Winning Space Race with Data Science

Timothy L. Clark

21 March 2023

https://github.com/clarkti5/IBM-Applied-Data-Science-Capstone

Outline

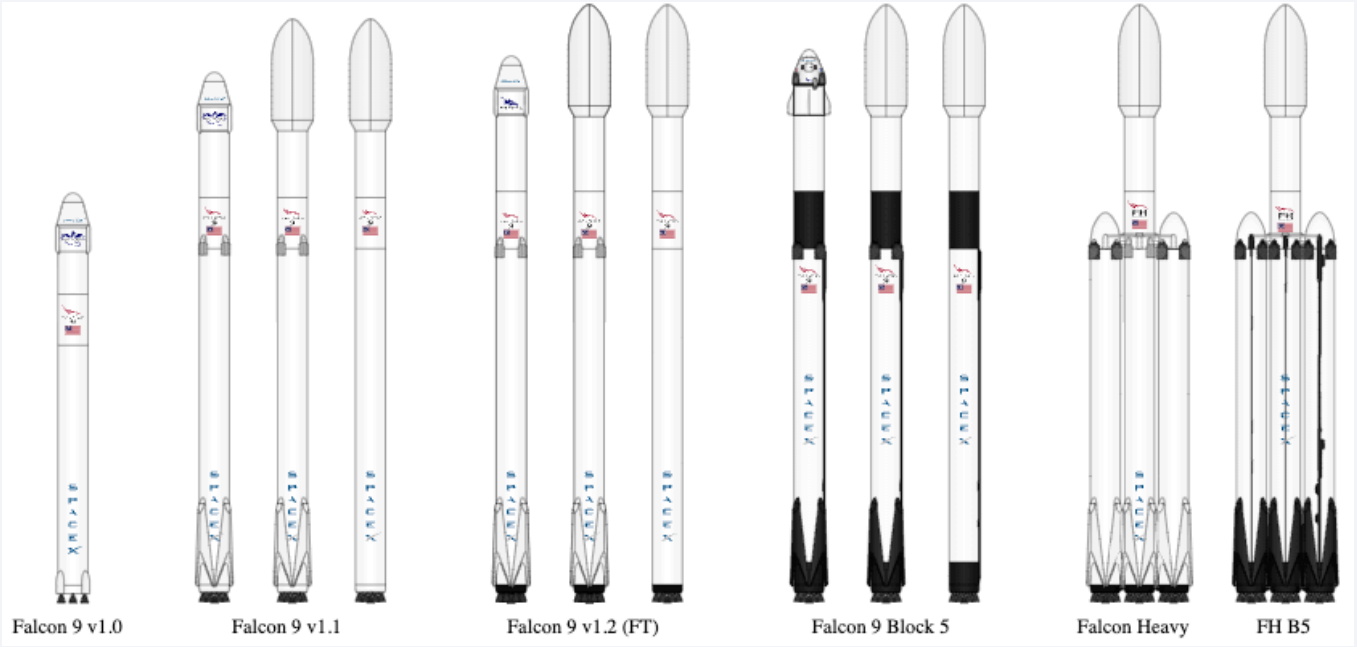
- Executive Summary
- Introduction
- Methodology and Results
- Insights
- Conclusion
- Appendix

Executive Summary

- We collected historical Falcon 9 launch data from the SpaceX launch API along with information available on Wikipedia.
- After data processing and exploratory data analysis with some basic visualizations and SQL queries, we performed features engineering on the data.
- We then utilized interactive analytics to investigate how the launch success rate is related to the data features, including flight number, launch site location, orbit type, and payload mass.
- Finally, we trained and tuned 5 machine learning models to predict whether the first stage of a Falcon 9 rocket will land successfully and be able to be reused.
- Each of the 5 models obtained similar performance, with an 83% prediction accuracy on the test data.

Introduction

- Much of the cost savings of the SpaceX Falcon 9 rocket come from the fact that the first stage can be reused if landed successfully.
- By predicting whether or not the first stage can be successfully recovered, we can estimate the cost of a launch.
- Can we predict whether or not the first stage of a Falcon 9 rocket will land successfully?



The Falcon 9 rocket family. Image by Lucabon (based on work of Markus Säynevirta and Craigboy and Rressi) - Falcon rocket family4.svg, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=66199394

Section 1

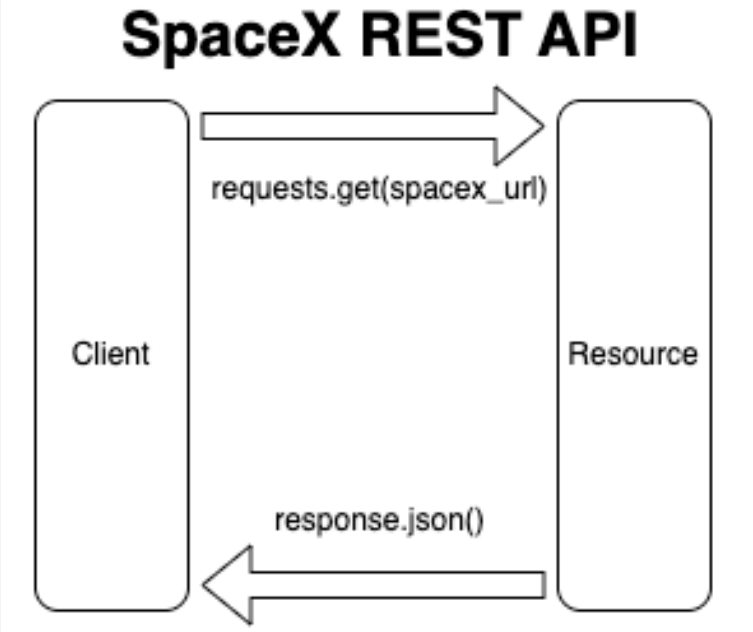
Methodology

Methodology Overview

- Data Collection
  - SpaceX API
  - Web Scraping
- Data Processing
- Exploratory Analysis
  - SQL
  - Visualization
- Features Engineering
- Visual Analytics
- Predictive Analytics

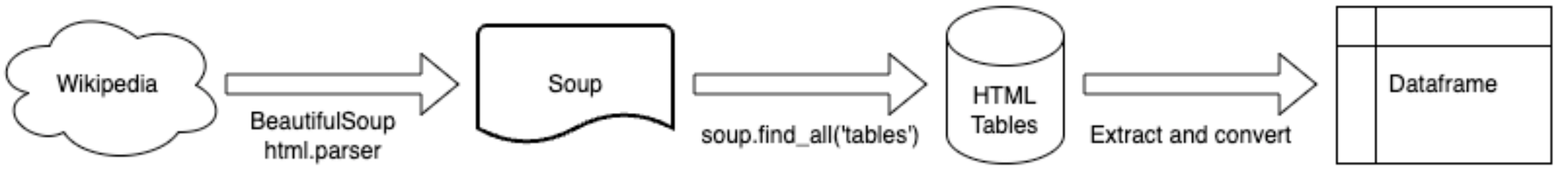
Data Collection — SpaceX API

- We made a request from the SpaceX REST API at `https://api.spacexdata.com/v4/launches/past` to collect historical launch data.
- The request was successful and the response was converted to a Pandas dataframe using `.json_normalize()`.



For more detail, see [Data collection via the SpaceX API.ipynb]

Data Collection — Web Scraping



- Additional data was scraped from Wikipedia tables at https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922 using `BeautifulSoup` and extracted to a list of HTML tables using `soup.find_all('tables')`.
- These tables were then converted to a dataframe.

For more detail, see `Data collection via web scraping.ipynb`

Data Processing — SpaceX API

- Much of the data collected from the SpaceX REST API referenced identification numbers. Several functions were used to get more easily interpretable information from the REST API.
- For example, `getBoosterVersion()` makes a separate request from the REST API to extract the booster version from the identification number in the original response data.
- There are analogous `getLaunchSite()`, `getPayloadData()`, and `getCoreData()` functions.

In [2]:
```
# Takes the dataset and uses the rocket column to call the API and append the data to the list
def getBoosterVersion(data):
    for x in data['rocket']:
       if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

For more detail, see Data collection via the SpaceX API.ipynb

Data Processing — SpaceX API

- After the identification numbers were replaced, several irrelevant columns were removed (e.g. `links.youtube_id`).
- Then, the data was filtered to include only the Falcon 9 launch data.
- Below is a preview of the resulting data frame named `data_falcon9`.

Out[25]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093

For more detail, see Data collection via the SpaceX API.ipynb

Data Processing — SpaceX API

- There were 5 missing payload values, which were replaced by the average payload.
- Missing values for `LandingPad` indicate when a landing pad was not used, so they were retained.

In [27]:
```
# Calculate the mean value of PayloadMass column
mean_PayloadMass = data_falcon9['PayloadMass'].mean()
# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'].replace(np.nan, mean_PayloadMass, inplace=True)

#Check the missing values in our dataset again
data_falcon9.isnull().sum()
```

Out[27]:
```
FlightNumber      0
Date              0
BoosterVersion    0
PayloadMass       0
Orbit             0
LaunchSite        0
Outcome           0
Flights           0
GridFins          0
Reused            0
Legs              0
LandingPad       26
Block             0
ReusedCount       0
Serial            0
Longitude         0
Latitude          0
dtype: int64
```

For more detail, see [Data collection via the SpaceX API.ipynb]

Data Processing — Web Scraped Data

- From the list of HTML tables scraped from `BeautifulSoup`, the relevant data was parsed into a Python dictionary named `launch_dict`, then converted into a Pandas dataframe, previewed below.

```
In [15]: df=pd.DataFrame(launch_dict)
         df.head()
```

Out[15]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing	Date	Time
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure	4 June 2010	18:45
1	2	CCAFS	Dragon	0	LEO	NASA	Success	F9 v1.0B0004.1	Failure	8 December 2010	15:43
2	3	CCAFS	Dragon	525 kg	LEO	NASA	Success	F9 v1.0B0005.1	No attempt\n	22 May 2012	07:44
3	4	CCAFS	SpaceX CRS-1	4,700 kg	LEO	NASA	Success\n	F9 v1.0B0006.1	No attempt	8 October 2012	00:35
4	5	CCAFS	SpaceX CRS-2	4,877 kg	LEO	NASA	Success\n	F9 v1.0B0007.1	No attempt\n	1 March 2013	15:10

For more detail, see Data collection via web scraping.ipynb

Exploratory Data Analysis — SQL

- Using SQL, we explored information regarding:
  - Launch site names.
  - Payload masses (e.g. average payload, total payload carried by NASA, boosters carrying the maximum payload).
  - Mission outcomes (e.g. total successful/failure missions).
  - Dates of successful/failure missions.

For more detail, see EDA with SQL.ipynb

Exploratory Data Analysis — Visualization

- Using `Seaborn`, we explored a variety of visualizations related to:
  - Flight number.
  - Launch site.
  - Payload mass.
  - Orbit type.
- Ultimately, we wanted to visualize how these features relate to one another as well as to the launch success rate.

For more detail, see `Visualization and features engineering.ipynb`

Features Engineering

- From the SpaceX API data, there was a column labeled `Outcome` which tracked the landing outcome for the first stage of a Falcon 9 launch.
- This column had 8 possibilities, shown here.

