IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Oleksandr
Zapadniuk
18/10/2023

Outline

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

Executive Summary

- Summary of methodologies
  - Collect data using SpaceX REST API and Web Scraping
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Build Models to predict landing outcomes
- Summary of all results
  - Exploratory Data Analysis result
  - Visualization/Analytics
  - Predictive Analytics result

Introduction

- Project background and context

  SpaceX is a leader in the space industry. His achievements include sending a spacecraft to the ISS, launching a satellite constellation that provides Internet access, and sending manned missions into space. SpaceX is advertising the launch of its $62 million Falcon 9 rocket on its website. Other suppliers are valued at more than $165 million. This difference is possible due to the reuse of the first stage of the Falcon 9 rocket. So if we can determine whether the first stage will land, we can determine the launch cost. This information could be used by a SpaceX competitor. The goal of the project is to create a machine learning pipeline that allows predicting the success of the first stage.

- Problems you want to find answers

  - How payload mass, launch site, number of flights and orbits affect first-stage landing success?
  - Rate of successful landings over time..
  - Best predictive model for successful landing.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Collect Data using SpaceX REST API and web scraping from Wikipedia
- Perform data wrangling
  - Wrangle Data by applying one hot encoding to prepare the data for analysis and modeling
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build Models to predict landing outcomes using classification models. Tune and evaluate models to find best model and parameters.

Data Collection

- From API
  - Request data from SpaceX API
  - Decode response using .json()
  - Convert to a dataframe using .json_normalize()
- From Wikipedia
  - Request data from Wikipedia
  - Create BeautifulSoup object from HTML response
  - Collect data from parsing HTML tables
  - Create dataframe from data

Exports data to csv files

Data Collection – SpaceX API

- Request data from SpaceX API
- Decode response and convert to dataframe
- Filter dataframe
- Replace missing values
- Export data to csv file

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

[7]: response = requests.get(spacex_url)
```

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

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

Calculate below the mean for the `PayloadMass` using the `.mean()`. Then use the mean and the `.replace()` function to replace

```
[28]: mean_value = data_falcon9['PayloadMass'].mean()

data_falcon9.loc[:, 'PayloadMass'].replace(np.NaN, mean_value, inplace=True)
```

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/1.%20Spacex-data-collection-api.ipynb

Data Collection - Scraping

- Request data from Wikipedia
- Create BeautifulSoup object
- Extract column names from HTML table header
- Collect data from HTML tables
- Create dataframe
- Export data to csv file

```
[3]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

Next, request the HTML page from the above URL and get a `response` object

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.

```
[4]: r = requests.get(static_url)
```

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

```
[5]: page = r.text
soup = BeautifulSoup(page)
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
```

