IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Martin Byrne
Feb 2022

Outline

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

Executive Summary

Summary of methodologies

Data Collection through API

Data Collection for Web Scraping

Data Wrangling

Exploratory Data Analysis with SQL

Exploratory Data Analysis with SQL

Interactive Visual Analytics with Folium and plotly Dash

Machine Learning Prediction

Summary of all results

Exploratory Data Analysis result

Interactive analytics in screenshots

Predictive Analytics result

Introduction

Project background and context

- The commercial space age is here, companies are making space travel affordable for everyone. Virgin Galactic is providing suborbital spaceflights. Rocket Lab is a small satellite provider. Blue Origin manufactures sub-orbital and orbital reusable rockets. Perhaps the most successful is SpaceX. SpaceX's accomplishments include: Sending spacecraft to the International Space Station. Starlink, a satellite internet constellation providing satellite Internet access. Sending manned missions to Space. One reason SpaceX can do this is the rocket launches are relatively inexpensive. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upwards of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This stage is quite large and expensive. Unlike other rocket providers, SpaceX's Falcon 9 Can recover the first stage.

Problems you want to find answers

- What influences if the rocket will land successfully?
- The effect each relationship with certain rocket variables will impact in determining the success rate of a successful landing
- What conditions does Space X have to achieveto get the best results and ensure the best rocket success landing rate

Section 1

Methodology

Methodology

Executive Summary

Data collection methodology:

- Data was collected using SpaceX API and web scraping from Wikipedia

Perform data wrangling

- One-hot encoding was applied to categorical features

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification models

- How to build, tune, evaluate classification models

Data Collection

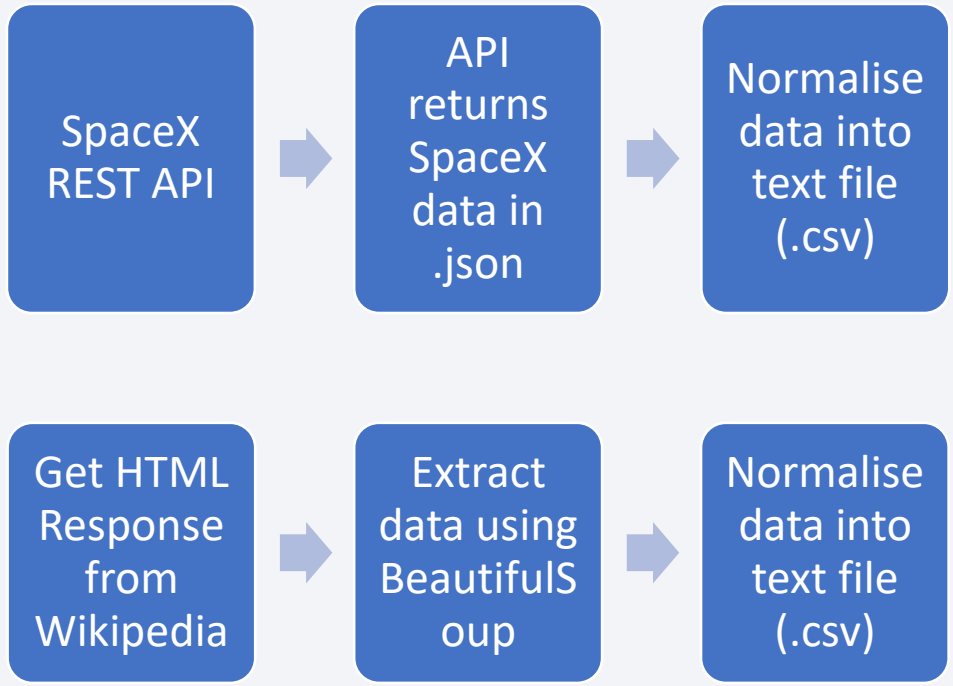
The datasets were collected by:

SpaceX REST API

- This API provides data about launches (e.g. rocket used, payload delivered, launch specifications, landing specifications, landing outcome)
- The response content was decoded as a Json using .json() function call and turned into a pandas dataframe using .json_normalize()
- Data was cleaned, checked for missing values and missing values inserted where required

SpaceX Webscraping

- Webscraping from Wikipedia for Falcon 9 launch records using BeautifulSoup
- extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis



Data Collection – SpaceX API

1. Get Response from API

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

2. Convert response to .json file

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

3. Apply custom functions

```
[16]: # Call getBoosterVersion
getBoosterVersion(data)
```

```
[18]: # Call getLaunchSite
getLaunchSite(data)
```

```
[19]: # Call getPayloadData
getPayloadData(data)
```

```
[20]: # Call getCoreData
getCoreData(data)
```

4. Assign list to dictionary then dataframe

```
[22]: # Create a data from launch_dict
df = pd.DataFrame.from_dict(launch_dict)
```