```
In [8]: landing_outcomes = df['Outcome'].value_counts()

In [9]: landing_outcomes

Out[9]: True ASDS       41
        None None       19
        True RTLS       14
        False ASDS       6
        True Ocean       5
        False Ocean      2
        None ASDS        2
        False RTLS       1
        Name: Outcome, dtype: int64
```

For more detail, see Visualization and features engineering.ipynb

Features Engineering

- Since we are only concerned with whether the landing was a success or not, we introduced a column `Class` that indicates a successful landing as 1 and a failure as 0.
- The `Outcome` column was then dropped.
- One-hot encoding was applied on the remaining categorical variables using `.get_dummies()`

```
In [14]: df.head(5)
```

Out[14]:

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude	Class
0	1	2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0003	-80.577366	28.561857	0
1	2	2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0005	-80.577366	28.561857	0
2	3	2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B0007	-80.577366	28.561857	0
3	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B1003	-120.610829	34.632093	0
4	5	2013-12-03	Falcon 9	3170.000000	GTO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B1004	-80.577366	28.561857	0

For more detail, see [Visualization and features engineering.ipynb]

Visual Analytics — Folium and Plotly Dash

- We visualized launch site locations using Folium to try to understand how launch sites are chosen and any relationship between launch site and launch success rate.
- Markers were added for launch site locations, along with indicators for total number of launches and launch outcomes.
- We investigated launch site proximities to coasts, cities, roads, and railroads.
- We also created an interactive dashboard using Plotly Dash to investigate the launch success rates as related to launch site and payload mass.

For more detail, see `Exploring launch site locations with Folium.ipynb` and `Launch records dashboard.ipynb`

Predictive Analytics — Training

- We trained 5 classification machine learning models in an attempt to predict the launch outcome.
- After normalizing with `StandardScaler` and splitting the data into training and testing sets with `train_test_split`, we trained the following models from `scikit-learn`:
  - Logistic Regression
  - Support Vector Machine
  - Decision Tree
  - *k*-Nearest Neighbors
  - XGBClassifier
- Note that the test data had only 18 entries, so the resulting models were difficult to distinguish from one another performance-wise.

```
In [11]: Y_test.shape
Out[11]: (18,)
```

For more detail, see Classification model comparisons.ipynb

Predictive Analytics — Tuning

- Hyperparameters were tuned using 10-fold cross validation with `GridSearchCV`.
- Model performance was evaluated using their accuracy score on the test data.

In [22]:
```
parameters = {'criterion': ['gini', 'entropy'],
     'splitter': ['best', 'random'],
     'max_depth': [2*n for n in range(1,10)],
     'max_features': ['auto', 'sqrt'],
     'min_samples_leaf': [1, 2, 4],
     'min_samples_split': [2, 5, 10]}

