IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

NG WEI YI

20th SEPTEMBER 2022

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using SpaceX API and web scraping
  - Data wrangling
  - Exploratory Data Analysis (EDA) with SQL and visualization
  - Interactive visual analytics with Folium
  - Machine learning prediction
- Summary of all results
  - Exploratory Data Analysis (EDA) result
  - Machine learning prediction result

Introduction

- Project background and context

  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 by an alternate company to bid against SpaceX for a rocket launch.

- Problems you want to find answers

  - Which factors determine the success or failure of a rocket landing?
  - What are the effects of each relationship of the rocket variables on the outcome?
  - Conditions that aid SpaceX to achieve highest rocket landing success rate.

Section 1

Methodology

Methodology

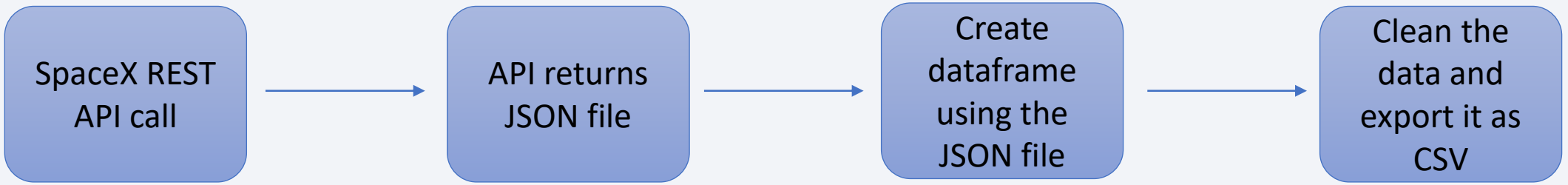
Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web scraping from Wikipedia
- Perform data wrangling
  - Dropping irrelevant columns
  - Applied one hot encoding to categorical variables
- 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

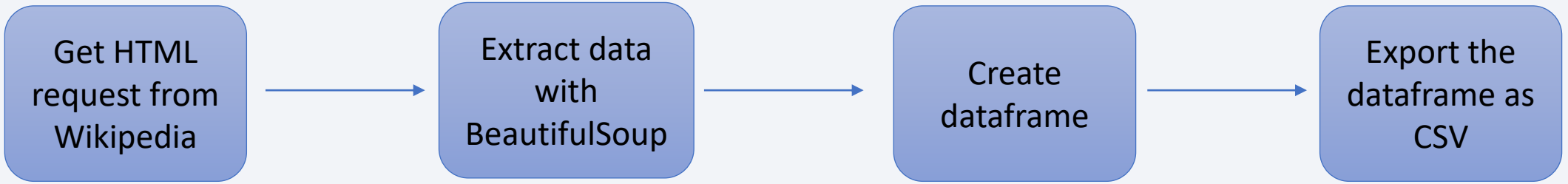
Data sets are collected using SpaceX REST API and web scraping from Wikipedia

- Data sets collected using SpaceX REST API (https://api.spacexdata.com/v4/)



Github link

- Data sets collected by web scraping from Wikipedia (https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches)



Github link

Data Collection – SpaceX API

1. Getting response from API

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

2. Convert response to JSON file

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

3. Extract data from JSON file

```
getBoosterVersion(data)
```

```
getLaunchSite(data)
```

```
getPayloadData(data)
```

```
getCoreData(data)
```

4. Combine the data and columns into a dictionary

```
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. Create dataframe

```
# Create a data from launch_dict
data2 = pd.DataFrame(launch_dict)
```

6. Filter dataframe

```
data_falcon9 = data2[data2['BoosterVersion'] != 'Falcon 1']
```

7. Export dataframe as CSV file

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

Github link

Data Collection - Scraping

1. Getting response from HTML

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
response = requests.get(static_url)
```

2. Create BeautifulSoup object

```
soup = BeautifulSoup(response.text, 'html')
```

3. Find all tables

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

4. Get column names

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

5. Create dictionary

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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. Add data to keys

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

7. Create a dataframe from dictionary

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

8. Export dataframe as CSV file

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

Github link

Data Wrangling

- Data wrangling is the process of removing errors and clean the data to make them more accessible and less complex to analyze.
- The outcomes of rocket landing are divided into two categories:
  - True Ocean, True RTLS, True ASDS mean a successful landing (denoted as 1).
  - False Ocean, False RTLS, False ASDS mean a failure landing (denoted as 0).

1. Calculate the total count for each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

2. Calculate the total count for each orbit type

```
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
```

3. Calculate the total count for each type Of outcome

```
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
```

4. Convert all the outcomes into two categorical values, 1 and 0

```
landing_class = df['Outcome'].map(lambda x: 0 if x in bad_outcomes else 1)
landing_class
```

5. Convert the dataframe to CSV file

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

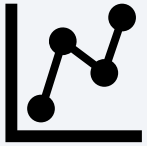
Github link

EDA with Data Visualization

- Three different charts are used to visualize the data: scatter, bar, and line charts.

Scatter Chart

- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload Mass vs. Launch Site
- Orbit vs. Flight Number
- Payload Mass vs. Orbit



Line Chart

- Success Rate vs. Year

Bar Chart

- Success Rate vs. Orbit

Github link

EDA with SQL

- SQL queries are performed to extract and understand major information about the data set. Queries performed are shown below:
  - 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 successful landing outcome in ground pad was achieved
  - 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
  - List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
  - 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

- One of the factors that affect the launch success rate is location. Hence, finding the optimal location for building a launch site is important.
- An interactive map can be built using Folium in Python:
  - A Folium Map object is created with initial center location to be NASA Johnson Space Center at Houston, Texas.
  - Red circle at NASA Johnson Space Center's coordinates with a label showing the name (*folium.Circle*, *folium.map.Marker*)
  - Red circle at each launch site's coordinates with a label showing the name (*folium.Circle*, *folium.map.Marker, folium.features.DivIcon*)
  - Success and failure launches for each site are marked, MarkerCluster object is used to simplify a map containing many markers having the same coordinate (*folium.plugins.MarkerCluster*)
  - Two colors are used for the outcome of rocket landing, Green is used for successful landing whereas Red is for failed landing (*folium.map.Marker, folium.Icon*)
  - The distances between launch sites and key locations (railway, highway, city) and a line is plotted between them (*folium.map.Marker, folium.PolyLine, folium.features.DivIcon*)

