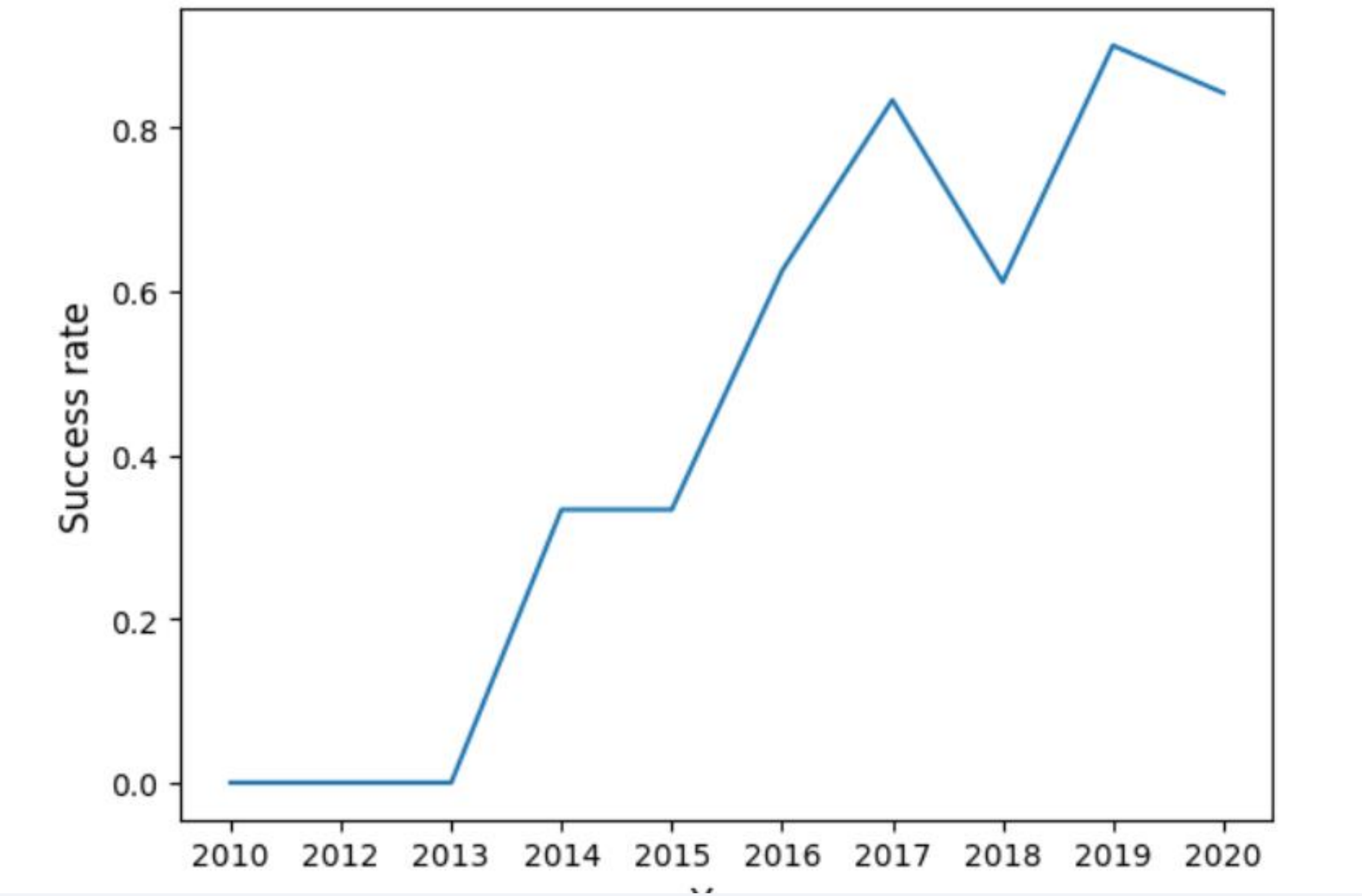
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
[19]: %sql Select Distinct "Launch_Site" from SPACEXTABLE

 * sqlite:///my_data1.db
Done.
```

[19]:

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

Launch Site Names Begin with 'CCA'

[23]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

Task 3 ¶

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

[24]:
```
%sql SELECT SUM("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE "Customer" LIKE '%NASA (CRS)%'
```

```
 * sqlite:///my_data1.db
Done.
```

[24]:

SUM("PAYLOAD_MASS__KG_")
48213

Average Payload Mass by F9 v1.1

```
[25]: %sql SELECT AVG("PAYLOAD_MASS__KG_") from SPACEXTABLE WHERE "Booster_Version" LIKE '%F9 v1.1%'

 * sqlite:///my_data1.db
Done.
```

[25]:

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

```
[27]: %sql SELECT MIN("Date") from SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Succ%'
```

```
 * sqlite:///my_data1.db
Done.
```

[27]:

MIN("Date")
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

[33]:
```
%sql SELECT DISTINCT "Booster_Version" from SPACEXTABLE WHERE "Landing_Outcome" LIKE '%Succ%' AND "Landing_Outcome" LIKE
```

```
 * sqlite:///my_data1.db
Done.
```

[33]:

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[42]:
```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") from SPACEXTABLE GROUP BY "Mission_Outcome"
```

```
 * sqlite:///my_data1.db
Done.
```

[42]:

Mission_Outcome	COUNT("Mission_Outcome")
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

[47]:
```
%sql SELECT DISTINCT "Booster_Version","PAYLOAD_MASS__KG_" FROM SPACEXTABLE WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOA
```

```
 * sqlite:///my_data1.db
Done.
```

[47]:

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

2015 Launch Records

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

[52]:
```
%sql SELECT substr(Date, 6,2), substr(Date,0,5), "Landing_Outcome", "Booster_Version", "Launch_Site" FROM SPACEXTABLE WHER
```

```
 * sqlite:///my_data1.db
Done.
```

[52]:

substr(Date, 6,2)	substr(Date,0,5)	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

Task 10 ¶

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.

[54]:
```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") from SPACEXTABLE WHERE "Date" BETWEEN '2010-06-04' and '2017-03-20' GRO
```

```
 * sqlite:///my_data1.db
Done.
