IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniel Alessi
27/01/24

Outline

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

Executive Summary

Summary of Methodologies

- Data Collection
- Data Wrangling
- EDA & Data Visualization
- EDA & SQL Querying
- Folium Interactive Map
- Plotly Dash Dashboard
- Predictive Machine Learning

Summary of Results

- Exploratory Data Analysis
- Interactive Map & Dashboard
- Predictive Analytics Classifier

Introduction

Project Context

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.
- If we can determine if the first stage will land, we can determine the launch cost.

Problem to Answer

- This project utilises data science approaches to analyse Space X rocket launches, transforming recorded data to predict future success of Falcon 9 rocket launches.

Section 1

Methodology

Methodology

Data Collection Methodology

- SpaceX REST API, Wikipedia Web Scraping

Perform Data Wrangling

- One Hot Encoding, Training Labels

Perform Exploratory Data Analysis & SQL

- Pandas, Matplotlib Visualizations & SQL

Perform Interactive Visual Analytics & Dashboard

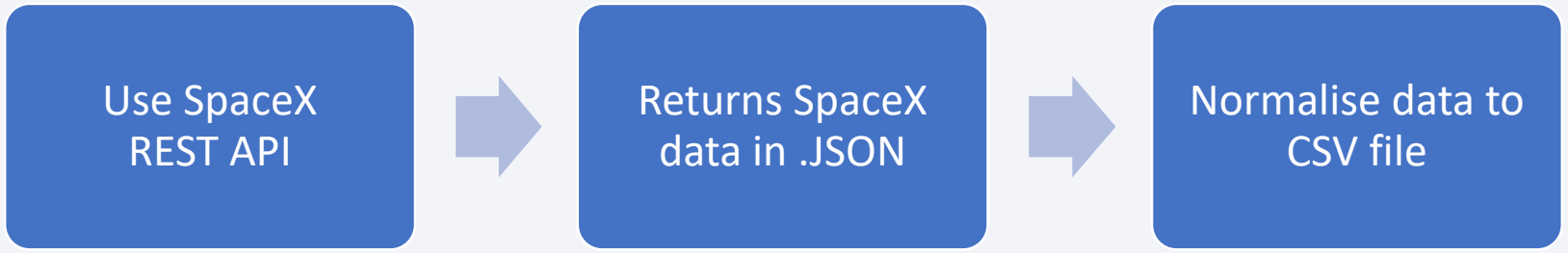
- Folium Interactive Map & Plotly Dash Dashboard

Perform Predictive Analytics using Machine Learning

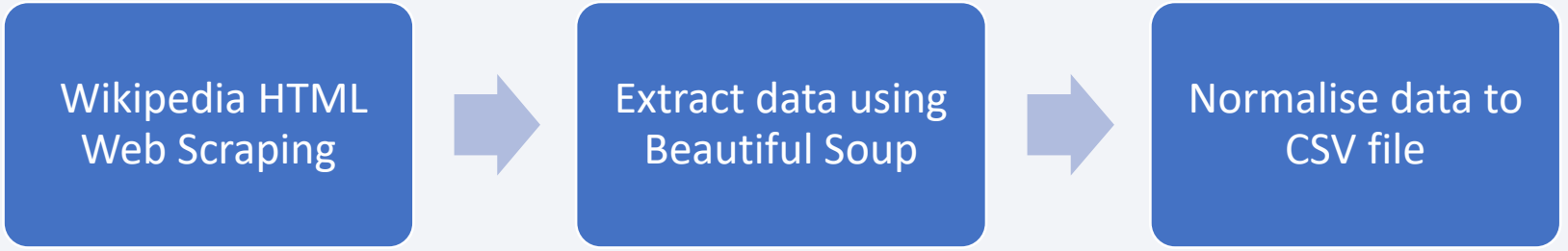
- ML Classification Models: LR, KNN, SVM, DT

Data Collection

SpaceX REST API



Wikipedia Web Scraping



Data Collection – SpaceX REST API

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

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

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

Finally lets construct our dataset using the data we have obtained. We we combine the columns into a dictionary.

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

Then, we need to create a Pandas data frame from the dictionary launch_dict.

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

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

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%201_Collecting%20Data%20%26%20SpaceX%20REST%20API.ipynb

Data Collection – Web Scraping

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

```
# 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`

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

Next, we want to collect all relevant column names from the HTML table header

```
html_tables = soup.find_all('table')
```

Next, we just need to iterate through the `<th>` elements

```
columns = []
headers = first_launch_table.find_all('th')
for header in headers:
    column = extract_column_from_header(header)
    if column is not None and len(column) > 0:
        columns.append(column)
```

Next create an empty dictionary with keys from the extracted column names

```
launch_dict= dict.fromkeys(columns)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

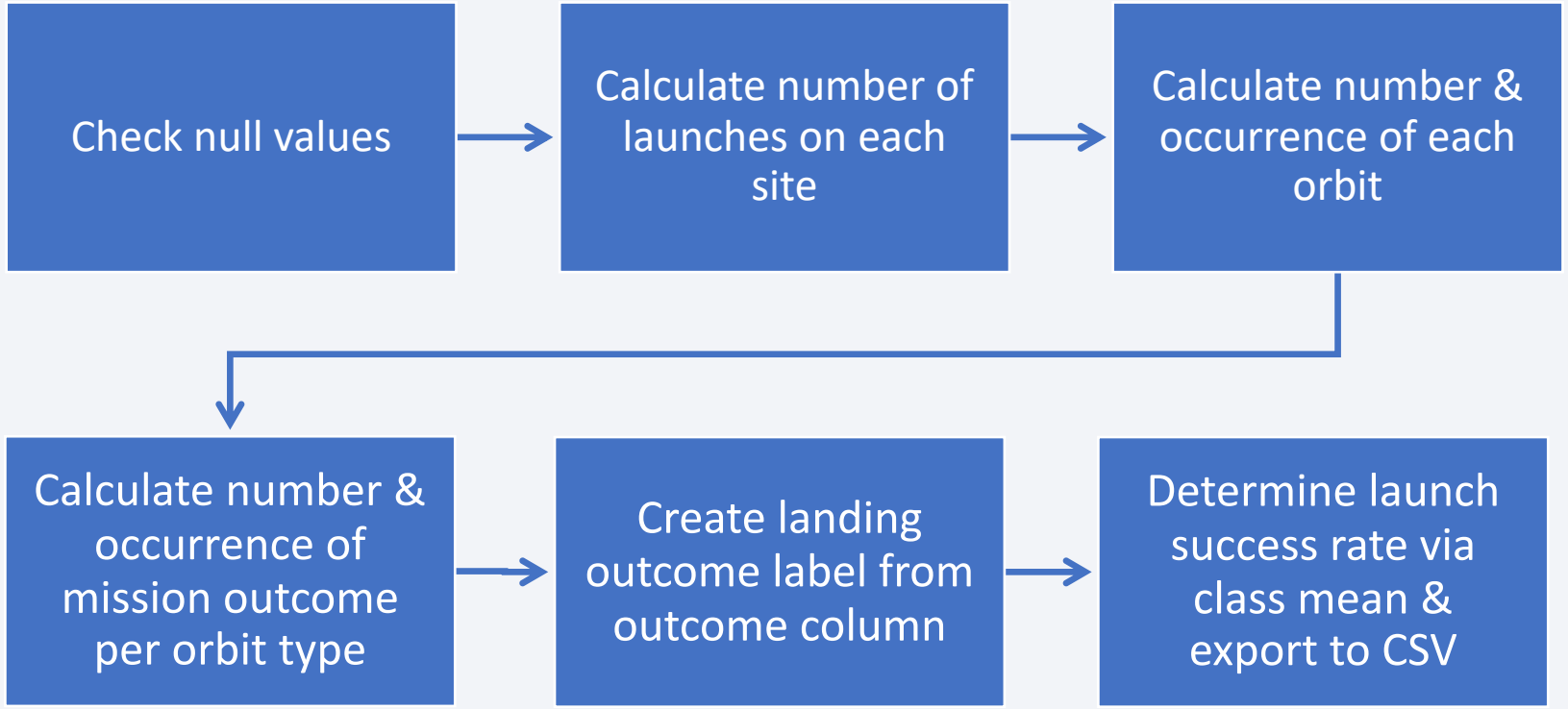
We can now export it to a CSV for the next section

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%201.1_Web%20Scraping%20from%20Wikipedia.ipynb

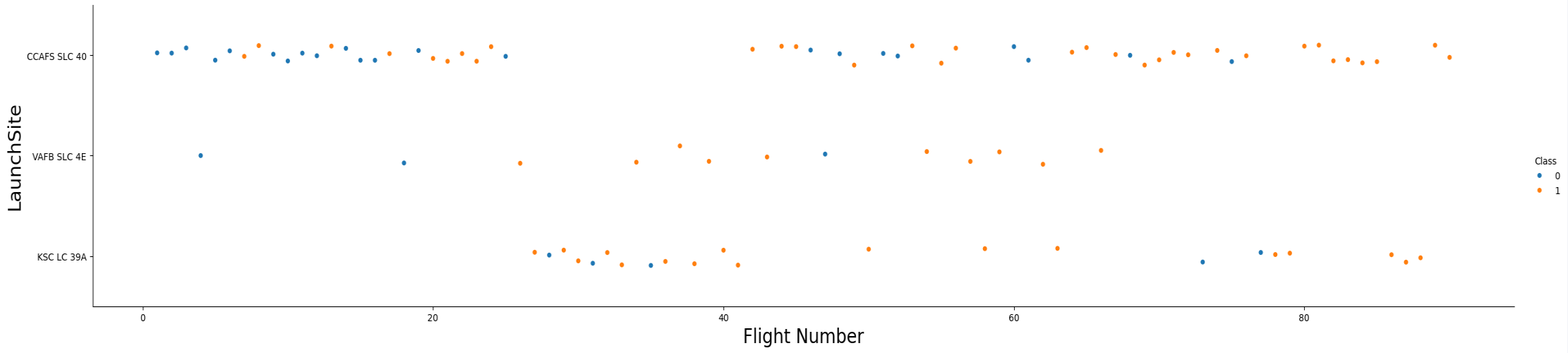
Data Wrangling

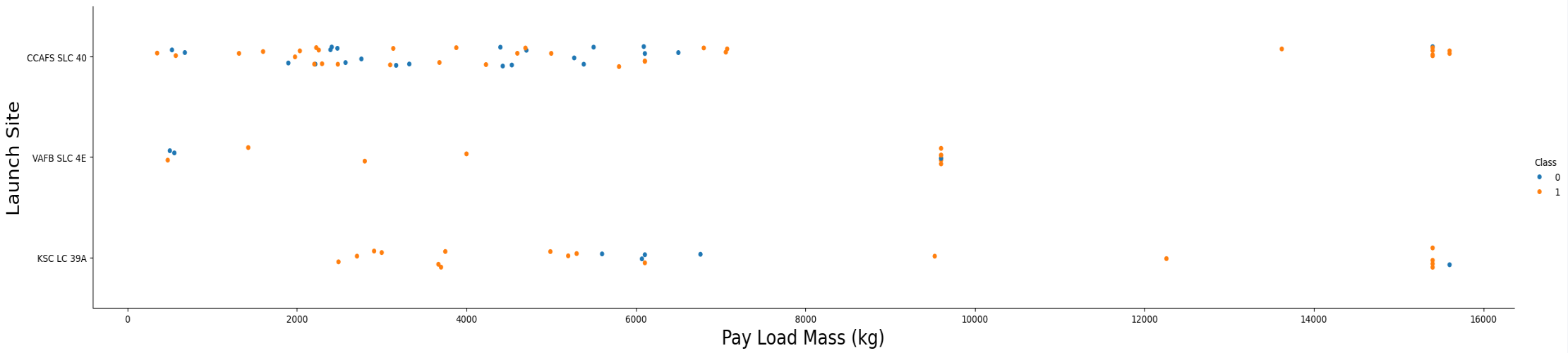
EDA Analysis



https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%202_Data%20Wrangling_EDA_Training%20Labels.ipynb

