IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rakibul Islam

03 September 2023

Outline

Executive Summary

In this project, we will predict whether a SpaceX Falcon 9 first stage will land successfully.

By predicting if the first stage will land successfully, we can estimate the cost of a launch. This will be achieved using Python and with the help of Machine Learning techniques. The following methodology steps are taken during the project:

1. Data Collection
2. Data Wrangling and Preprocessing
3. Exploratory Data Analysis
4. Data Visualization
5. Machine Learning Prediction

During the analytical process, it indicates that there are some features of rocket launches that have correlation with the success or failure of launches.

Introduction

The primary goal of this capstone project is to predict whether a SpaceX Falcon 9 first stage will land successfully. SpaceX prides itself in being able to reuse the first stage of a rocket launch so much so that they advertise on their website that their rocket launches cost USD $62 million while other provides cost upward USD $165 million. Much of these savings are down to the first stage's reusability.

If we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

This brings us to our main question that we are trying to answer :

For a given set of features about a Falcon 9 rocket launch, will the first stage of the rocket land successfully?

Source: Wikipedia

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology

  Data was collected through two methods: requesting data from the SpaceX API and web scraping launch data from a Wikipedia page.

- Perform data wrangling

  Data wrangling was then performed to transform and clean the data using Python's Pandas library.

- Perform exploratory data analysis (EDA) using visualization and SQL

  With the clean data, exploratory data analysis (EDA) was performed using visualization tools such as Python's matplotlib and seaborn libraries, as well as answering questions using SQL queries. Python's interactive visualization packages were used to answer some analytical questions.

- Perform interactive visual analytics using Folium and Plotly Dash

  Folium was used for creating maps while Plotly Dash was used to create interactive data visualizations.

- Perform predictive analysis using classification models

  Four different machine learning classification models were used for the predictive analysis. The models that were used are logistic regression, support vector machines, k-nearest neighbor and decision tree classifier. Each model was trained, tuned and evaluated to find the best one.

Data Collection

The data was collected using below methods:

- Data collection was done using get request to the SpaceX API . Then, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize()
- We then cleaned the data, checked for missing values and fill in missing values where necessary.
- In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with the help of BeautifulSoup.
- The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for the future analysis.

Data Collection – SpaceX API

Below major steps taken during data collection:

- Request and parse the SpaceX launch data using the GET request
- Normalize JSON response into a dataframe
- Extract only useful columns using auxiliary functions
- Create new pandas dataframe from dictionary
- Filter dataframe to only include Falcon 9 launches
- Handle missing values

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/1.%20IBM%20Data%20Science_SpaceX-Data-Collection-API_Rakib.ipynb

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

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

        Check the content of the response

In [8]: print(response.content)

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":

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

        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 .jso

In [11]: # 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

In [13]: # Get the head of the dataframe
         data.head()

Out[13]:
```

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures

Data Collection - Scraping

Below major steps taken during webscraping:

- Request rocket launch data from Wikipedia page
- Applied webscraping techniques with BeautifulSoup
- Parsed the table and converted into a pandas dataframe.

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/2.%20IBM%20Data%20Science_SpaceX_Webscraping_Rakib.ipynb

To keep the lab tasks consistent, you will be asked to scrape the data from a snapshot of the `List of Falcon 9 and Falcon Heavy launches` Wikipage updated on `9th June 2021`

In [4]:
```
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.

In [5]:
```
# use requests.get() method with the provided static_url
# assign the response to a object

data  = requests.get(static_url).text
```

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

In [6]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(data)
```

In [16]:
```
df=pd.DataFrame(launch_dict)
df.head()
```

Out[16]:

	Flight No.	Launch site	Payload	Payload mass	Orbit	Customer	Launch outcome	Version Booster	Booster landing
0	1	CCAFS	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success\n	F9 v1.0B0003.1	Failure
1	2	CCAFS	Dragon	0	LEO	NASA (COTS)\nNRO	Success	F9 v1.0B0004.1	Failure

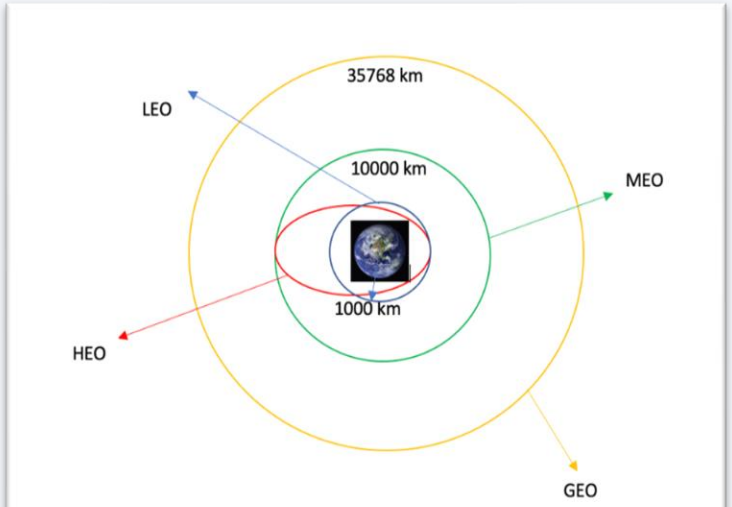
Data Wrangling

Below major steps taken during Data Wrangling:

- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column using one-hot encoding

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/3.%20IBM%20Data%20Science_SpaceX%20Data_Wrangling_Rakib.ipynb




```
In [8]: # Apply value_counts on Orbit column
        df['Orbit'].value_counts()

