IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jeffrey Budiman
July 1, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data collection using API and web scraping
  - Data wrangling
  - Exploratory Data Analysis using SQL and visualization
  - Interactive Visual Analytics using Folium
  - Predictive Analysis / Classification
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics 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 if an alternate company wants to bid against SpaceX for a rocket launch.

  The goal of this project is to make prediction if the first stage will land successfully.

- Problems you want to find answers
  - Understand which factors (variables) affect landing success
  - Use machine learning model to make prediction if first stage landing is a success

Section 1

Methodology

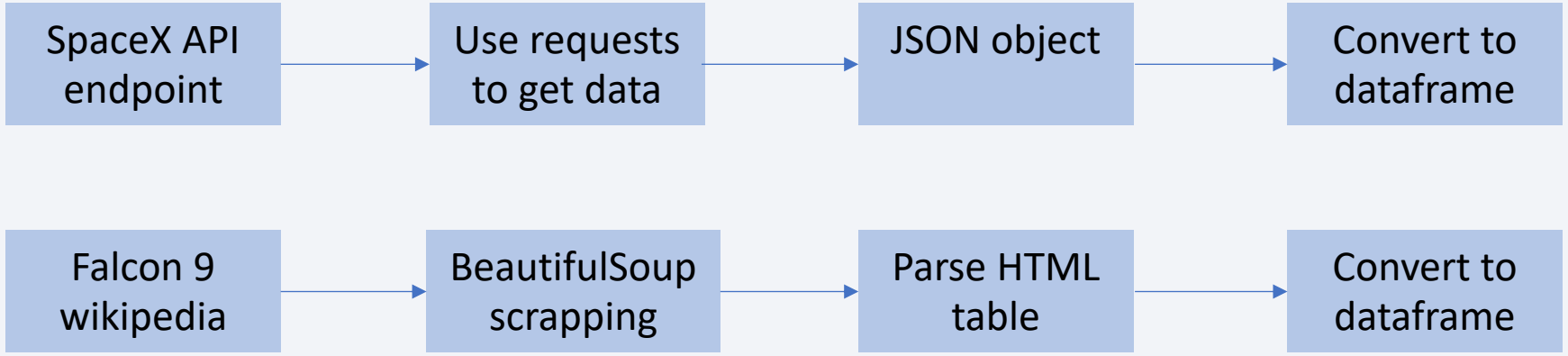
Methodology

Executive Summary

- Data collection methodology:
  - The data was collected using API (to get launches data) and Web Scraping (to get additional data on Falcon 9 launches)
- Perform data wrangling
  - The data was processed by converting the launch 'Outcome' values to class 0 (failure) and 1 (success)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Dataset is split into test and train.
  - Various machine learning model types will be built with the train data to see which one performs the best. Each will go through hyperparameter tuning using grid search to find the optimum point.
  - Accuracy score and confusion matrix will be used for model evaluation.

Data Collection

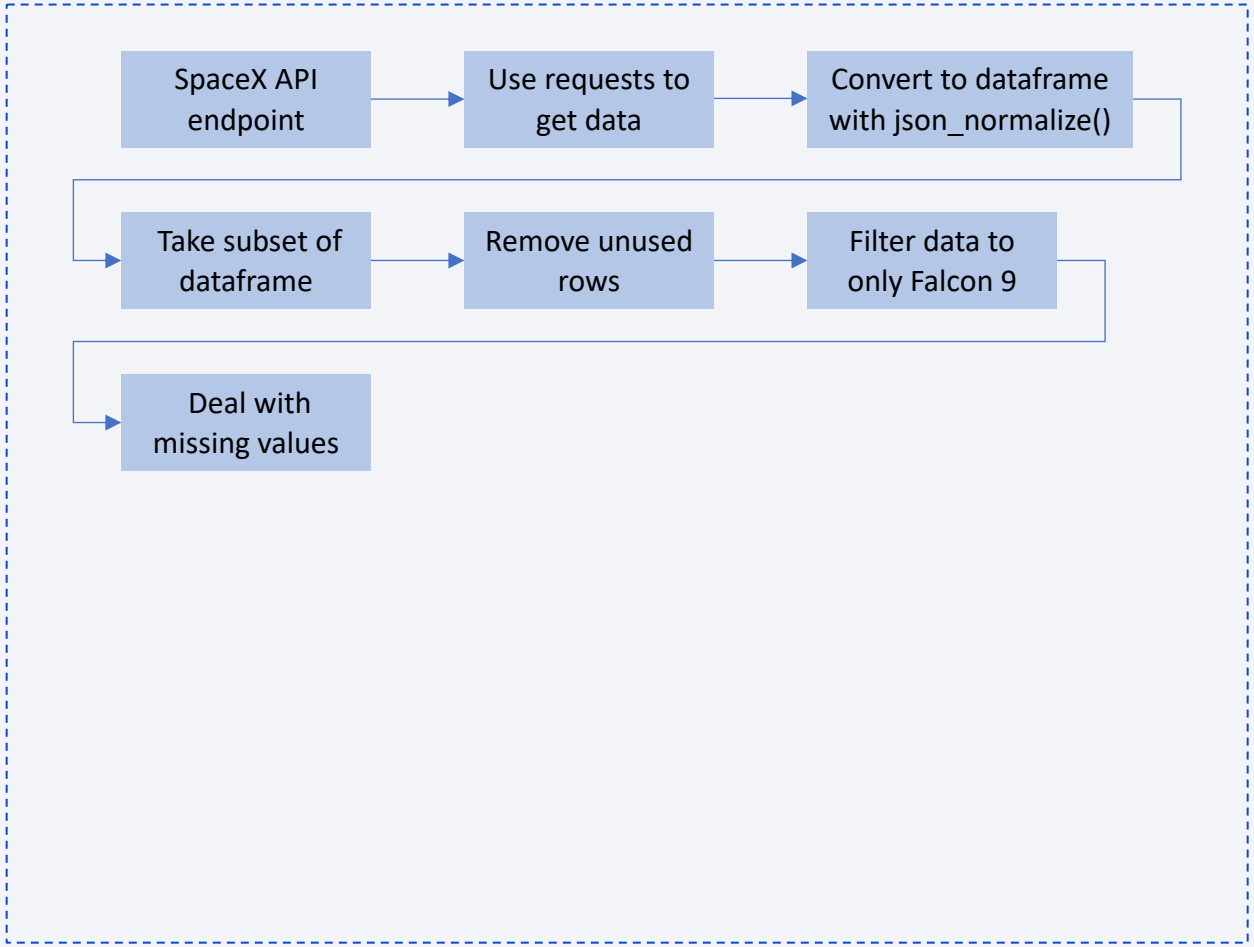
- How data sets were collected
  - Data was gathered by sending GET request to SpaceX API. The response is decoded as JSON object and converted to dataframe. Data cleaning is performed to ensure integrity. In addition, Wikipedia web scraping using BeautifulSoup was done to get Falcon 9 launch data. The scrapped HTML data is then converted to dataframe.
- Flowcharts