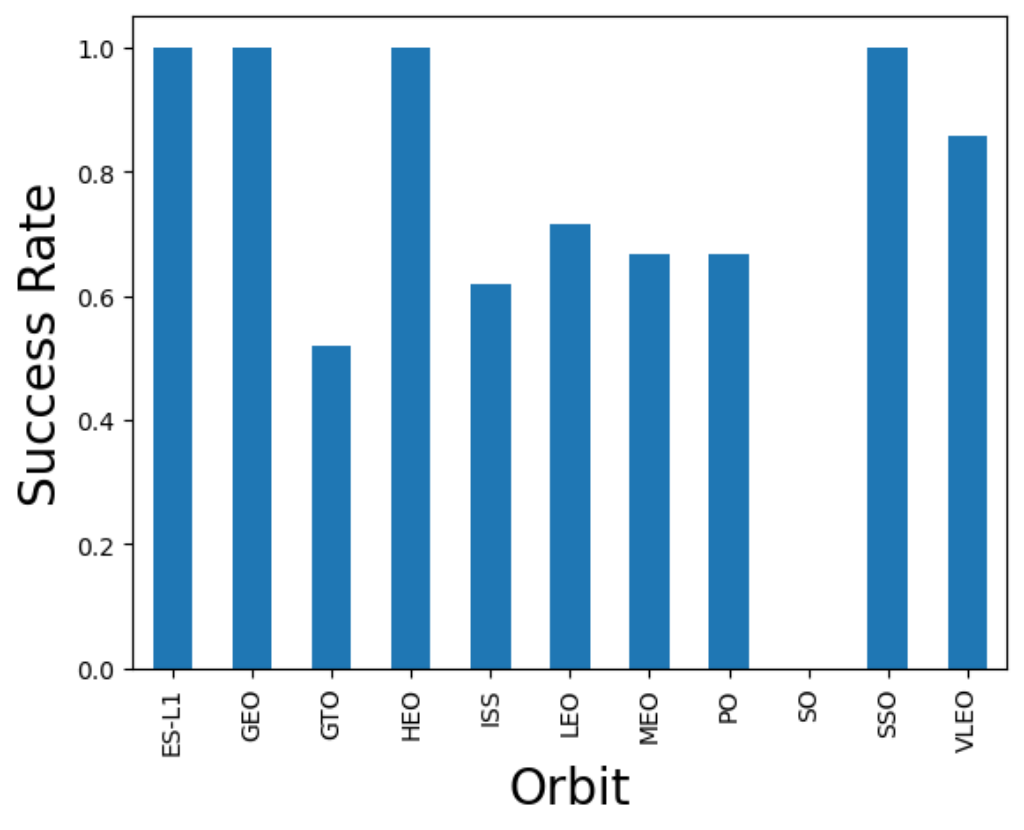
EDA with Data Visualization (1)

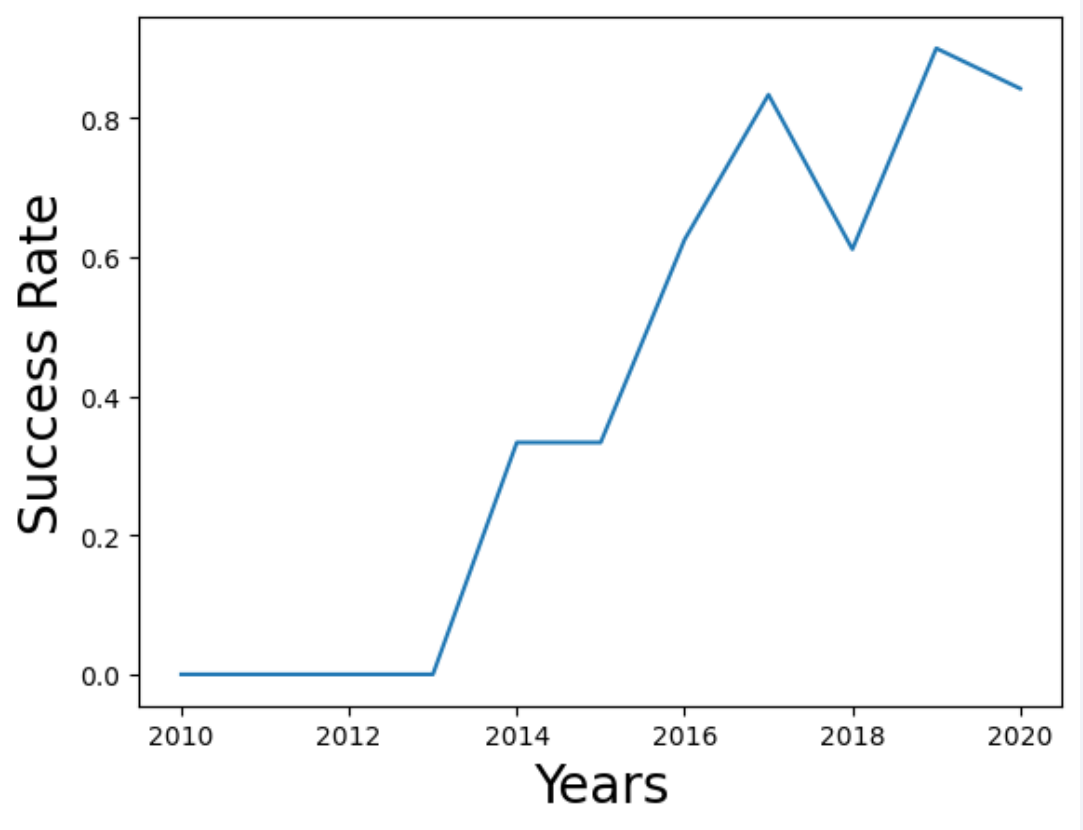




https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%203.1_EDA%20with%20Pandas%20%26%20Matplotlib.ipynb

EDA with Data Visualization (2)





https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%203.1_EDA%20with%20Pandas%20%26%20Matplotlib.ipynb

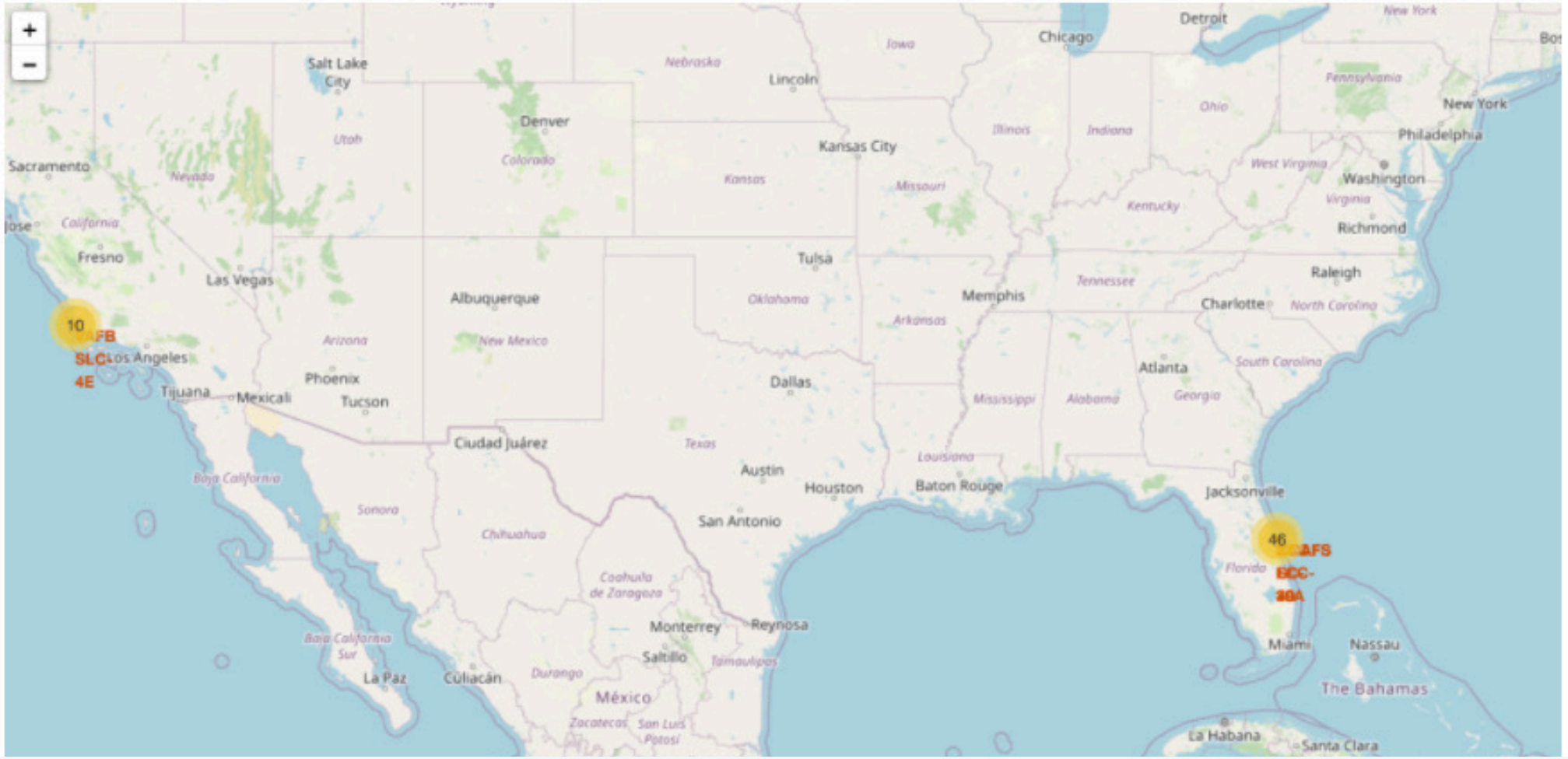
EDA with SQL

SQL Queries Performed

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string '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 successful 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 maximum payload mass. Used a subquery
- List the records which display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for 2015.
- Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%203_Load%20to%20Db2_SQL%20Lite%20EDA.ipynb

