IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pietro Cipolla
17 May 2023

Outline

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

Executive Summary

- Summary of methodologies
    - Data Collection through API
    - Data Collection with Web Scraping
    - Data Wrangling
    - Exploratory Data Analysis with SQL
    - Exploratory Data Analysis with Data Visualization
    - Interactive Map with Folium
    - Machine Learning Prediction

- Summary of all results
    - Exploratory Data Analysis results
    - Interactive maps and dashboard
    - Predictive Analytics result from Machine Learning Lab

Introduction

- Project background and context

  In this capstone, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  - Identifying all factors that influence the landing outcome.
  - The relationship between each variables and how it is affecting the outcome.
  - The best condition needed to increase the probability of successful landing.

Section 1

Methodology

Methodology

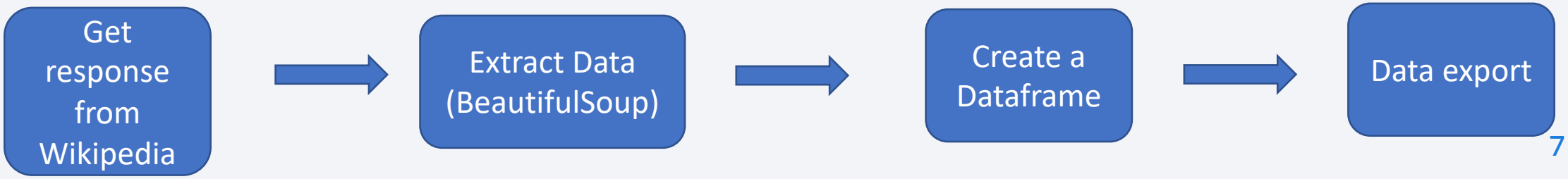
Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping (from Wikipedia)
- Perform data wrangling
  - Eliminate useless columns
  - Classification models with Hot Encoding
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Collected datasets from Rest SpaceX API, we obtain the following information: rocket, launches and payload information. SpaceX Rest API URL

- From the webscraping of Wikipedia page we obtain the following information: launches, landing, payload information. Wikipedia page



Data Collection – SpaceX API

1. GET response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Convert file into a JSON

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

3. Data transformation

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

4. Dictionary creation

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

5. Dataframe creation

```
data = pd.DataFrame.from_dict(launch_dict)
```

6. Dataframe filtering

```
data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7. Data export

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

[Data collection API Git Hub URL]

Data Collection - Scraping

1. GET response from HTML

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

2. BeautifulSoup object

```
soup = BeautifulSoup(response.text, "html5lib")
```

3. Extract tables from HTML

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

4. Extract columns name

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

5. Dictionary creation

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Dictionary filling

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7. Dataframe creation

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

8. Data export

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

[Web scraping Git Hub URL]

Data Wrangling

- In this process we calculate the number of launches on each site, calculate the number and the occurrence of each orbit, Calculate the number and occurence of mission outcome per orbit type, Create a landing outcome label from Outcome column and we export the data.

1. Number of launch

```
In [7]: # Apply value_counts() on column LaunchSite
        df['LaunchSite'].value_counts()

Out[7]: CCAFS SLC 40     55
        KSC LC 39A       22
        VAFB SLC 4E      13
        Name: LaunchSite, dtype: int64
```

2. Orbit number

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

3. Mission outcome

