IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Yusuke Funaki
20th-Oct-2022

Outline

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

Executive Summary

- Summary of methodologies
    1. Data Collection (from the Space X API and web scraping)
    2. Data Wrangling
    3. EDA with SQL
    4. EDA with Data Visualization
    5. Building an Interactive Map with Folium
    6. Building a Dashboard with Ploty Dash
    7. Machine Learning Prediction (Classification)
- Summary of all results
    - EDA results
    - Interactive Map and Dashboard
    - Analytic results with Machine Learning

Introduction

- Project background and context

  Space X 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 Space X 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 space X for a rocket launch.

- Problems you want to find answers

  What factors are behind the failure of the missions.

  What factors are behind the success of the missions and its accuracy.

Section 1

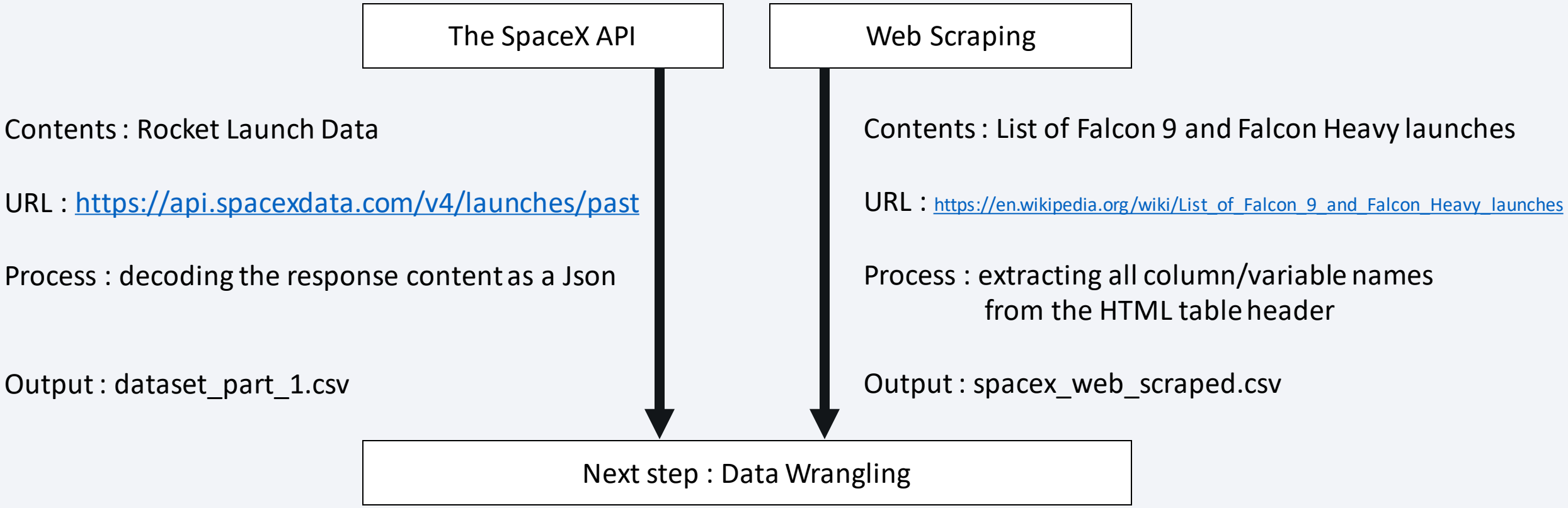
Methodology

Methodology

Executive Summary

- Data collection methodology:
  - From SpaceX API
  - With Web Scraping from wikipedia
- Perform data wrangling
  - Calculating the number of launches on each site, the number and occurrence of each orbit, the number and occurrence of mission outcome per orbit type
  - Creating a landing outcome label from outcome column 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
  - logistic regression, support vector machine, a decision tree and k nearest neighbors

Data Collection



Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API
   https://api.spacexdata.com/v4/launches/past

2. Requesting and parse the SpaceX launch data
   https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json

3. Decoding the response content as a Json and turn it into a Pandas dataframe

4. Pre-processing for cleaning data

5. Filtering the dataframe to only include Falcon 9 launches

dataset_part_1.csv

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/02ddfd3fae342f255a4fc3b1b9944f046b388379/01_jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1. Performing an HTTP GET method to request the Falcon9 Launch HTML page
   https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

2. Creating a BeautifulSoup object from the HTML response

3. Extracting all column/variable names from the HTML table header

4. Creating a data frame by parsing the launch HTML tables

spacex_web_scraped.csv

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/02ddfd3fae342f255a4fc3b1b9944f046b388379/02_jupyter-labs-webscraping.ipynb

Data Wrangling

1. Loading Space X dataset : **dataset_part_1.csv**

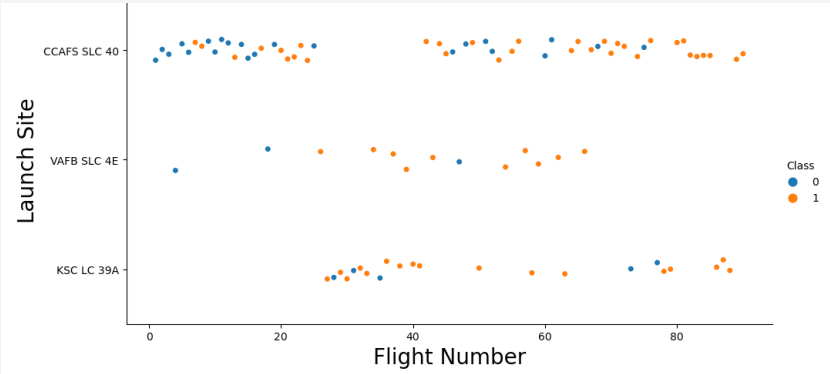
2. Calculate the number of launches on each site
3. Calculate the number and occurrence of each orbit
4. Calculate the number and occurence of mission outcome per orbit type

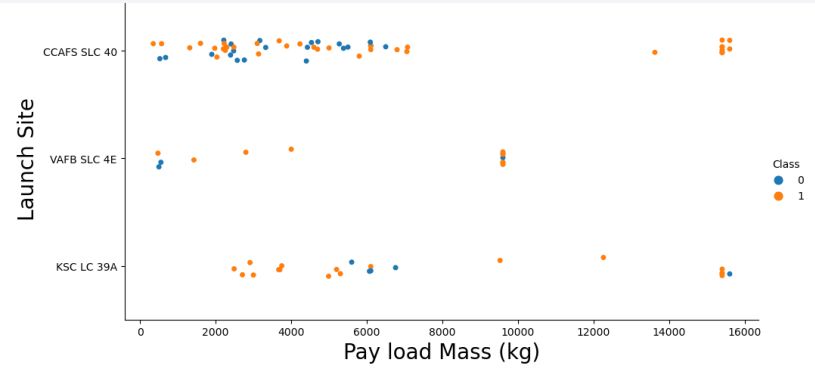
5. Create a landing outcome label from Outcome column

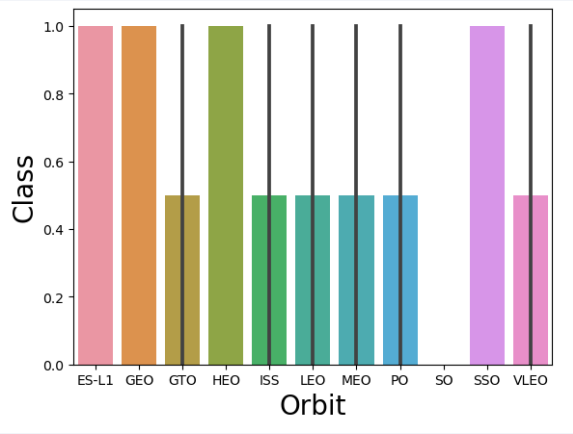
dataset_part¥_2.csv

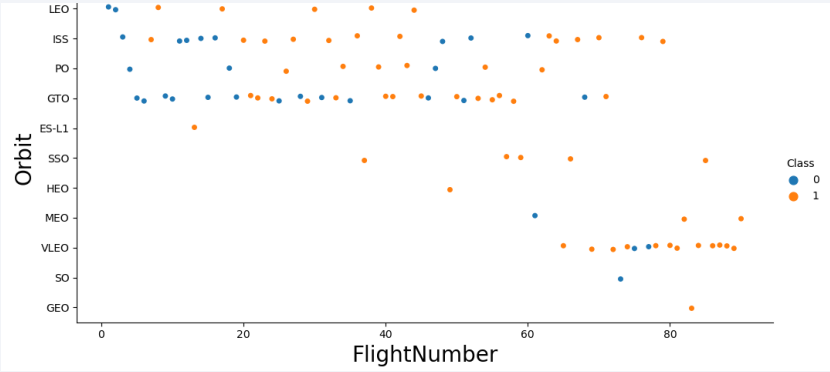
GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/02ddfd3fae342f255a4fc3b1b9944f046b388379/03_labs-jupyter-spacex-Data%20wrangling.ipynb