Github link

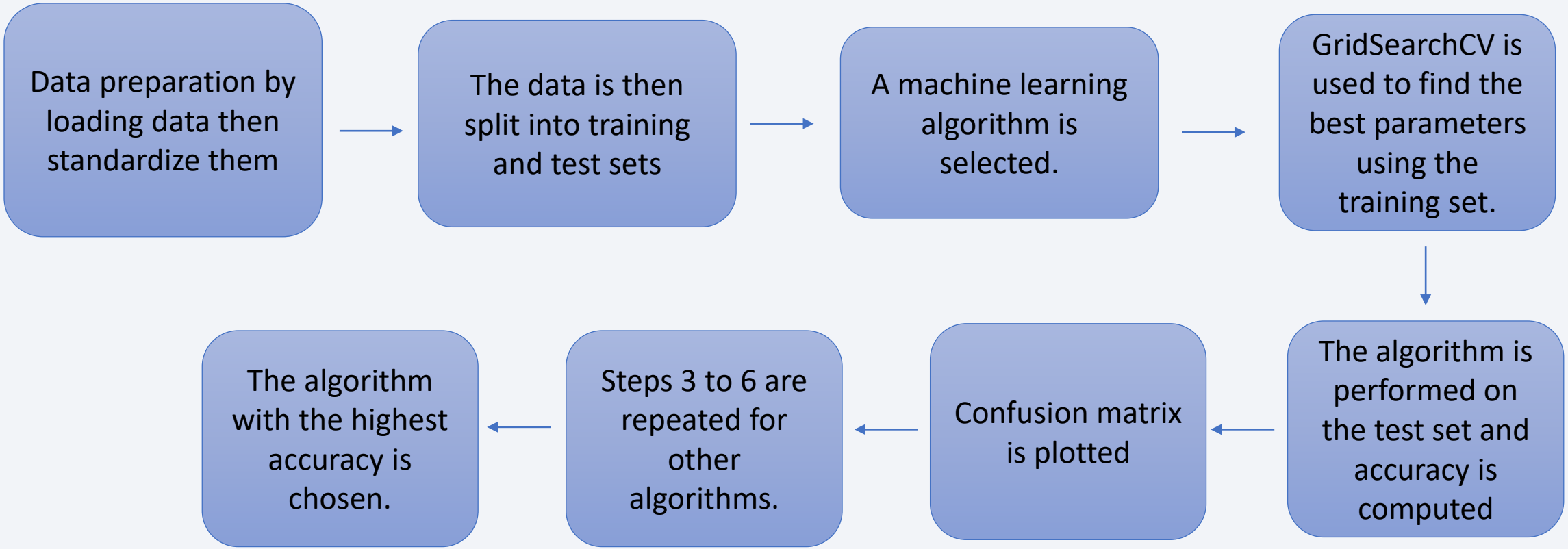
Build a Dashboard with Plotly Dash

- The dashboard built has dropdown to choose a site or all sites, pie chart, scatter plot, and range slider:
  - The Dropdown feature allows user to choose a site or all sites (*dash.dcc.Dropdown*)
  - Pie chart shows the total success and failure rate for a launch site using the Dropdown feature (*plotly.express.pie*)
  - Scatter plot shows the user the relationship between success rate and payload mass (*plotly.express.scatter)*
  - Range slider allows a user to select a specific range of payload mass for the scatter plot (*dash.dcc.RangeSlider*)

Github link

Predictive Analysis (Classification)

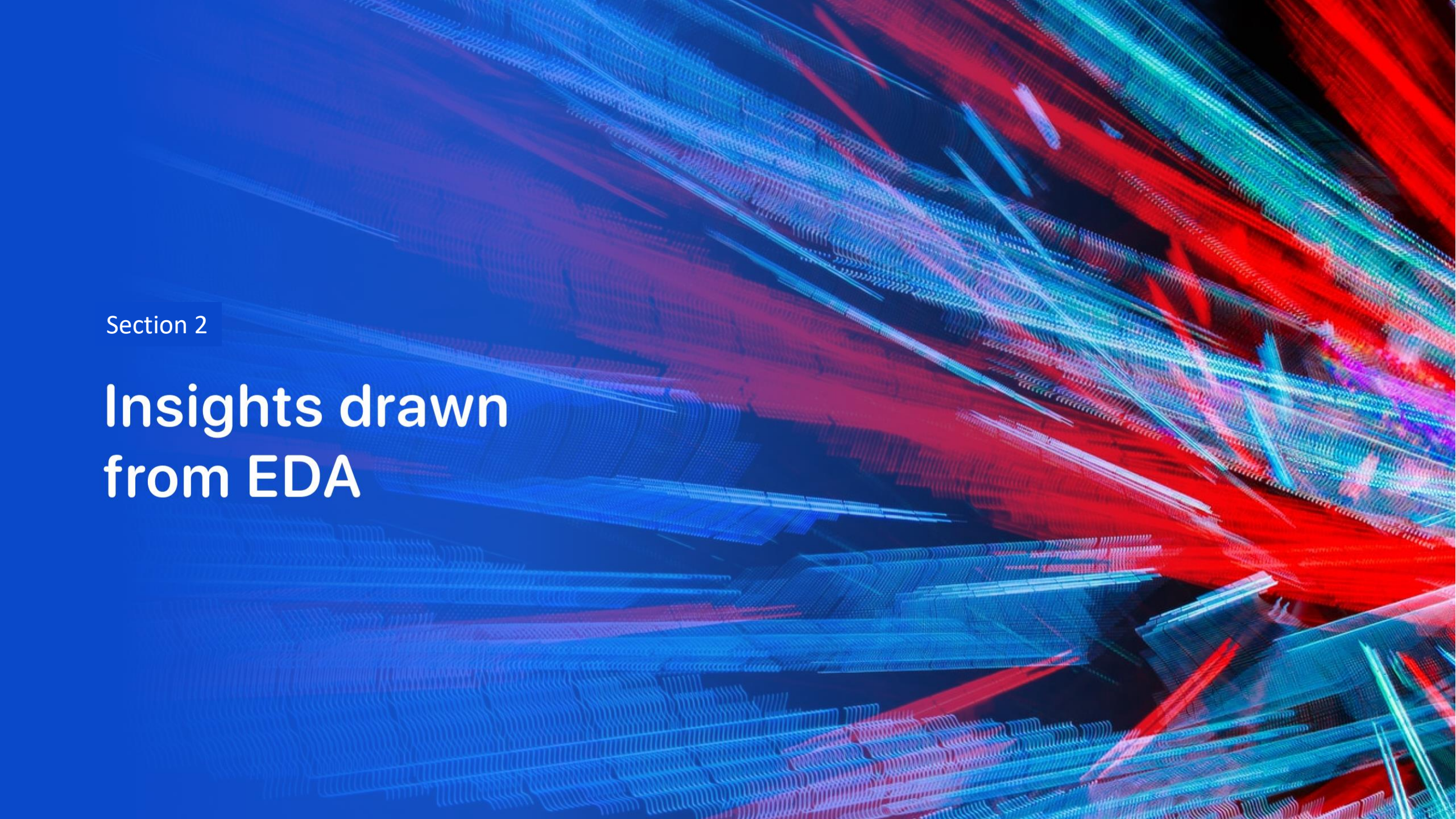
- The flowchart below shows the model development process:



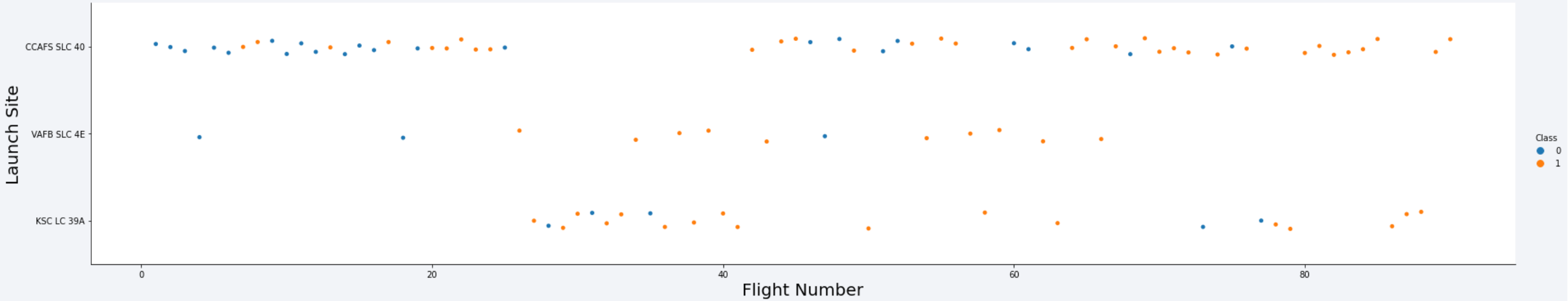
Github link

Results

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

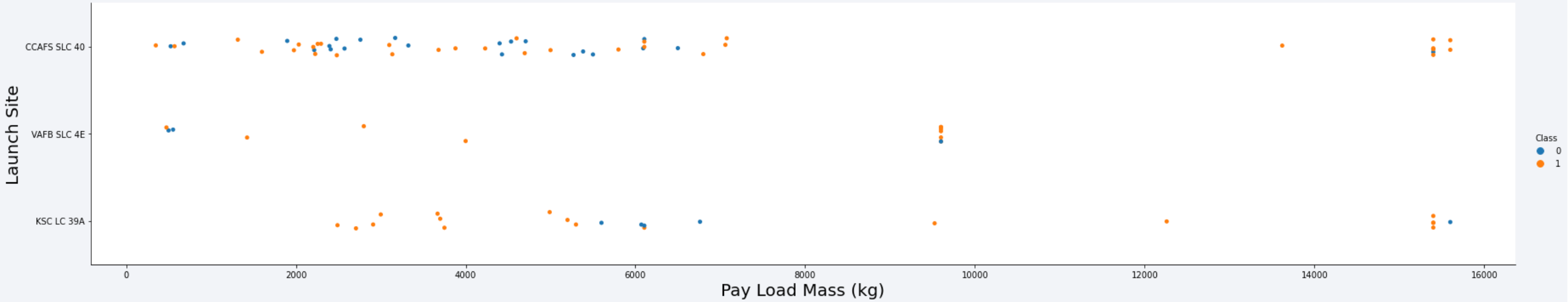

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



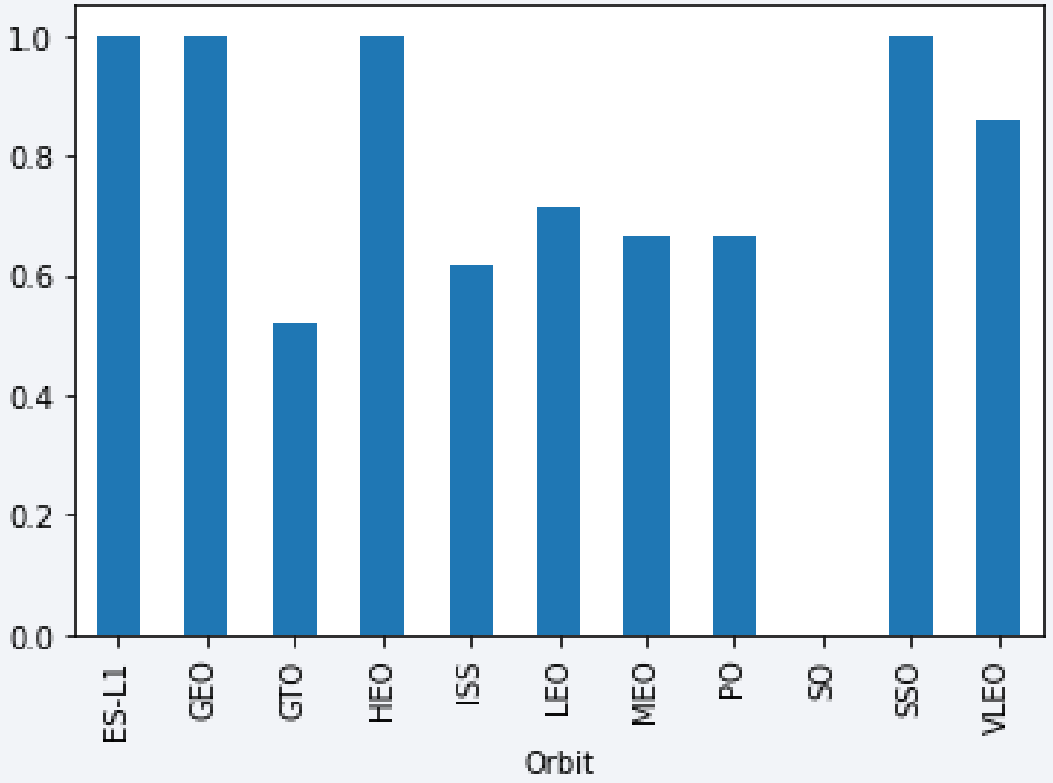
- We can observe that the success rate is increasing for each site.

Payload vs. Launch Site



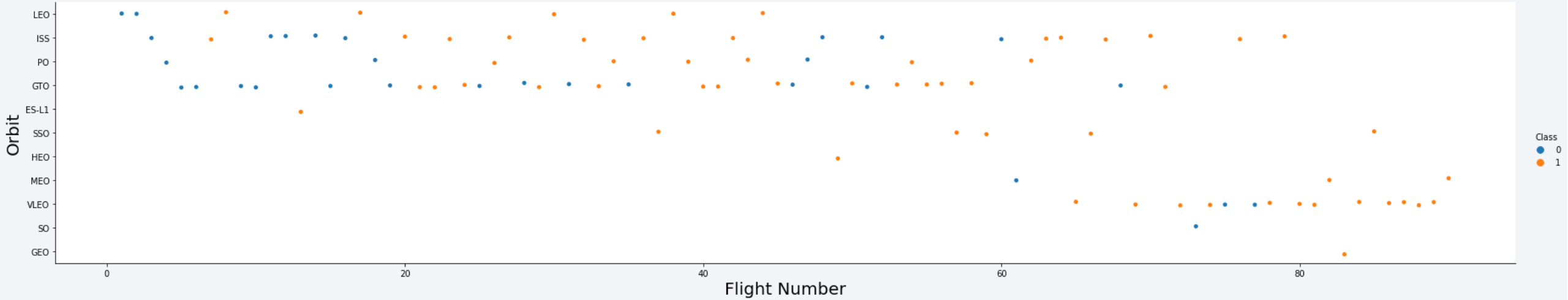
- Heavier payloads seem to be having higher success rate as compared to lighter payloads. However, this depends on launch site.