```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
landing_outcomes

True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: Outcome, dtype: int64
```

4. Label outcome creation

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []
for key, value in df['Outcome'].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

5. Data export

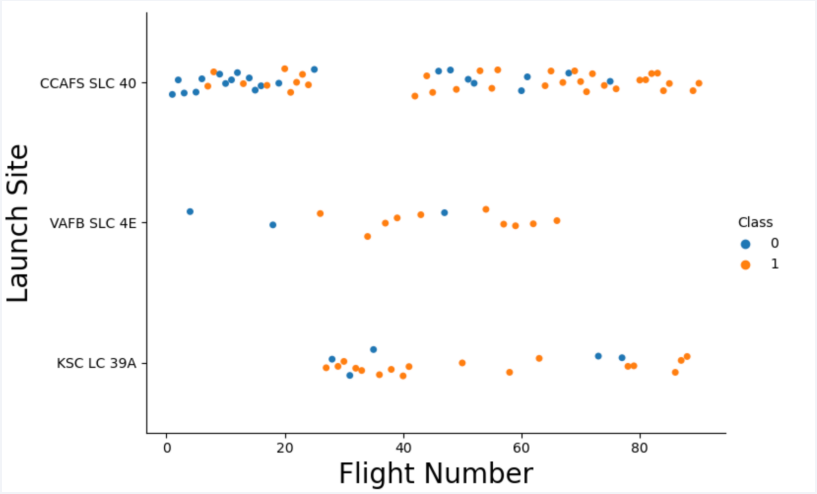
```
df.to_csv("dataset_part_2.csv", index=False)
```

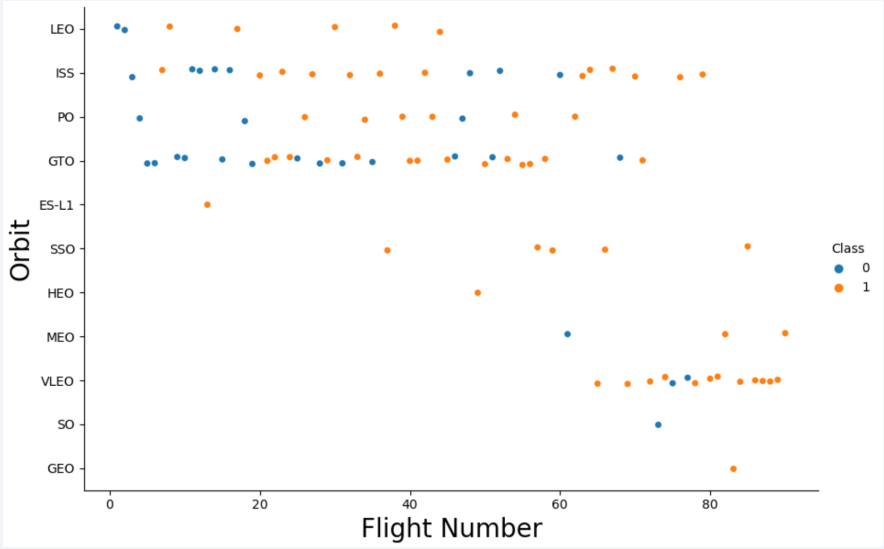
Data wrangling URL

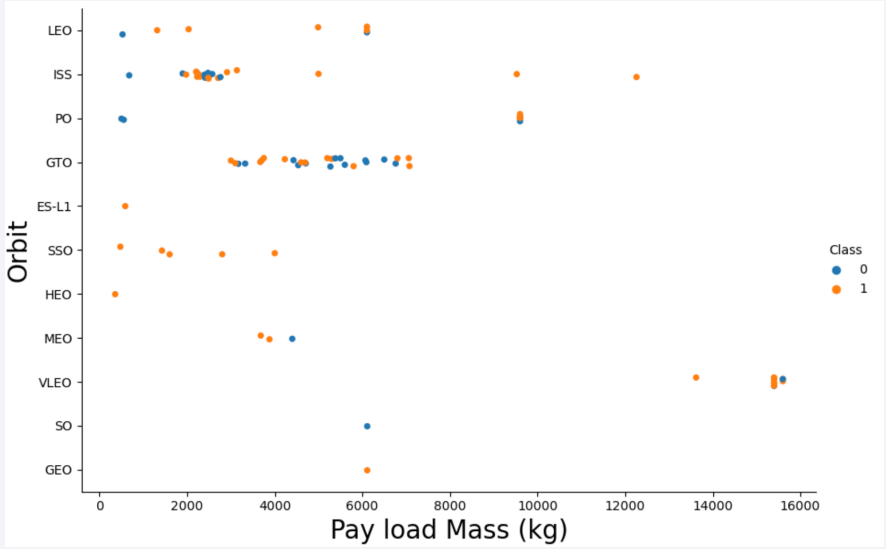
EDA with Data Visualization

We used the scatter plot to find relationship between:

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

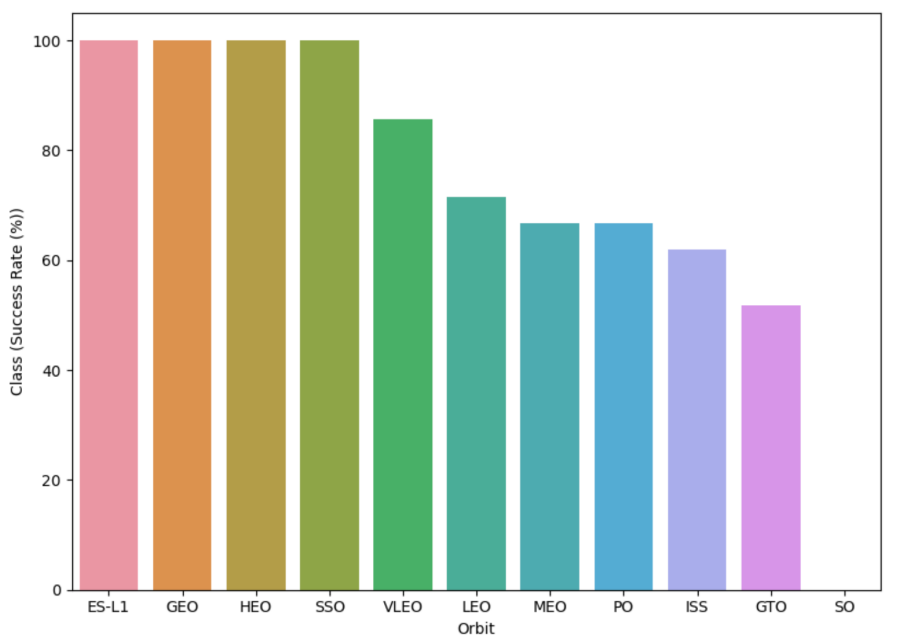


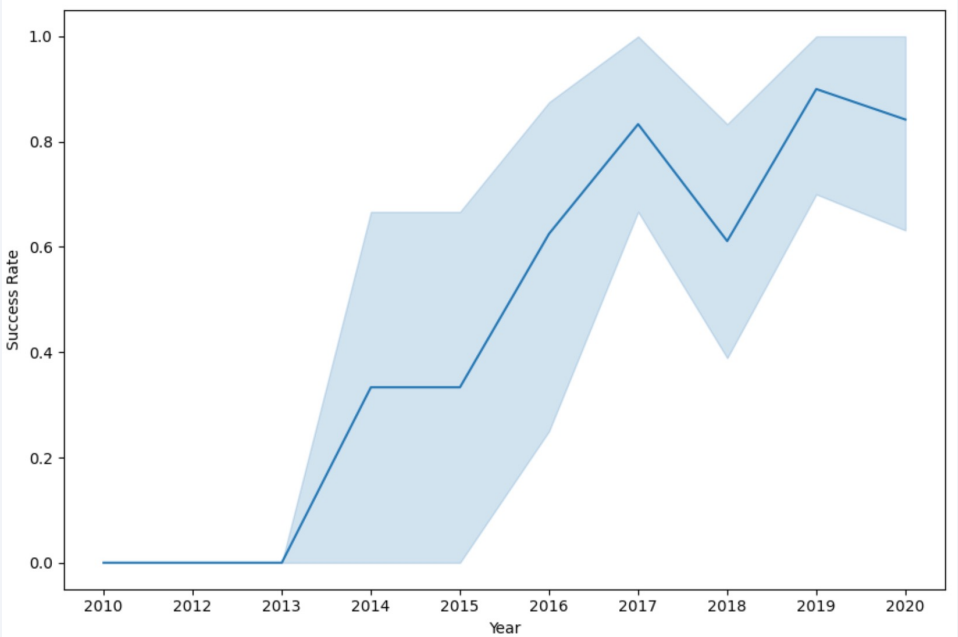




EDA with Data Visualization

Then we other visualization tools such as bar graph to determine which orbits have the highest probability of success, line plots graph to determine pattern of the attribute over time which in this case, is used for see the launch success yearly trend.





EDA Data Visualization URL

EDA with SQL

SQL queries:

- Names of the unique launch sites in the space mission.
- Top 5 launch sites whose name begin with the string 'CCA'.