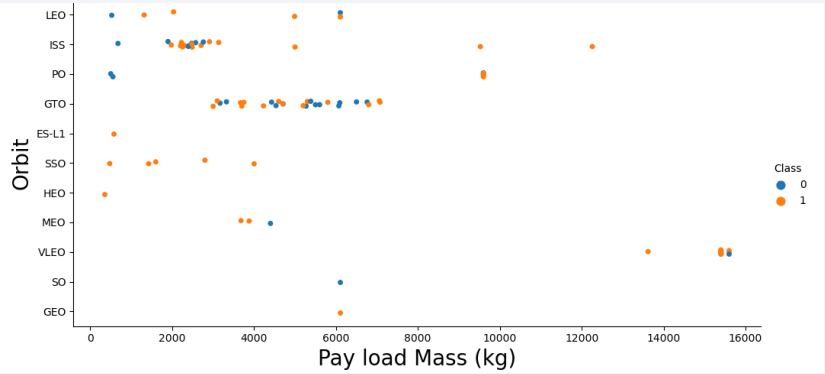
EDA with Data Visualization

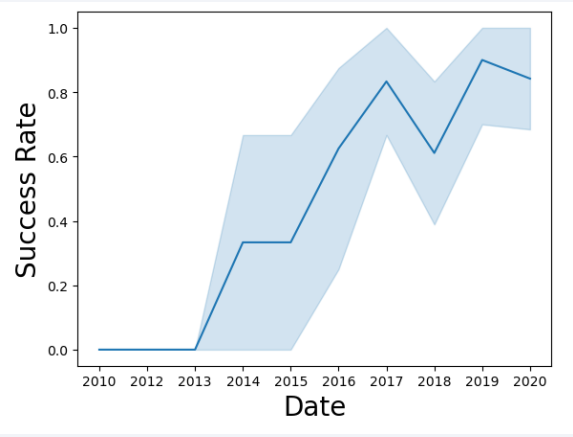












GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/04c03bb89f6e874db1d9fc79e981fdb679ee75d7/06_lab_jupyter_launch_site_location.ipynb

EDA with SQL

10 tasks to answer with SQL queries :

- 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 succesful landing outcome in ground pad was acheived.
- 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 records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/02ddfd3fae342f255a4fc3b1b9944f046b388379/04%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

spacex_launch_geo.csv

- folium.Map() : to mark all launch sites on a map
- folium.Circle() : to add a highlighted circle area with a text label on a specific coordinate
- folium.map.Marker() : to add a circle for each launch site
- folium.Icon() : to indicate if this launch was successed or failed
- folium.PolyLine() : to draw a PolyLine between a launch site to the selected coastline point

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/02ddfd3fae342f255a4fc3b1b9944f046b388379/06_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

building a Plotly Dash application to see launch success rate

https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/spacex_launch_dash.csv
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_3/spacex_dash_app.py

1. Pie-Chart

- Showing success rate of each site
- having dropdown component for selecting the site

2. Scatter-Plot

- Showing plots of payload mass and success index
- having rangeslider component used for selecting the range of payload mass

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/6793de036f382d461d947a08ea7e1e9c353e2965/07_spacex_dash_app.py

Predictive Analysis (Classification)

dataset_part_2.csv, dataset_part_3.csv

1. Loading the data
2. Standardizing the data
3. Splitting the data X and Y into training and test data

4. Buiding the models

logistic regression | support vector machine | a decision tree | k nearest neighbors

5. Calculating the accuracy on the test data

6. Creating the confusion matrix

GitHub URL of whole process:
https://github.com/YusukeF0912/ibm_ds_pga_course10/blob/80877dd59e0042dbe804ad41ab72d7eb0adf4ef3/08_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results

  The Orbit, ES-L1, GEO, HEO and SSO, have 100% success rate.

  The Orbit, SO, has 0% success rate.

  The success rate is about 80% recently.

- Interactive analytics demo in screenshots

  A launch with heavier payload mass has higher success rate.

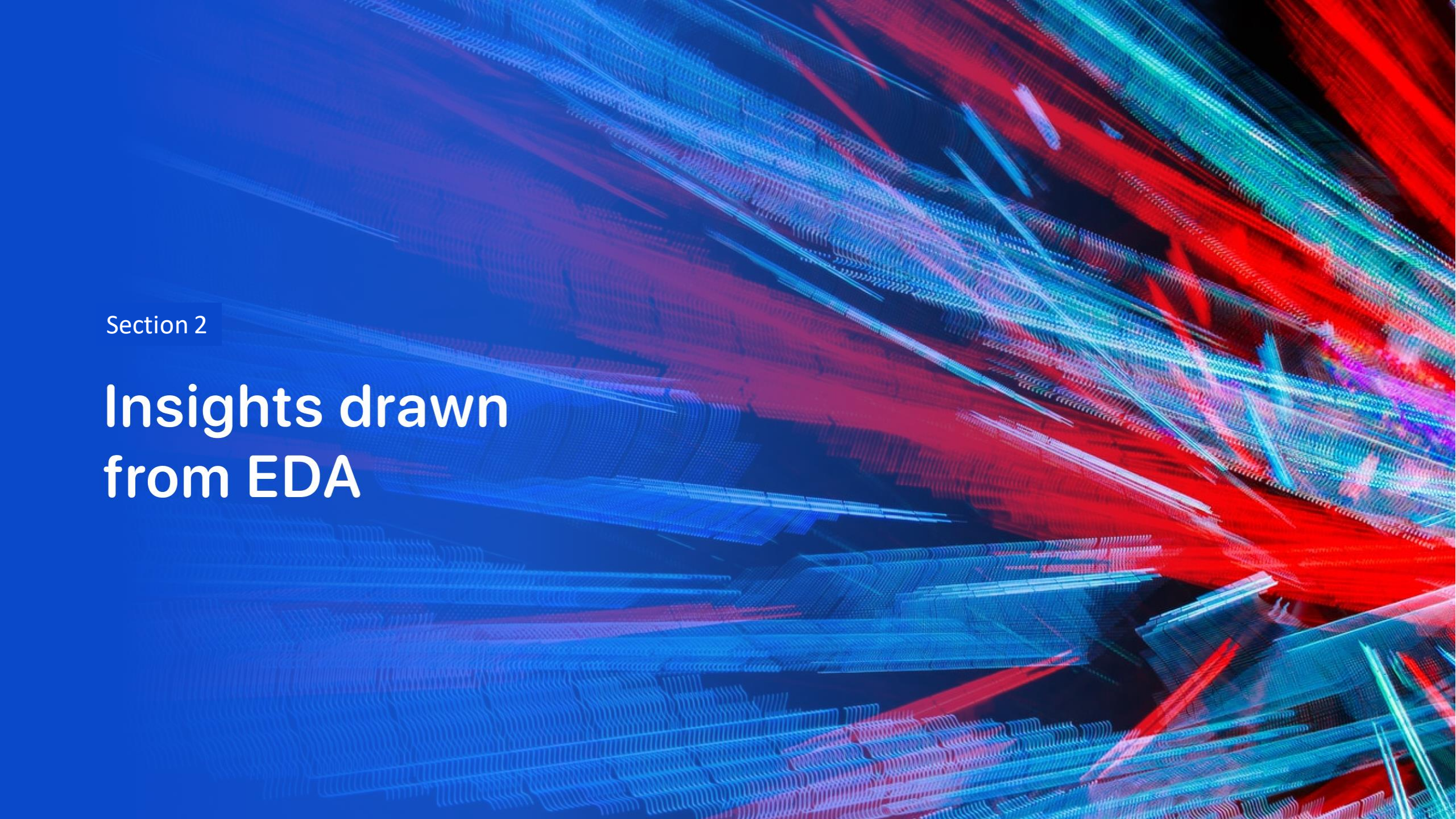
  A launch from KSC LC-39A has the highest success rate.

