IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joanna Ruszczyk
01.06.2024

Outline

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

Executive Summary

Summary of methodologies:

- **Data Collection using API and Web Scrapping**
- **Data Wrangling**
- **Dataset analysis with SQL**
- **EDA Data Visualization**
- **Launch Sites Locations Analysis with Folium**
- **Dashboard Application with Plotly Dash**
- **Machine Learning Prediction**

Summary of all results:

- **Exploratory Analysis**
- **Visualization Analysis - Dashboards**
- **Predictive Analysis - Classification**

Introduction

Companies are making space travel more accessible.

SpaceX stands out with achievements such as sending spacecraft to the ISS, launching the Starlink internet constellation, and conducting manned missions. SpaceX's cost-effective **Falcon 9** launches, priced at $62 million due to reusable first stages, are much cheaper than other providers' $165 million launches.

This presentation shows:

- the analysis of SpaceX data and
- prediction if the Falcon 9’s first stage will be reused
- estimatimation of the cost of launches

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Using SpaceX REST API and web scraping
- Perform data wrangling
  - Filtering and cleaning the data
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning and evaluating the models

Data Collection

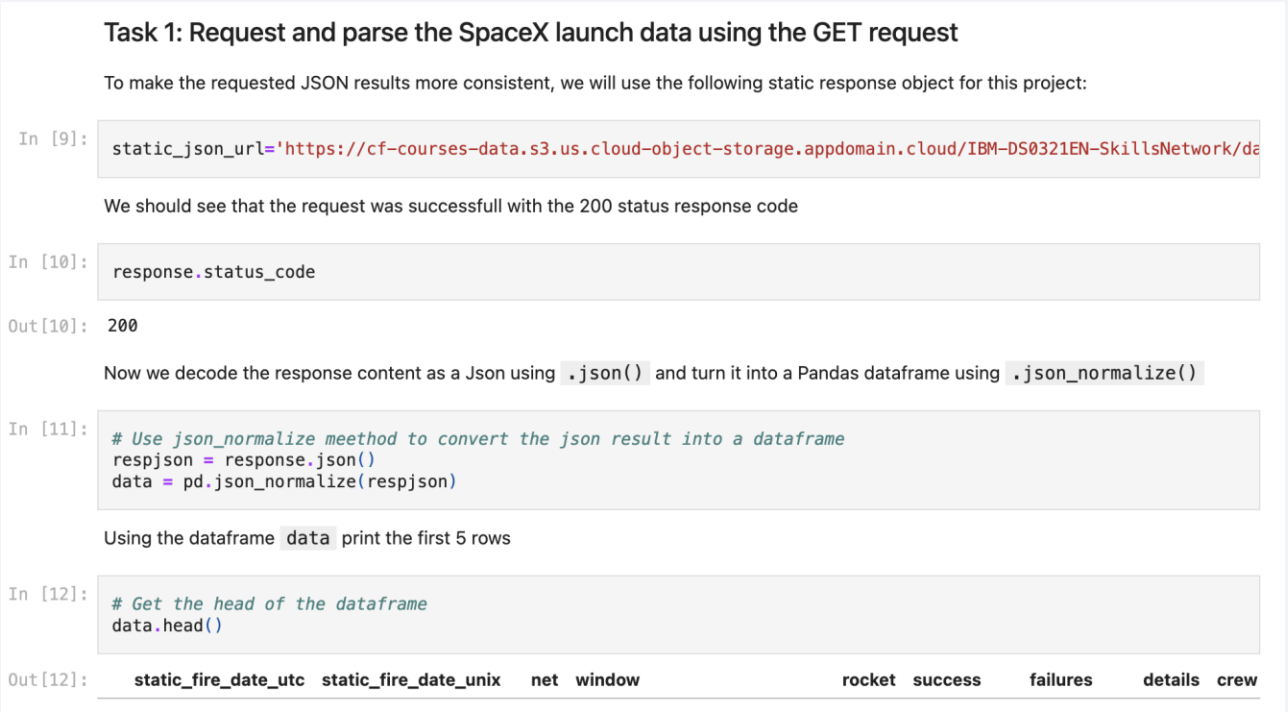
- Request and parse the SpaceX launch data using the GET request
- Using the API to get information about the launches using the IDs given for each launch: rocket, payloads, launchpad, and cores
- Filter the dataframe to only include Falcon 9 launches
- Dealing with missing values by replacing them
- Export data to CSV. file

Data Collection – SpaceX API

- Data collection with SpaceX REST

GitHub URL:

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/1.%20Spacex-data-collection-api.ipynb



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:

In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/da
```

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

In [10]:
```
response.status_code
```

Out[10]: 200

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

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

Using the dataframe `data` print the first 5 rows

In [12]:
```
# Get the head of the dataframe
data.head()
```

Out[12]:

static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew

Data Collection - Scraping

Web scraping process with BeautifulSoup

- Extraction of a Falcon 9 launch records HTML table from Wikipedia
- Parsing the table and converting it into a Pandas data frame

GitHub URL:

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/2.%20SpaceX%20webscraping.ipynb

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 [6]:

```
# use requests.get() method with the provided static_url
# assign the response to a object

import requests

# URL of the Falcon 9 Launch HTML page
url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"

# Perform the GET request
response = requests.get("https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&ol
```

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

In [7]:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

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

In [8]:

```
# Use soup.title attribute
title = soup.title
print("Title of the page:", title.text)
```

