IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Travis Deck
20240227

Outline

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

Executive Summary

- Summary of Results:
  - Exploratory Data Analysis
  - Analytical Dashboards
  - Predictive Analytics Results
- Methodologies
  - Data Collection via API
  - Data Collection using Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis through Data Visualization
  - Interactive Visual Analytics using Folium
  - Machine Learning Prediction

Introduction

- Project Background and Context

  SpaceX promotes Falcon 9 rocket launches on its website at a price of $62 million, significantly lower than other providers, whose costs can reach upwards of $165 million per launch. The primary reason for this substantial cost difference lies in SpaceX's ability to reuse the first stage of the rocket. Hence, predicting the first stage's landing success can directly influence the launch cost estimation. This predictive capability becomes particularly valuable when competing against SpaceX for rocket launch contracts. The project aims to develop a machine learning pipeline specifically designed to forecast the first stage's landing outcome.

- Problems Seeking Answers

  What are the determining factors behind a successful rocket landing?

  The intricate interplay among various factors influencing the likelihood of a successful first stage landing.

  What operational conditions must be met to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

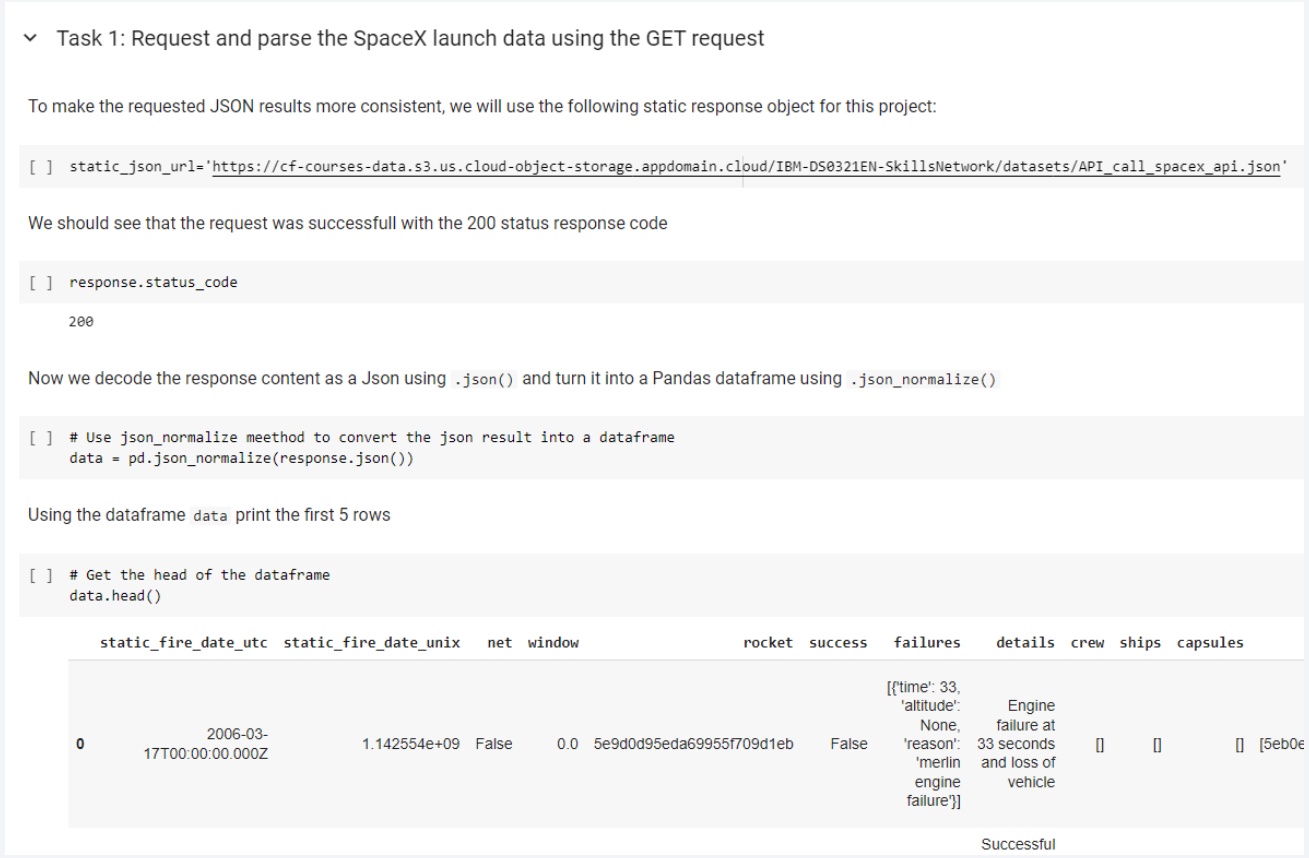
- Data collection methodology:
  - Data was collected using SpaceX queries and Wikipedia web scraping.
- Perform data wrangling
  - Normal ETL, standardization and one hot encoding to categorical data sets
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, fit, refine, and evaluate classification models

Data Collection

- Data collection involved:
  - Utilizing a GET request to access the SpaceX API.
  - Decoding response content into JSON format using the .json() function.
  - Transforming JSON data into a pandas dataframe using .json_normalize().
- Data cleaning steps included:
  - Identifying and addressing missing values as necessary.
- Web scraping from Wikipedia was performed:
  - Objective: Retrieve Falcon 9 launch records.
  - Tools used: BeautifulSoup.
  - Process: Extracted launch records presented as an HTML table, parsed it, and converted it into a pandas dataframe for analysis.

Data Collection – SpaceX API

- We utilized the GET request on the SpaceX API to gather data, processed the requested information, and conducted initial data cleaning, as well as basic wrangling and formatting tasks.
- GitHub URL of the completed SpaceX API calls



Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

```
[ ] static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
```

We should see that the request was successfull with the 200 status response code

```
[ ] response.status_code
```

```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

```
[ ] # Use json_normalize meethod to convert the json result into a dataframe
    data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

```
[ ] # Get the head of the dataframe
    data.head()
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e

Successful

Data Collection - Scraping

- We employed web scraping using BeautifulSoup to extract Falcon 9 launch records. After parsing the table, we transformed it into a pandas dataframe.
- GitHub URL of the completed web scraping notebook

TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [ ]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
data  = requests.get(static_url).text
```

In [ ]:
```
type(data)
```

Out[ ]:
```
str
```

Create a `BeautifulSoup` object from the HTML `response`

In [ ]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data, 'html5lib')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [ ]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Data Wrangling

- We conducted exploratory data analysis to identify the training labels. This involved calculating the number of launches at each site, as well as the frequency of each orbit. Additionally, we generated a landing outcome label based on the outcome column and exported the results to a CSV file.
- GitHub URL of completed data wrangling related notebook

Out[ ]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
5	6	2014-01-06	Falcon 9	3325.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
6	7	2014-04-18	Falcon 9	2296.000000	ISS	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0
7	8	2014-07-14	Falcon 9	1316.000000	LEO	CCAFS SLC 40	True Ocean	1	False	False	True	NaN	1.0
8	9	2014-08-05	Falcon 9	4535.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0
9	10	2014-09-07	Falcon 9	4428.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0

We can use the following line of code to determine the success rate:

In [ ]:
```
df["Class"].mean()
```

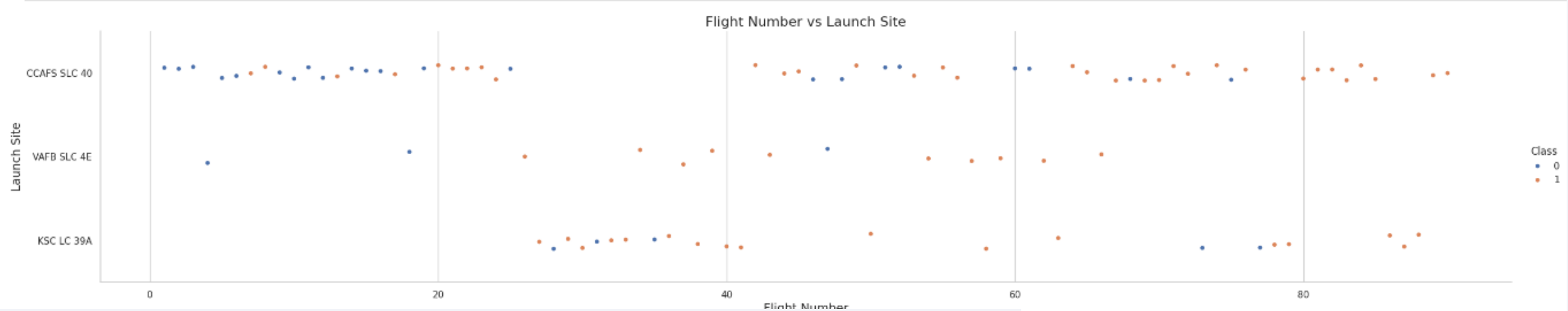
Out[ ]:
```
1.2222333334445679e+81
```

We can now export it to a CSV for the next section,but to make the answers consistent, in the next lab we will provide data in a pre-selected date range.