```

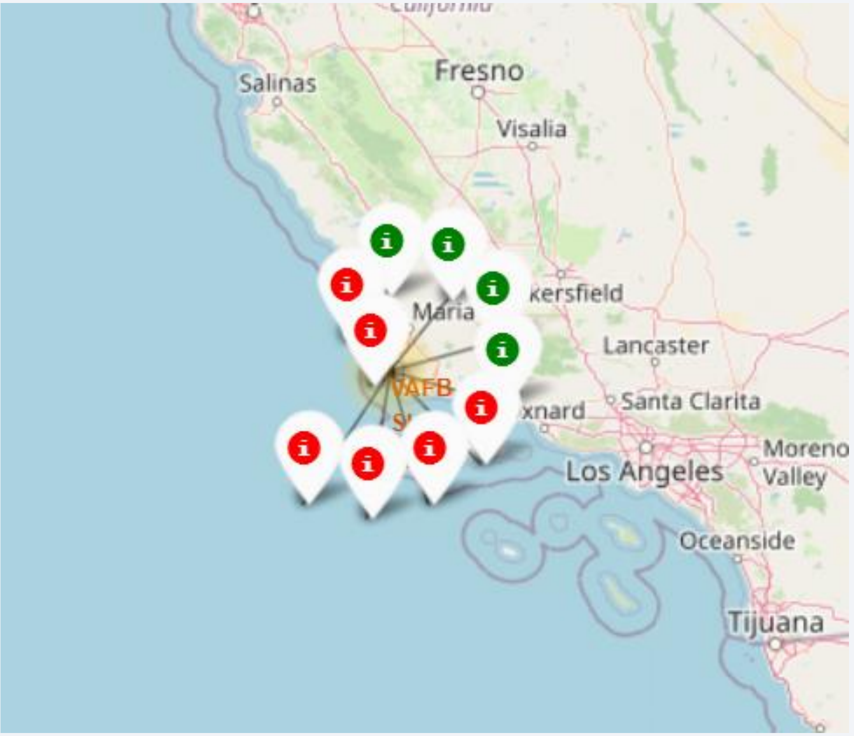
[54]:

Landing_Outcome	COUNT("Landing_Outcome")
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

Locations of launching sites

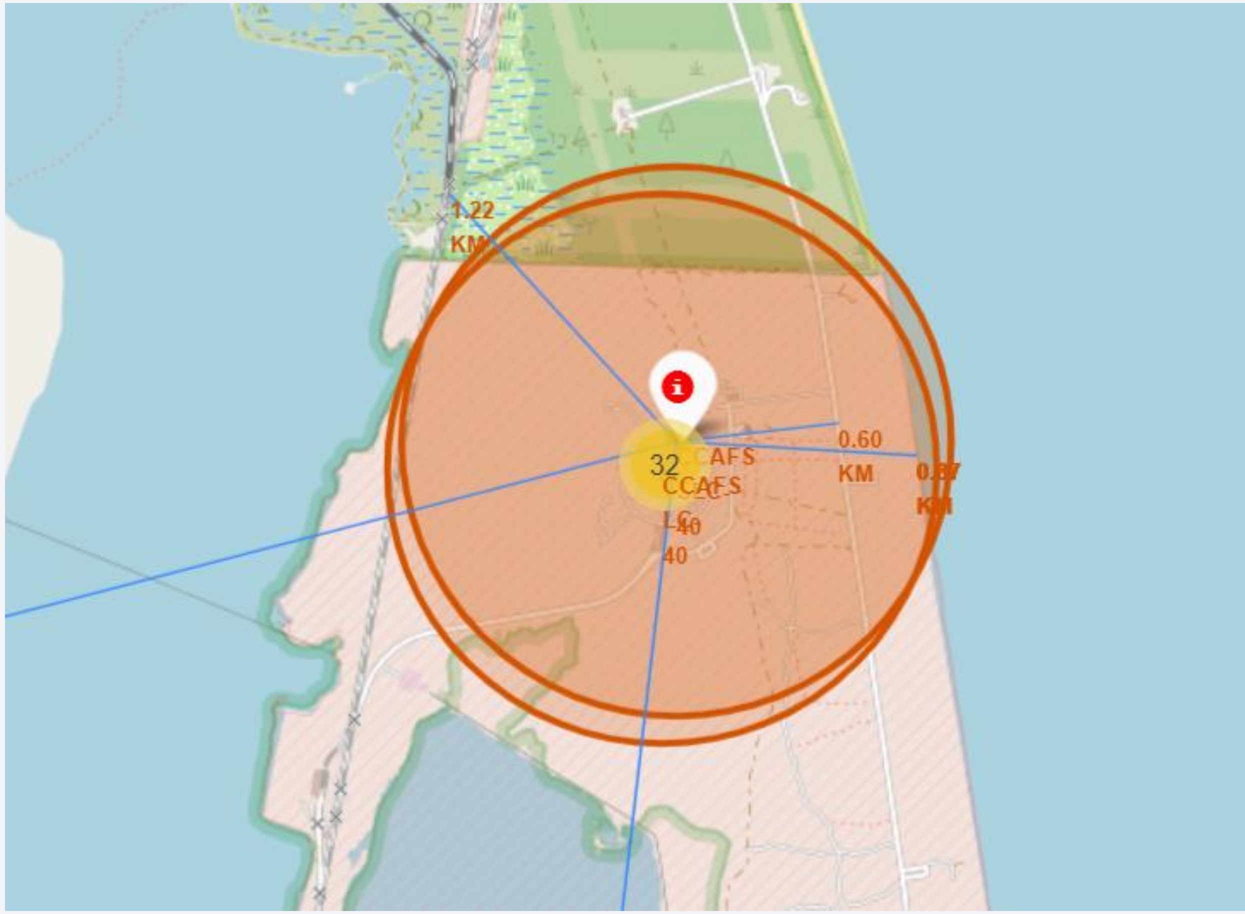
- Launch sites are usually close to ocean, and distant from cities. They are also close to Equator line

Landing Outcome of Launch sites



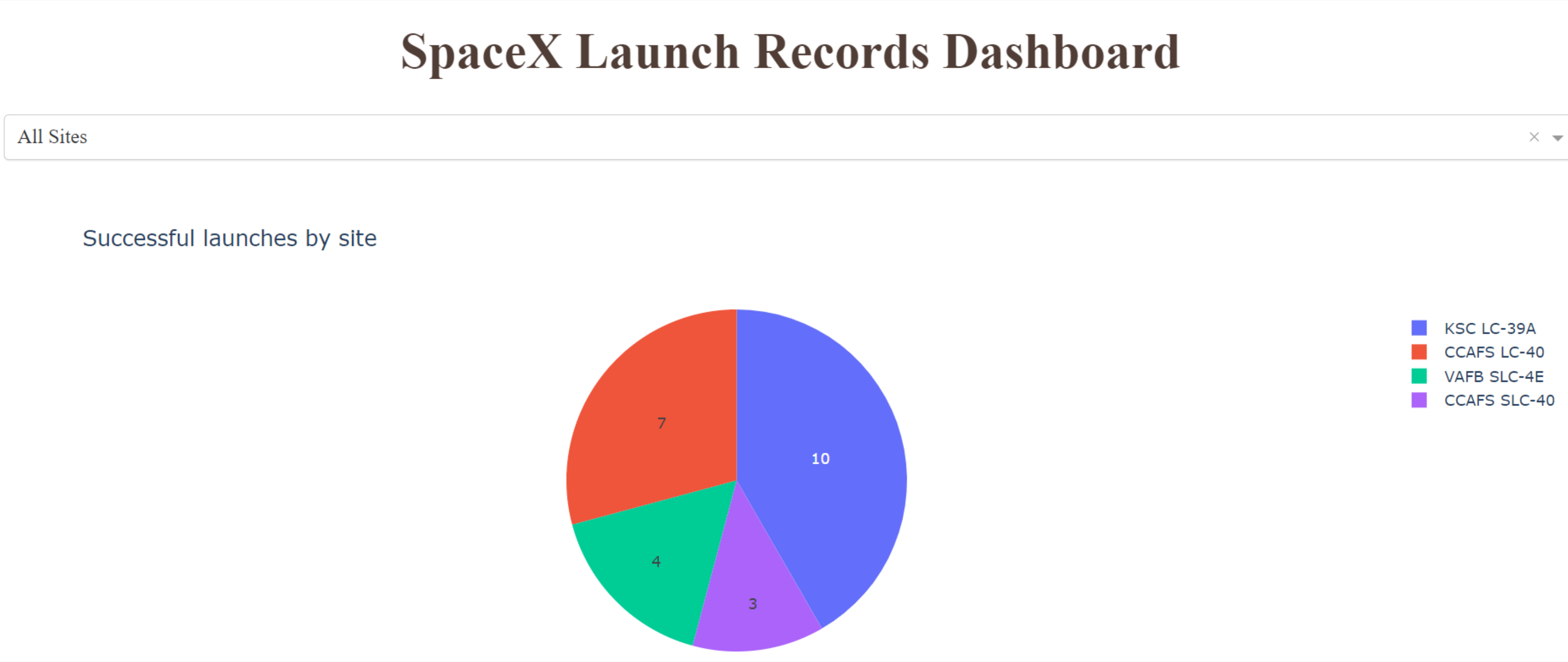