```
Title of the page: List of Falcon 9 and Falcon Heavy launches - Wikipedia
```

Data Wrangling

- Performing Exploratory Data Analysis (EDA)
- Converting outcomes of landing into Training Labels
- Calculations of successful landing
- Determination of the the success rate

- GitHub URL

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/3.%20SpaceX-Data%20wrangling.ipynb

EDA with Data Visualization

- Performing exploratory Data Analysis and Feature Engineering using Pandas and Matplotlib:
  - FlightNumber vs LaunchSite
  - Payload Vs. Launch Site
  - FlightNumber vs. Orbit type
  - Payload vs. Orbit
- In order to obtain some preliminary insights about how each important variable would affect the success rate

GitHub URL:

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/5.%20SpaceX%20EDA%20Data%20Visualization.ipynb

EDA with SQL

- Displaying the names of the unique launch sites in the space mission

%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;

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

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

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

%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';

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

%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';

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

%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";

GitHub URL

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/4.%20SpaceX-eda-sql_sqllite.ipynb

EDA with SQL part 2

- 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 DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

- List the total number of successful and failure mission outcomes

%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";

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

%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);

- 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.

%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';

- 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.

%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;

GitHub URL
https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/4.%20SpaceX-eda-sql_sqllite.ipynb

Build an Interactive Map with Folium

- Creation of folium map with launch sites and map objects such as markers, circles, lines, etc. In order to mark the success or failure of the launch site.
  - Green – successful
  - Red - failure

* Marker clusters help to simplify a map containing many markers having the same coordinate.

GitHub URL

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/6.%20SpaceX%20launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Launch Site drop-down input component
- Callback function to render success-pie-chart based on selected site drop/down
- Ranger Slider to select payload
- Callback function to render the success-payload-scatter-char scatter plot

GitHub URL

https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/7.%20SpaceX_dash_app.py

Predictive Analysis (Classification)

- Loading the dataframe
- Creating a NumPy array
- Standardizing the data
- Using the function train_test_split to split the data X and Y into training and test data
- Creating a decision support vector machine object and tree classifier object and Create a k nearest neighbors object
- Calculating the accuracy on the test data
- Finding which method performs best