5. Deal with missing values

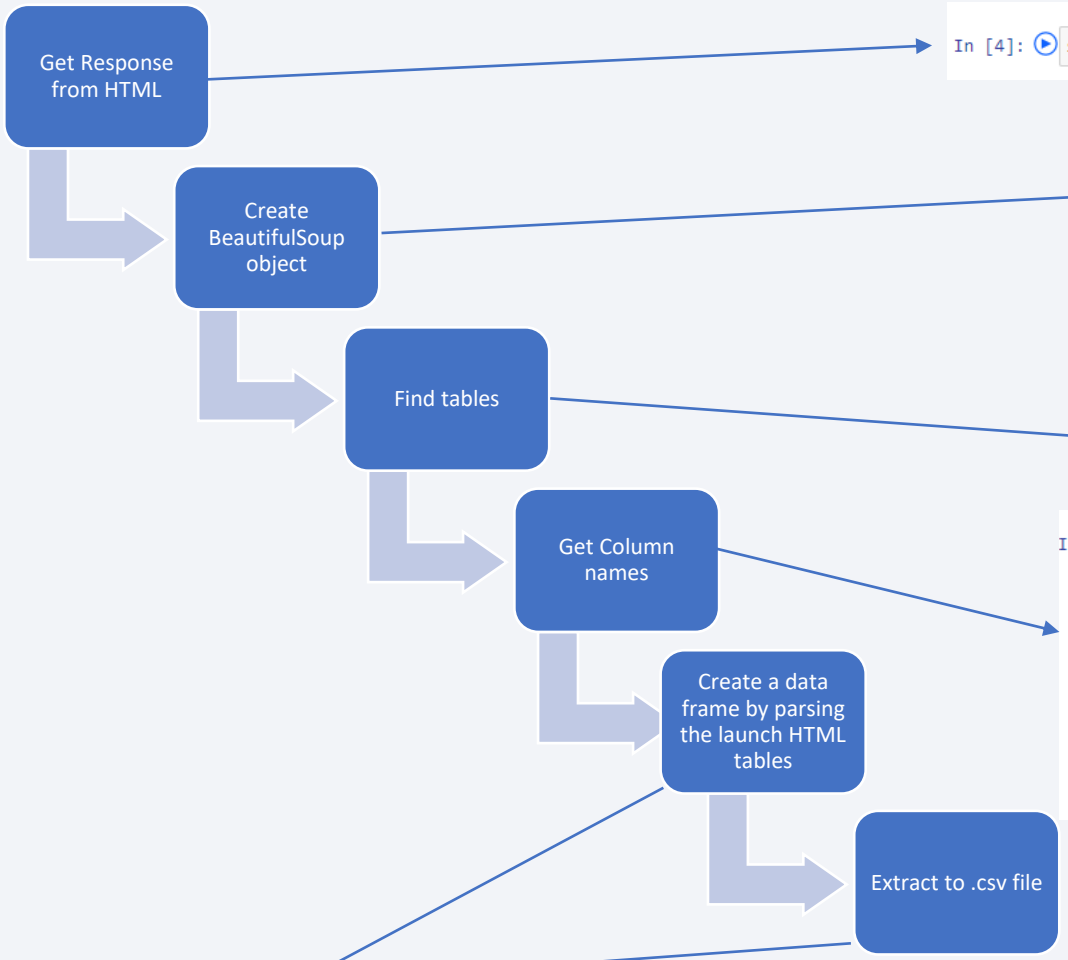
```
[29]: # Calculate the mean value of PayloadMass column
PayloadMass_Mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].fillna(PayloadMass_Mean)
```

6. Export to .csv

```
[29]: data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/01%20SpaceX_Data_Collection_API.ipynb

Data Collection - Scraping




```
In [4]: static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

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

Out[5]: 200
```

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

```
In [6]: # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
        soup = BeautifulSoup(page.text, 'html.parser')
```

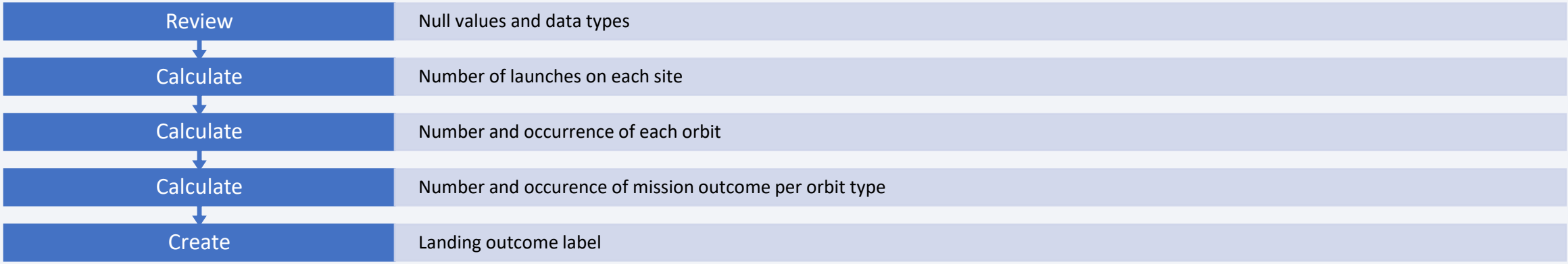
```
In [8]: # Use the find_all function in the BeautifulSoup object, with element type `table`
        # Assign the result to a list called `html_tables`
        html_tables = soup.find_all('table')
```

```
In [11]: # Apply find_all() function with `th` element on first_launch_table
         # Iterate each th element and apply the provided extract_column_from_header() to get a column name
         # Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
         column_names = []
         temp = soup.find_all('th')
         for x in range(len(temp)):
             try:
                 name = extract_column_from_header(temp[x])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/01%20SpaceX_Data_Collection_API.ipynb

Data Wrangling

In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad.True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship.



Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/03%20SpaceX_Data_Wrangling.ipynb

EDA with Data Visualization

- Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation . Scatter plots usually consist of a large body of data.
  - Flight Number vs Payload Mass, Flight Number vs Launch Site, Payload vs Launch Site, Orbit vs Flight Number, Payload vs Orbit Type, Orbit vs Payload Mass
- A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.
  - Mean vs Orbit
- Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded
  - Success Rate vs Year

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/05%20EDA%20with%20SQL%20Visualisation.ipynb

EDA with SQL

- Performed SQL queries to gather information about the dataset
  - Displaying the names of the unique launch sites in the space mission
  - Displaying 5 records where launch sites begin with the string 'KSC’
  - Displaying the total payload mass carried by boosters launched by NASA (CRS)
  - Displaying average payload mass carried by booster version F9 v1.1
  - Listing the date where the successful landing outcome in drone ship was achieved.
  - Listing the names of the boosters which have success in ground pad and have payload mass greater than 4000 but less than 6000
  - Listing the total number of successful and failure mission outcomes
  - Listing the names of the booster_versions which have carried the maximum payload mass.
  - Listing the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
  - Ranking the count of successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/04%20jupyter-labs-eda-sql-coursera.ipynb

Build an Interactive Map with Folium

- To visualize the Launch Data into an interactive map. We took the Latitude and Longitude Coordinates at each launch site and added a Circle Marker around each launch site with a label of the name of the launch site.
- We assigned the dataframe launch_outcomes(failures, successes) to classes 0 and 1 with Green and Red markers on the map in a MarkerCluster()
- Using Haversine's formula we calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks
- Example of some trends in which the Launch Site is situated in.
  - Are launch sites in close proximity to railways? No
  - Are launch sites in close proximity to highways? No
  - Are launch sites in close proximity to coastline? Yes
  - Do launch sites keep certain distance away from cities? Yes

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/06%20Interactive%20Visual%20Analytics%20with%20Folium%20lab.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash
- Graphs
  - ➢ Pie Chart showing the total launches by a certain site/all sites
    - display relative proportions of multiple classes of data.
    - size of the circle can be made proportional to the total quantity it represents.
  - ➢ Scatter Graph showing the relationship with Outcome and Payload Mass (Kg) for the different Booster Versions
    - It shows the relationship between two variables.
    - It is the best method to show you a non-linear pattern.
    - The range of data flow, i.e. maximum and minimum value, can be determined.
    - Observation and reading are straightforward.