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

```
[6]: soup.title
```

```
[6]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

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

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/2.%20Webscraping.ipynb

Data Wrangling

- Calculate the number of launches on each site use value_counts()
- Calculate the number and occurrence of each orbit use value_counts()
- Calculate the number and occurence of mission outcome per orbit type use value_counts()
- Create a landing outcome label from Outcome column
- Export data to csv file

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/3.%20Data_Wrangling.ipynb

EDA with Data Visualization

- Charts
  - Flight number and payload
  - Flight number and launch location
  - Payload mass (kg) depending on the launch site
  - Payload mass (kg) depending on the type of orbit
- Analysis
  - View relationships using scatter plots
  - Show comparisons between individual categories using histograms

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/5.%20Data_Visualization.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch site begins with ‘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 succesful 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 max payload mass
- 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
- 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

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/4.%20Sql.ipynb

Build an Interactive Map with Folium

- Added red circles at all launch sites coordinates with a popup label showing its name using its name using its latitude and longitude coordinates Colored Markers of Launch Outcomes
- Added colored markers of successful (green) and unsuccessful (red) launches at each launch site to show which launch sites have high success rates
- Added colored lines to show distance between launch site CCAFS SLC-40 and its proximity to the nearest coastline, railway, highway, and city

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/6.%20Launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Dropdown List with Launch Sites

- Allow user to select all launch sites or a certain launch site

Pie Chart Showing Successful Launches

- Allow user to see successful and unsuccessful launches as a percent of the total

Slider of Payload Mass Range

- Allow user to select payload mass range

Scatter Chart Showing Payload Mass vs. Success Rate by Booster Version

- Allow user to see the correlation between Payload and Launch Success

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Create NumPy array from the Class column
- Standardize the data with StandardScaler. Fit and transform the data.
- Split the data using train_test_split
- Create a GridSearchCV object with cv=10 for parameter optimization
- Apply GridSearchCV on different algorithms: logistic regression, support vector machine, decision tree, K-Nearest Neighbor
- Calculate accuracy on the test data using .score() for all models
- Assess the confusion matrix for all models

https://github.com/nevermind311/DS-Capstone-SpaceX/blob/main/7.%20Machine_Learning.ipynb

Results

Exploratory Data Analysis

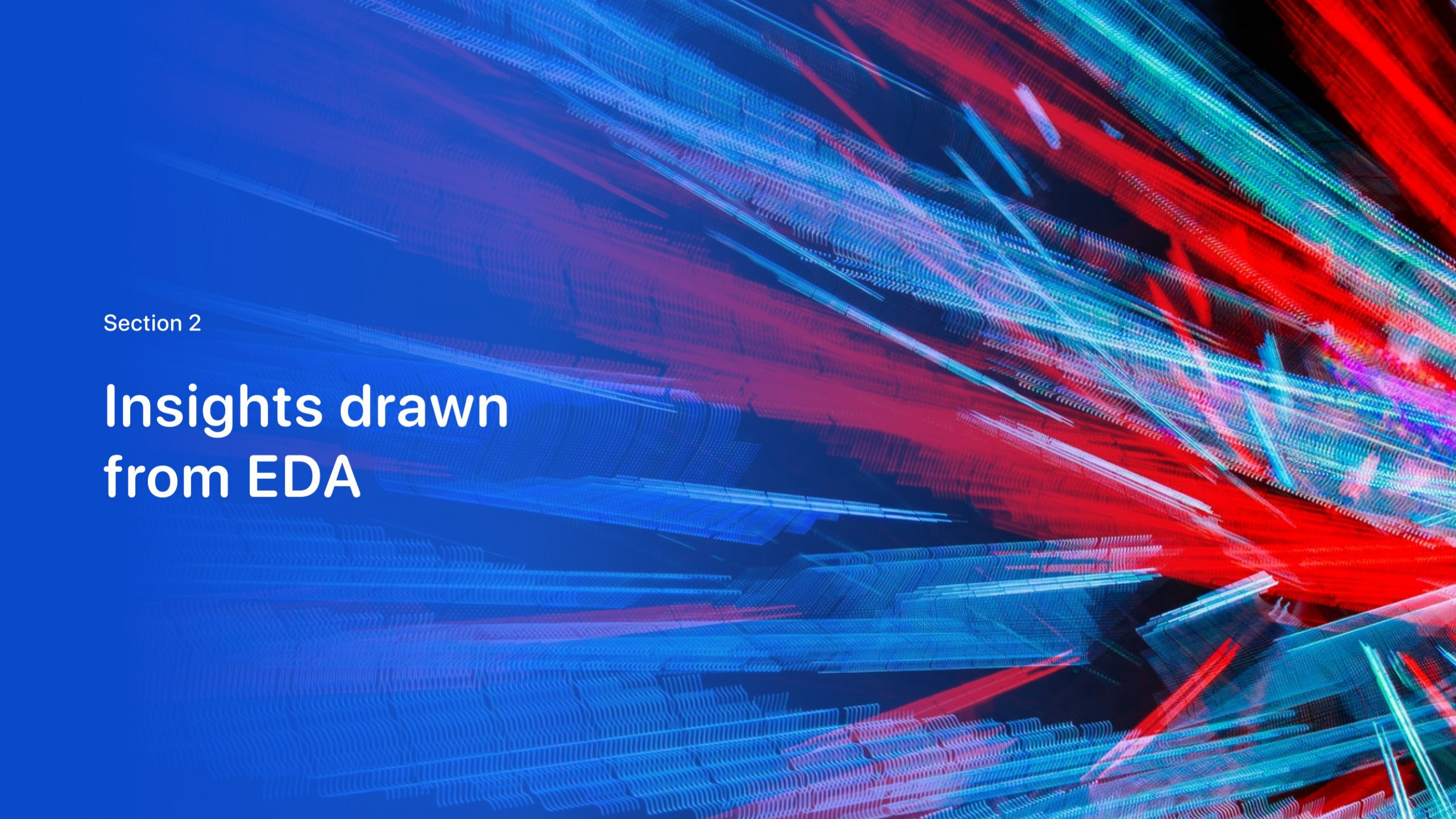
- Launch success has improved over time
- KSC LC-39A has the highest success rate among landing sites
- Orbits ES-L1, GEO, HEO and SSO have a 100% success rate