tree = DecisionTreeClassifier()
```

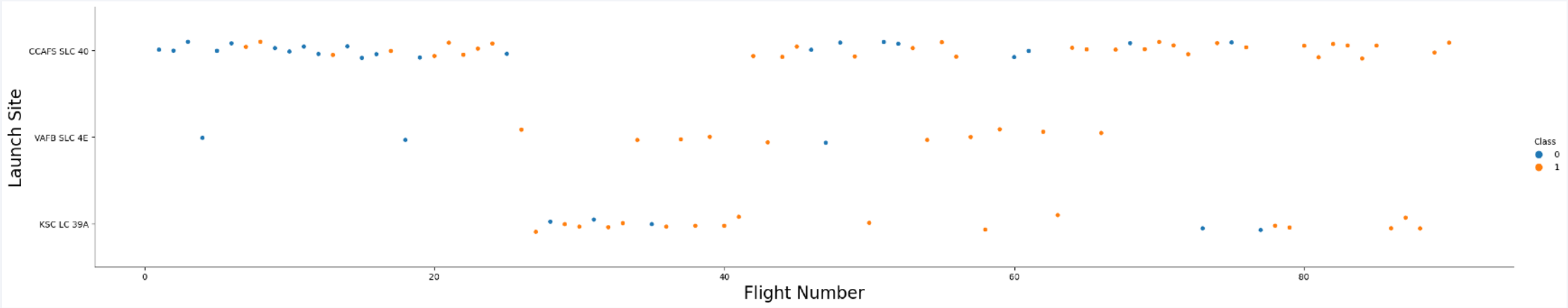
In [23]:
```
tree_cv = GridSearchCV(tree, parameters, cv=10).fit(X_train, Y_train)
```

In [24]:
```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter':
'random'}
accuracy : 0.8892857142857145
```

For more detail, see [Classification model comparisons.ipynb]

Results — Exploratory Analysis



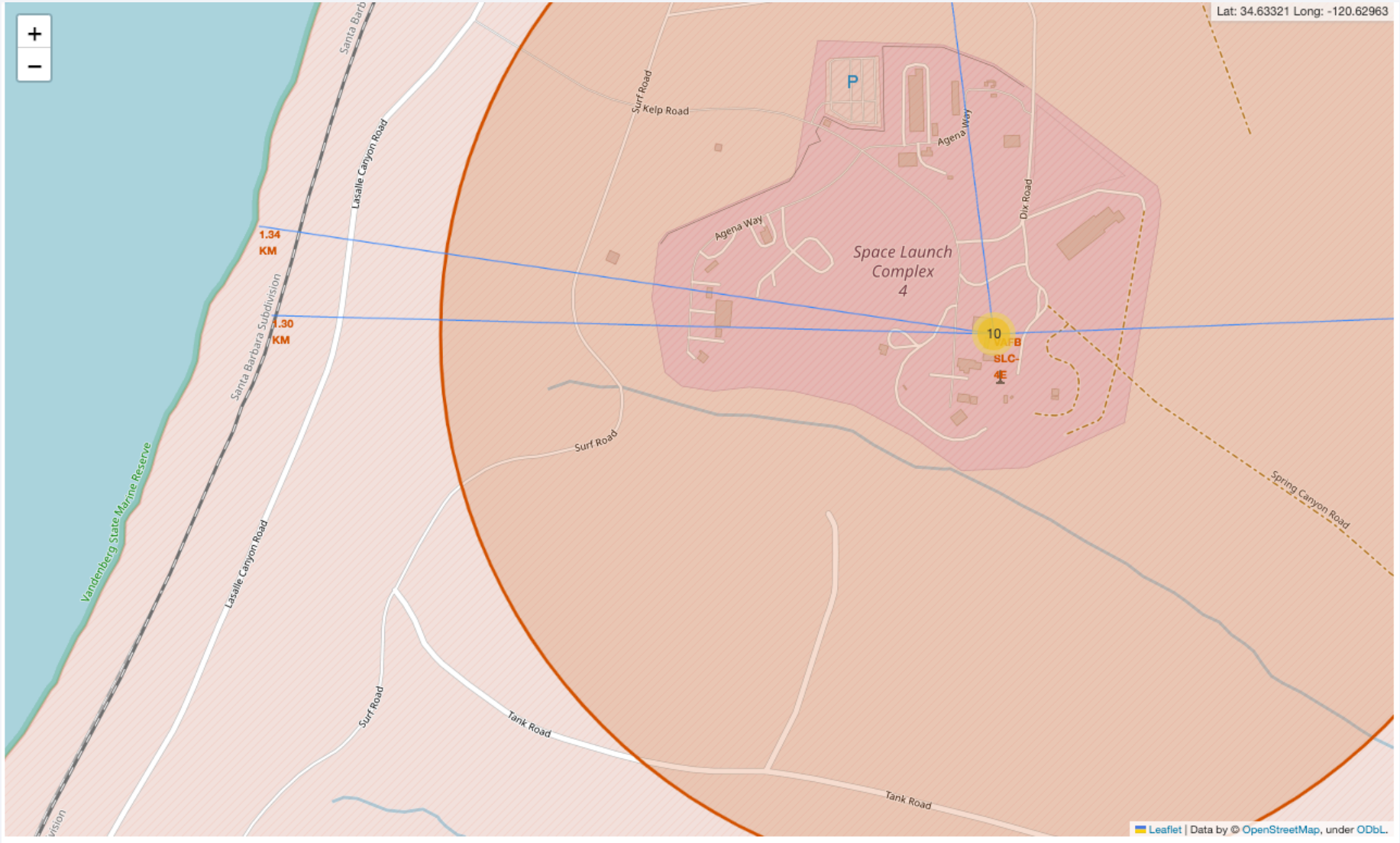
In [16]:
```
%%sql
SELECT SUBSTRING("Mission_Outcome", 1, 7) AS 'Mission Outcome', COUNT("Mission_Outcome") AS 'Mission Outcome Total'
FROM SPACEXTBL
GROUP BY SUBSTRING("Mission_Outcome", 1, 7);
```

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

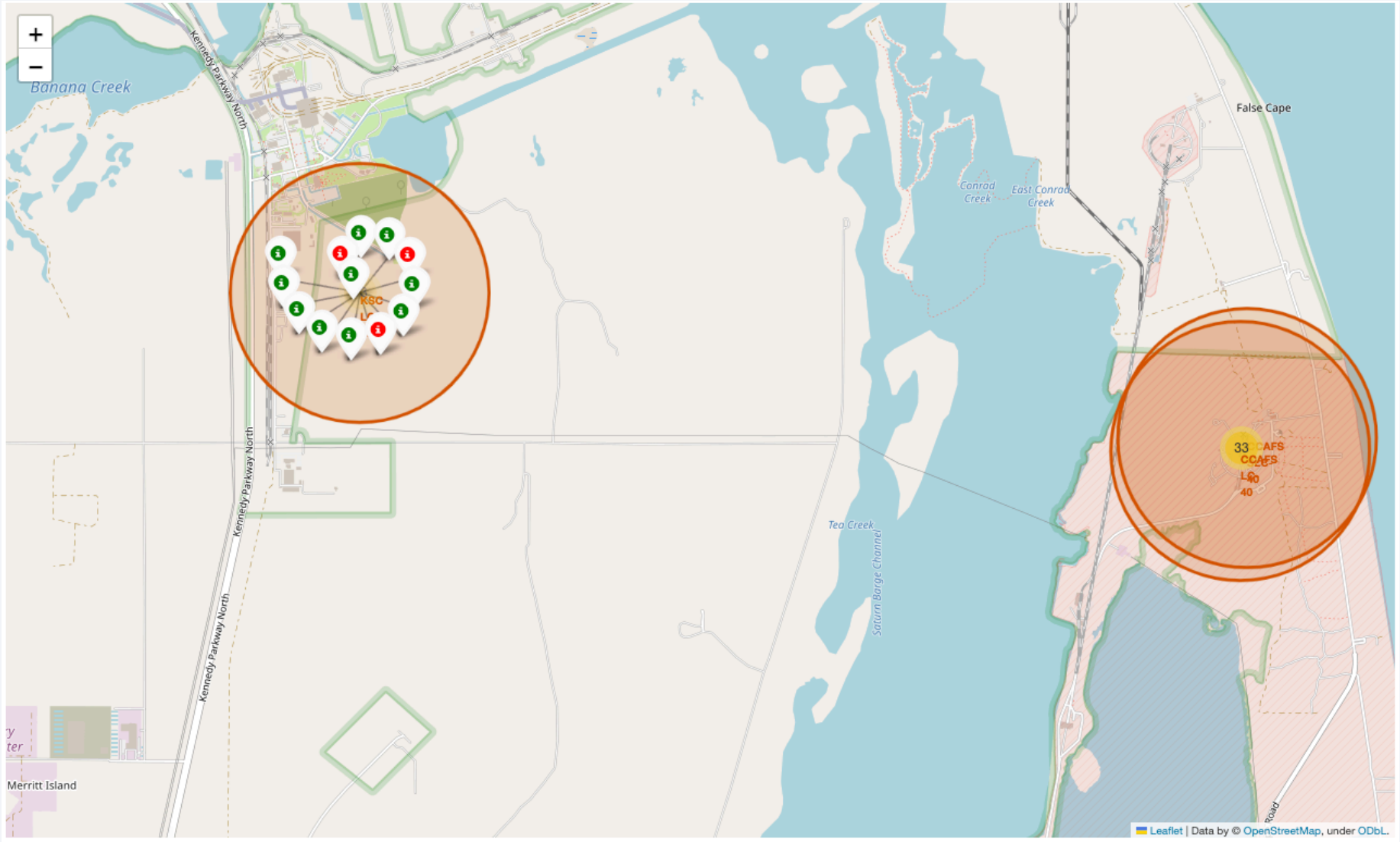
Out[16]:

Mission Outcome	Mission Outcome Total
Failure	1
Success	100

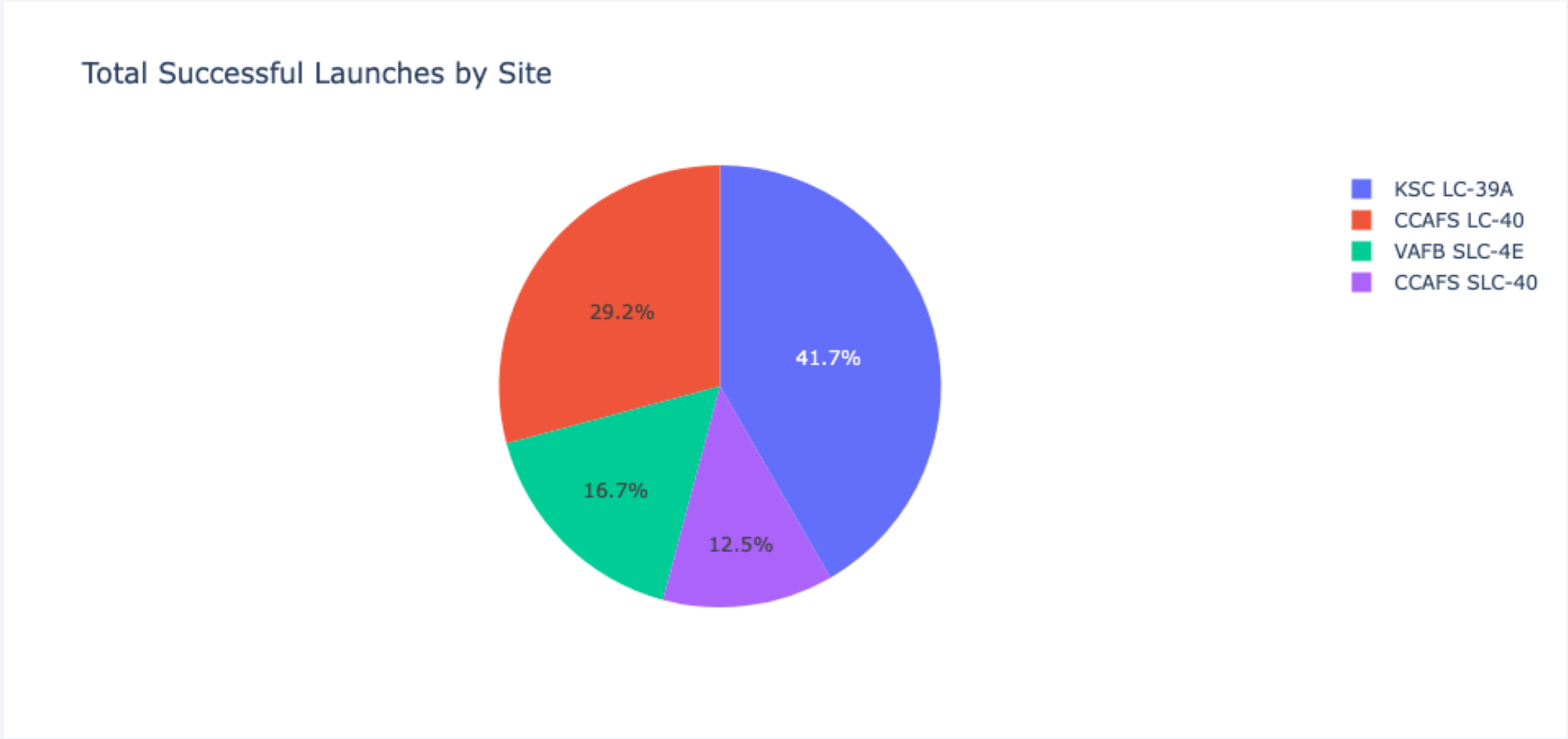