- Cape Canaveral LC39A concentrating success outcome

Service Infrastructure close to the launch site



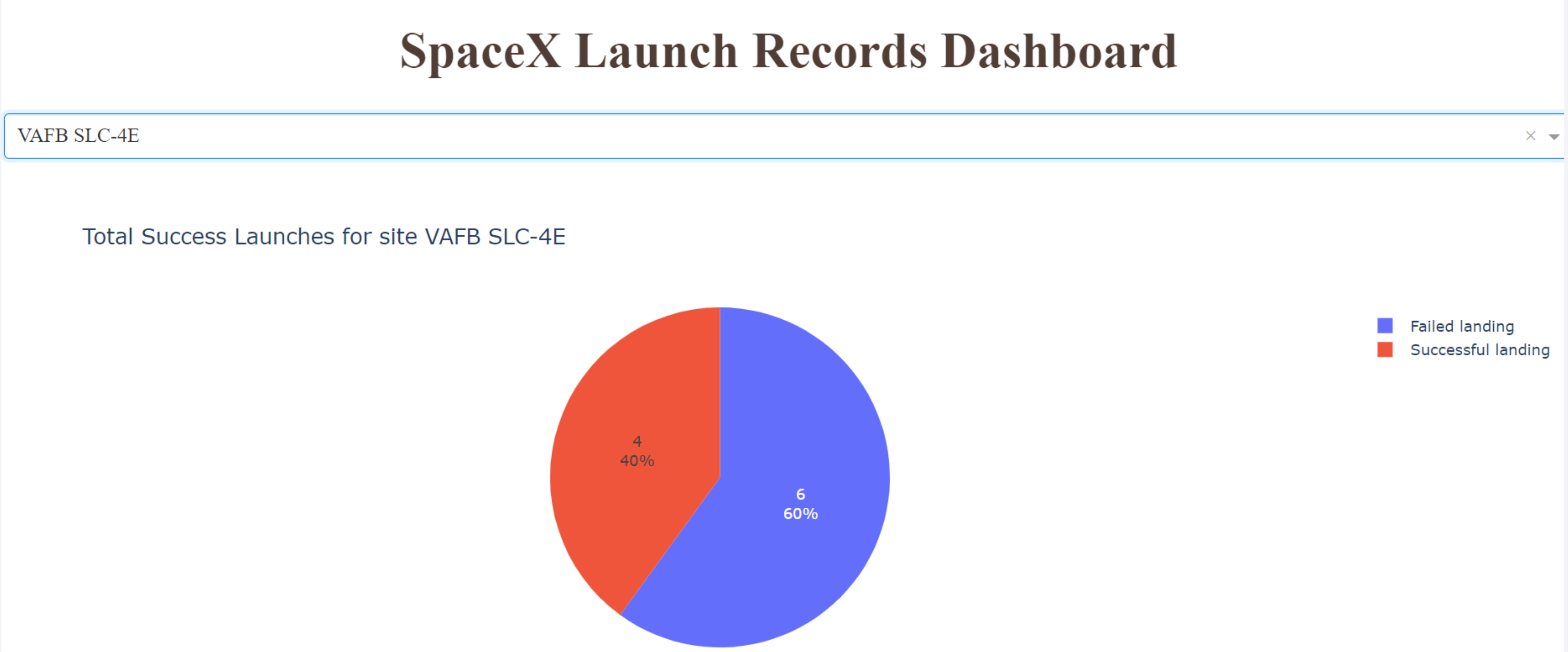
- Railway, roads and airport closer to the launch site are for use of the base
- Closest cities are more than 15 km from the base

First Stage success landing per site

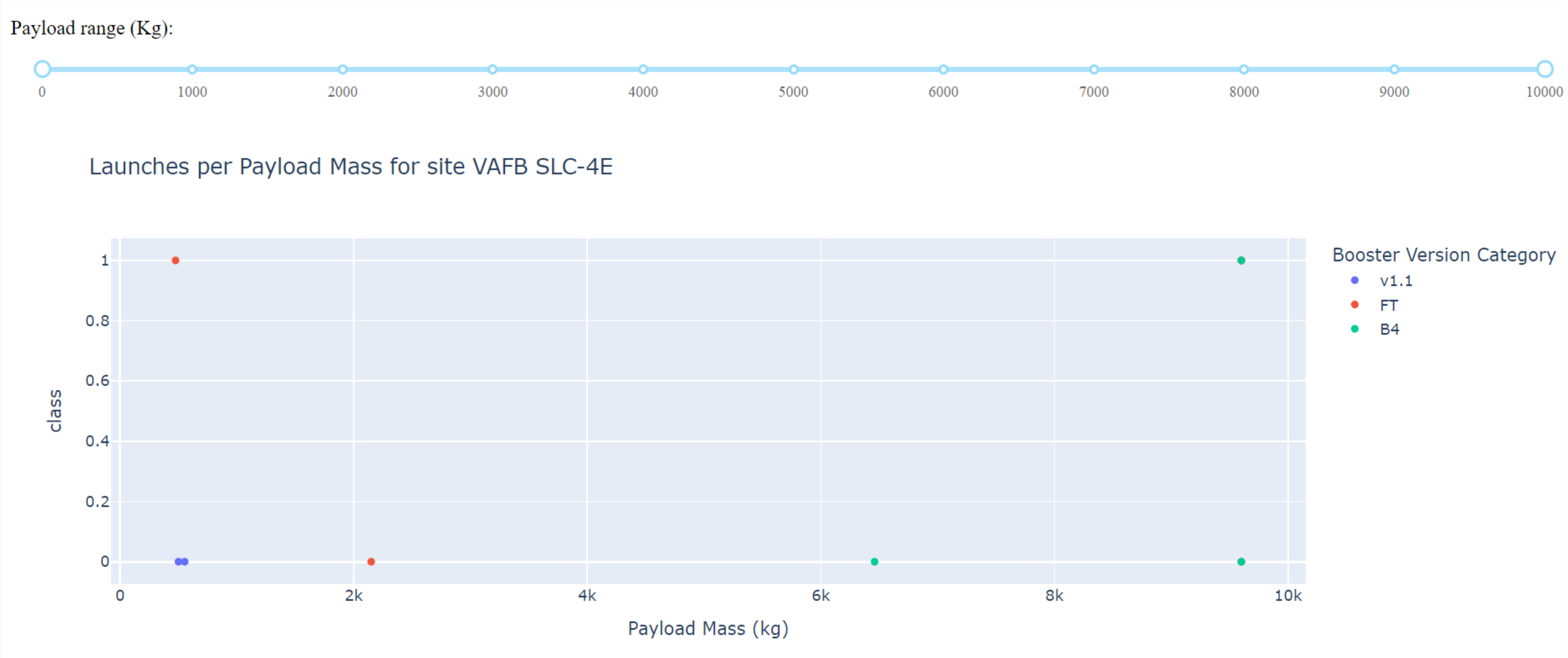


First Stage landing x payload and booster version