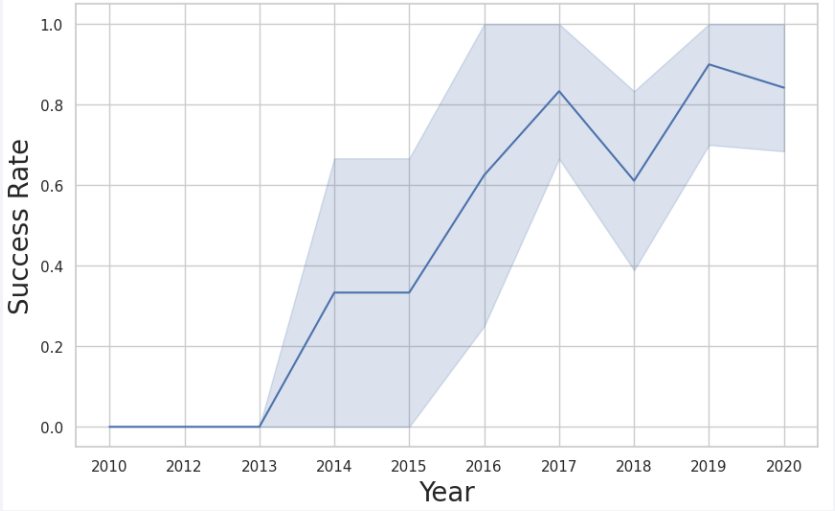
```
df.to_csv("dataset_part_2.csv", index=False)
```