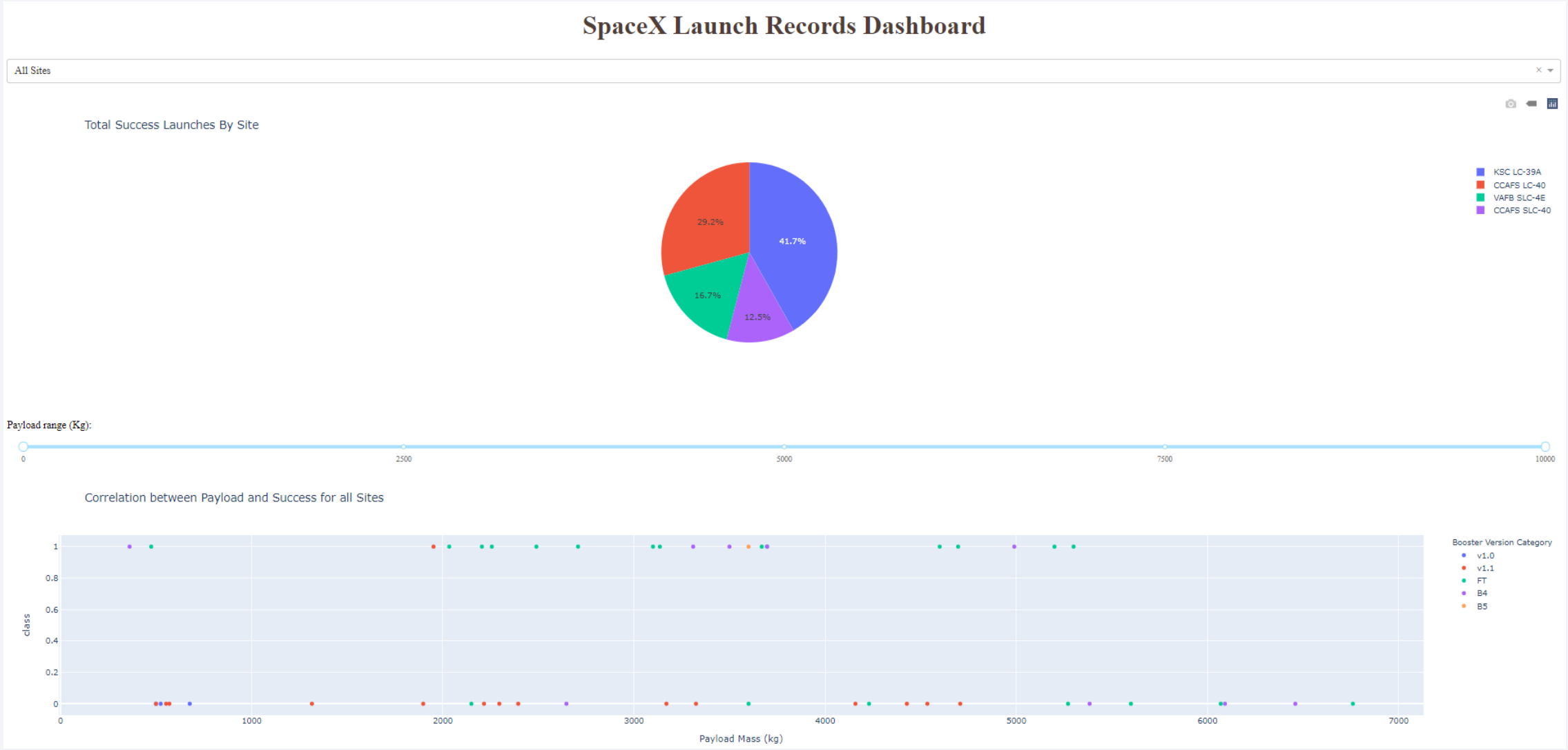
Build an Interactive Map with Folium



Markers added to Folium map to interactively visualise optimal launch locations.

https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%204_Folium%20Interactive%20Location%20Analytics.ipynb

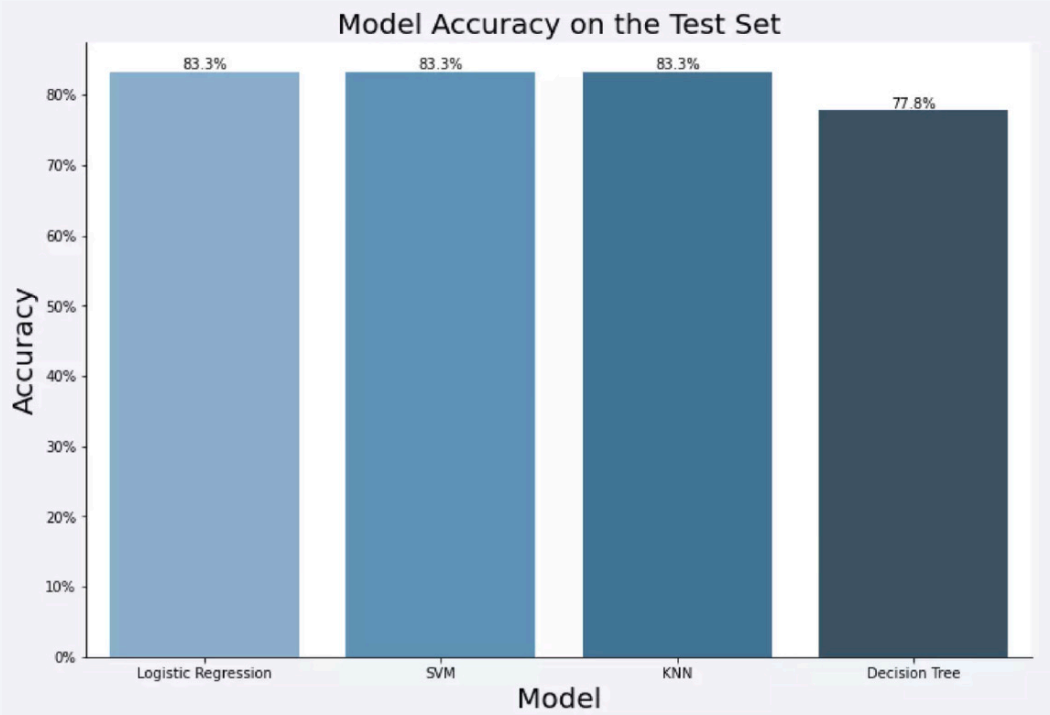
Build a Dashboard with Plotly Dash

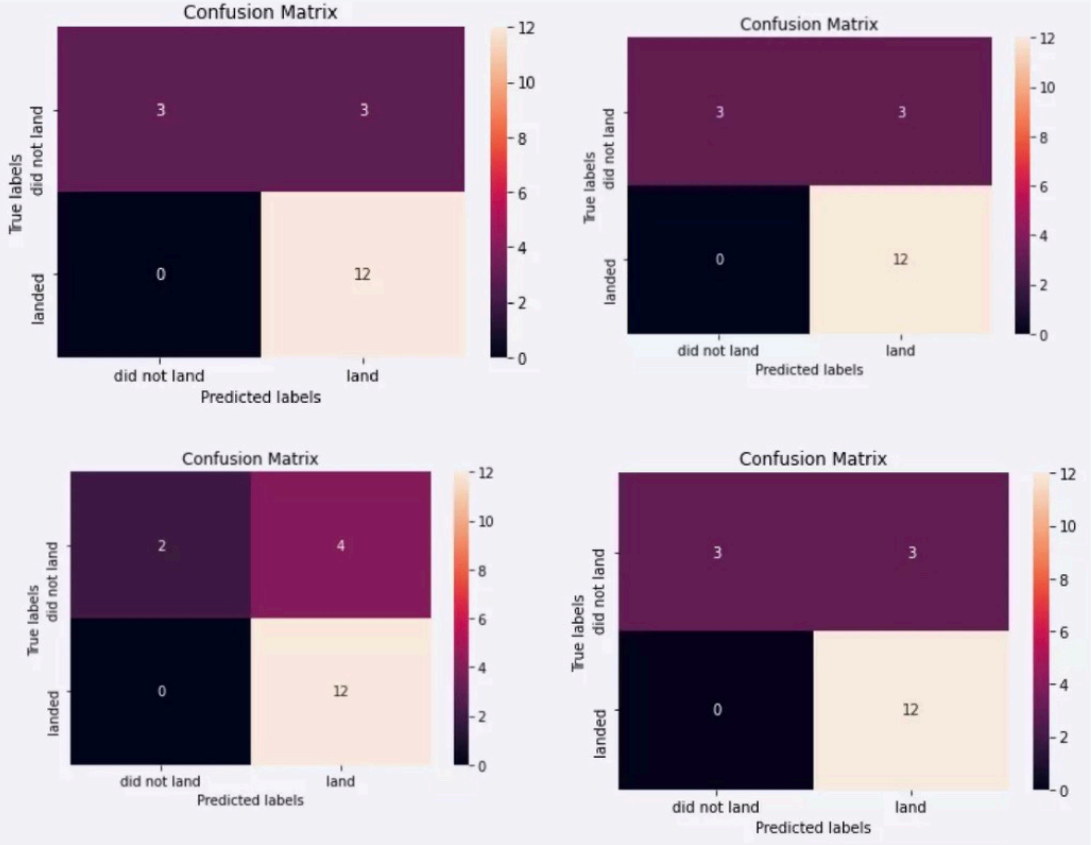


https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%204.1_Plotly%20Dashboard%20App.py

Predictive Analysis (Classification)

- Standardised and split data into training & test sets with `train_test_split` function.
- Built and evaluated Logistic Regression, SVM, KNN, and Decision Tree classifiers.





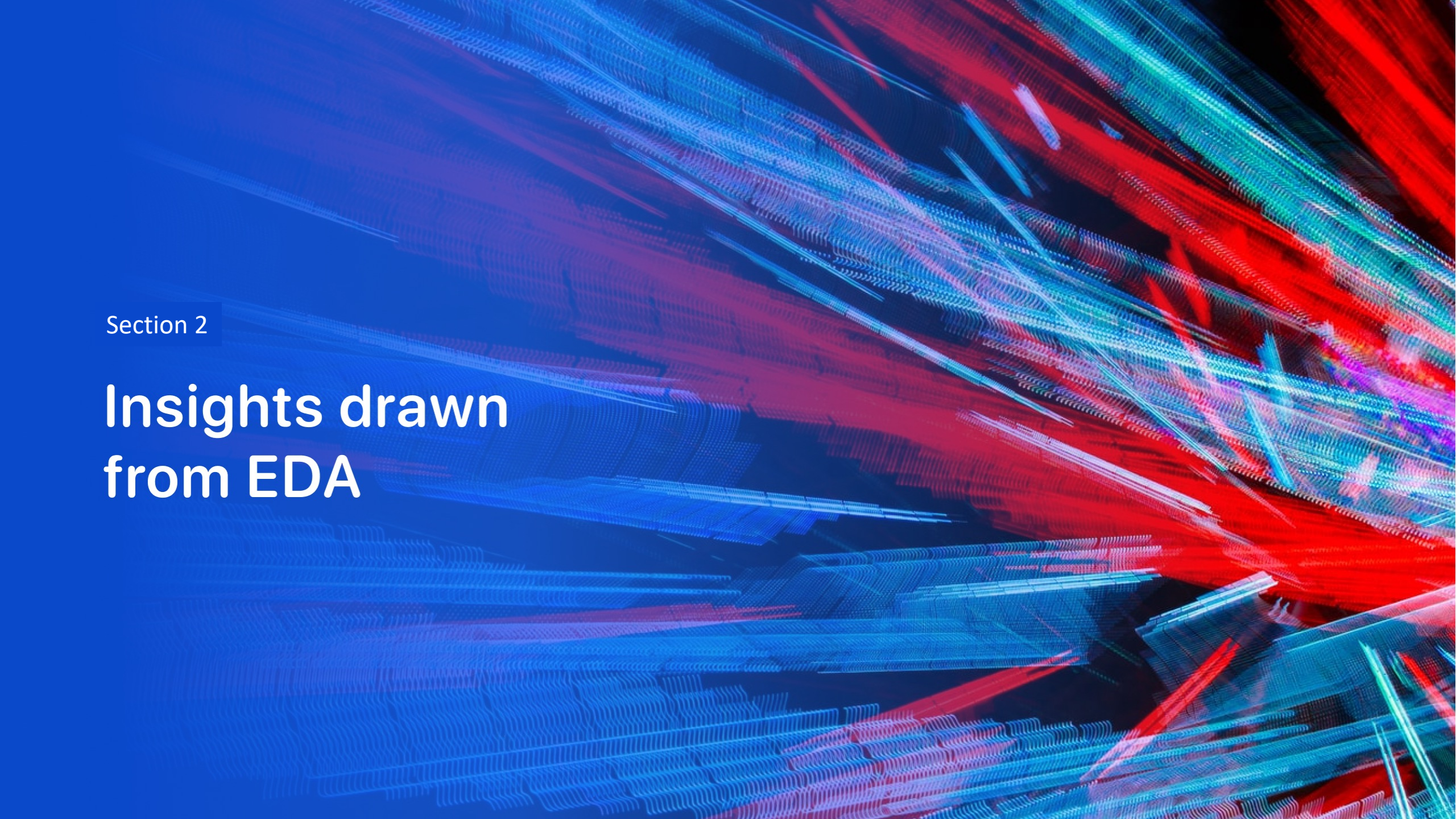
https://github.com/dalessi26/IBM-Data-Science-Capstone/blob/main/Lab%205_ML%20Prediction%20Model%20Evaluation.ipynb