Data Collection – SpaceX API

- Data collection flowcharts with SpaceX REST calls is shown on the right

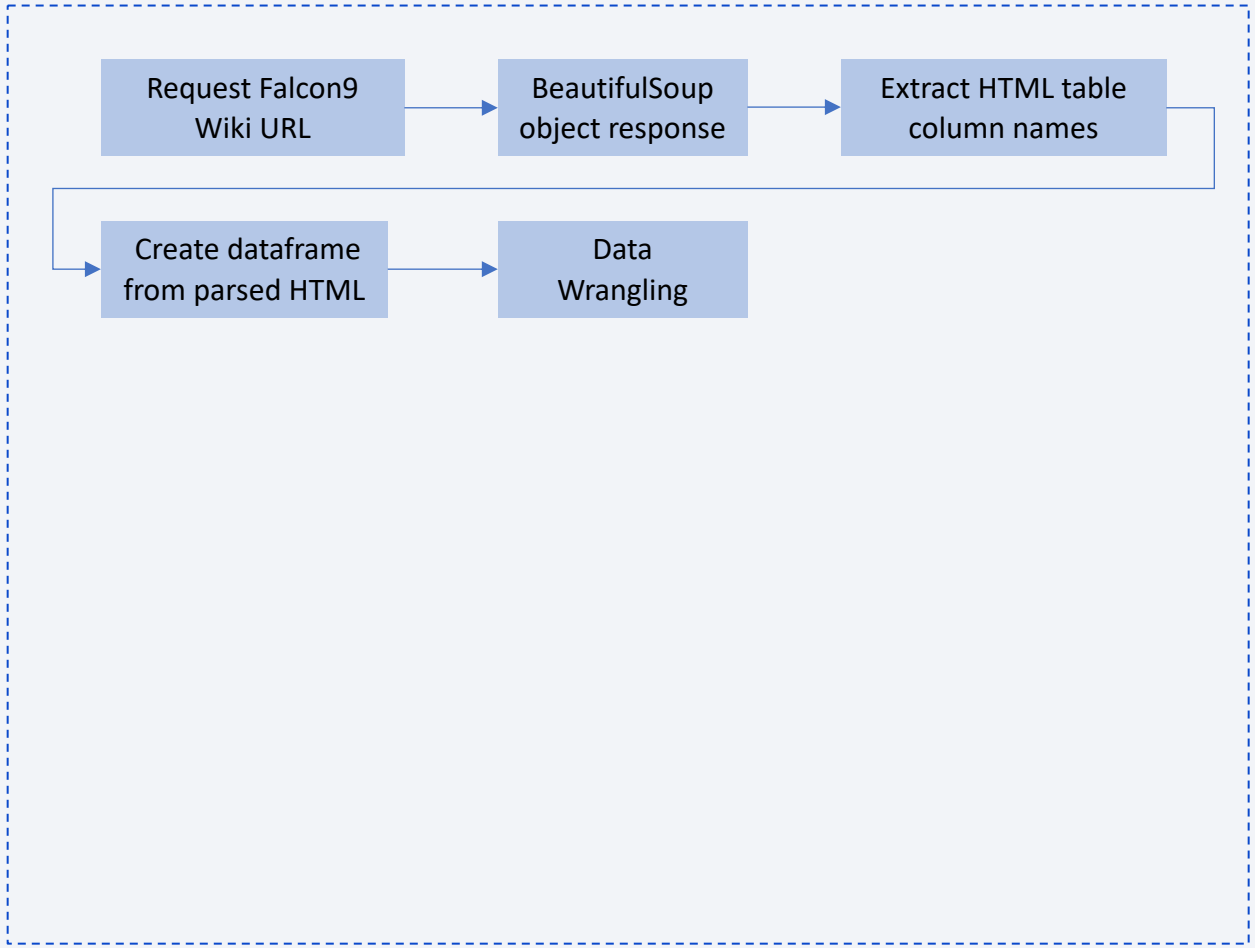
- GitHub URL of the completed SpaceX API calls notebook



Data Collection - Scraping

- Web scraping process is shown on the right

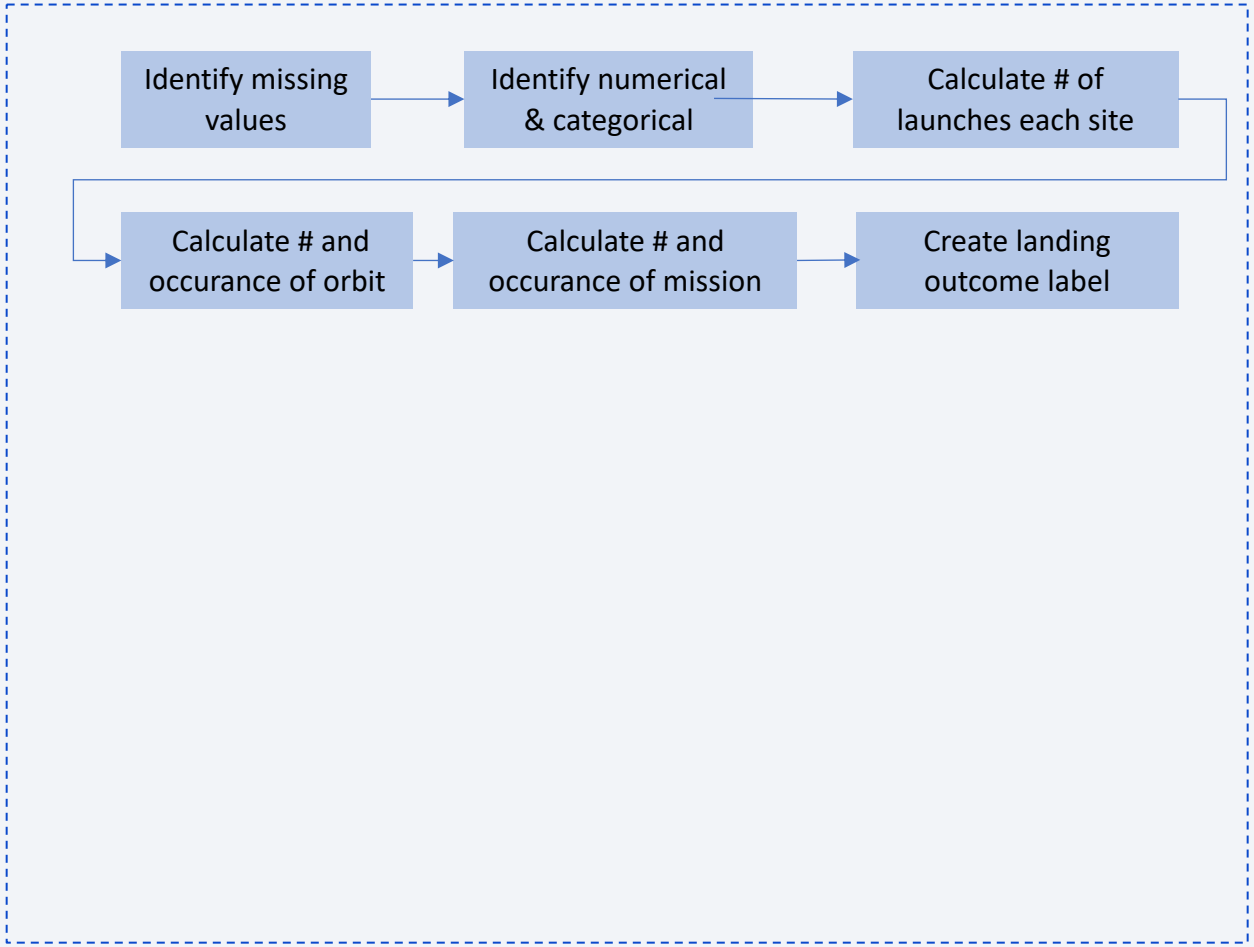
- GitHub URL of the completed web scraping notebook



Data Wrangling

- Data Wrangling process is shown on the right

- GitHub URL of the completed Data Wrangling notebook



EDA with Data Visualization

- Summarize what charts were plotted:

- Catplot to visualize the relationship between Flight Number and Payload
- Catplot to visualize the relationship between Flight Number and Launch Site
- Catplot to visualize the relationship between Payload and Launch Site
- Bar chart to visualize the relationship between success rate of each Orbit type
- Catplot to visualize the relationship between Flight Number and Orbit type
- Catplot to visualize the relationship between Payload and Orbit type
- Line chart to visualize the launch success yearly trend

- GitHub URL of the completed EDA with data visualization notebook

EDA with SQL

- Task 1: Display the names of the unique launch sites in the space mission
  SELECT DISTINCT launch_site FROM SPACEXTBL;
- Task 2: Display 5 records where launch sites begin with the string 'CCA'
  SELECT * FROM SPACEXTBL WHERE launch_site LIKE 'CCA%' LIMIT 5;
- Task 3: Display the total payload mass carried by boosters launched by NASA (CRS)
  SELECT SUM(payload_mass__kg_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE customer='NASA (CRS)';
- Task 4: Display average payload mass carried by booster version F9 v1.1
  SELECT AVG(payload_mass__kg_) AS AVG_PAYLOAD_MASS FROM SPACEXTBL WHERE booster_version='F9 v1.1';
- Task 5: List the date when the first succesful landing outcome in ground pad was acheived.
  SELECT MIN(DATE) AS first_successful_landing FROM SPACEXTBL WHERE (landing_outcome)='Success (ground pad)';
- Task 6: List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  SELECT booster_version, payload_mass__kg_, landing_outcome FROM SPACEXTBL WHERE landing_outcome='Success (drone ship)' AND (payload_mass__kg_ BETWEEN 4000 AND 6000);
- Task 7: List the total number of successful and failure mission outcomes
  SELECT mission_outcome, COUNT(mission_outcome) AS TOTAL FROM SPACEXTBL GROUP BY mission_outcome;

EDA with SQL

- Task 8: List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  SELECT DISTINCT (booster_version), (SELECT MAX(payload_mass__kg_) AS "maximum_payload_mass" FROM SPACEXTBL) FROM SPACEXTBL LIMIT 5;