Success Rate vs. Orbit Type



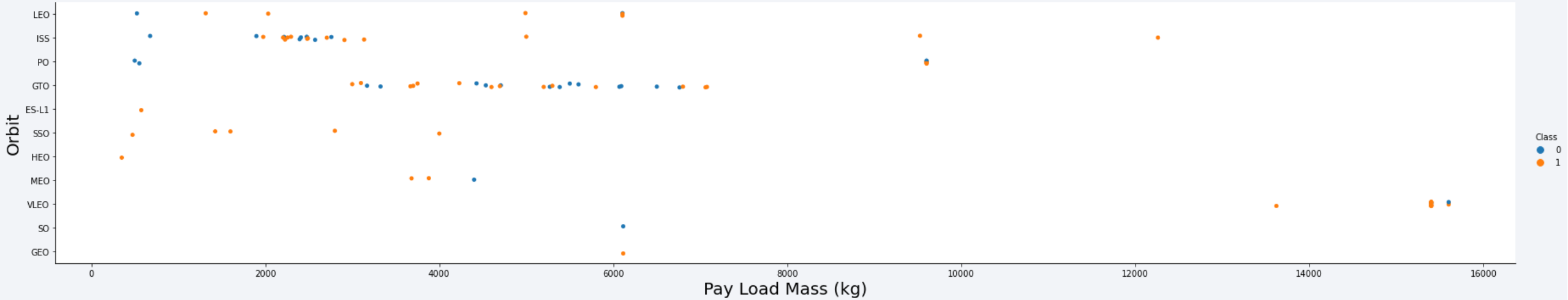
- ES-L1, GEO, HEO, and SSO orbit types have higher success rate as compared to other orbit types.

Flight Number vs. Orbit Type



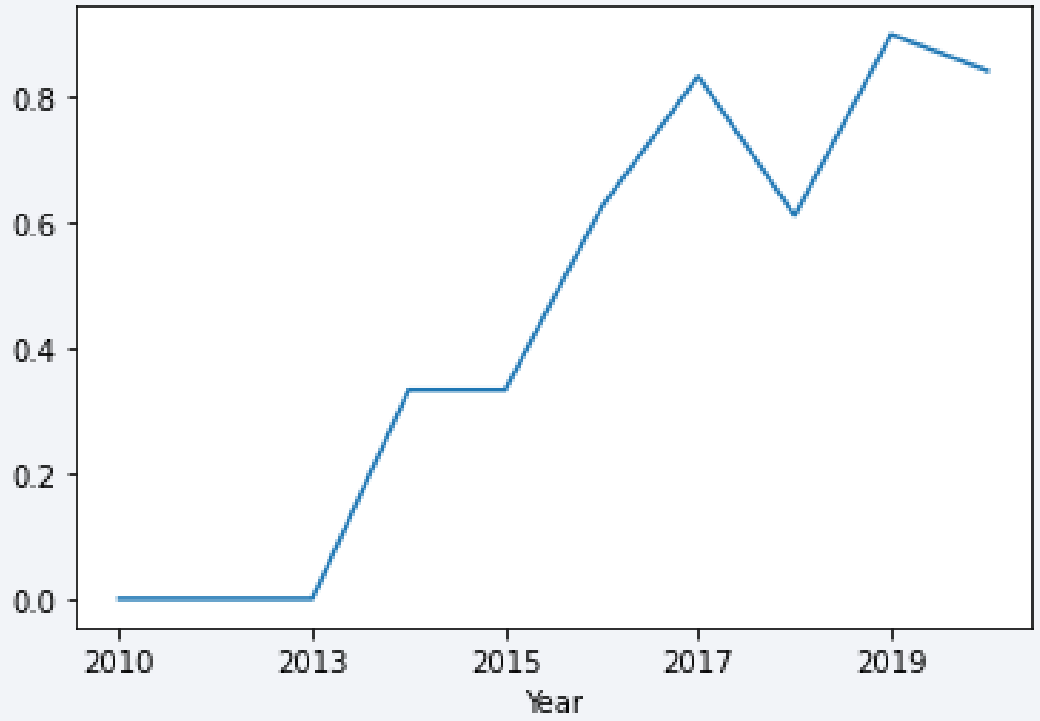
- In general, the success rate has been improved with increasing flight number.
- The obvious case is that the success rate increases with flight number for the VLEO orbit type.

Payload vs. Orbit Type



- The success rate of the launches has been greatly influenced by the weight of payloads in certain orbit types. For example, heavier payloads have higher success rate for VLEO and ISS orbit types.

Launch Success Yearly Trend



- The success rate has been increasing every year as shown in the line chart.

All Launch Site Names

```
%sql SELECT DISTINCT(LAUNCH_SITE) FROM SPACEXTBL;
```

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

- DISTINCT is used in the query to show unique launch sites from the data.

Launch Site Names Begin with 'CCA'

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

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The WHERE and LIKE clauses are used to filter launch sites that contain the substring ‘CCA’ and the query limits to only 5 records.

Total Payload Mass

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

1
45596

- SUM() function is used to calculate the sum of all payload masses where the customer is NASA (CRS).

Average Payload Mass by F9 v1.1

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

1
2928

- The AVG() function is used to calculate the average of all payload masses where the booster version is F9 v1.1.

First Successful Ground Landing Date

```
%%sql SELECT MIN(DATE) FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

1
2015-12-22

- The MIN() function here is used to find the earliest date where the landing outcome of ground pad is successful.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT BOOSTER_VERSION FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

- The WHERE and AND clauses are used to filter the landing outcomes which have successfully landed on a drone ship and having the payload mass between 4000 KG and 6000 KG.

Total Number of Successful and Failure Mission Outcomes

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

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

- The COUNT() function counts the number of rows in the data and by grouping the mission outcomes, there are in total of 100 successful outcomes and 1 failed outcome.

Boosters Carried Maximum Payload

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

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

- A subquery is used to find the booster versions that have the maximum payload mass from the data.

2015 Launch Records

```
%%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEXTBL
WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

- The WHERE and AND clauses are used to filter the booster version and launch site having landing outcomes which have failed to land on a drone ship and it happened in the year 2015.

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

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

landing__outcome	2
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

- The COUNT() function is used to count the total number of rows and the WHERE clause is used to filter the date set between 2010-06-04 and 2017-03-20. The query is then grouped by the landing outcome using GROUP BY clause and the ORDER BY clause is used to order the grouped landing outcome in descending order.

