IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Richeng Piao
Sunday, November 7, 2021

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection
  - Data wrangling
  - EDA with data visualization
  - EDA with SQL
  - Building an interactive map with Folium
  - Building a Dashboard with Plotly Dash
  - Predictive analysis (Classification)
- Summary of all results
  - Exploratory data analysis results
  - Interactive analytics demo in screenshots
  - Predictive analysis results

Introduction

- **Project background and context**

This project is intended to 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 I can determine if the first stage will land, I 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**

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Rest API
  - Web scrapping from Wikipedia
- Perform data wrangling
  - Wrangling data using an API / Sampling Data / Dealing with Nulls
- 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

- Describe how data sets were collected.
  - Source 1: SpaceX API
    - API includes **launch data,** for example, rocket information, payload delivered, launch specifications, landing specifications, and landing outcomes.
    - The **goal** is to use this data to predict whether SpaceX will attempt to land a rocket or not.
    - url → *requests* → *response.json()* → *json_normalize*
  - Source 2: Web Scaping
    - Web scaping similar launch data from related Wiki pages.
    - *BeautifulSoup* → *Panda dataframe*
- You need to present your data collection process use key phrases and flowcharts

SpaceX API: Request and parse the launch data using the GET request → Decode the response content as a Json and turn it into a Panda dataframe. → Filter the dataframe and dealing with missing values.

Web Scraping: Request the Falcon9 Launch Wiki page from its URL → Extract all variable names from the HTML table header → Create a dataframe by parsing the launch HTML tables

Data Collection – SpaceX API

1. **Request and parse the SpaceX launch data using the GET request.**

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

2. **Decode the response content as a Json and turn it into a Panda dataframe.**

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

3. **Apply Custon functions to clean data**

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

4. **Assign list to dictinary then dataframe**

```
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. **Filter the dataframe to only include Falcon 9 launches**

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

6. **Dealing with Missing Values.**

```
payloadmass_mean = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].fillna(payloadmass_mean, inplace=True)
```

>>> GitHub Link <<<

Data Collection - Scraping

1. **Request the Falcon9 Launch Wiki page from its URL, and create a BeautifulSoup object.**

```
url="https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches"
response = requests.get(url)
soup = BeautifulSoup(response_url.text, 'html.parser')
```

2. **Extract all variable names from the HTML table header**

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

3. **Create a dictionary**

```
column_names = []
for i in range(0,len(first_launch_table.find_all('th'))):
   name = extract_column_from_header(first_launch_table.find_all('th')[i])
   if name != None and if len(name) > 0:
         column_names.append(name)
```

4. **Parsing the launch HTML tables**

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
                                                    … …
                booster_landing = landing_status(row[8])
                launch_dict["Booster landing"].append(booster_landing)
```