- Task 9: 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.
  SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL WHERE landing_outcome LIKE '%Failure (drone ship)%' AND (DATE LIKE '2015%');

- Task 10: Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order.
  SELECT landing_outcome, COUNT(landing_outcome) AS "total" FROM SPACEXTBL WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY landing_outcome ORDER BY "total" DESC;

- GitHub URL of the completed EDA with SQL notebook

Build an Interactive Map with Folium

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map:
- folium.Circle and folium.Marker to add a highlighted circle area with a text label on a specific coordinate for each launch site on the site map.
- MarkerCluster object for simplify a map containing many markers having the same coordinate.
- MousePosition on the map to get coordinate for a mouse over a point on the map.
- Folium. PolyLine object to draw a line between a launch site to its closest city, railway and highway

- GitHub URL of completed interactive map with Folium map

Build a Dashboard with Plotly Dash

- Summarize what plots/graphs and interactions you have added to a dashboard
  This dashboard application contains input components such as a dropdown list and a range slider to interact with a pie chart and a scatter point chart.
- A launch Site Drop-down Input Component.
  There are four different launch sites and a dropdown menu let us select different launch sites.
- A callback function to render success-pie-chart based on selected site dropdown.
  The general idea of this callback function is to get the selected launch site from site-dropdown and render a pie chart visualizing launch success counts.
- A range Slider to Select Payload.
  The Slider is to be able to easily select different payload range and see if we can identify some visual patterns

- GitHub URL of completed Plotly Dash lab

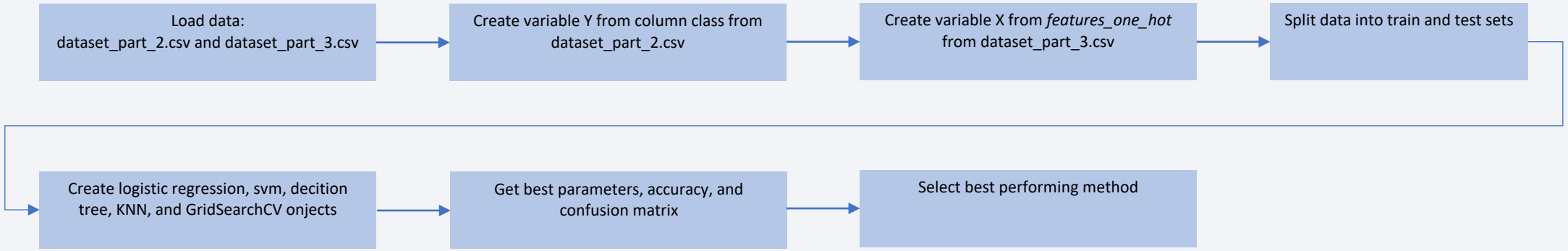
Predictive Analysis (Classification)

Summary on how the best performing classification model was built, evaluated, improved, and found:

- Creation of a NumPy array from the column Class in data.
- Data standardization.
- Use of the function train_test_split to split the data X and Y into training and test data.
- Searching for the best Hyperparameters for Logistic Regression, SVM, Decision Tree and KNN classifiers.
- Searching for the method that performs best using test data

Predictive Analysis (Classification)

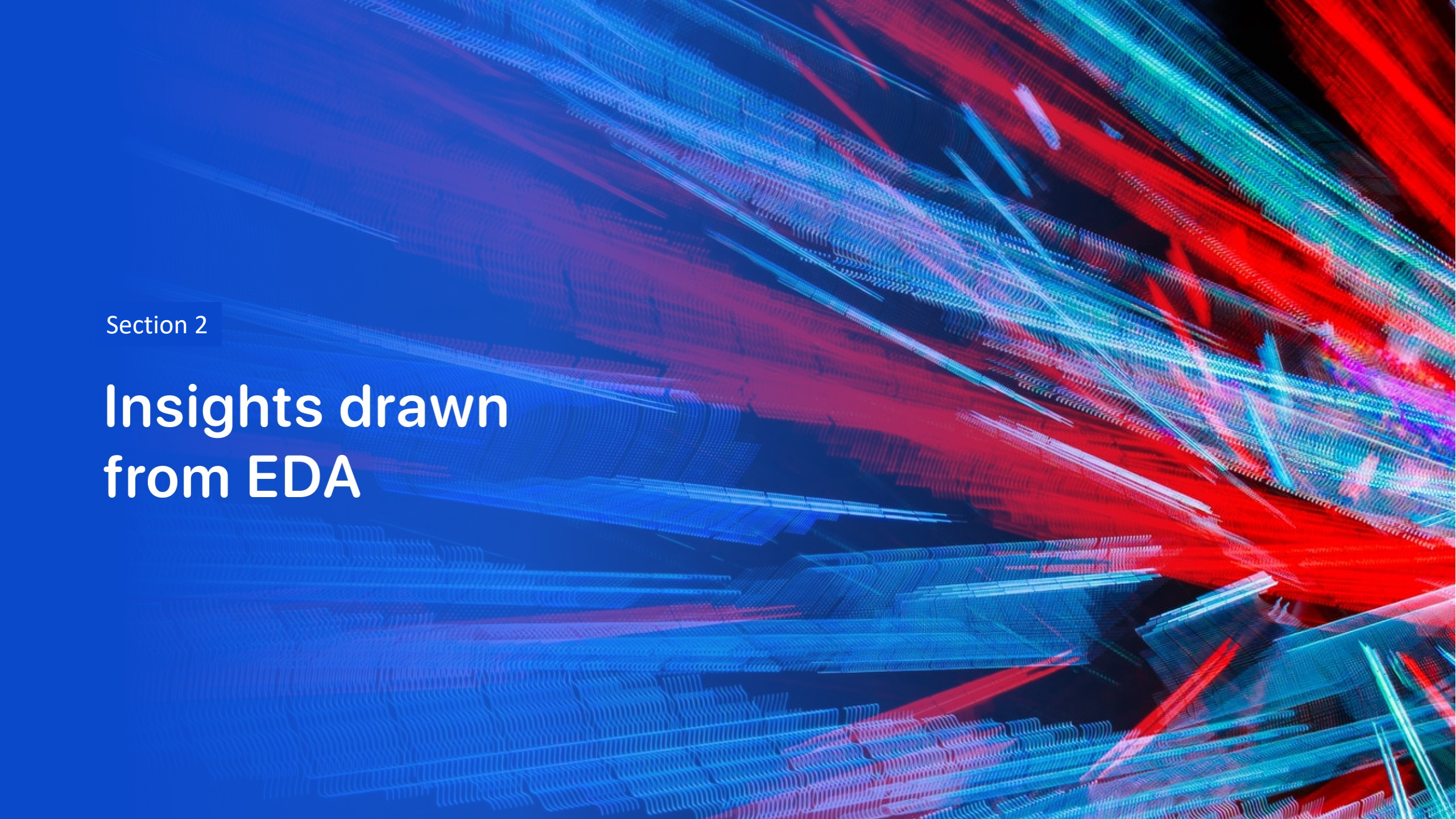
- Flowchart



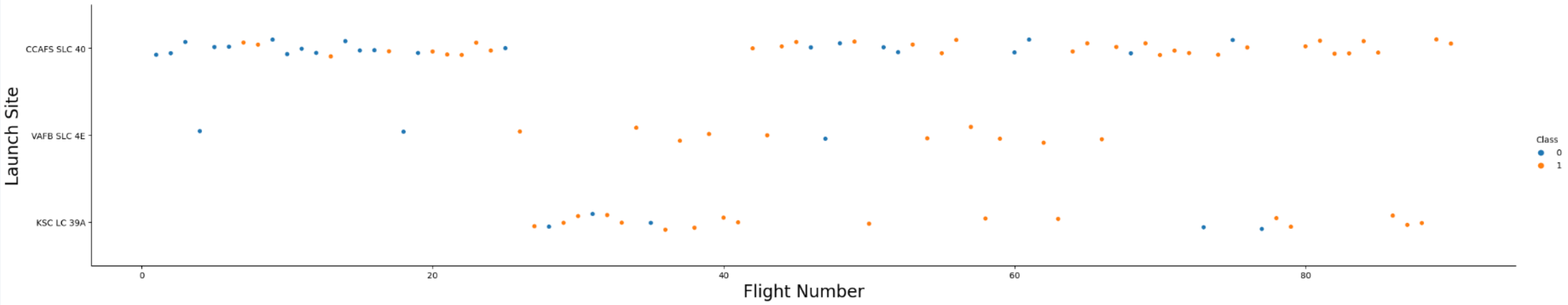
- GitHub URL of completed predictive analysis lab