GitHub URL

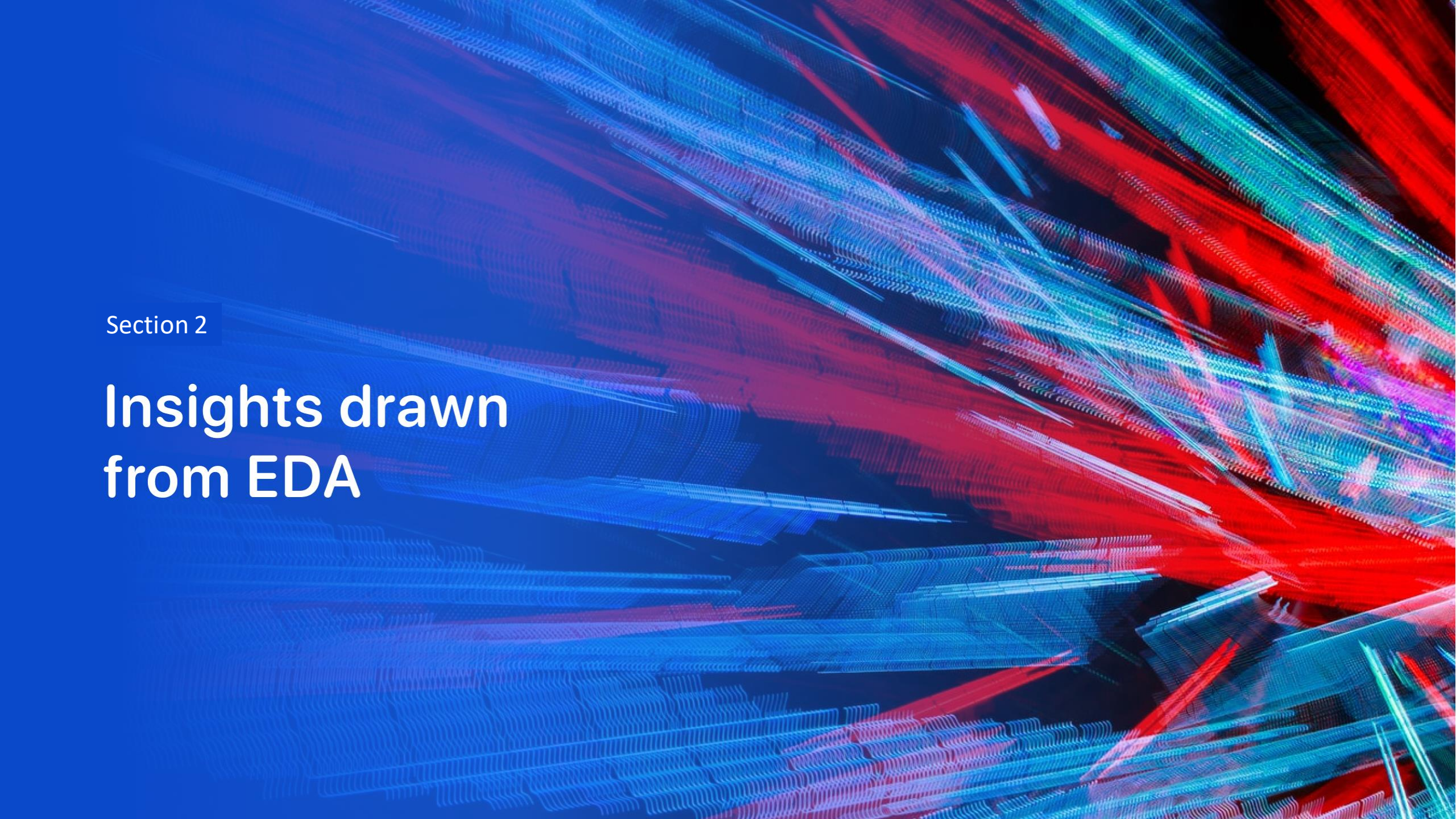
https://github.com/YoaRiz/SpaceX-Falcon-9-first-stage-Landing-Prediction/blob/399eba7631afc6a4cc589914d035667f2fb2c301/8.%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

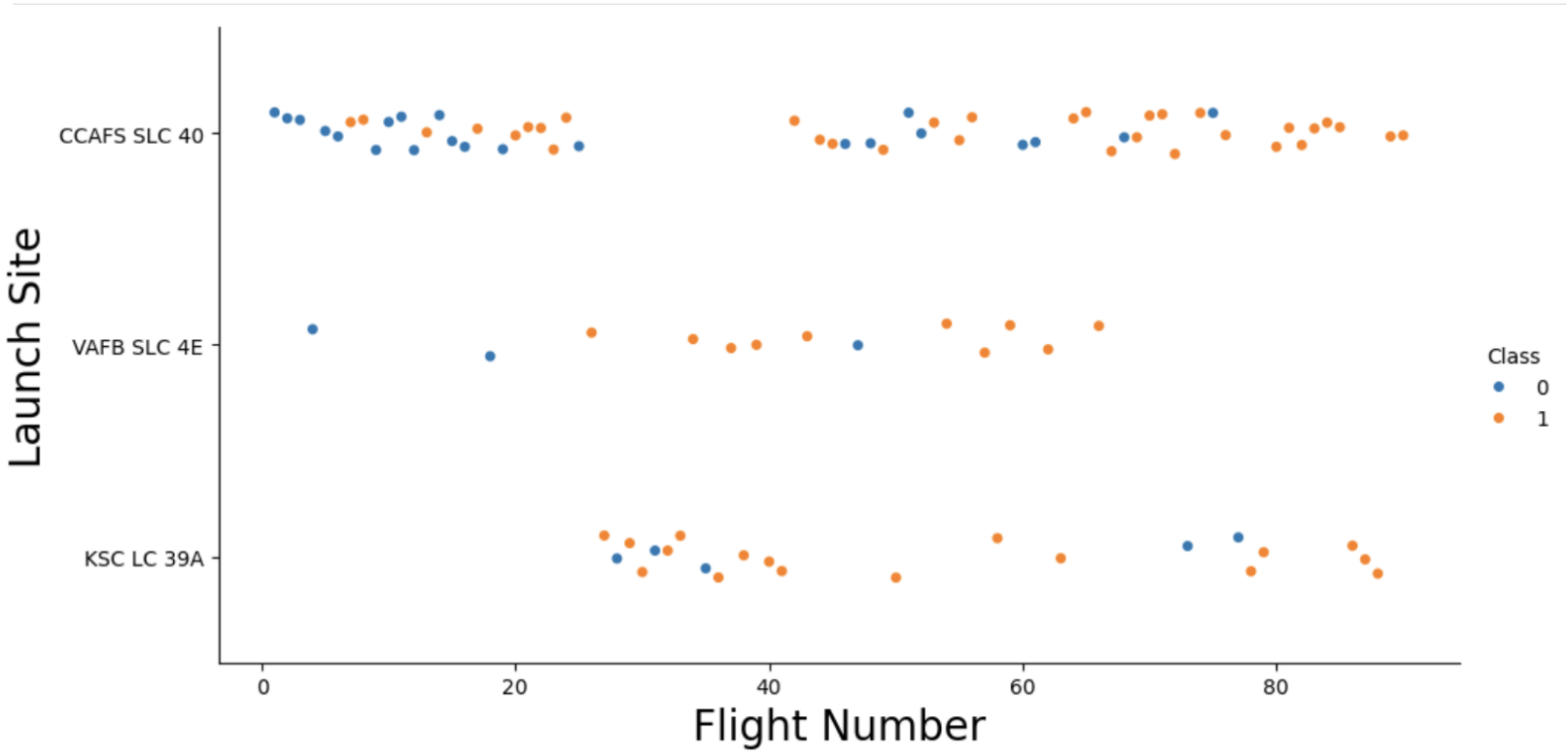
Section 2

Insights drawn from EDA



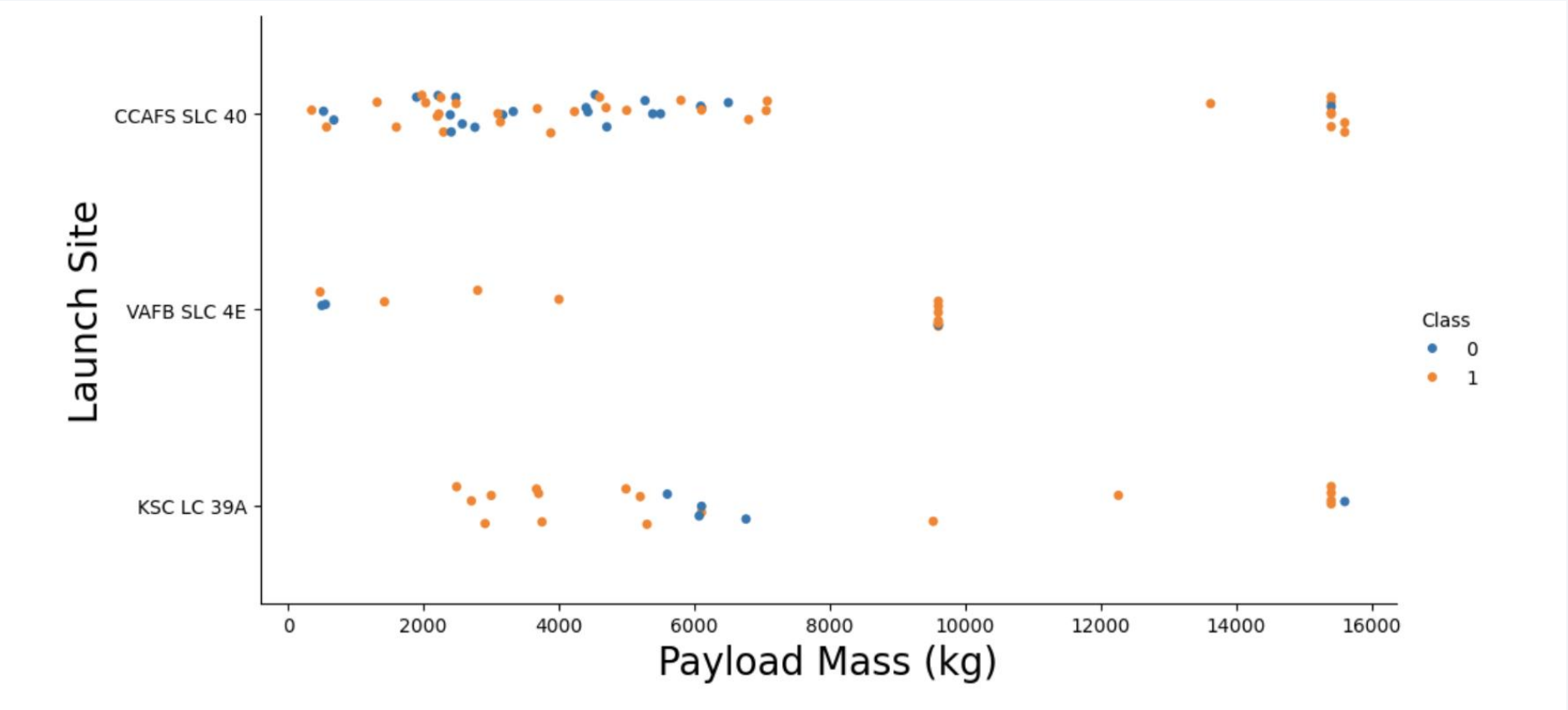
Flight Number vs. Launch Site

Explanation: The later the flights the higher the success rate. The new launches have a higher success rate



Payload vs. Launch Site

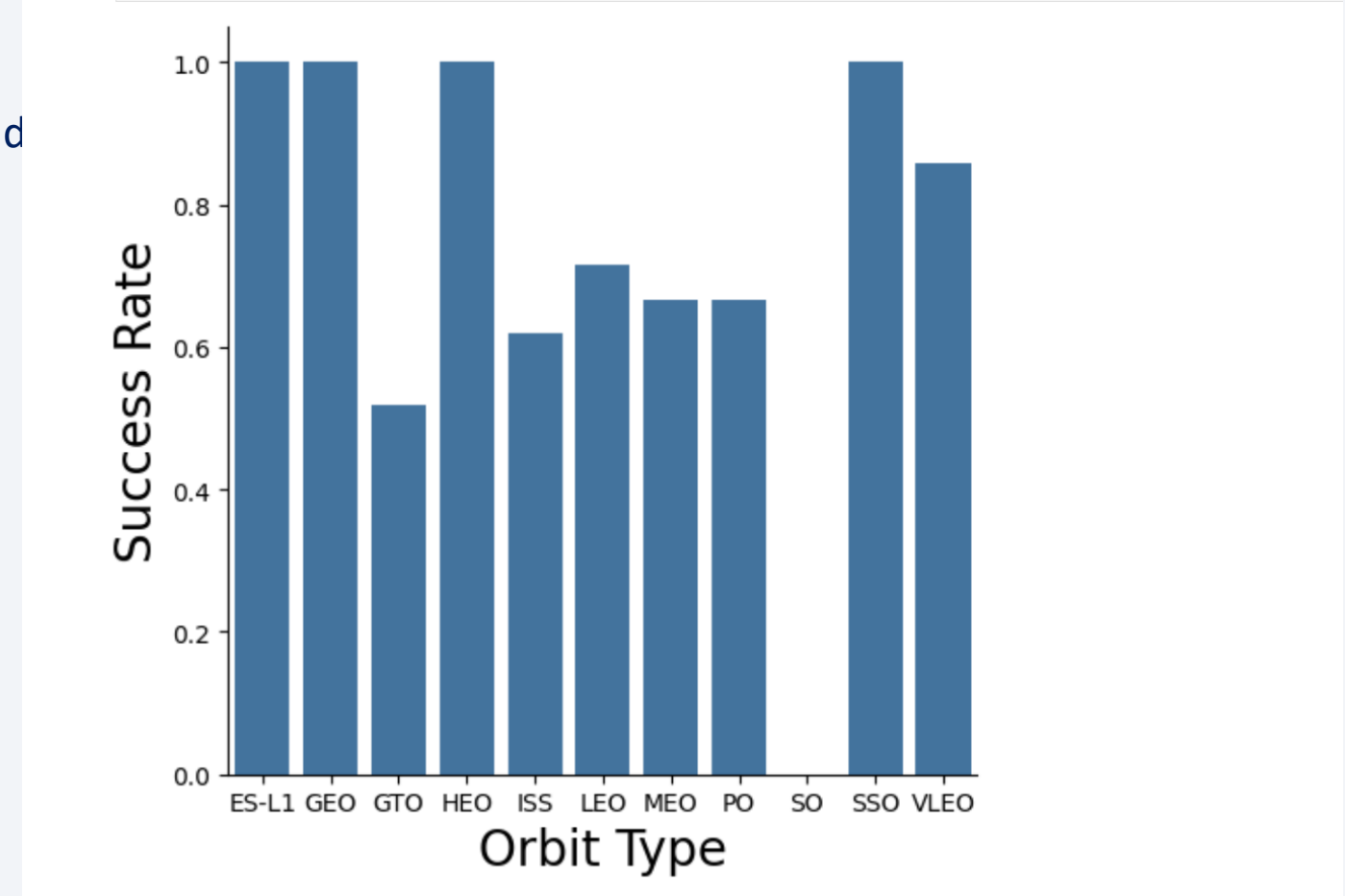
- Explanation: the higher the payload mass the higher the success



Success Rate vs. Orbit Type

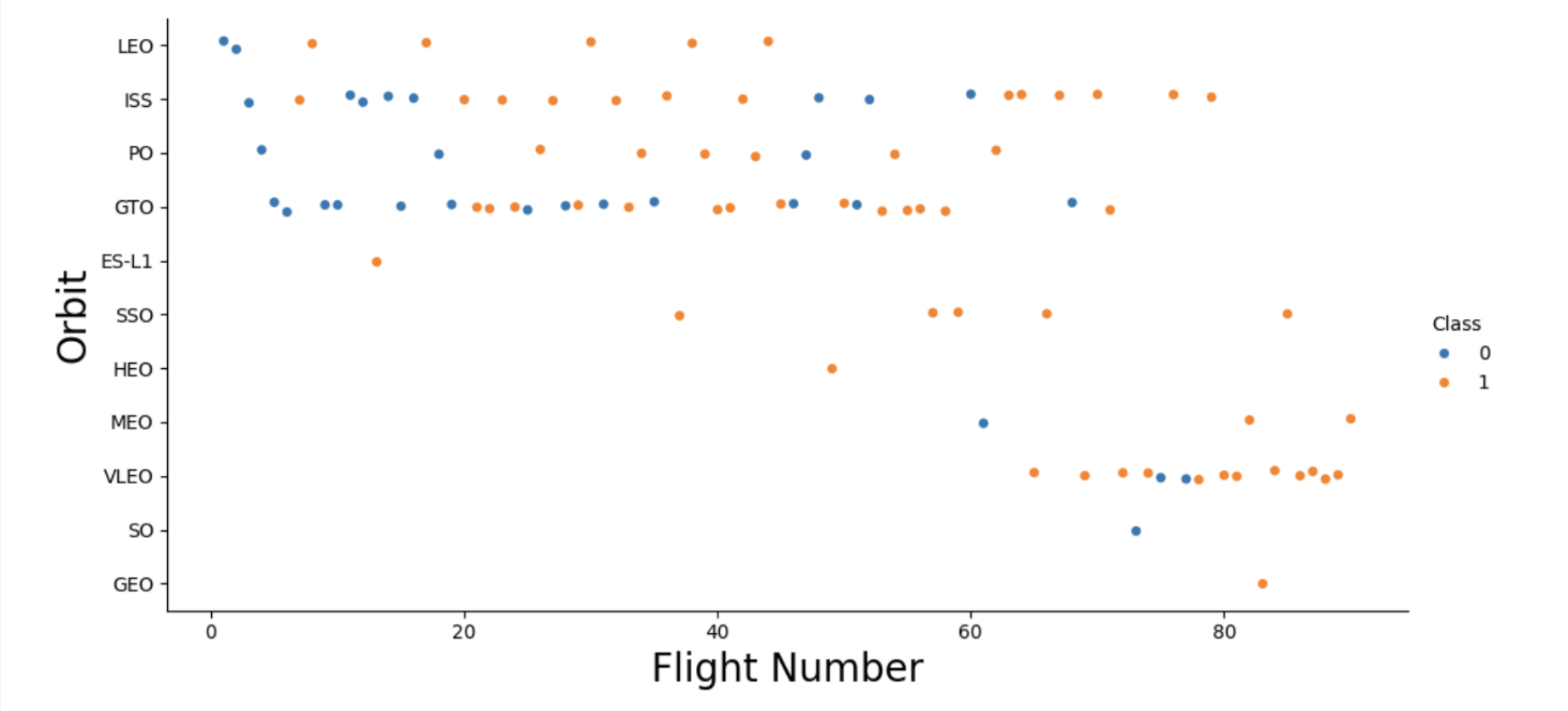
Explanation: Orbits ES-L1, GEO, HEO and SSO have the highest success rate at 100%.

Orbit SO has 0% of success rate.



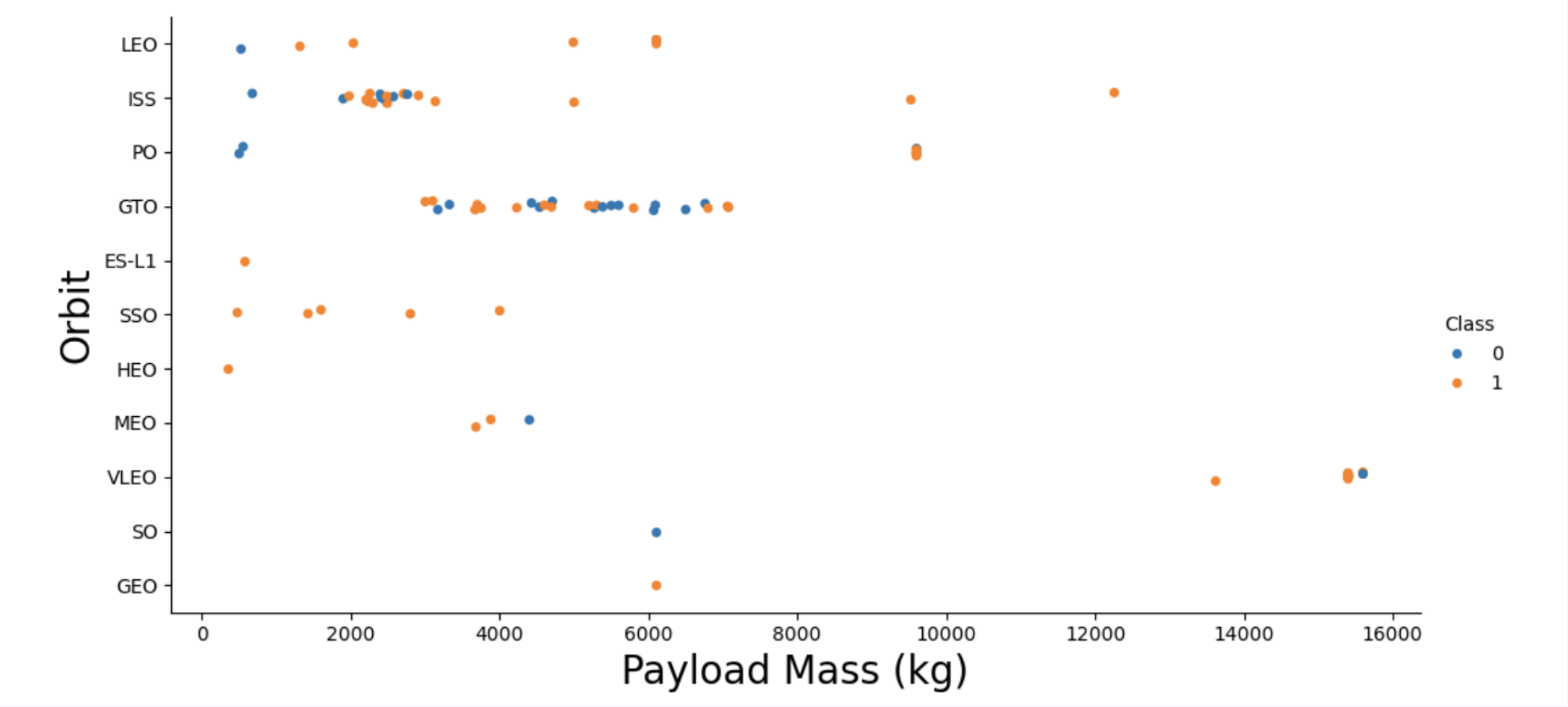
Flight Number vs. Orbit Type

Explanation: The success rate increases with the number of flights for every orbit. The GTO orbit doesn't follow this trend.



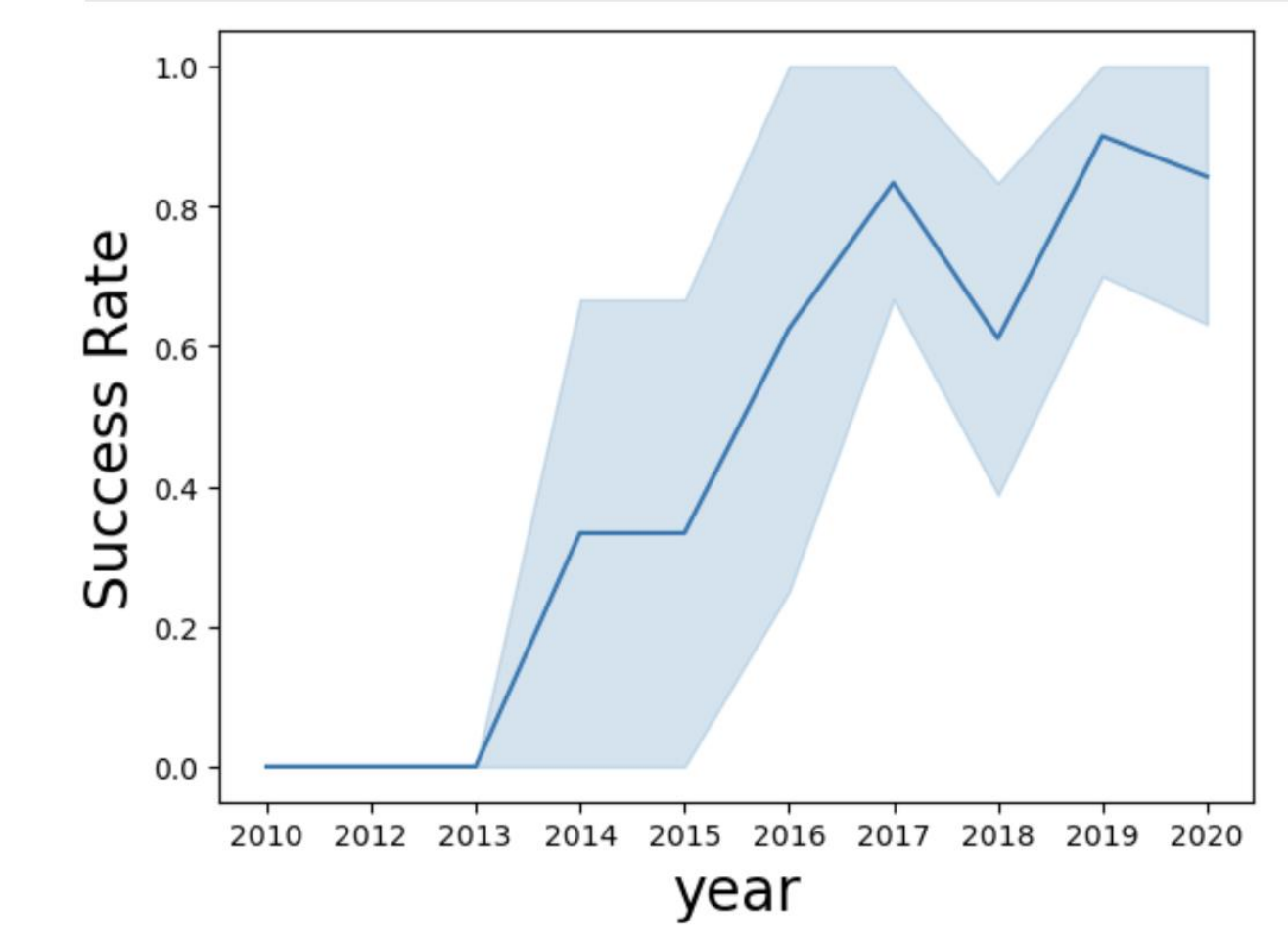
Payload vs. Orbit Type

Explanation: Heavy payloads perform better with the orbits: LEO, ISS and PO. The GTO orbit has successes as well as failures independently from the payload.



Launch Success Yearly Trend

We can see that the success is increasing since 2013, with a decrease in the years 2017-2018 and after 2019.



All Launch Site Names

The SQL query and result of unique Launch Site Names:

Task 1

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

In [26]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[26]:

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

Launch Site Names Begin with 'CCA'

5 records where launch sites begin with `CCA`