3. **Save to dataframe**

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

```
launch_dict= dict.fromkeys(column_names)

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'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

>>> GitHub Link <<<

Data Wrangling

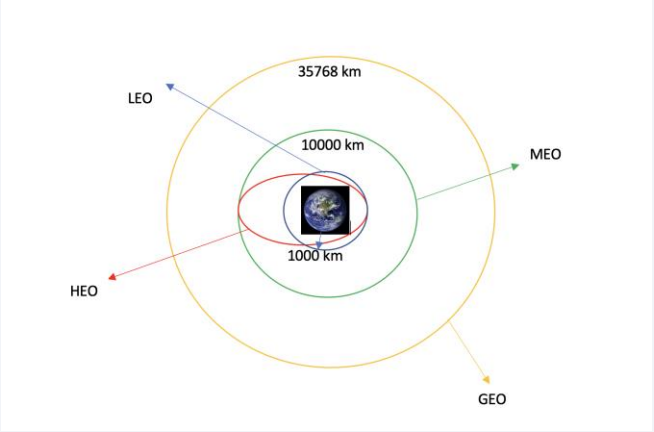
Process:

Perform some Exploratory Data Analysis (EDA) to find some patterns in the data → Calculate the number of launches on each site → Calculate the number and occurrence of each orbit → Calculate the number and occurrence of mission outcome per orbit type → Create a landing outcome label from Outcome column → Determine the success rate.

Example of successful landing?

- True (Class = 1)
  - E.g., ***True Ocean*** means the mission outcome was successfully landed to a specific region of the ocean
- False (Class = 0)
  - E.g., ***False RTLS*** means the mission outcome was unsuccessfully landed to a ground pad.

Examples of orbit types:



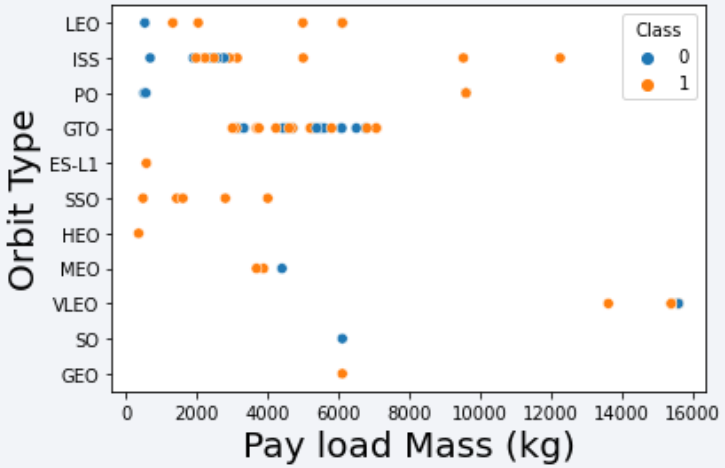
>>> GitHub Link <<<

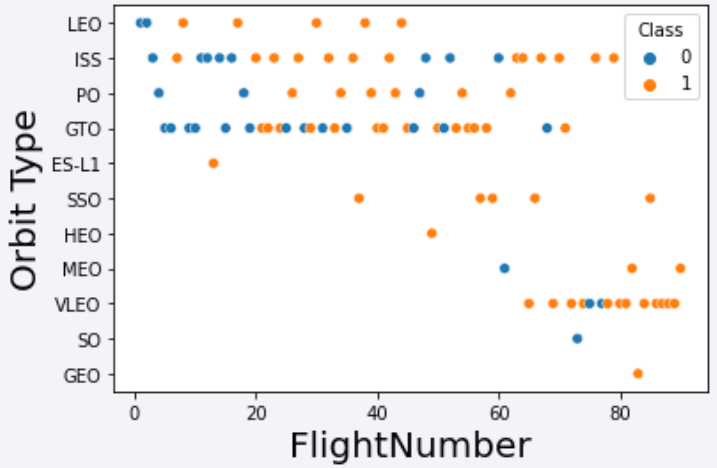
EDA with Data Visualization

Scatter Plots:

- Flight Number vs. Pay load Mass
- Flight Number vs Launch Site
- Pay load Mass vs Launch Site
- Flight Number vs Orbit type
- Pay load Mass vs Orbit type

** Understand relationship between variables.*

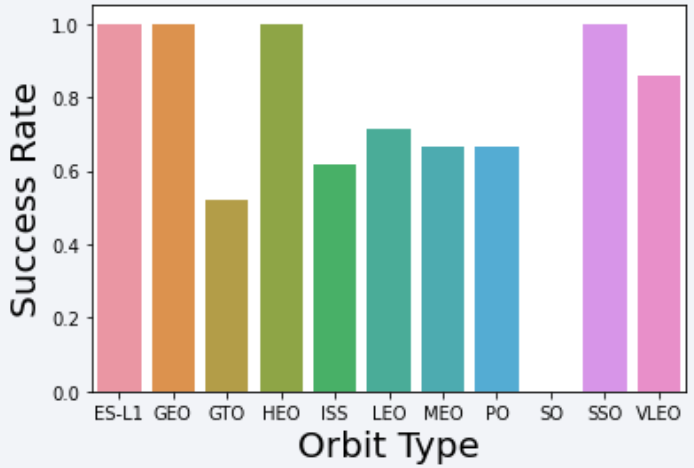




Bar Charts:

- Mean success rate vs Orbit type.

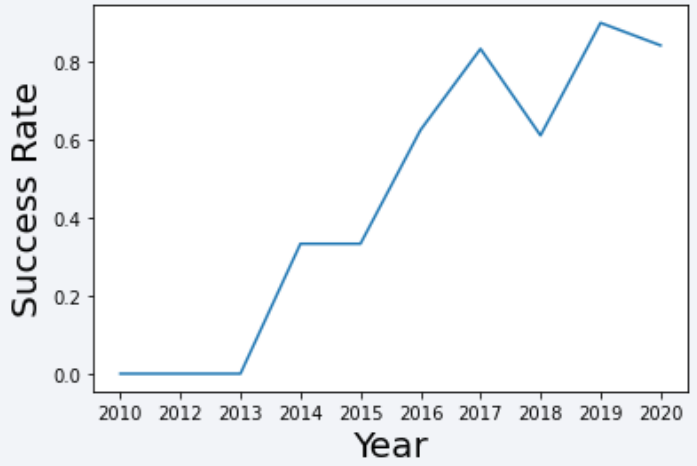
** Understand relationship between success rate and orbit type.*



Line Graph:

- Launch success yearly trend

** Understand yearly trend of success rate.*



>>> GitHub Link <<<

EDA with SQL

SQL queries:

1. Display the names of the unique launch sites in the space mission.
2. Display 5 records where launch sites begin with the string 'CCA’.
3. Display the total payload mass carried by boosters launched by NASA (CRS).
4. Display average payload mass carried by booster version F9 v1.1.