Results

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

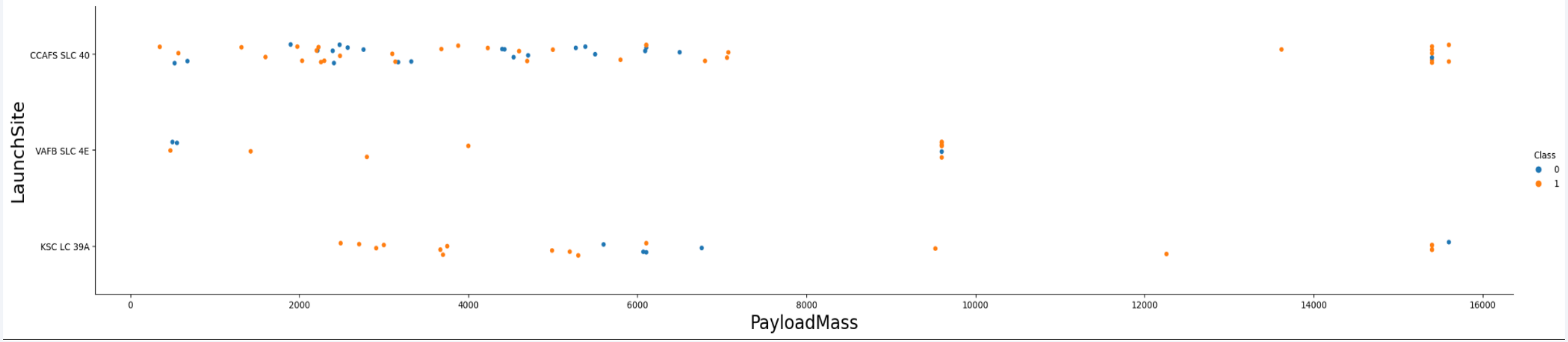

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



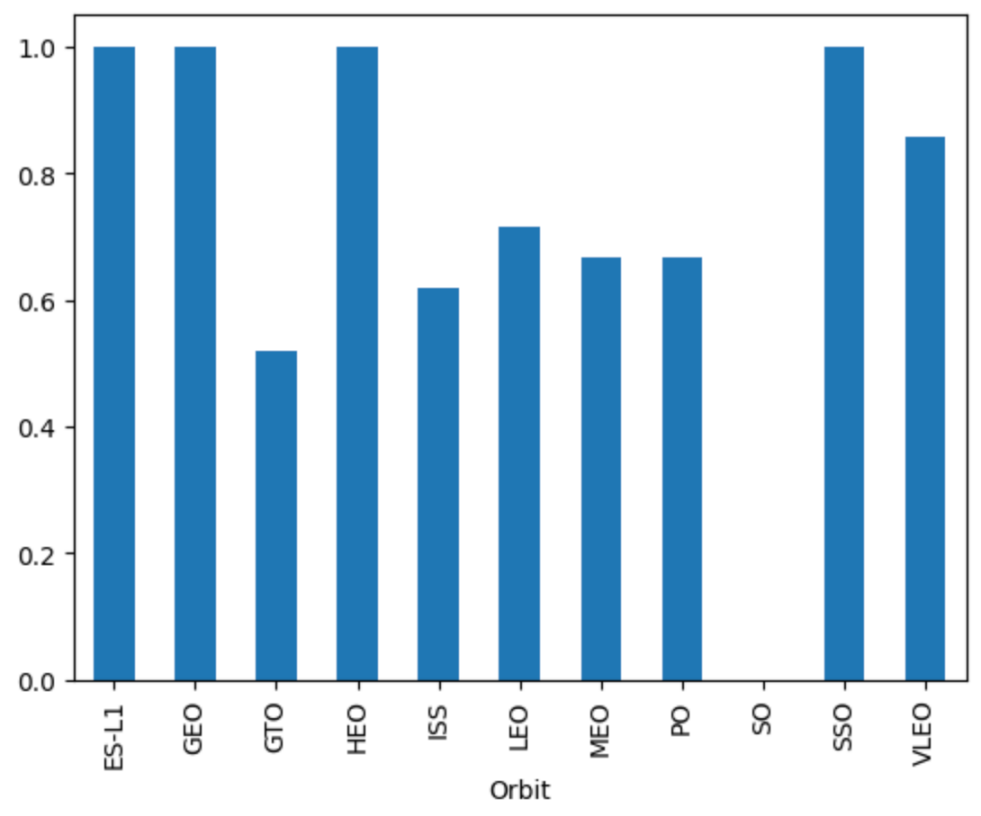
- From the scatter plot, the greater the flight number, the higher the success rate for all launch sites. This means as improvement is made in progression, later flights will have higher success chance.

Payload vs. Launch Site



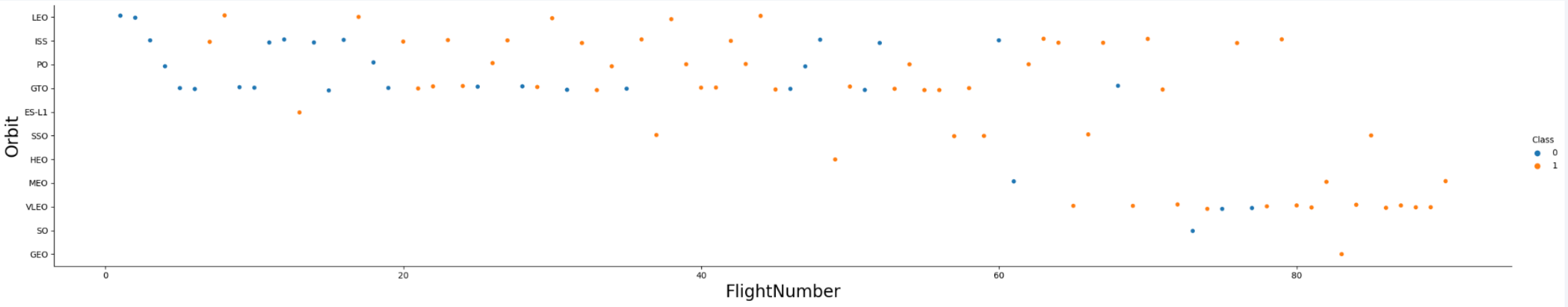
- CCAFS SLC has launched rockets less than 7500kg and more than 13000kg payloadmass but not in between
- In VAFB-SLC launch site there are no rockets launched for heavy payloadmass (greater than 10000 kg)
- In KSC LC launch site there are no rockets launched for lower payloadmass (less than 2500kg

Success Rate vs. Orbit Type



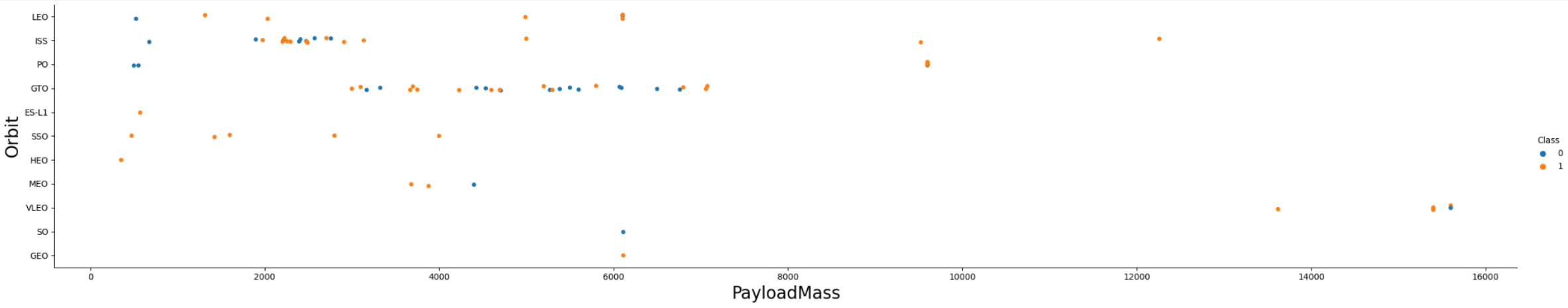
- The orbits with highest success rates are ES-L1, GEO, HEO and SSO