Results

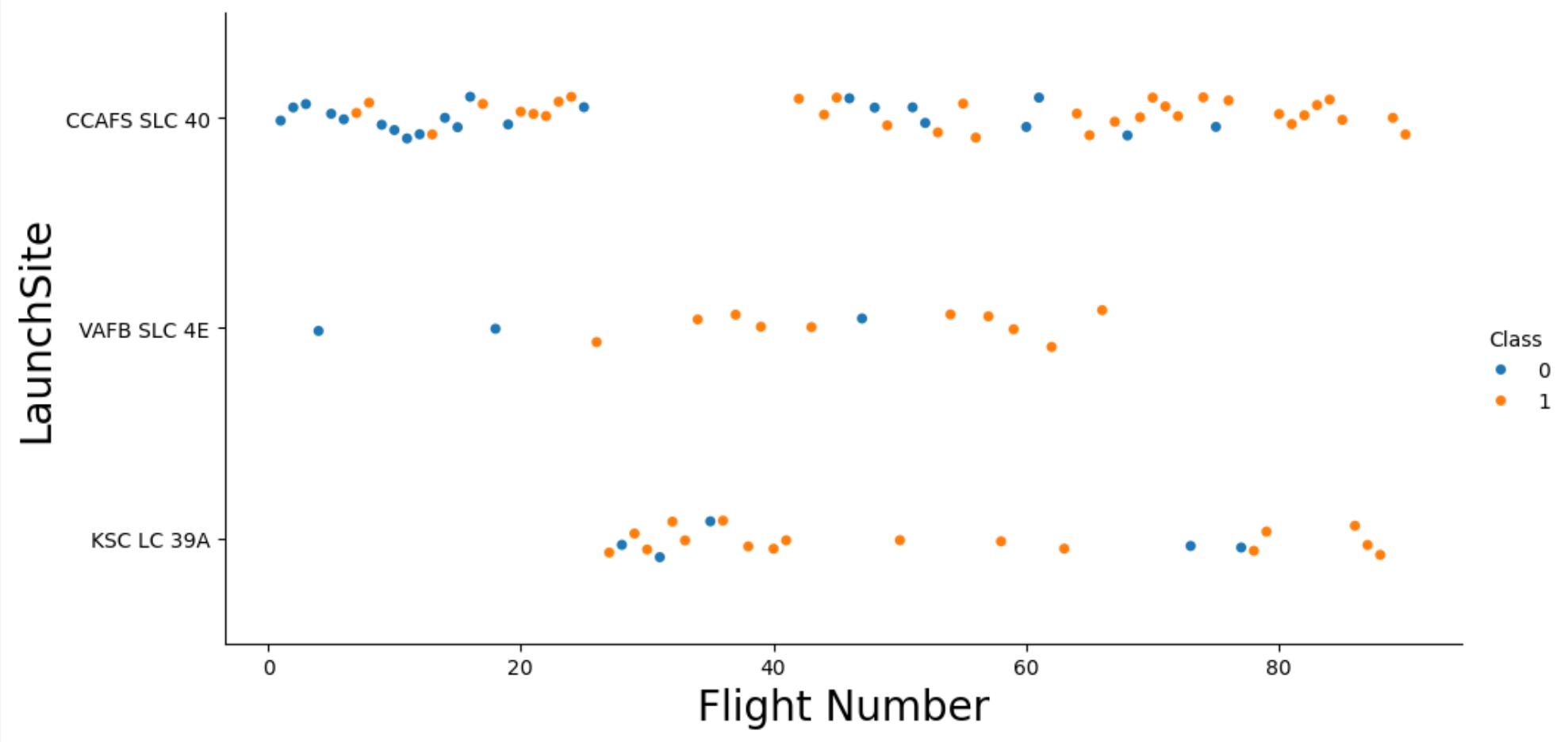
- The Logistic Regression, SVM, and KNN models scored as the most accurate predictive classifiers for the dataset at 83.33%.

- Low weighted payloads performed better than heavier weighted payloads.
- Success rate for launches increased over time since launches were recorded.

- KSC LC 39A achieved the most successful launches.
- Orbits GEO, HEO, SSO ES L1 recorded the best success rate.

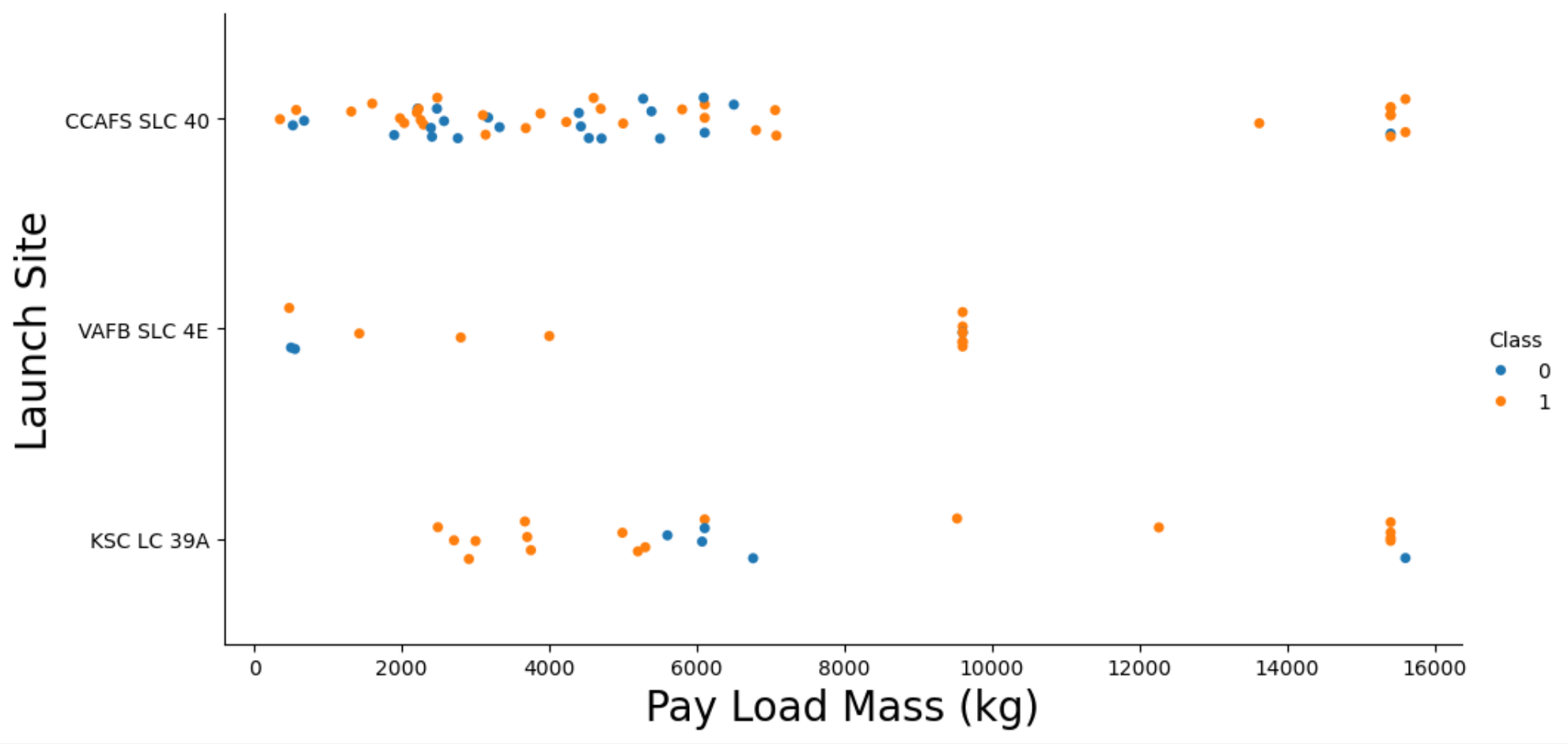

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



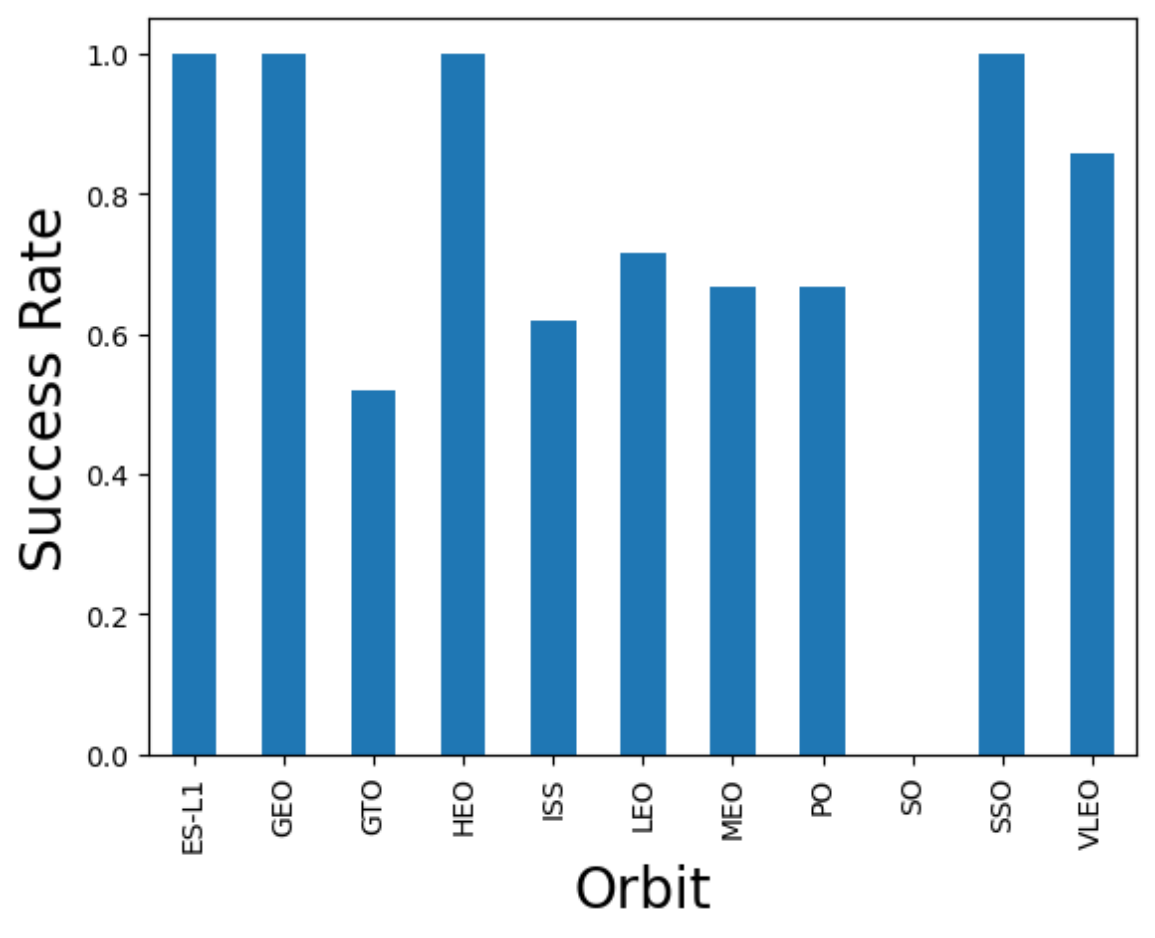
- Launches from CCAFS SLC 40 were higher than launches from other sites.