- Total payload mass carried by boosters launched by NASA.
- Average payload mass carried by booster version F9 v1.1.
- Date of first successfull landing outcome in ground pad.
- Names of successfull boosters in drone ship with payload mass between 4k and 6k kg.
- Total number of successful and failure mission outcomes.
- Names of the booster versions which have carried the maximum payload mass.
- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- Rank of the count of landing outcomes between the date 2010-06-04 and 2017-03-20.

EDA with SQL URL

Build an Interactive Map with Folium

- Markers indicate points like launch sites;
- Circles indicate highlighted areas around specific coordinates, like NASA Johnson Space Center;
- Marker clusters indicates groups of events in each coordinate, like launches in a launch site; and
- Lines are used to indicate distances between two coordinates.

Interactive map URL

Build a Dashboard with Plotly Dash

We used the following graphs and plots to visualize data:

- Percentage of launches by site
- Payload range

This was usefull *to find the relations beetween launch sites and payloads and determine which was the best place as launch site.*

Plotly Dash URL

Predictive Analysis (Classification)

Data preparation:

- Load dataset
- Normalize data
- Split data: training/test sets.

Model preparation:

- Finding a machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Training GridSearchModel models with training dataset

Model evaluation:

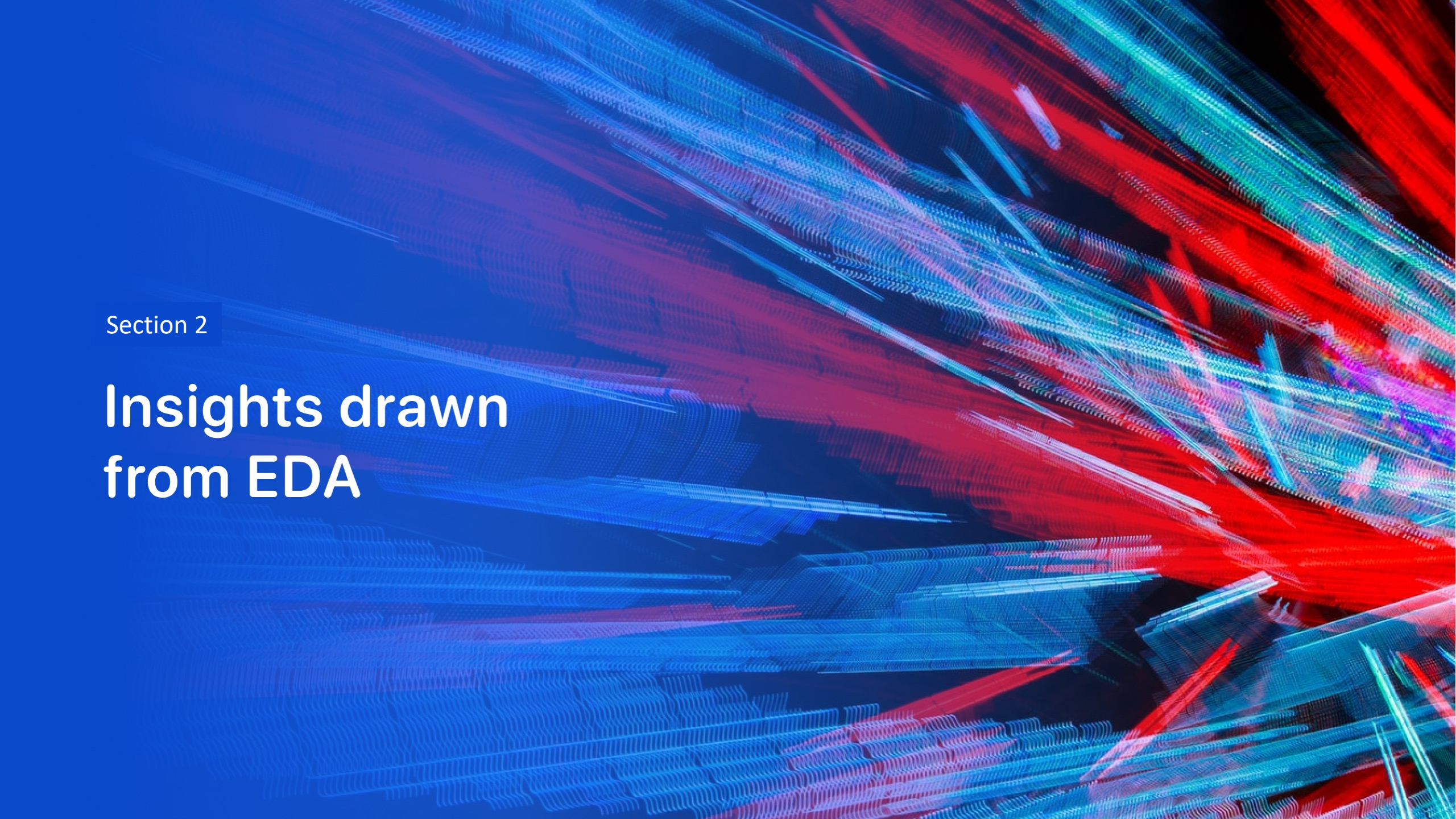
- Find hyperparameters for each type of model
- Compute accuracy for each model with test dataset
- Plot Confusion Matrix

Model comparison:

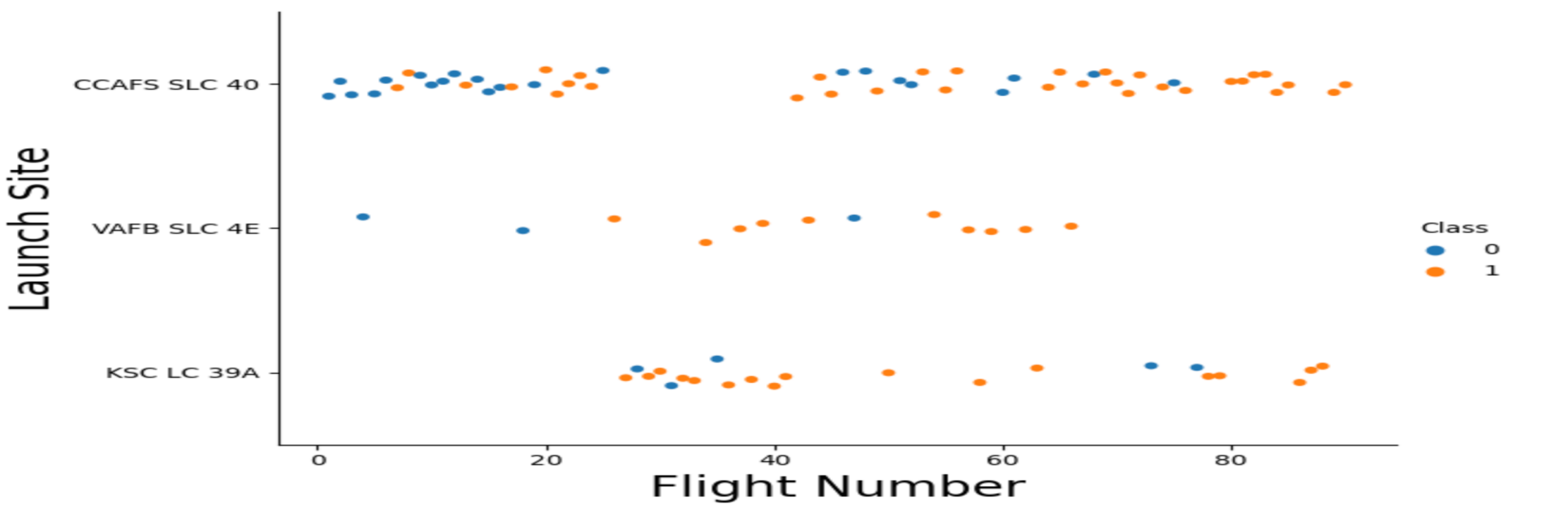
- Comparison of models according to their accuracy
- Find the best accuracy model

Results

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

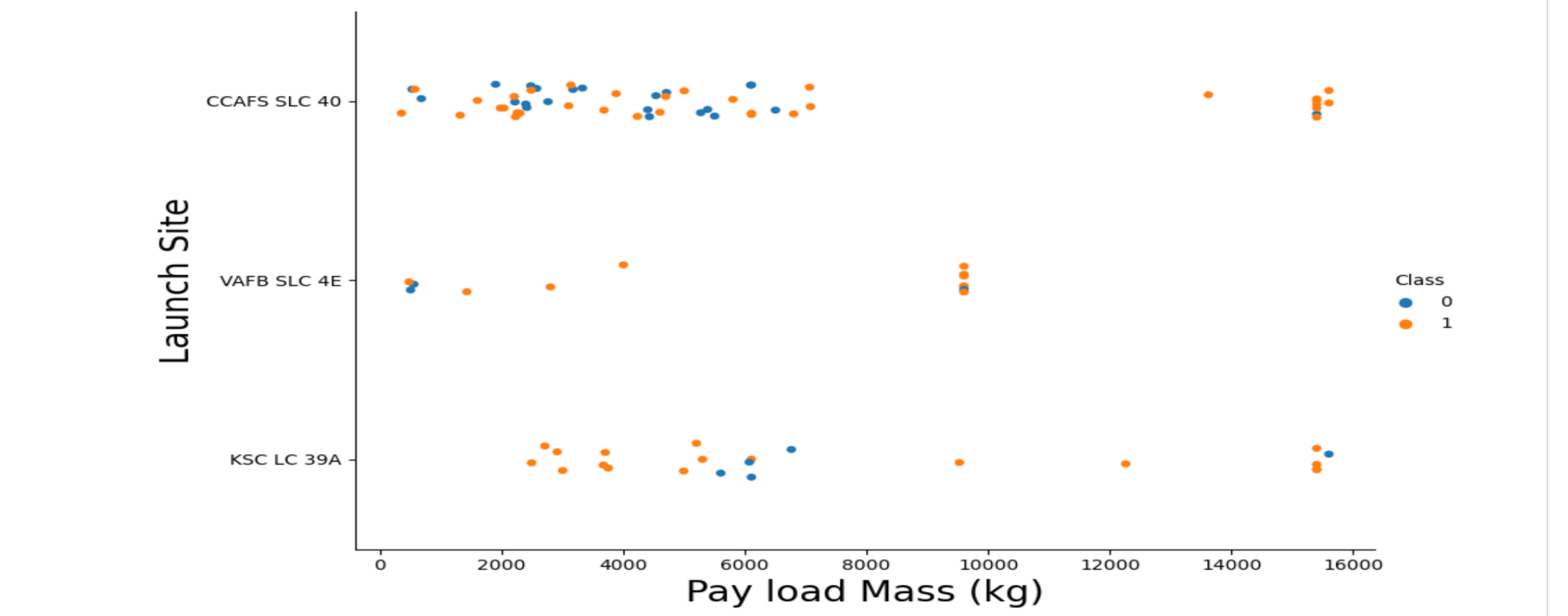

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



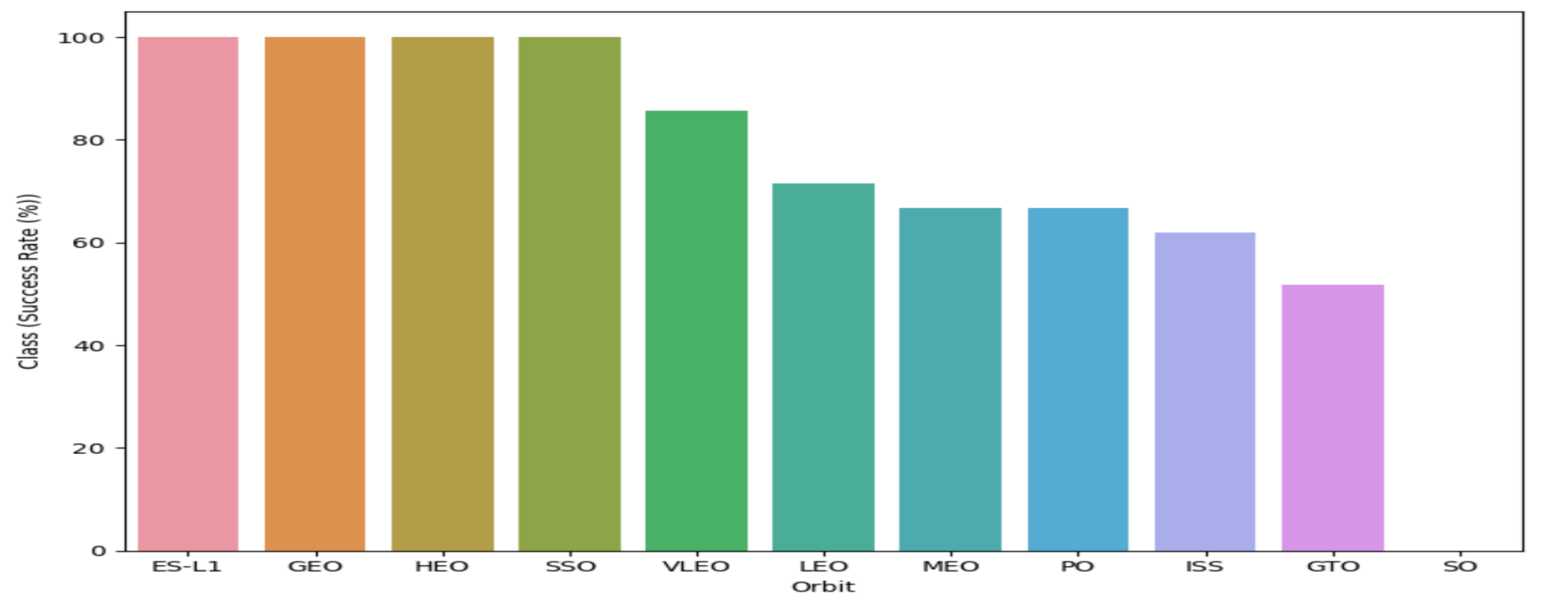
The success rate is increasing for each site

Payload vs. Launch Site



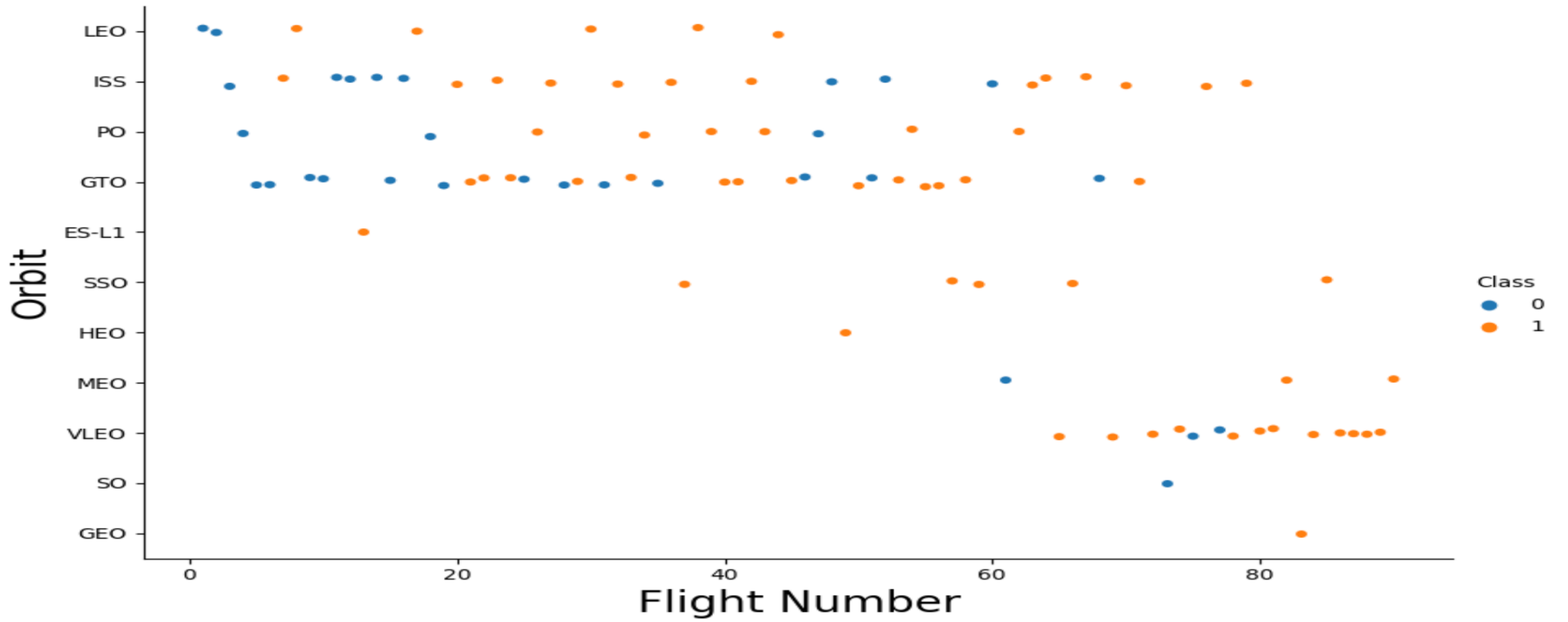
A heavier payload usually has more successful landing

Success Rate vs. Orbit Type

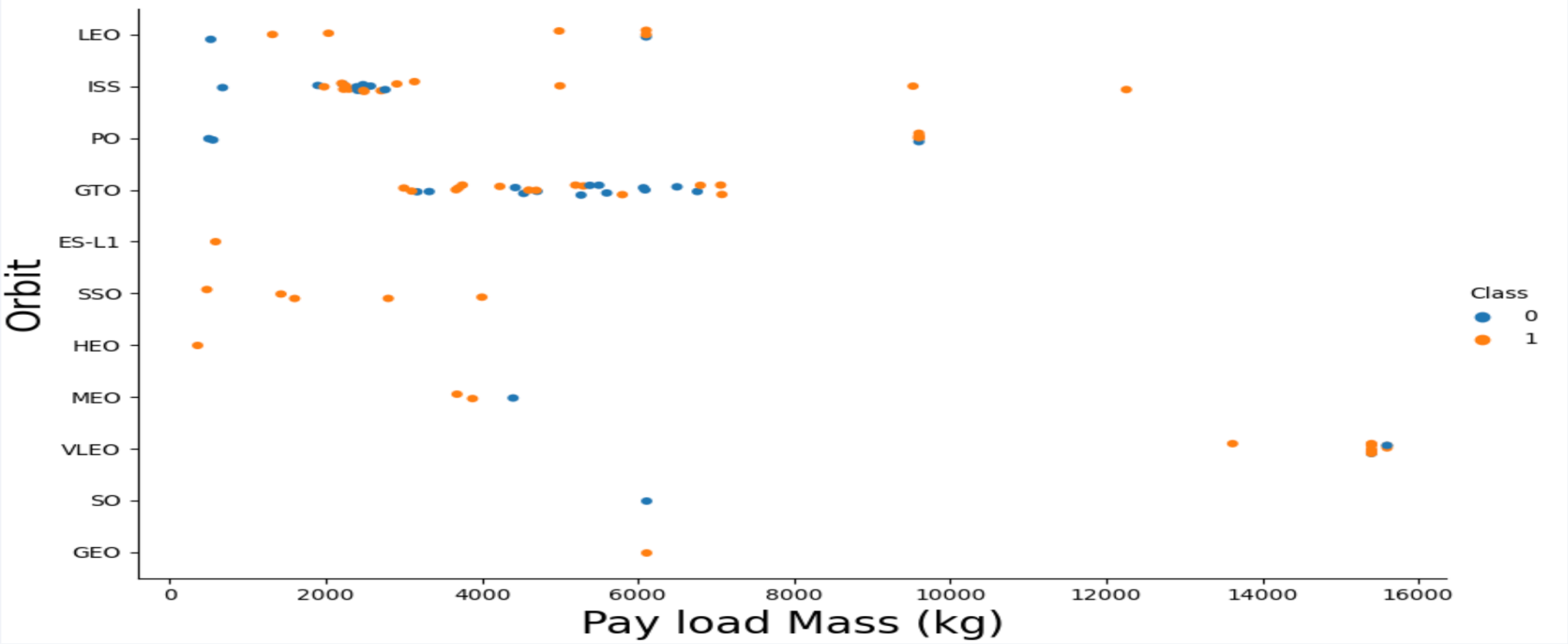


Best success rate: ES-L1, GEO, HEO, SSO

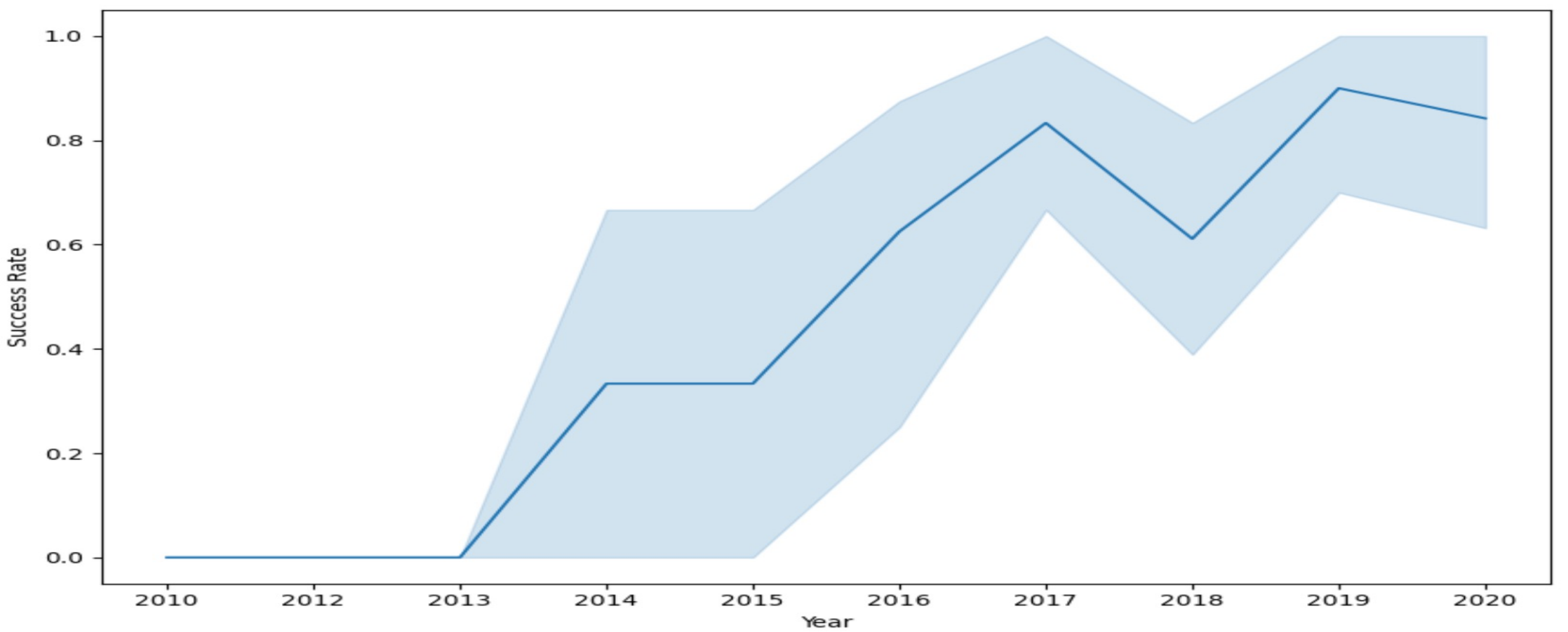
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



The success rate keep increasing since 2013.

All Launch Site Names