5. List the date when the first successful landing outcome in ground pad was acheived.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
7. List the total number of successful and failure mission outcomes.
8. List the names of the booster_versions which have carried the maximum payload mass. Use a subquery.
9. List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
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.

>>> GitHub Link <<<

Build an Interactive Map with Folium

- Circle Marker: to visualize the launch site data into interactive map.
  - Added each site's location on a map using site's latitude and longitude coordinates
- Market Cluster: to add the launch outcome for each site.
  - Use marker cluster simplify the map by grouping same coordinate.
  - Successful = green marker or Failed = red marker
- Lines: to explore and analyze landmarks of launch sites.
  - Calculate the distance between site and its landmarks.
  - Draw a Polyline between a launch site to the selected landmark(i.e., railway, highway, coastline, etc.).
  - Use marker to display distance.

>>> GitHub Link <<<

Build a Dashboard with Plotly Dash

- Pie Chart: to show the total launches by sites
  - Display relative proportions of multiple classes (sites) of data.
  - Size of the circle can be made proportional to the total quantity it represents.
- Scatter Graph: to show the relationship with outcome and payload mass (kg) for the different booster versions.
  - Visually observe how payload may be correlated with mission outcomes for selected site(s).
  - Shows relationship between two variables.
  - Show non-linear pattern.

>>> GitHub Link <<<

Predictive Analysis (Classification)

Building Model

1. Load the dataset into dataframe.
2. Transform Data.
3. Split the data into training and testing sets.
4. Pick a machine learning algorithms.
5. Set parameters and algorithms for the GridSearchCV objects.