Results — Visual Analytics with Folium



Results — Visual Analytics with Folium

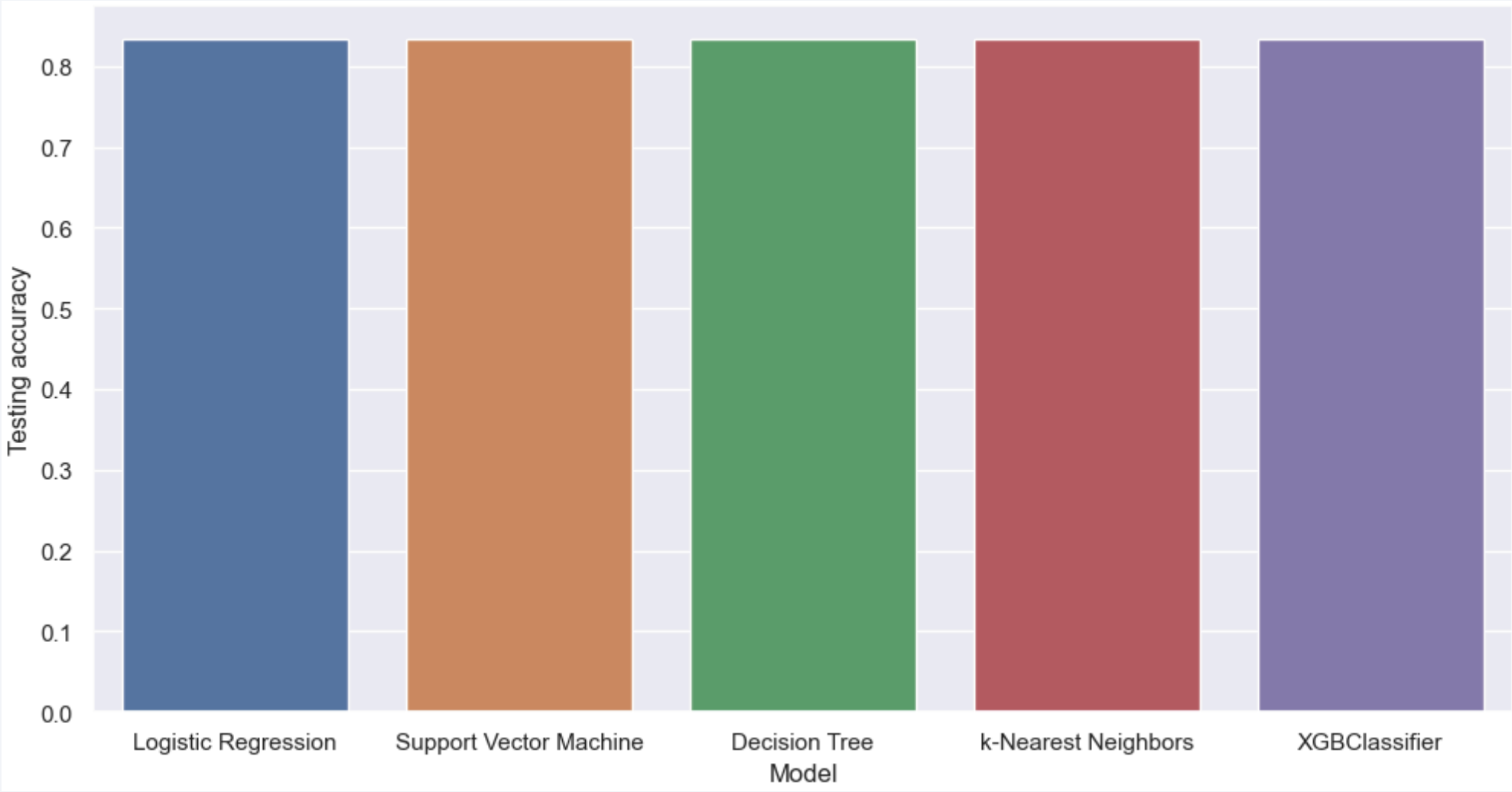


Results — Visual Analytics with Plotly Dash

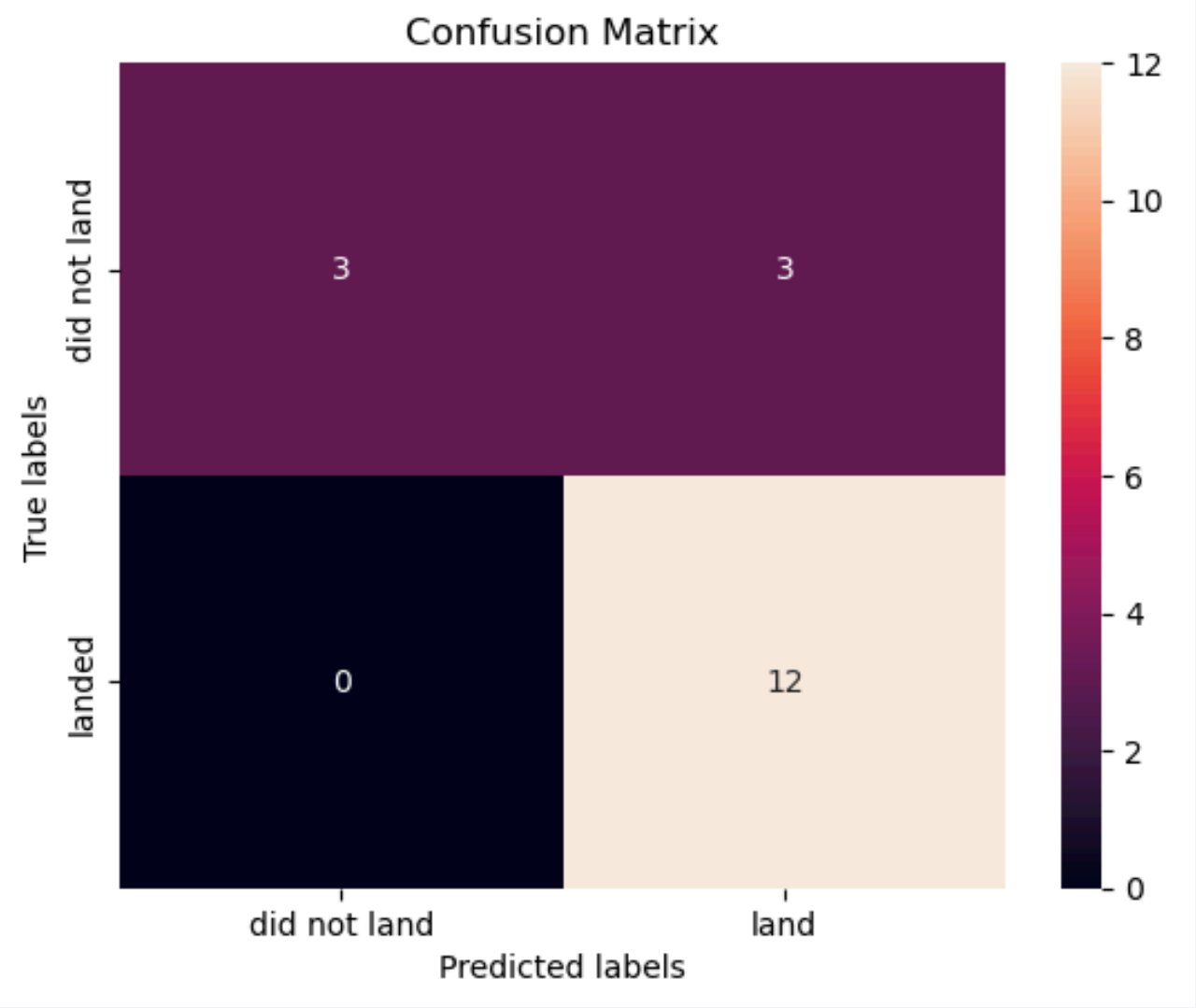


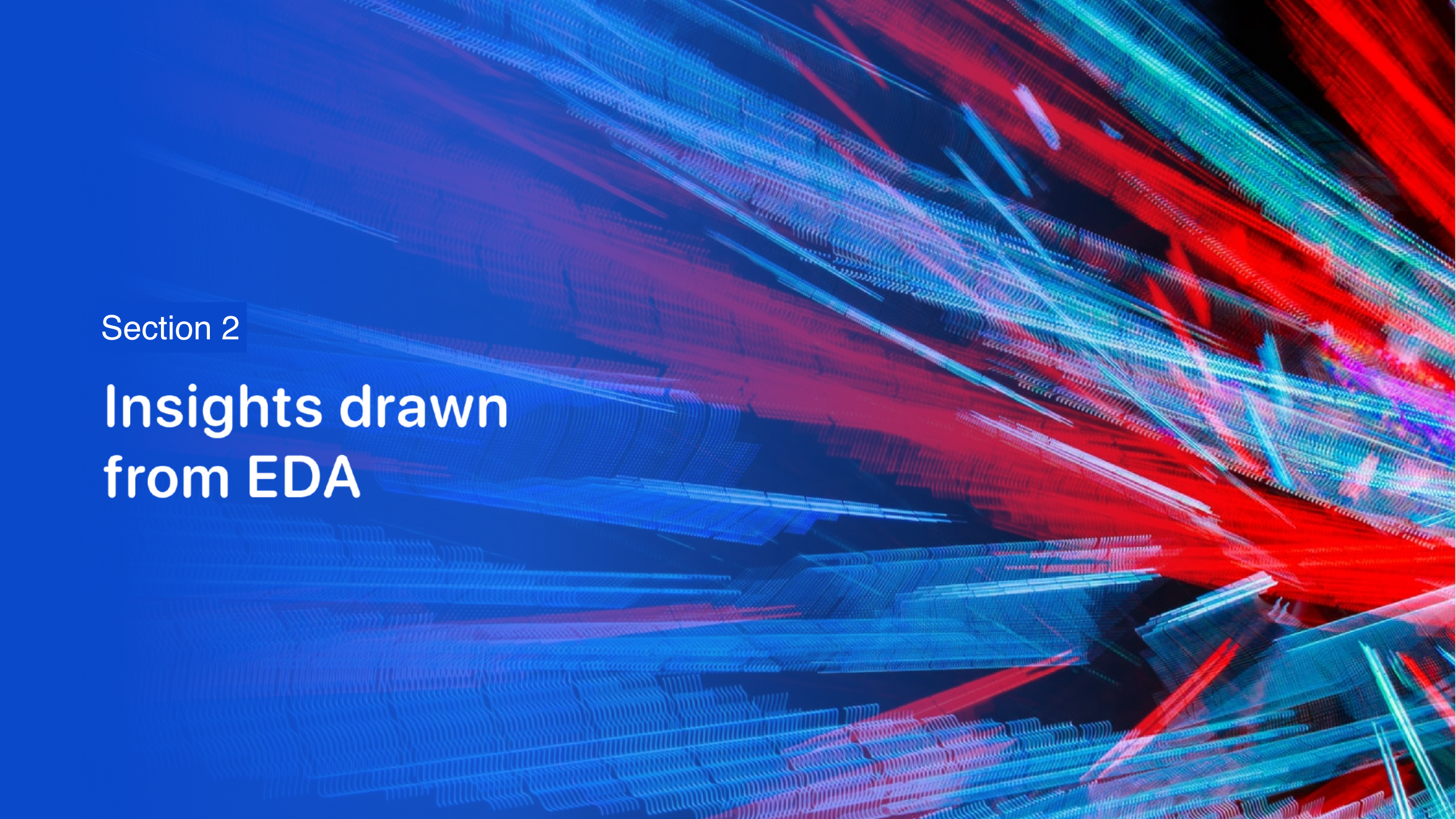
Results — Visual Analytics with Plotly Dash

Results — Predictive Analytics

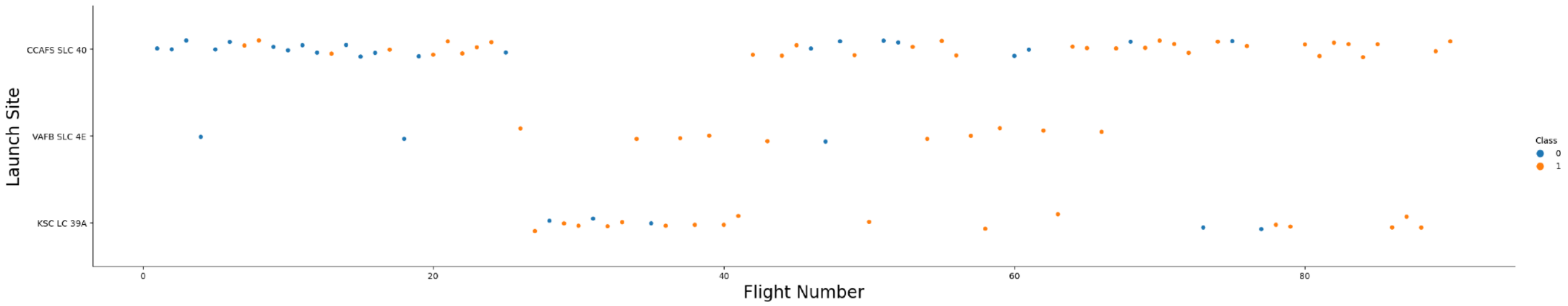


Results — Predictive Analytics



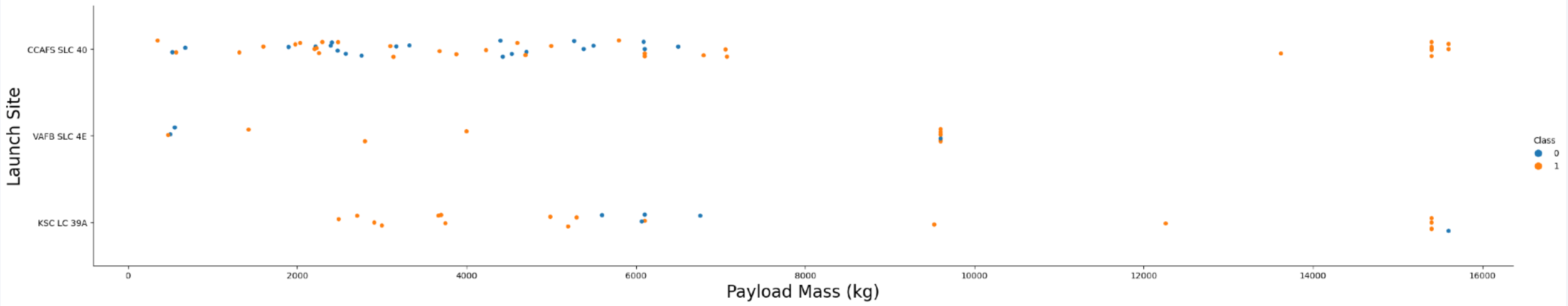

Section 2
Insights drawn
from EDA

Flight Number vs. Launch Site



- The success rate appears to increase with the flight number and different launch sites have different success rates.
- Observe how the choice of launch site changes with flight number.

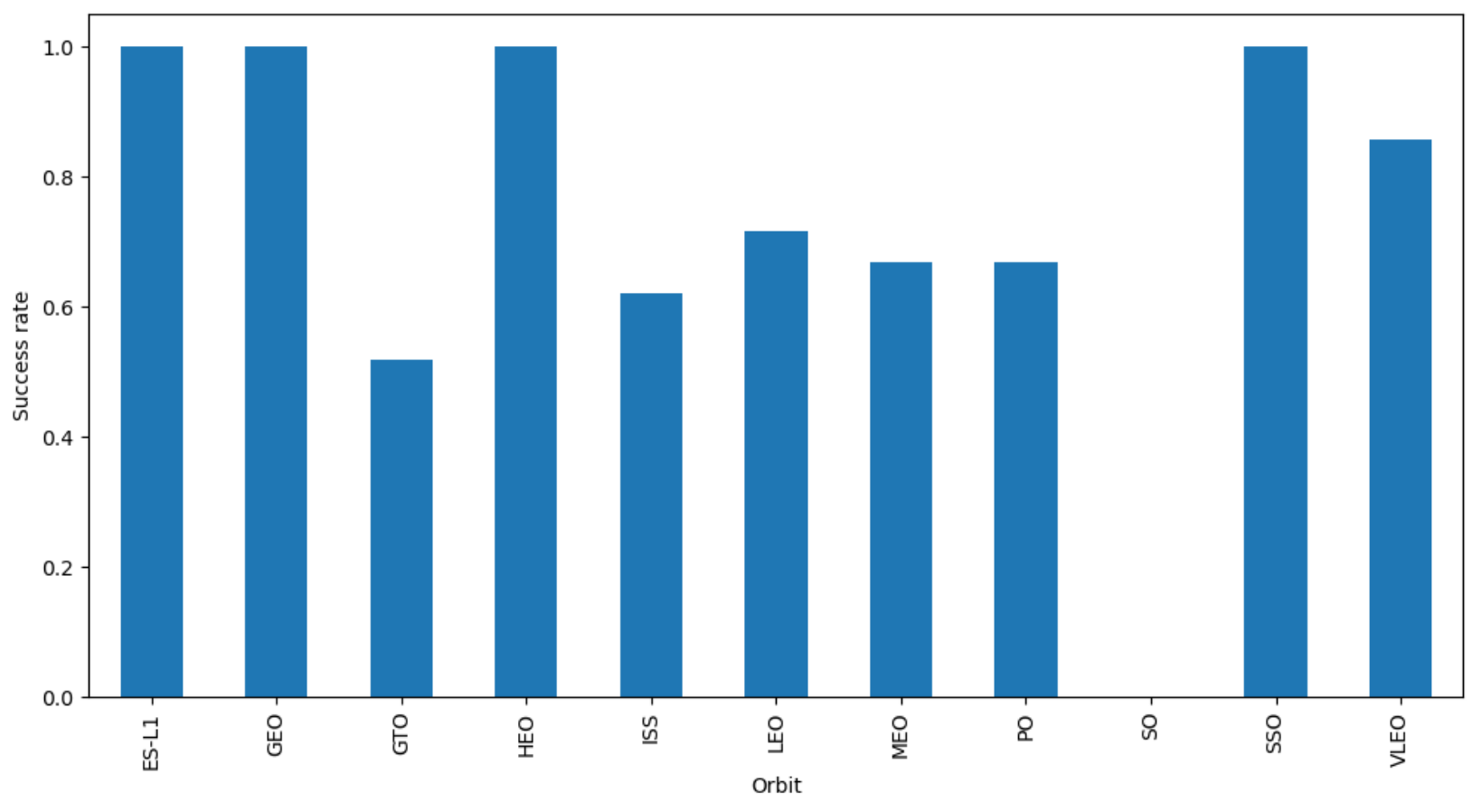
Payload vs. Launch Site



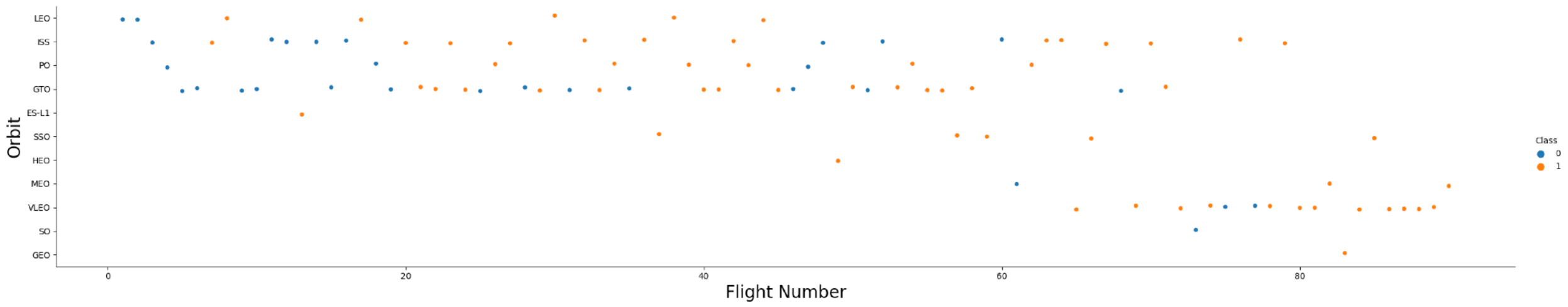