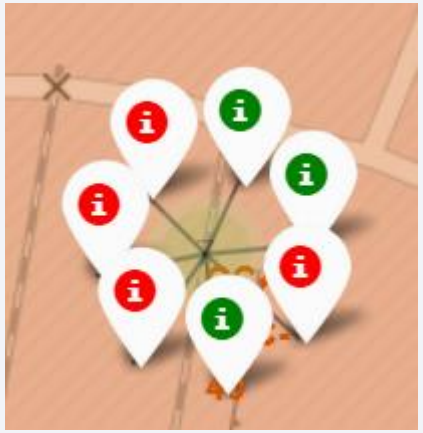
Section 3

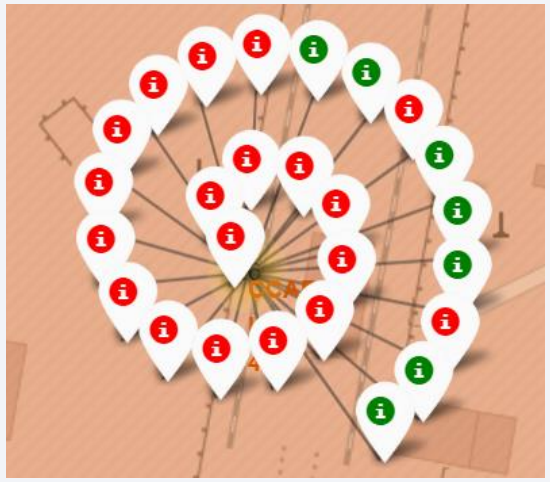
Launch Sites Proximities Analysis

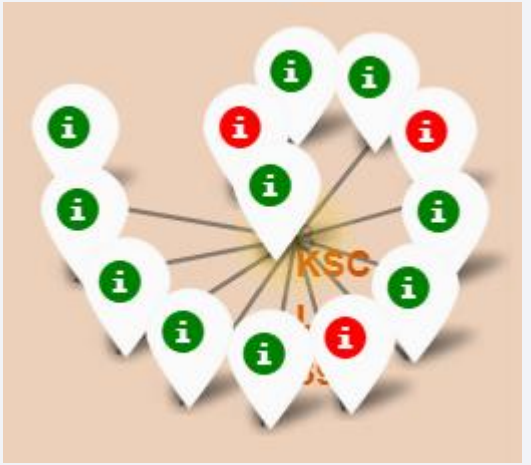
All Launch Sites on Folium Map

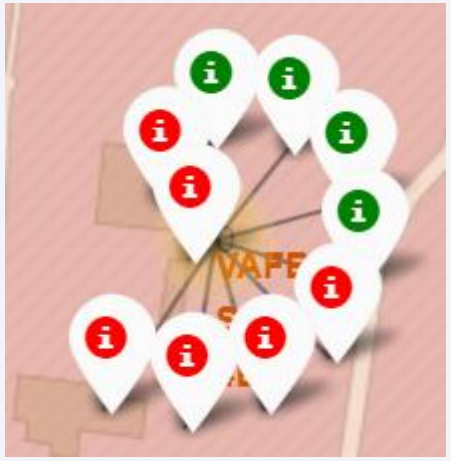
- Based on the screenshot shown above, the launch sites are located on the coasts of USA.