- Predictive analysis results

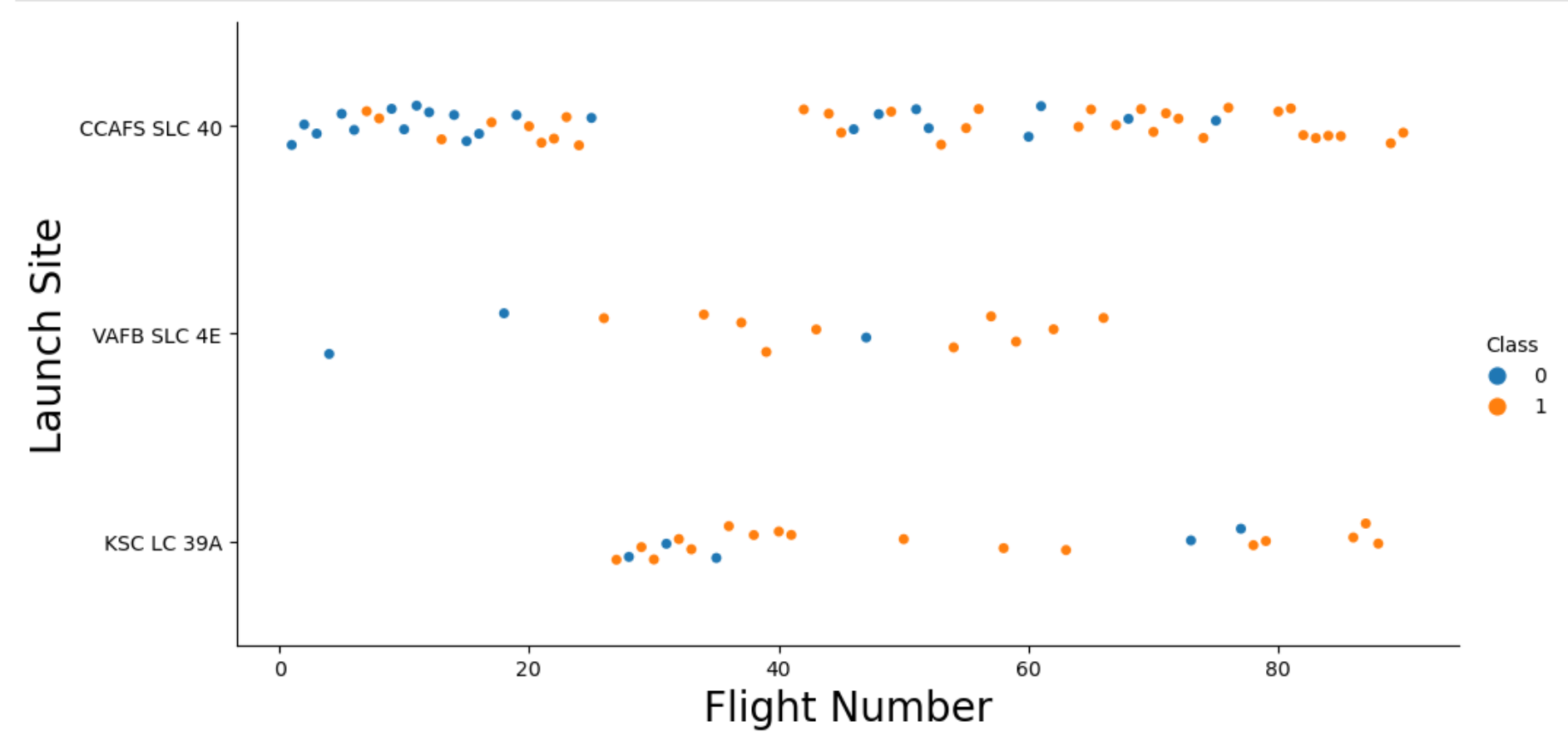
  All 4 machine leaning classification models (logistic regression, support vector machine, a decision tree and k nearest neighbors) have the same accuracy, 0.83333…..

Section 2

Insights drawn from EDA

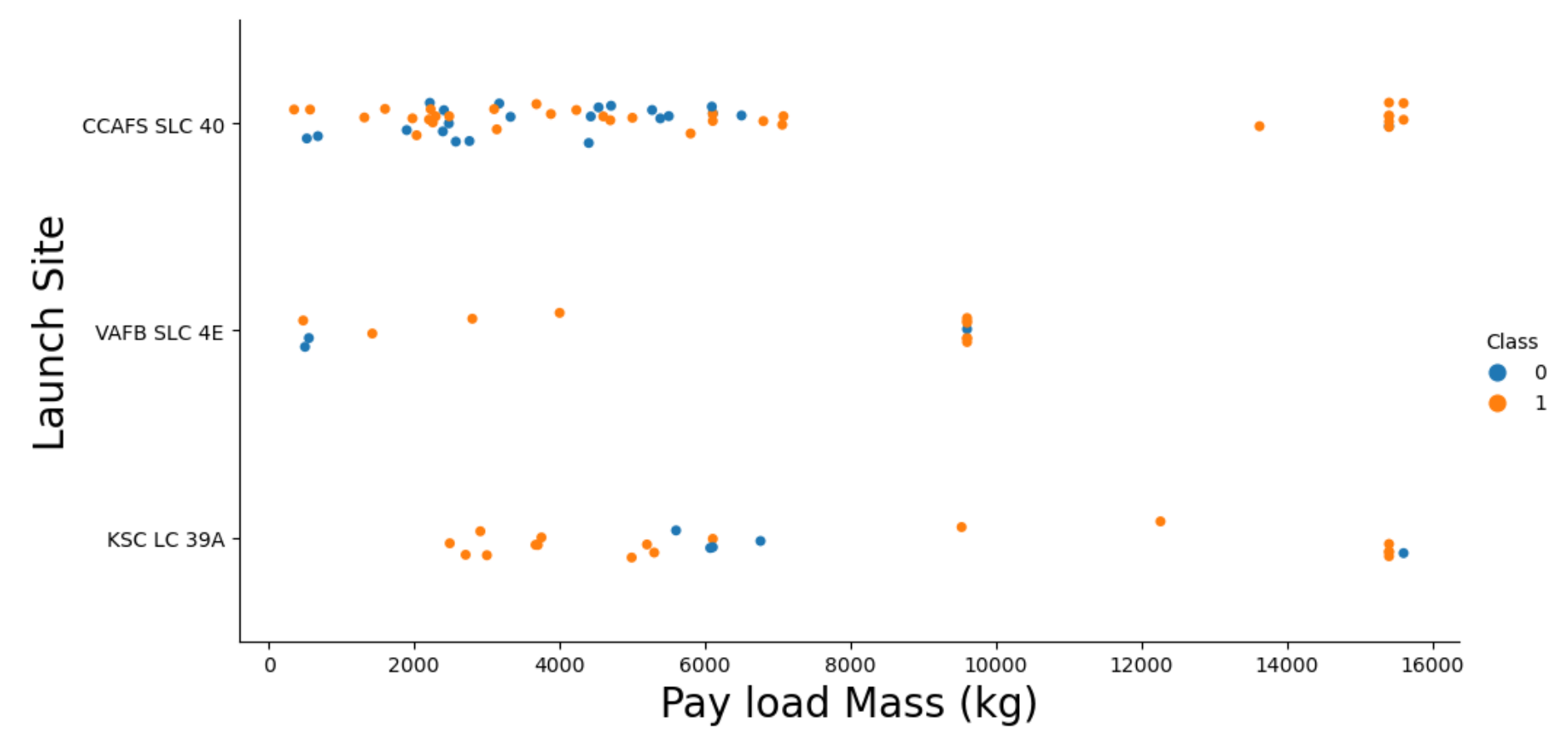


Flight Number vs. Launch Site



At all launch site, success rate increases while Flight Number increases.

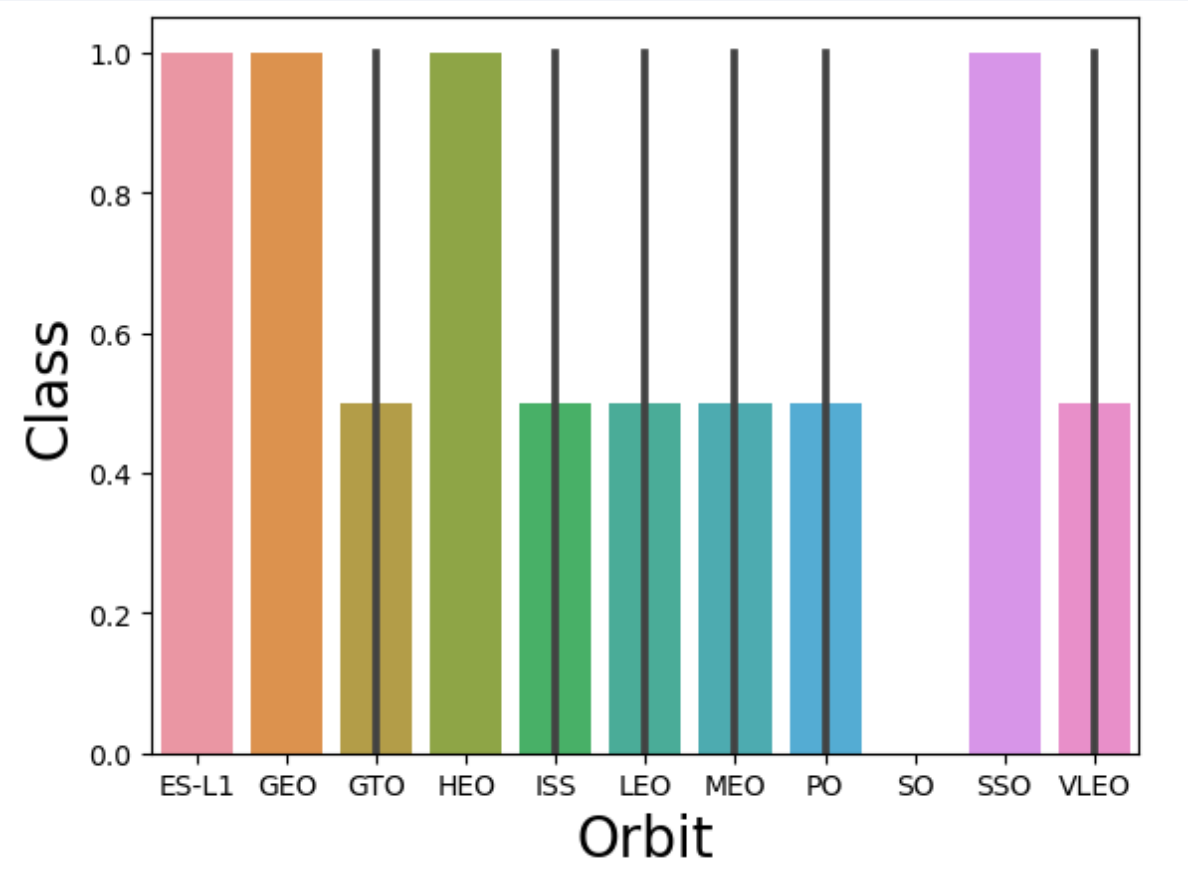
Payload vs. Launch Site



Most launches with Lower Payload mass are performed at CCAFS SLC 40.