- Launch site VAFB SLC-4E has no launches with payload greater than 10,000 kg.

Success Rate vs. Orbit Type

- The orbits with the highest success rate are ES-L1, GEO, HEO, and SSO.

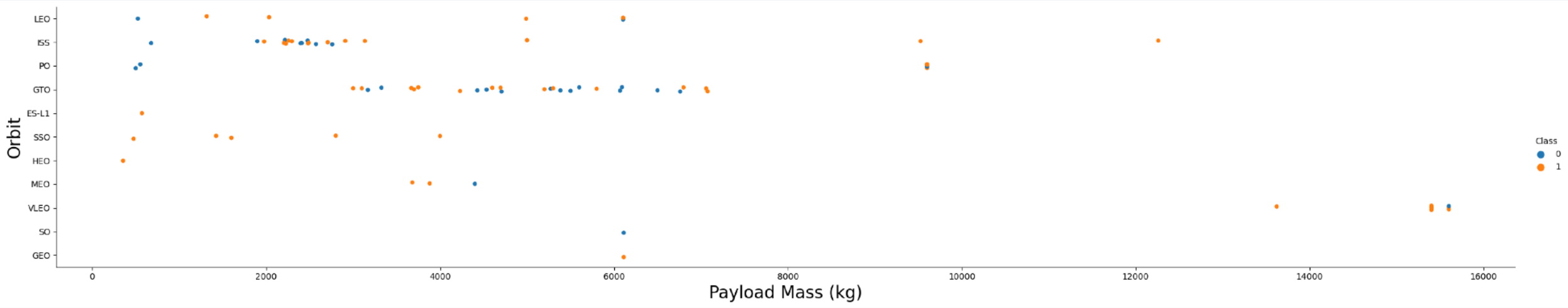


Flight Number vs. Orbit Type



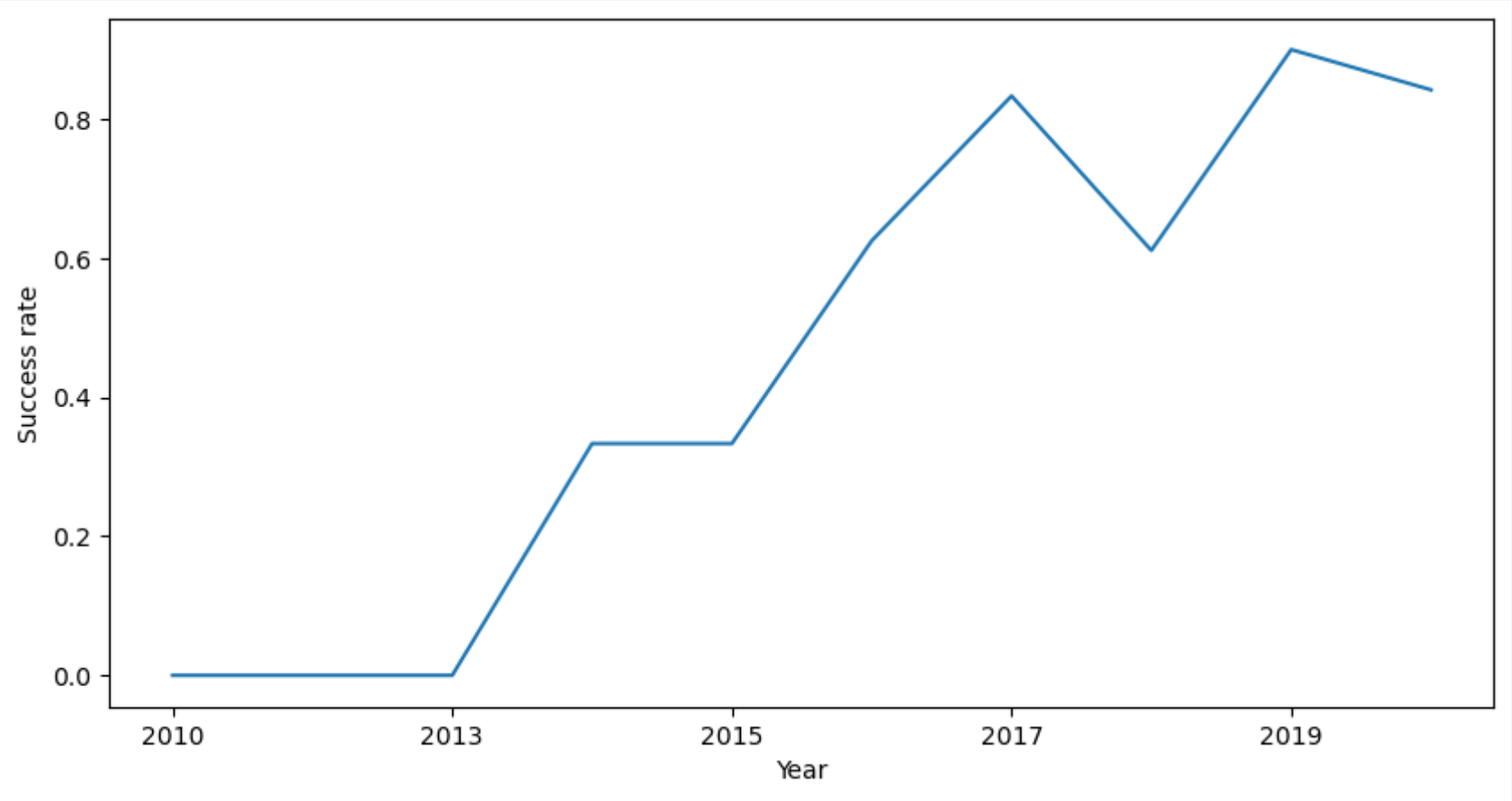
- The success rate for LEO increases with flight number.
- However, there does not appear to be a relationship between flight number and success rate for GTO launches.

Payload vs. Orbit Type



- Heavy payloads (greater than 10,000 kg) have more success with LEO and ISS launches.

Launch Success Yearly Trend



- The launch success rate has increased from 2010 to 2020.

All Launch Site Names

- There are four distinct launch sites used by SpaceX for Falcon 9 launches, as shown here.

In [7]:
```
%%sql
SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