Payload vs. Launch Site



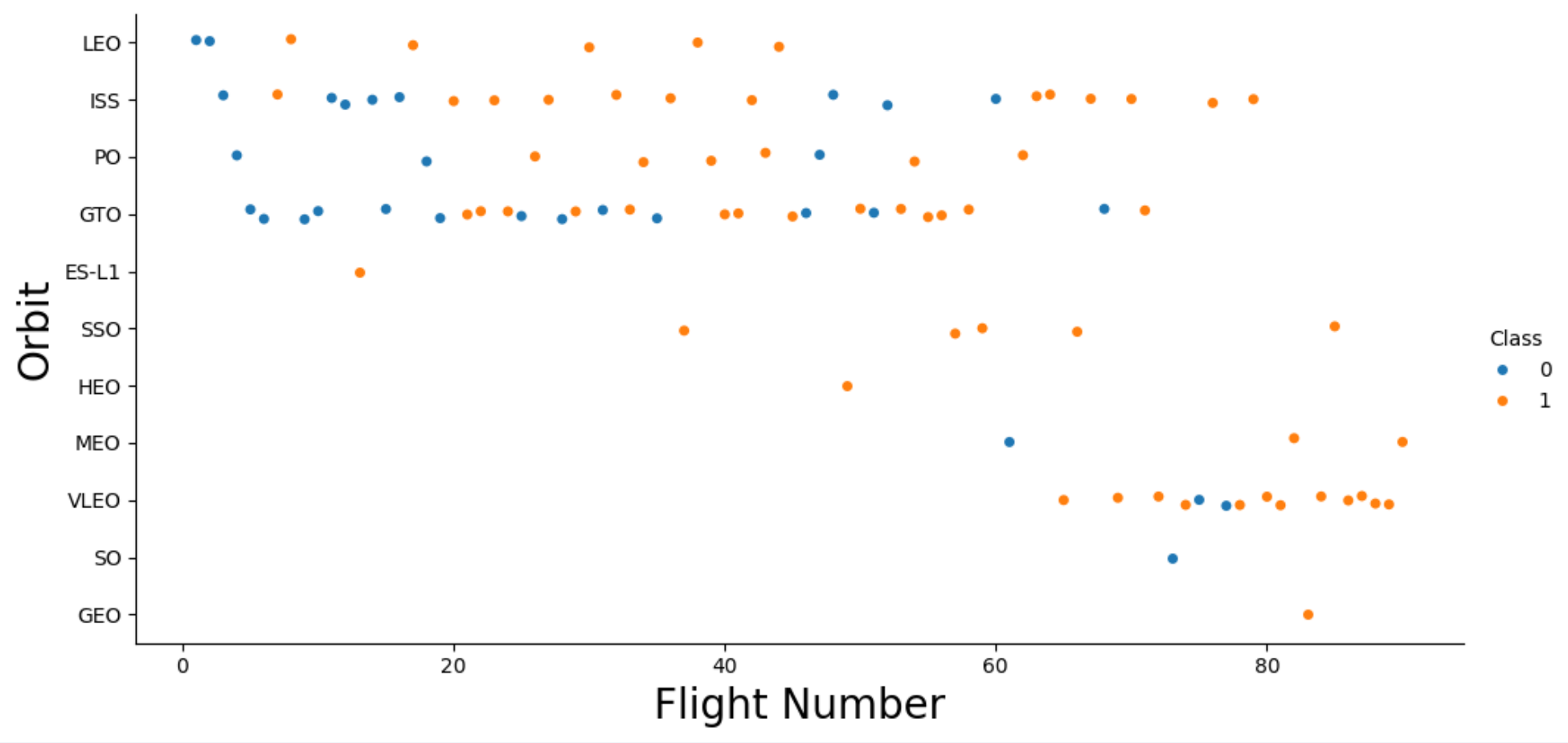
- Majority of Pay Loads with lower mass were launched from CCAFS SLC 40.

Success Rate vs. Orbit Type



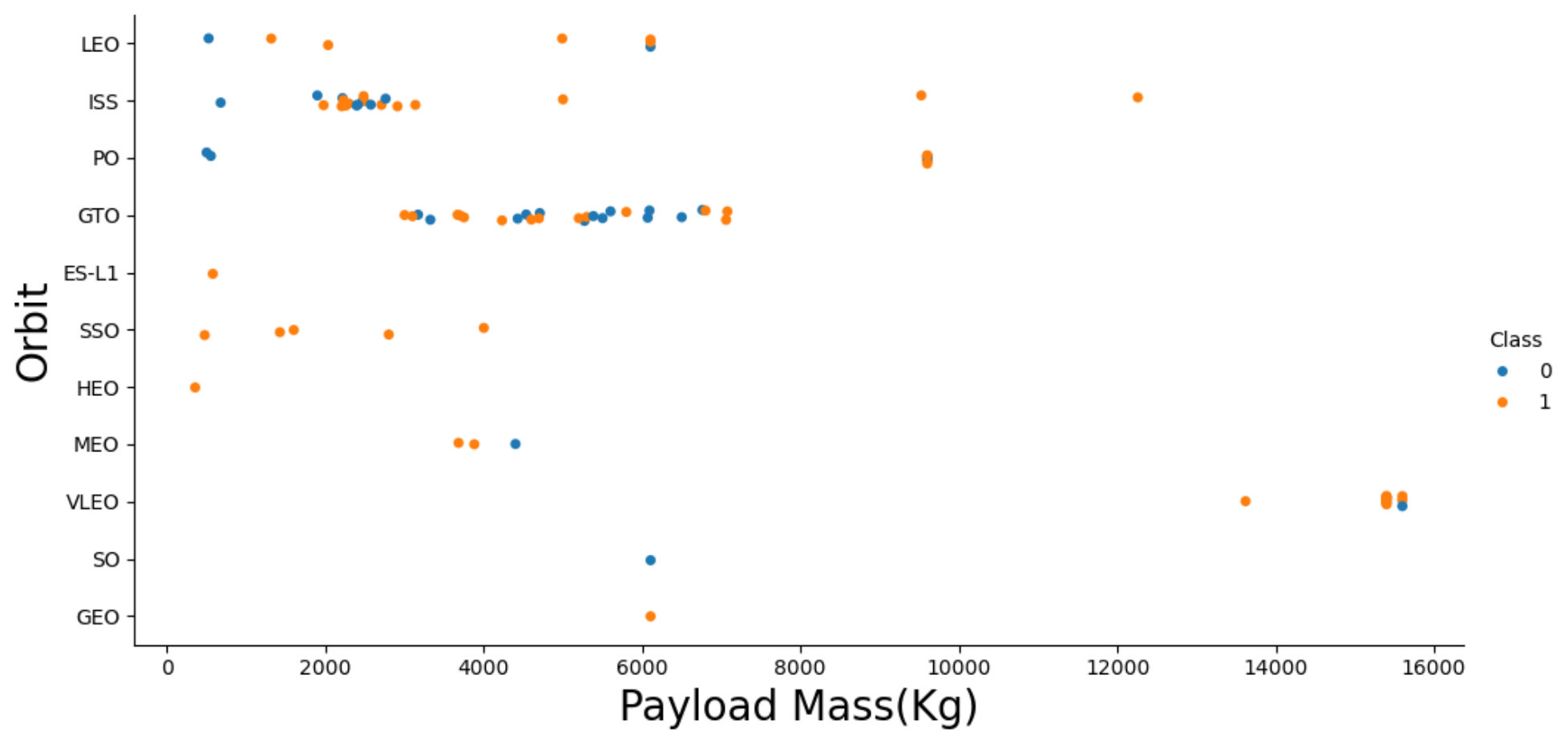
- Orbit types ES-L1, GEO, HEO, SSO had highest success rates.

Flight Number vs. Orbit Type



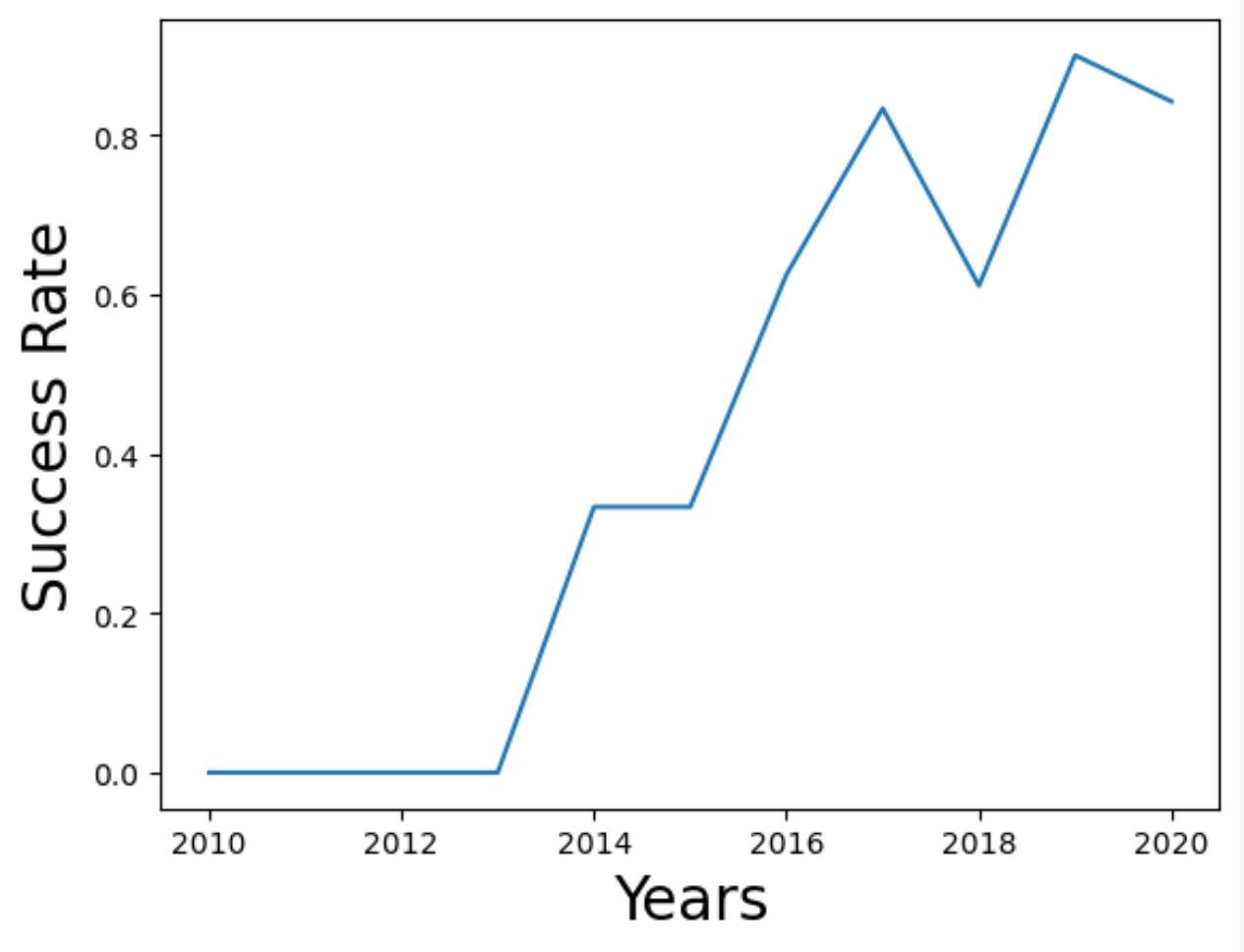
- Observe the trend that VLEO was the most frequent Orbit in recent launches.

Payload vs. Orbit Type



- Strong correlation for ISS at Payload Mass 2000 – 4000kg, GTO dispersed.