```
In [7]: sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1;
 * sqlite:///my_data1.db
Done.
```

Out[7]:

Launch_Site
None
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

The SELECT command allows to eliminate the duplicate Launch site.

Launch Site Names Begin with 'CCA'

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

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Lan
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Fai
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Fai
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

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

In [9]:
```
sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXTBL WHERE PAYLOAD  LIKE '%CRS%';
```

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

Out[9]:

TOTAL_PAYLOAD
111268.0

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 AVG_PAYLOAD FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

AVG_PAYLOAD
2928.4

First Successful Ground Landing Date

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

Hint:Use min function

In [12]:
```
sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXTBL WHERE LANDING_OUTCOME = 'Success (ground pad)';
```

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

Out[12]:

FIRST_SUCCESS_GP
01/08/2018

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 DISTINCT BOOSTER_VERSION FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING_OUTCOME = 'Success (ground pad)';
```

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

Booster_Version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXTBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

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

Mission_Outcome	QTY
None	898
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

```
%%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
FROM SPACEXTBL)
ORDER BY BOOSTER_VERSION;
```

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

Booster_Version
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

```
%%sql SELECT SUBSTR(Date, 4, 2), DATE, BOOSTER_VERSION, LAUNCH_SITE, LANDING_OUTCOME FROM SPACEXTBL
      WHERE LANDING_OUTCOME = 'Failure (drone ship)' AND SUBSTR(DATE,7,4)='2015';
```

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

SUBSTR(Date, 4, 2)	Date	Booster_Version	Launch_Site	Landing_Outcome
10	01/10/2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	14/04/2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

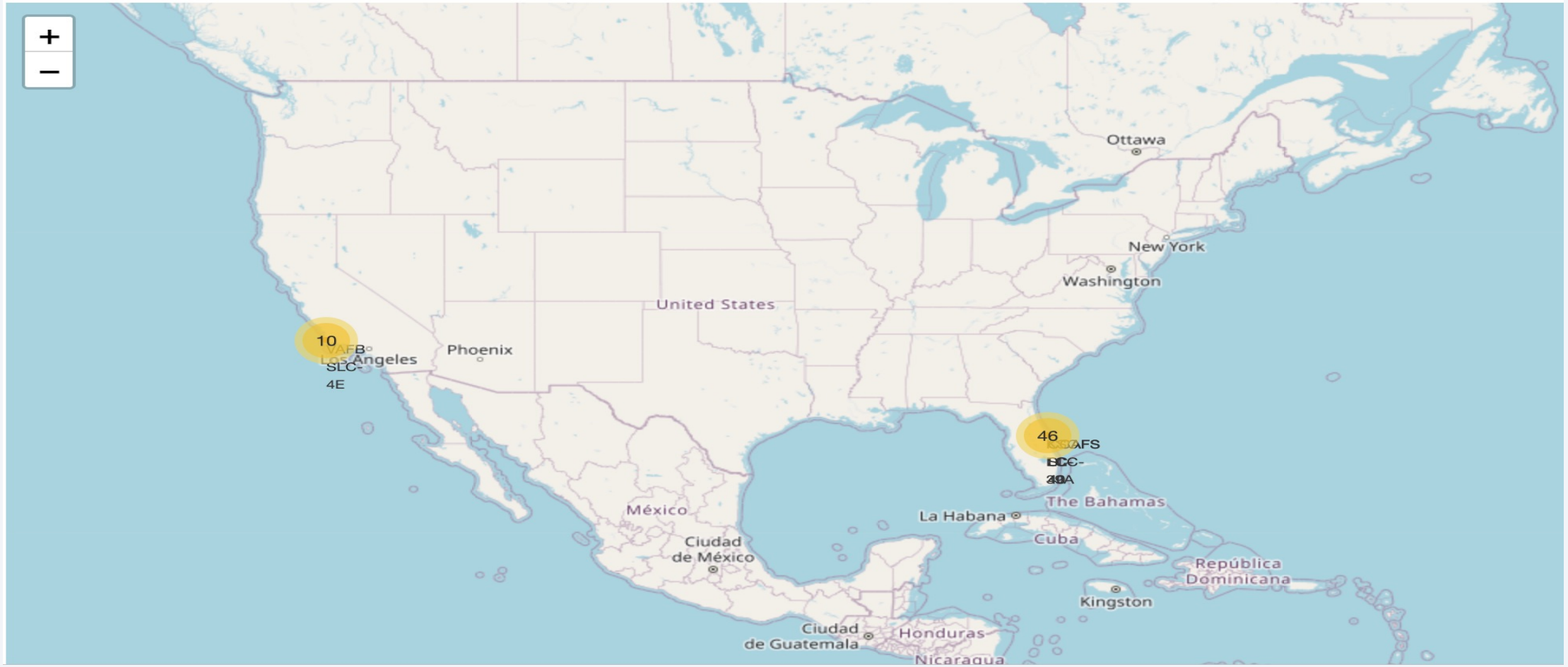
```
%%sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_OUTCOMES FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME ORDER BY COUNT_OUTCOMES DESC;
```