Out[8]: GTO     27
        ISS     21
        VLEO    14
        PO       9
        LEO      7
        SSO      5
        MEO      3
        ES-L1    1
        HEO      1
        SO       1
        GEO      1
        Name: Orbit, dtype: int64
```

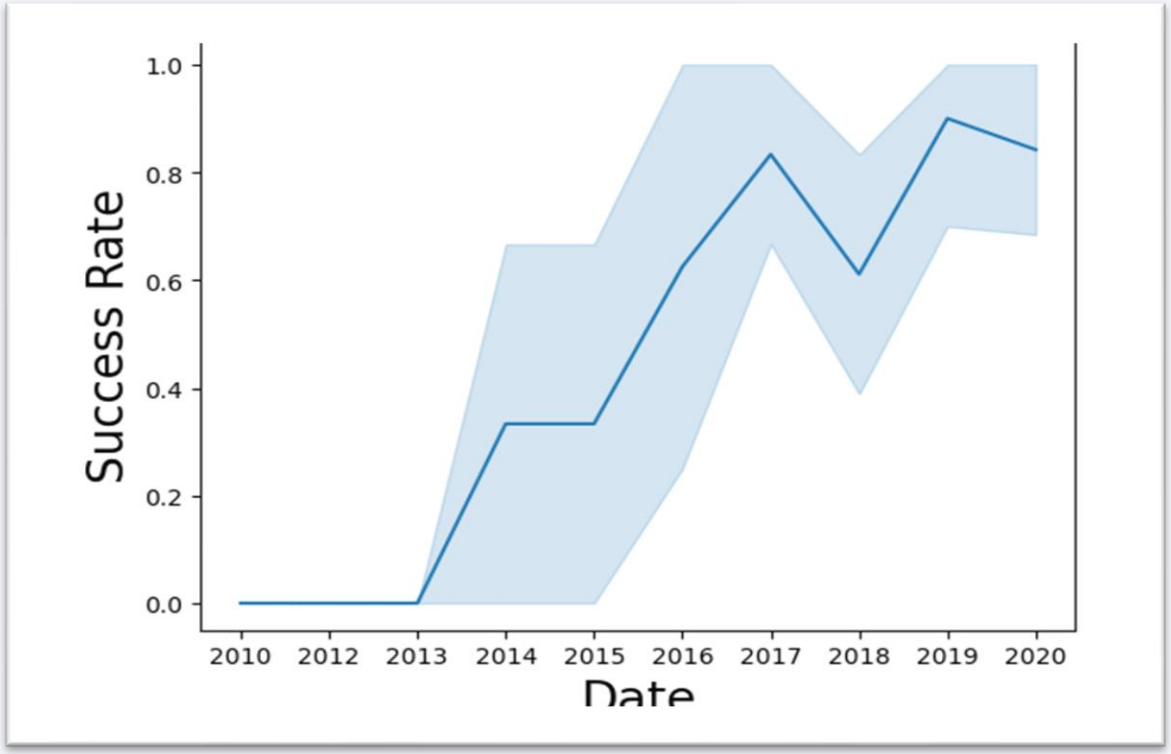
EDA with Data Visualization

Below major steps taken during Data Visualization:

- Scatter plot, Bar chart and Line charts are used in this stage
- The data are explored by visualizing the relationship between flight number and launch site, payload and launch site, success rate of each orbit type, flight number and orbit type. Also analyzed the launch success yearly trend.

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/5.%20IBM%20Data%20Science_SpaceX%20EDA-Dataviz_Rakib.ipynb



EDA with SQL

Below steps were taken during this SQL stage:

- Loaded SpaceX dataset into a PostgreSQL database while still using Jupyter notebook.
- Applied EDA with SQL queries to get following insights from the dataset:
  - Find names of the unique launch sites in the space mission
  - Total payload mass carried by boosters launched by NASA (CRS)
  - The date when the first successful landing outcome in ground pad was achieved
  - Total number of successful and failure mission outcomes
  - The names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - The names of the booster versions which have carried the maximum payload mass
  - 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

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/4.%20IBM%20Data%20Science_SpaceX%20EDA-SQL_Rakib.ipynb

Build an Interactive Map with Folium

Below steps were taken during this Interactive Map with Folium stage:

- Objects (markers, circles, lines) were created and added to a Folium map. Marker clusters were used to show all launch sites on a map as well as the successful/failed launches for each site on the map. Line objects were used to calculate the distances between a launch site to its proximities.