```
%sql SELECT * FROM 'SPACEXTBL' 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_
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (p
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 (p
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	N
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	N
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	N

Total Payload Mass

%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';

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

In [11]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA
```

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

Out[11]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';

```
In [12]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booste
 * sqlite:///my_data1.db
Done.
```

Out[12]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";

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

Hint:Use min function

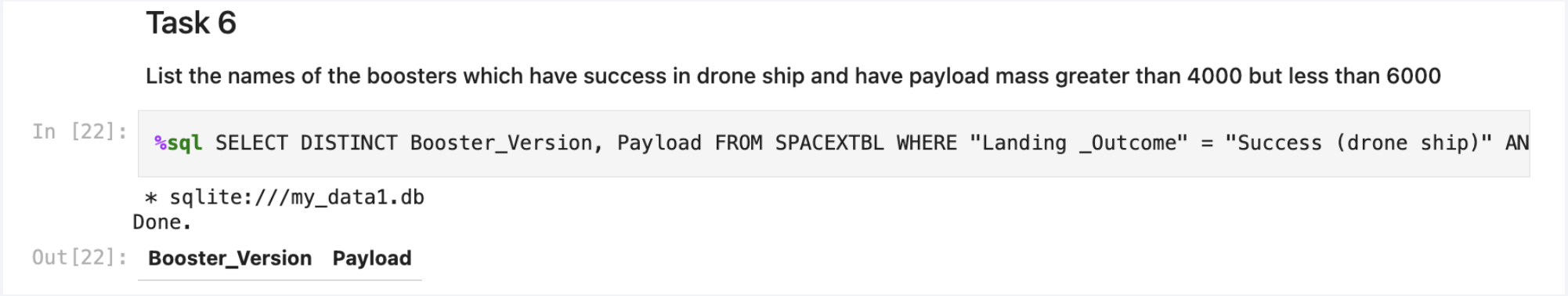
```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

MIN(DATE)
None

Successful Drone Ship Landing with Payload between 4000 and 6000

%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;



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 [22]:
```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AN
```

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

Out[22]:

Booster_Version	Payload

Total Number of Successful and Failure Mission Outcomes

%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";

Task 7

List the total number of successful and failure mission outcomes

In [15]:
```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

Out[15]:

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

Boosters Carried Maximum Payload

%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);

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

In [16]:
```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX
```

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

Out[16]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

2015 Launch Records

%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';

```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_",
```

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

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	"Landing _Outcome"

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

%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2017') ORDER BY Date DESC;

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.

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing _Outcome" LIKE 'Success%' AND (Date BETWEEN '04-06-2010' AND '20-03-2
```

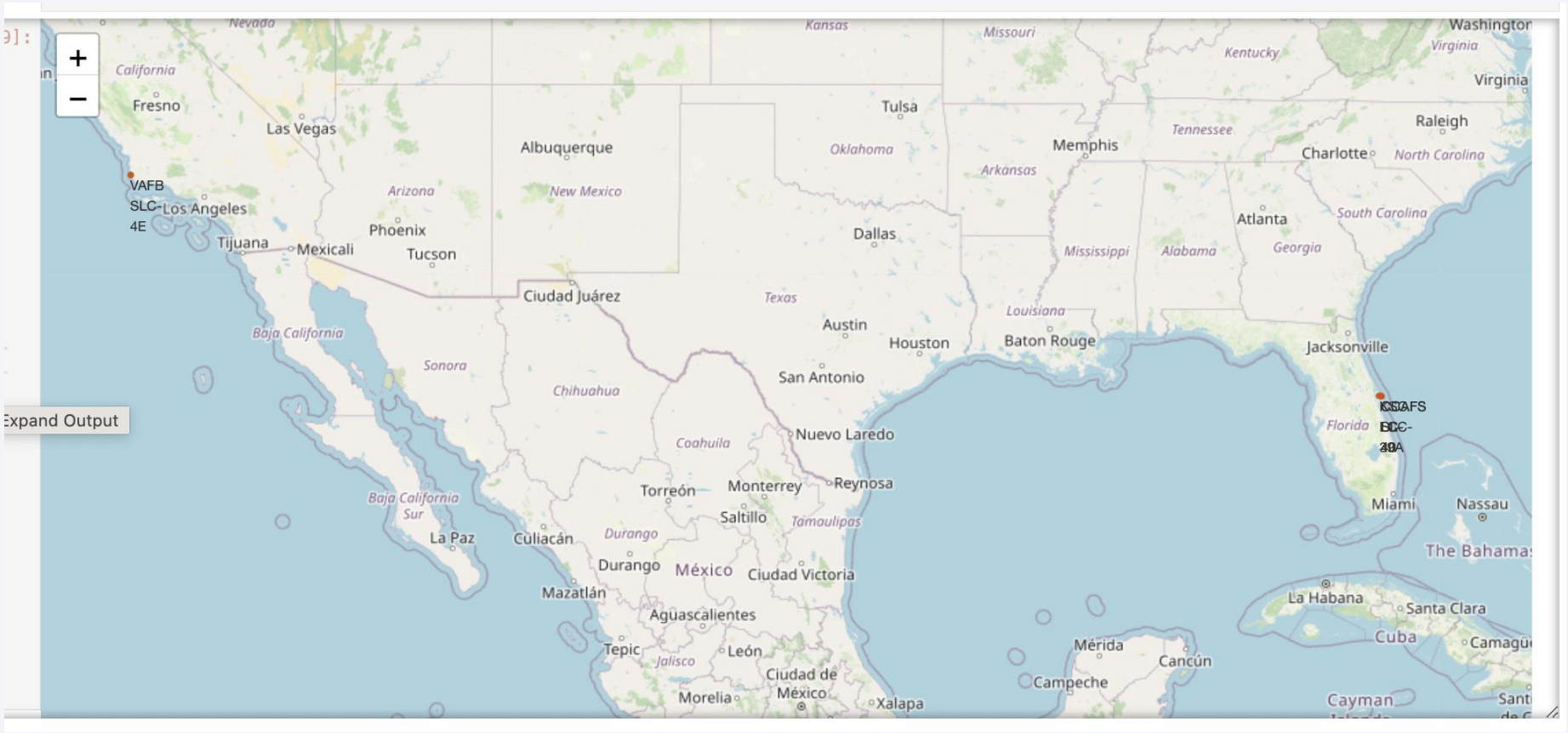
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcon


Section 3
Launch Sites
Proximities Analysis

Launch sites

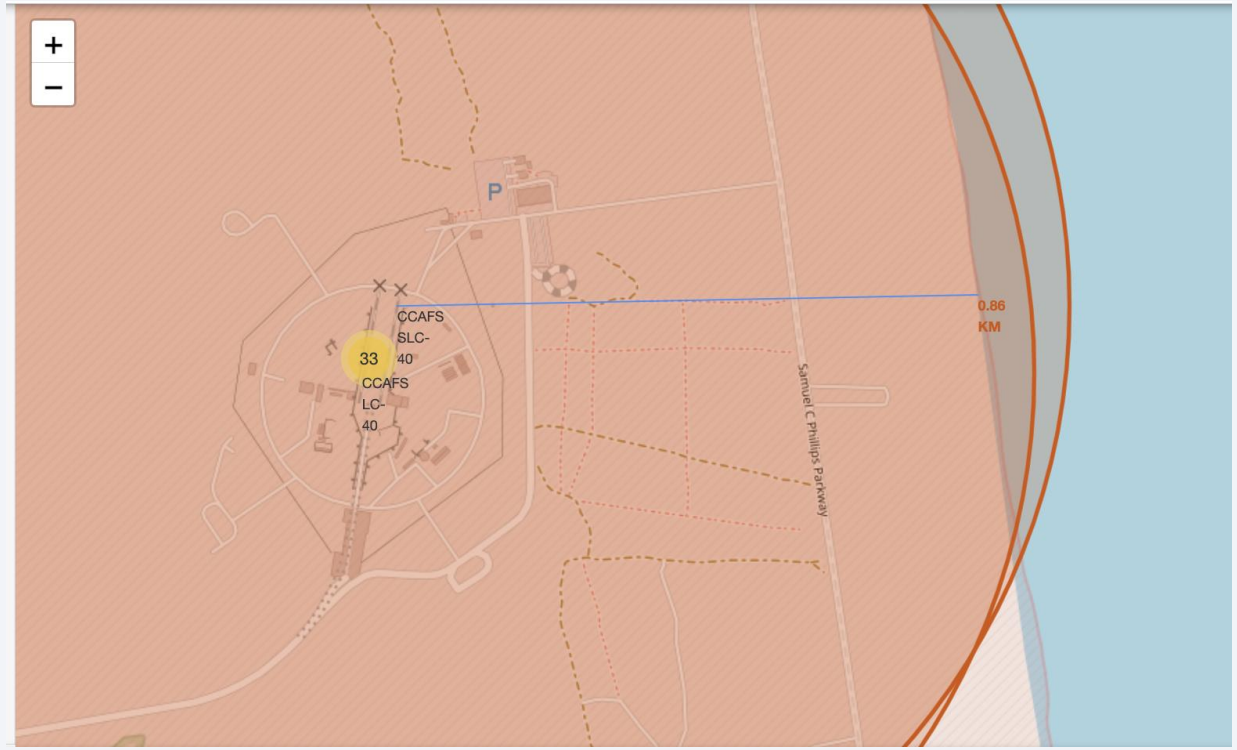
All launch sites are near the Equator, what gives the rockets a natural boost and saves the fuel cost.