In [ ]:
```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization



- We delved into the data by creating visualizations to examine various relationships, including the correlation between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type, as well as the yearly trend in launch success.
- GitHub URL of your completed EDA with data visualization notebook



EDA with SQL

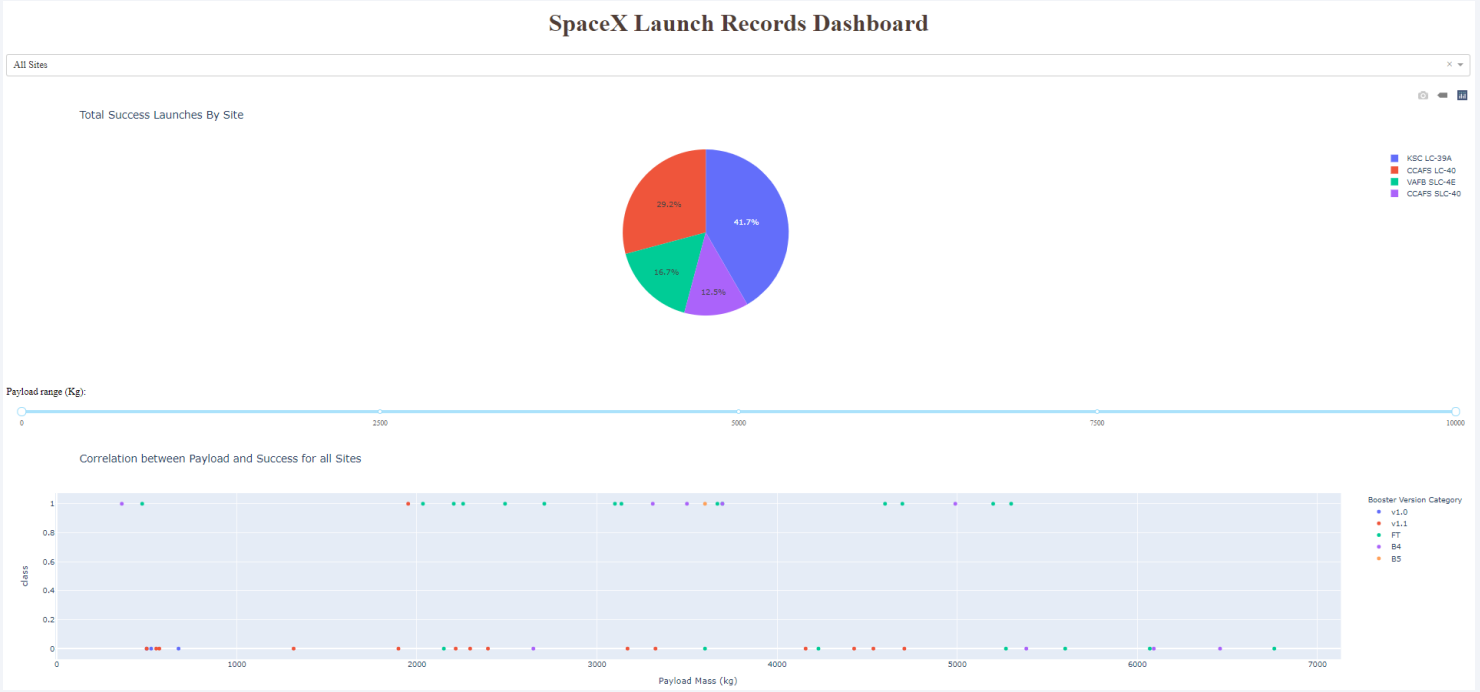
- We seamlessly imported the SpaceX dataset into a PostgreSQL database directly within the Jupyter Notebook environment. Utilizing SQL for exploratory data analysis, we extracted valuable insights from the dataset. For instance, we crafted queries to determine:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS).
  - The average payload mass carried by booster version F9 v1.1.
  - The total number of successful and failed mission outcomes.
  - The failed landing outcomes on the drone ship, along with their corresponding booster versions and launch site names.
- GitHub URL of completed EDA with SQL notebook

Build an Interactive Map with Folium

- We annotated all launch sites on the map and incorporated map objects like markers, circles, and lines to indicate the success or failure of launches at each site using Folium. We categorized launch outcomes (failure or success) as class 0 and 1, with 0 denoting failure and 1 denoting success.
- By utilizing color-coded marker clusters, we discerned which launch sites exhibited relatively high success rates. Additionally, we computed the distances between each launch site and its surroundings, allowing us to address questions such as:
- Are launch sites situated near railways, highways, and coastlines?
- Do launch sites maintain a certain distance from populated cities?
- GitHub URL of completed interactive map with Folium map

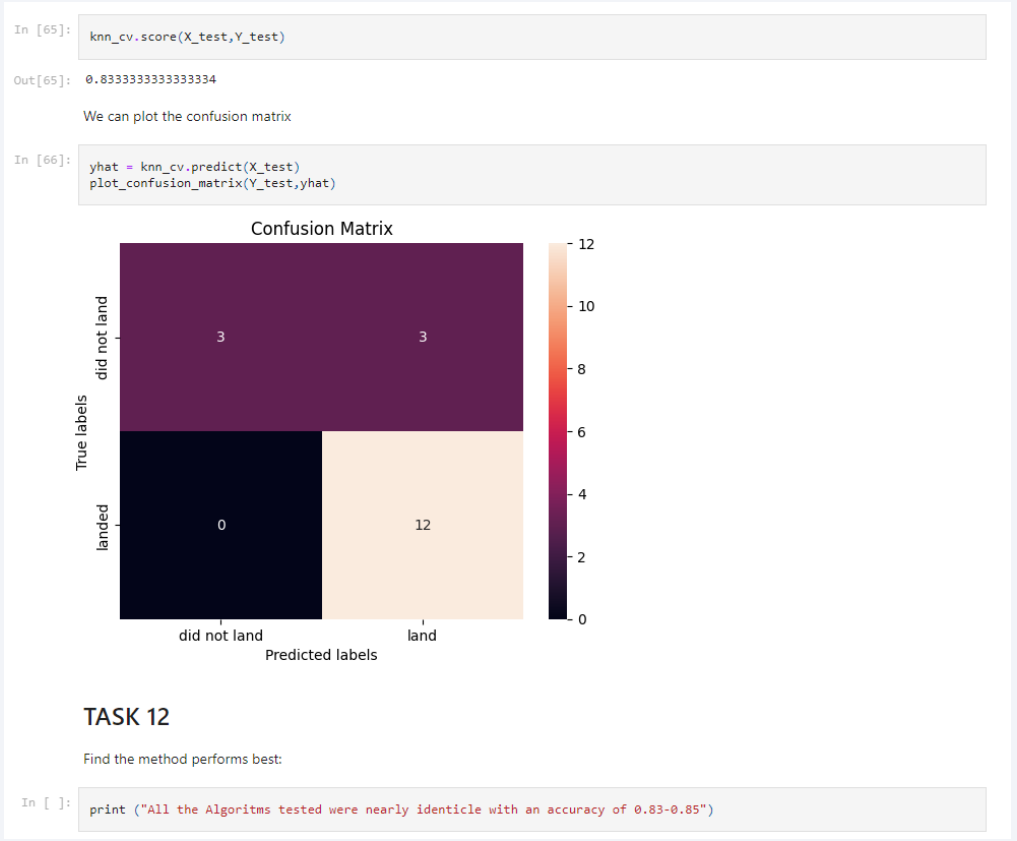
Build a Dashboard with Plotly Dash

- We constructed an interactive dashboard using Plotly Dash, where we included pie charts illustrating the total launches from specific sites. Additionally, we created scatter graphs to visualize the relationship between Outcome and Payload Mass (Kg) for various booster versions.