The Launces with heavier payload mass perform well (the number is smaller but success rate is higer)

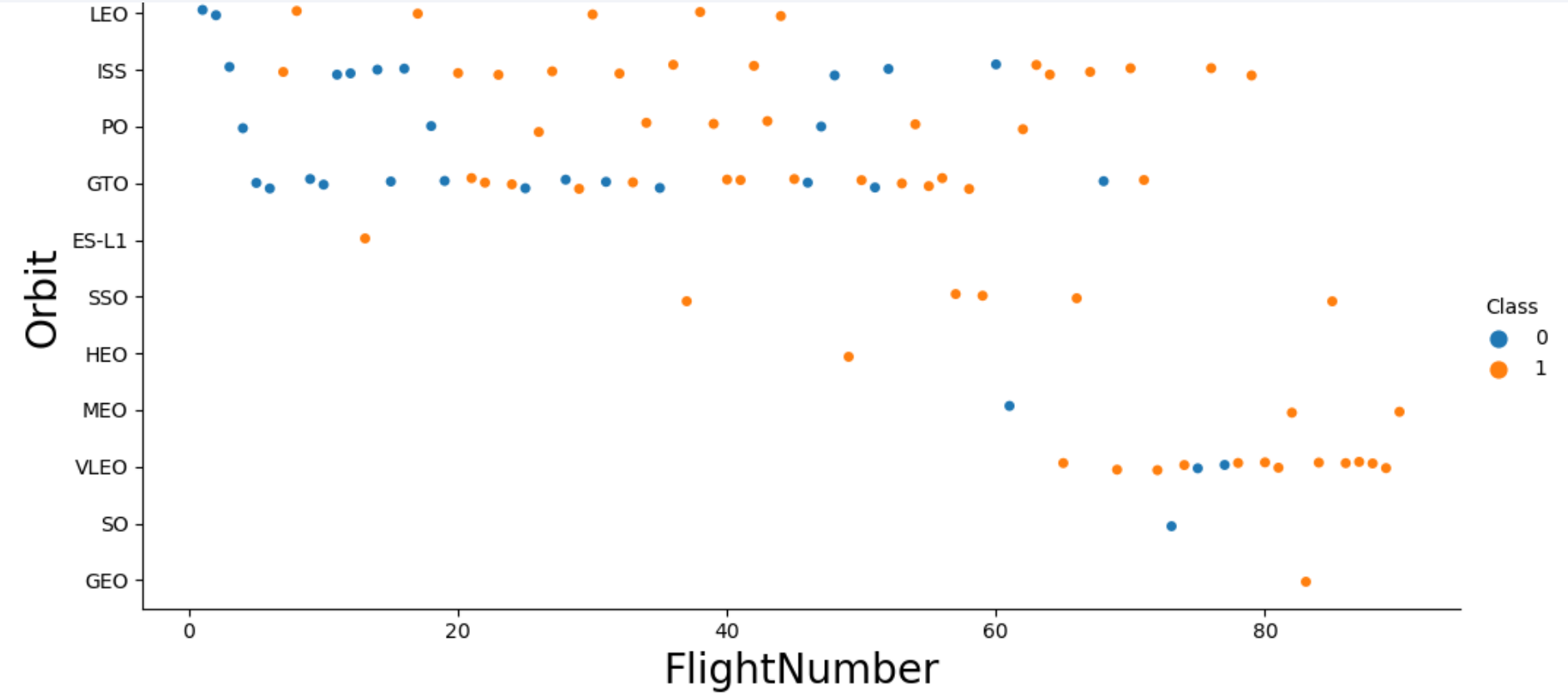
Success Rate vs. Orbit Type



The Orbit, ES-L1, GEO, HEO and SSO, have 100% success rate.

The Orbit, SO, has 0% success rate.

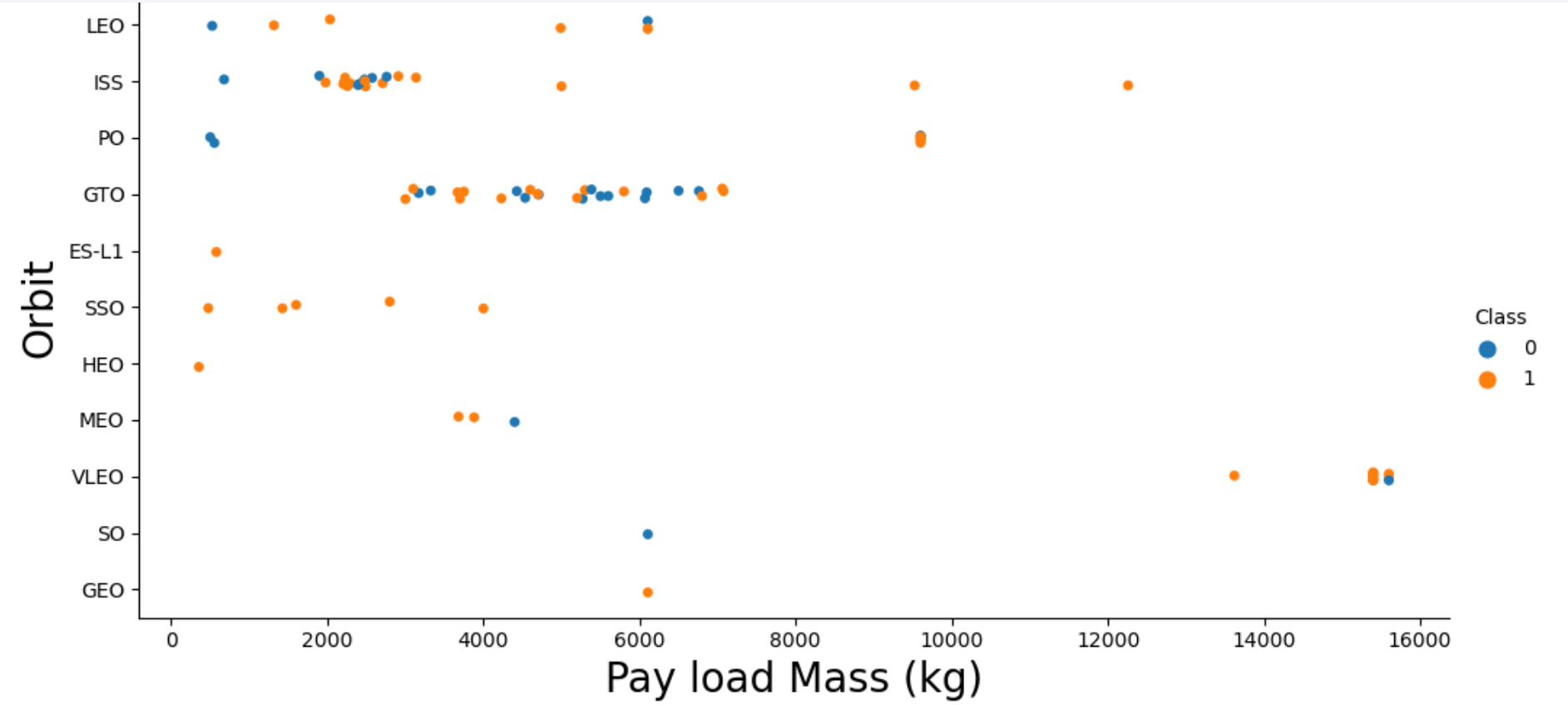
Flight Number vs. Orbit Type



The Orbit has been shifted from LEO, ISS, PO and GTO to ISS and VLEO.

The success rate has been getting better.