- By adding these objects, following geographical patterns about launch sites are found:
  - Are launch sites in close proximity to railways, highways and coastlines? Yes.
  - Do launch sites keep certain distance away from cities? Yes

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/6.%20IBM%20Data%20Science_SpaceX%20Launch_Site_Location_Rakib.ipynb

Build a Dashboard with Plotly Dash

Below steps were taken during this Plotly Dash stage:

- We build an interactive dashboard with Plotly dash
- A pie chart that shows the successful launch by each site. This chart is useful as you can visualize the distribution of landing outcomes across all launch sites or show the success rate of launches on individual sites.
- A scatter chart that shows the relationship between landing outcomes and the payload mass of different boosters. The dashboard takes two inputs, namely the site(s) and payload mass. This chart is useful as you can visualize how different variables affect the landing outcomes

My completed python notebook is available via GitHub link below:

https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/IBM%20Data%20Science_SpaceX_Plotly_App_Rakib.py

Predictive Analysis (Classification)

Below steps were taken during this Machine Learning Predictive Analysis:

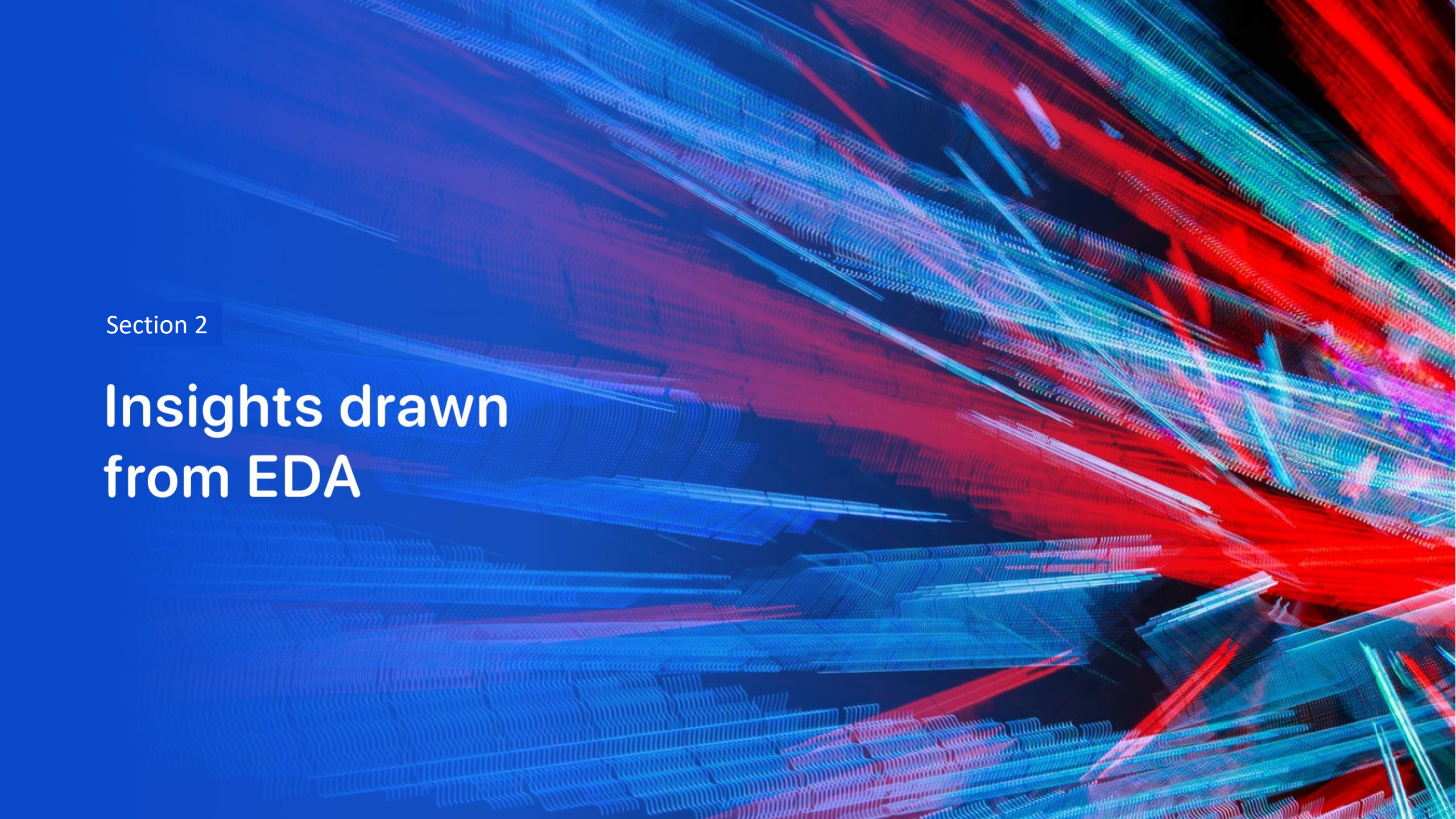
- We loaded the data using numpy and pandas, transformed the data, split the data into training and testing
- We built different machine learning models (for SVM, Decision Tree, K-Nearest Neighbours and Logistic Regression) and tune different hyperparameters using GridSearchCV
- Use test data to evaluate models based on their accuracy scores and confusion matrix