- GitHub URL of completed Plotly Dash lab



Predictive Analysis (Classification)

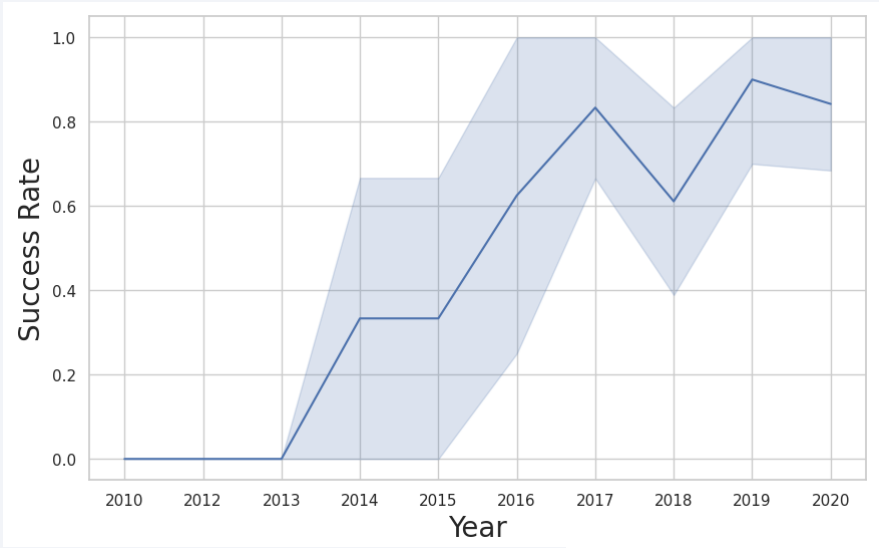
- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model
- GitHub URL of the completed predictive analysis lab

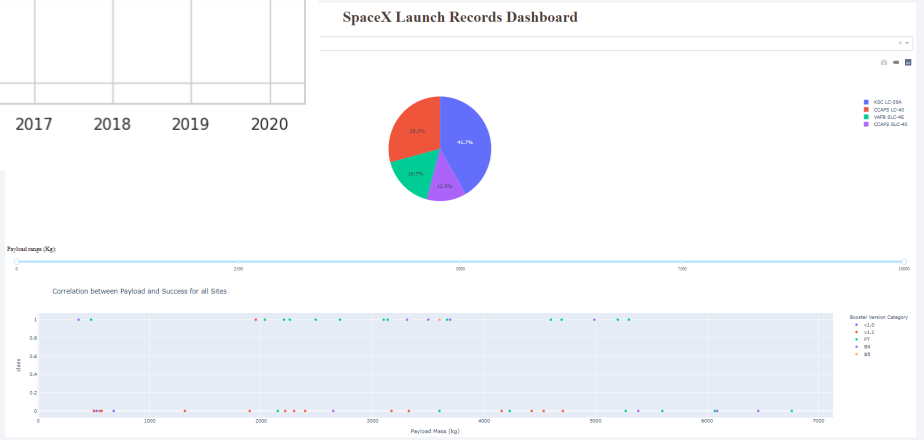


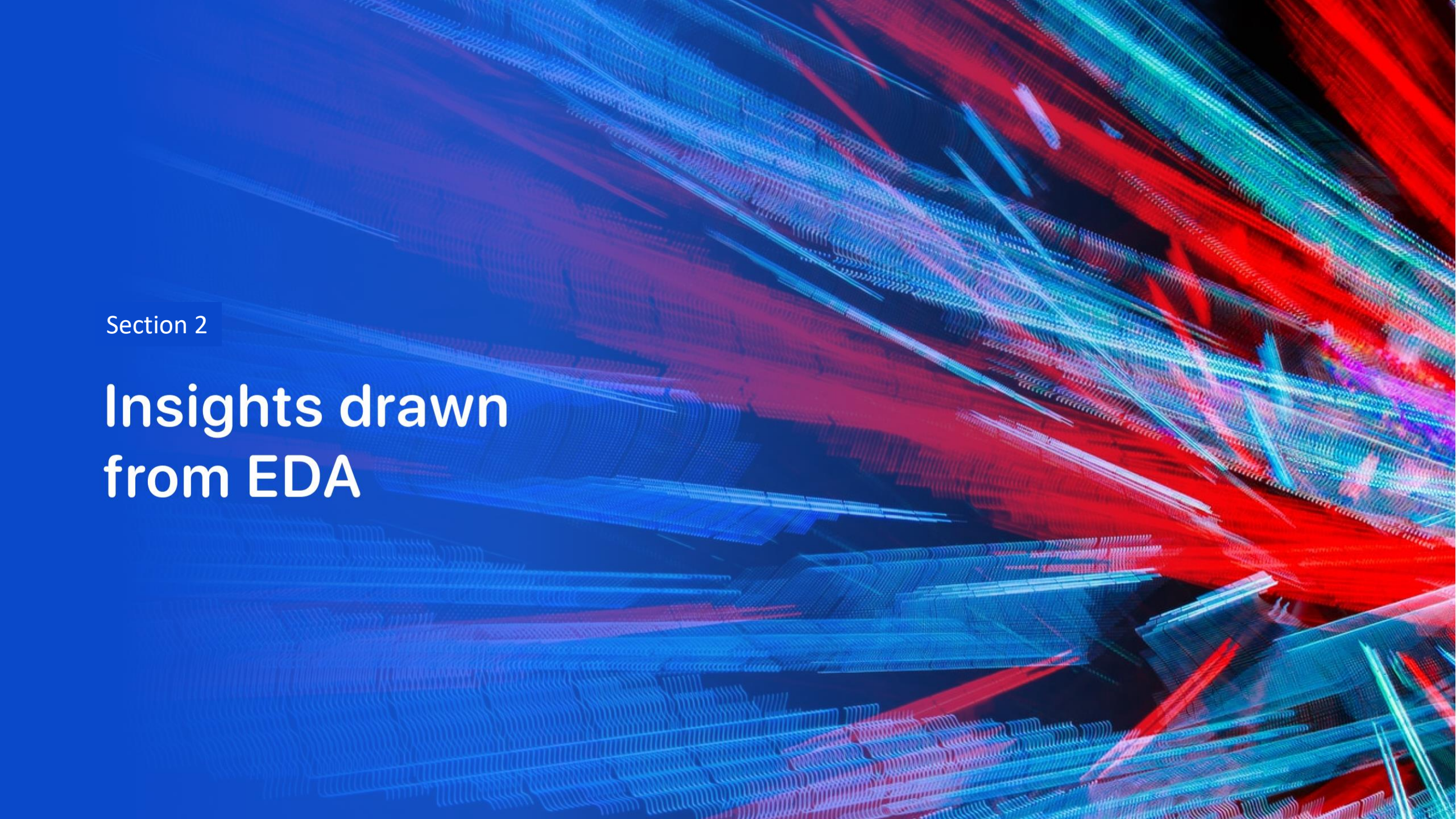
"All the Algoritms tested were nearly identicle with an accuracy of 0.83-0.85"

Results

- Exploratory data analysis results
  - heavy payloads the successful landing or positive landing rate are more for Polar
  - The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing
- Interactive analytics demo in screenshots
- Predictive analysis results
  - All the Algoritms tested were nearly identicle with an accuracy of 0.83-0.87

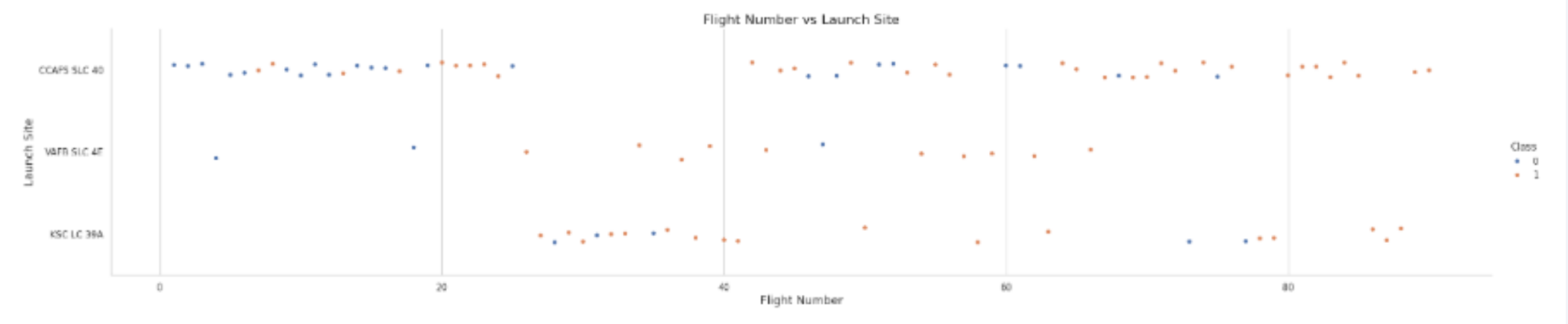





Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

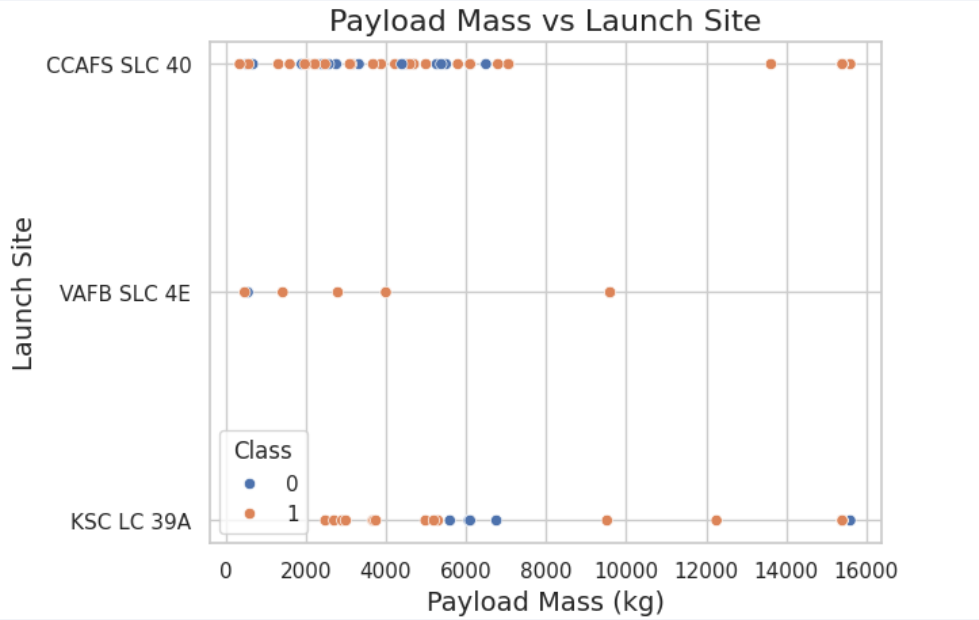
- Flight Number vs. Launch Site Scatter Plot



- More successes were apparent at each Launch site as flights continued

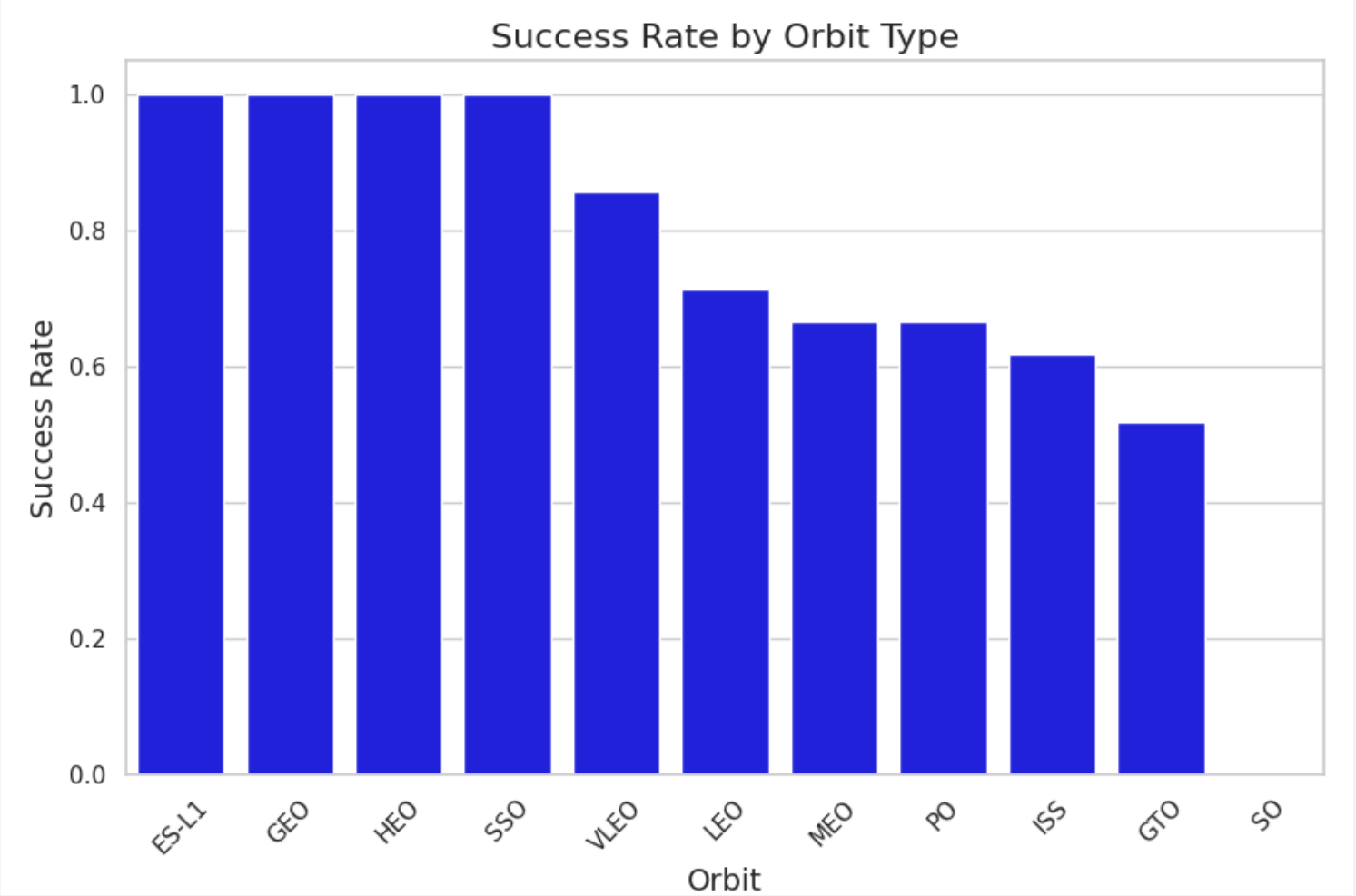
Payload vs. Launch Site

- Payload vs. Launch Site Scatter Plot
  - Lower Payloads are more frequent, but higher payloads seem to be more successful



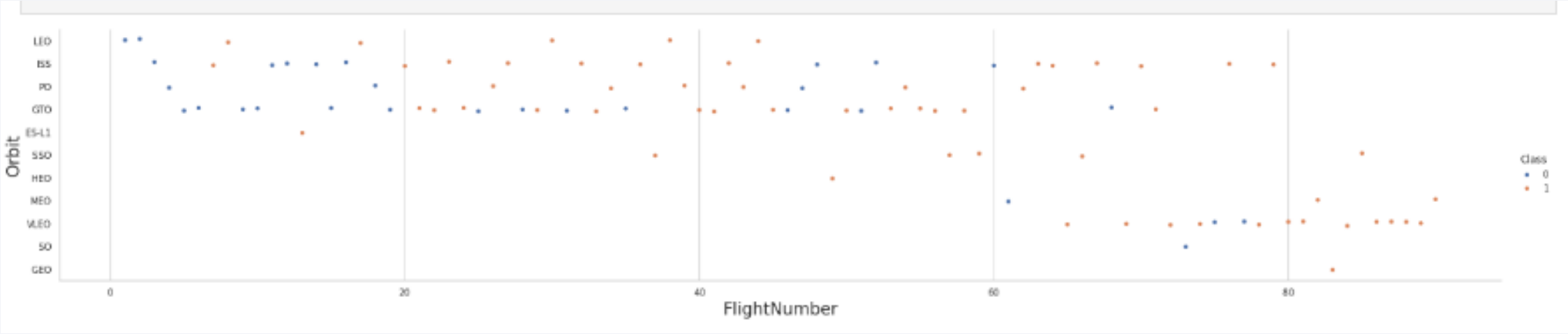
Success Rate vs. Orbit Type

- Success rate of each orbit type Bar Chart
  - Shown are the orbit type success rate, indicating potential risks by orbit type



Flight Number vs. Orbit Type

- Flight number vs. Orbit type



- Flights steadily grew more successful as flights continued

Payload vs. Orbit Type

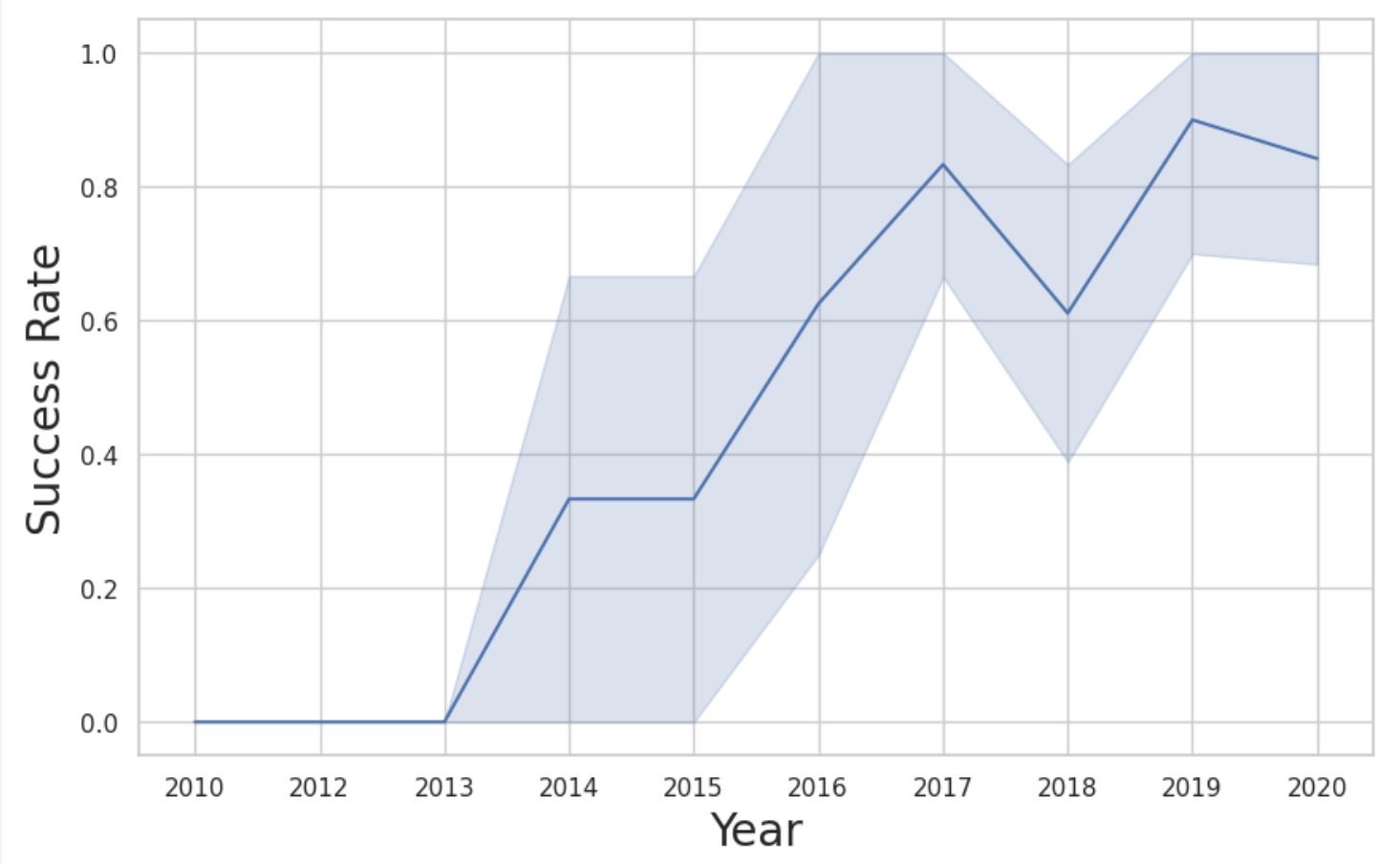
- Payload vs. orbit type Scatter Plot

- Respective orbits indicate some correlation to payload size

Launch Success Yearly Trend

- Yearly average success rate line chart

- The success rate since 2013 kept increasing till 2017 (stable in 2014) and after 2015 it started increasing



All Launch Site Names

- Names of the unique launch sites
- We utilized the keyword DISTINCT to display only unique launch sites from the SpaceX data

In [5]:

```
# Select relevant sub-columns: `Launch Site`, `Lat(Latitude)`, `Long(Longitude)`, `class`
spacex_df = spacex_df[['Launch Site', 'Lat', 'Long', 'class']]
launch_sites_df = spacex_df.groupby(['Launch Site'], as_index=False).first()
launch_sites_df = launch_sites_df[['Launch Site', 'Lat', 'Long']]
launch_sites_df
```

Out[5]:

	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

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

In [17]:
```
%sql Select * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' Limit 5
```

Running query in 'sqlite:///my_data1.db'

Out[17]:

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

Total Payload Mass

- The total payload carried by boosters from NASA was 45,596 Kg

Task 3

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

In [30]:
```
#%sql UPDATE SPACEXTBL SET your_column = TRIM(  PAYLOAD_MASS__KG_)
```

In [44]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

Running query in 'sqlite:///my_data1.db'

Out[44]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1 was ~2,535 Kg

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

In [49]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%'
```

Running query in 'sqlite:///my_data1.db'

Out[49]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- The first successful landing outcome on ground pad was December 12, 2022

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

Hint:Use min function

In [73]:
```
%sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
```

Running query in 'sqlite:///my_data1.db'

Out[73]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 are listed below

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

In [78]:
```
%sql select Booster_Version from SPACEXTBL where "Landing_Outcome" = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4
```

Running query in 'sqlite:///my_data1.db'

Out[78]:

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

Total Number of Successful and Failure Mission Outcomes

- Displayed below, the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

In [89]:
```
%sql SELECT Mission_Outcome , count() FROM SPACEXTBL group by Mission_Outcome
```

Running query in 'sqlite:///my_data1.db'

Out[89]:

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass are listed below

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

In [91]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

Running query in 'sqlite:///my_data1.db'

Out[91]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

- Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 listed below

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.

Note: SQLLite 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 [105…

```
%sql SELECT substr(Date,6,2) AS Month , "Landing_Outcome", Booster_Version, Launch_Site from SPACEXTBL where "Landing_Outcor
```

Running query in 'sqlite:///my_data1.db'

Out[105…

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

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

- The ranked 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 listed below

In [112…

```
%%sql
SELECT ROW_NUMBER() OVER (ORDER BY COUNT(*) DESC) AS Ranking, Landing_Outcome, COUNT(*) AS Landing_count
FROM SPACEXTBL
WHERE substr(Date,0,11) BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY COUNT(*) DESC
```

Running query in 'sqlite:///my_data1.db'

Out[112…

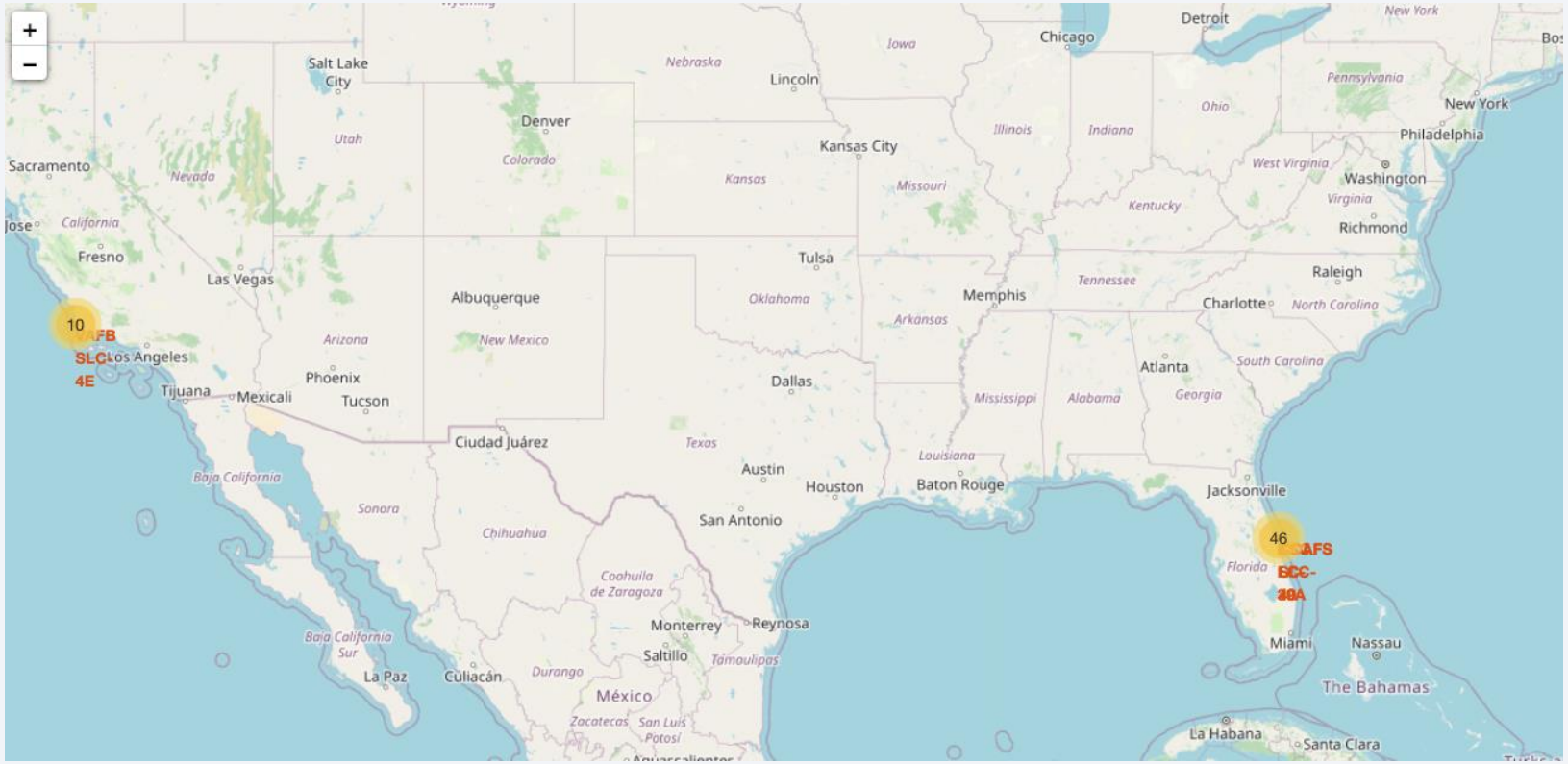
Ranking	Landing_Outcome	Landing_count
1	No attempt	10
2	Success (drone ship)	5
3	Failure (drone ship)	5
4	Success (ground pad)	3
5	Controlled (ocean)	3
6	Uncontrolled (ocean)	2
7	Failure (parachute)	2
8	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