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/07%20spacex_dash_app.ipynb

Predictive Analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Transform Data
- Split our data into training and test data sets
- Check how many test samples we have
- Decide which type of machine learning algorithms we want to use
- Set our parameters and algorithms to GridSearchCV
- Fit our datasets into the GridSearchCV objects and train our dataset.

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithms
- Plot Confusion Matrix

IMPROVING MODEL

- Feature Engineering
- Algorithm Tuning

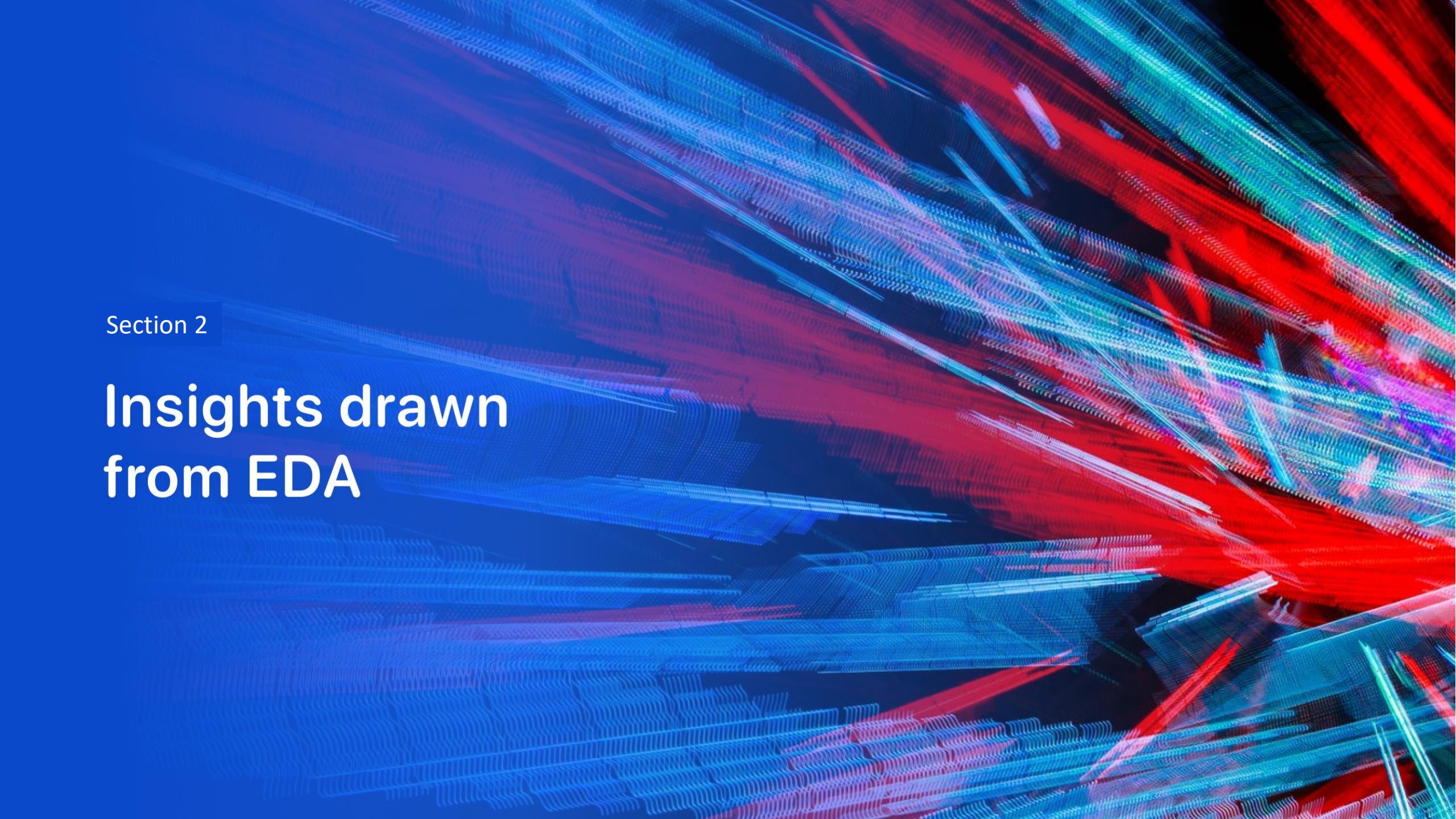
FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the best accuracy score wins the best performing model
- In the notebook there is a dictionary of algorithms with scores at the bottom of the notebook.

Link to GitHub: https://github.com/MartinB107/IBM-Data-Science-Project/blob/main/08%20Machine%20Learning%20Prediction.ipynb