Flight Number vs. Orbit Type



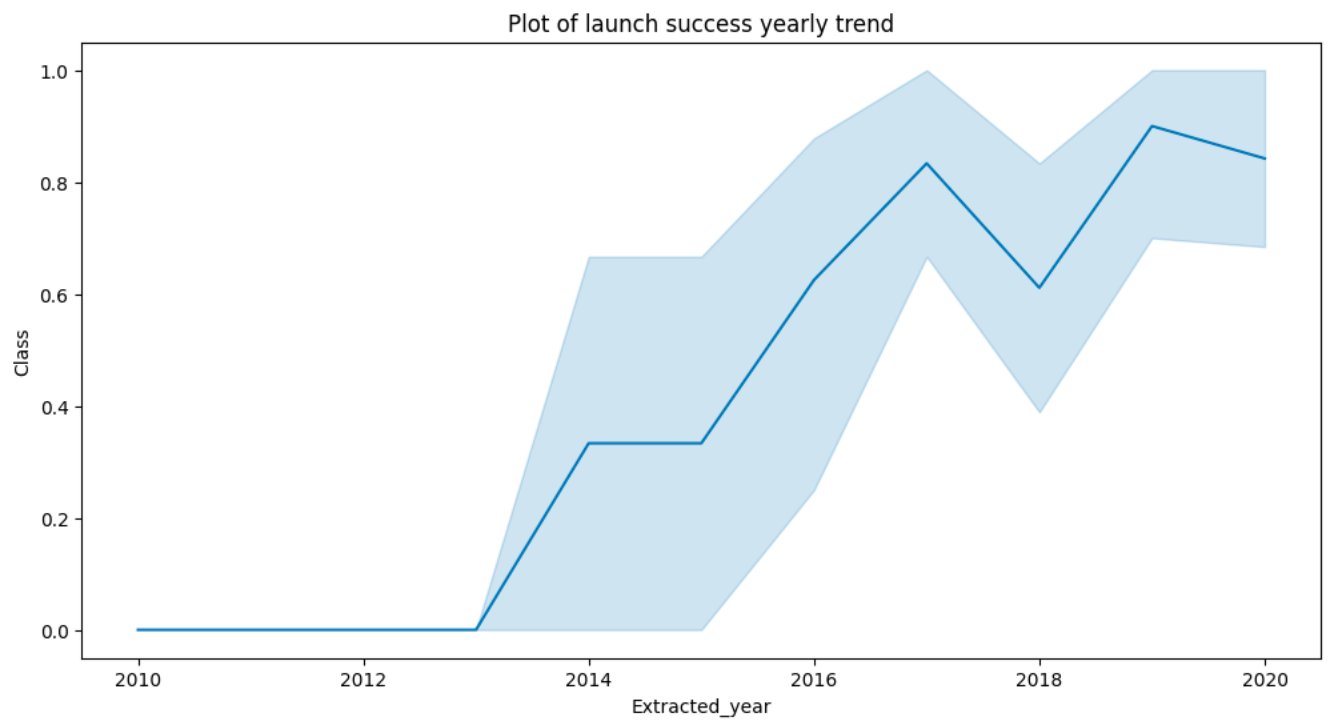
- For LEO, higher flight number has greater success rate

Payload vs. Orbit Type



- For LEO, ISS, and PO have greater success rates at higher payload mass

Launch Success Yearly Trend



- Success rate trend increases since 2013 until 2020

All Launch Site Names

Task 1

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

In [10]:
```
%sql SELECT DISTINCT launch_site FROM SPACEXTBL;
```

File display

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

Out[10]:

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

- DISTINCT is used to get distinct names of the launch sites

Launch Site Names Begin with 'CCA'

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

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

Python

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

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

- LIKE ‘CCA’ is used to filter the SELECT results that begins with ‘CCA’ and LIMIT 5 is used to display five records.

Total Payload Mass

Task 3

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

In [13]:
```
%sql SELECT SUM(payload_mass__kg_) AS TOTAL_PAYLOAD_MASS FROM SPACEXTBL WHERE customer='NASA (CRS)';
```

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

Out[13]:

TOTAL_PAYLOAD_MASS
45596

- SUM is used to calculate total payload mass with WHERE customer='NASA (CRS)' as condition to select boosters launched by NASA (CRS)

Average Payload Mass by F9 v1.1

Task 4

File display

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

```
In [15]: %sql SELECT AVG(payload_mass__kg_) AS AVG_PAYLOAD_MASS FROM SPACEXTBL WHERE booster_version='F9 v1.1';
 * sqlite:///my_data1.db
Done.
```

Out[15]:

AVG_PAYLOAD_MASS
2928.4

- AVG is used to calculate average payload mass with booster version specified in the WHERE condition

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [17]:
```
%sql SELECT MIN(DATE) AS first_successful_landing FROM SPACEXTBL WHERE (landing_outcome)='Success (ground pad)';
```

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

Out[17]:

first_successful_landing
2015-12-22

- MIN is used to get the first date of the successful landing outcome which is specified in the WHERE condition.

Successful Drone Ship Landing with Payload between 4000 and 6000

File display

Task 6

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

In [18]:
```
%sql SELECT booster_version, payload_mass__kg_, landing_outcome FROM SPACEXTBL WHERE landing_outcome='Success (dr
```

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

Out[18]:

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- `%sql SELECT booster_version, payload_mass__kg_, landing_outcome FROM SPACEXTBL WHERE landing_outcome='Success (drone ship)' AND (payload_mass__kg_ BETWEEN 4000 AND 6000);`
- Payload range is specified in the WHERE condition

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [19]:
```
%sql SELECT mission_outcome, COUNT(mission_outcome) AS TOTAL FROM SPACEXTBL GROUP BY mission_outcome;
```

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

Out[19]:

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

- COUNT is used to calculate the total # of missions and GROUP BY is used to categorized / split the outcome based on success or failure

Boosters Carried Maximum Payload

Task 8

File display

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

In [24]:
```
%sql SELECT DISTINCT (booster_version), (SELECT MAX(payload_mass__kg_) AS "maximum_payload_mass" FROM SPACEXTBL)
```

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

Out[24]:

Booster_Version	(SELECT MAX(payload_mass__kg_) AS "maximum_payload_mass" FROM SPACEXTBL)
F9 v1.0 B0003	15600
F9 v1.0 B0004	15600
F9 v1.0 B0005	15600
F9 v1.0 B0006	15600
F9 v1.0 B0007	15600

- MAX is used to select maximum payload mass and then DISTINCT (booster_version) is used to list all distinct / unique booster version names among all maximum payload mass.

2015 Launch Records

Task 9

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.

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

In [25]:
```
%sql SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL WHERE landing_outcome LIKE '%Failu
```

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

Out[25]:

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

- %sql SELECT landing_outcome, booster_version, launch_site, DATE FROM SPACEXTBL WHERE landing_outcome LIKE '%Failure (drone ship)%' AND (DATE LIKE '2015%');
- Conditions inside WHERE statement gets the Failure outcome in 2015. The columns in SELECT then displays the outcomes

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

Task 10

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

In [26]:
```
%sql SELECT landing_outcome, COUNT(landing_outcome) AS "total" FROM SPACEXTBL WHERE (DATE BETWEEN '2010-06-04' AN
```

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

Out[26]:

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

File display

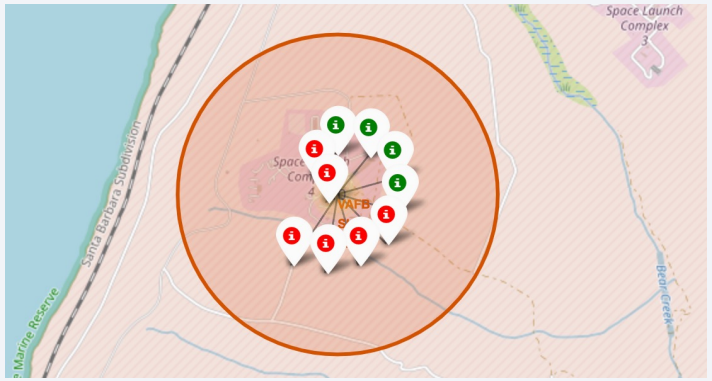
- %sql SELECT landing_outcome, COUNT(landing_outcome) AS "total" FROM SPACEXTBL WHERE (DATE BETWEEN '2010-06-04' AND '2017-03-20') GROUP BY landing_outcome ORDER BY "total" DESC;
- Conditions inside WHERE statement gets the desired date range. Within this range, each landing outcome type is counted. Then they are sorted using ORDER BY statement.