Launch Success Yearly Trend



- Launch success rate first lifted from 0% in 2013
- Stabilized at ~33% success rate in 2014 to 2015
- Climbed from 2015 and reached ~80% in 2017

All Launch Site Names

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

```
%sql Select distinct Launch_Site from SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string '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_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE Customer='NASA (CRS)'
```

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

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE BOOSTER_VERSION LIKE 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

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)'
```

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

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

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 SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND \
PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 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

List the total number of successful and failure mission outcomes

```
%sql SELECT DISTINCT(MISSION_OUTCOME),COUNT(*) from SPACEXTBL GROUP BY MISSION_OUTCOME
```

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

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

Boosters Carried Maximum Payload

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

```
%sql SELECT Booster_Version from SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

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

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

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in 2015.

```
%sql SELECT substr(Date,6,2) as month, DATE, Booster_Version, LAUNCH_SITE, Landing_Outcome FROM SPACEXTBL \
where Landing_Outcome = 'Failure (drone ship)' and substr(Date,0,5)='2015';
```

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

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

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

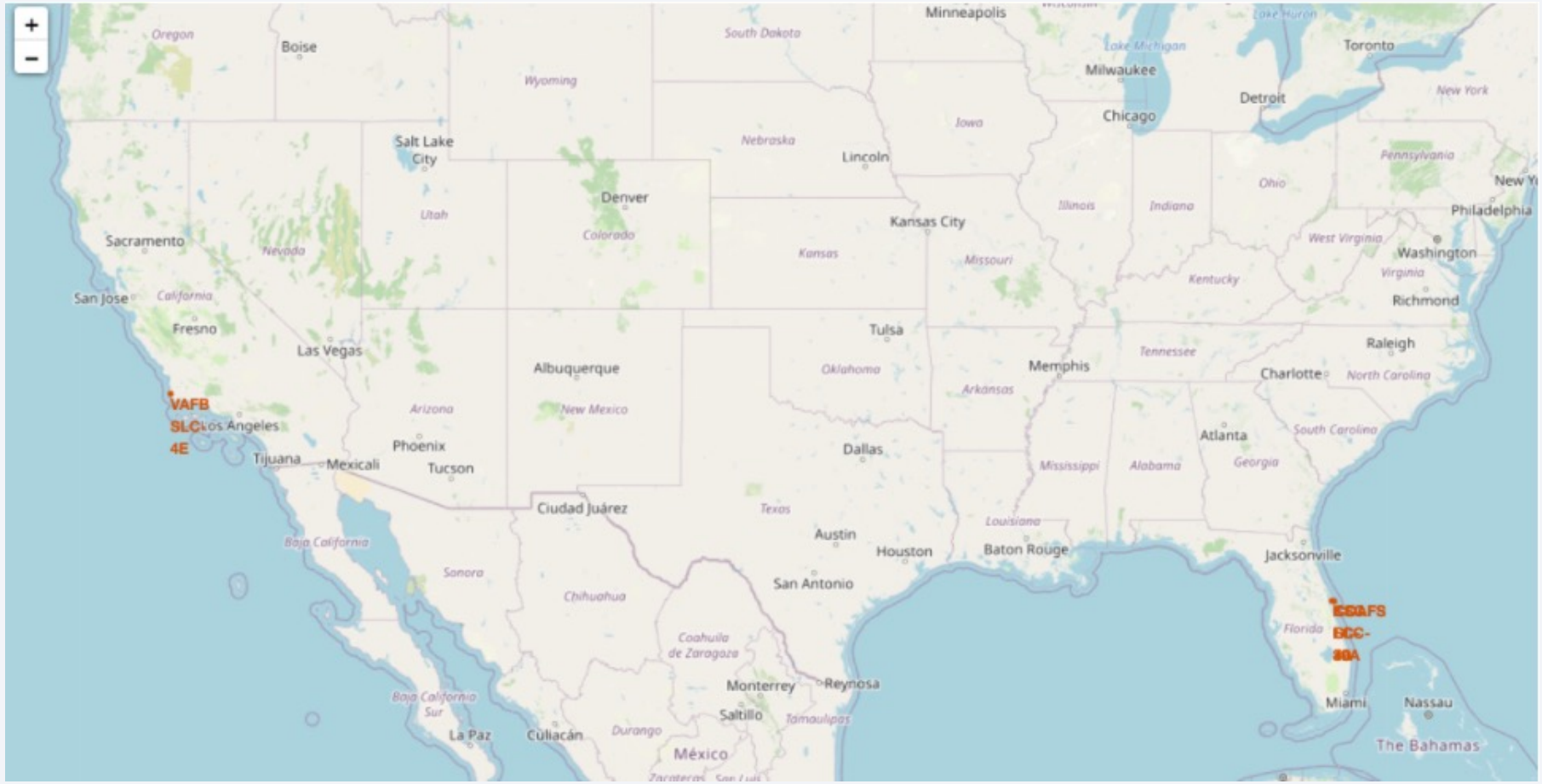
```
%sql SELECT DISTINCT Landing_Outcome, COUNT(*) as ct from SPACEXTBL \
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY ct DESC;
```

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

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

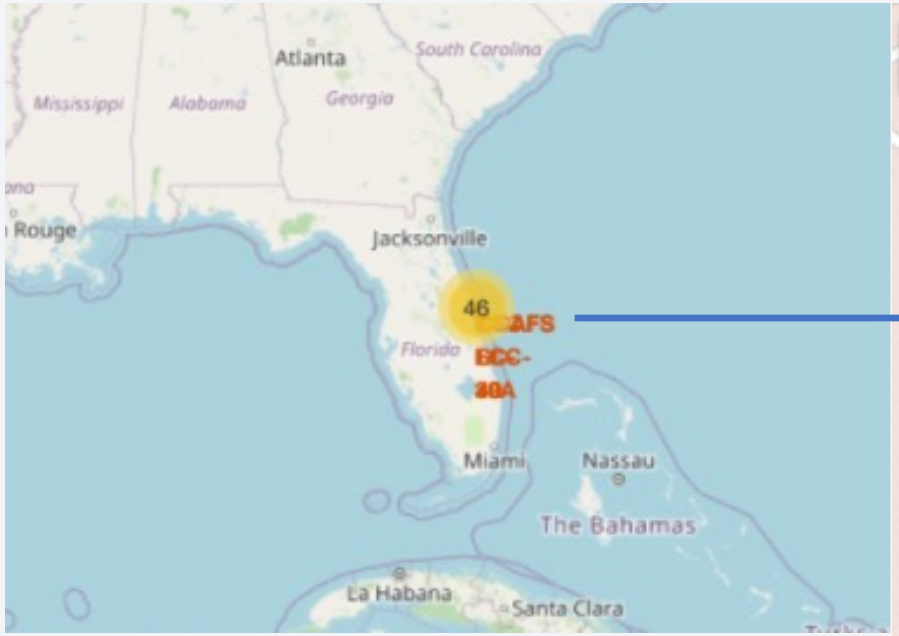

Section 3
Launch Sites
Proximities Analysis

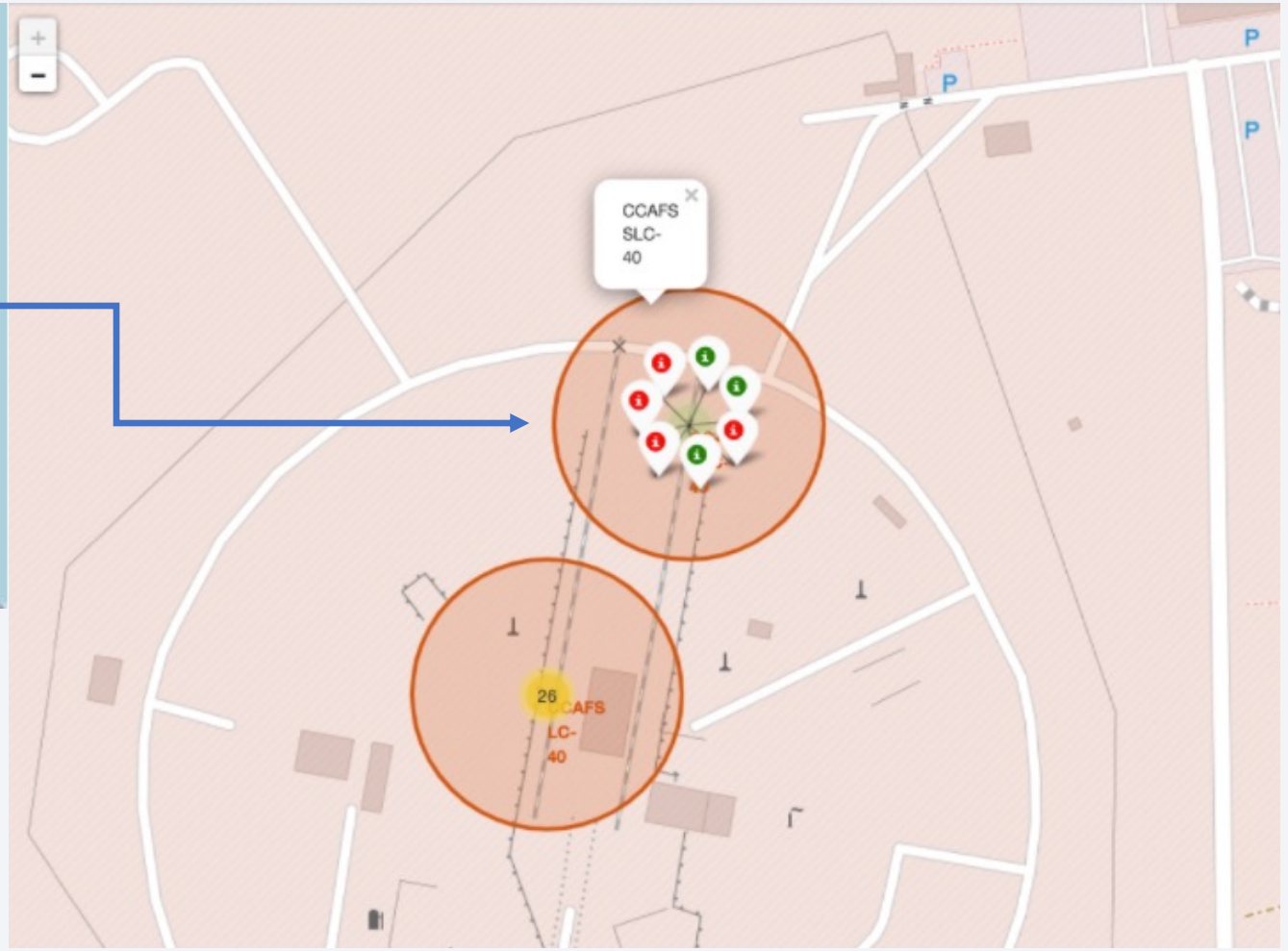
All Launch Sites in Folium



- All launch sites are in proximity to the equator, and a coastline.