6. Fit the dataset into the GridSearchCV objects and train the dataset.

Evaluation

1. Check the accuracy
2. Tune the hyperparameters
3. Plot confusion Matrix

Improve the model

- Feature Engineering
- Algorithm Tuning

Finding the best performing model

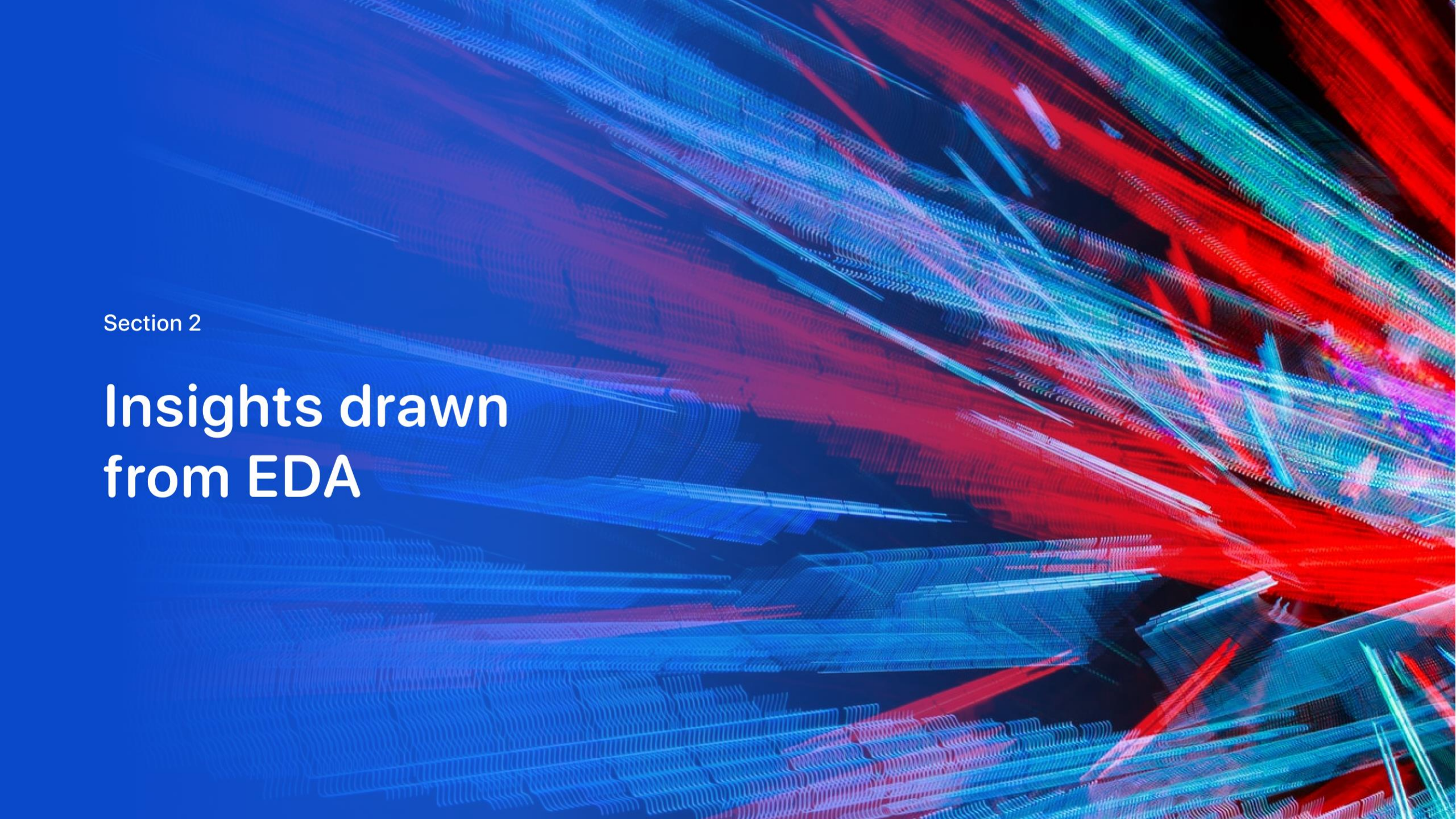
- 1. Based on the accuracy score

Results

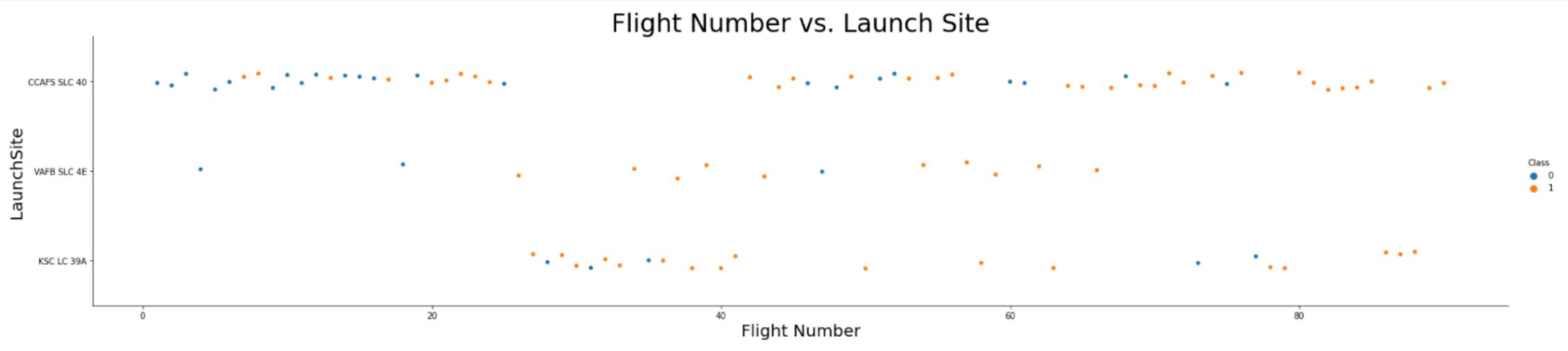
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

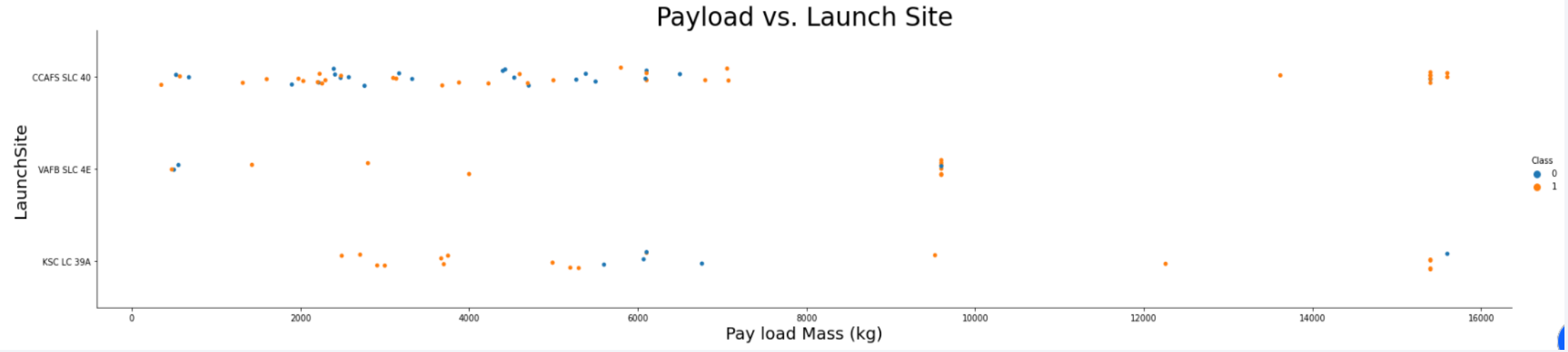


Flight Number vs. Launch Site



- The site “CCAFS SLC 40” launched most flights, and this site’s success rate is increasing with flight numbers.
- The site “VAFB SLC 4E” launched least flights, and most of launches were successful.
- The site “KSC LC 39A” also has great success rate.

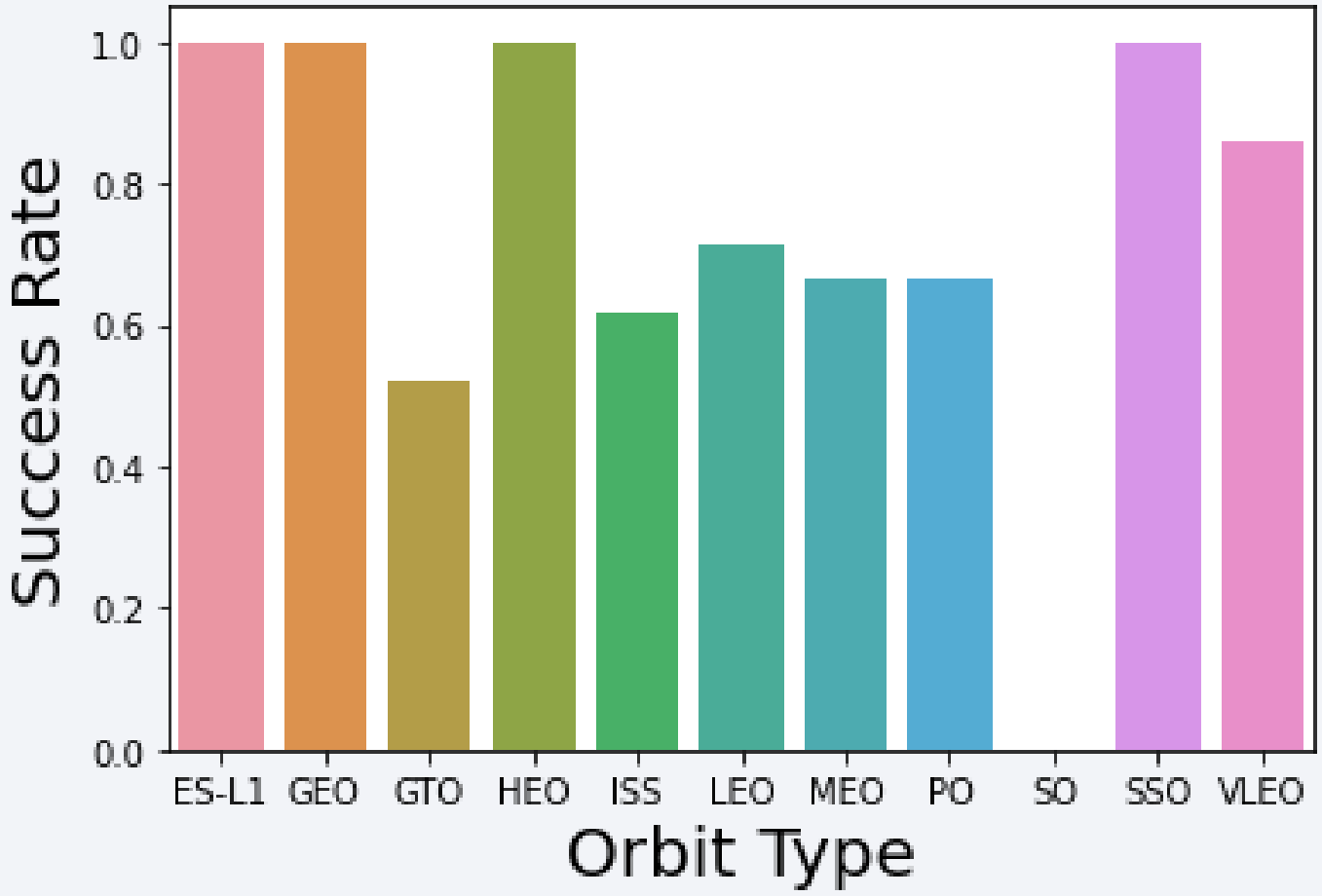
Payload vs. Launch Site



- The site “CCAFS SLC 40” has highest success rate (100%) in launching heavy payload(12000+kg). It launched most of rockets.
- “VAFB-SLC 4E” never launched a rockets with mass greater 10000kg. It launched lease rockets.
- Overall, heavy rockets has higher success rate compared to lighter one. The “KSC LC39A” has good record of launching sub-5000 kg rockets.

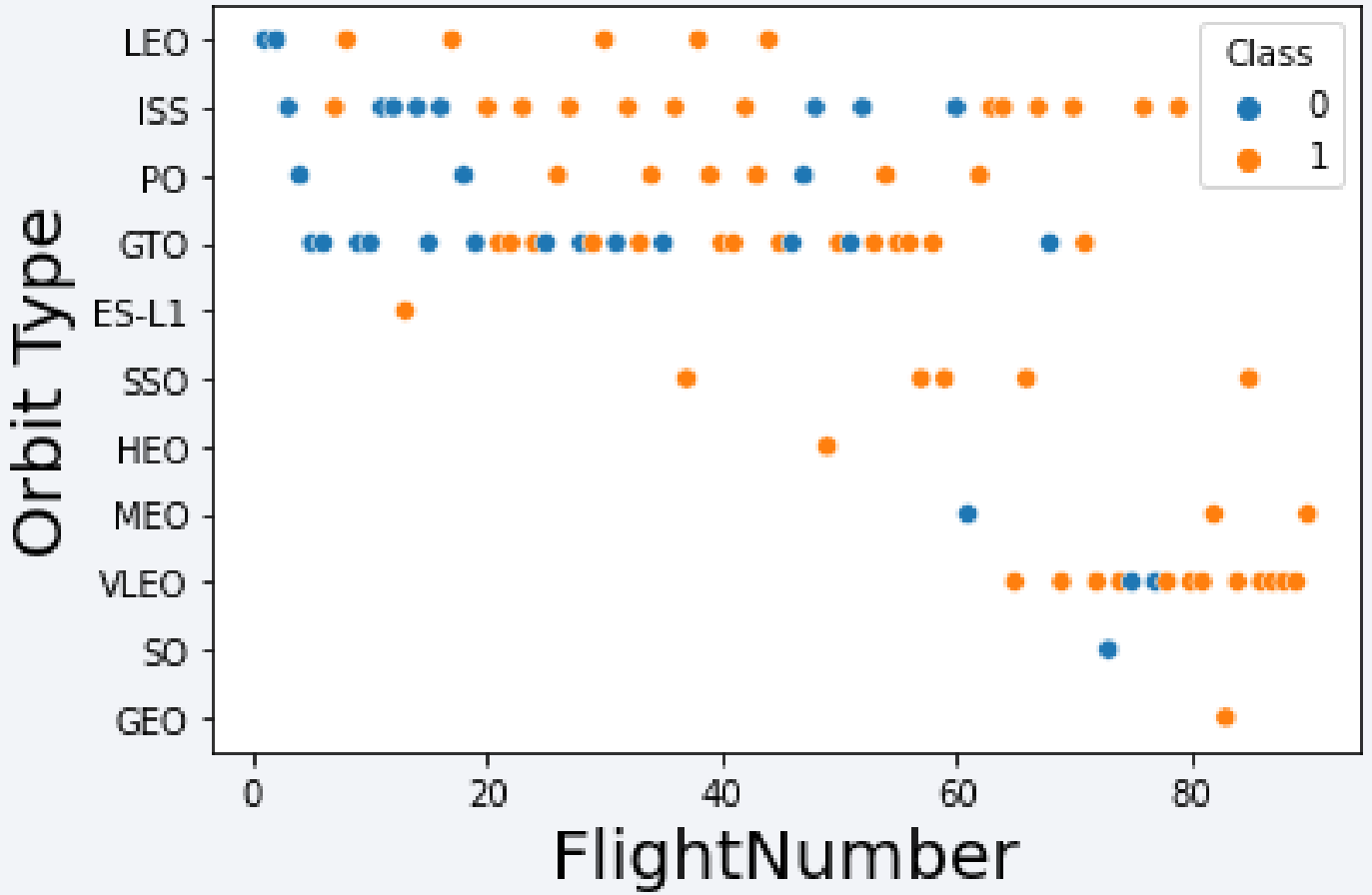
Success Rate vs. Orbit Type

- Orbit type “EL-L1”, “GEO”, “HEO”, and ”SSO” has 100 percent success rate.
- Orbit type “SO” has zero success rate.



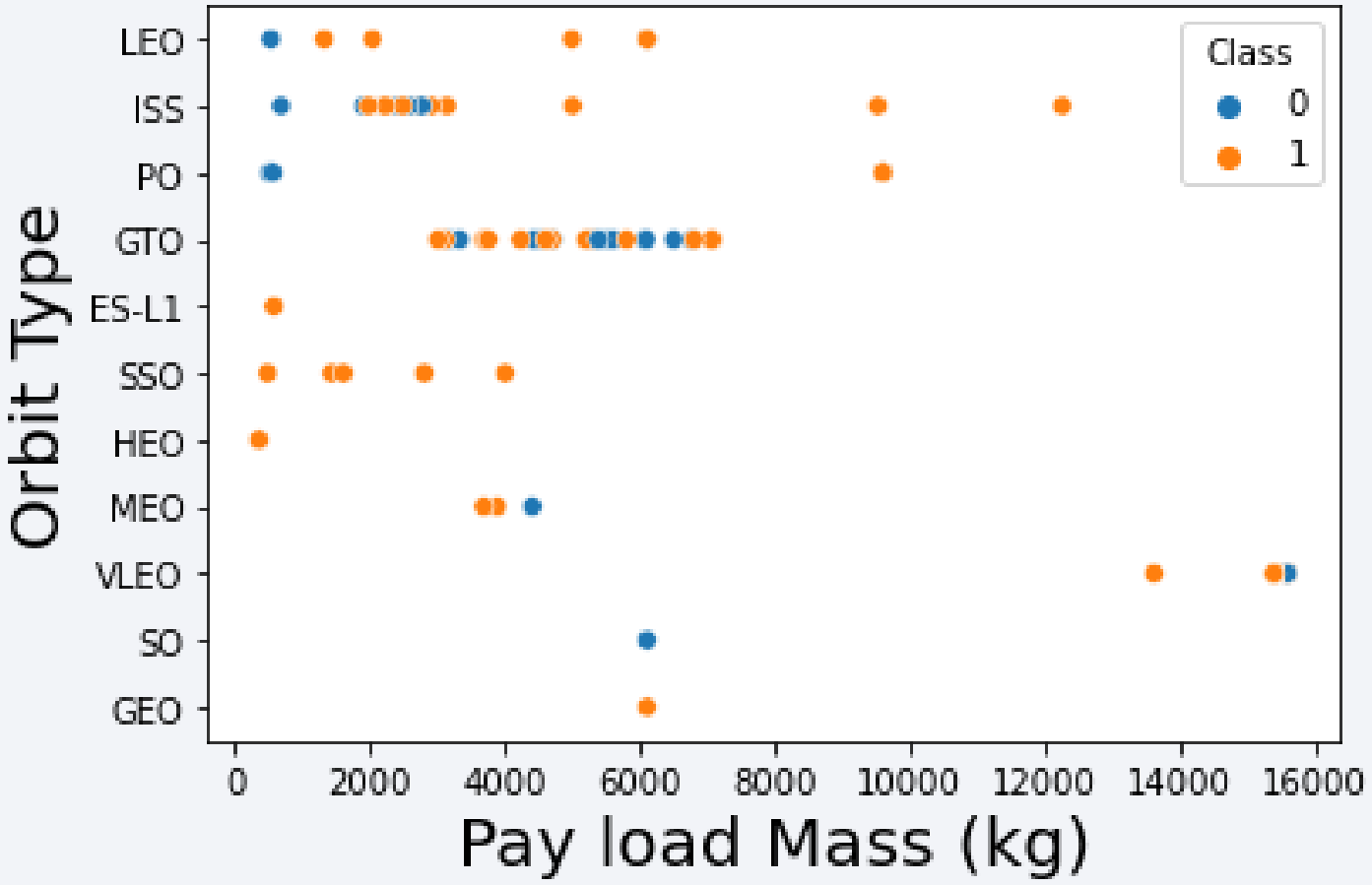
Flight Number vs. Orbit Type

- LEO orbit the Success appears related to the number of flights.
- There seems to be no relationship between flight number when in GTO orbit.



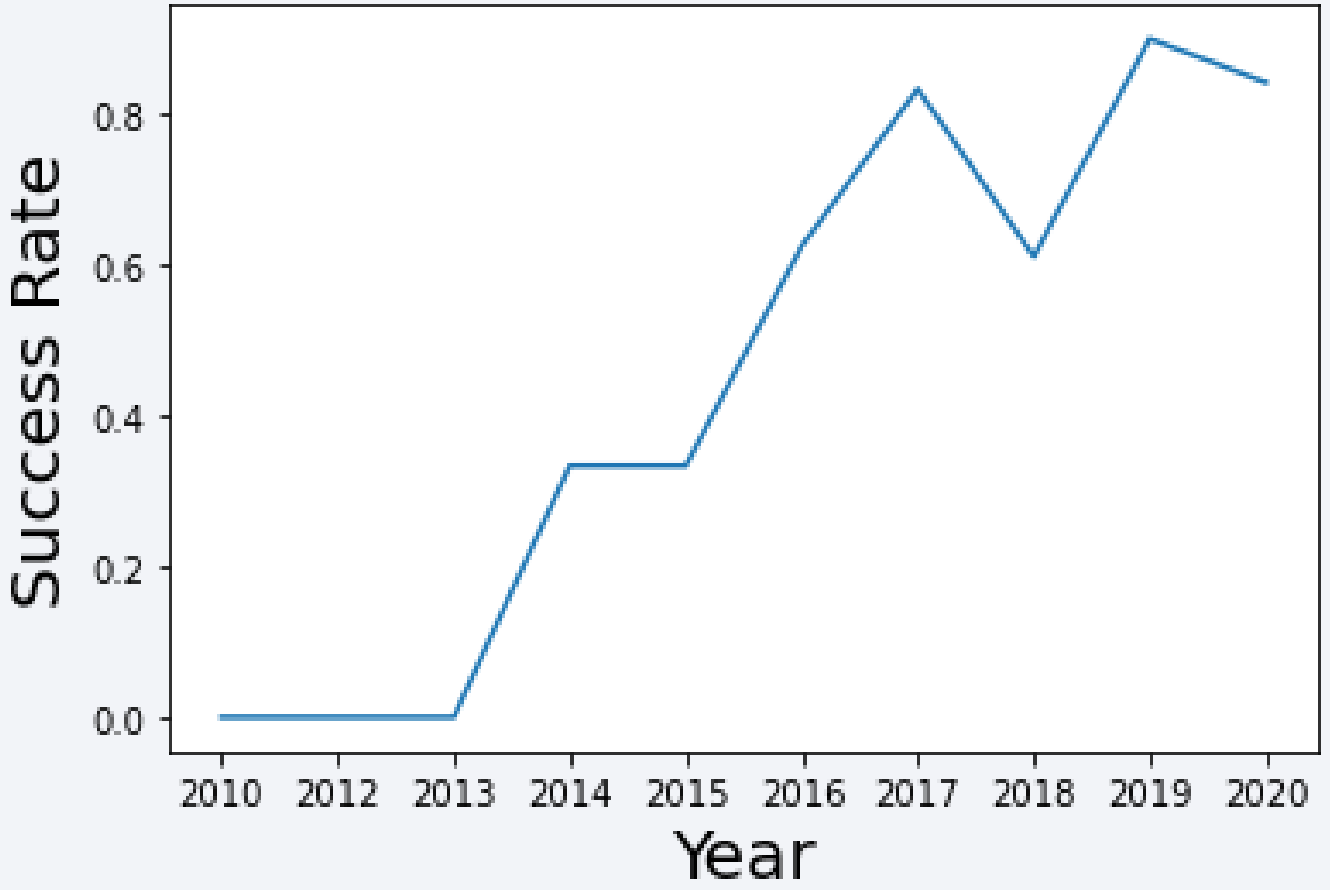
Payload vs. Orbit Type

- 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 unsuccessful landing are both there here.



Launch Success Yearly Trend

- Success rate since 2013 kept increasing till 2020
- There was a big drop in 2018, from 80% to 60%.



All Launch Site Names

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