Color Labeled Markers on Folium Map



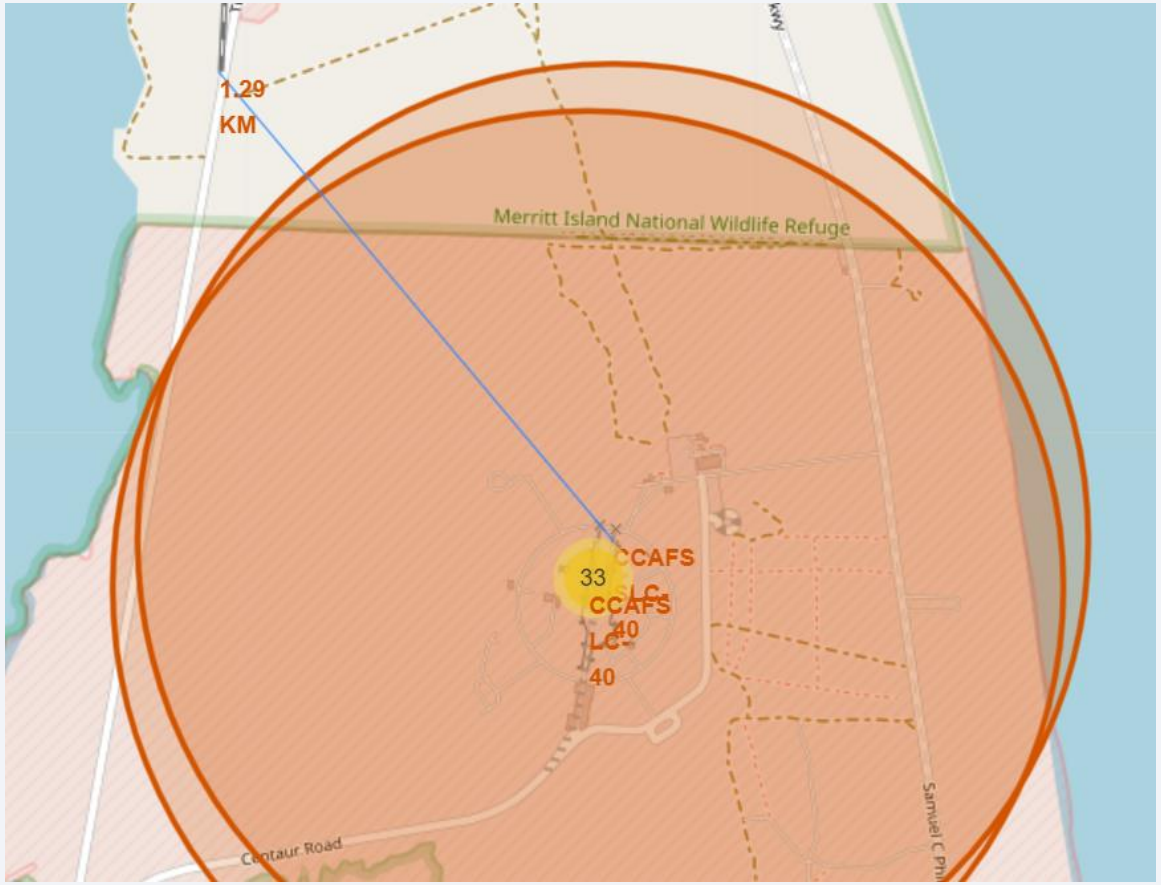




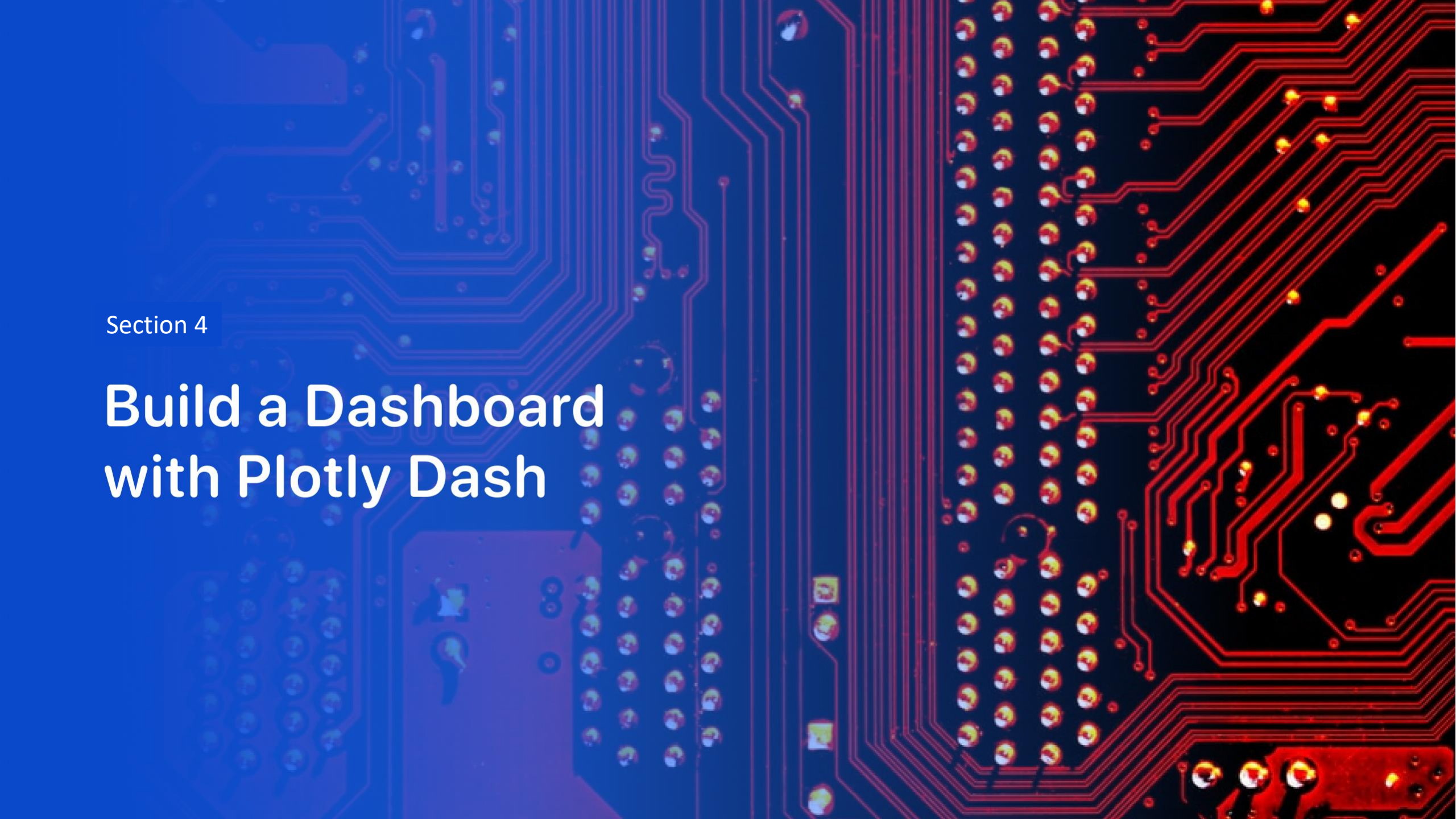


- The Green markers indicate successful launches whereas Red markers indicate failed launches in each site on the map.

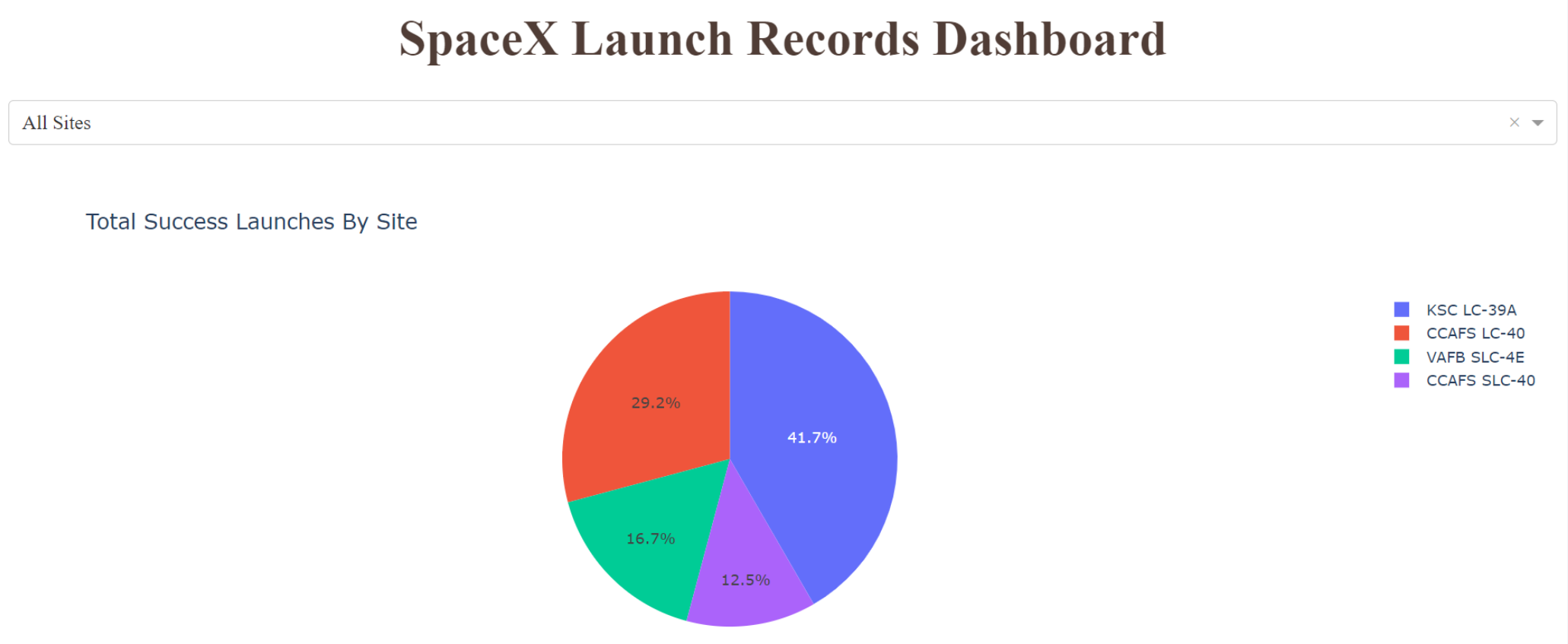
Distance of Launch Site to Other Locations



- The screenshot shows the distance between one of the sites, CCAFS SLC-40 to the nearest railway, which is about 1.29 KM away.