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

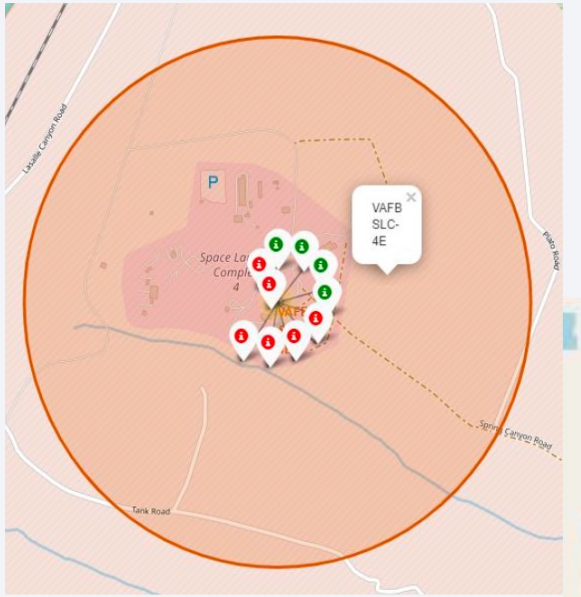
Landing_Outcome	COUNT_OUTCOMES

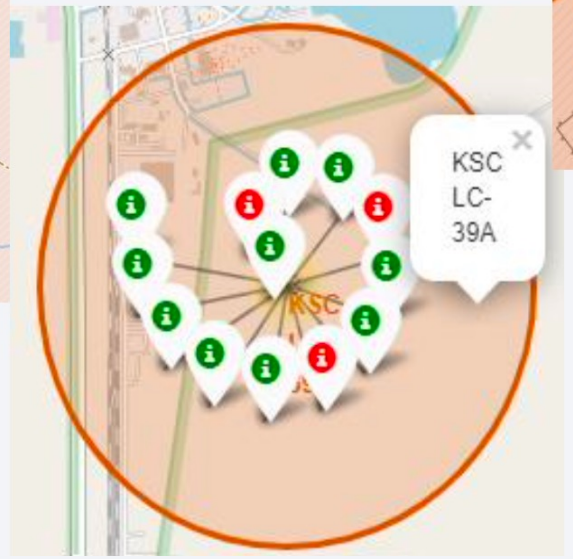

Section 3
Launch Sites
Proximities Analysis

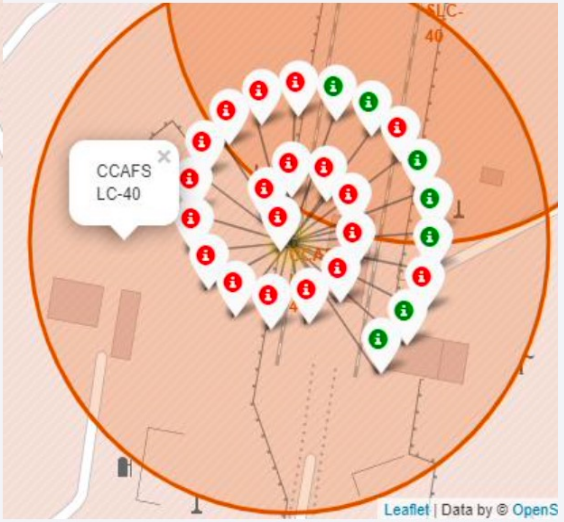
Launch sites



Detailed Launch sites outcomes

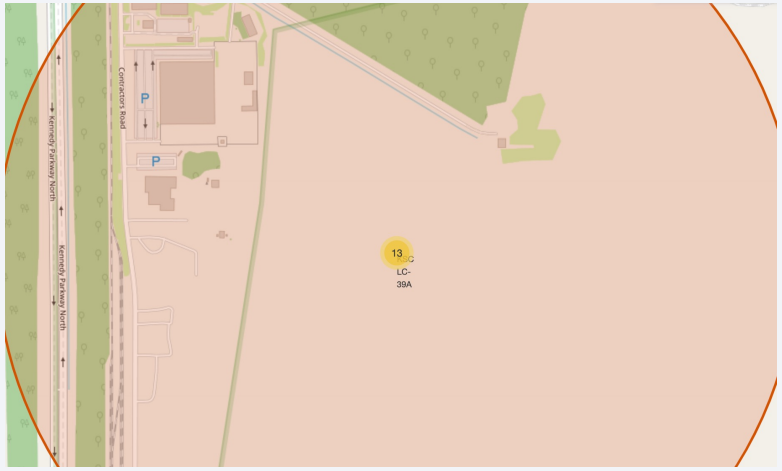


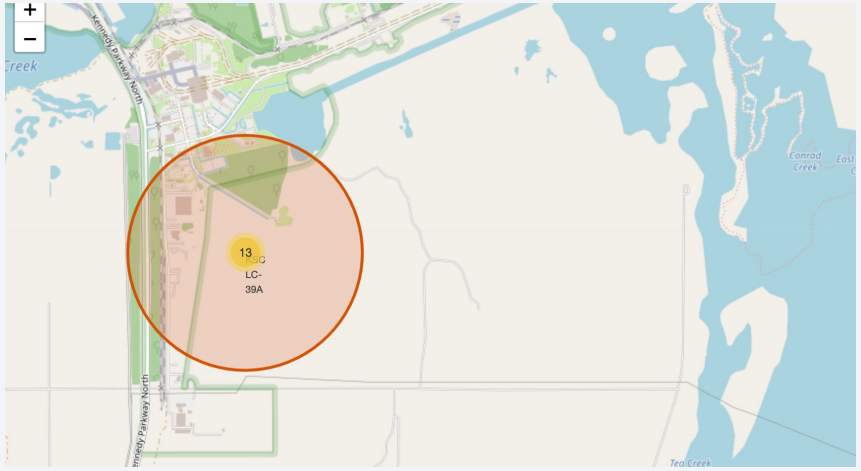




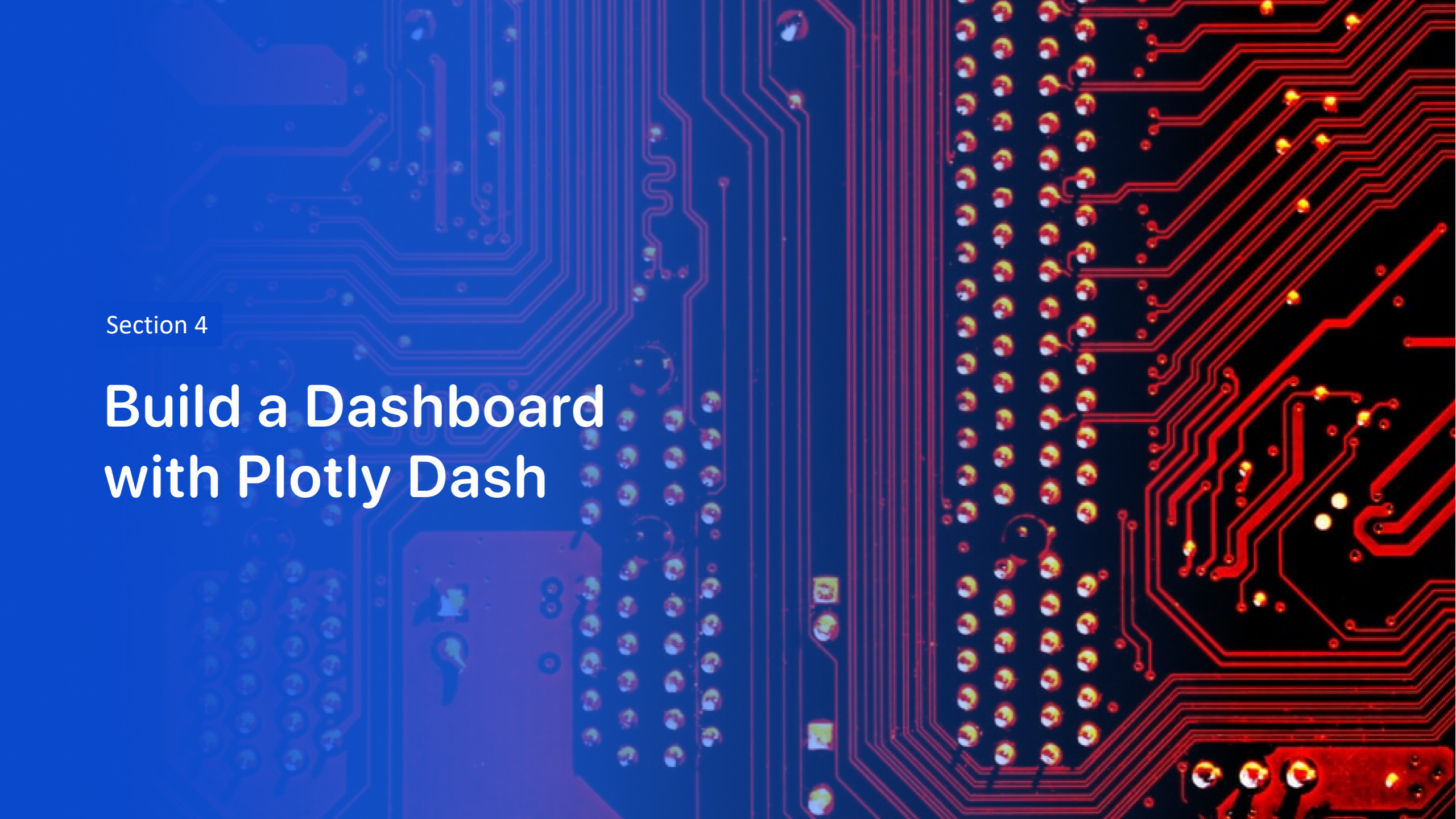
Green markers = successful launches
Red markers = failed launches