Section 3
Launch Sites
Proximities Analysis

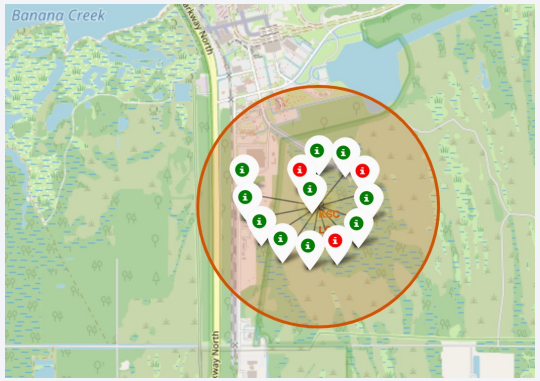
All Launch Sites

- All launch sites are strategically located at the coastal area.

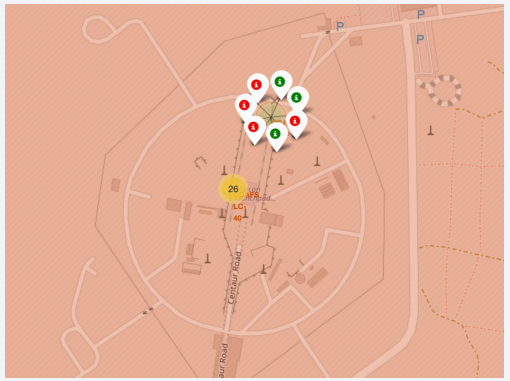
Launch sites with success / failure color markers