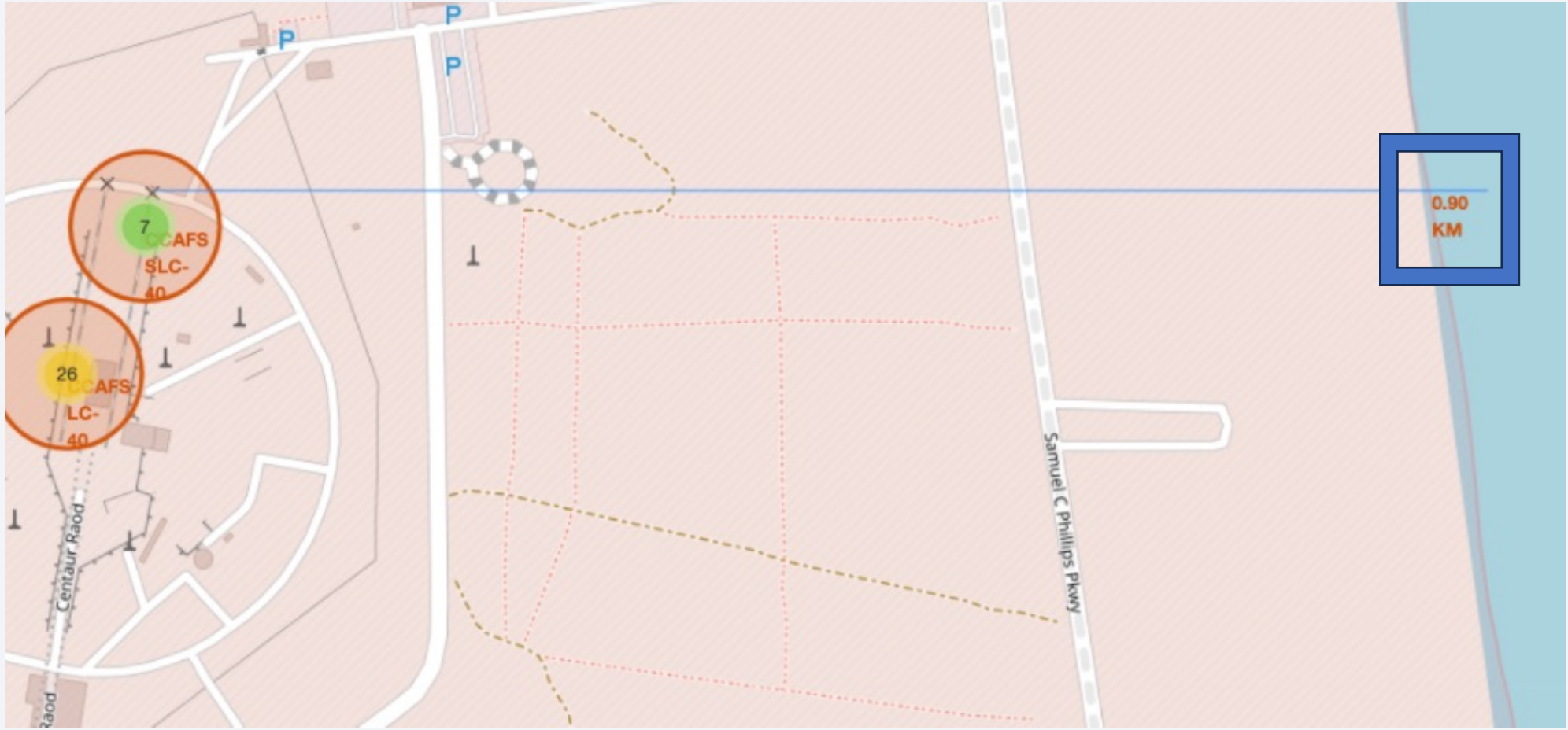
Successful & Failed Launch Sites



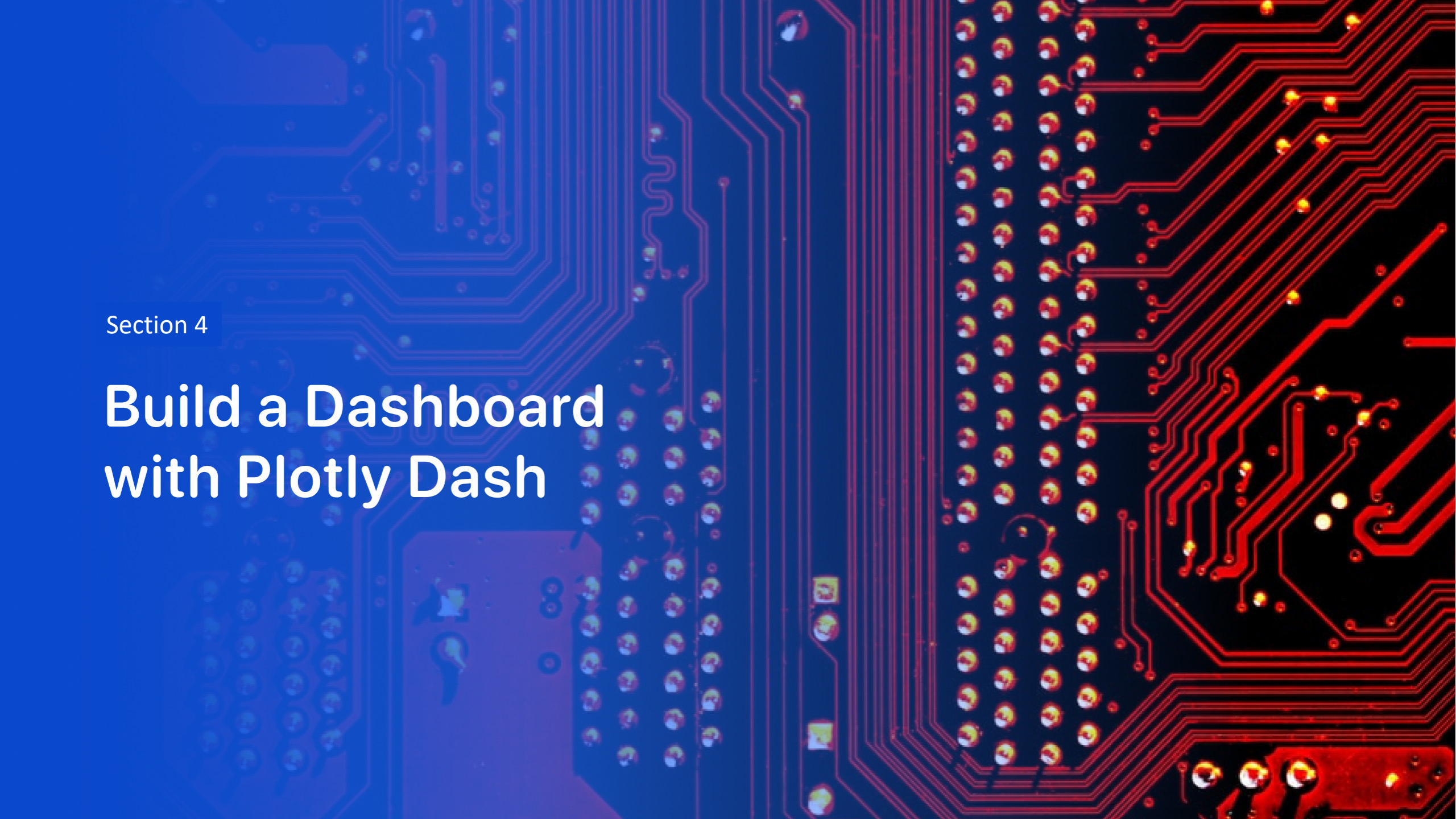


- Zoom in on launch sites provides coloured markers in clusters to help easily identify sites with high success rates

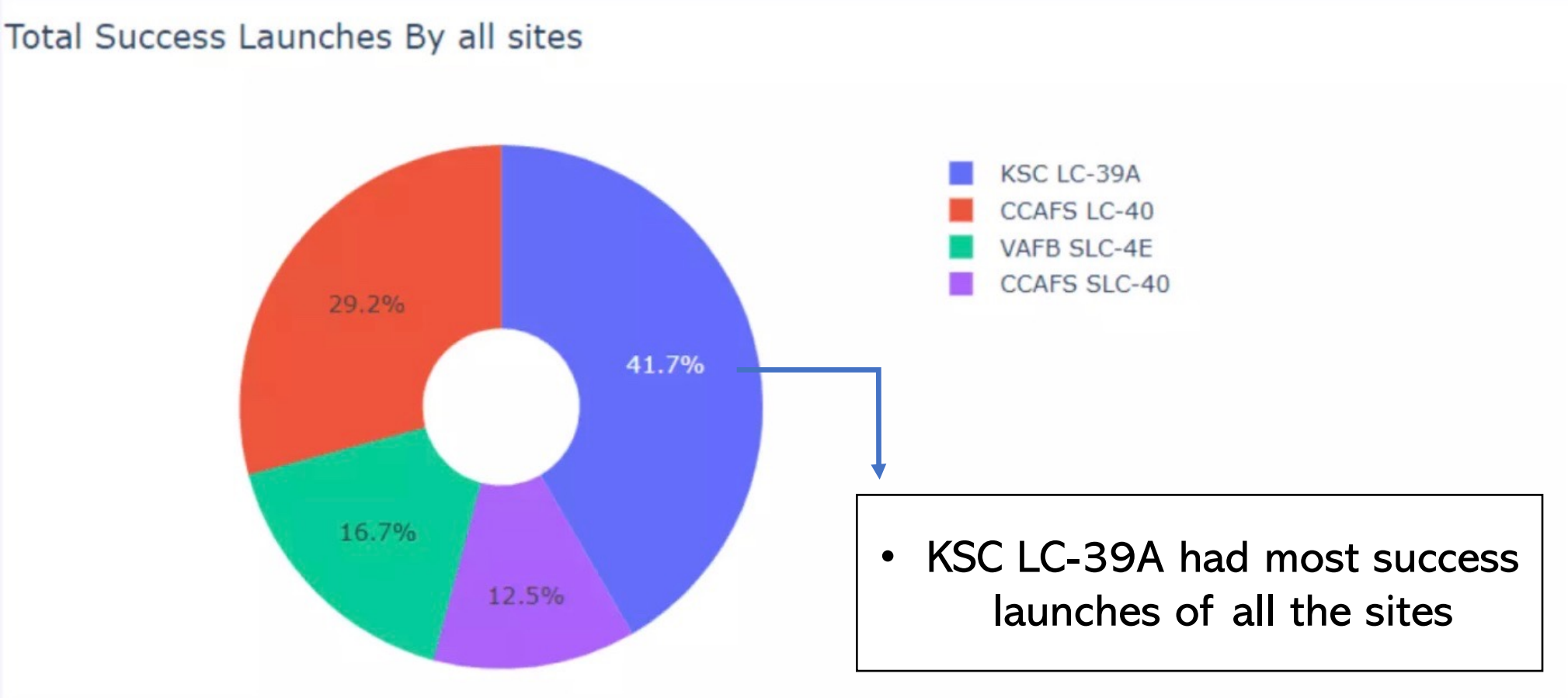
Interactive Proximities to Launch Sites



- Interactive distance lines confirm proximity to railway, highway, and coastline.