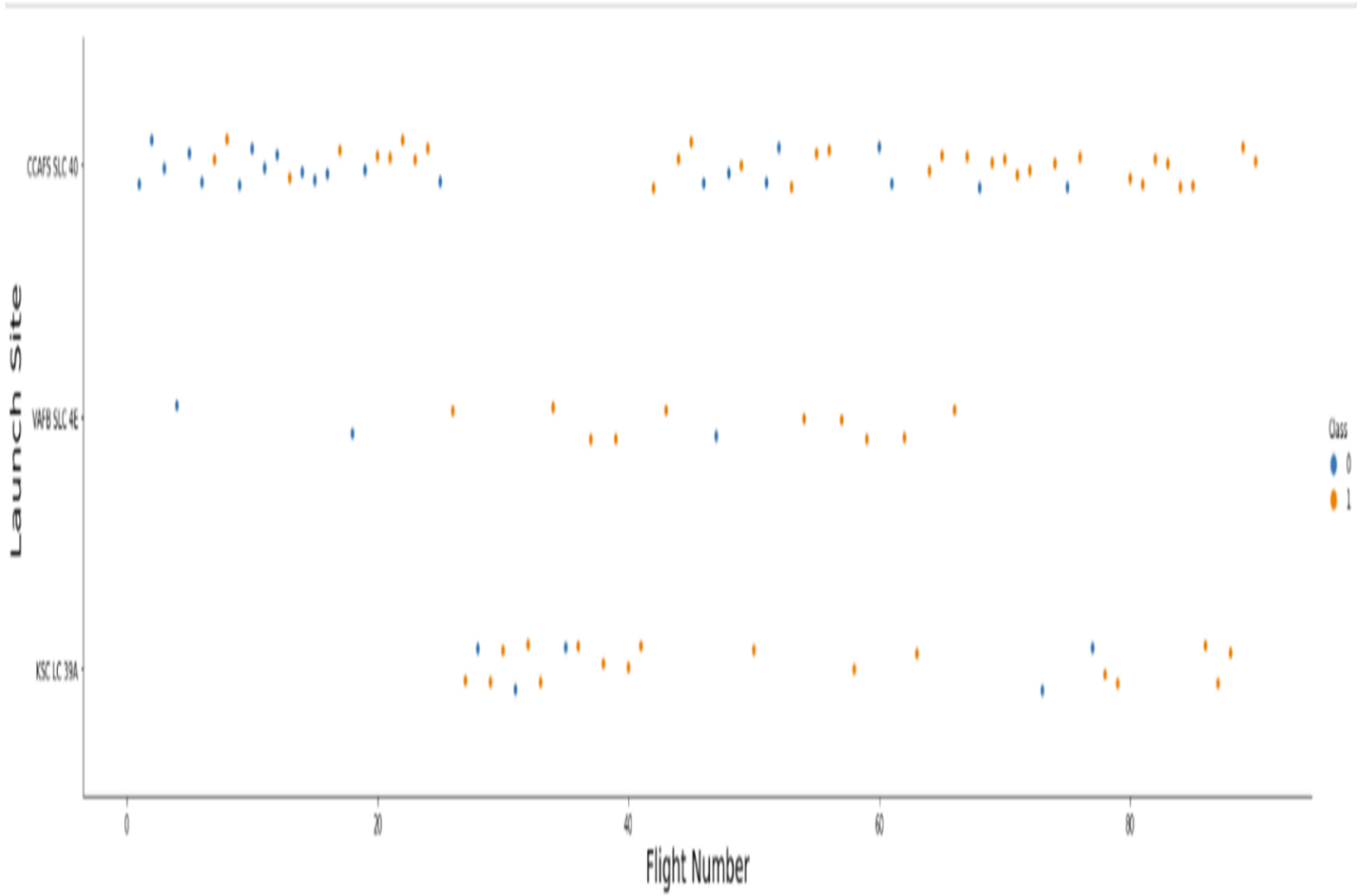
Results

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


Section 2
Insights drawn from EDA

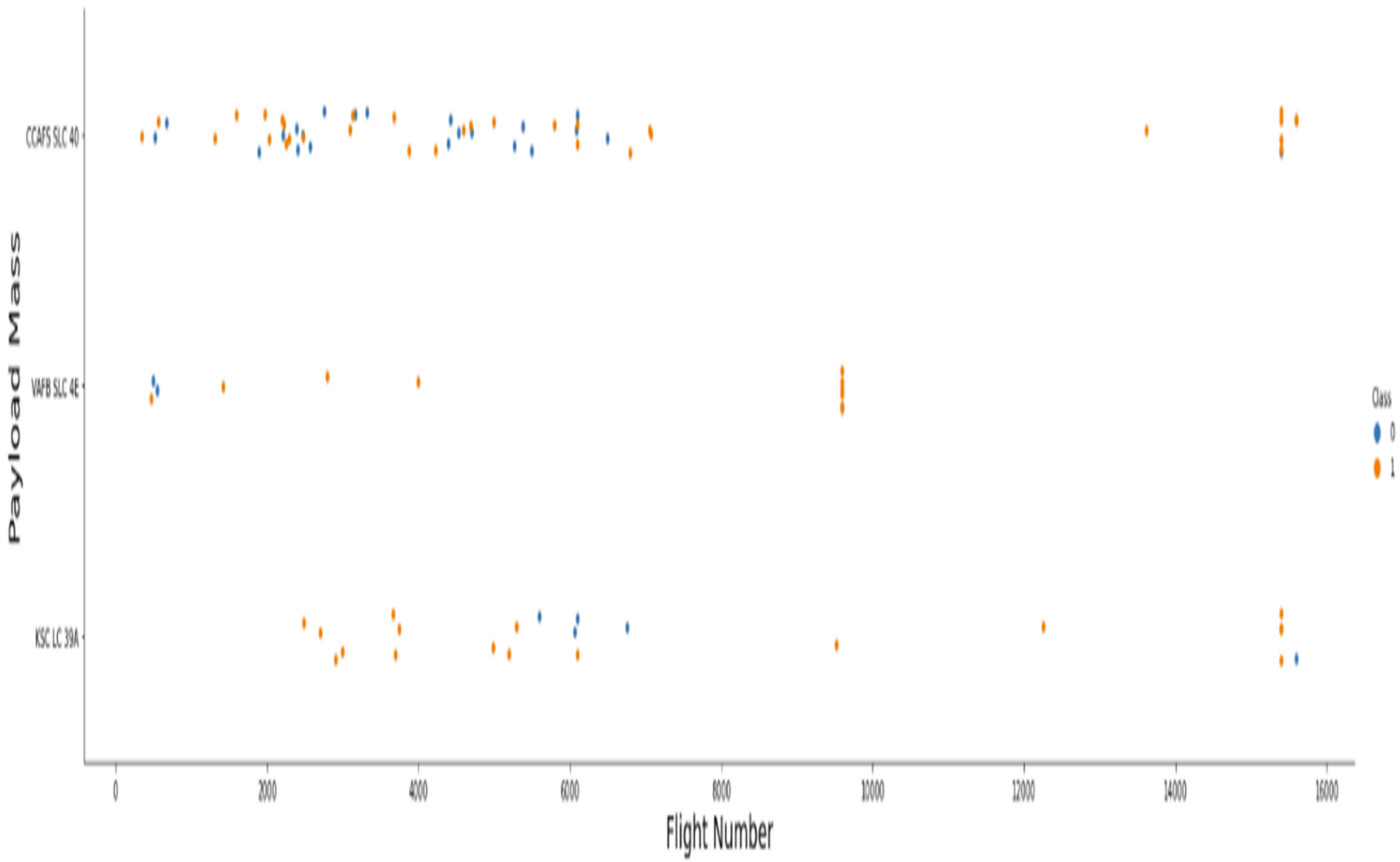
Flight Number vs. Launch Site

The more amount of flights at a launch site the greater the success rate at a launch site.