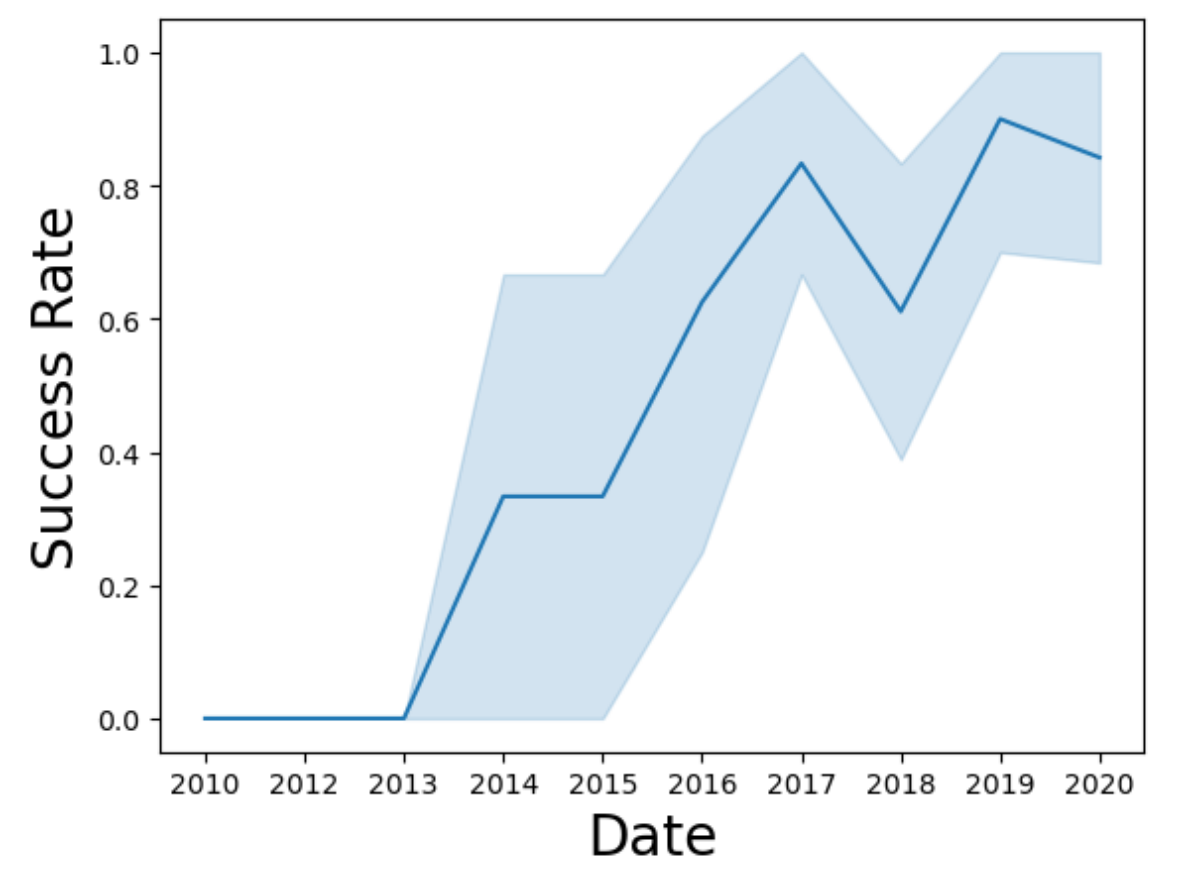
Payload vs. Orbit Type



Most of the range of Pay Load Mass is 2000～4000 [kg] for ISS.

Most of the range of Pay Load Mass is 3000～7000 [kg] for GTO.

Launch Success Yearly Trend



Can see drastic increase from 2013.

The success rate is about 80% recently.

All Launch Site Names

In [7]:

```
%%sql
SELECT Distinct LAUNCH_SITE
FROM SPACEXTBL
```

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

Out[7]:

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

Having 4 Launch Sites

Launch Site Names Begin with 'CCA'

In [8]:

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

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

CCAFS LC-40

Total Payload Mass

- Calculating the total payload carried by boosters from NASA

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE CUSTOMER='NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculating the average payload mass carried by booster version F9 v1.1

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE BOOSTER_VERSION='F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- Finding the dates of the first successful landing outcome on ground pad

```
%%sql
SELECT DATE
FROM SPACEXTBL
WHERE [Landing _Outcome]='Success (ground pad)'
```

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

Date
22-12-2015
18-07-2016
19-02-2017
01-05-2017
03-06-2017
14-08-2017
07-09-2017
15-12-2017
08-01-2018

2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- Listing the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_
between 4000 and 6000 AND [LANDING _OUTCOME]='Success (drone ship)'
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Calculating the total number of successful and failure mission outcomes

```
%%sql
SELECT COUNT(*)
FROM SPACEXTBL
WHERE MISSION_OUTCOME LIKE '%Success%'
```

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

COUNT(*)
100

Success

```
%%sql
SELECT COUNT(*)
FROM SPACEXTBL
WHERE MISSION_OUTCOME LIKE '%Failure%'
```

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

COUNT(*)
1

Failure

Boosters Carried Maximum Payload

- Listing the names of the booster which have carried the maximum payload mass

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ =
 (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Booster_Version
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

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

```
%%sql
SELECT
    substr(Date, 4, 2) AS MONTH_NAME,
    [LANDING _OUTCOME] AS LANDING_OUTCOME,
    BOOSTER_VERSION AS BOOSTER_VERSION,
    LAUNCH_SITE AS LAUNCH_SITE
FROM SPACEXTBL
WHERE [LANDING _OUTCOME] = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'
```