Visual Analytics

- Launch sites are far enough away from anything a failed launch can damage (city, highway, railway), while still close enough to bring people and material to support launch activities

Predictive Analytics

- Decision Tree model is the best predictive model for the dataset

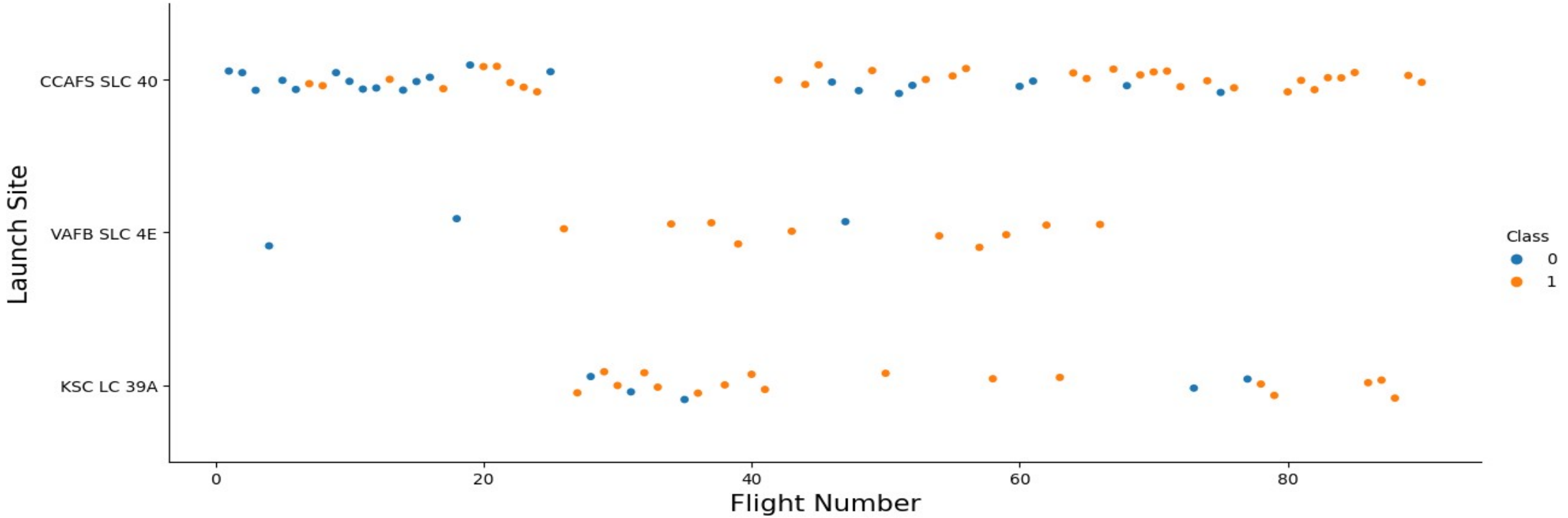


Section 2

Insights drawn from EDA

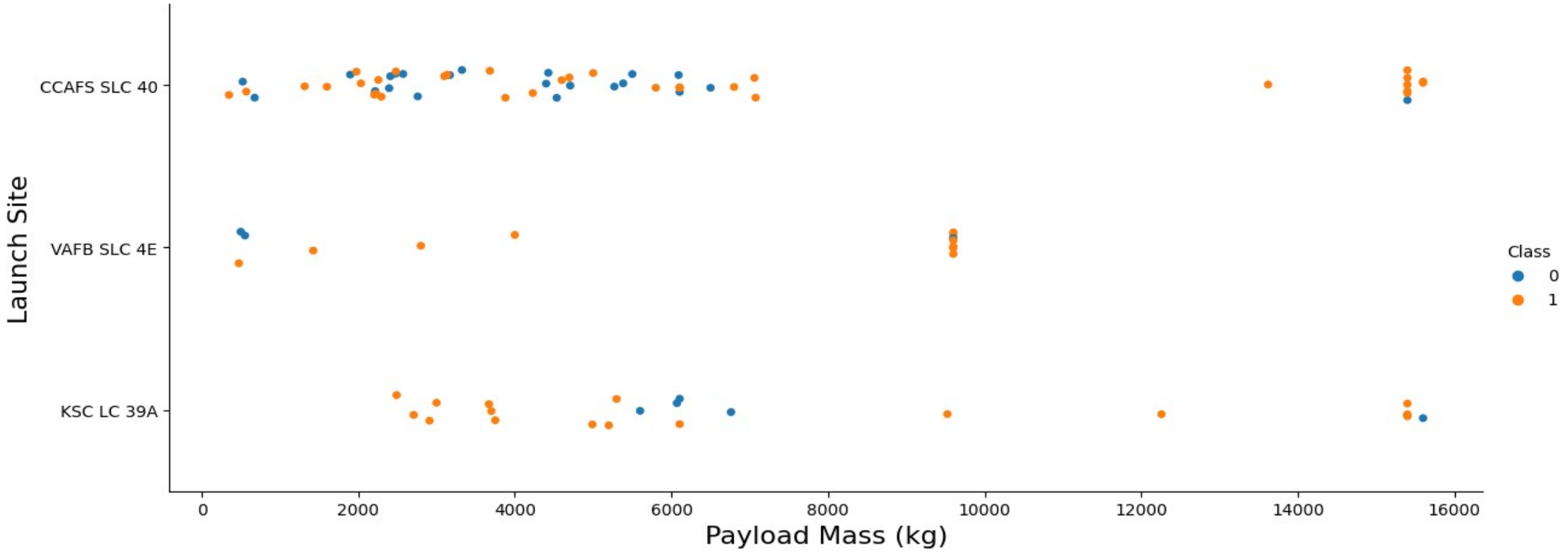
Flight Number vs. Launch Site

- Earlier flights have a lower success rate but later flights have a higher success rate
- VAFB SLC 4E and KSC LC 39A have higher success rates



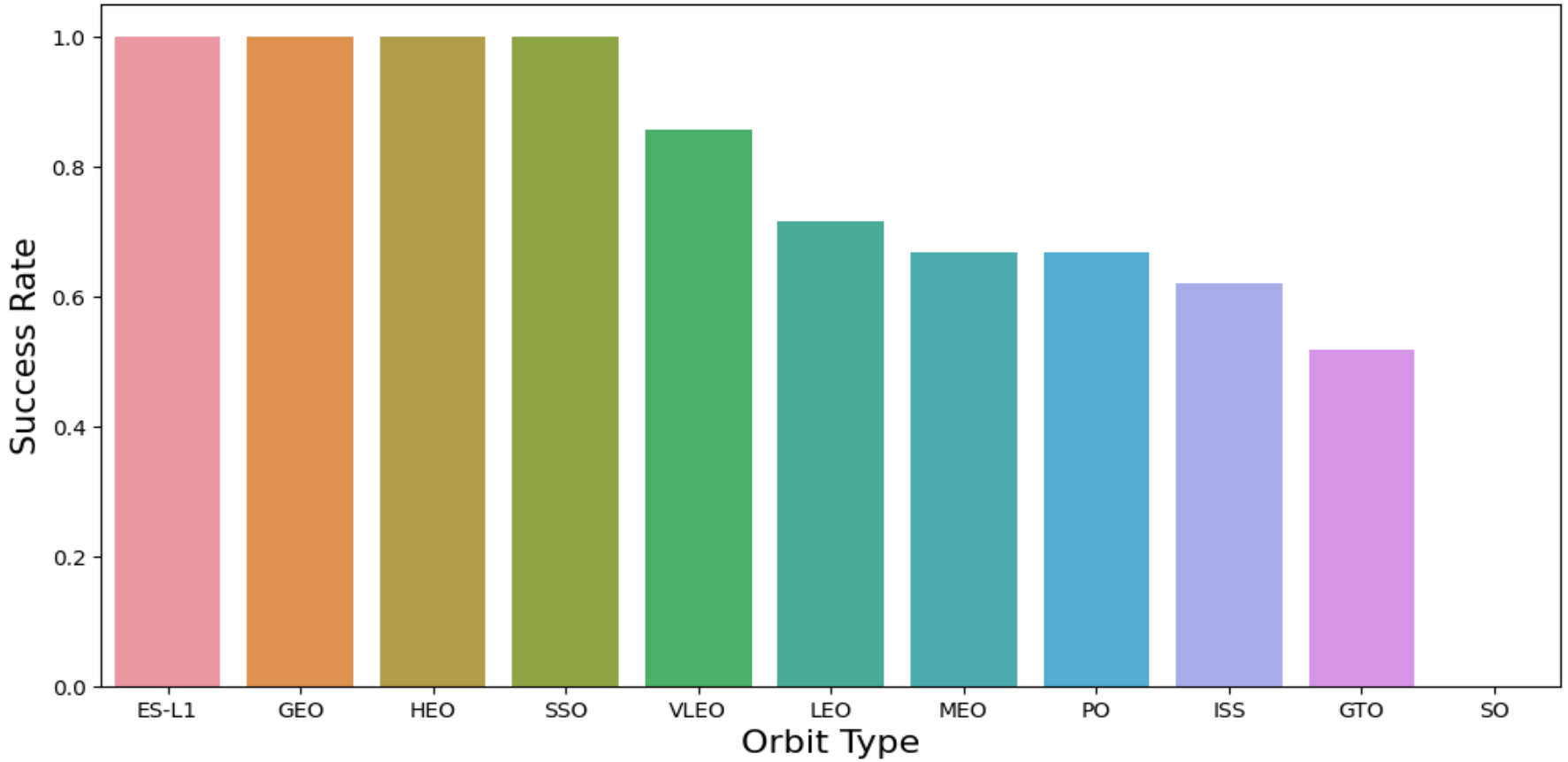
Payload vs. Launch Site

- Most launches with a payload mass of more than 7,000 kg were successful
- No rockets with a payload mass of more than 10,000 kg were launched from the VAFB SLC 4E launch pad



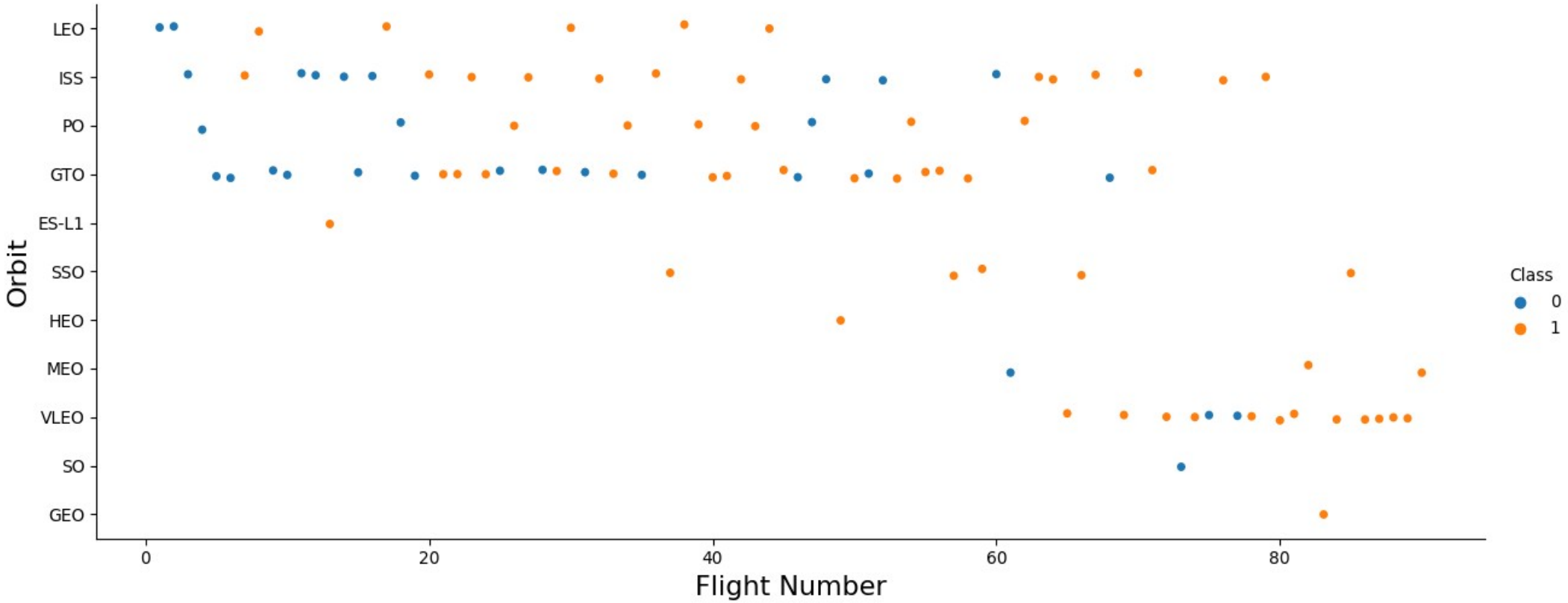
Success Rate vs. Orbit Type

- Orbit types ES-L1, GEO, HEO, SSO have a 100% Success rate
- Orbit type SO have a 0% Success rate



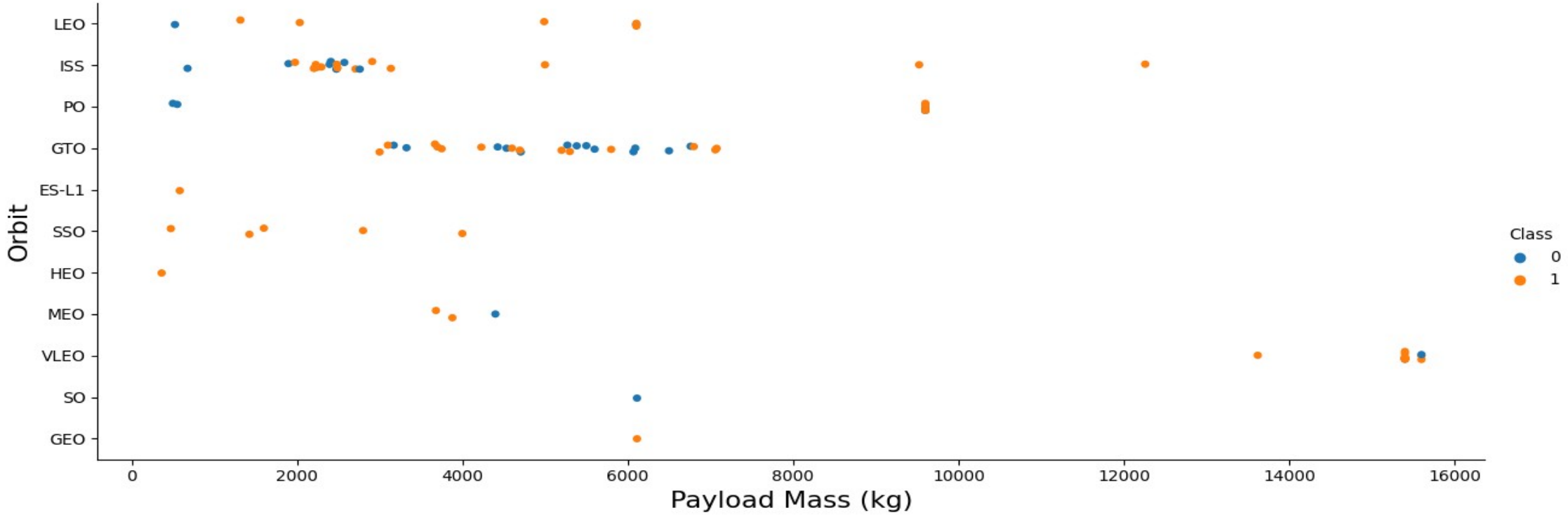
Flight Number vs. Orbit Type

- In most cases, the probably of success increases with the number of flights to each orbit
- In the case of the GTO orbit, this trend is not observed



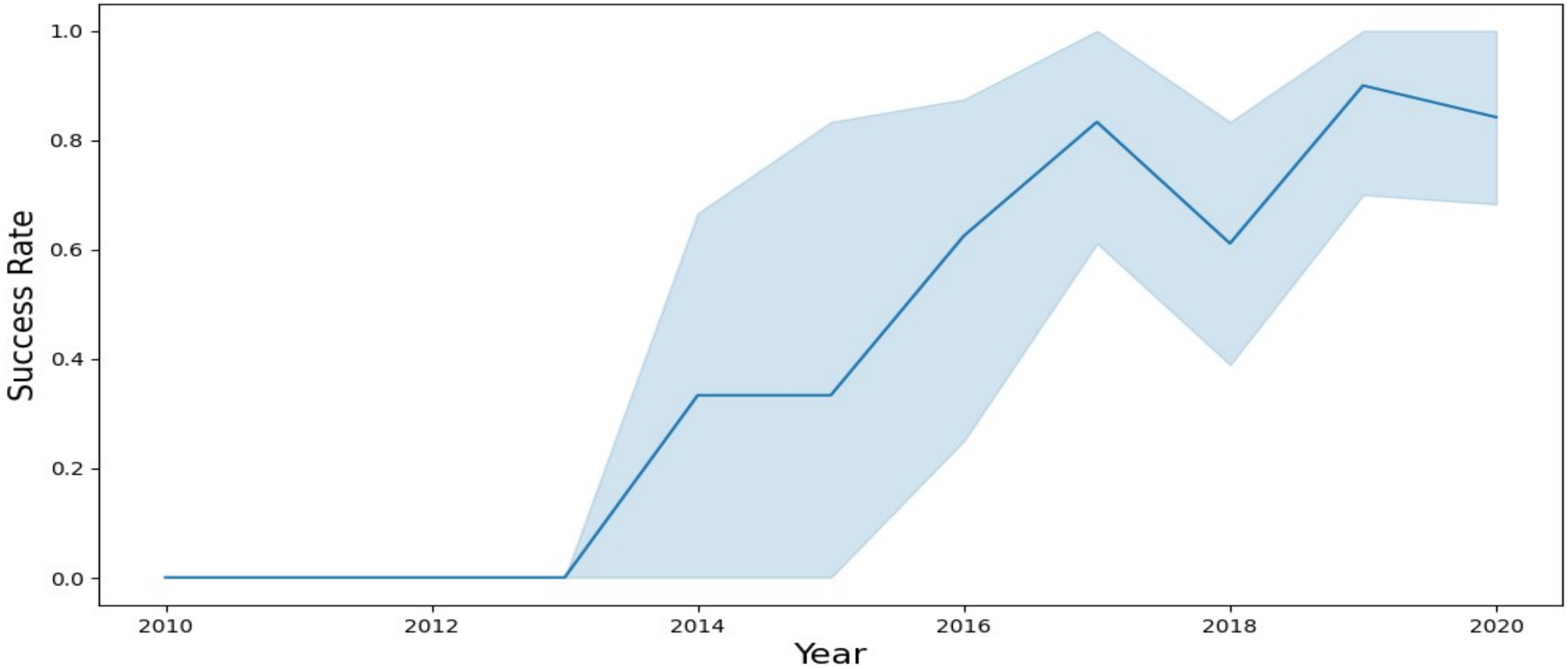
Payload vs. Orbit Type

- Launches from LEO, ISS and PO orbits with heavy payloads are successful
- Heavy payload GTO launches have had mixed success



Launch Success Yearly Trend

- The Success Rate since 2013 kept increasing till 2017 (stable 2014)



All Launch Site Names

Launch Site Names

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

Task 1

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