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

Out[7]:

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

Launch Site Names Begin with 'CCA'

- Two of these sites (CCAFS LC-40 and CCAFS SLC-40) are located in close proximity to each other near Cape Canaveral, Florida.
- Below are the first 5 launch records from these sites.

In [8]:
```
%%sql
SELECT * FROM SPACEXTBL
WHERE "Launch_Site" LIKE 'CCA%'
LIMIT 5;
```

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

Out[8]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass for NASA

- The total payload mass carried for NASA (CRS) was 45,956 kg.
- The total payload carried for NASA in general was 107,010 kg.

In [9]:
```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS 'Total payload mass for NASA (CRS)' FROM SPACEXTBL
WHERE "Customer" = 'NASA (CRS)';
```

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

Out[9]:

Total payload mass for NASA (CRS)
45596

In [10]:
```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") AS 'Total payload mass for NASA' FROM SPACEXTBL
WHERE "Customer" LIKE '%NASA%';
```

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

Out[10]:

Total payload mass for NASA
107010

Average Payload Mass by F9 v1.1

- The average payload mass carried by the F9 v1.1 booster was 2,535 kg.

In [11]:
```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") AS 'Average payload mass for F9 v1.1' FROM SPACEXTBL
WHERE "Booster_Version" LIKE 'F9 v1.1%';
```

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

Out[11]:

Average payload mass for F9 v1.1
2534.6666666666665

Boosters Carrying Maximum Payload

- The maximum payload was 15,600 kg carried by the boosters listed here.