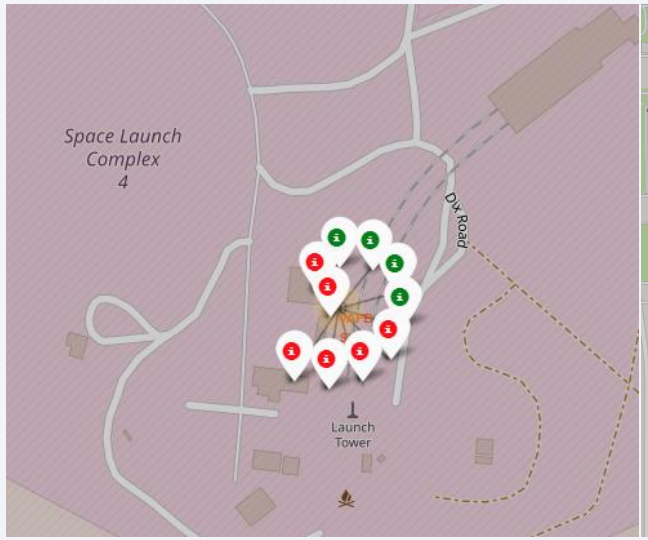
SpaceX launch site global map markers

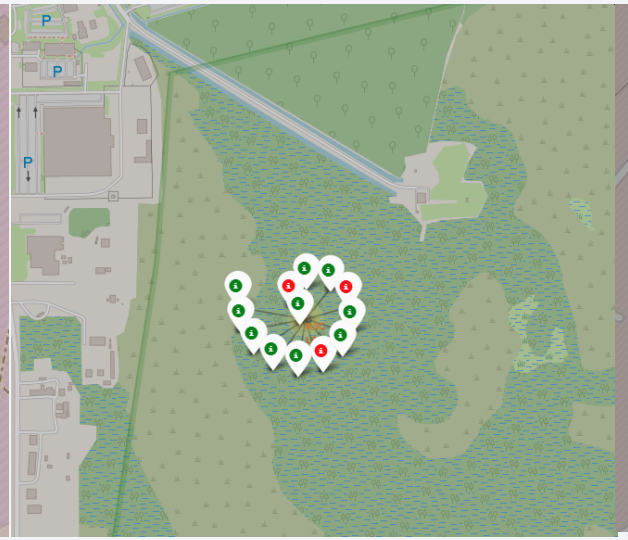
- SpaceX launch sites are only in the United States coastlines

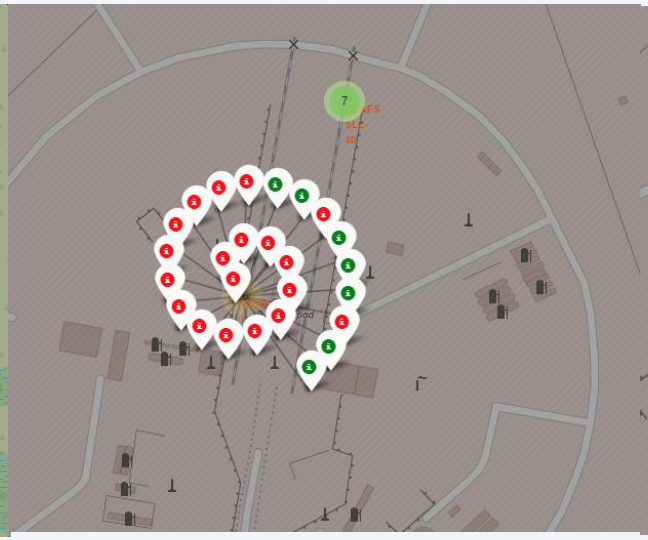


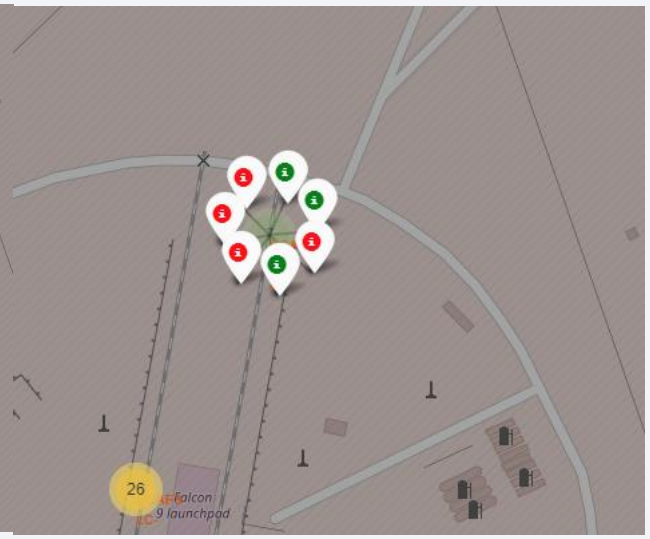
Launch Sites

- All Four Launch Sites have similar configurations
  - Later launch sites (end of tail) seem to have more successful landings