My completed python notebook is available via GitHub link below:

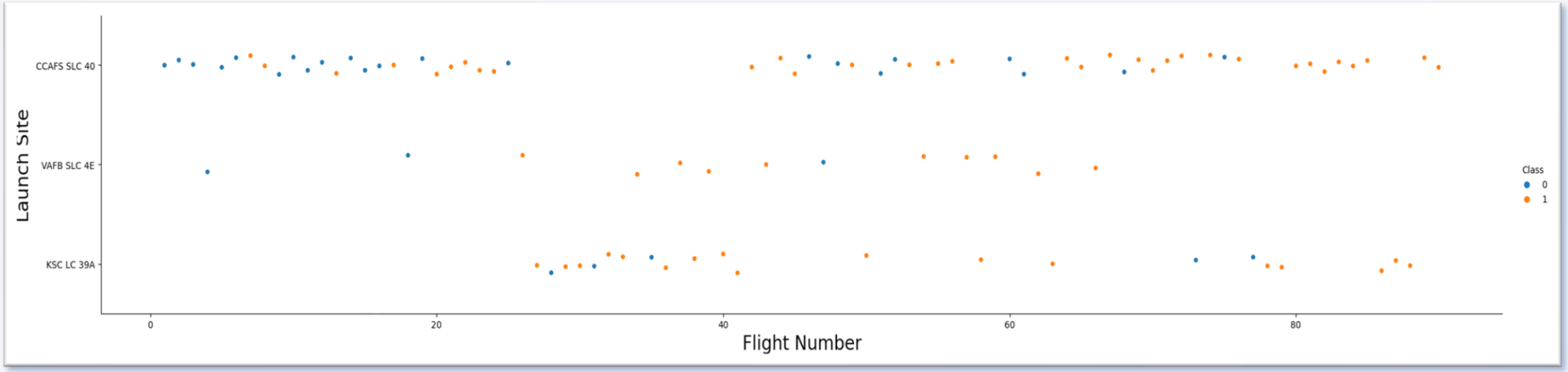
https://github.com/rakibsumon/IBM_Data_Science_Capstone/blob/main/7.%20IBM%20Data%20Science_SpaceX_Machine_Learning_Prediction_Rakib.ipynb

Results (in the following sections)

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

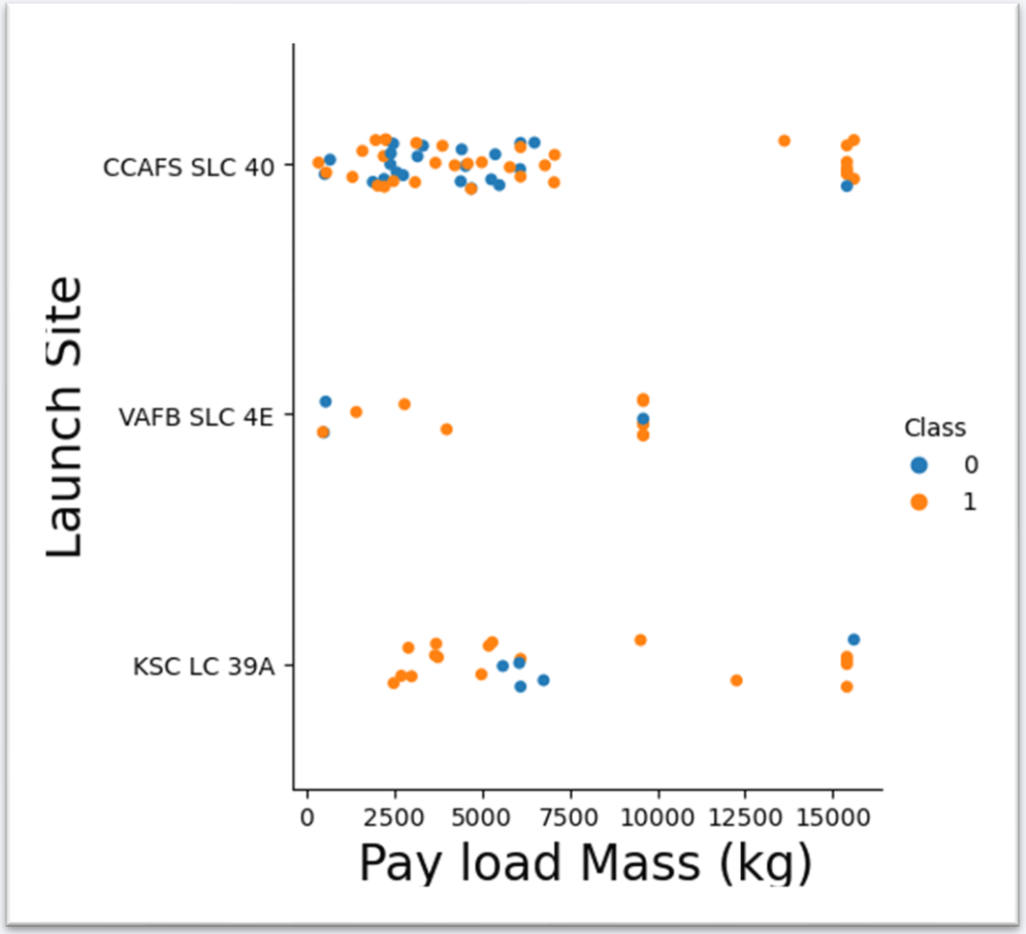

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



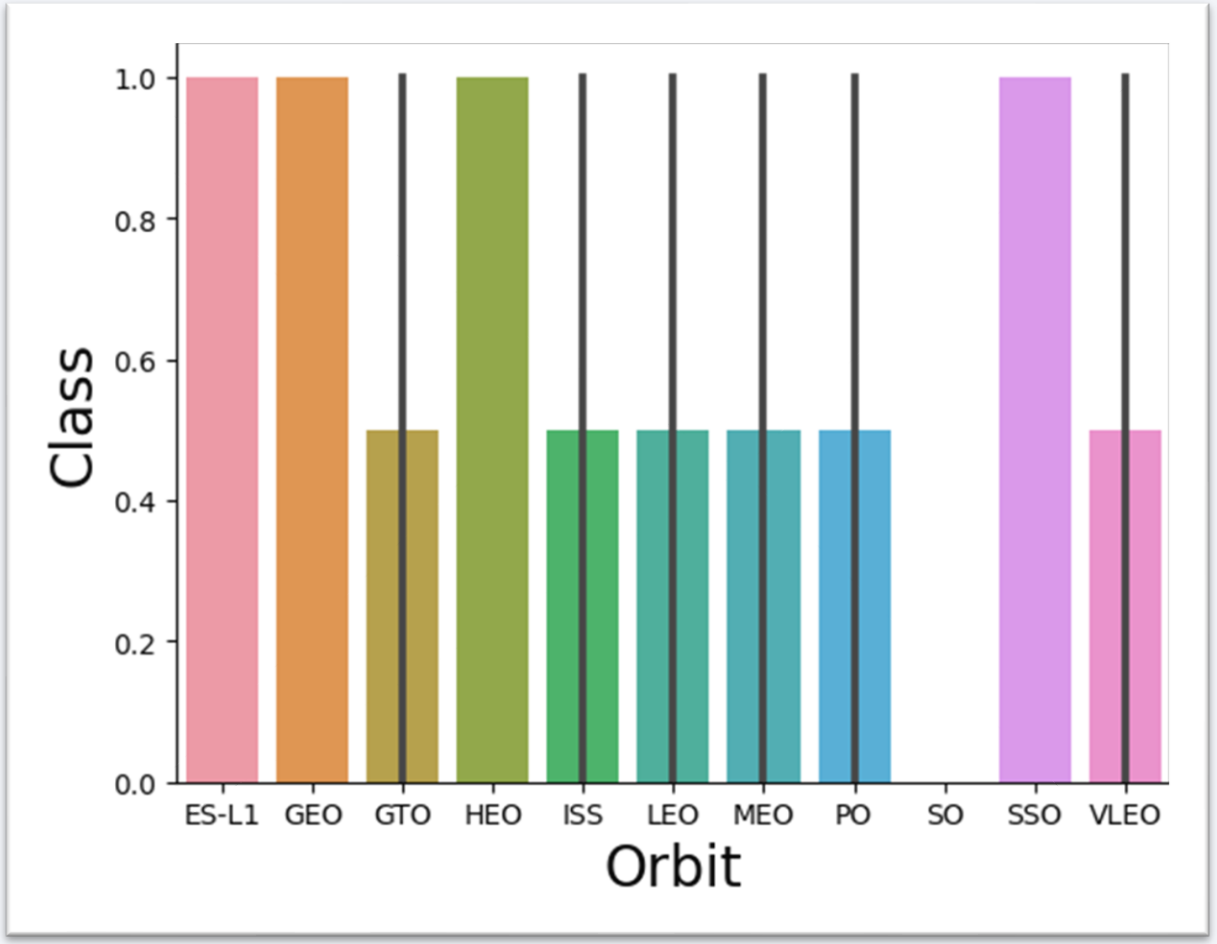
- Above plot shows that success rate increased as the number of flights increased.
- There seems to be an increase in successful flights after the 40th launch.

Payload vs. Launch Site



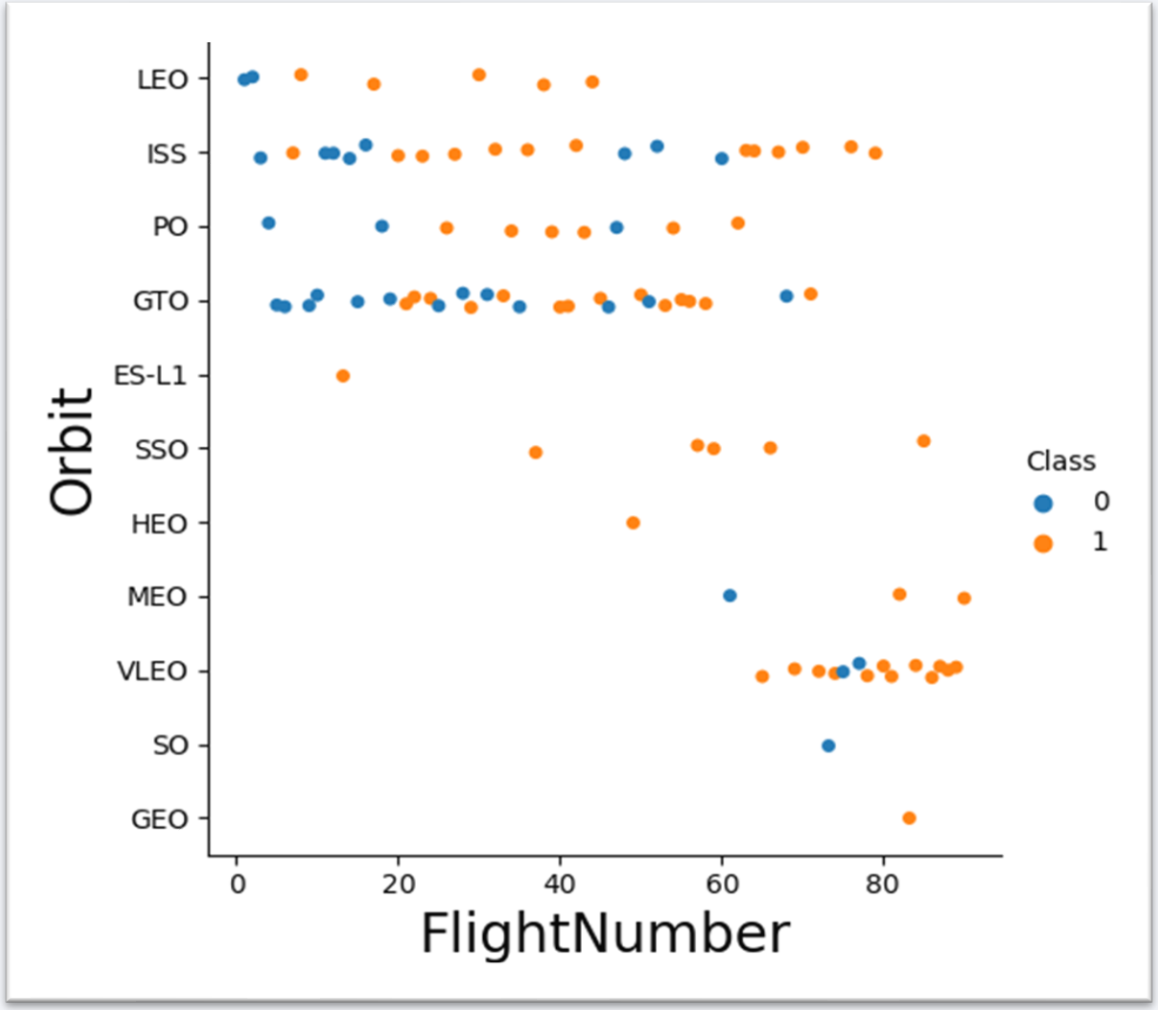
- Apparently, for the VAFB SLC 4E launch site, there are no rockets launched with heavy payload mass.
- There seems to be a weak correlation between Payload and Launch Site.

Success Rate vs. Orbit Type



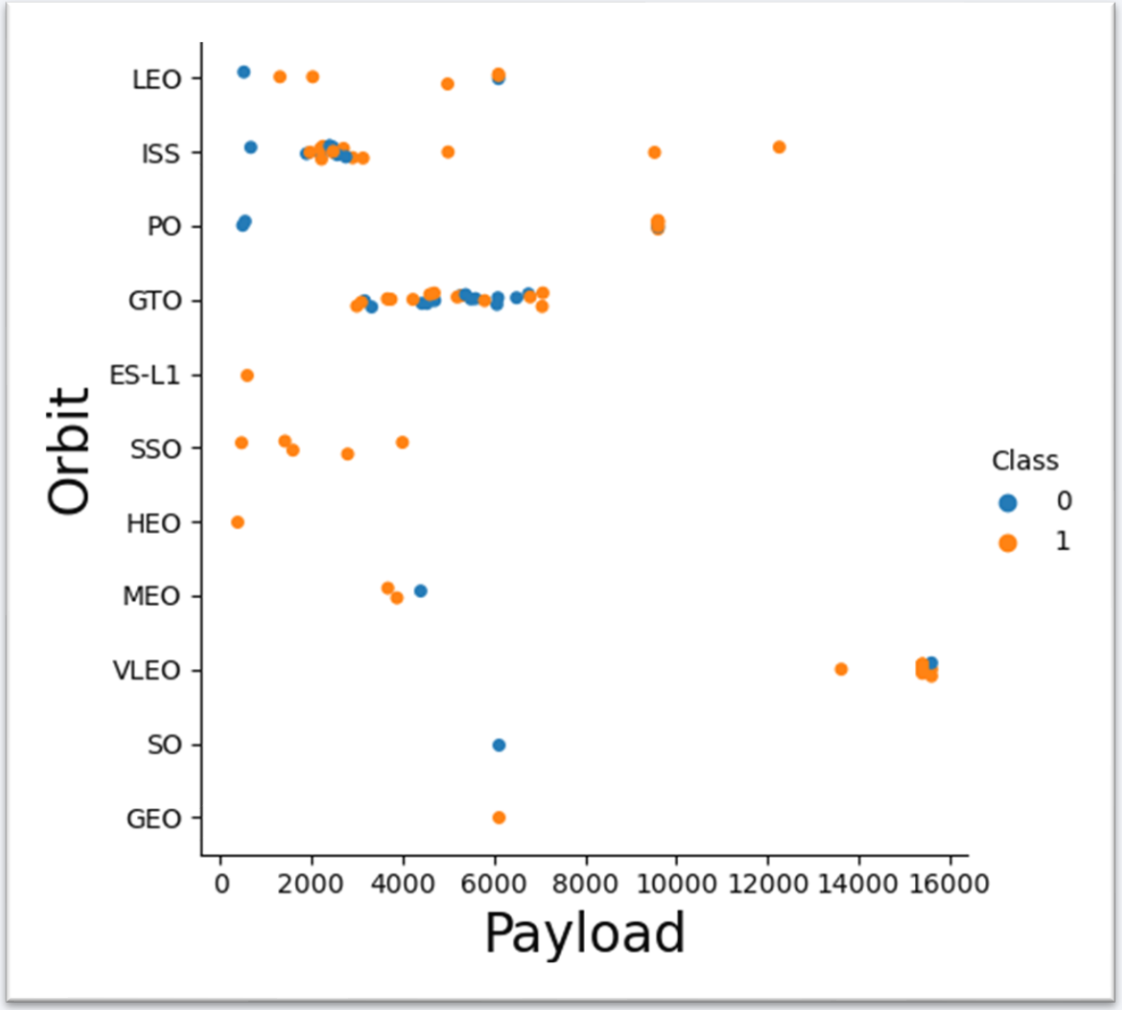
- Orbits ES-L1, GEO, HEO and SSO have the most success rates.
- SO orbit did not have any successful launches.

Flight Number vs. Orbit Type



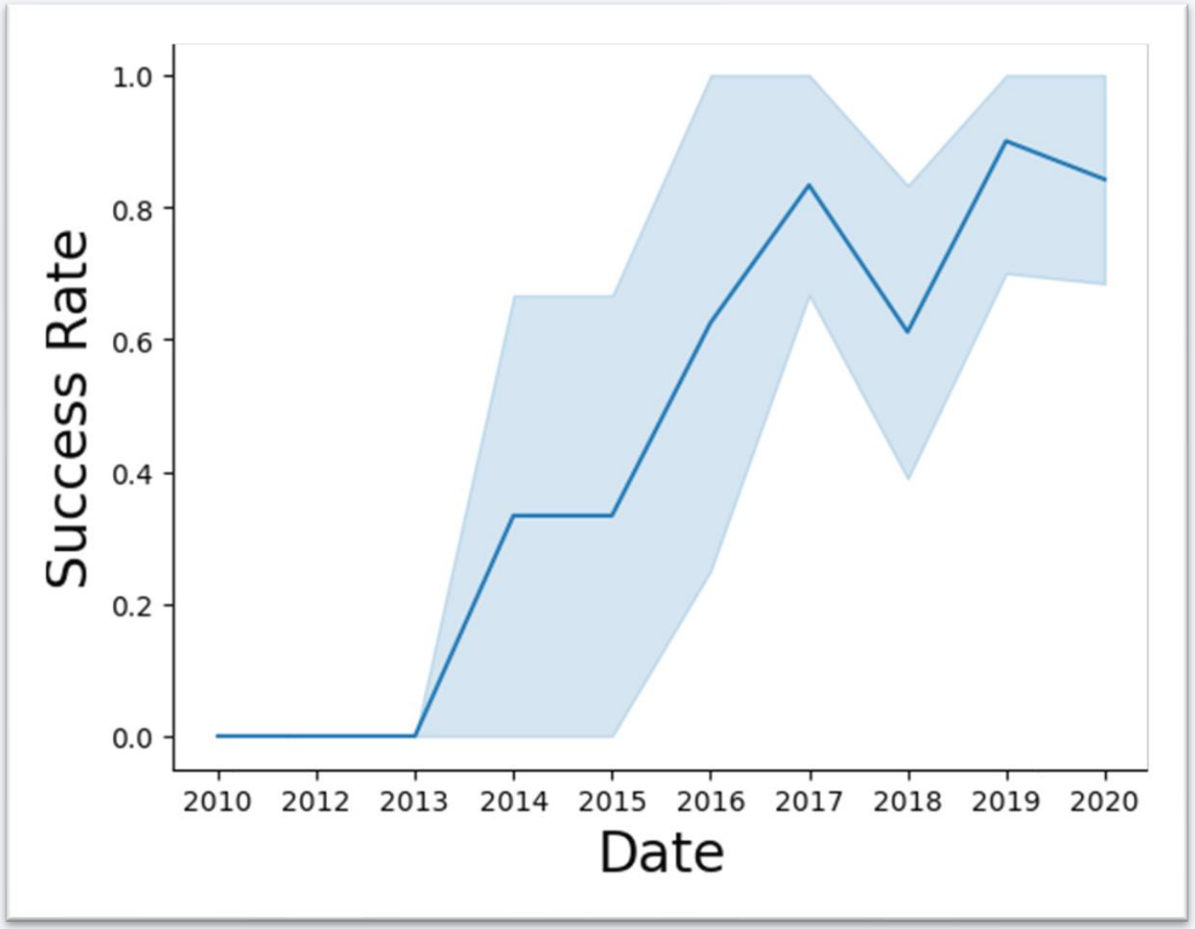
- In the LEO orbit, the success is positively correlated to the the number of flights.
- There seems to be no relationship between flight number in the GTO orbit.
- The SSO orbit has a 100% success rate.
- Overall, success rate increases for fights numbers greater than 40.

Payload vs. Orbit Type



- As the payloads get heavier, the success rate increases in the PO, SSO, LEO and ISS orbits.
- There seems to be no direct correlation between orbit type and payload mass for GTO orbit.

Launch Success Yearly Trend



- From the yearly trend, as plotted in the line chart, we observe that launch success rates are increasing after 2013.

All Launch Site Names

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