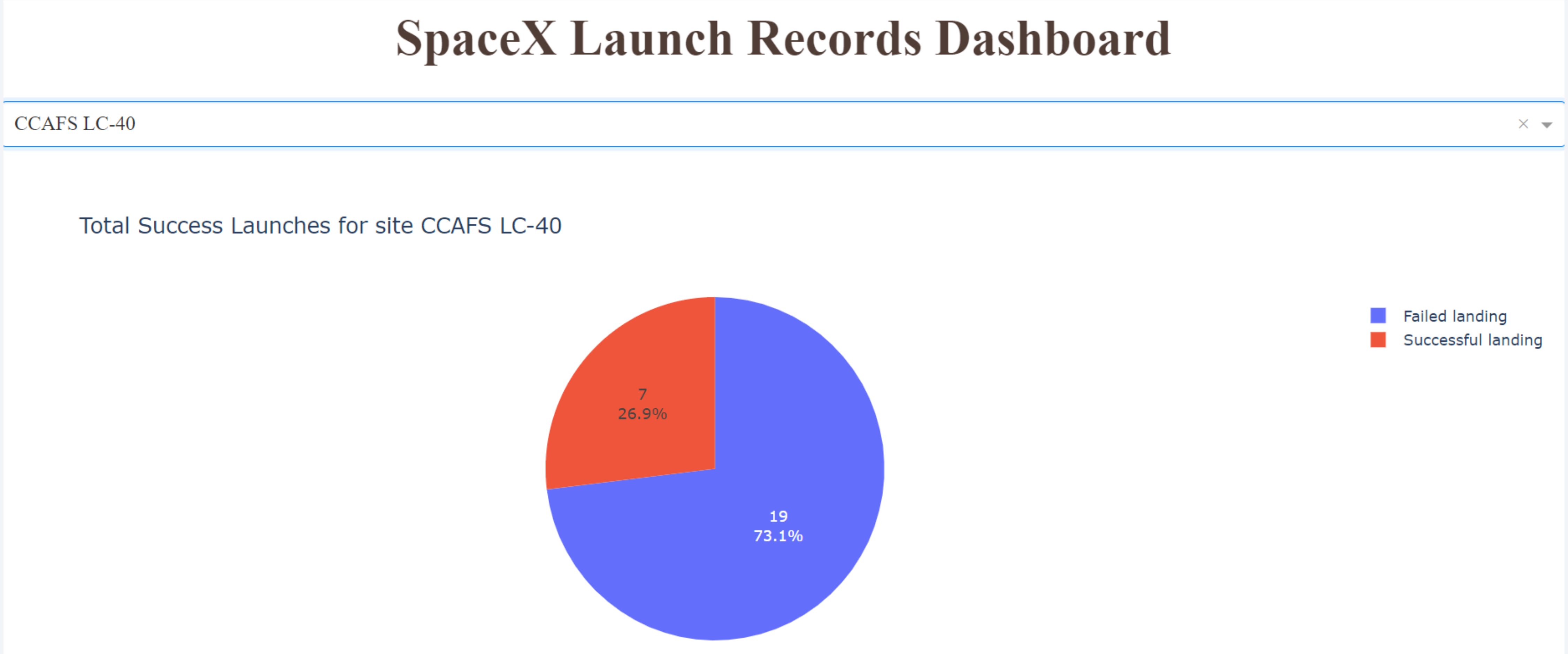
VAFB SLC 4E


SpaceX Launch Records Dashboard
VAFB SLC-4E
Total Success Launches for site VAFB SLC-4E
4
40%
6
60%
Failed landing
Successful landing

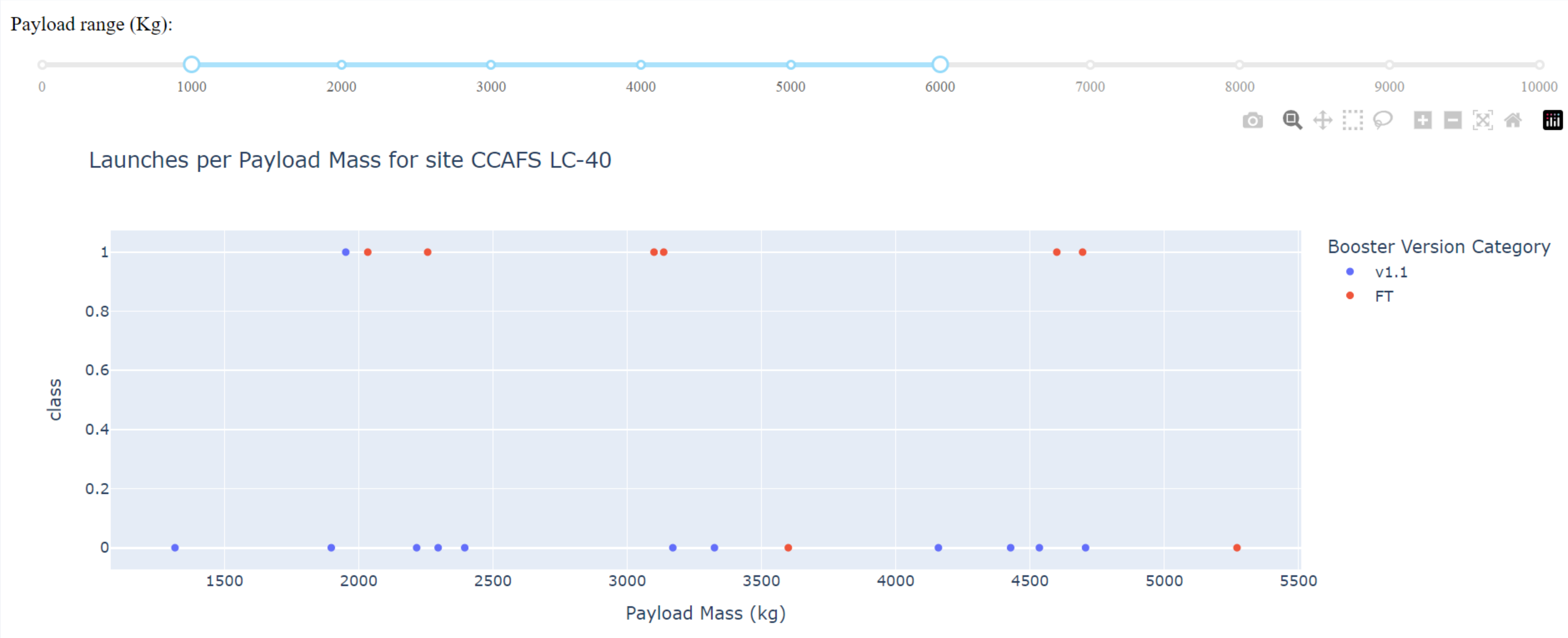
VAFB SLC 4E x Payload and Booster version



CCAAFS LC40

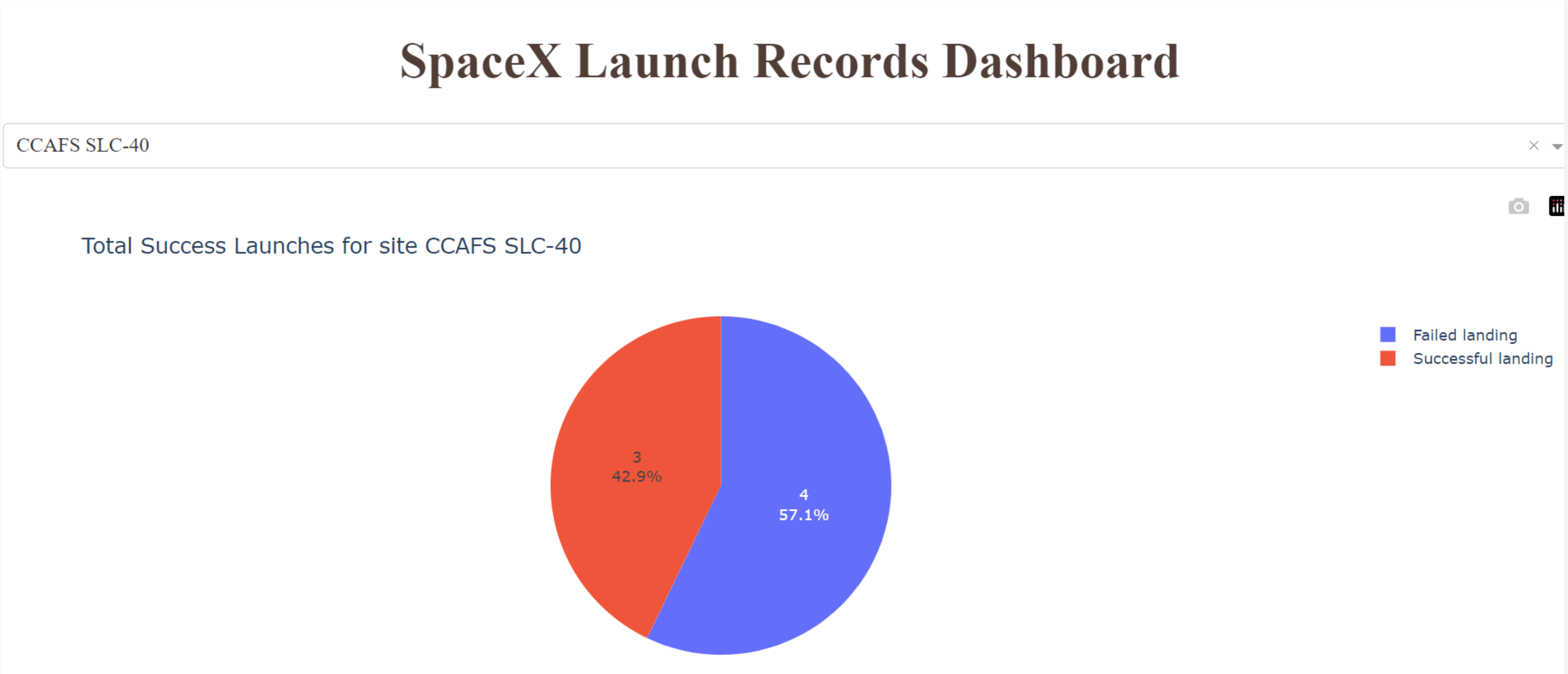

SpaceX Launch Records Dashboard
CCAFS LC-40