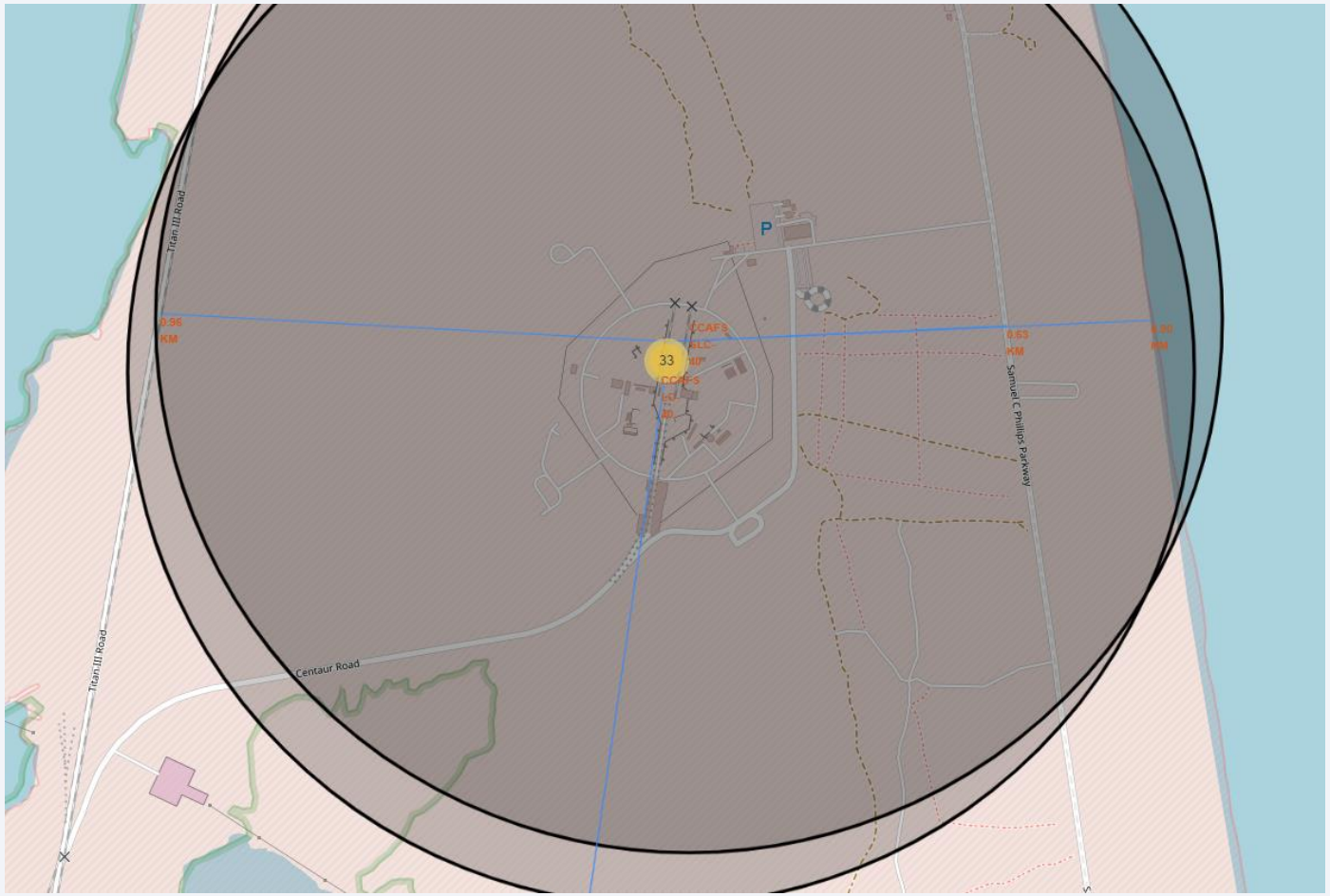


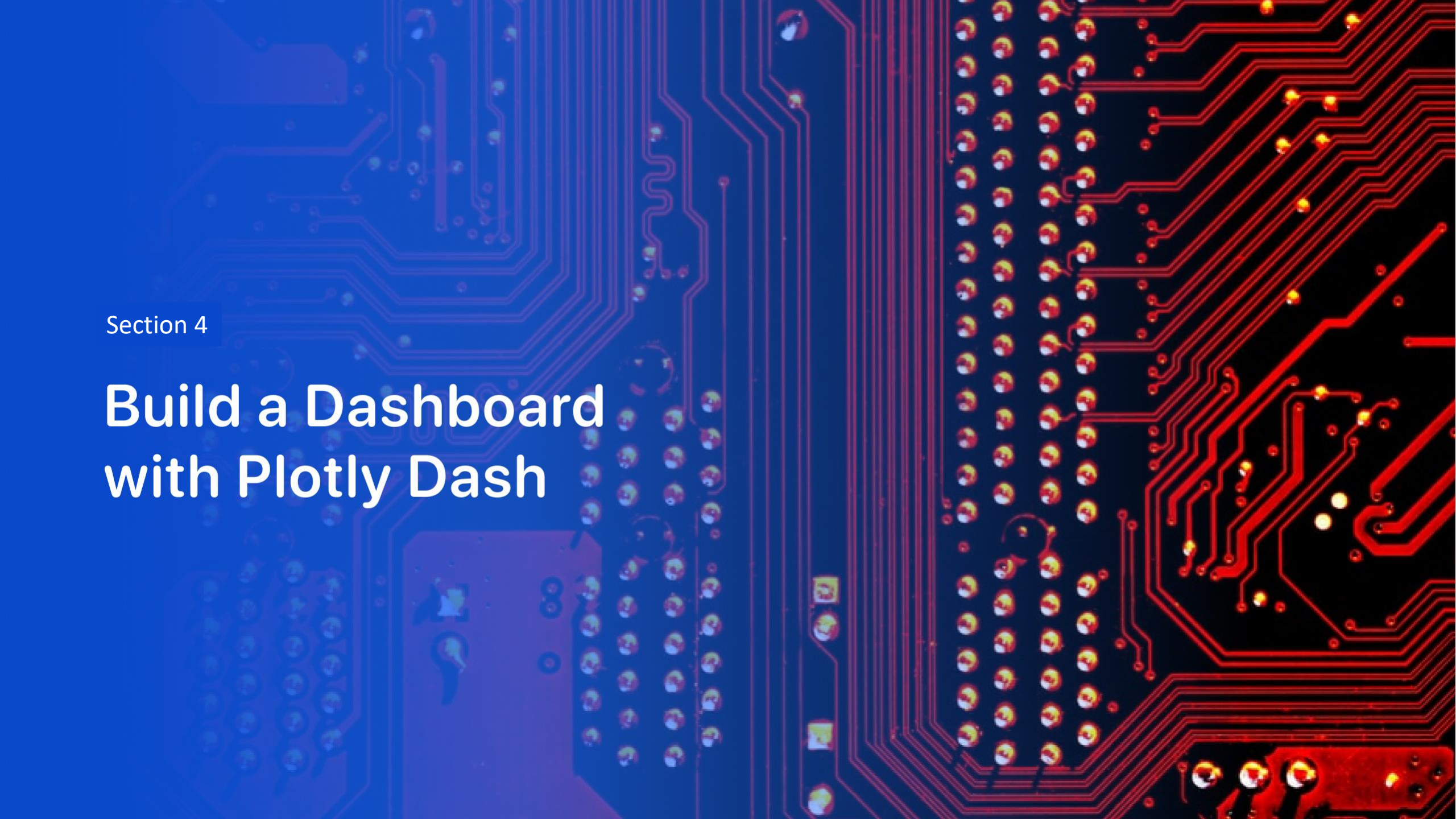




Safety Zoning

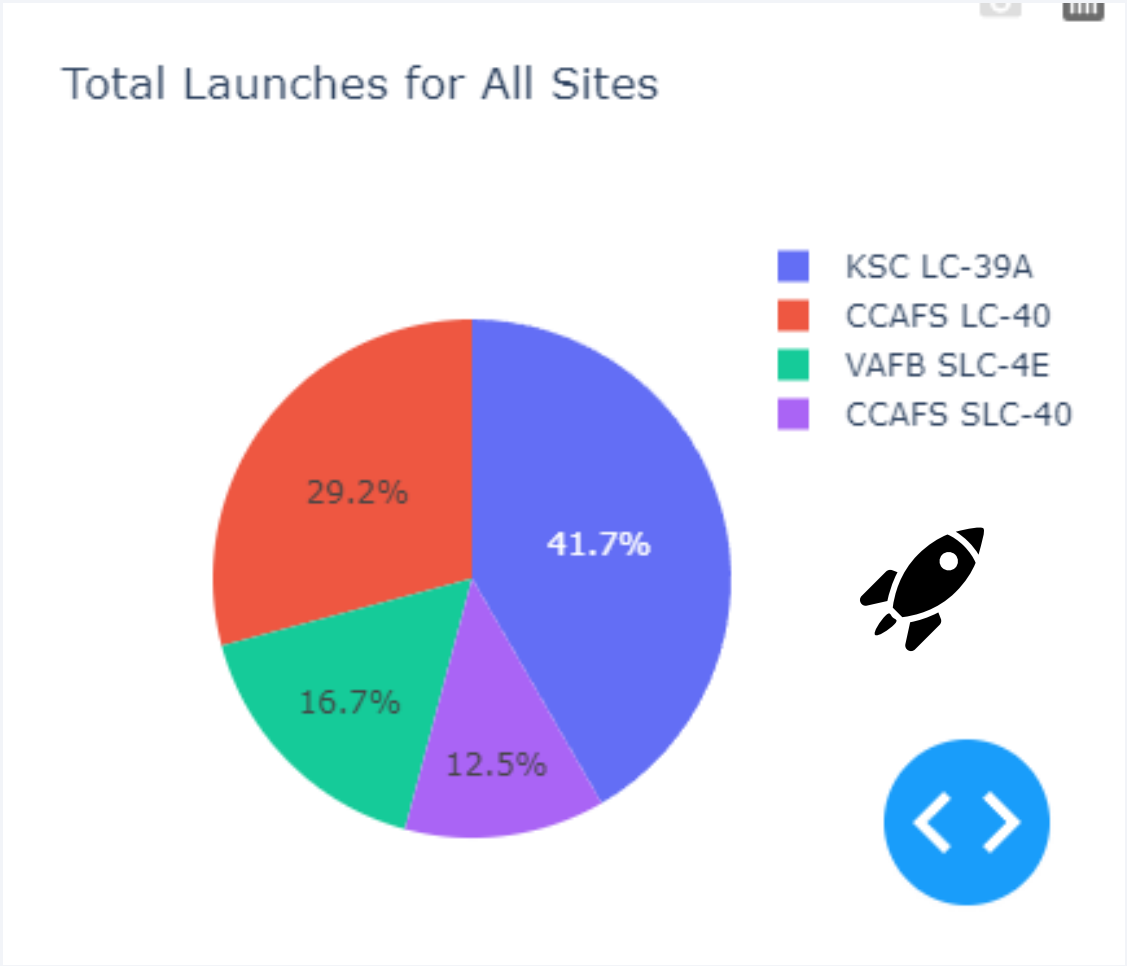
- Nearest Highway, Railroad, Coastline, and City atleast 0.63 km away from launch site