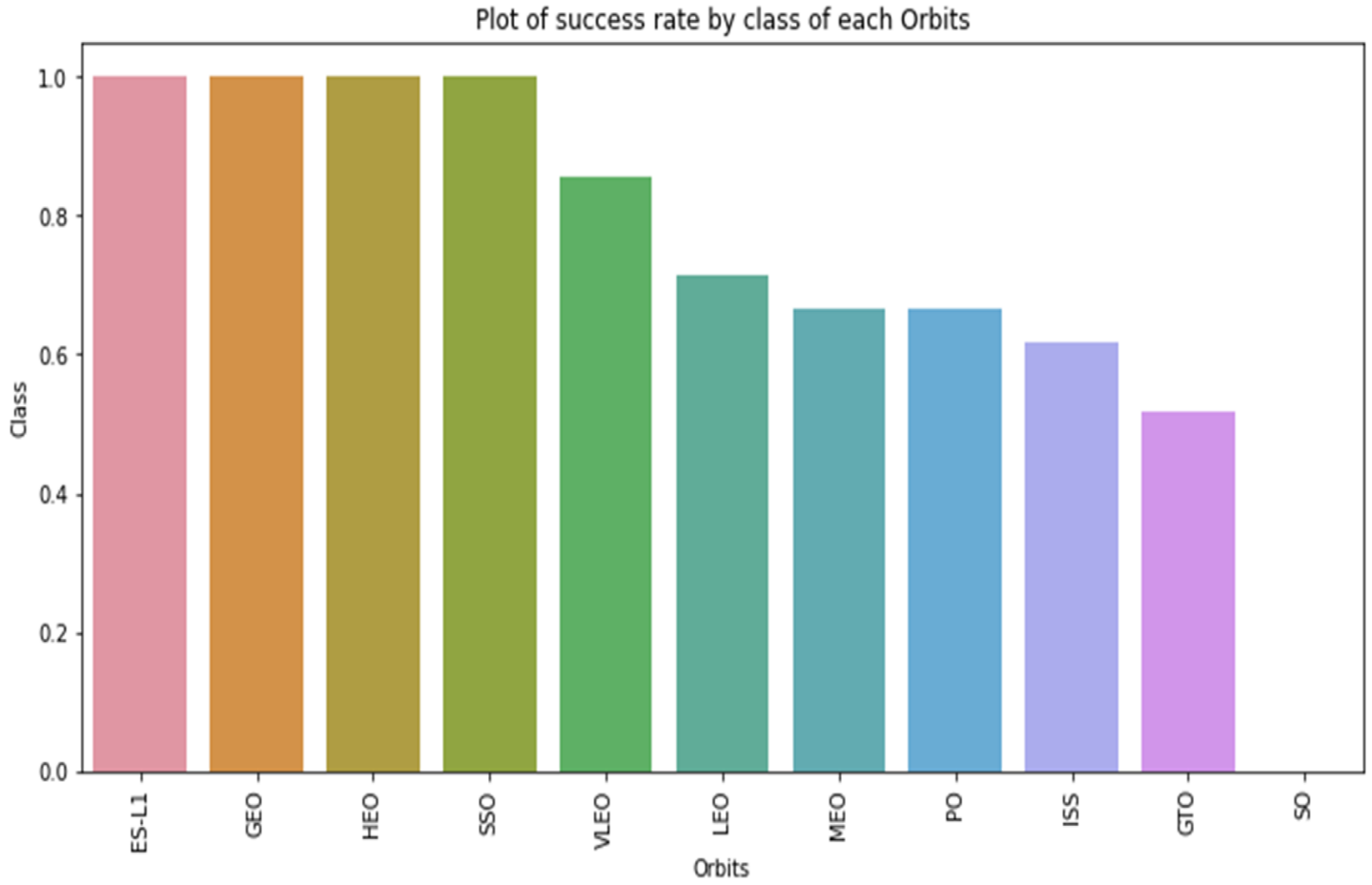
Payload vs. Launch Site

The greater the payload mass for Launch Site CCAFS SLC 40 the higher the success rate for the Rocket. There is not quite a clear pattern to be found using this visualization to make a decision if the Launch Site is dependant on Pay Load Mass for a success launch.