```
[6]: %sql select distinct Launch_Site from SPACEXTABLE;
 * sqlite:///my_data1.db
Done.
```

[6]:

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

Launch Site Names Begin with 'CCA'

To perform this task, use the following query:

%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;

Task 2

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

```
%sql select * from SPACEXTABLE where Launch_Site like 'CCA%' limit 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	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)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	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

Total Payload Mass

Total payload Mass = 45,596 kg carried by boosters launched by NASA (CRS)

Task 3

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

```
[8]: %sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer = 'NASA (CRS)';
```

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

[8]:

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Average Payload Mass = 2,928 kg carried by booster version F9 v1.1

Task 4

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

```
[10]: %sql select avg(PAYLOAD_MASS__KG_) from SPACEXTABLE where Booster_Version = 'F9 v1.1';
```

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

[10]:

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

First successful landing outcome in ground pad was 2015-12-22

Task 5

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

Hint:Use min function

```
[10]: %sql select min(Date) from SPACEXTABLE where Landing_Outcome = 'Success (ground pad)'
```

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

[10]:

min(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000:

- F9 FT B1022, F9 FT B1026, F9 FT B1021.2, F9 FT B1031.2

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

```
%sql select Booster_Version from SPACEXTABLE\
where Landing_Outcome = "Success (drone ship)" and PAYLOAD_MASS__KG_ between 4000 and 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- Success – 99
- Success (payload status unclear) – 1
- Failure (in flight) - 1

Task 7

List the total number of successful and failure mission outcomes

[12]:
```
%sql SELECT Mission_Outcome, COUNT(*) as count from SPACEXTABLE\
GROUP BY Mission_Outcome ORDER BY count DESC;
```

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

[12]:

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

Boosters Carried Maximum Payload

Carrying Max Payload

- 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

Task 8

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

```
4]: %sql select Booster_Version from SPACEXTABLE\
    where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
 * sqlite:///my_data1.db
Done.
```

4]:

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

Showing month, date, booster version, launch site and landing outcome

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

[15]:
```
%sql select substr(Date, 6, 2) as month, Landing_Outcome, Booster_Version, Launch_Site from SPACEXTABLE\
where substr(Date, 1, 4) = '2015' and Landing_Outcome = 'Failure (drone ship)'
```

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

[15]:

month	Landing_Outcome	Booster_Version	Launch_Site
10	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

Rank the count of landing outcomes between 2010-06-04 and 2017-03-20 in descending order