```
In [8]: %sql SELECT distinct "Launch_Site" FROM SPACEXTBL
```

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

Out[8]:

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

- List of unique launch sites were achieved by using DISTINCT

Launch Site Names Begin with 'CCA'

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

In [9]:
```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

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

Out[9]:

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

- The desired result is achieved by using LIKE ‘CCA%’ and LIMIT

Total Payload Mass

- We calculated the total payload carried by boosters from NASA is 45,596 KG

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

In [11]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer = 'NASA (CRS)'
```

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

Out[11]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 is 2,928.4 KG.

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

In [13]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION = 'F9 v1.1'
```

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

Out[13]:

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

- We found the date of the first successful landing outcome on ground pad was on 22nd December 2015.

```
In [14]: %sql SELECT min(DATE) FROM SPACEXTBL WHERE "Landing_Outcome"  = 'Success (ground pad)'
 * sqlite:///my_data1.db
Done.
Out[14]: min(DATE)
         2015-12-22
```

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used WHERE, BETWEEN & AND to get the list of the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
In [17]: %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000 AND "Landing_Outcome"  = 'Succe
 * sqlite:///my_data1.db
Done.
```

Out[17]:

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

Total Number of Successful and Failure Mission Outcomes

- We calculated the total number of successful and failure mission outcomes using WHERE, LIKE and using wildcard %

List the total number of successful and failure mission outcomes

In [18]:
```
%sql SELECT COUNT(*) FROM SPACEXTBL WHERE "Mission_Outcome" LIKE '%Success%' OR "Mission_Outcome" LIKE '%Failure%'
```

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

Out[18]:

COUNT(*)
101

Boosters Carried Maximum Payload

```
In [19]: %sql SELECT "Booster_Version" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

Out[19]:

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

- We used WHERE clause and MAX() to get the list of the names of the booster which have carried the maximum payload mass.

2015 Launch Records

- We used WHERE clause, LIKE, AND conditions to get failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

In [51]:
```
%sql SELECT STRFTIME("%m-%Y",Date) AS MONTH_YEAR, \
    "Landing_Outcome" AS LANDING__OUTCOME, \
    "Booster_Version" AS BOOSTER_VERSION, \
    "Launch_Site" AS LAUNCH_SITE \
    FROM SPACEXTBL WHERE "Landing_Outcome" = 'Failure (drone ship)' AND Date LIKE '%2015%'
```

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

Out[51]:

MONTH_YEAR	LANDING__OUTCOME	BOOSTER_VERSION	LAUNCH_SITE
10-2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04-2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

In [20]:
```
%sql SELECT "Landing_Outcome", COUNT("Landing_Outcome") FROM SPACEXTBL \
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' \
    GROUP BY "Landing_Outcome" \
    ORDER BY COUNT("Landing_Outcome") DESC
```

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

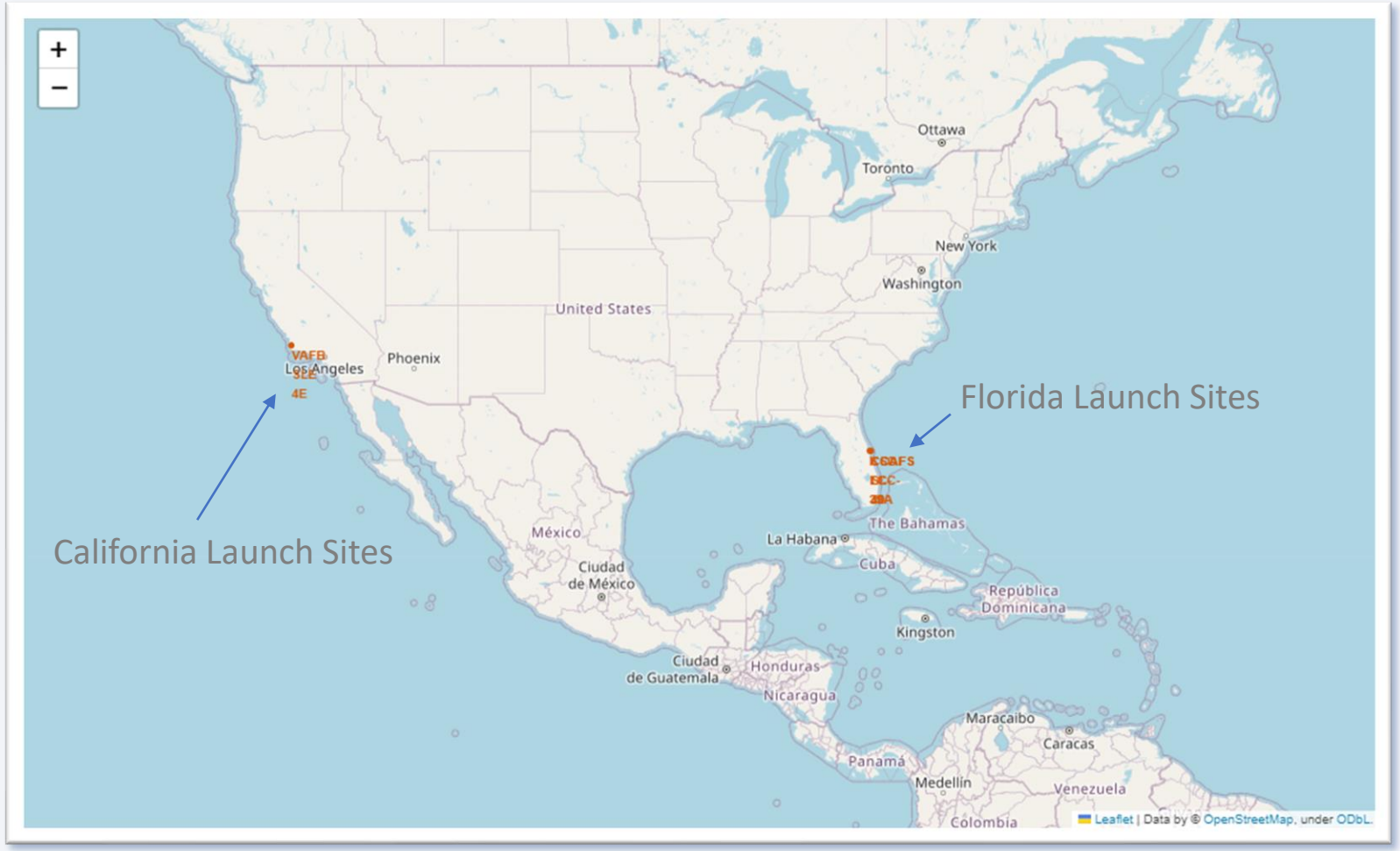
Out[20]:

Landing_Outcome	COUNT("Landing_Outcome")
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

- We used COUNT, WHERE, BETWEEN, GROUP BY and ORDER BY to get the desired result.