KSC LC-39A Launch site details

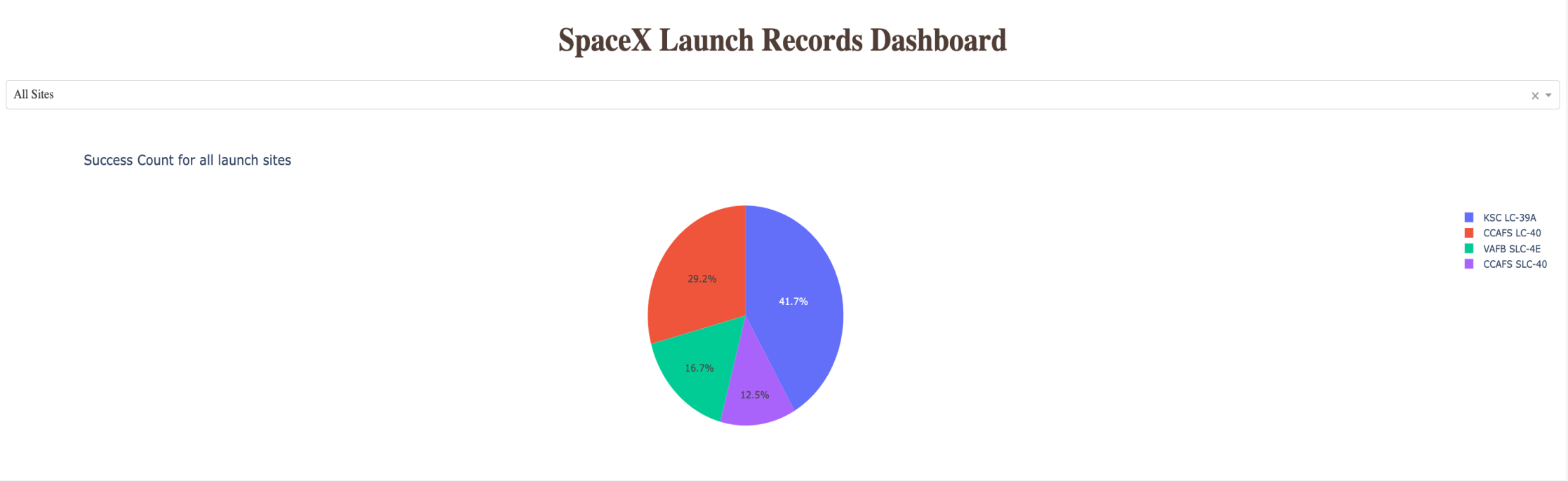




- Are launch sites in close proximity to railways? Yes
- Are launch sites in close proximity to highways? Yes
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? No