```
[15]: %sql select Landing_Outcome, count(*) as count_outcomes \
from SPACEXTABLE\
where Date between '2010-06-04' and '2017-03-20'\
group by Landing_Outcome\
order by count_outcomes Desc;
 * sqlite:///my_data1.db
Done.
```

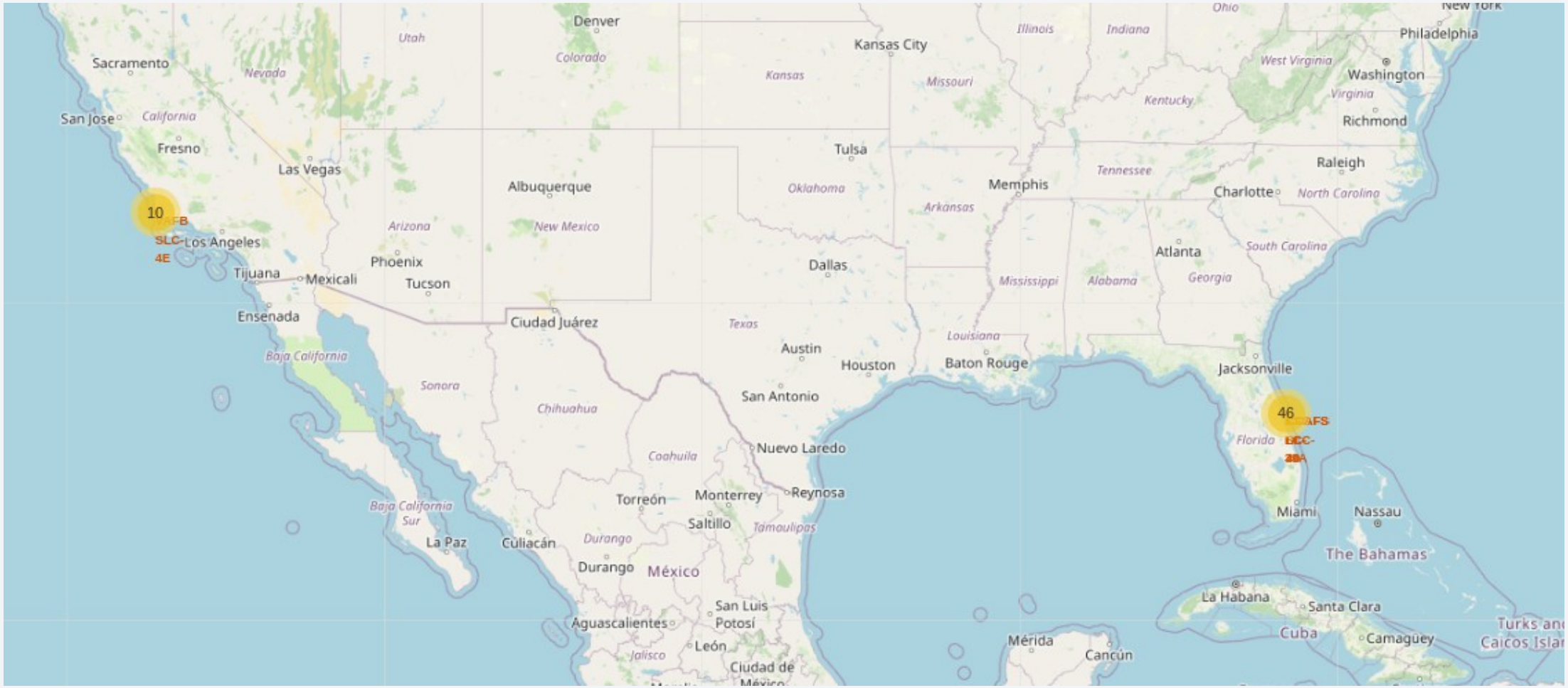
[15]:

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

Section 4

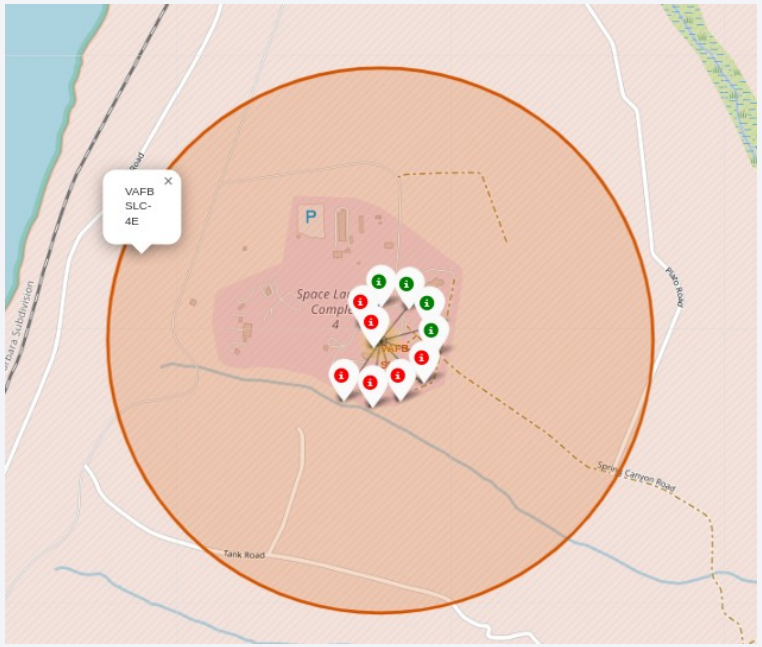
Launch Sites Proximities Analysis

All launch sites map markers



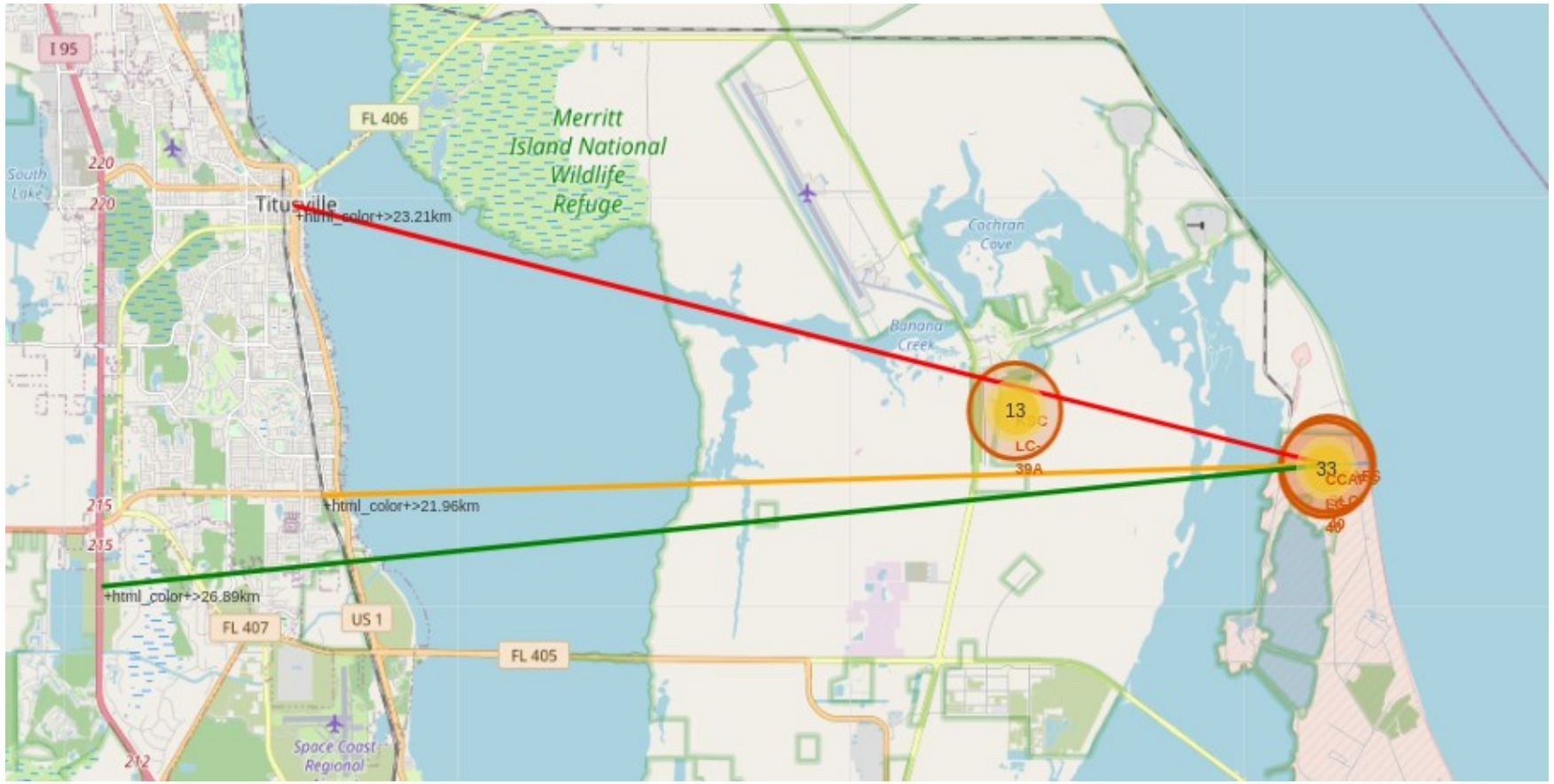
Markers showing launch sites with color labels

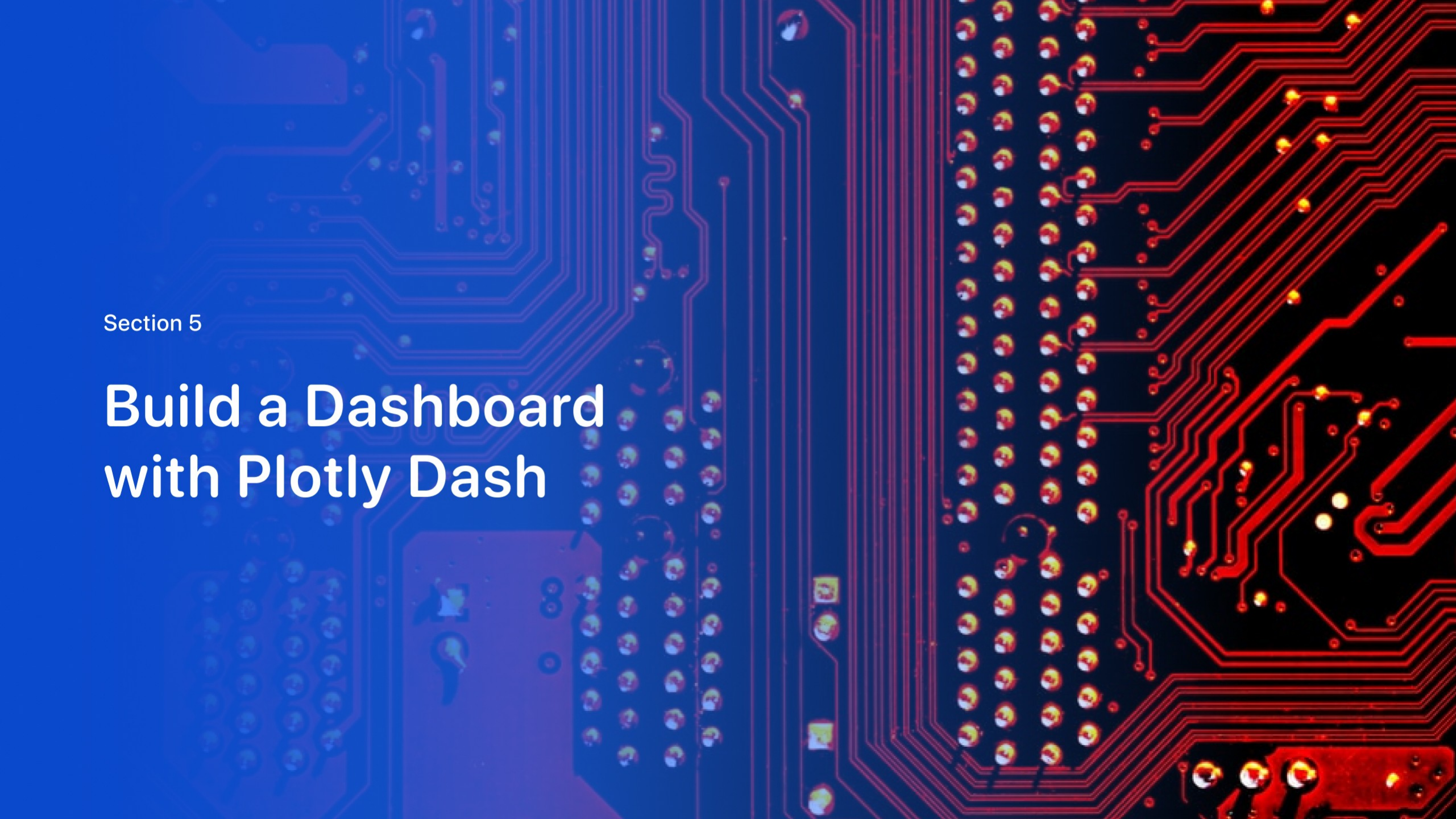
- Green markers for successful launches
- Red markers for unsuccessful launches



Launch Site distance to landmarks

- City Distance 23.21 km
- Railway Distance 21.96 km
- Highway Distance 26.89 km
- Coastline Distance 0.87 km



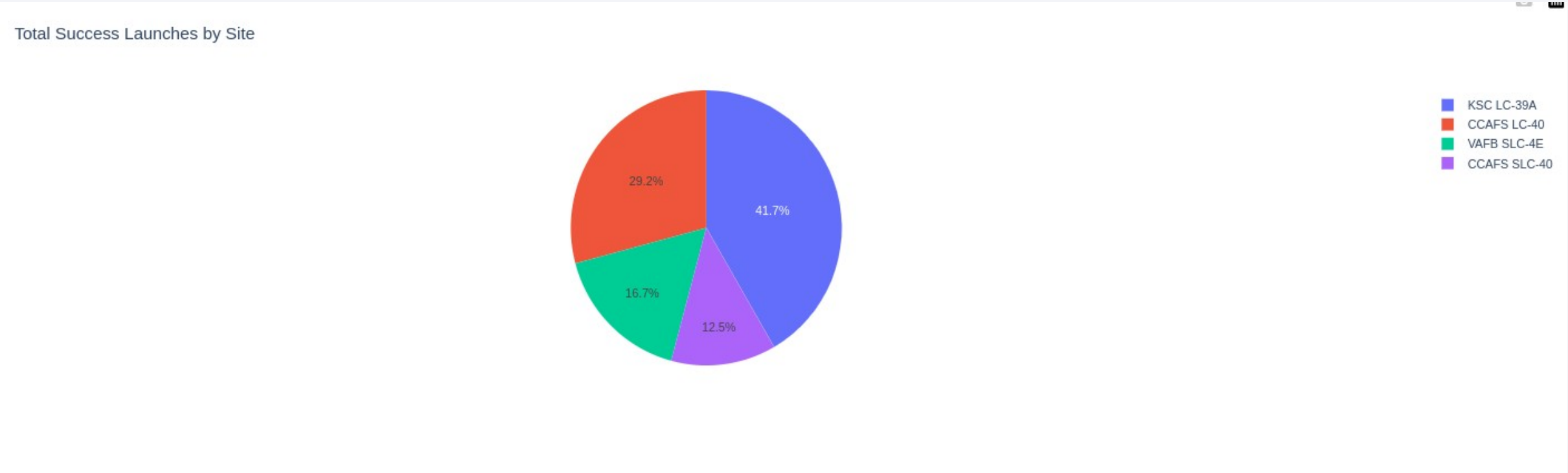