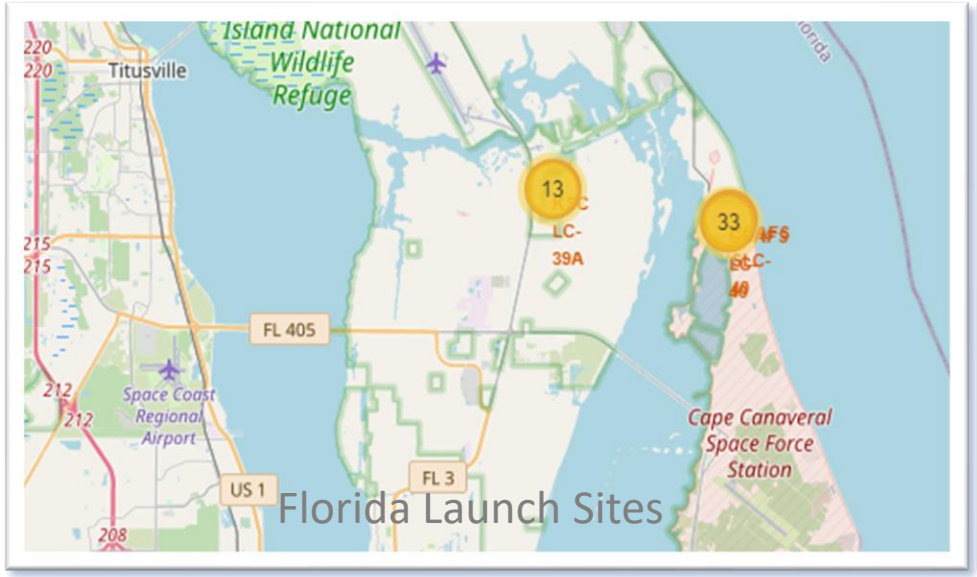

Section 3
Launch Sites
Proximities Analysis

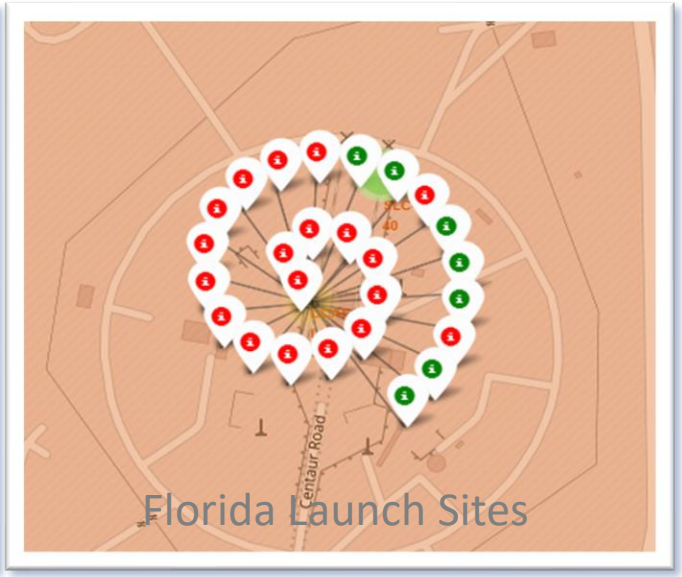
All SpaceX Launch Sites

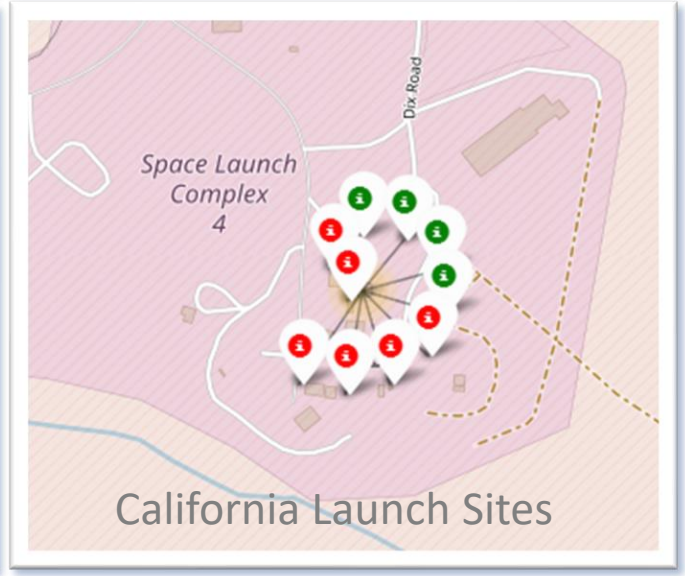


- SpaceX launch sites are located in Florida and California in the United States of America.

Markers Showing Different Launch Sites

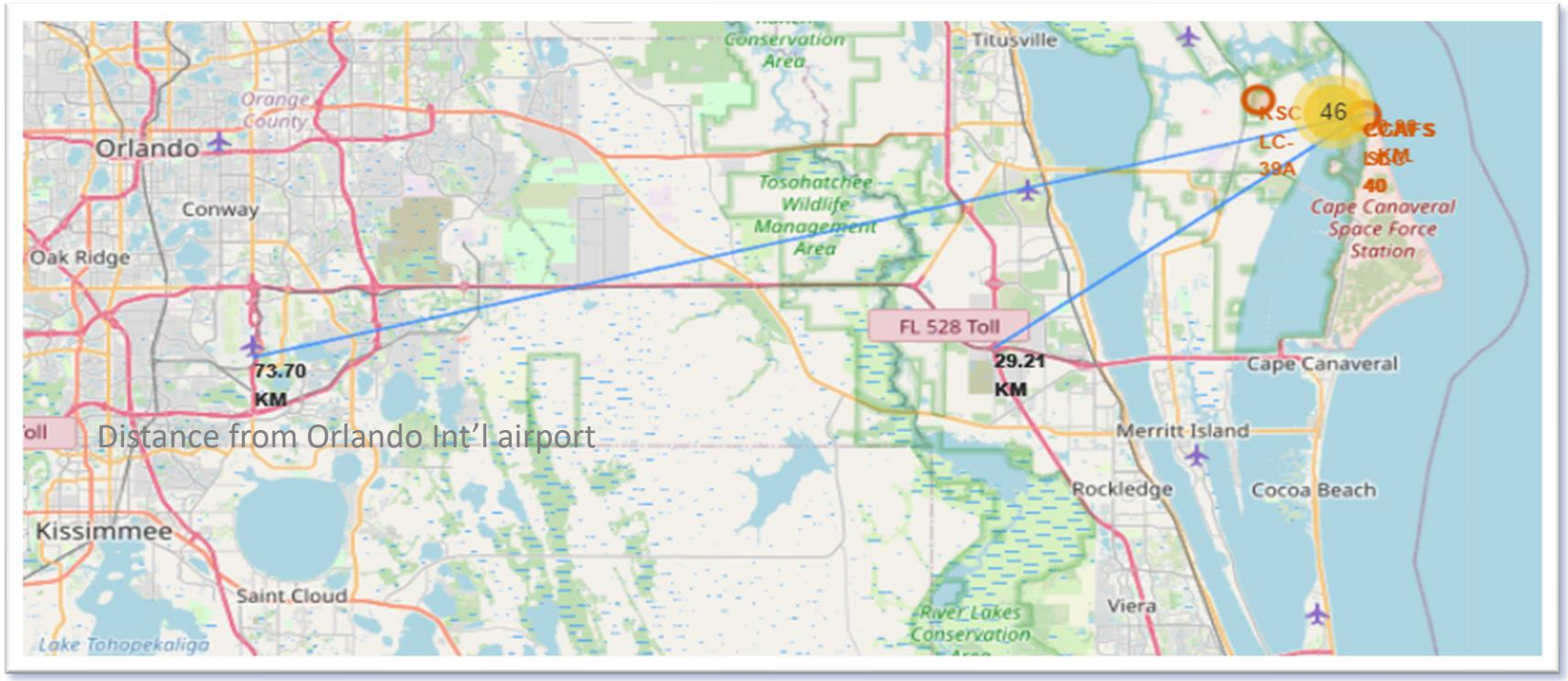


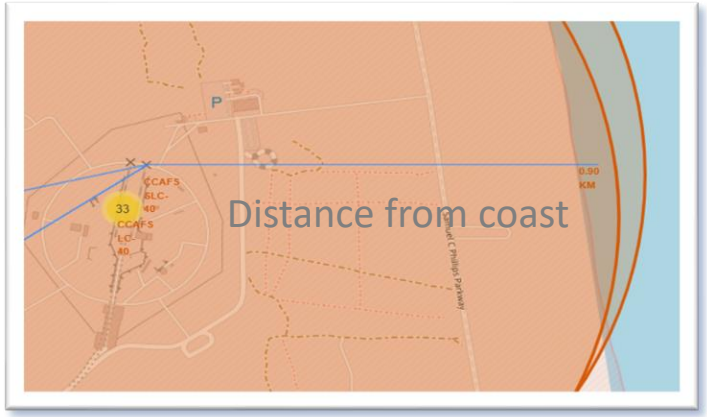


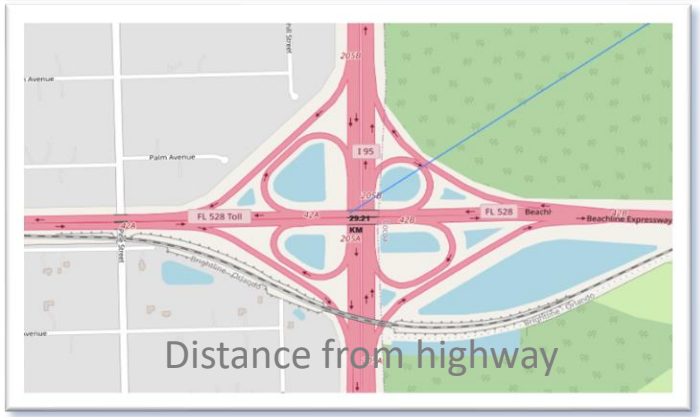


Green marker shows successful launches and Red shows failures.