```
%%sql
SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

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

DISTINCT will show unique values in variable **lanunch_site.**

Launch Site Names Begin with 'CCA'

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

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

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

LIKE will pick out the site name start with string “CCA”.

Limit 5 will restrict data to top 5.

Total Payload Mass

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

```
%%sql
SELECT SUM(payload_mass__kg_) FROM SPACEXDATASET
WHERE CUSTOMER LIKE 'NASA%';
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

1
36679

SUM() to adding up all payload mass with restricted customer “NASA”.

Average Payload Mass by F9 v1.1

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

```
%%sql
SELECT AVG(payload_mass__kg_) FROM SPACEXDATASET
WHERE booster_version LIKE 'F9 v1.1';
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

1
3676

AVG() to get average payload for booster version F9 V1.1.

First Successful Ground Landing Date

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

```
%%sql
SELECT MIN(DATE) FROM SPACEXDATASET
WHERE landing__outcome LIKE '%ground pad%';
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

1
2017-01-05

Min() with date shows the earliest launch for ground pad.

Successful Drone Ship Landing with Payload between 4000 and 6000

List 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 SPACEXDATASET
WHERE landing__outcome LIKE '%Success (drone ship)%'
and payload_mass__kg_ BETWEEN 4000 AND 6000
;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

3]:

booster_version
F9 FT B1022
F9 FT B1031.2

Use LIKE to filter out successfully drone ship lading. Combine with AND and BETWEEN to filter the payload mass.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes.

```
%%sql
SELECT count(mission_outcome)
FROM SPACEXDATASET
WHERE mission_outcome LIKE '%Success%'
;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

]:

1
45

```
%%sql
SELECT count(mission_outcome)
FROM SPACEXDATASET
WHERE mission_outcome LIKE '%Failure%'
;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

]:

1
0

Successful Missions = 45 and Failure Missions = 0

Boosters Carried Maximum Payload

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

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
where payload_mass__kg_ = (select max(payload_mass__kg_) from SPACEXDATASET);
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

7]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

Find the max payload rockets using MAX function. Then find corresponding booster version for it.

2015 Launch Records

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

```
%%sql
SELECT booster_version
FROM SPACEXDATASET
WHERE landing__outcome LIKE '%Failure (drone ship)%' AND DATE LIKE '%2015%'
;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

]:

booster_version
F9 v1.1 B1012

Find the drop ships that failed in 2015 using LIKE and AND function. Then find corresponding booster version for it.

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, COUNT(*) as outcomes
FROM SPACEXDATASET
GROUP BY landing__outcome
ORDER BY outcomes DESC
;
```