Section 5

Build a Dashboard with Plotly Dash

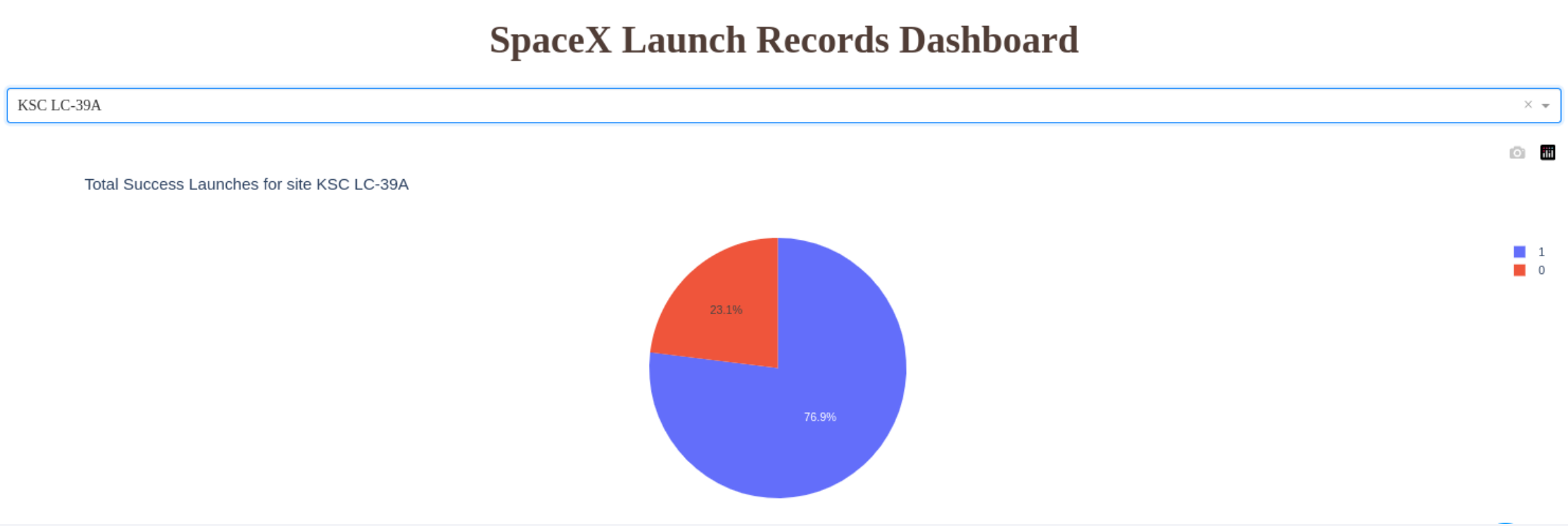
Launch Success by Site

The piechart displays the percentage of the total number of successful launches
KSC LC-39A has the highest rate - 41.7%



Launch Success (KSC LC-39A)

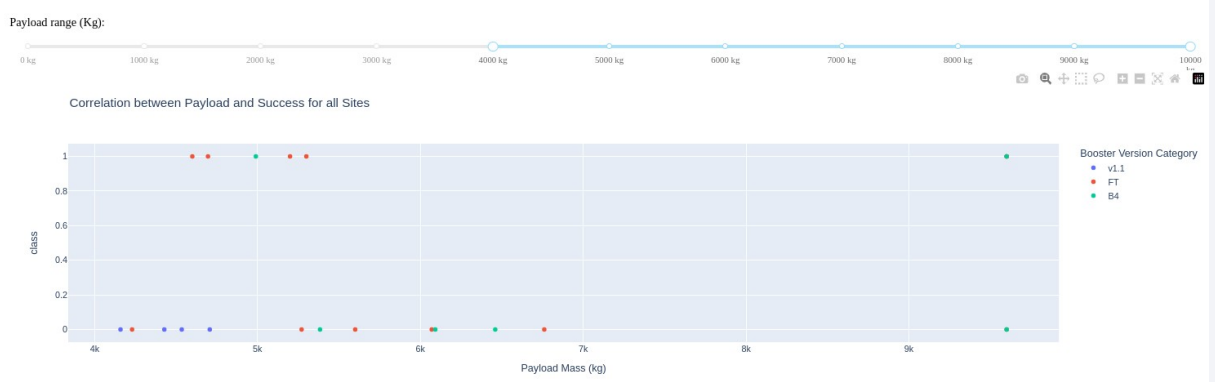
KSC LC-39A has the highest success rate - 76.9%



Payload Mass and Success

Payloads between 1,000 kg and 4,000 kg have the highest success rate

Payloads between 4,000 kg and 10,000 kg have the highest success rate



Section 6

Predictive Analysis (Classification)

Classification Accuracy

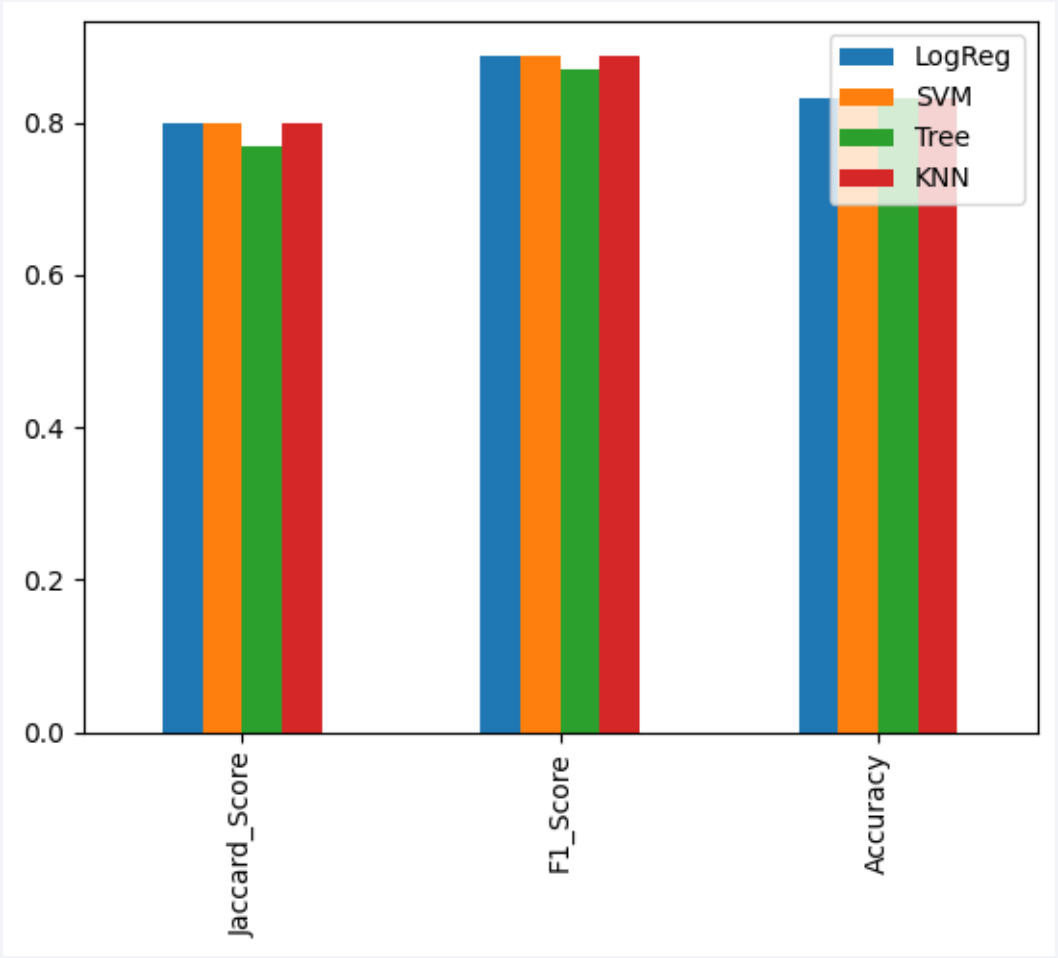
Best model is DecisionTree with a score of 0.8892857142857142

Best params is : {'criterion': 'entropy', 'max_depth': 16, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}