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

MONTH_NAME	LANDING_OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Ranking the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql
SELECT DATE, [LANDING _OUTCOME] AS 'Landing Outcome', COUNT([LANDING _OUTCOME]) as COUNT
FROM SPACEXTBL
WHERE [LANDING _OUTCOME] LIKE '%Success%' AND NOT DATE LIKE '%2018%' AND NOT DATE LIKE '%2019%' AND NOT DATE LIKE '%2020%' AND NOT DATE LIKE '%04-
GROUP BY DATE
ORDER BY COUNT([LANDING _OUTCOME]) DESC
```

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

Date	Landing Outcome	COUNT
30-03-2017	Success (drone ship)	1
27-05-2016	Success (drone ship)	1
22-12-2015	Success (ground pad)	1
19-02-2017	Success (ground pad)	1
18-07-2016	Success (ground pad)	1
14-08-2016	Success (drone ship)	1
14-01-2017	Success (drone ship)	1
08-04-2016	Success (drone ship)	1
06-05-2016	Success (drone ship)	1

Because of data loading issue about date format,
Couldn’t make the adequate query.

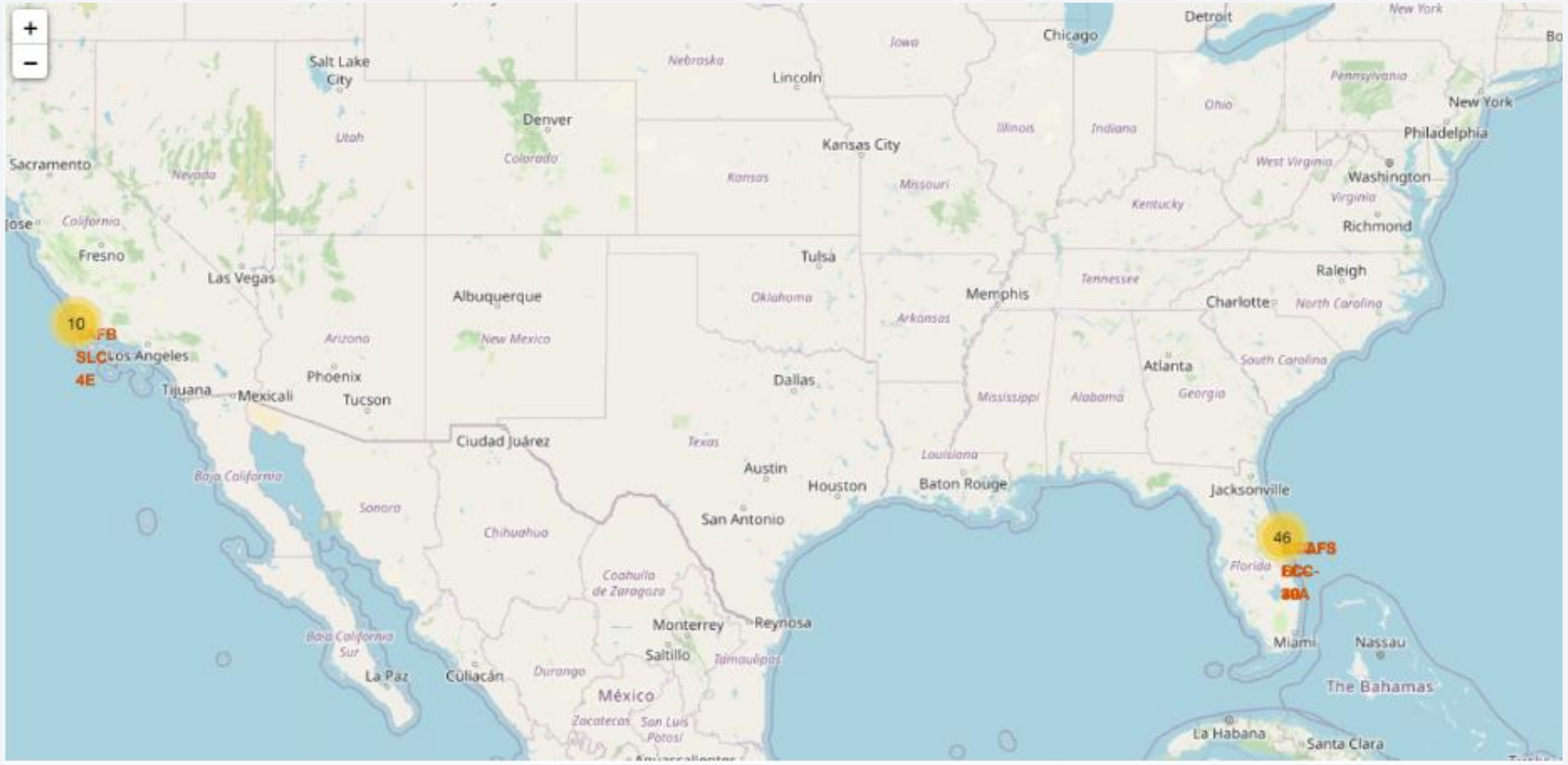
Section 3