```
 * ibm_db_sa://mhl21482:***@54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud:32733/BLUDB
Done.
```

9]:

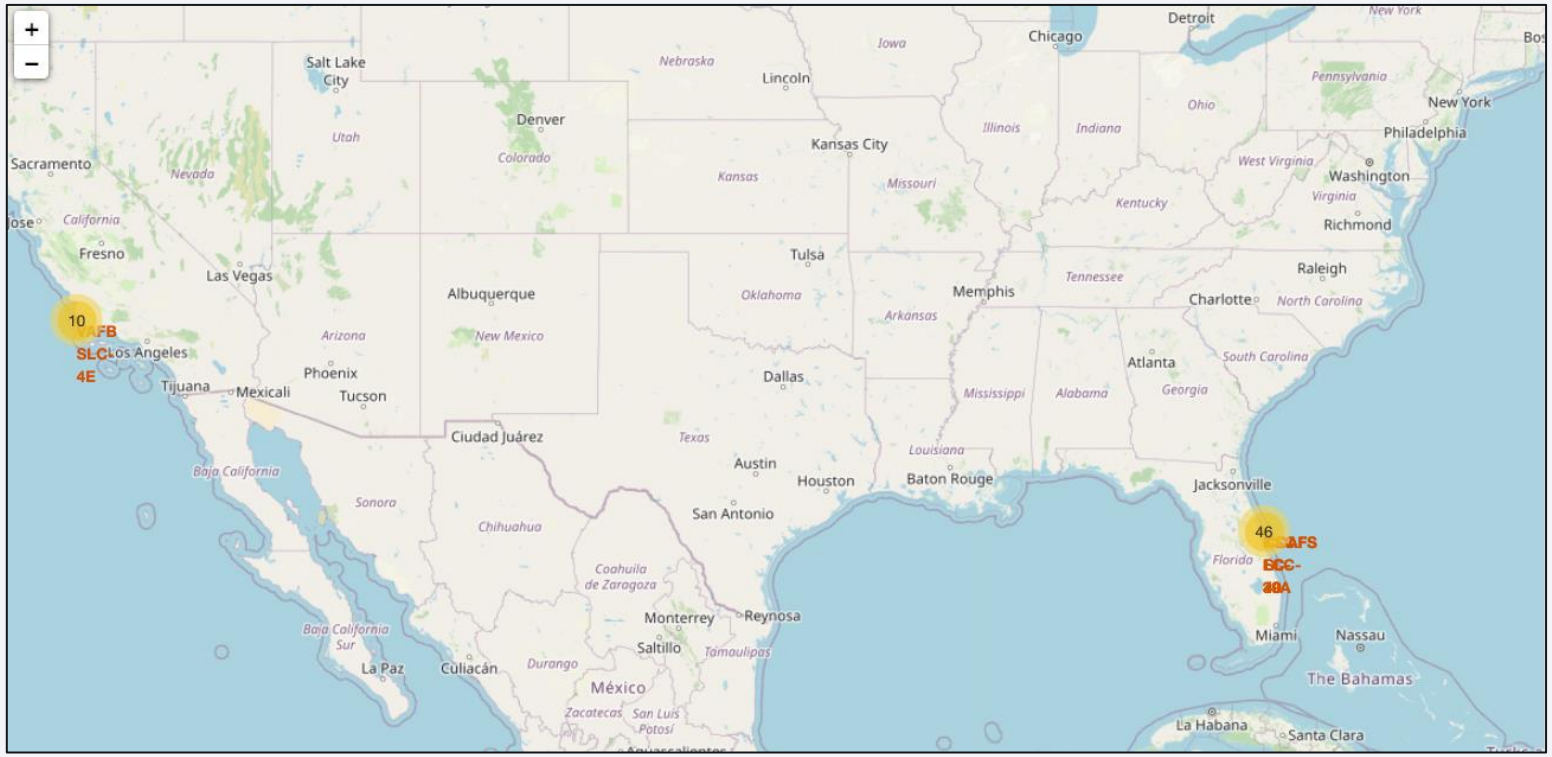
landing__outcome	outcomes
Success	18
No attempt	12
Success (drone ship)	5
Success (ground pad)	4
Failure (drone ship)	2
Failure (parachute)	2
Controlled (ocean)	1
Failure	1

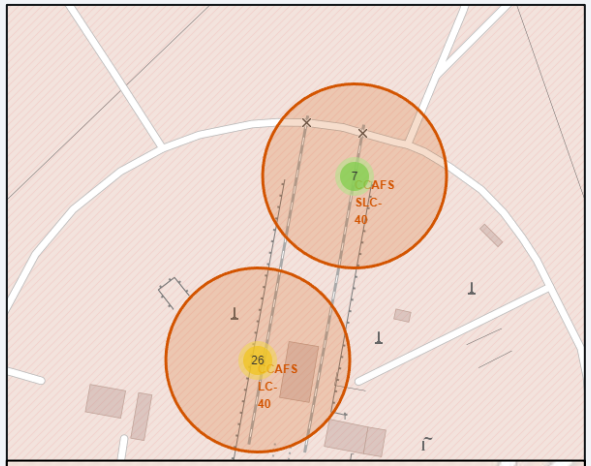
Section 4

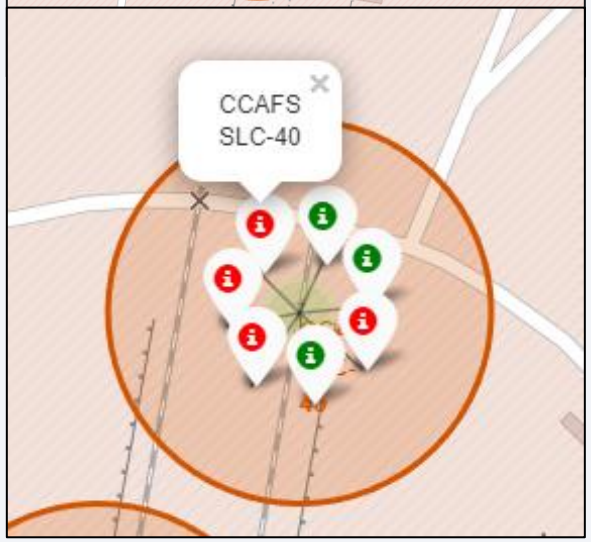
Launch Sites Proximities Analysis

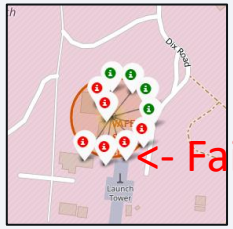
All Launch Sites

Mark result of launches for each site







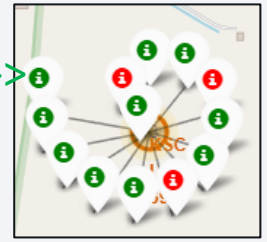


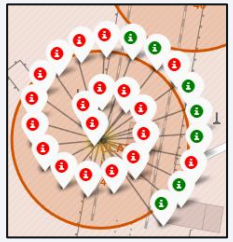
<- California Launch Site

<- Failures (Red)

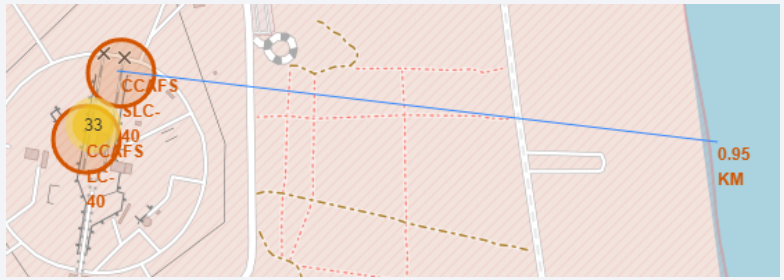
Successful-> (Green)