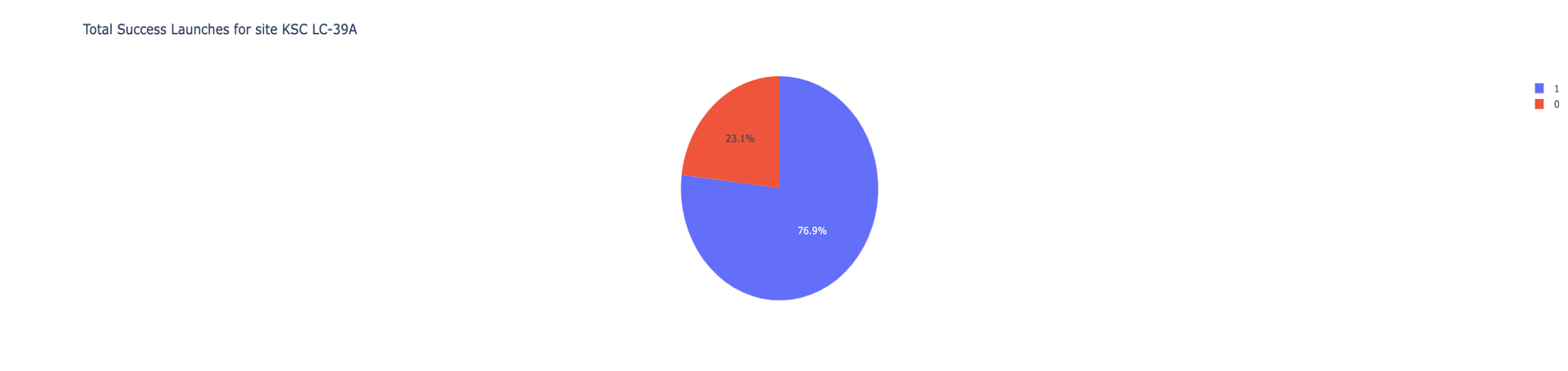

Section 4
Build a Dashboard with Plotly Dash

Total success for each site



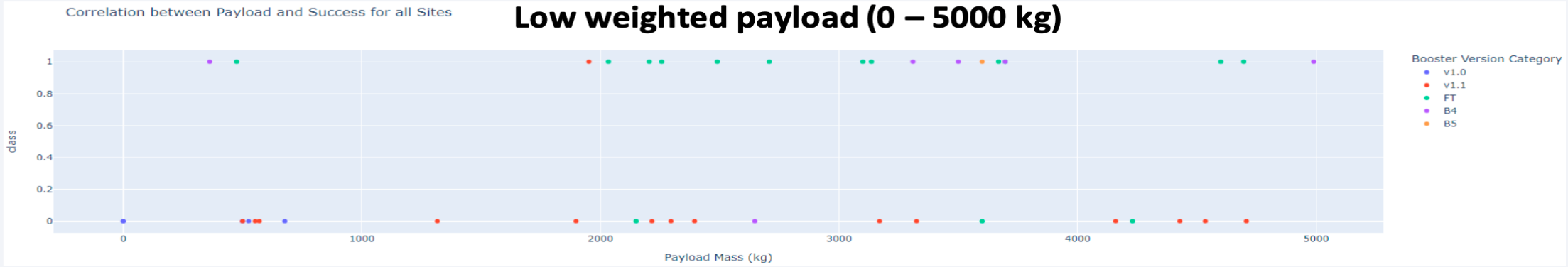
KSC LC-39A has the best success rate

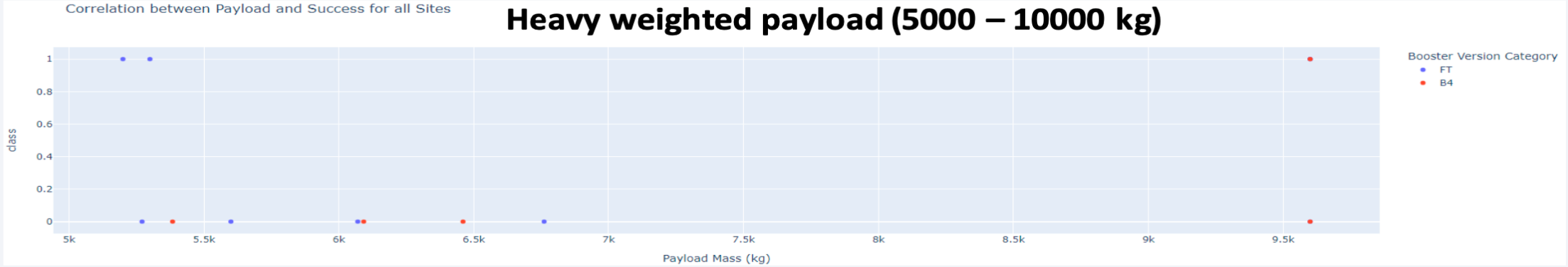
KSC LC-39A



- KSC LC-39A has 76.9% success rate

Payload and launch outcomes scatter plot



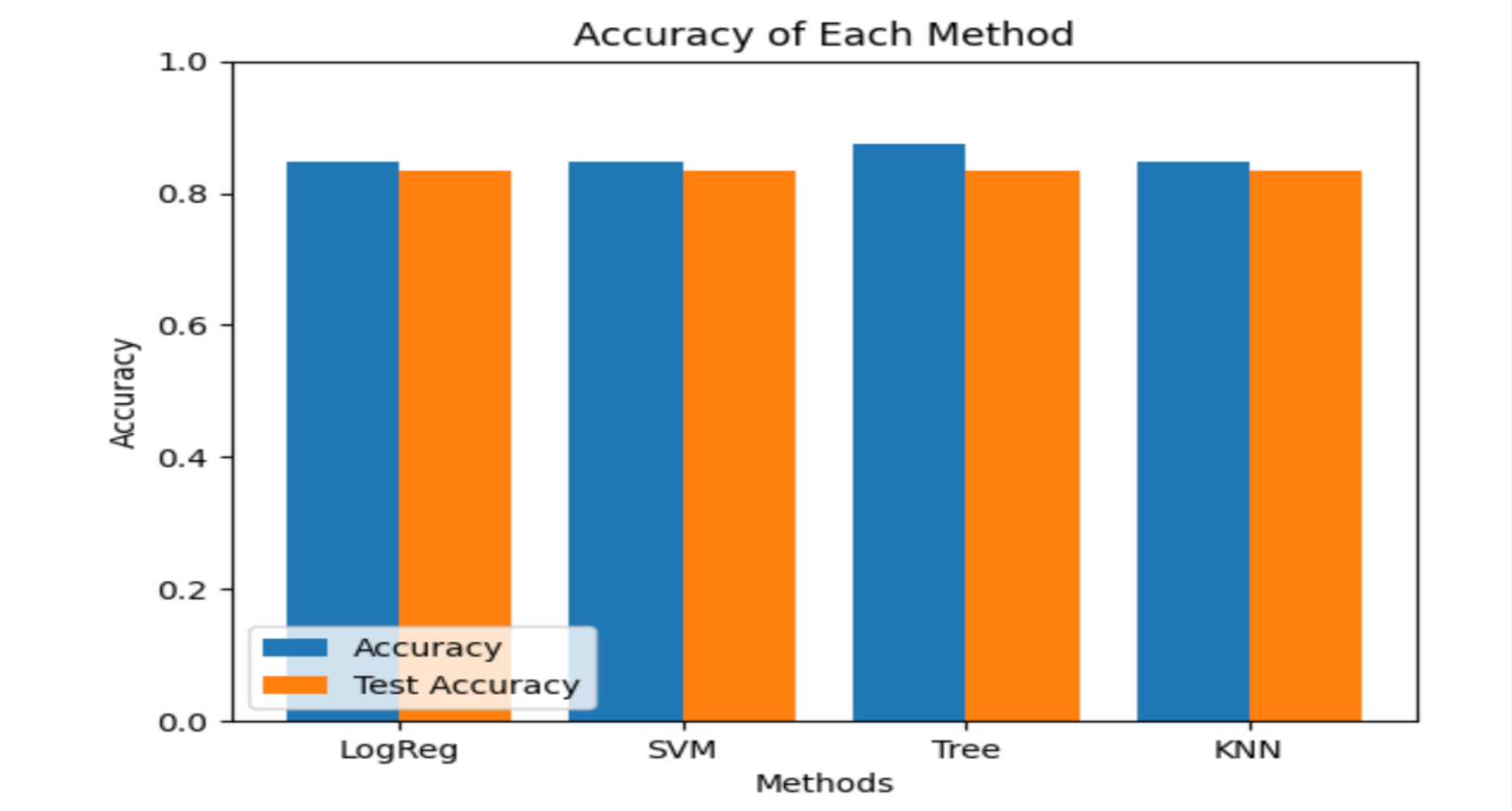


The success rate for low weighted payload is higher than heavy weighted payload

Section 5

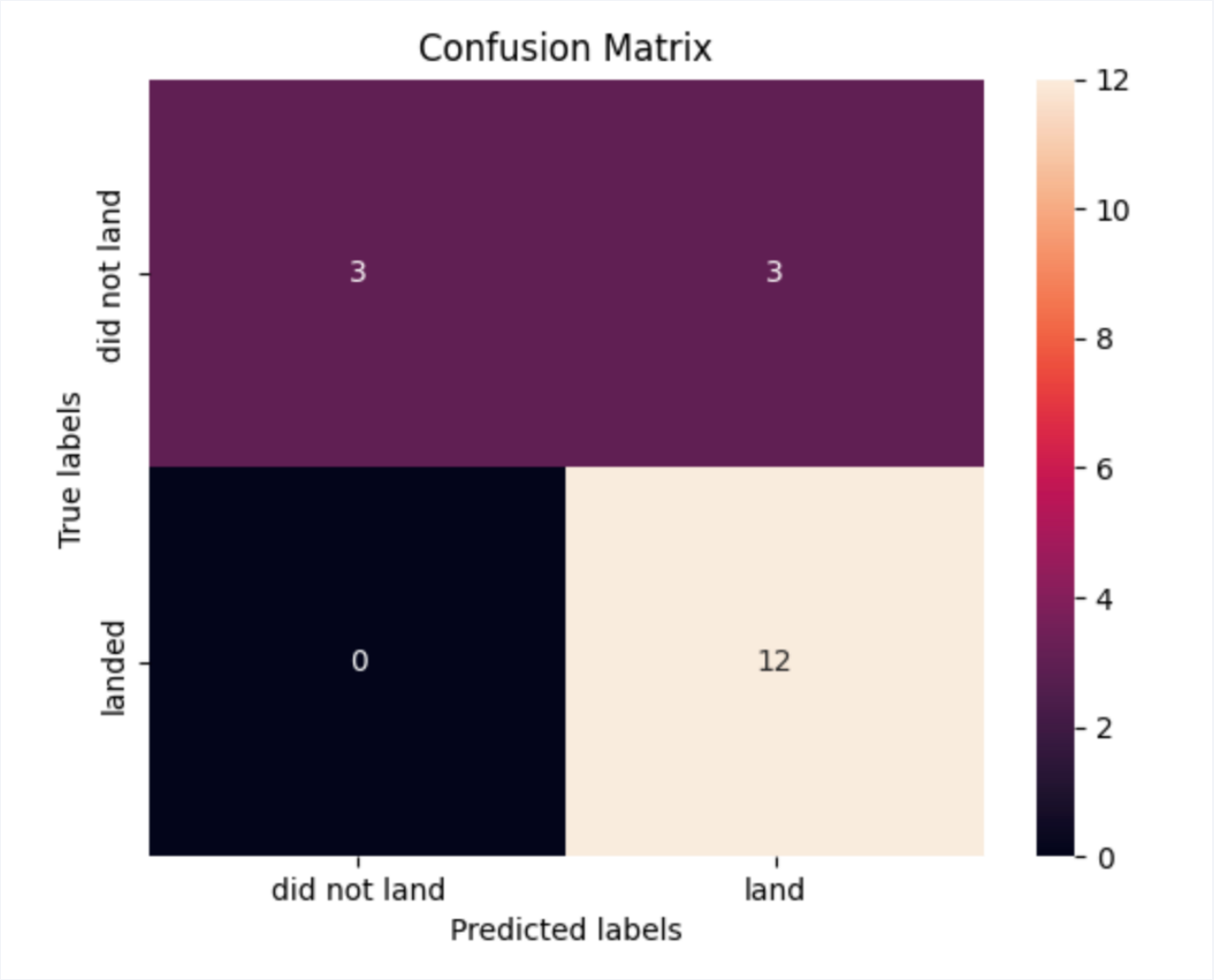
Predictive Analysis (Classification)

Classification Accuracy



The best classification model is decision Tree model with an accuracy of 87%

Confusion Matrix



Conclusions

- The best launch site is KSC LC-39A
- The success rate is increasing since 2013
- Decision Tree classifier is the most accurate model for this prediction

Thank you!