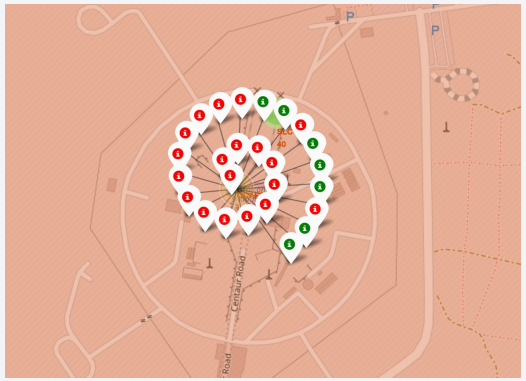
California sites



Florida sites



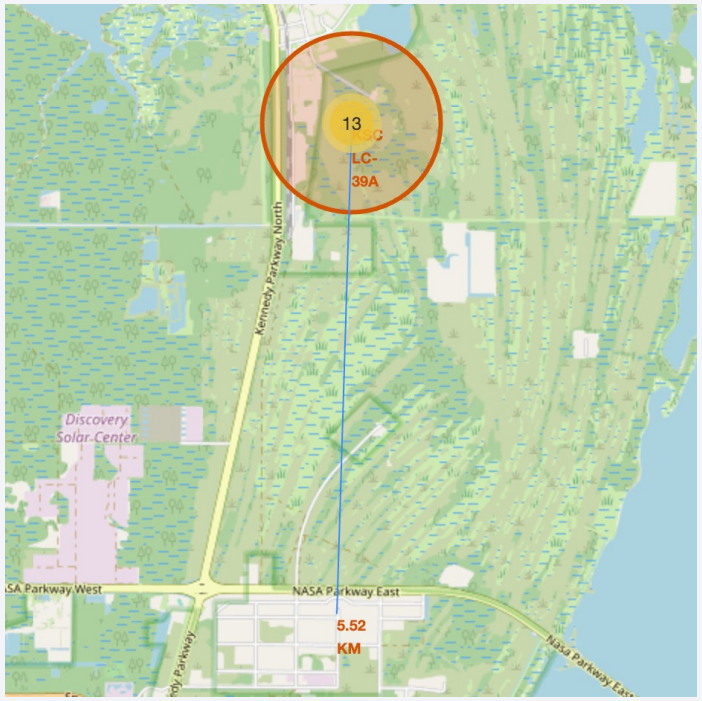
Florida sites



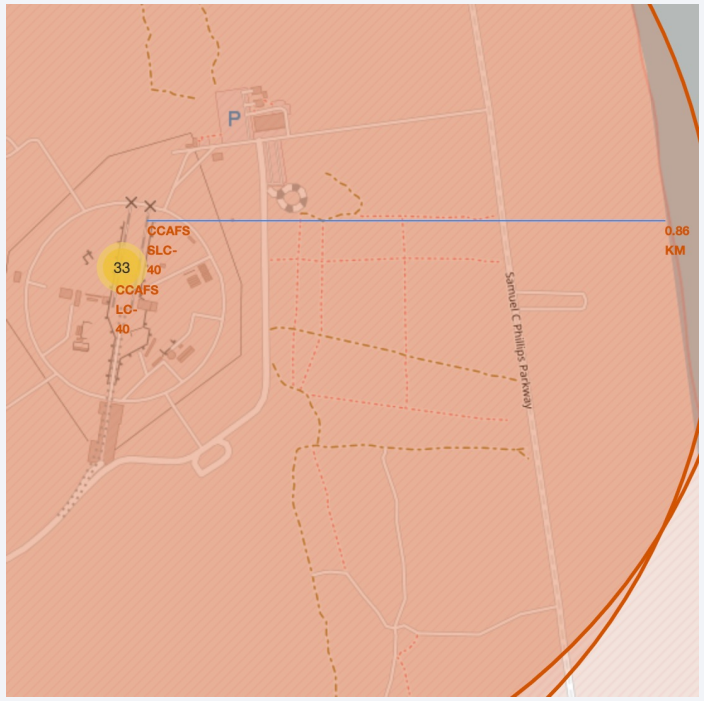
Florida sites

- Green color indicates successful sites and Red indicates failed sites

Launch site distance to proximity

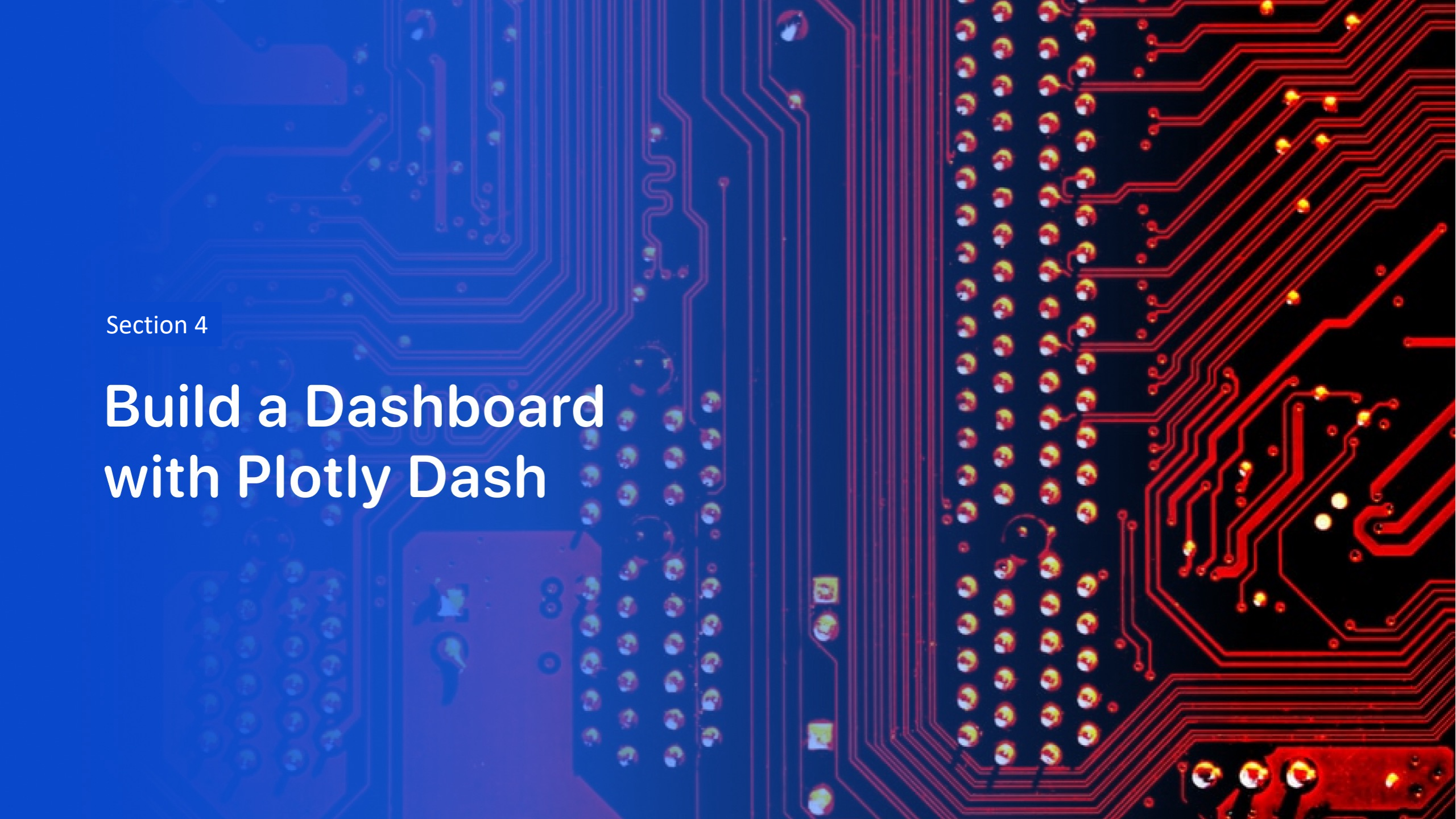


Distance to city

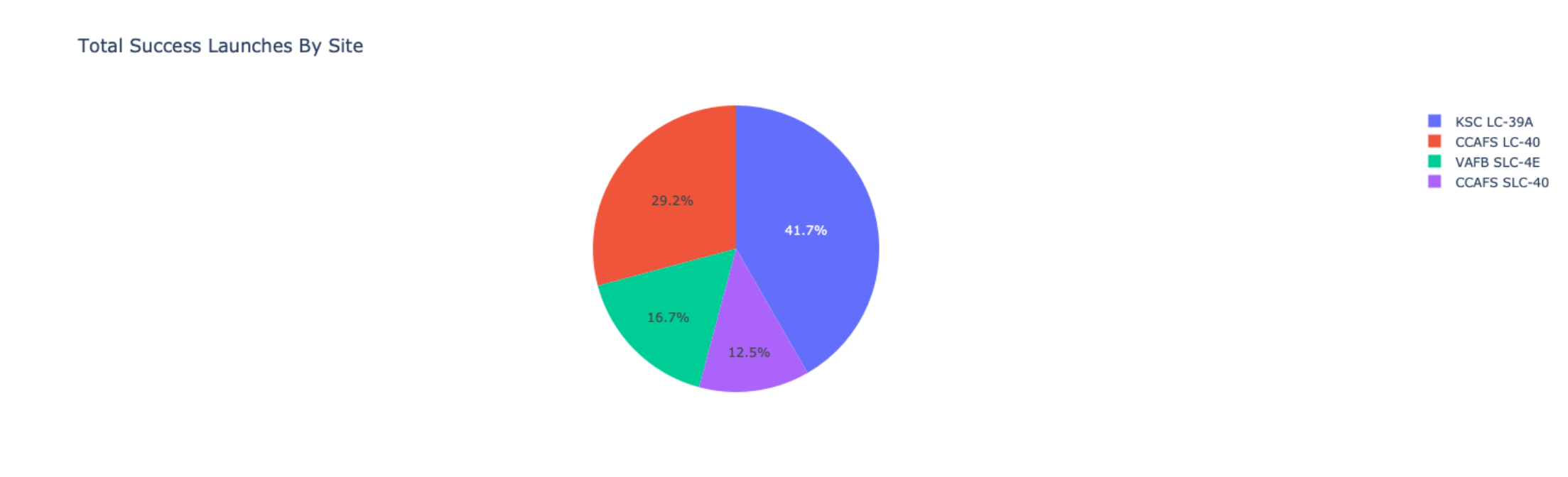


Distance to coastline

- The left screenshot shows a launch site keeps a distance to a city while the right screenshot shows a launch site is close to the coastline

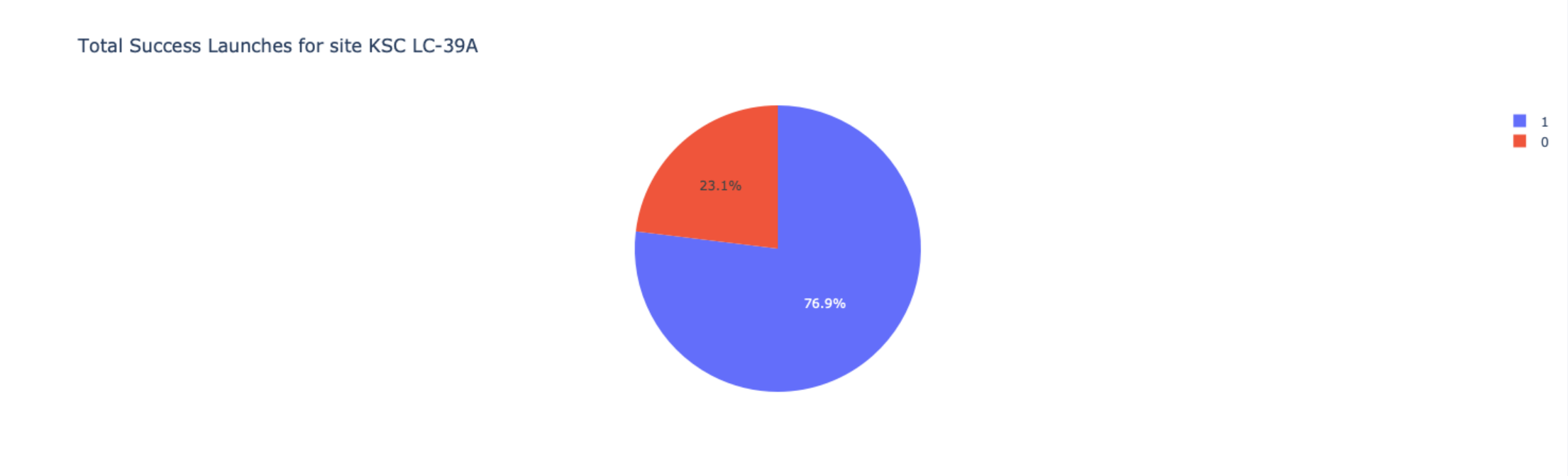

Section 4
Build a Dashboard with Plotly Dash

Success percentage by site



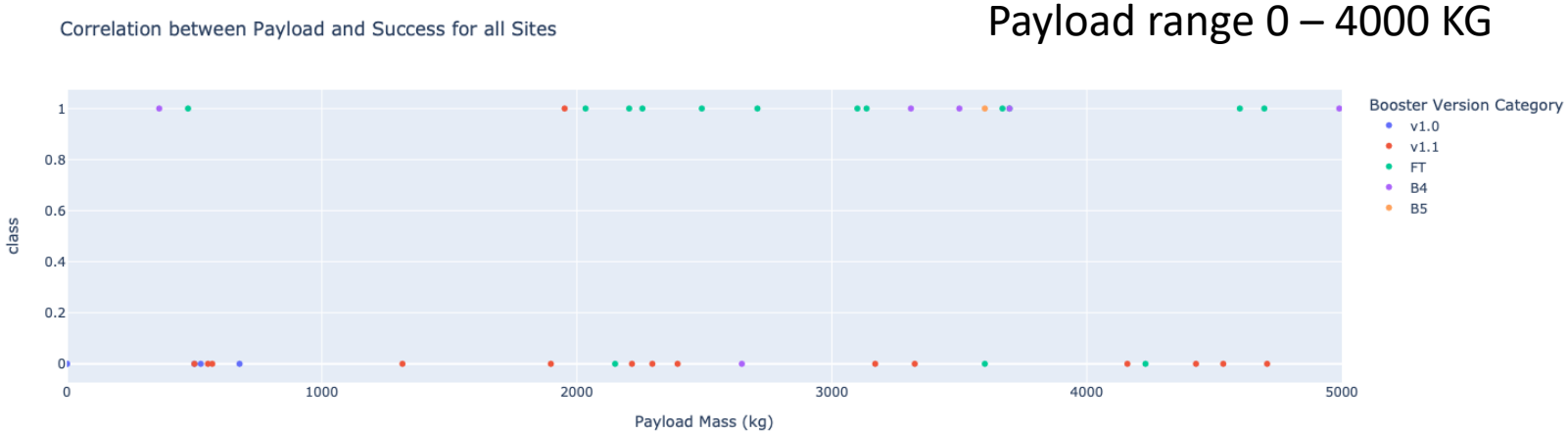
- KSC LC-39A has the most successes among all sites