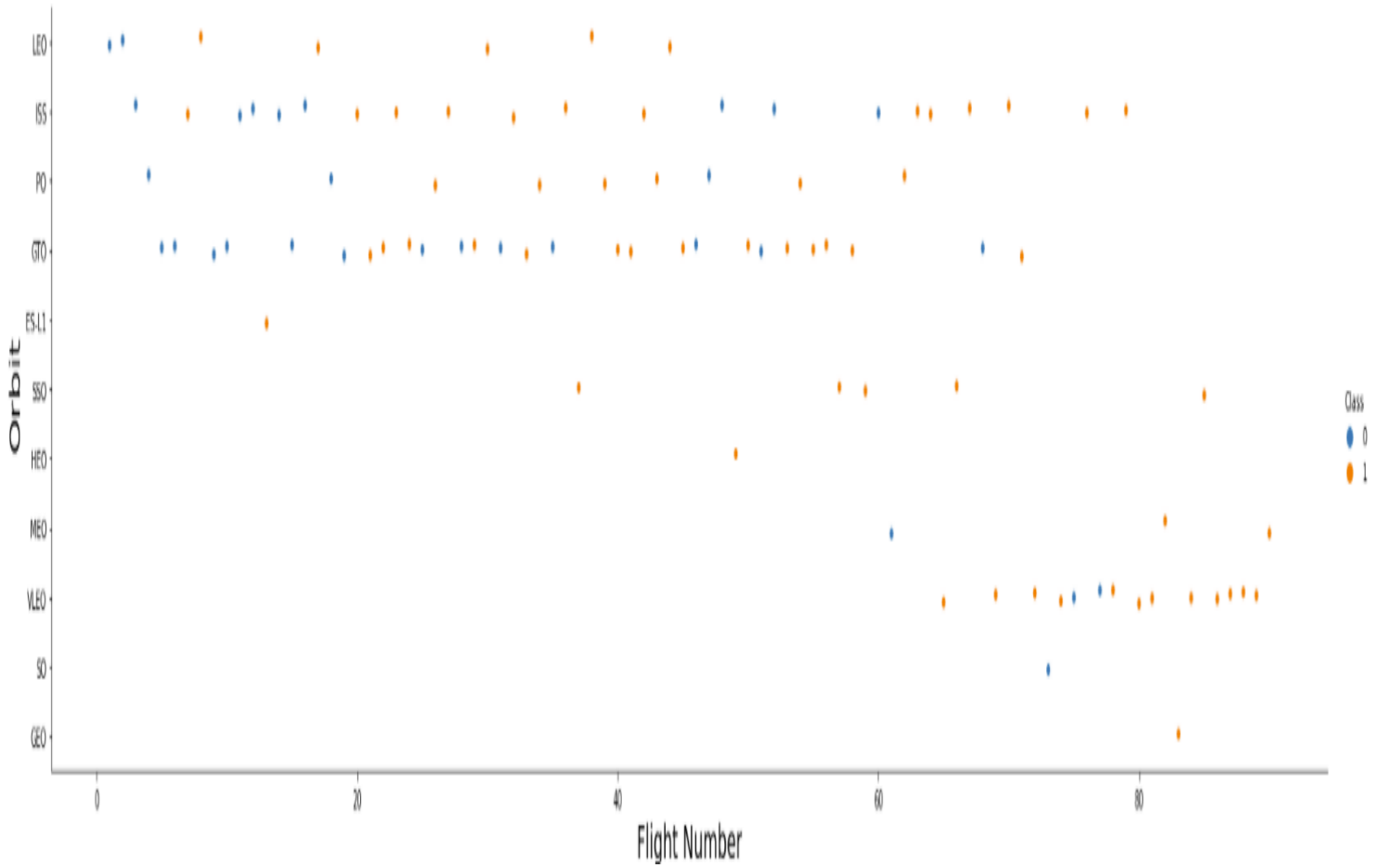
Success Rate vs. Orbit Type

Orbit ES-L1 , GEO, HEO, SSO has the best Success Rate



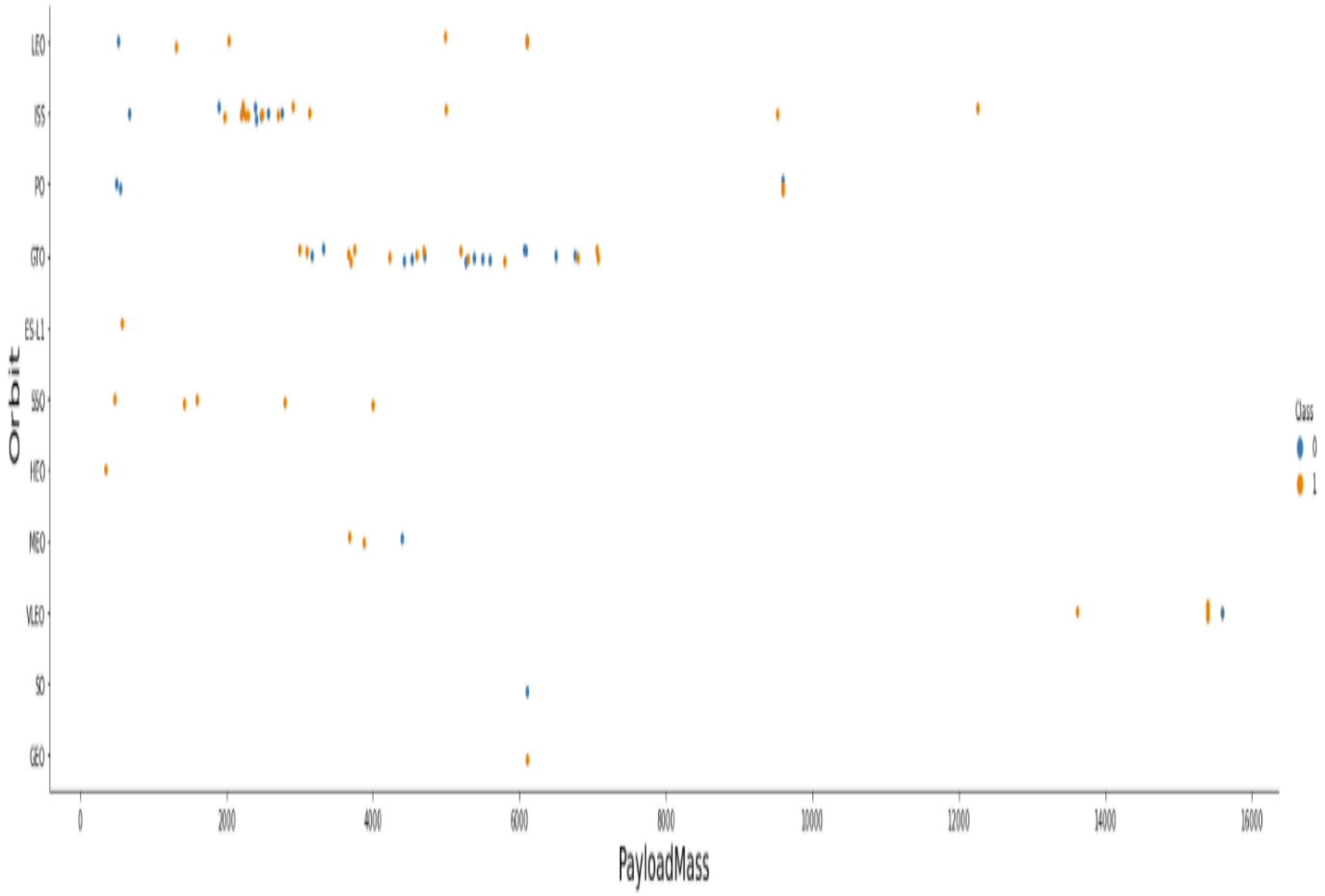
Flight Number vs. Orbit Type

You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



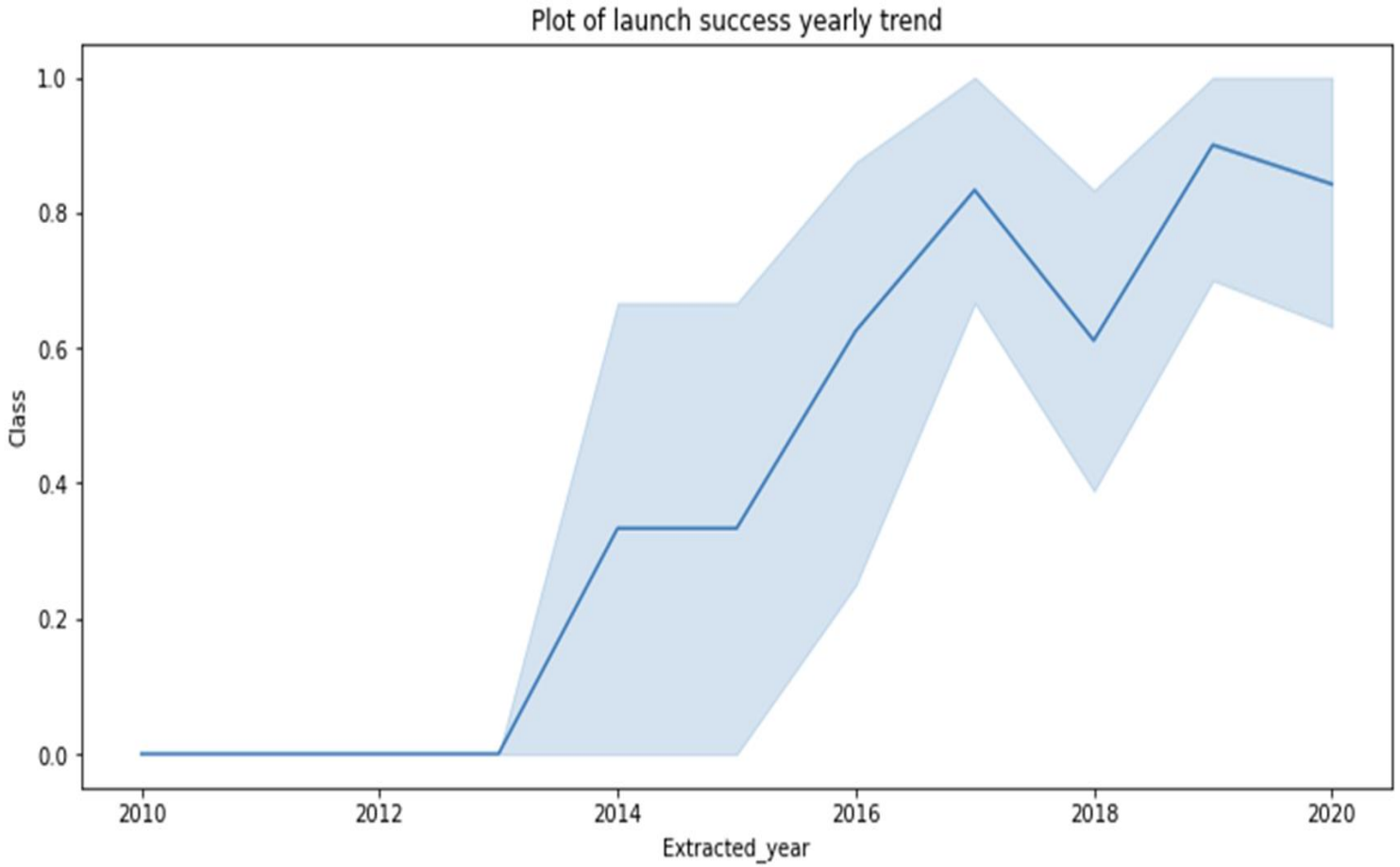
Payload vs. Orbit Type

You should observe that Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits



Launch Success Yearly Trend

The success rate since 2013 has kept increasing till 2020



All Launch Site Names

- The names of the unique launch sites using the key word DISTINCT to show only unique launch sites from the SpaceX data

```
SELECT DISTINCT launch_site FROM SPACEXTBL;
```

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

Launch Site Names Begin with 'CCA'

- First we want to return all columns from the raw data. We then identify only the launch site that starts with “CCA” followed by wildcard characters. We have only returned the 5 records.

```
SELECT * FROM SPACEXTBL

  WHERE launch_site LIKE 'CCA%'

  limit 5;
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

- The SUM function is used to add all of the payloads together. The WHERE clause only includes NASA customers

```
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE CUSTOMER = 'NASA (CRS)';
```

Result set 1
1
45596

Average Payload Mass by F9 v1.1

- The AVG function is used to calculate the average of the payloads. The WHERE clause only includes F9 v1.1 boosters

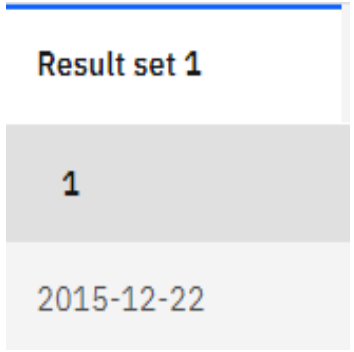
```
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL
WHERE BOOSTER_VERSION = 'F9 v1.1';
```

Result set 1
1
2928

First Successful Ground Landing Date