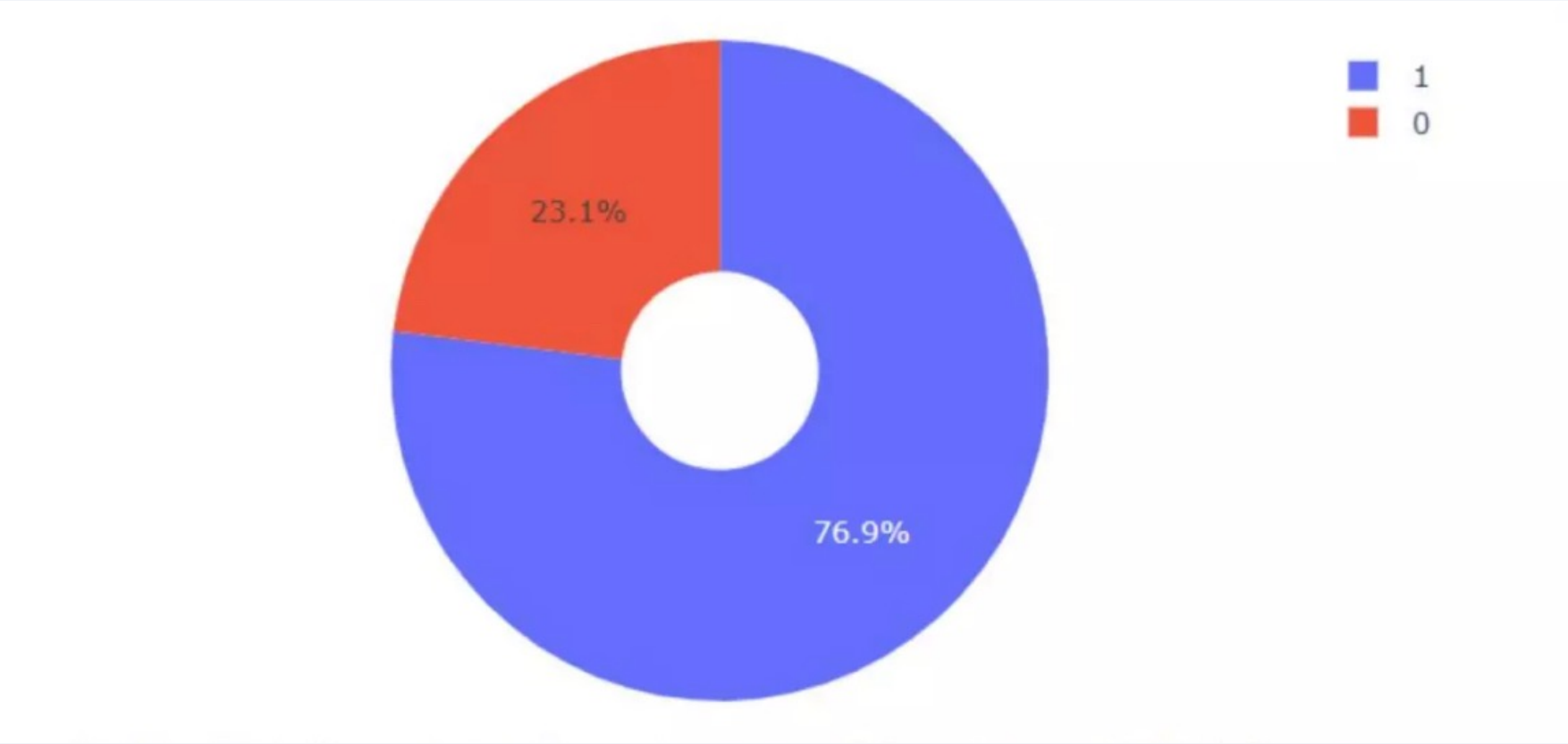

Section 4
Build a Dashboard with Plotly Dash

Success Launches by All Sites



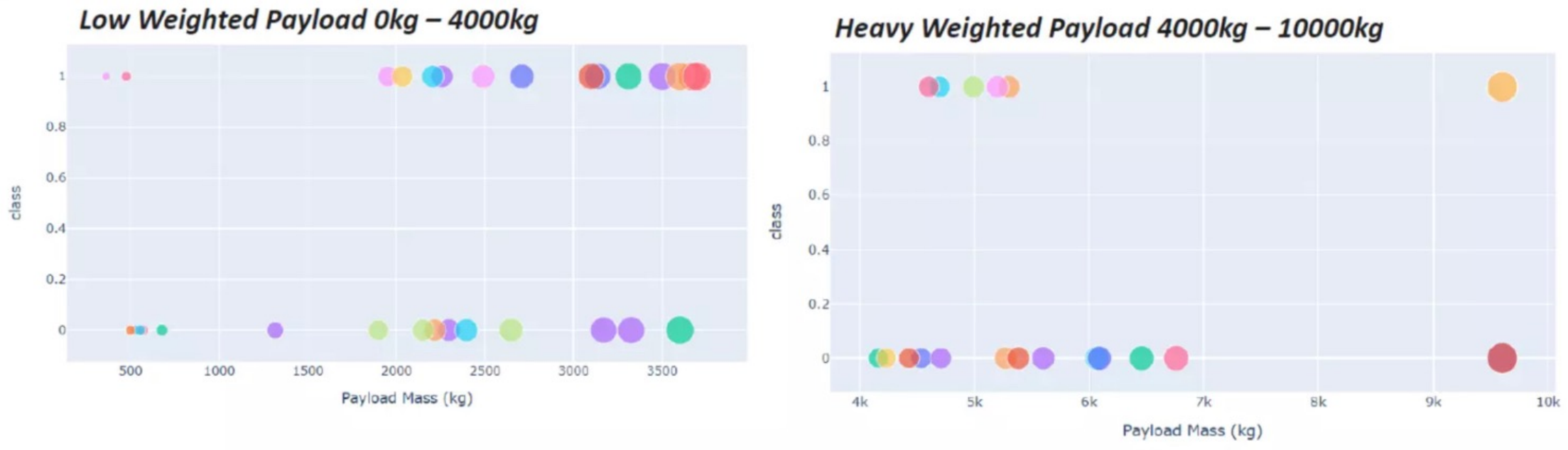
- KSC LC-39A had most success launches of all the sites

Success Rate by Site - KSC LC-39A



- KSC LC-39A recorded a 76.9% success rate, and failure of 23.1%

Payload vs. Launch Outcome

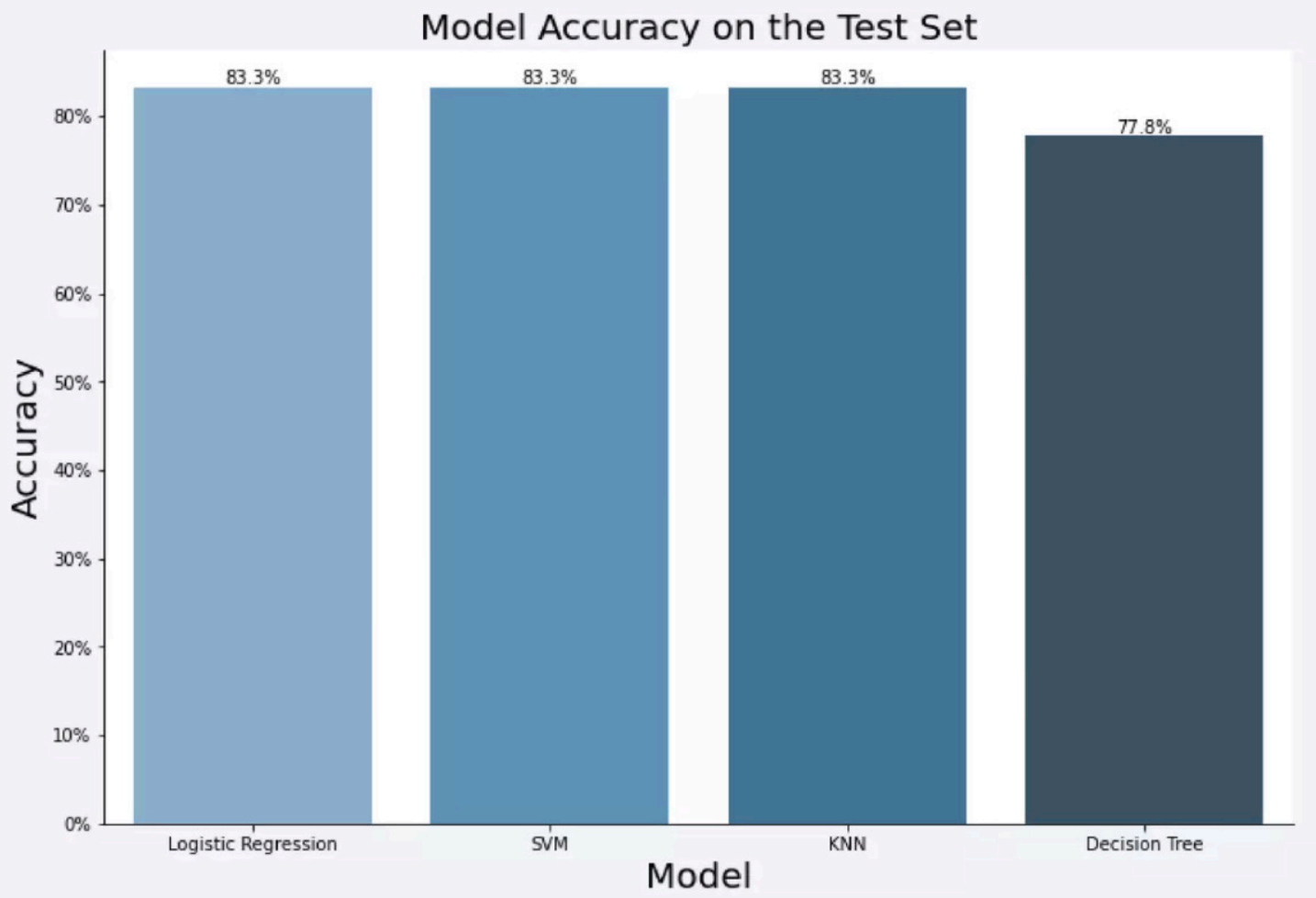


- Higher success rates for Low Weighted Payloads compared to Heavy Weighted

Section 5

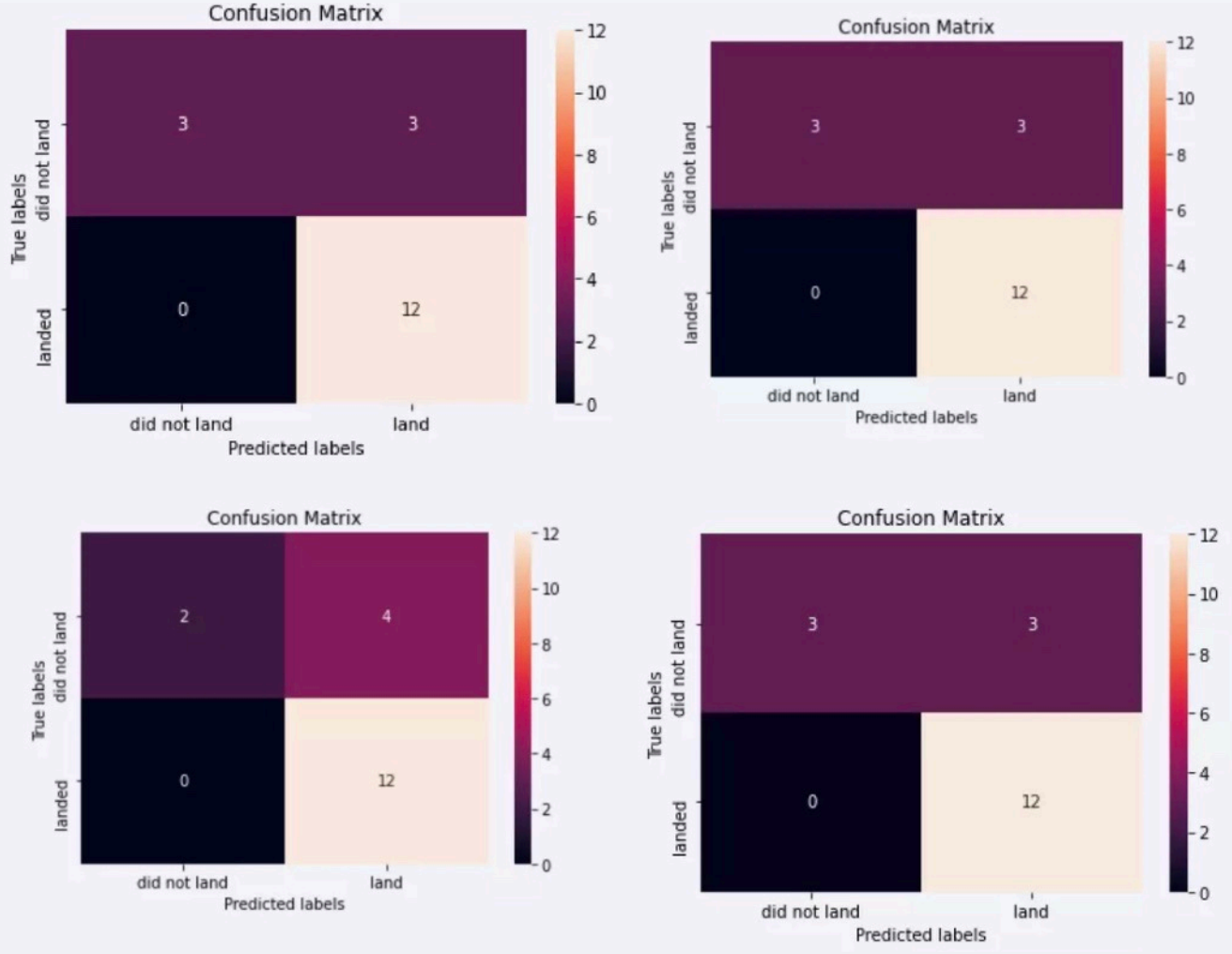
Predictive Analysis (Classification)

Classification Accuracy



- The Logistic Regression, SVM, and KNN models scored as the most accurate predictive classifiers for the dataset at 83.33%
- The Decision Tree model scored lowest accuracy for the dataset at 77.8%

Confusion Matrix



- The Logistic Regression, SVM, and KNN confusion matrix all scored the same figures for True Labels and Predicted Labels due to 83.33% model accuracy.
- The Decision Tree confusion matrix displayed (bottom left) two different upper figures due to 77.8% model accuracy.

Conclusions

- The Logistic Regression, SVM, and KNN models scored as the most accurate predictive classifiers for the dataset at 83.33%.

- Low weighted payloads performed better than heavier weighted payloads.
- Success rate for launches increased over time since launches were recorded.

- KSC LC 39A achieved the most successful launches.
- Orbits GEO, HEO, SSO ES L1 recorded the best success rate.

Appendix

Thank you!