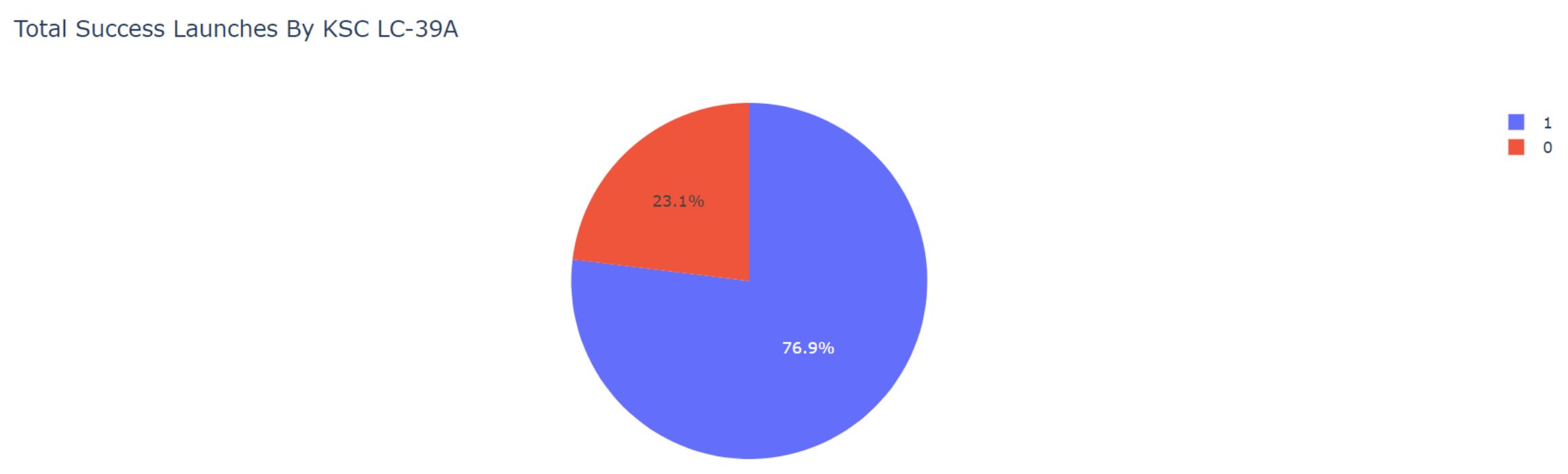

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by All Sites



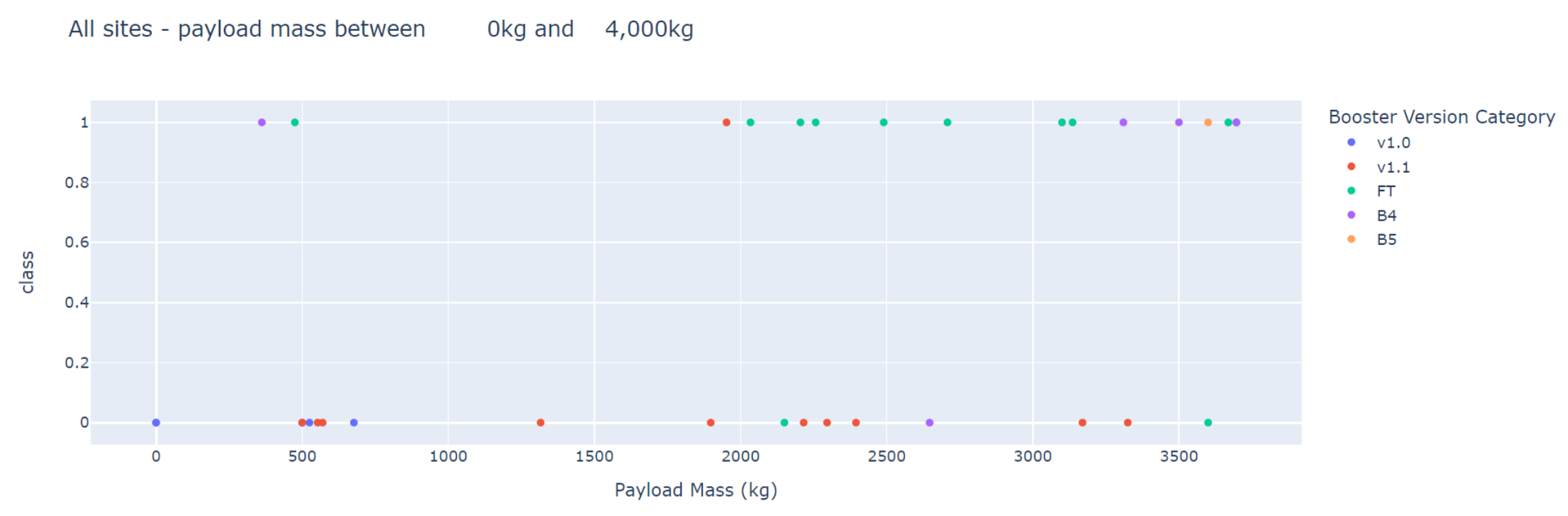
- We can clearly see that KSC LC-39A has the most successful launches out of all the sites, 41.7%.

Launch Success ratio for KSC LC-39A



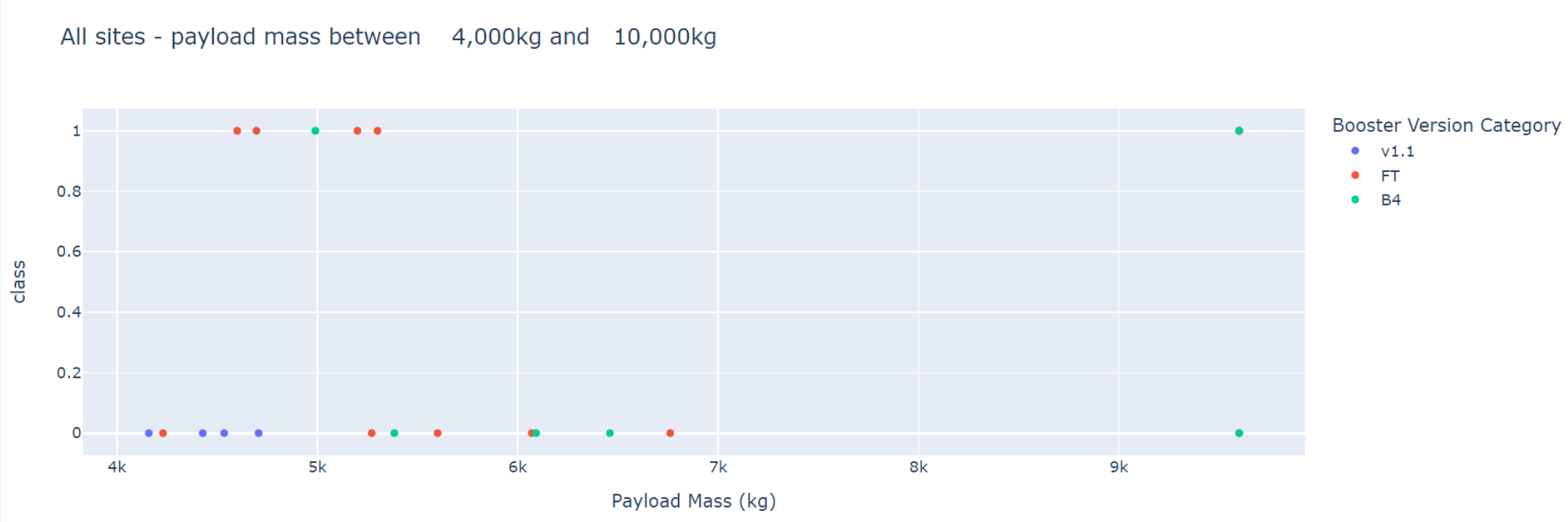
- KSC LC-39A has a 76.9% success rate and a 23.1% failure rate.