- The MIN function is used to call out the earliest date. The WHERE clause only includes Successful ground pad landings

```
SELECT MIN(DATE) FROM SPACEXTBL

  WHERE LANDING__OUTCOME = 'Success (ground pad)';
```



Successful Drone Ship Landing with Payload between 4000 and 6000

- Returning only booster versions. WHERE clause is used to only return Successful landings on the drone ship AND payload mass is greater than 4000 but less than 6000

```
SELECT BOOSTER_VERSION FROM SPACEXTBL
  WHERE LANDING__OUTCOME = 'Success (drone ship)'
  AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000);
```

Result set 1

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

Total Number of Successful and Failure Missi on Outcomes

- 100 successful and 1 failed mission
- Using the GROUPBY function allows us to create categories of mission outcomes We can then use the COUNT function for each occurrence of that mission outcome

```
SELECT MISSION_OUTCOME, COUNT(*) FROM SPACEXTBL

GROUP BY MISSION_OUTCOME;
```

Result set 1

MISSION_OUTCOME	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function

```
SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXTBL

WHERE PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

BOOSTER_VERSION	PAYLOAD_MASS__KG_
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

2015 Launch Records

- There were only 2 records of failed landing on drone ship in the year 2015
- We identified the 3 columns we wanted to return in the output. A WHERE clause with AND statement to ensure that only failed drone ship landings and dates starting in year 2015 are returned

```
SELECT BOOSTER_VERSION, LAUNCH_SITE, LANDING__OUTCOME FROM SPACEXTBL

  where (LANDING__OUTCOME like 'Failure (drone ship)')

  and (DATE like '2015%');
```