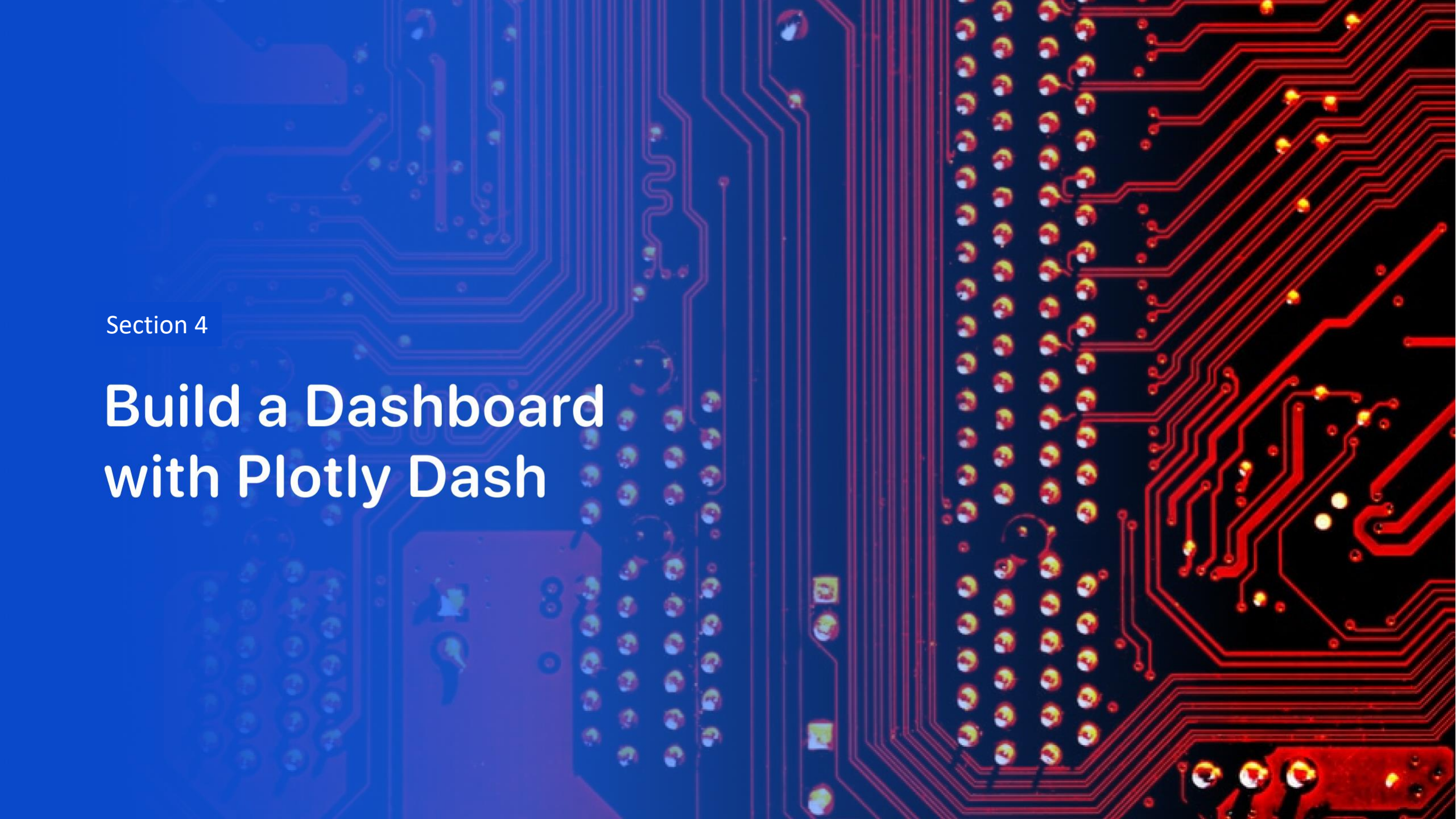
Markers Showing Different Launch Sites



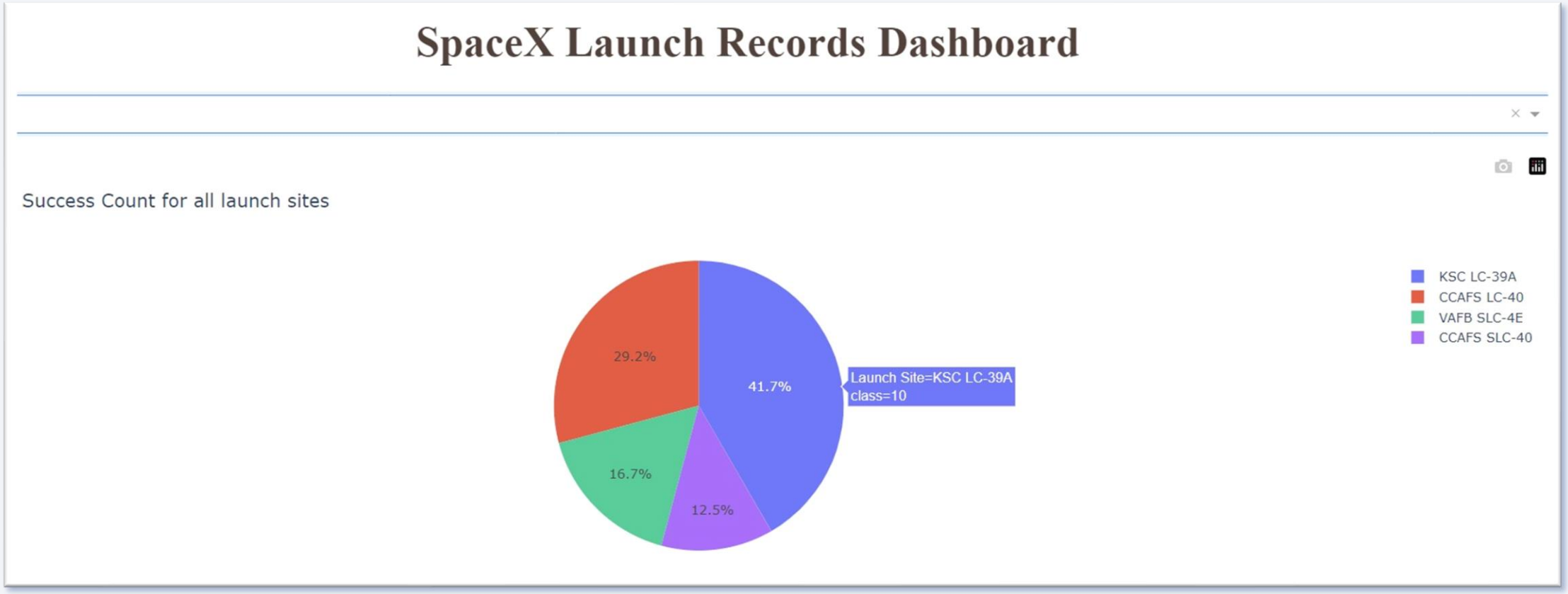




- Are launch sites in close proximity to railways? - Yes
- Are launch sites in close proximity to highways? - Yes
- Are launch sites in close proximity to coastline? - Yes
- Do launch sites keep certain distance away from cities? - Yes