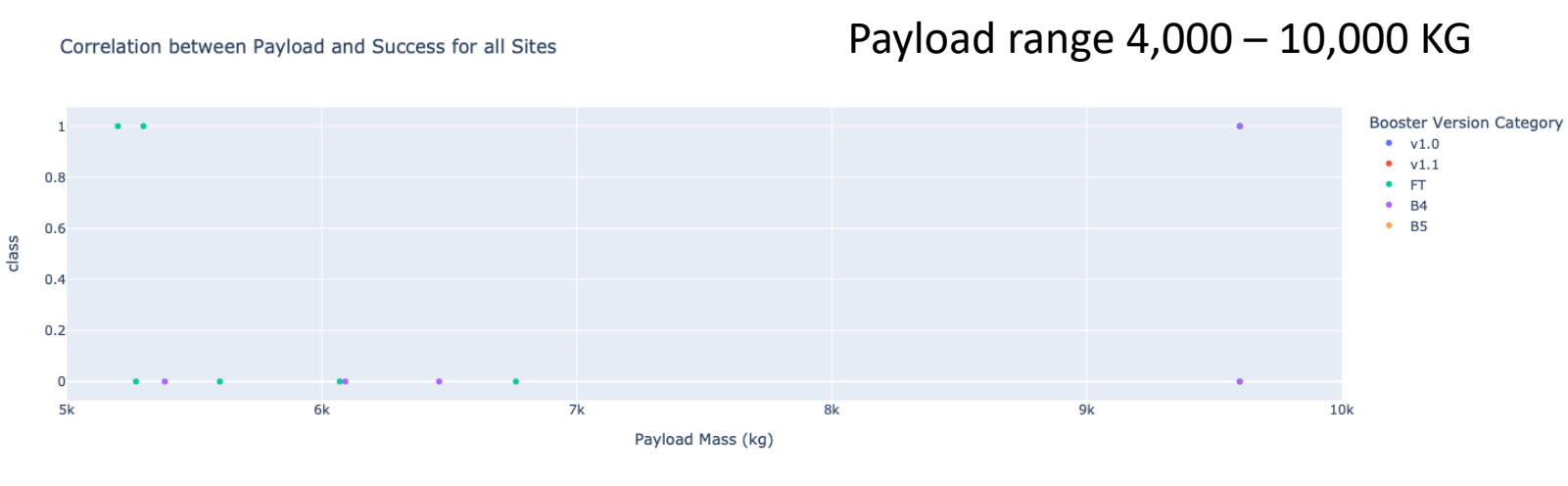
Chart for the launch site with highest launch success ratio



- KSC LC-39A has 76.9% success rate

Payload vs Launch Outcome scatter plot for all sites



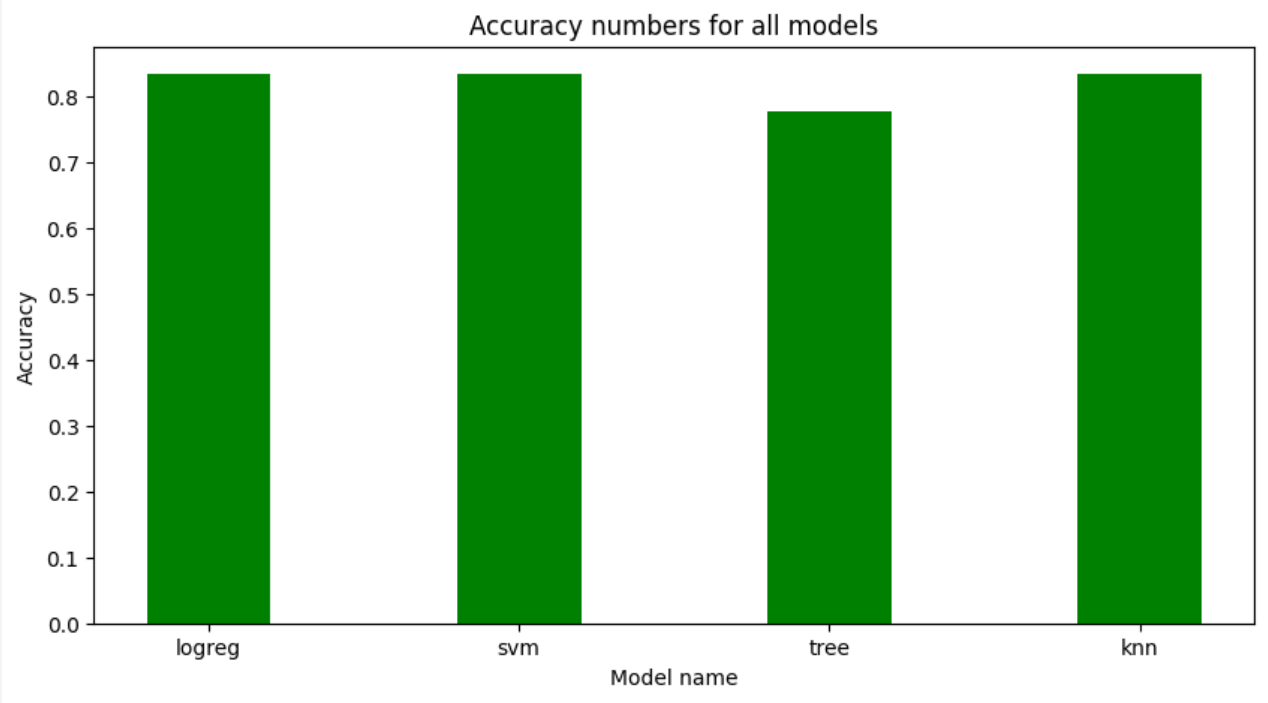


- The success rate at lower payload range is higher than at higher payload range

Section 5

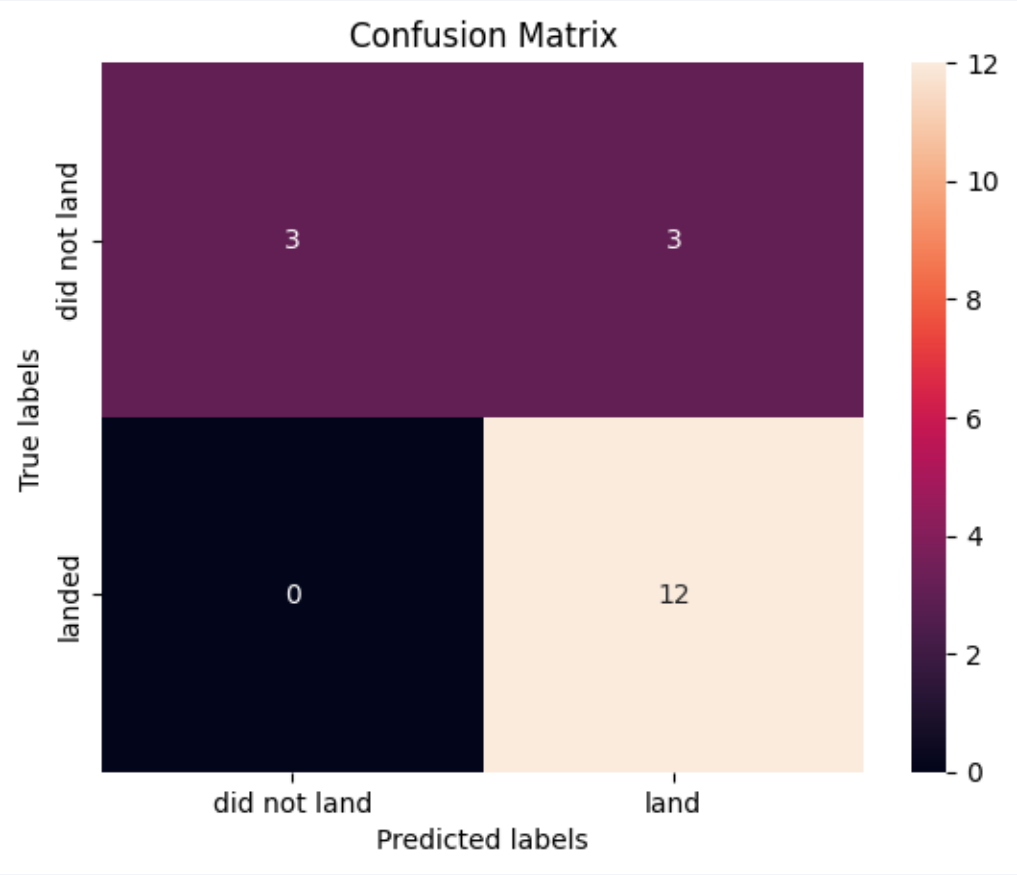
Predictive Analysis (Classification)

Classification Accuracy



- In general, the accuracy numbers are fairly close for all models due to relatively small dataset

Confusion Matrix



- All models have identical confusion matrices

Conclusions

- The results are practically the same. This is because the dataset is small and has lesser values.
- The correctly predicted data points might be very easy for all models to predict and the incorrectly predicted data might be very hard for all models to predict.
- By using our machine learning model, we can predict if the first stage of our competitor will land and determine the cost of a launch.

Appendix

- GitHub repository link

Thank you!