Florida Launch Site ->



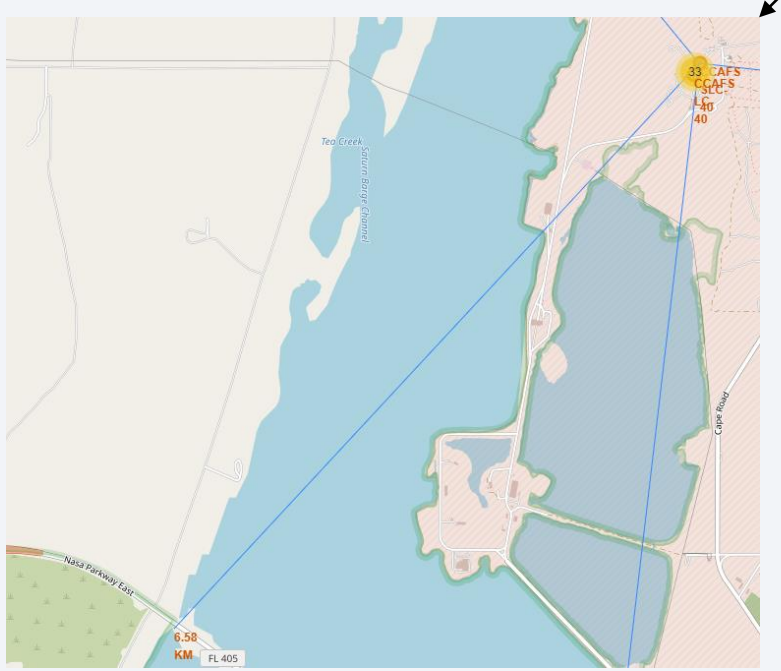


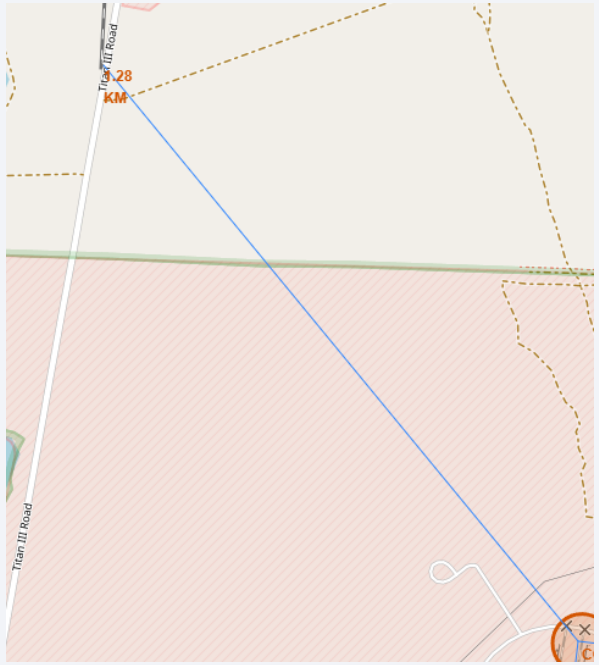
Explore and analyze the proximities of launch sites

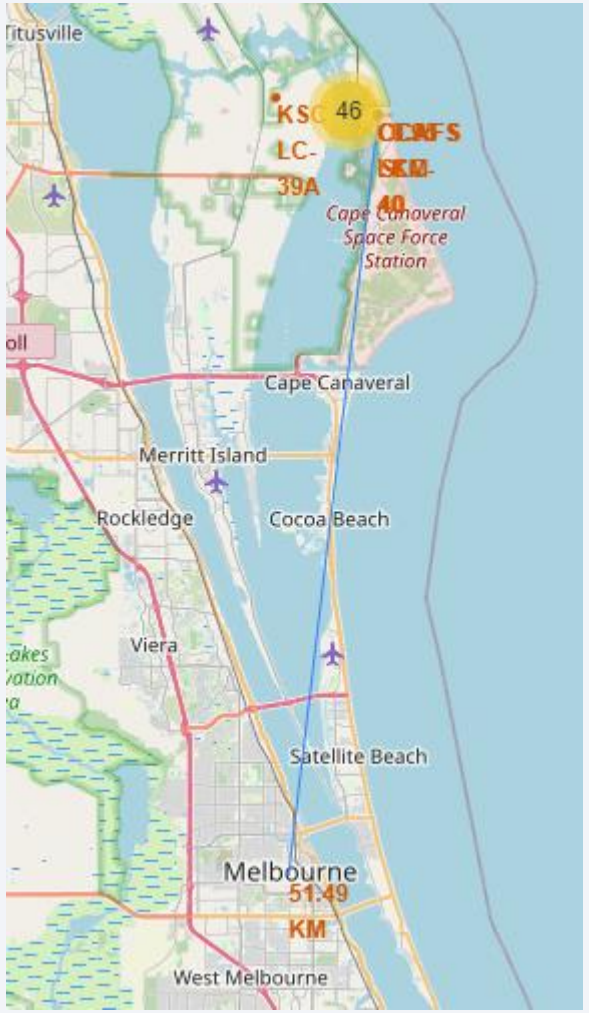


Costline
Highway

City
Railway

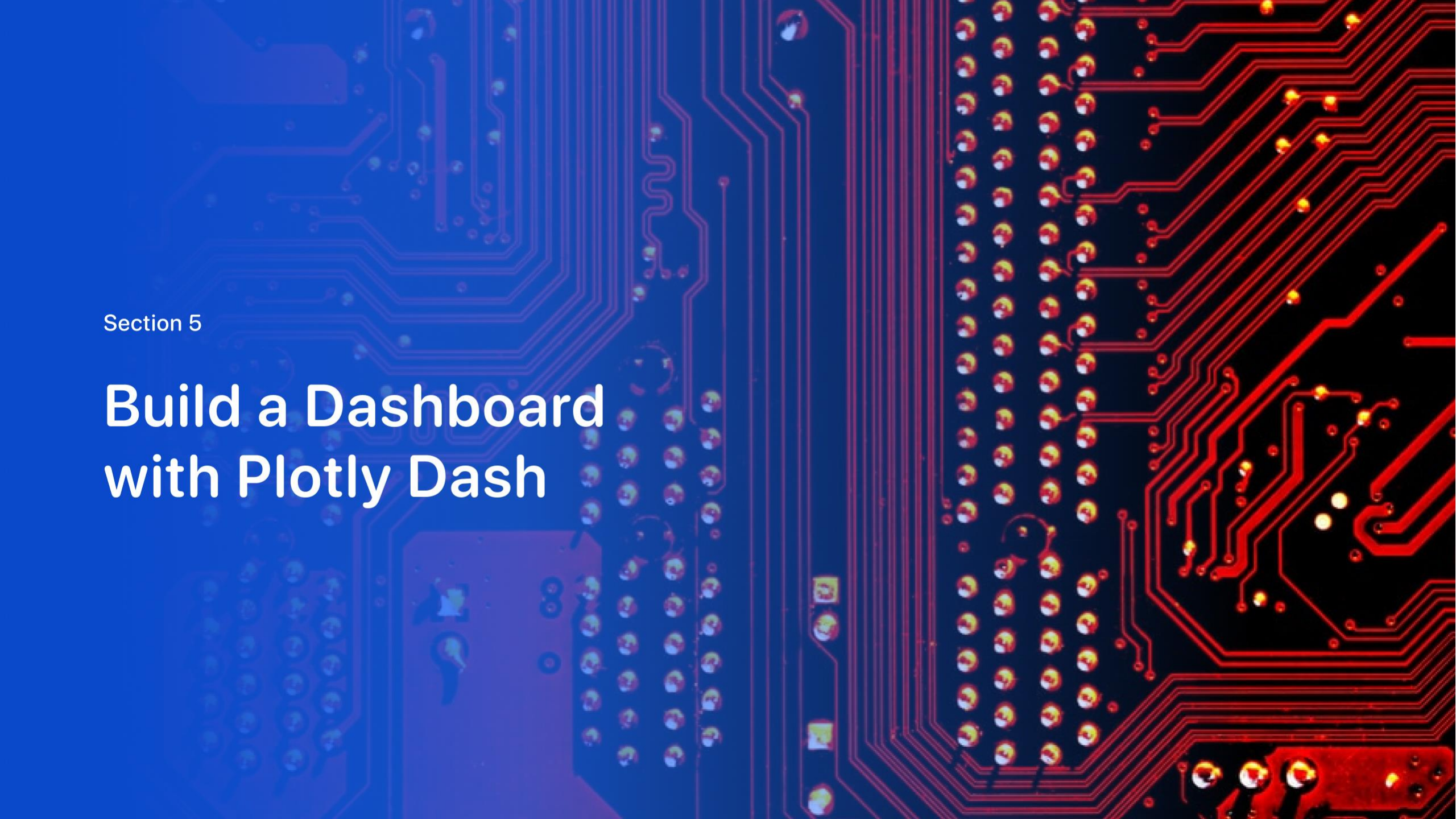




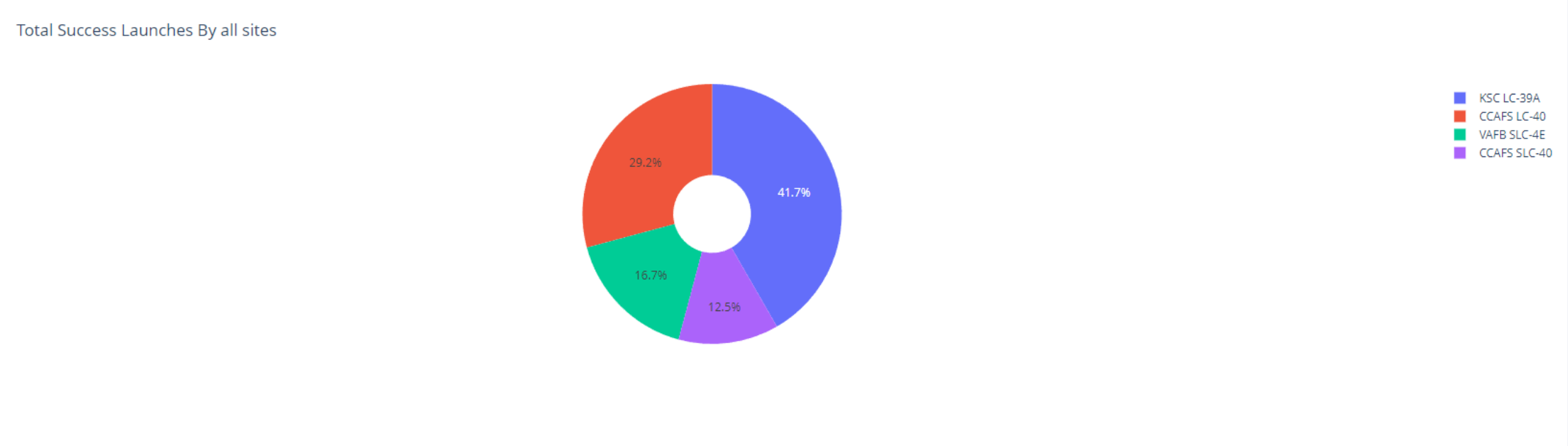


Section 5

Build a Dashboard with Plotly Dash



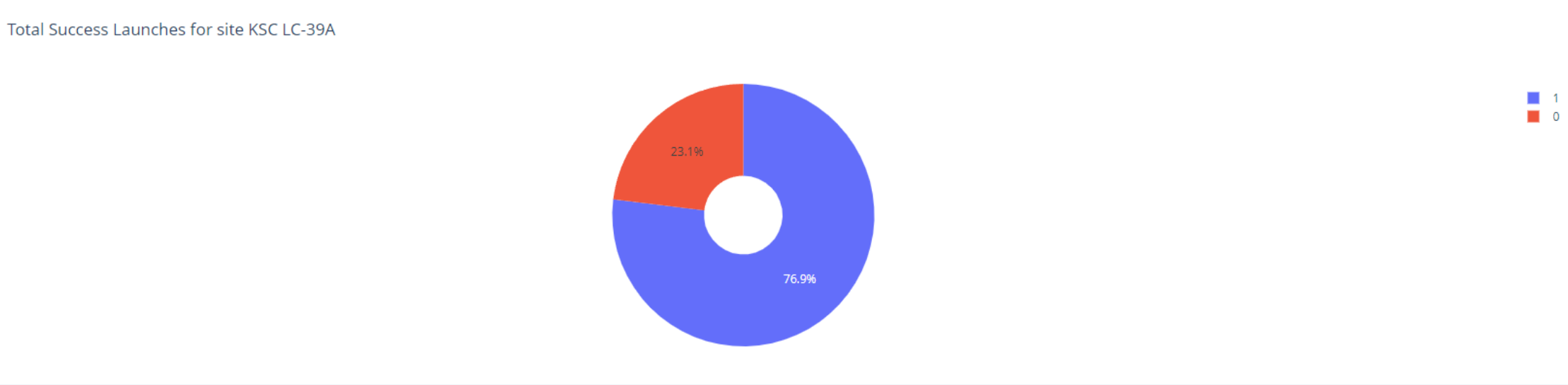
Pie Chart: Launch success count for all sites



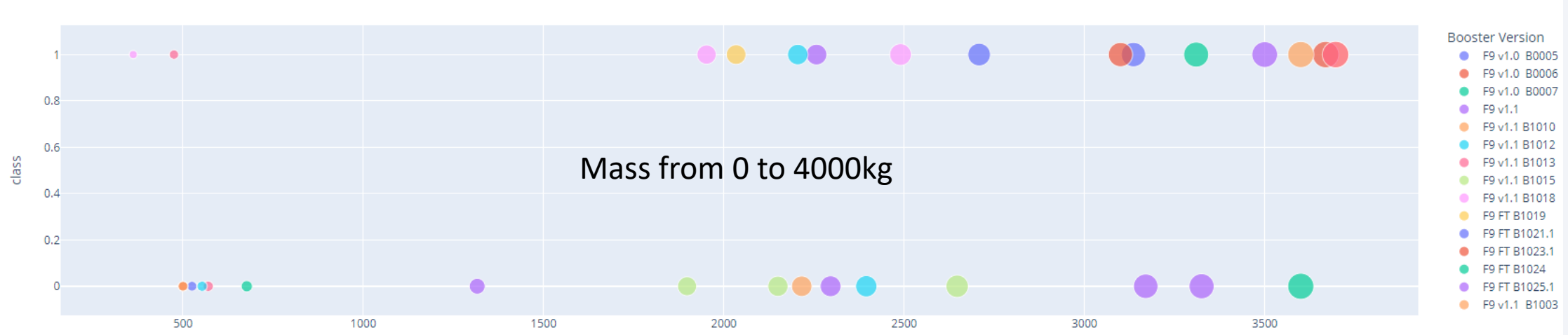
KSC LC-39A has most successful launches from all the sites.
VAFB SLC-4E has least successful launches.

Total Success Launches for site KSC LC-39A

The KSC LC-39A has 76.9 percent of success rate.



Dashboard – Payload vs. Launch Outcome

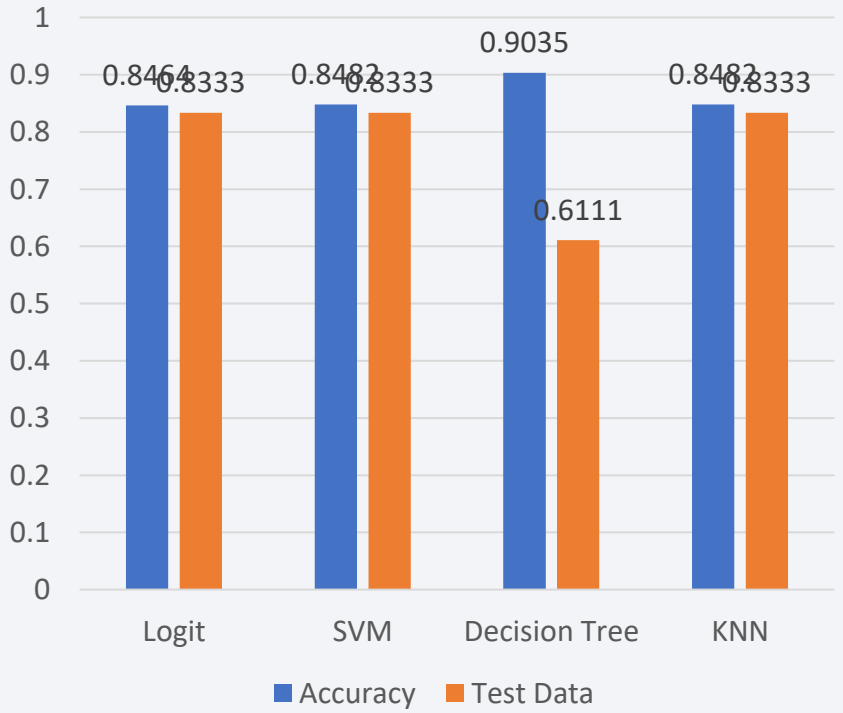