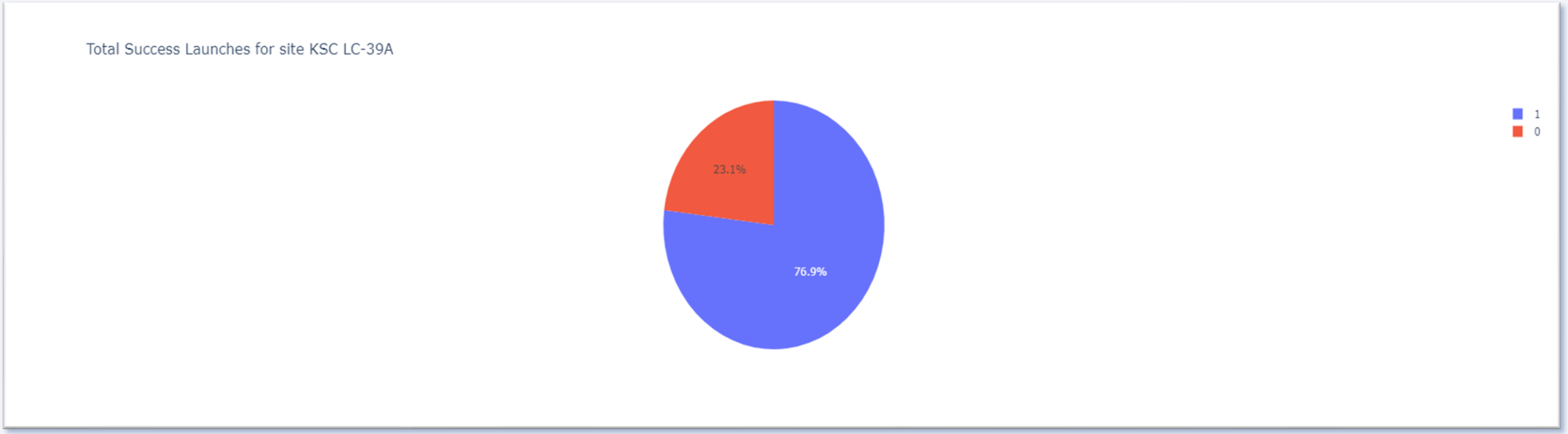

Section 4
Build a Dashboard with Plotly Dash

Total Successful Launches by Sites



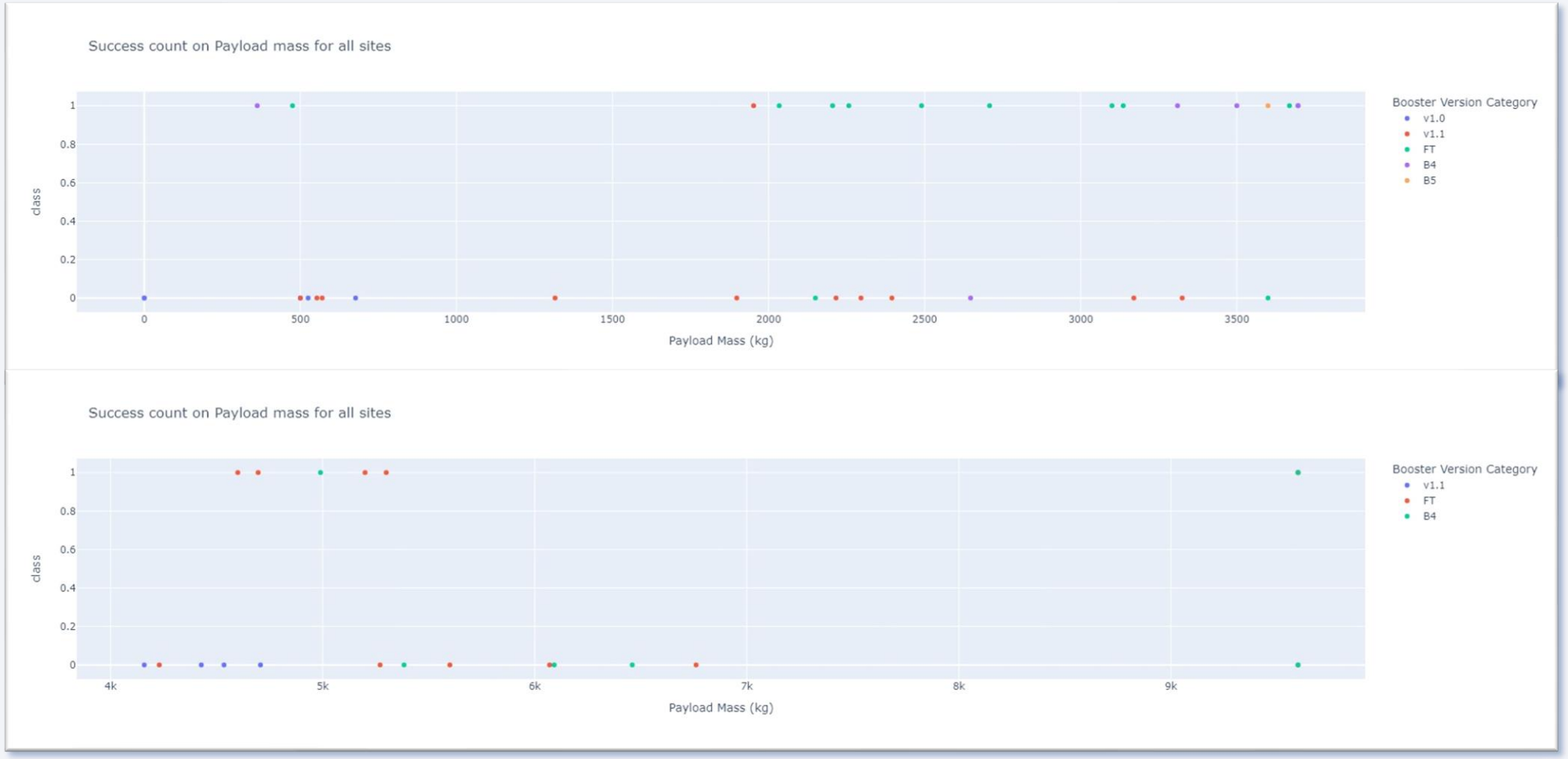
- The KSC LC-39A Launch site has the most successful launches with 10 in total.

Launch Site With Highest Success Ratio



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

Payload vs. Launch Outcome



- We observe that success rates for low weighted payloads is higher than the heavy payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

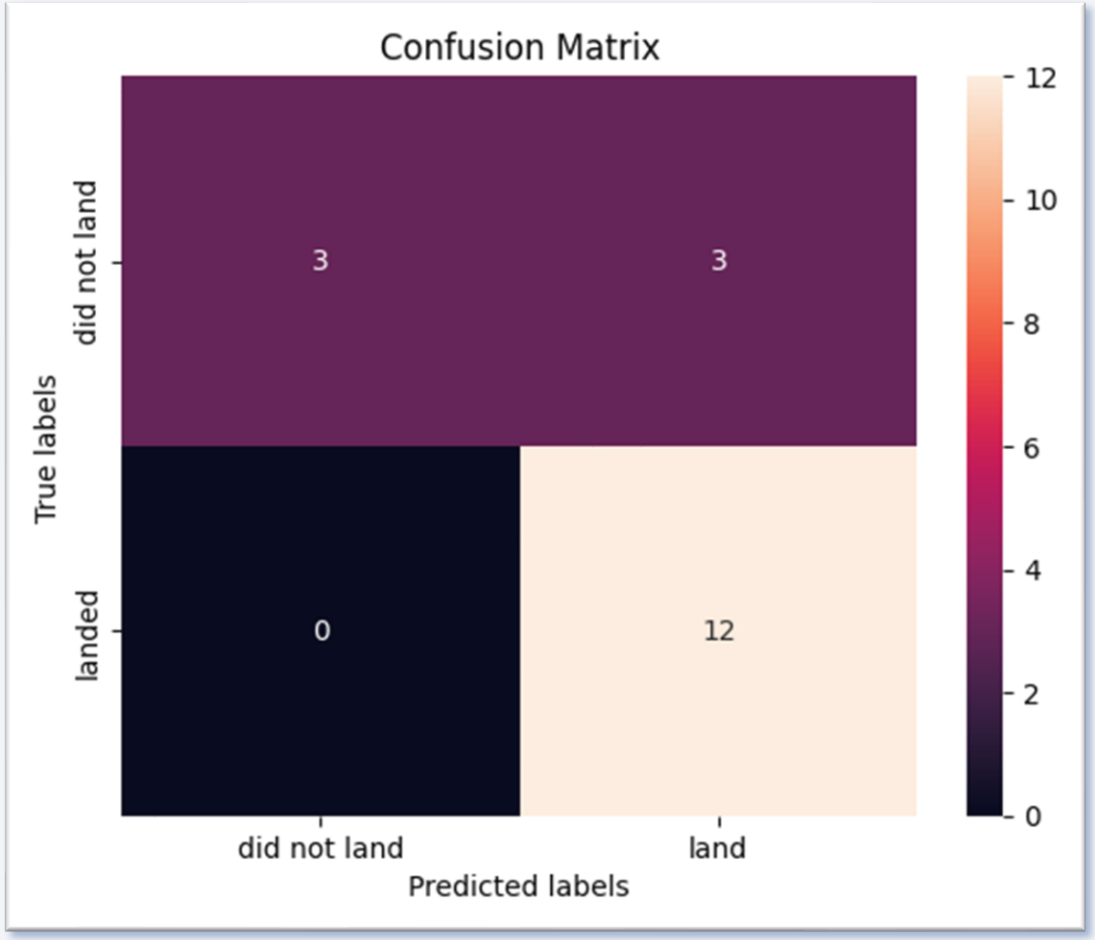
Find the method performs best:

In [49]:
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearest neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.8333333333333334
Accuracy for K nearest neighbors method: 0.8333333333333334
```

- During my analysis, I found all models have the same level of classification accuracy at 83.34%. This is likely due to small dataset.

Confusion Matrix



- Confusion matrix summarizes the performance of a classification algorithm.
- Confusion matrix outcomes for all models:
  - 12 True positive
  - 3 True negative
  - 3 False positive
  - 0 False Negative

Conclusions

After all the analysis, we can conclude that:

- There is a positive correlation between number of flights and success rate as the success rate has improved over the years.
- Launch success rate started to increase since 2013.
- Orbits ES-L1, GEO, HEO, SSO, VLEO have the most success rates.
- Launch site KSC LC-39A has the most successful launches.
- All the models performed similarly on the test set.

Appendix

- More information on SpaceX Falcon 9 launches: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches
- Link to my GitHub repository: https://github.com/rakibsumon/IBM_Data_Science_Capstone

Thank you!