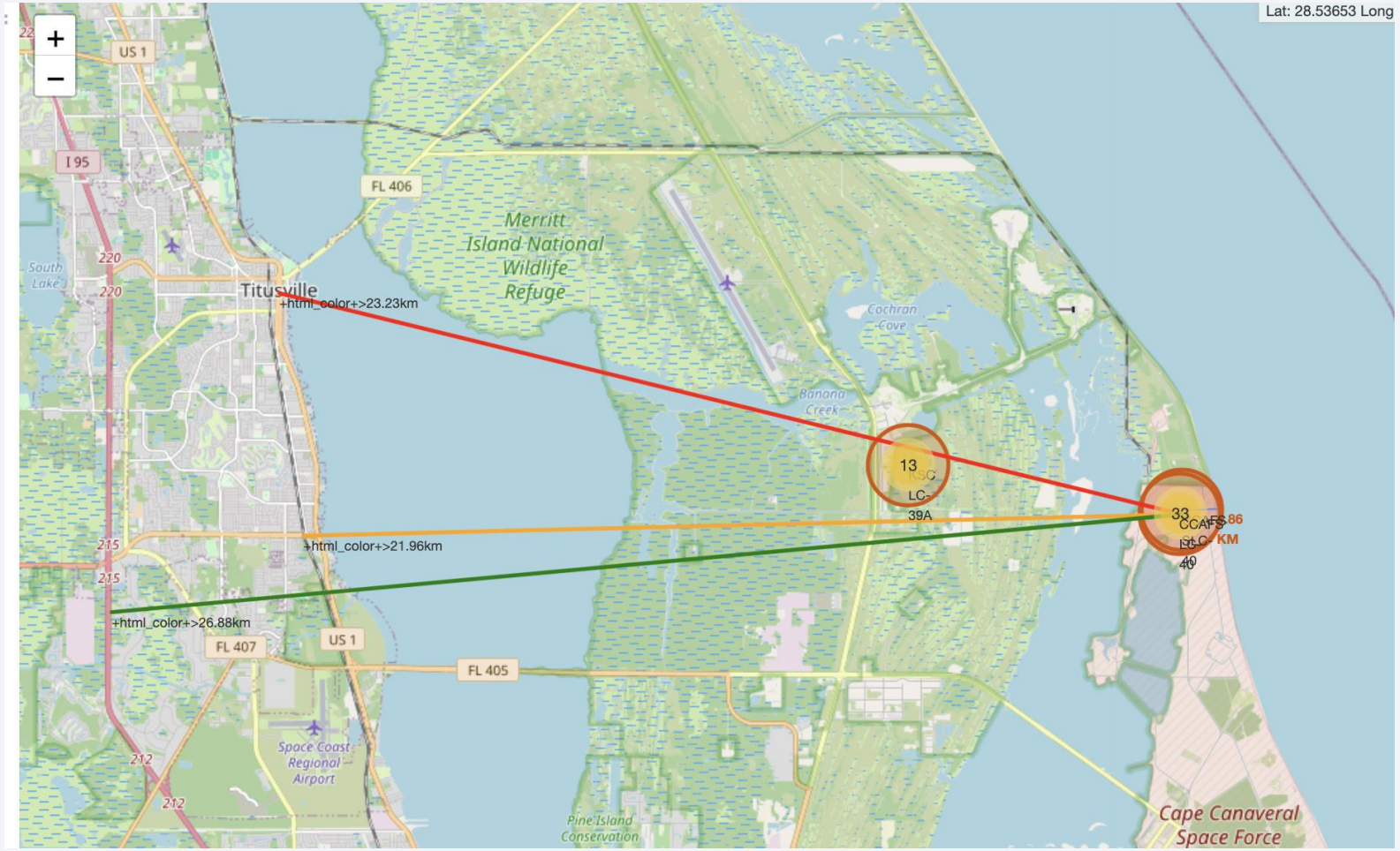
Launch site Outcomes

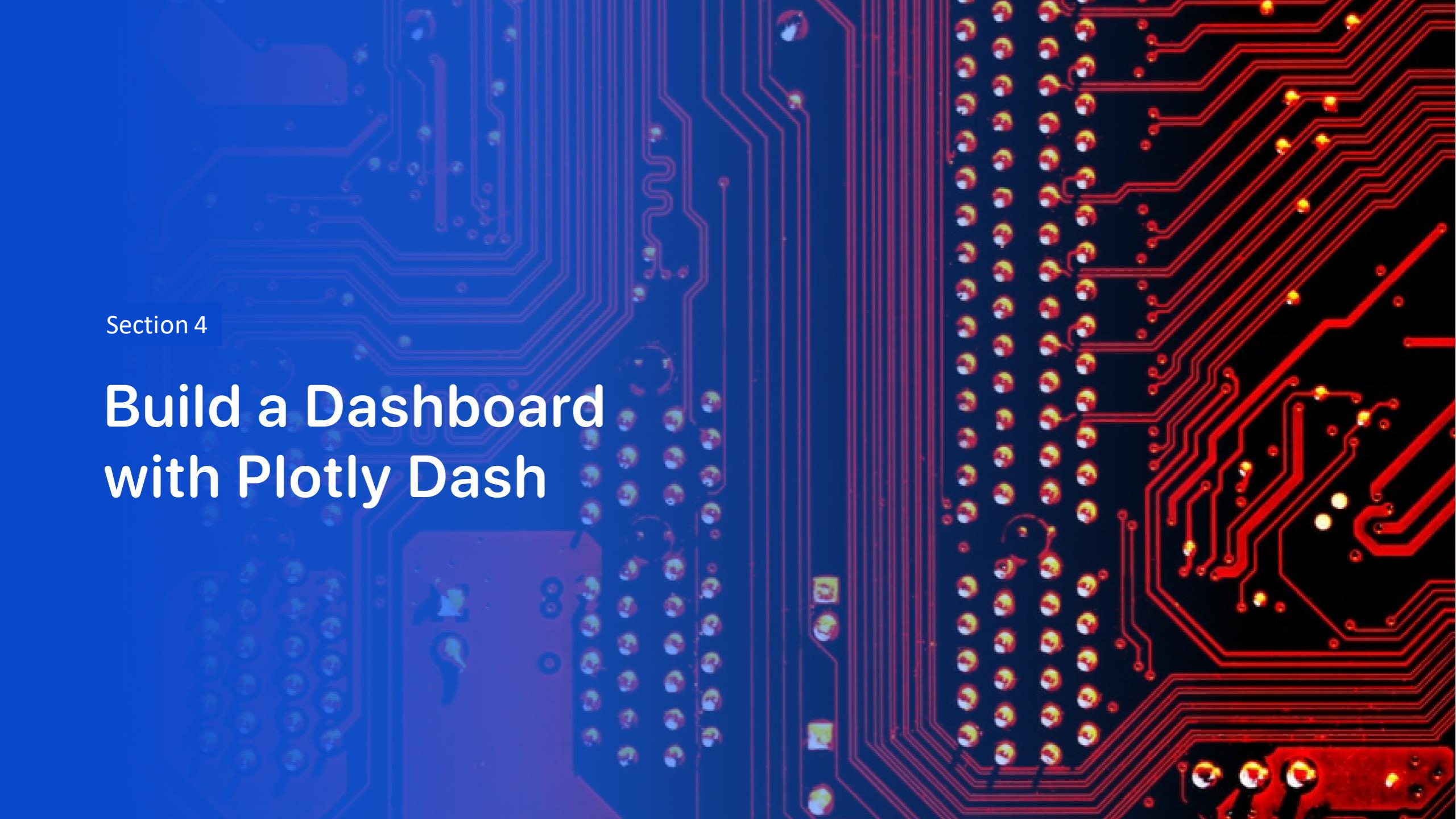
Green markers show the success and red markers the failed launches.