```
[36]: 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])
```

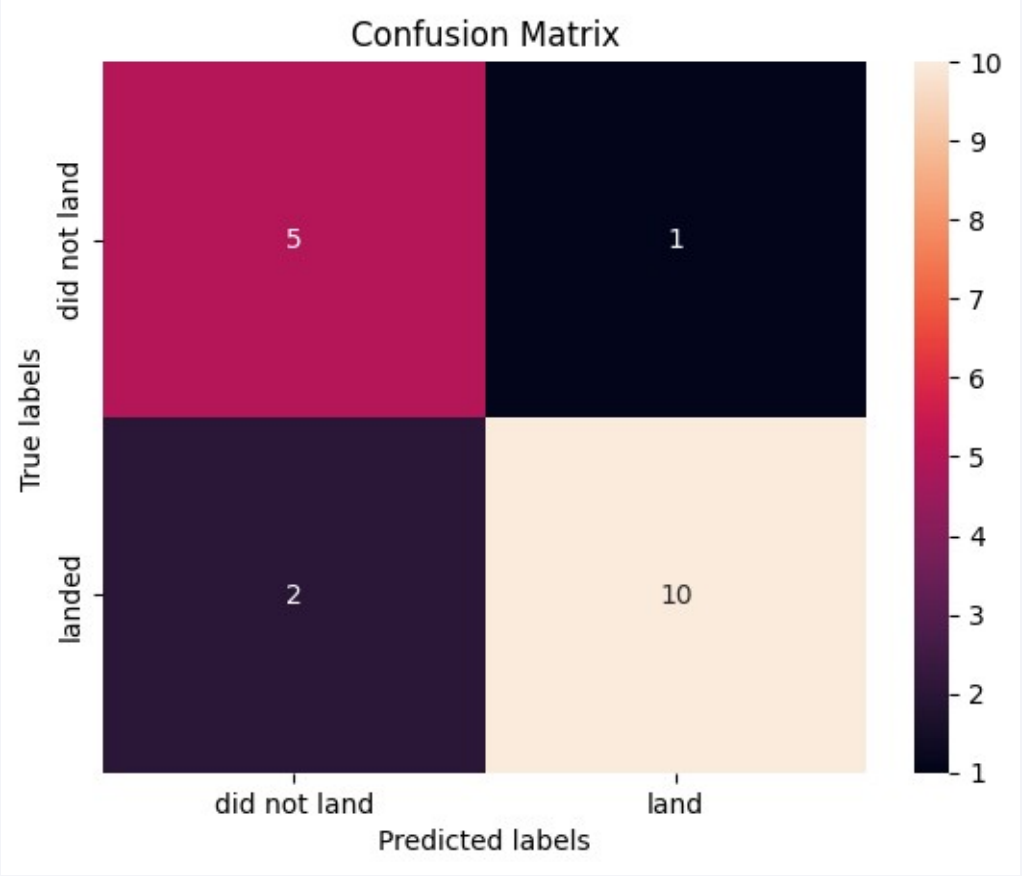
```
Best model is DecisionTree with a score of 0.8892857142857142
```



Confusion Matrix

Confusion matrix for decision tree classifier shows that the classifier can distinguish between different classes.

The main problem is false positives, when an unsuccessful landing is marked as a successful landing by the classifier



Conclusions

- The Decision tree classifier is the best machine learning algorithm for this task
- Coast: All the launch sites are close to the coast
- Launch Success: Increases over time
- KSC LC-39A: Has the highest success rate among launch sites.
- Orbits: ES-L1, GEO, HEO, and SSO have a 100% success rate
- Payload Mass: Across all launch sites, the higher the payload mass (kg), the higher the success rate

Thank you!