Section 4
Build a Dashboard with Plotly Dash

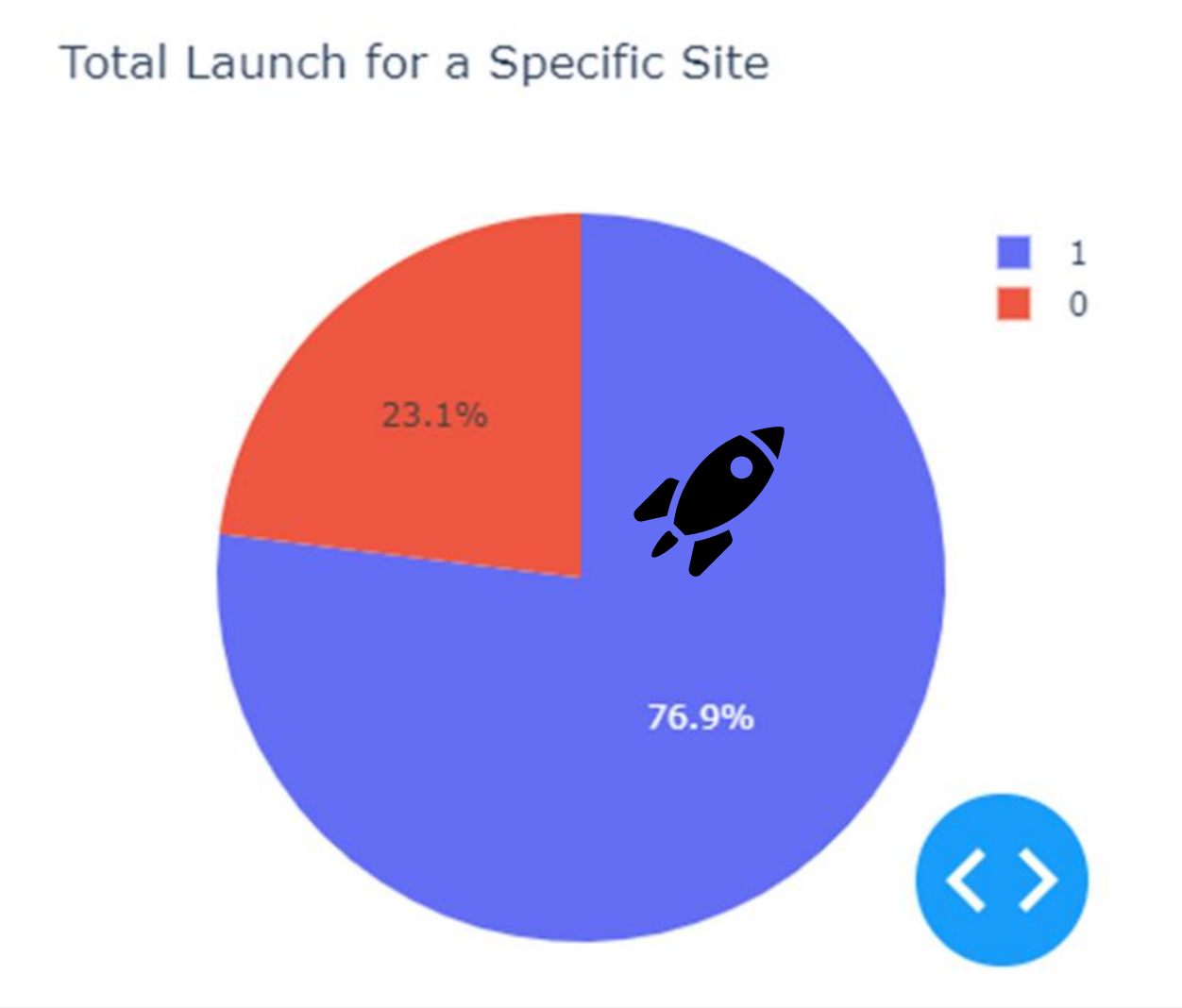
Launch Site Percentage of Success

- KSC LC-39A had the greatest successful launches



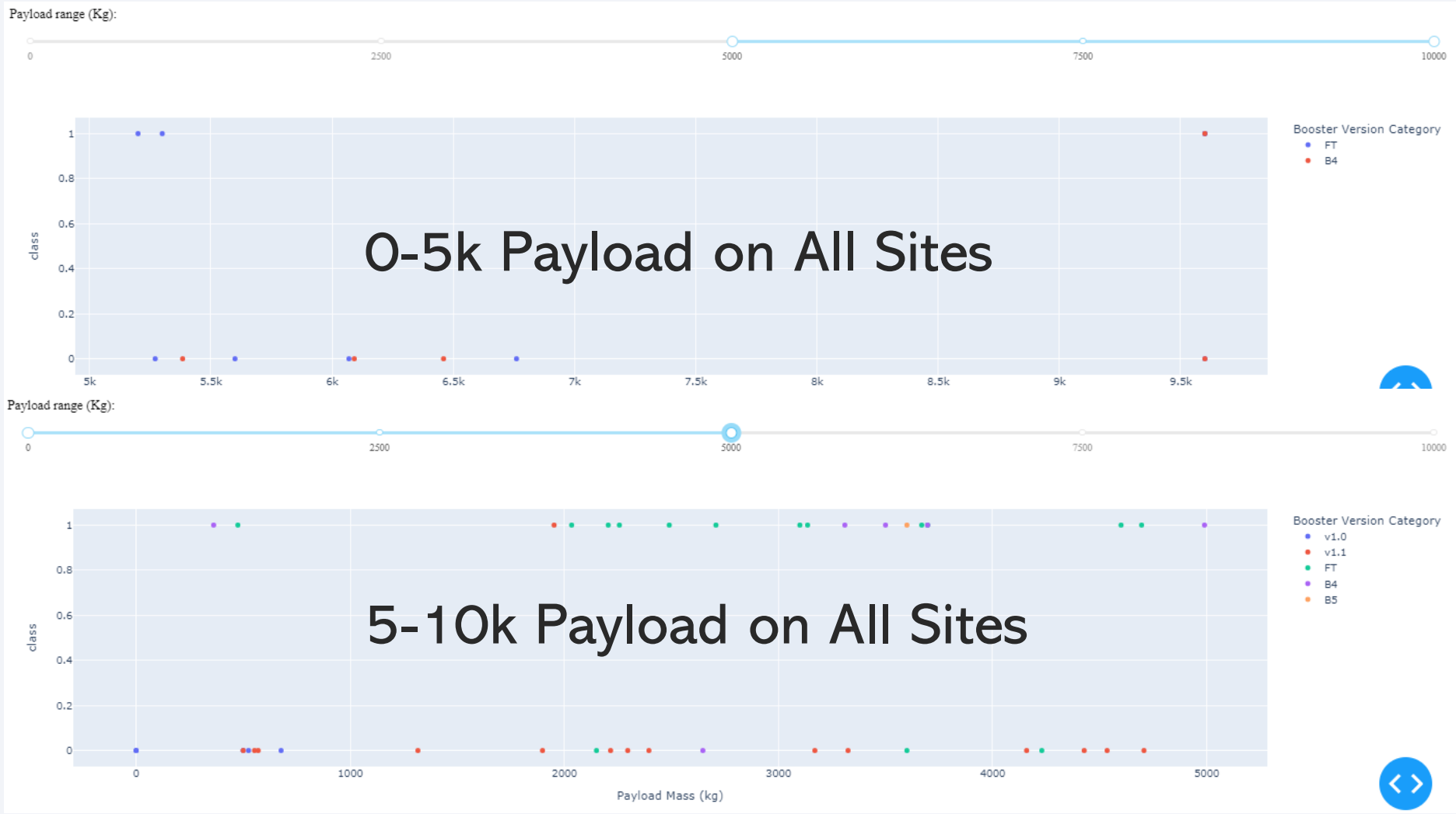
Highest Success Launch Ratio by Site

- KSC LC-39A again comes out on top with a ~77:23% success/failure rate



Payload vs Launch Outcome for varied ranges

- Replace <Dashboard screenshot 3> title with an appropriate title



- Success Rates are higher, across all sites, with lower weighted payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The model with the highest classification accuracy is the decision tree classifier

- The decision tree classifier model had an accuracy score of 0.8732

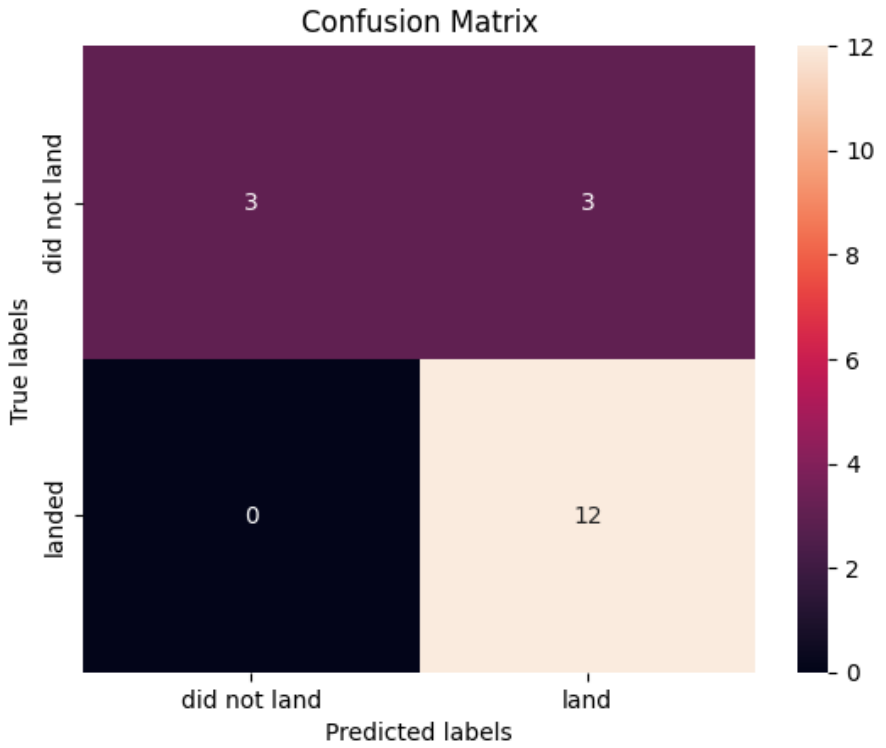
```
models = {'KNeighbors':knn_cv.best_score_,
              'DecisionTree':tree_cv.best_score_,
              'LogisticRegression':logreg_cv.best_score_,
              'SupportVector': svm_cv.best_score_}

bestalgorithm = max(models, key=models.get)
print('Best model is', bestalgorithm,'with a score of', models[bestalgorithm])
if bestalgorithm == 'DecisionTree':
    print('Best params is :', tree_cv.best_params_)
if bestalgorithm == 'KNeighbors':
    print('Best params is :', knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best params is :', logreg_cv.best_params_)
if bestalgorithm == 'SupportVector':
    print('Best params is :', svm_cv.best_params_)
```

```
Best model is DecisionTree with a score of 0.8732142857142856
Best params is : {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'mi
```

Confusion Matrix

- The confusion matrix for the decision tree classifier indicates that the classifier can effectively differentiate between the various classes. However, a notable issue arises with false positives, where unsuccessful landings are incorrectly classified as successful landings by the classifier.

```
[ ]  yhat = knn_cv.predict(X_test)
     plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- Based on our analysis, we can draw the following conclusions:

- There is a positive correlation between the number of flights at a launch site and the success rate at that site, suggesting that higher flight volumes contribute to increased success rates.
- The launch success rate has shown a consistent upward trend from 2013 to 2020, indicating an overall improvement in mission success over this period.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO exhibit the highest success rates, suggesting that missions targeting these orbits tend to be more successful.
- KSC LC-39A stands out as the launch site with the highest number of successful launches, indicating its effectiveness and reliability in hosting successful missions.
- Based on our evaluation, the Decision Tree classifier emerges as the best machine learning algorithm for this task, given its ability to effectively distinguish between different classes and provide accurate predictions.

Thank you!