Distance to the coastline



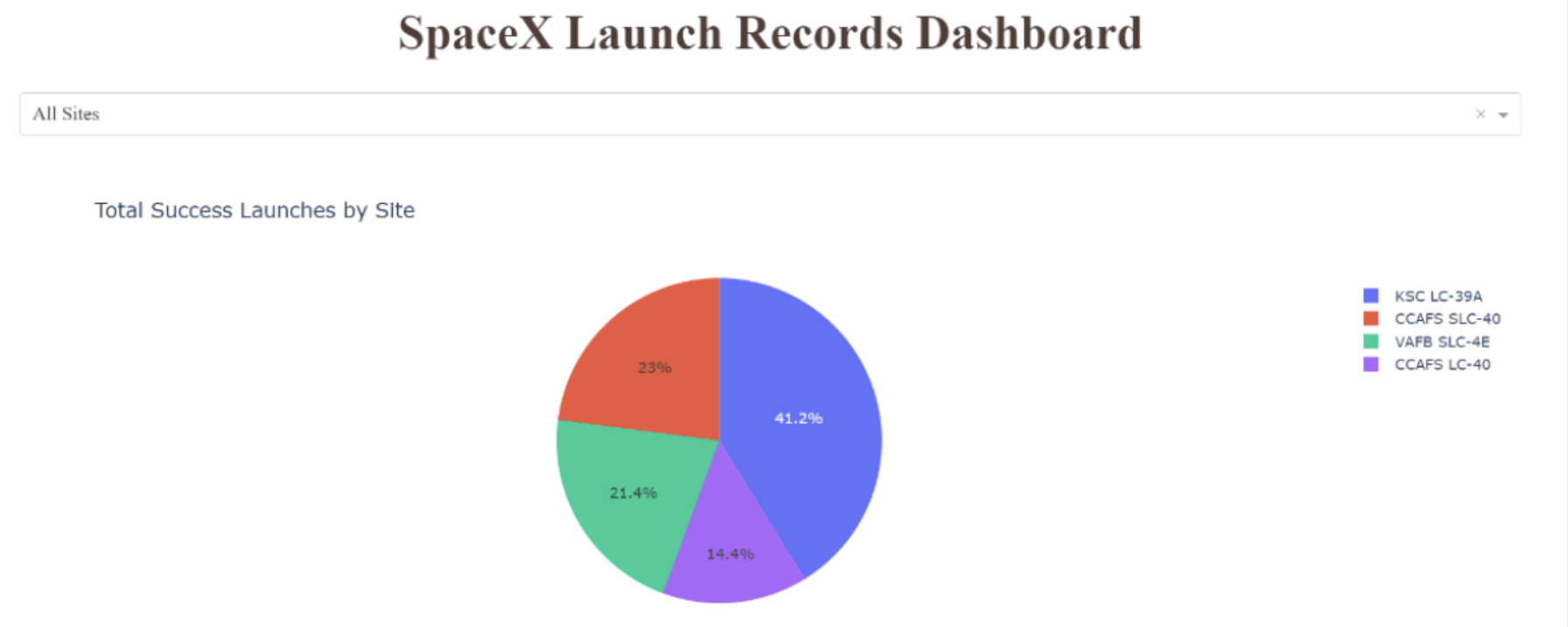
Distance to the proximities




Section 4
Build a Dashboard with Plotly Dash

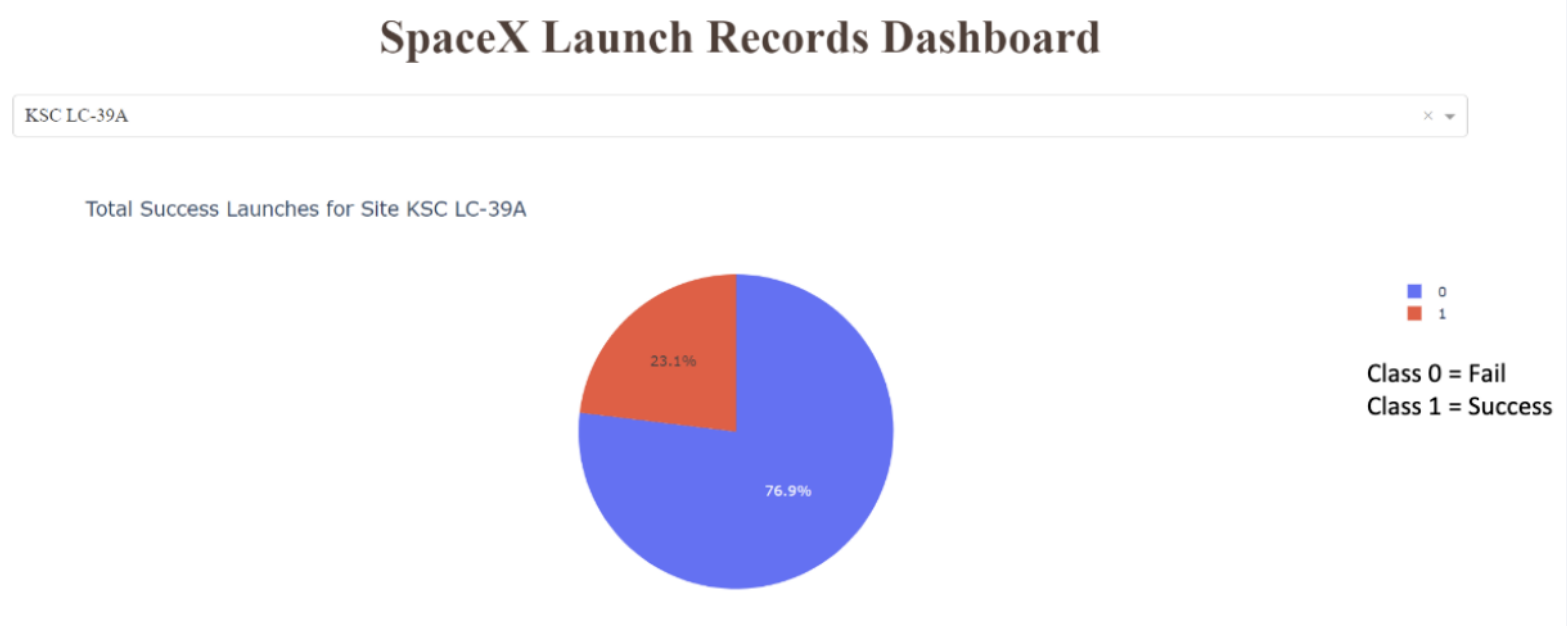
Total success launches by site in %

KSC LC-39A is the most successful launch site with 41.2%. CCAFS SLC-40 has the lowest success rate at 14.4%



Launch site KSC LC-39A

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



Success count on Payload Mass

Payloads between 2000kg and 5000kg are the most successful.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

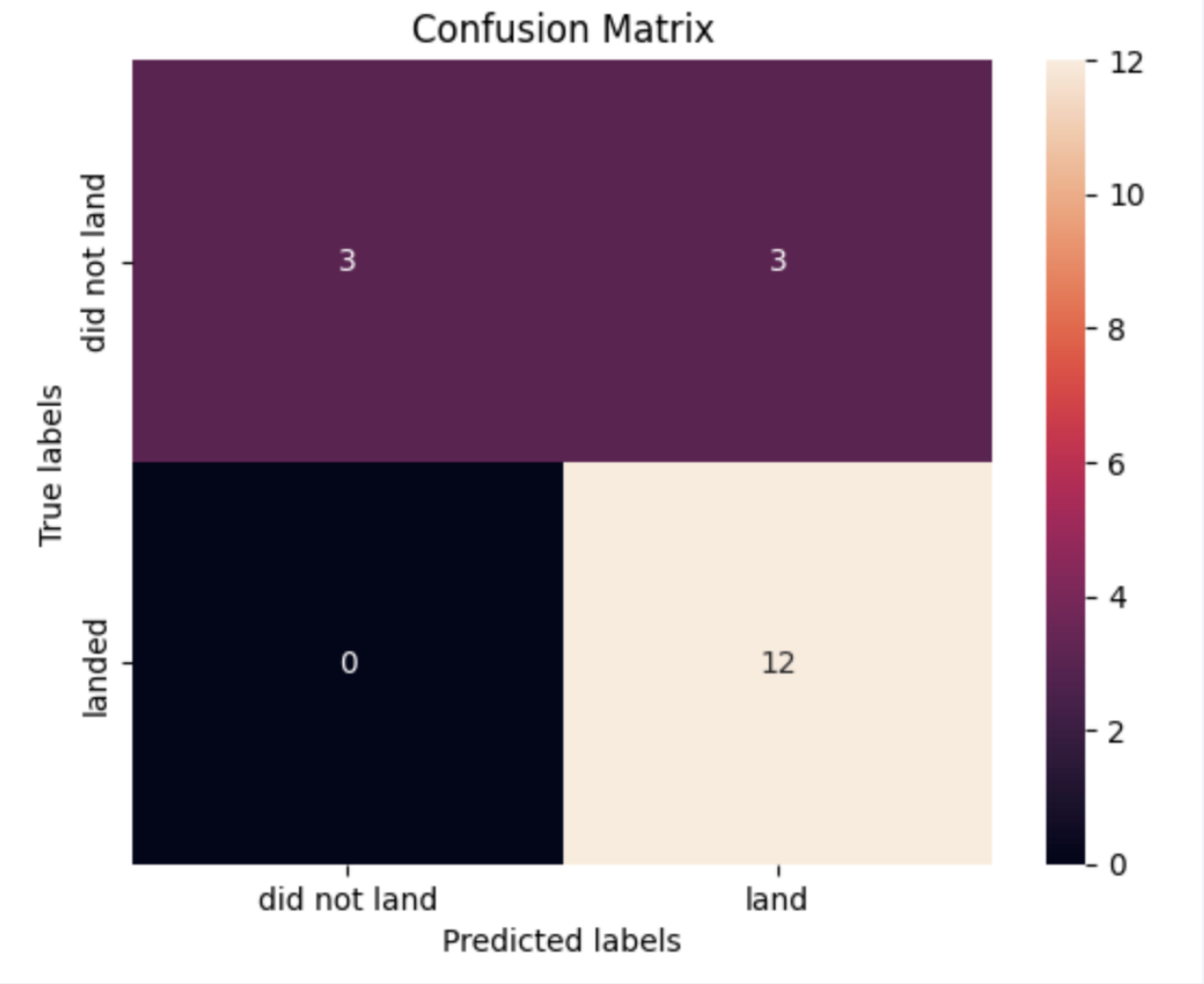
All models have the same accuracy 0.83. The small data amount could be the reason why all models performed at the same level

```
Report = pd.DataFrame({'Method' : ['Test Data Accuracy']})

knn_accuracy=knn_cv.score(X_test, Y_test)
Decision_tree_accuracy=tree_cv.score(X_test, Y_test)
SVM_accuracy=svm_cv.score(X_test, Y_test)
Logistic_Regression=logreg_cv.score(X_test, Y_test)

Report['Logistic_Reg'] = [Logistic_Regression]
Report['SVM'] = [SVM_accuracy]
Report['Decision Tree'] = [Decision_tree_accuracy]
Report['KNN'] = [knn_accuracy]

Report.transpose()
```

	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

Confusion Matrix

All the confusion matrixes were identical.



Conclusions

- Most of the launch sites are located near the equator. It gives an additional natural boos and saves fuel costs.
- The higher flight number, the higher the success rate
- The most successful launch site was KSC LC-39A
- ES-L1, GEO, HEO and SSO orbits had a 100% success rate
- The higher the payload mass, the higher the success rate.
- The success rate increases since 2013 till 2020
- Payloads between 2000kg and 5000kg are the most successful.
- All models performed very similar having the same accuracy rate.

Thank you!