In [17]:
```
%%sql
SELECT DISTINCT("Booster_Version"), "PAYLOAD_MASS__KG_" AS 'Payload_Mass' FROM SPACEXTBL
WHERE "PAYLOAD_MASS__KG_" =
    (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Out[17]:

Booster_Version	Payload_Mass
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

Successful Drone Ship Landing with Payload between 4,000–6,000 kg

- The boosters that have successfully landed on a drone ship with payload between 4,000 and 6,000 kg are shown here.

In [13]:
```
%%sql
SELECT "Booster_Version", "PAYLOAD_MASS__KG_" AS 'Payload_Mass' FROM SPACEXTBL
WHERE ("Landing _Outcome" = 'Success (drone ship)') AND ("PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000);
```

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

Out[13]:

Booster_Version	Payload_Mass
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

First Successful Ground Landing Date

- The first successful landing on a ground pad was on 2015-12-22.

In [12]:
```
%%sql
SELECT "Date", "Landing _Outcome" FROM SPACEXTBL
WHERE "Landing _Outcome" = 'Success (ground pad)'
ORDER BY "Date" DESC
LIMIT 1;
```

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

Out[12]:

Date	Landing _Outcome
22-12-2015	Success (ground pad)

Total Number of Successful and Failure Mission Outcomes

- In total, there were 100 successful landings and 1 failure.

In [16]:
```
%%sql
SELECT SUBSTRING("Mission_Outcome", 1, 7) AS 'Mission Outcome', COUNT("Mission_Outcome") AS 'Mission Outcome Total'
FROM SPACEXTBL
GROUP BY SUBSTRING("Mission_Outcome", 1, 7);
```

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

Out[16]:

Mission Outcome	Mission Outcome Total
Failure	1
Success	100

2015 Failed Launch Records

- The following describes the failed drone ship landings in 2015.

In [18]:
```
%%sql
SELECT SUBSTRING("Date", 4, 2) AS Month, "Landing _Outcome", "Booster_Version", "Launch_Site"
FROM SPACEXTBL
WHERE (SUBSTRING("Date", 7, 4) = '2015') AND ("Landing _Outcome" = 'Failure (drone ship)');
```

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

Out[18]:

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

- Below is the count of successful landing outcomes between 2010-06-04 and 2017-03-20.

In [19]:
```
%%sql
SELECT "Date", "Landing _Outcome", COUNT("Landing _Outcome") AS 'Landing_Outcome_Count' FROM SPACEXTBL
WHERE ("Landing _Outcome" LIKE '%Success%') AND (CAST(SUBSTRING("Date", 7, 4) AS Year) BETWEEN 2010 AND 2017)
GROUP BY "Landing _Outcome"
ORDER BY 'Landing_Outcome_Count' DESC;
```

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

Out[19]:

Date	Landing _Outcome	Landing_Outcome_Count
22-12-2015	Success (ground pad)	8
08-04-2016	Success (drone ship)	12


Section 3
Launch Sites