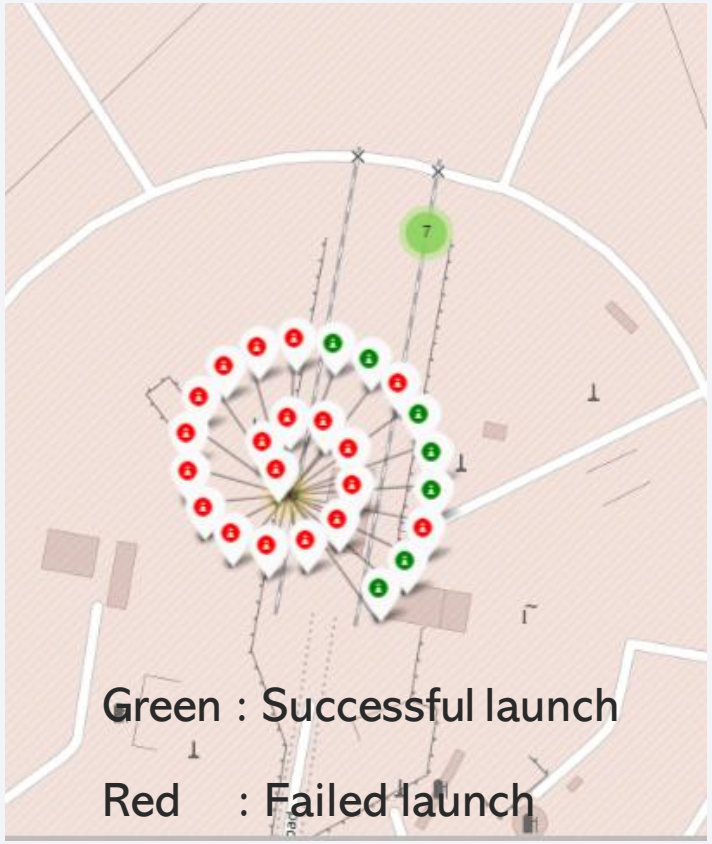
Launch Sites Proximities Analysis

All Launch Site on the map

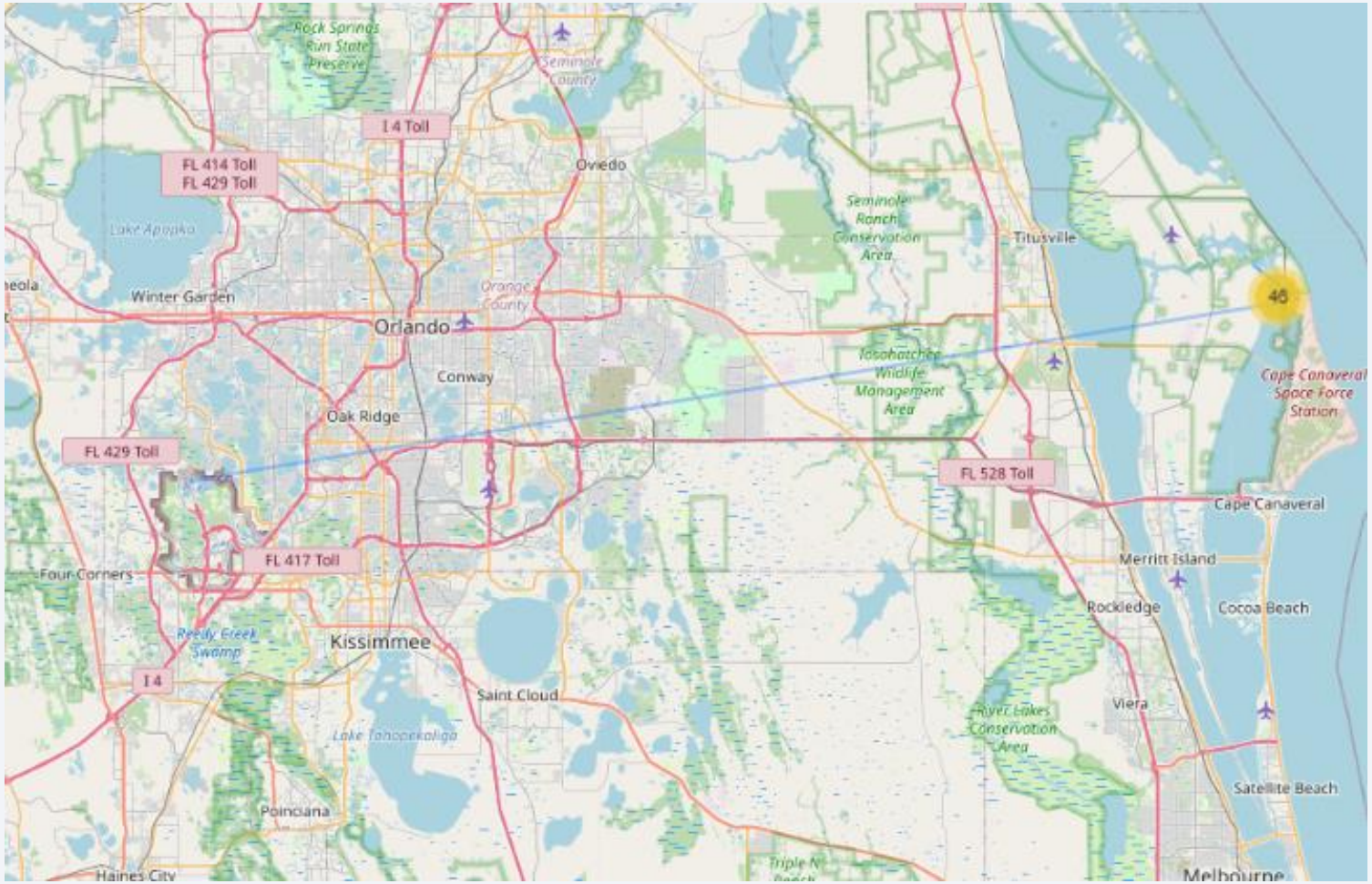
The location of all launch sites are in Florida or California.

The color-labeled launch outcomes on the map

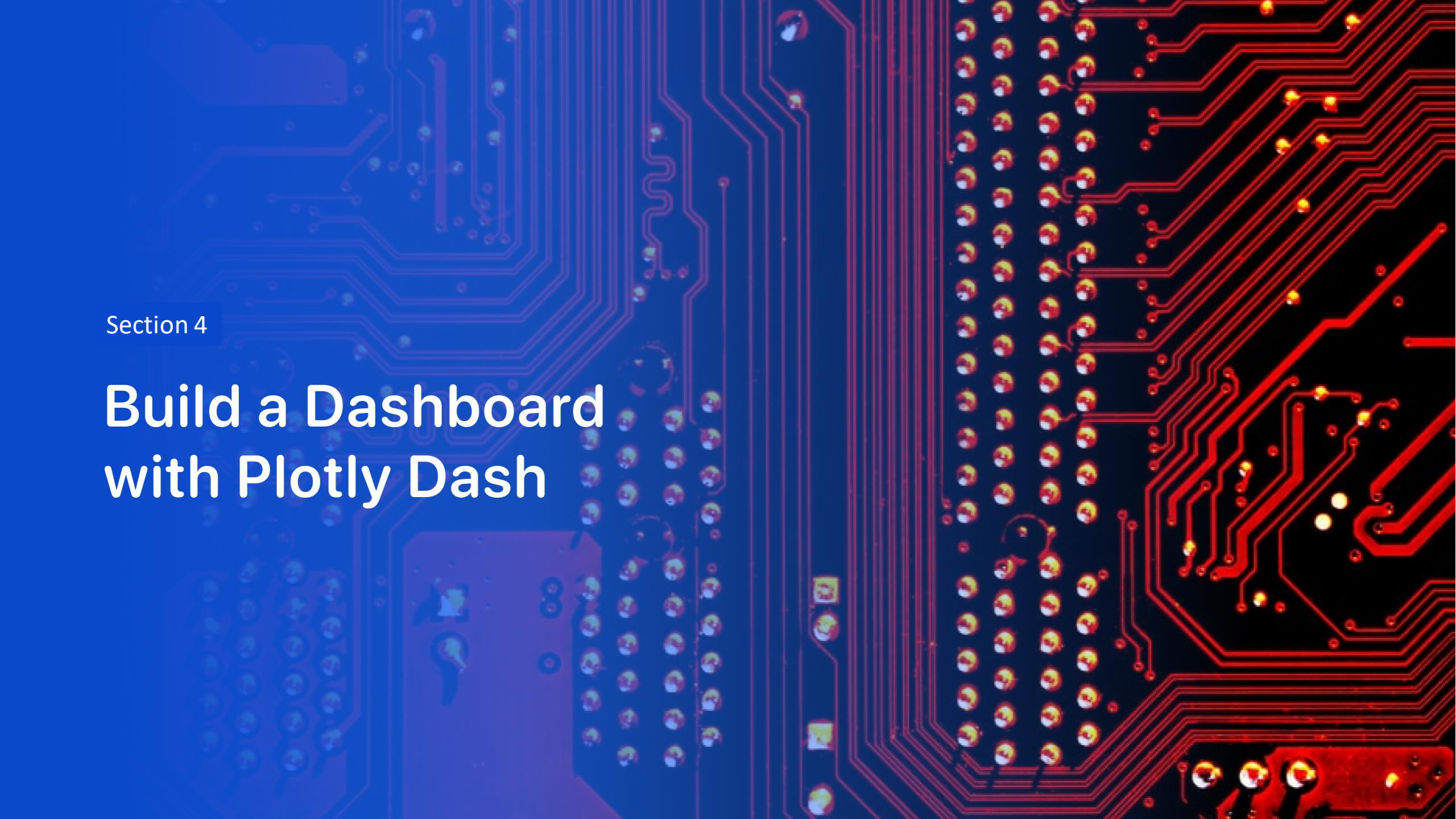




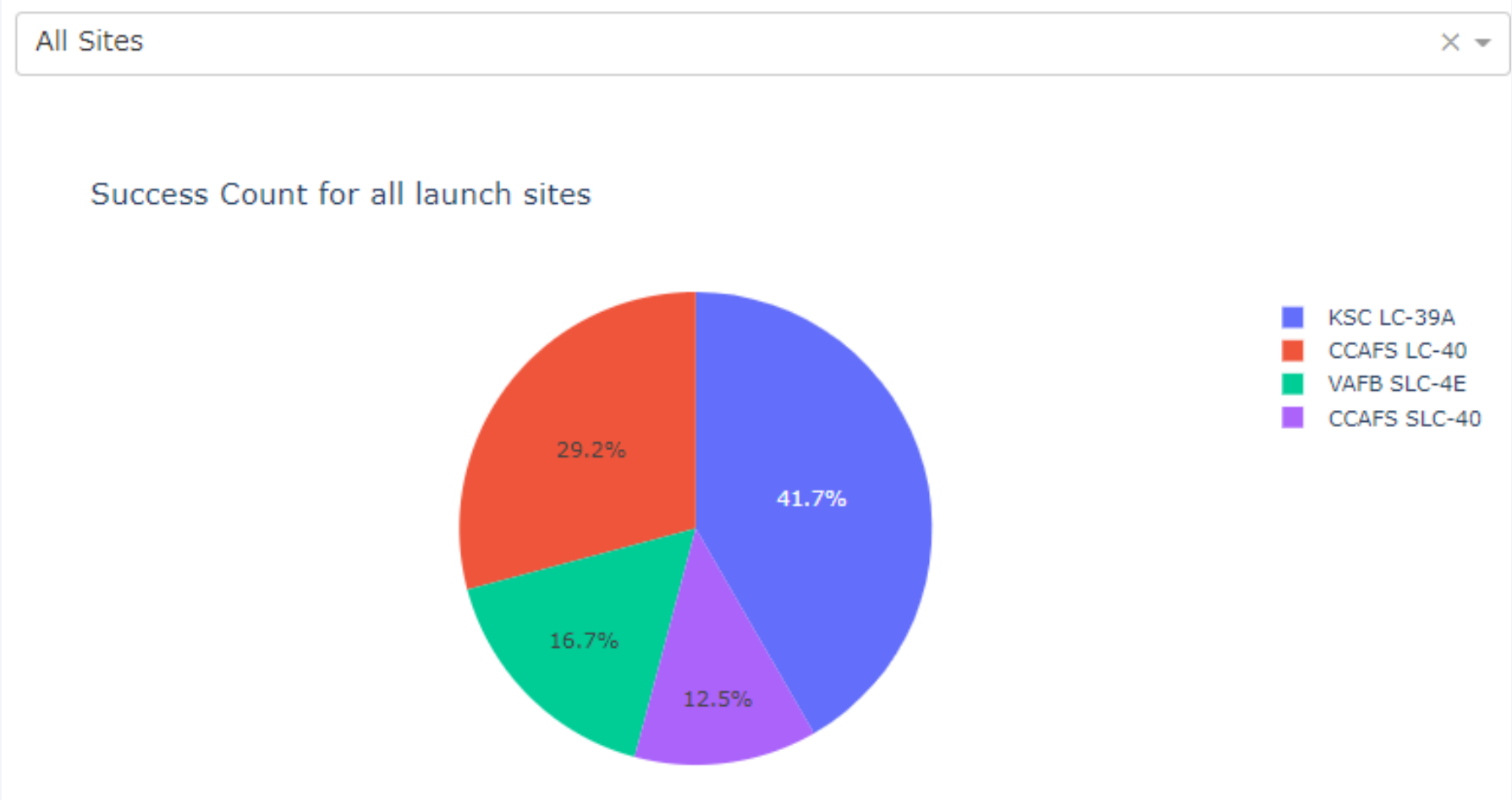
Launch Site to it proximities



The distance from the Cinderella Castle in Walt Disney World

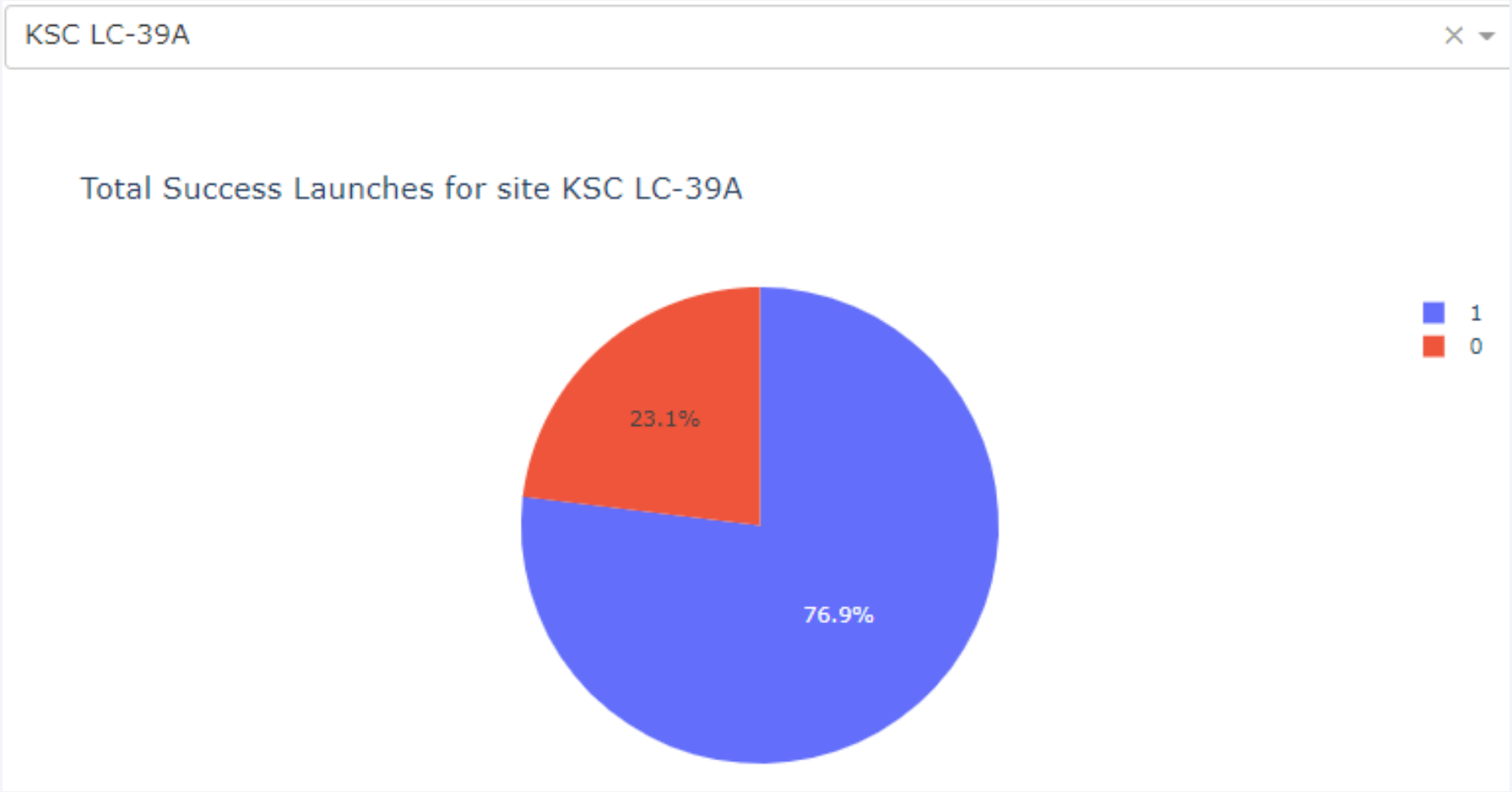

Section 4
Build a Dashboard with Plotly Dash

Dashboard : Success count for all launch site



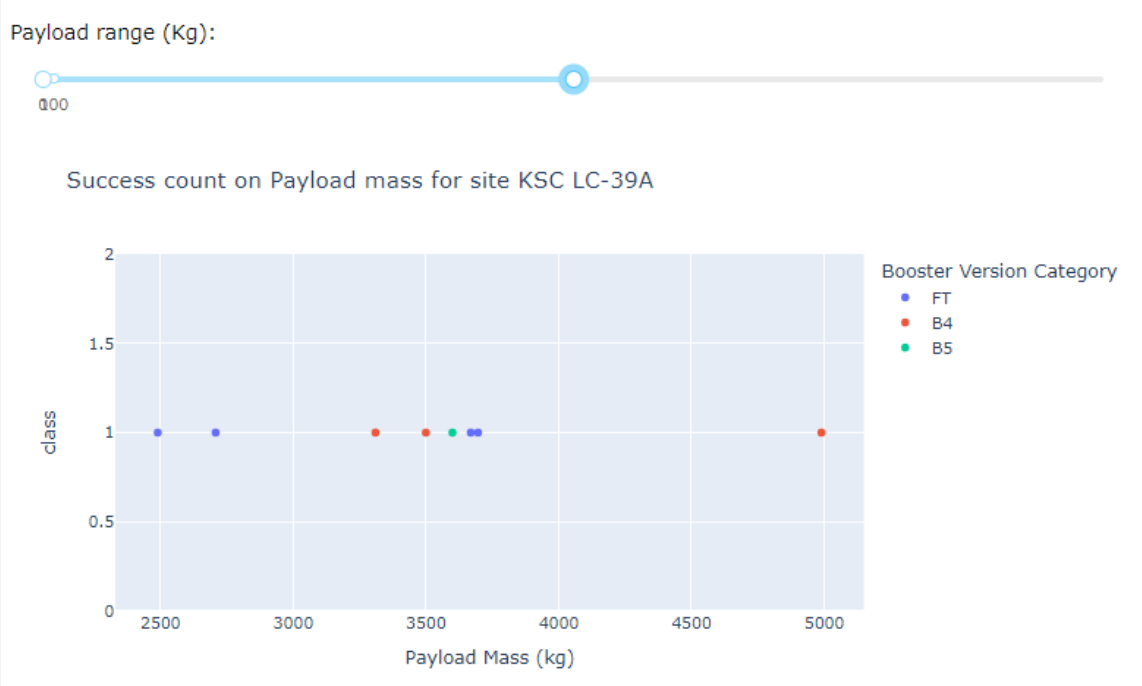
KSC LC-39A has the highest success rate and CCAFS SLC-40 has the lowest.

Dashboard : the launch site having the highest success rate

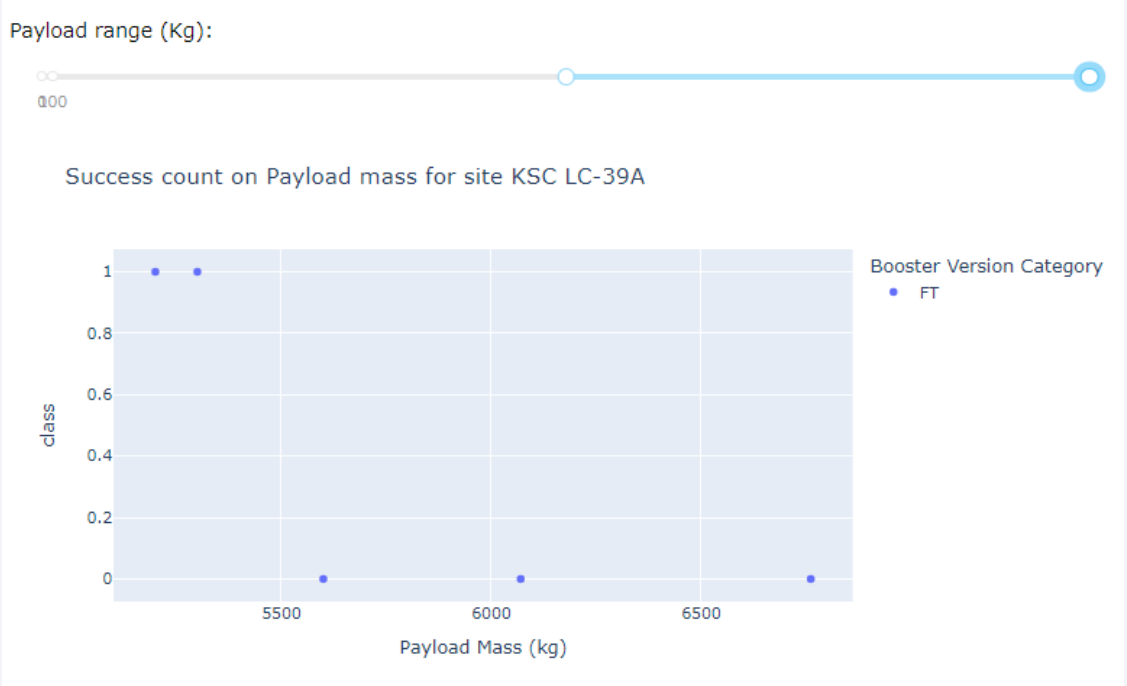


KSC LC-39A has the highest success rate and it is 76.9%

Dashboard : Payload Mass and Launch Outcomes



Range of Payload Mass : 0 kg ~ 5,000 kg



Range of Payload Mass : 6,000 kg ~ 10,000 kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