Low Payload Mass vs. Launch Outcome Scatter Plot for All Sites



- Low weighted payloads have higher success rate.

High Payload Mass vs. Launch Outcome Scatter Plot for All Sites



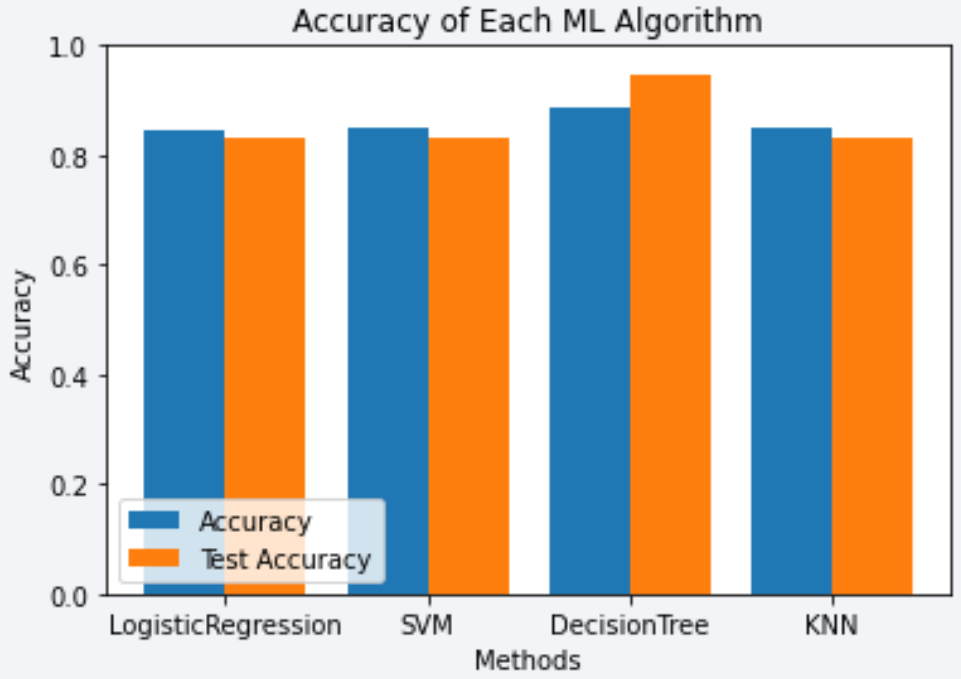
- High weighted payloads have lower success rate.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- Based on the bar chart and the table, it is clear that the best model with the highest accuracy is Decision Tree Classifier, 89.0% training accuracy and 94.0% test accuracy. The best parameters of this classifier is shown below.

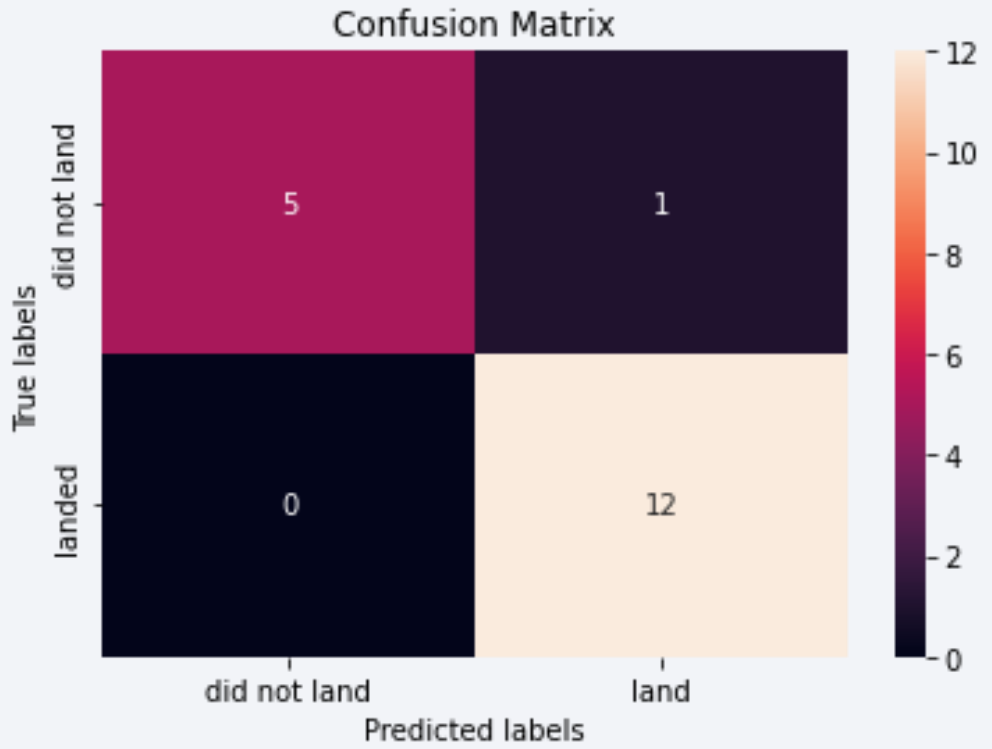


	ML Algorithms	Training Accuracy	Test Accuracy
0	LogisticRegression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	DecisionTree	0.887500	0.944444
3	KNN	0.848214	0.833333

```
The best parameters are {'criterion': 'gini', 'max_depth': 18, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 10, 'splitter':
'best'}
```

Confusion Matrix

- The confusion matrix of Decision Tree Classifier is shown here.
- Only one sample has been misclassified by the algorithm.
- This shows that there is a big number of true positive and true negative as compared to false positive and false negative.



Conclusions

- The success rate increases with an increased in flight number.
- KSC LC-39A has the most successful launches of any site.
- ES-L1, GEO, HEO, and SSO orbit types have higher success rate.
- The launch success rate has been increasing throughout the years from 2013 to 2020.
- Decision Tree Classifier can be used to predict the outcome of a landing and increase the profit.

Thank you!