Section 6

Predictive Analysis (Classification)

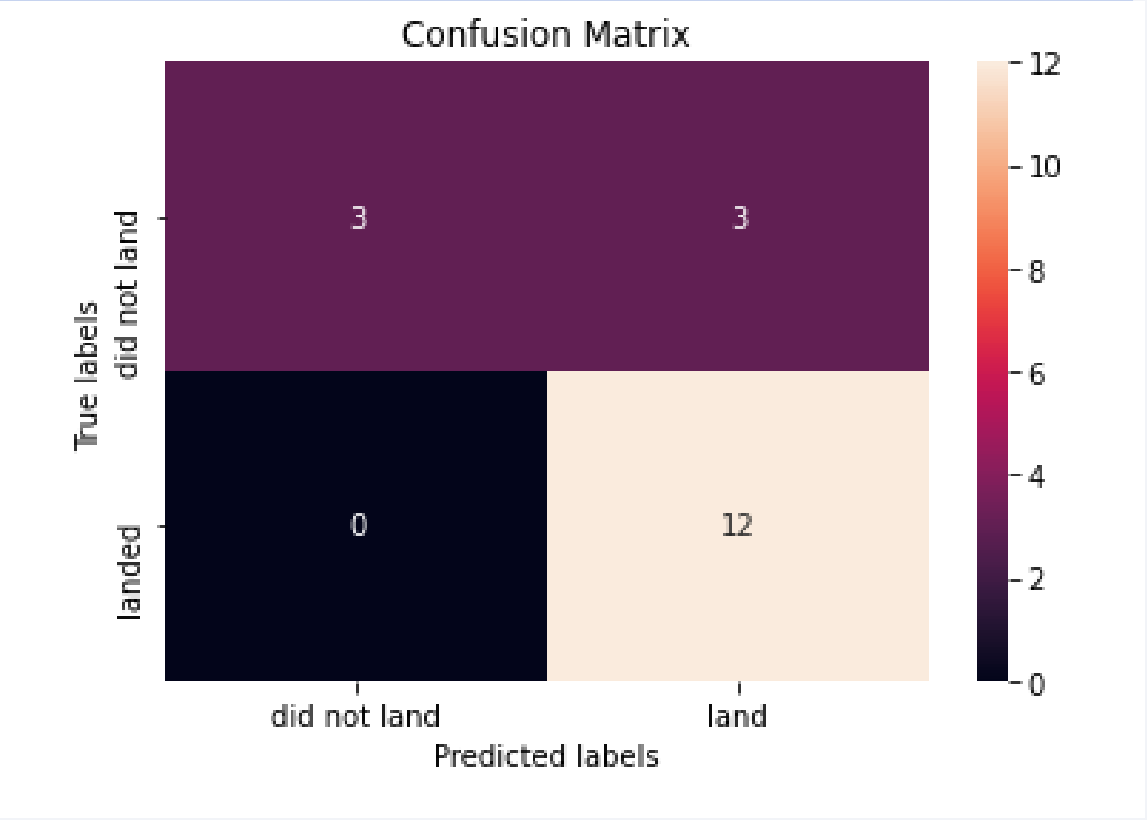
Classification Accuracy

- Logit, SVM, and KNN are extremely close. They all have accuracy score of 0.8333 from test data.
- Decision Tree has highest accuracy of 0.9036. The accuracy score of test data is 0.6111.



Confusion Matrix

- Show the confusion matrix of the decision tree model.
- The result is not clear when true label is “Did not land”.



Conclusions

- The Decision Tree classifier algorithm is the best for this case.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches increase over the years.
- KSC LC-39A has most successful launches from all the sites.
- Orbit type “EL-L1”, “GEO”, “HEO”, and ”SSO” has 100 percent success rate.

Thank you!