Total Success Launches for site CCAFS LC-40
7
26.9%
19
73.1%
Failed landing
Successful landing

CCAAFS LC40

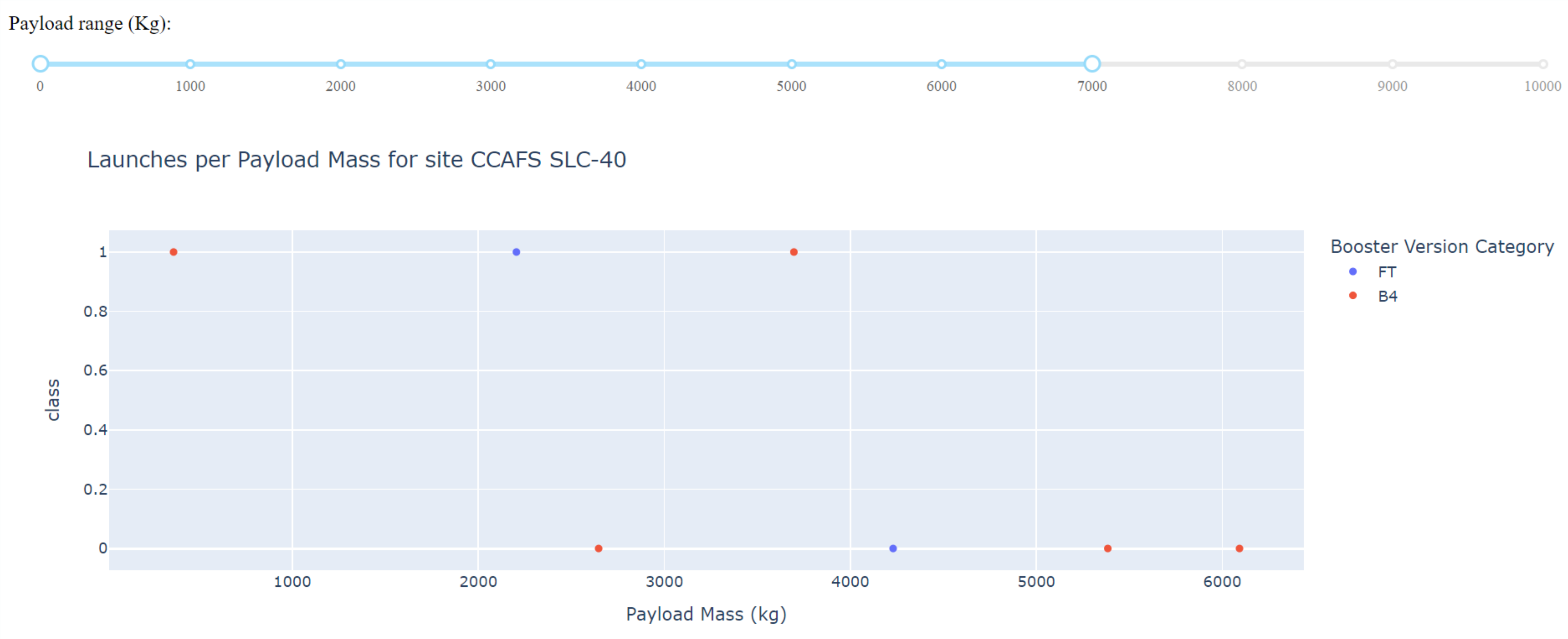


Payload range (Kg):
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
Launches per Payload Mass for site CCAFS LC-40
class
1
0.8
0.6
0.4
0.2
0
1500
2000
2500
3000
3500
4000
4500
5000
5500
Payload Mass (kg)
Booster Version Category
v1.1
FT

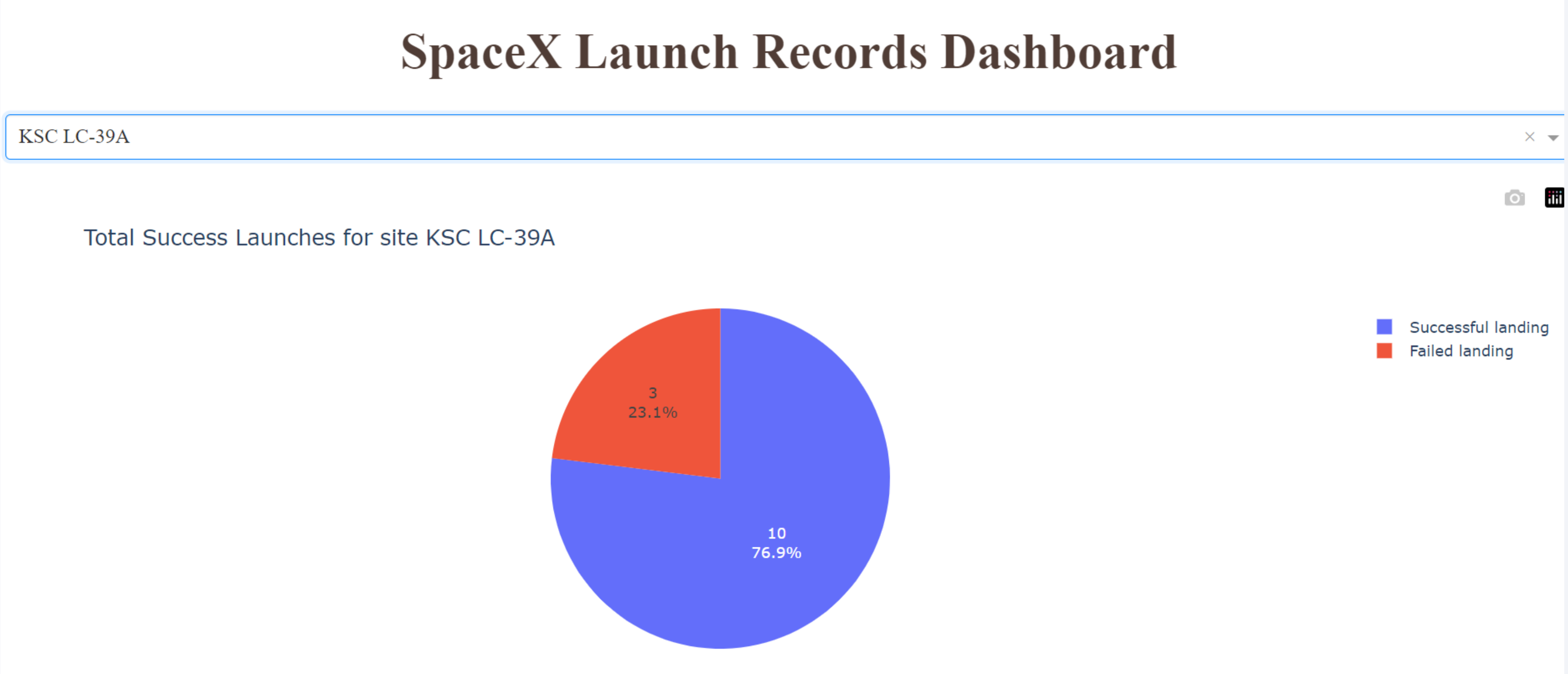