Proximities Analysis

Launch Site Locations

- There are 4 distinct launch sites.
- 1 is in California with the rest in close proximity to one another near Cape Canaveral, Florida.

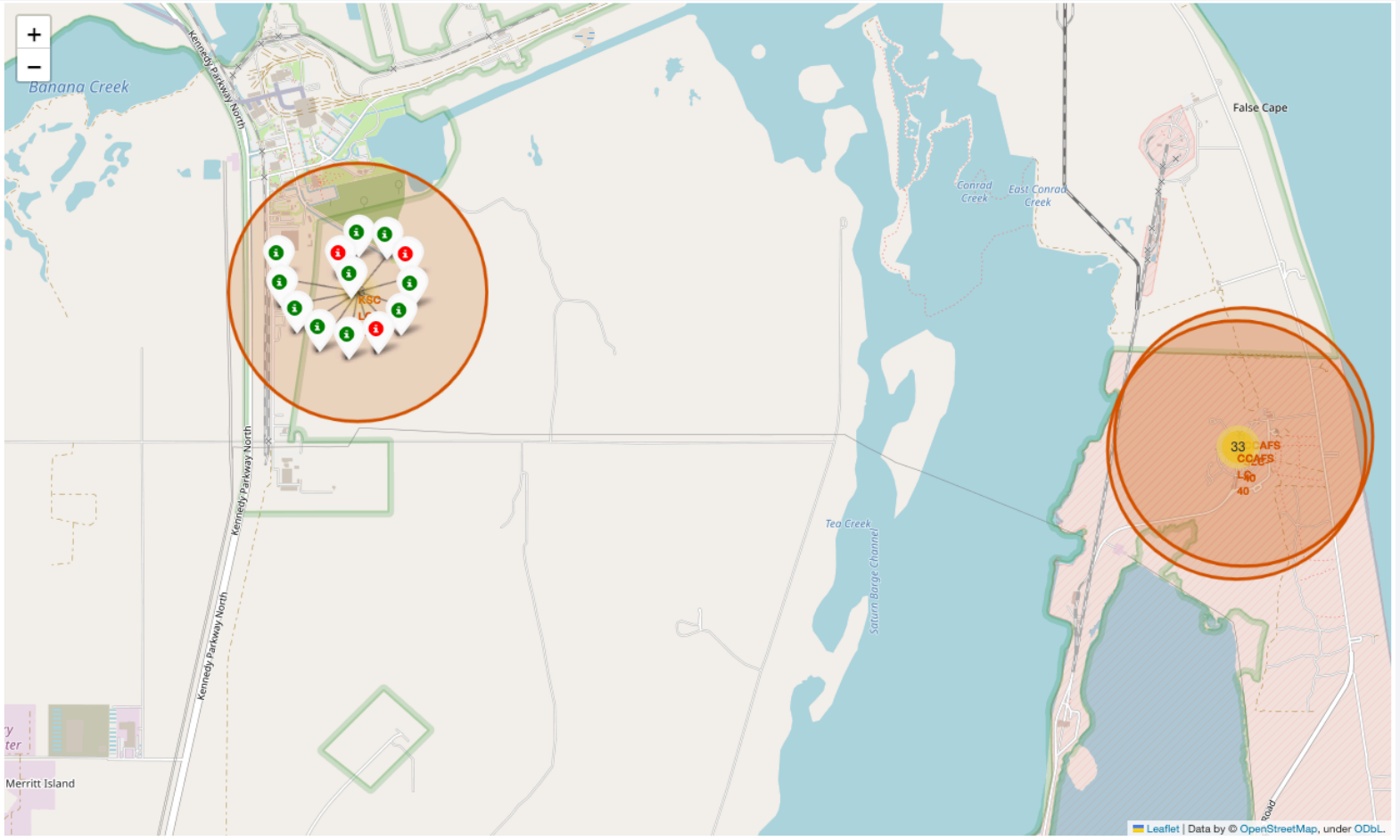
Launch Outcomes by Launch Site

- The launch site in California has 10 launches, while the Florida launch sites have a combined 46 launches.

Launch Outcomes by Launch Site

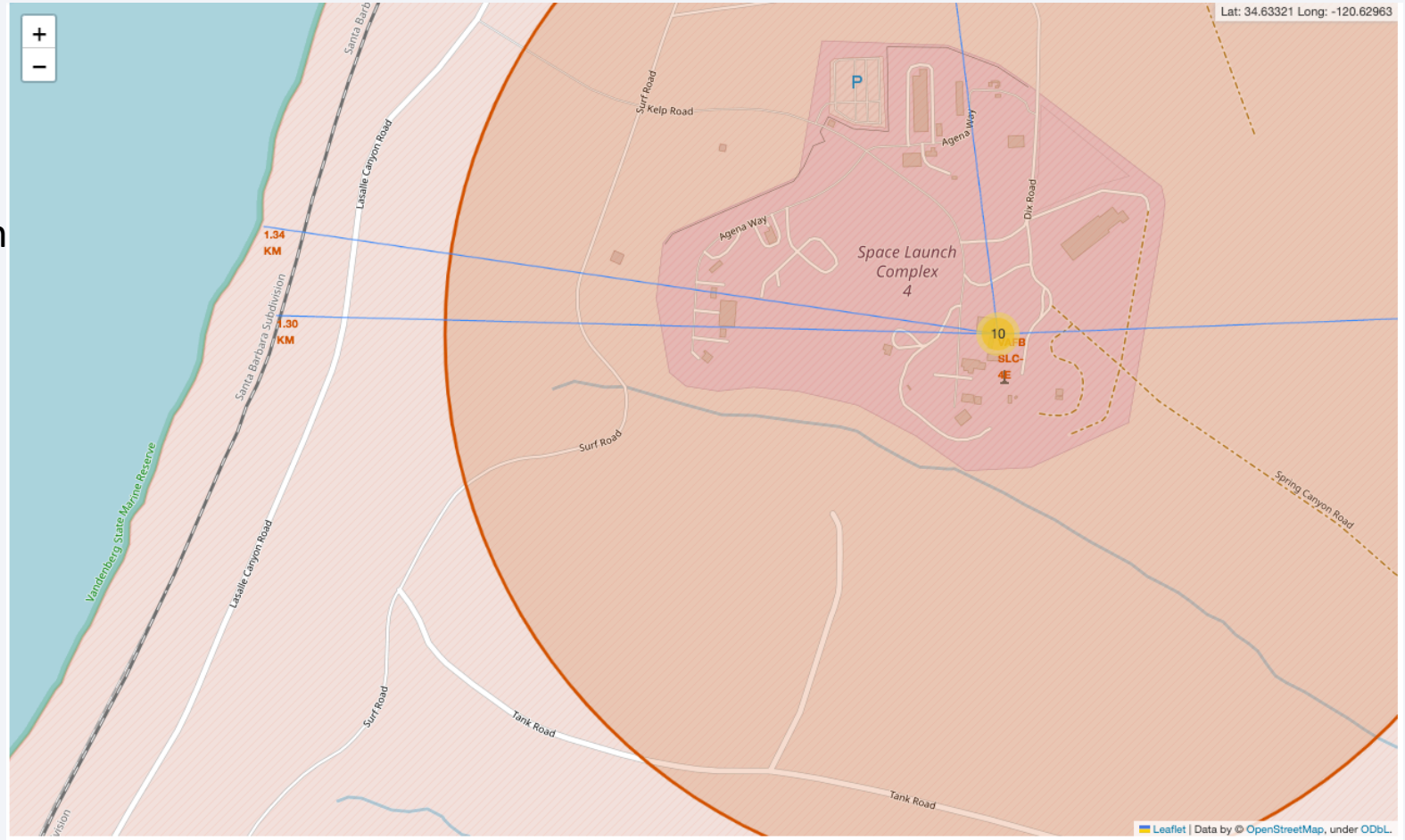
- Clicking on the marker, successful launches are colored green.

Launch site	Success rate on Folium Map
CCAFS LC-40	27%
CCAFS SLC-40	43%
KSC LC-39A	77%
VAFB SLC-4E	40%



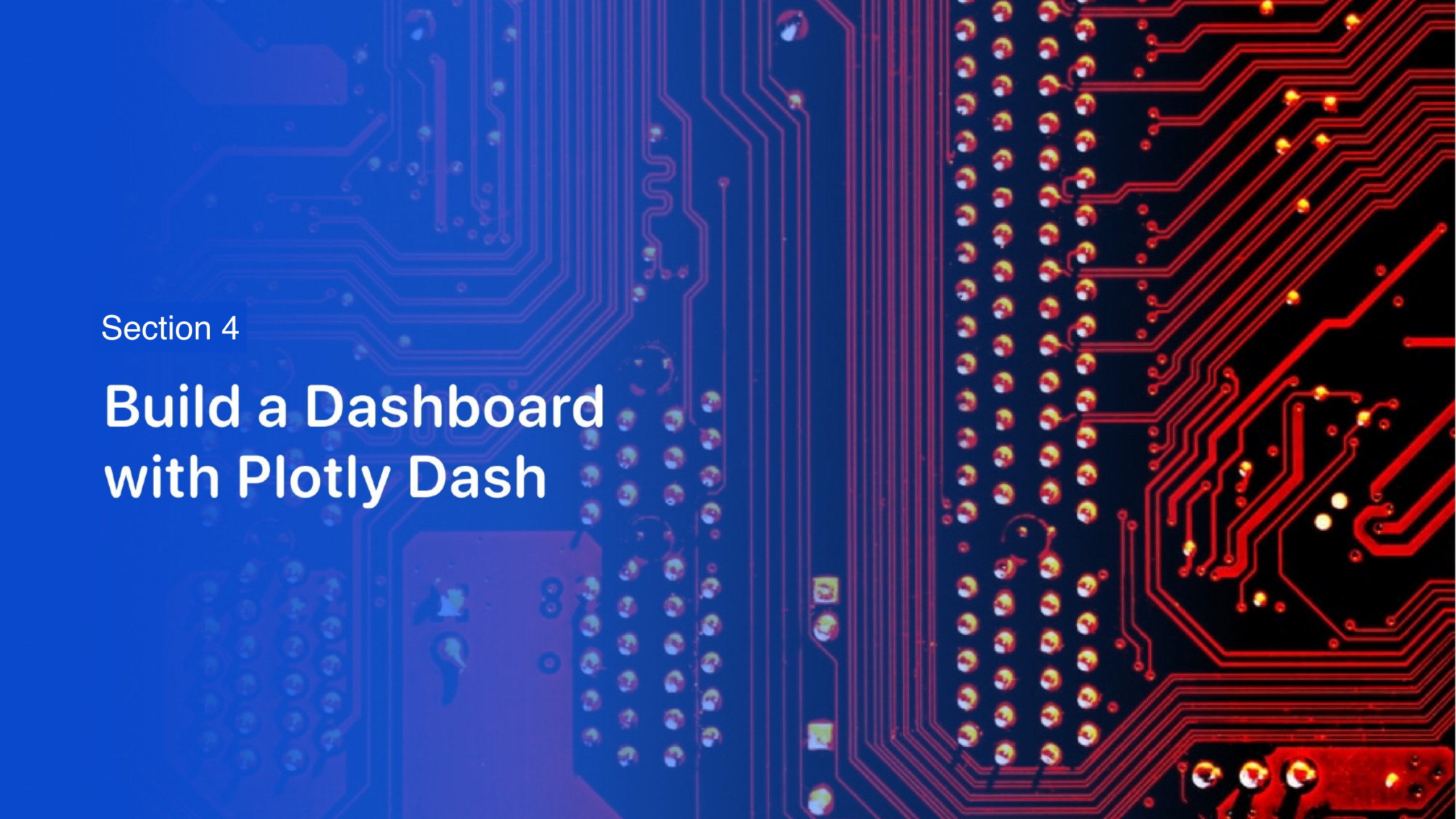
Launch Site Proximities

- Launch sites tend to be close to railroads, highways, and coastlines. This makes it easier to transport resources (including people) to and from the launch site.
- Launch sites tend to be close to coastlines but at least 15 km away from cities. These constraints can help minimize the risk of collateral damage in the event of a malfunction.



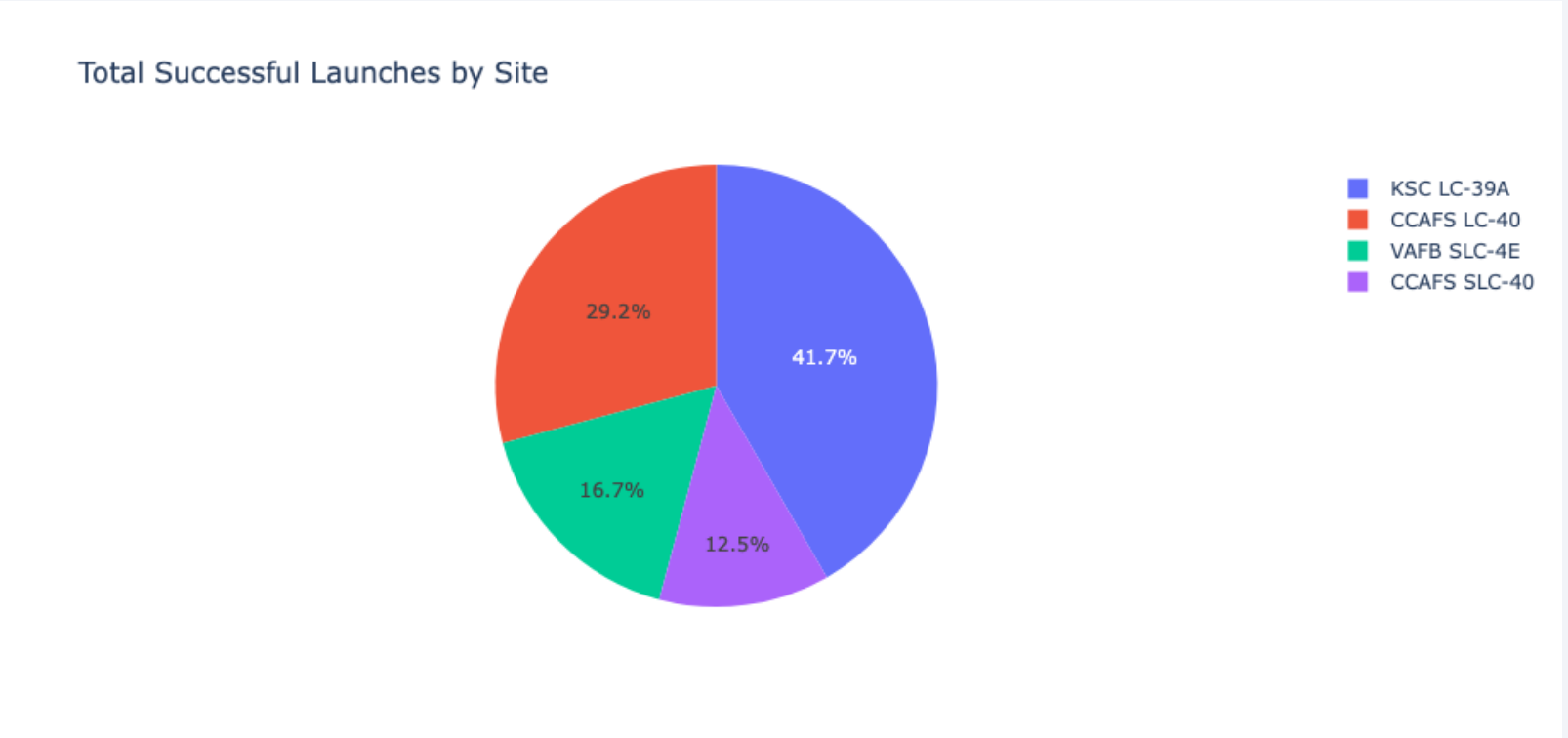
Section 4

Build a Dashboard with Plotly Dash



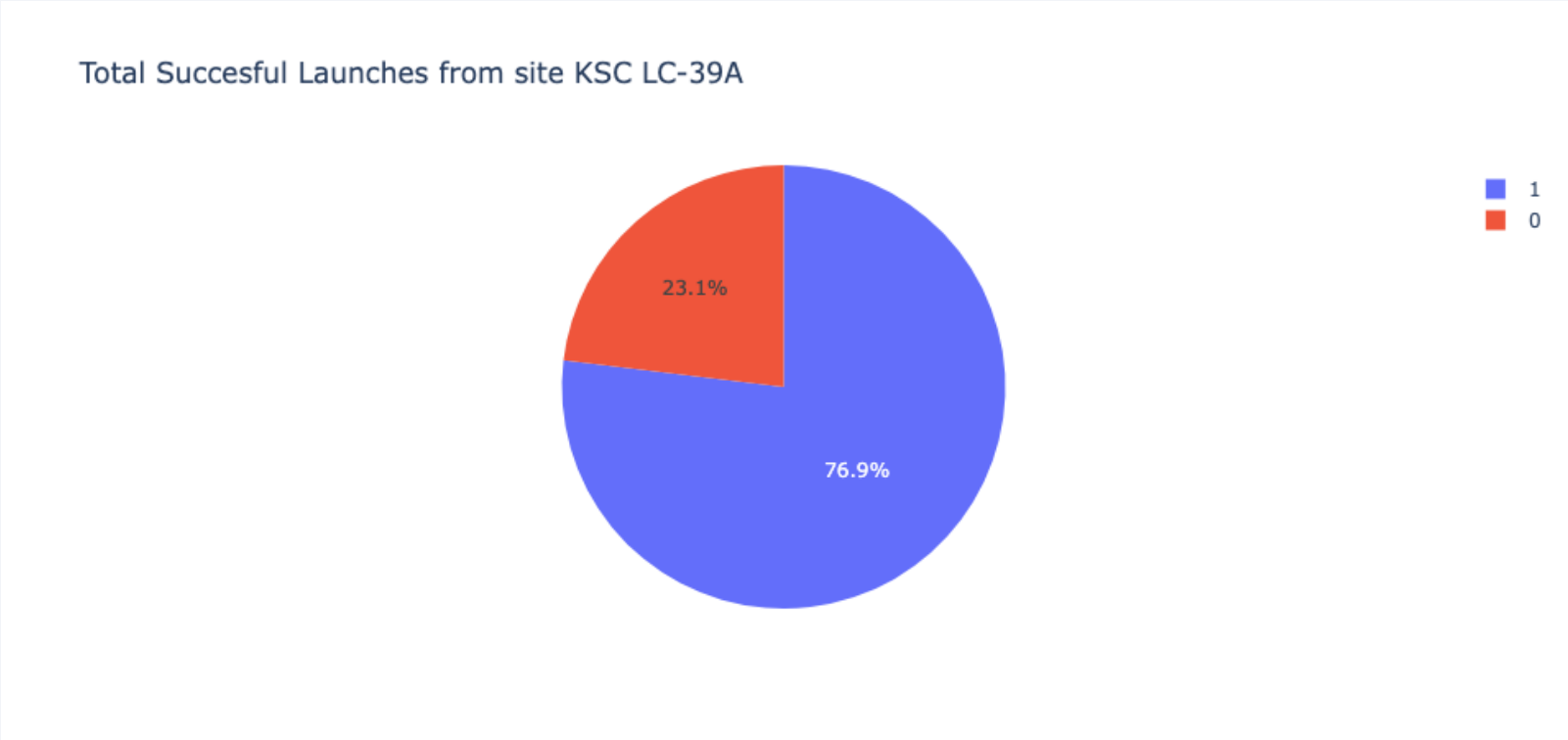
Total Successful Launches by Site

- KSC LC-39A has the highest proportion of successful launches.



Total Successful Launches from KSC LC-39A

- KSC LC-39A also has the highest launch success rate of a about 77%.



Payload Range vs. Launch Outcome — All Sites

- Generally, as payload increases, the launch success rate decreases.

Payload Range vs. Launch Outcome — Light Payload

- For the relatively light payloads shown here, the success rate is about 67%.

Payload Range vs. Launch Outcome — Heavy Payload

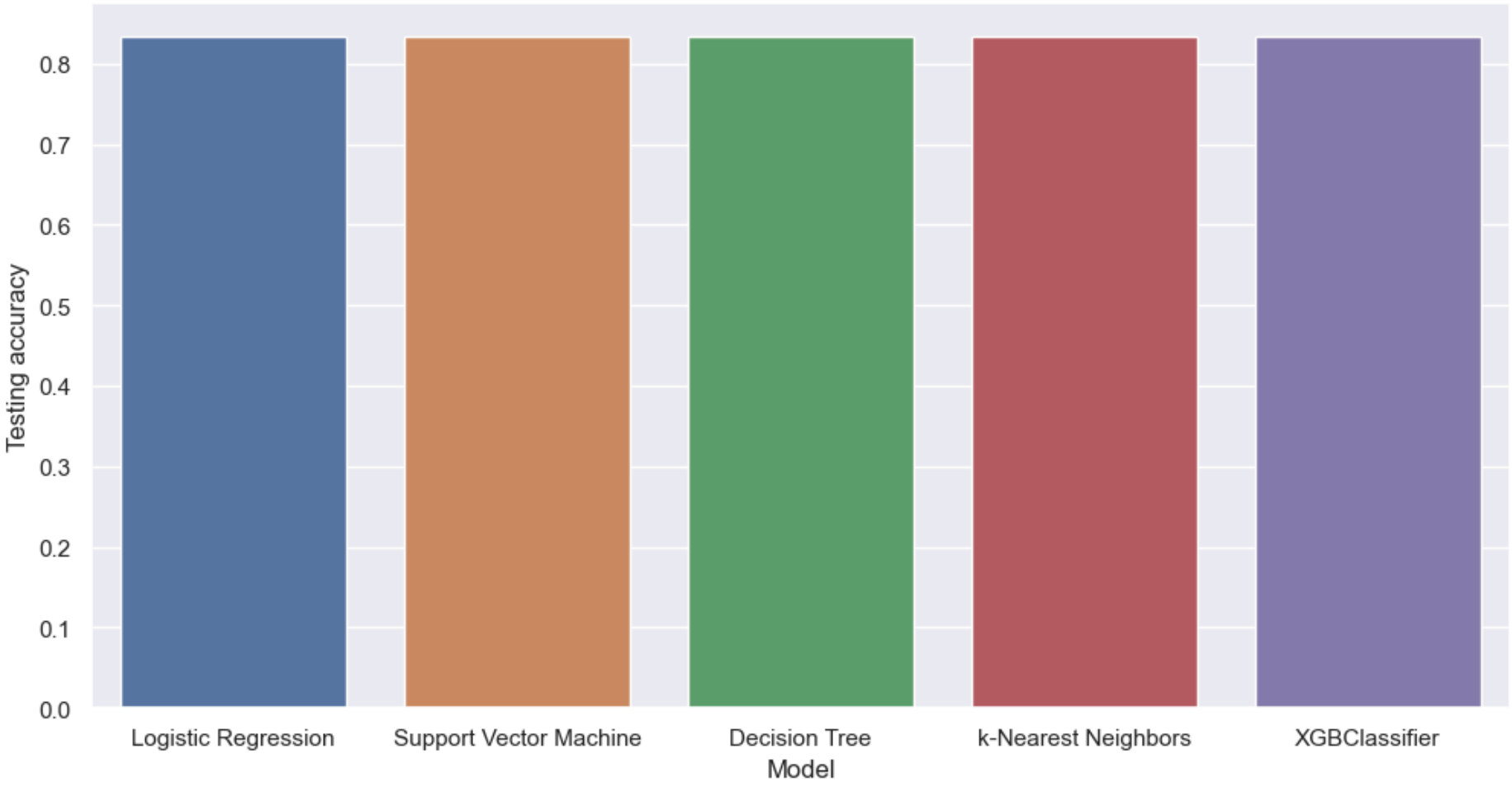
- For the heavier payloads shown here, the success rate is 0%.

Section 5

Predictive Analysis (Classification)

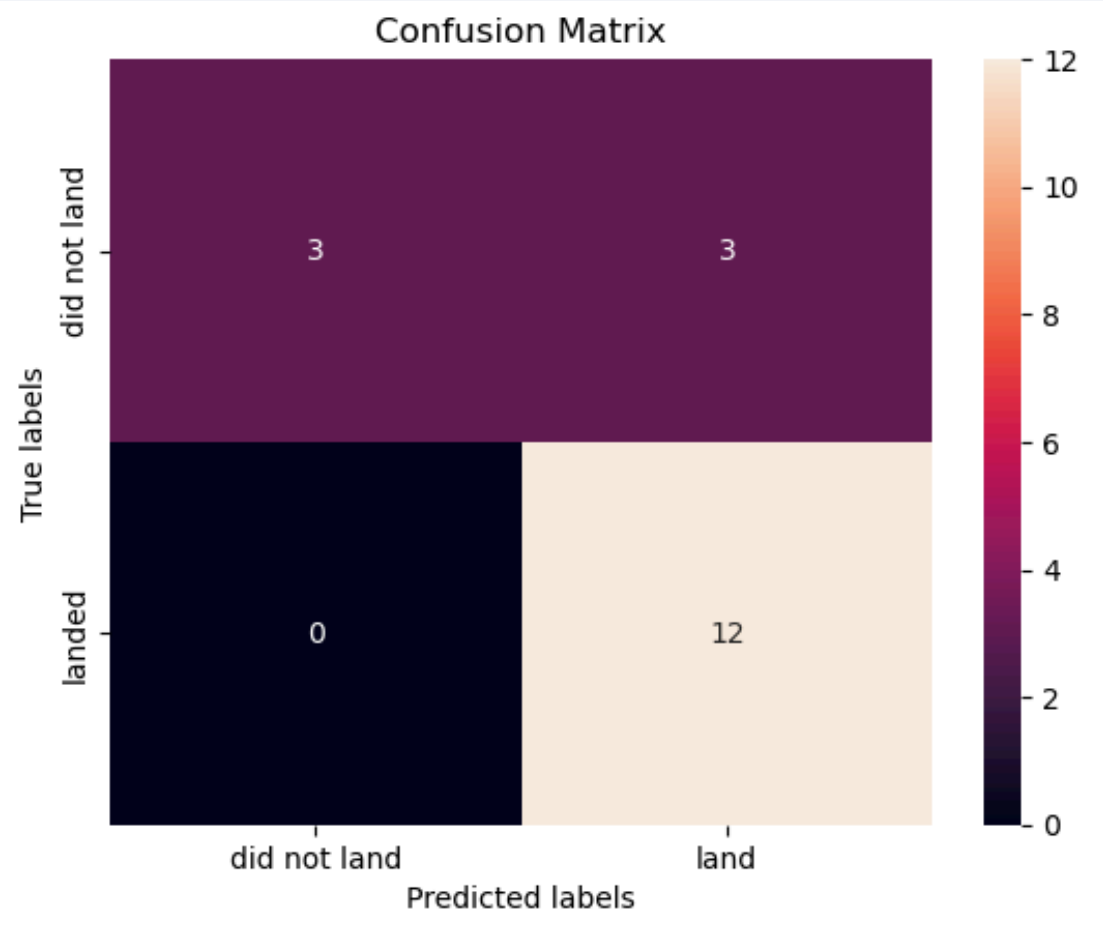
Classification Accuracy

- All models performed similarly, with an 83% testing accuracy.



Confusion Matrix

- All models also produced the same confusion matrix.
- The matrix indicates that classification is possible, the main issue being false positives.



Conclusions

- It appears possible to predict whether or not a Falcon 9 rocket's first stage can be reused using machine learning.
- However, the models should be trained and tested on a larger dataset if we would actually like to determine optimal performance.

Appendix

- This presentation was completed in satisfaction of the requirements for the Final Project from IBM's Applied Data Science Capstone course on Coursera.
- All project files, including Jupyter Notebooks, can be found at:
  - https://github.com/clarkti5/IBM-Applied-Data-Science-Capstone

Thank you!