```
print('the score of knn_cv =',knn_cv.score(X_test, Y_test))
print('the score of svm_cv =',svm_cv.score(X_test, Y_test))
print('the score of logreg_cv =',logreg_cv.score(X_test, Y_test))
print('the score of tree_cv =',tree_cv.score(X_test, Y_test))
```

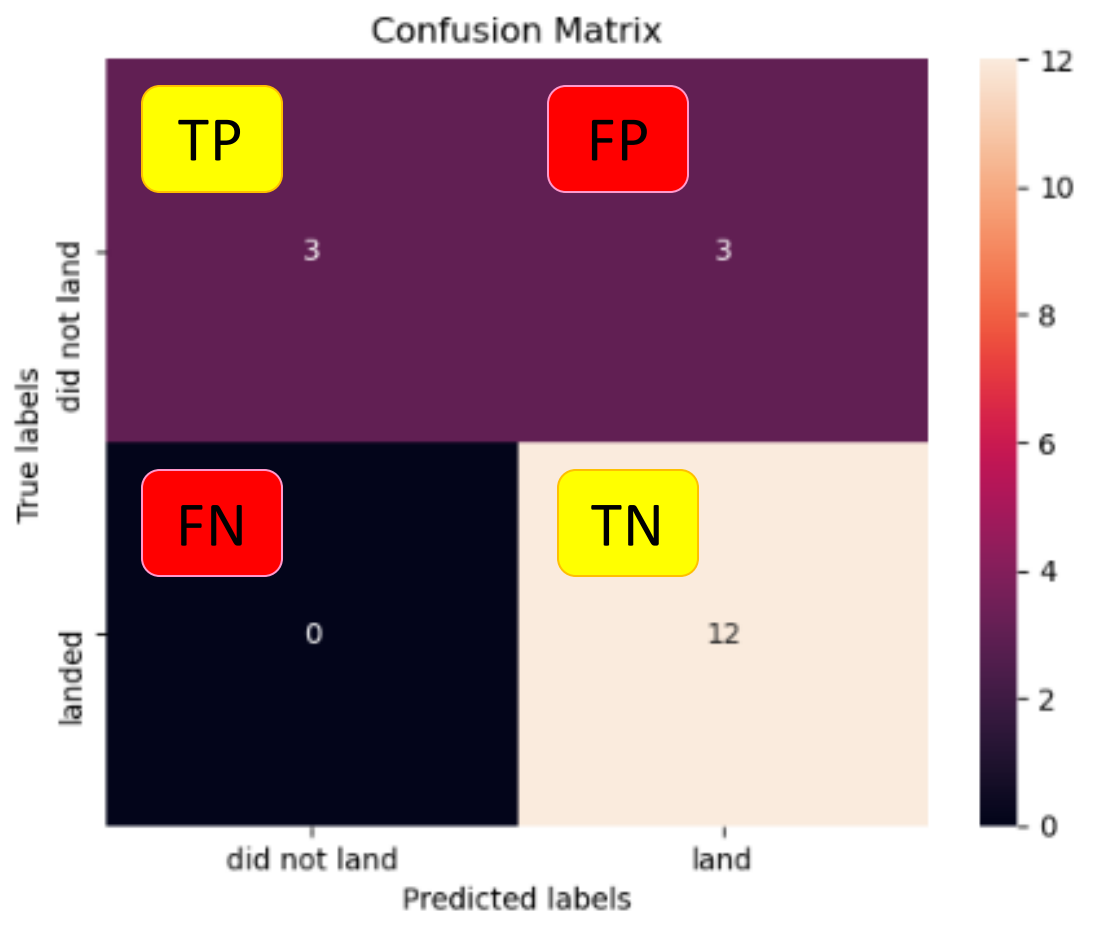
```
the score of knn_cv = 0.8333333333333334
the score of svm_cv = 0.8333333333333334
the score of logreg_cv = 0.8333333333333334
the score of tree_cv = 0.8333333333333334
```

All 4 machine leaning classification models

(logistic regression, support vector machine, a decision tree and k nearest neighbors)

have the same accuracy, 0.8333333333333334

Confusion Matrix



	Formula	Result
Sensitivity	TP/(TP+FN)	1.00
Specificity	TN/(TN+FP)	0.20
Accuracy	(TP+TN)/(TP+TN+FP+FN)	0.83
Negative Predictive Value	TN/(TN+FN)	1.00
Precision	TP/(TP+FP)	0.50

Conclusions

- Point1

  The Orbit, ES-L1, GEO, HEO and SSO, have 100% success rate.

  The Orbit, SO, has 0% success rate.

  The success rate is about 80% recently.

- Point 2

  A launch with heavier payload mass has higher success rate.

  A launch from KSC LC-39A has the highest success rate.

- Point 3

  All 4 machine leaning classification models (logistic regression, support vector machine, a decision tree and k nearest neighbors) have the same accuracy, 0.83333…..

Thank you!