CCAAFS SLC40



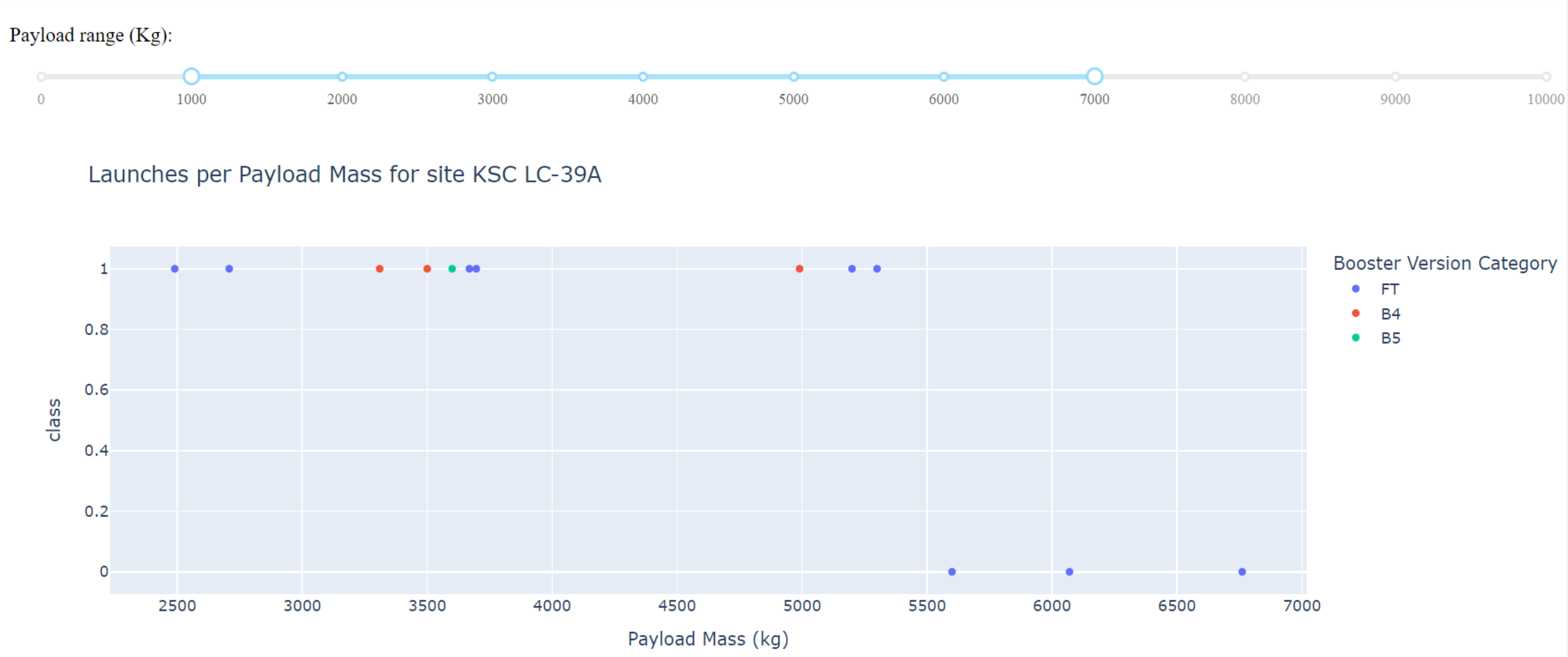
CCAAFS SLC40 x Payload and Booster Version



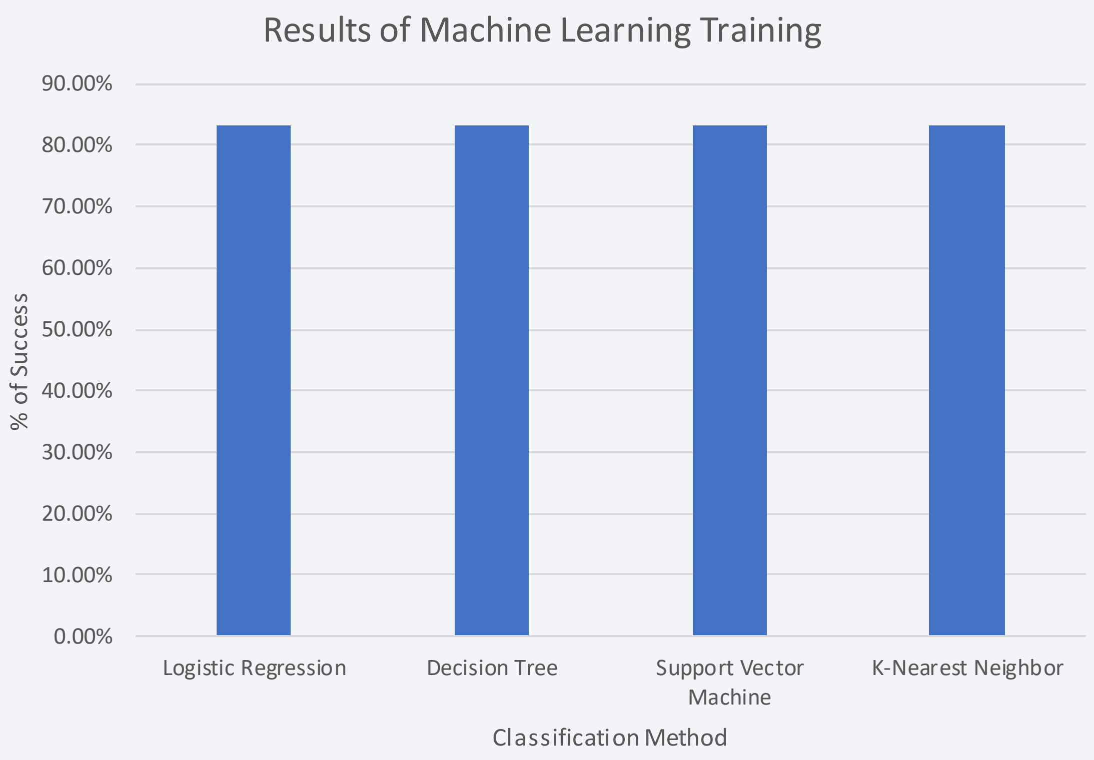
KSC LC39A


SpaceX Launch Records Dashboard
KSC LC-39A
Total Success Launches for site KSC LC-39A
3
23.1%
10
76.9%
Successful landing
Failed landing

KSC LC39A x Payload and Booster version

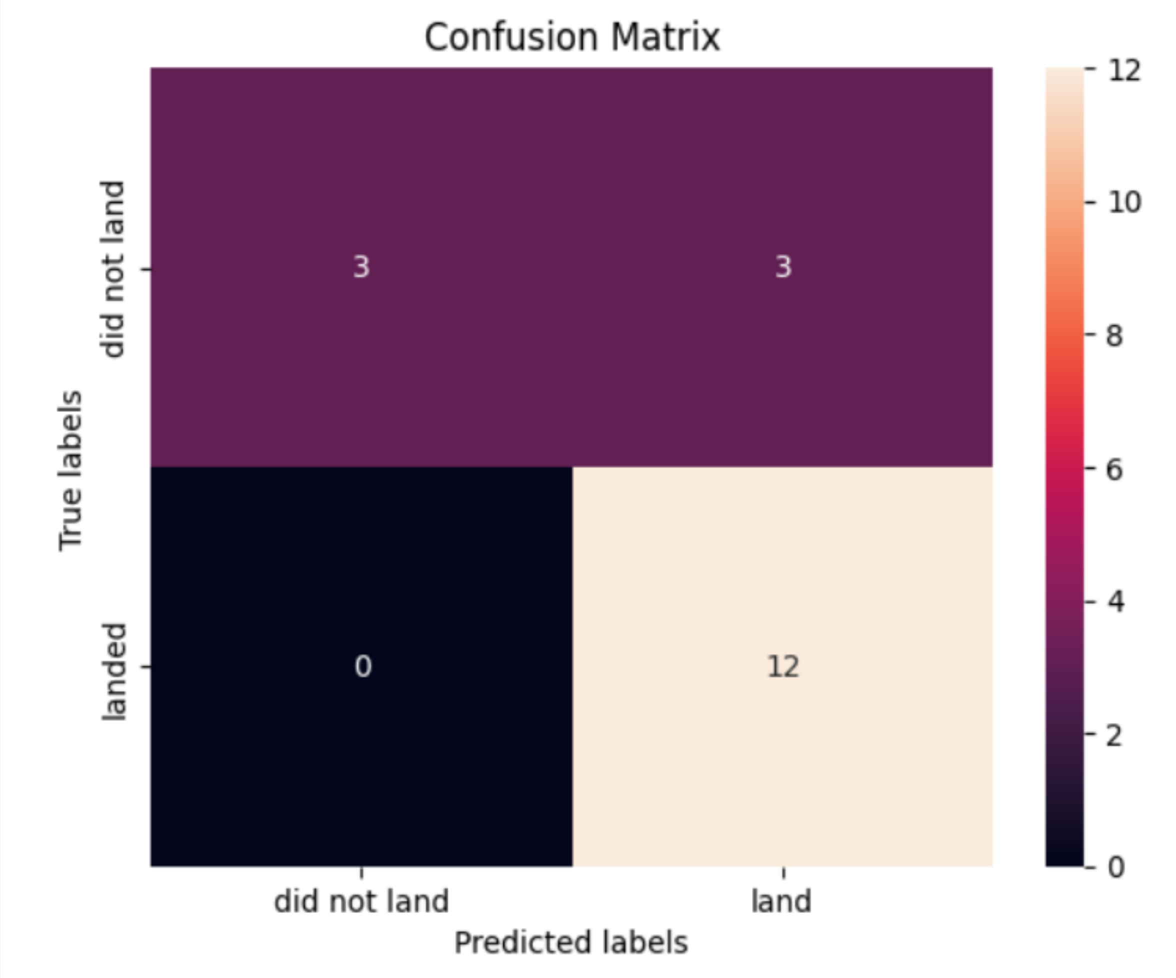


Classification Accuracy



- All methods had a similar accuracy of 83,33%, in predicting the landing outcome.

Confusion Matrix



- All four models tested resulted in similar results on our Test Set
- No false negatives were predicted, but we had 3 false positives out of 18 samples resulting in an accuracy of 83,33%

Conclusions

The best set of parameters for each method are:

- Logistic regression: {'C': 0.01, 'penalty': 'l2', 'solver': 'lbfgs'}
- SVM: {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
- Decision Tree: {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'splitter': 'best'}
- KNN: {'algorithm': 'auto', 'n_neighbors': 10, 'p': 1}

Highest accuracy on the Train set was achieved at Decision Tree (88.9%) the other methods achieves 84%.

On the Test Set the accuracy achieved was 83.3%.