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

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

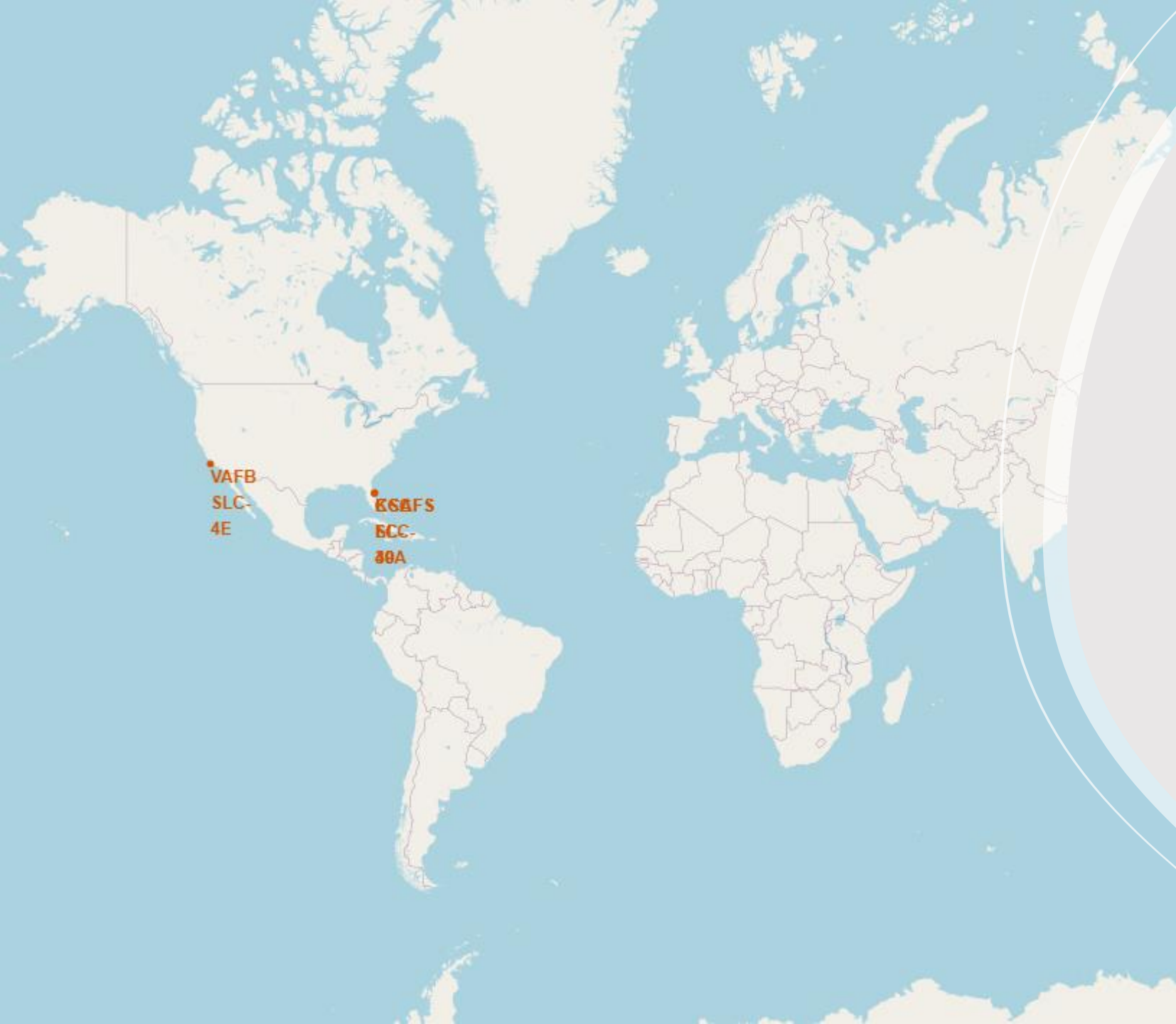
- We only wanted to return landing outcome and the COUNT of each landing outcome. The WHERE clause to filter for landing outcomes BETWEEN given dates. GROUP BY clause for each landing outcome and ORDER BY clause to have the final output in descending order

```
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;
```

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

Section 3

Launch Sites Proximities Analysis

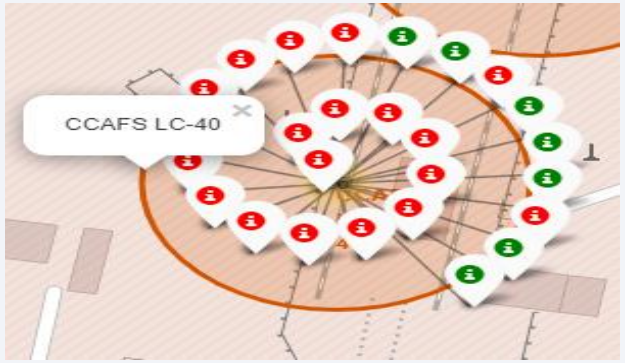


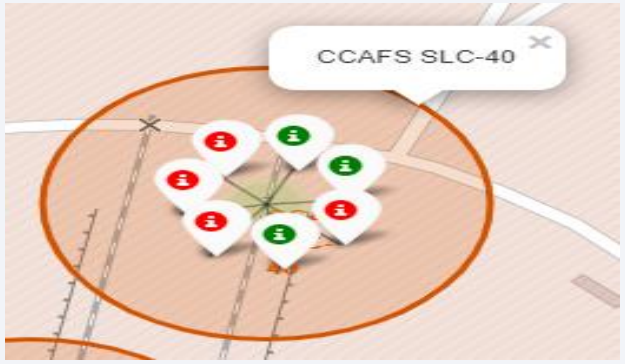
All Launch Sites Global map markers

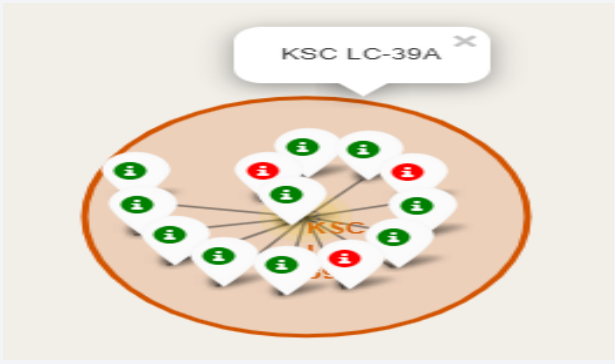
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

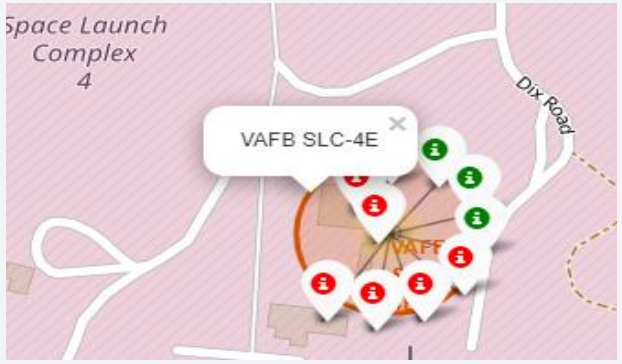
Florida Launch Site





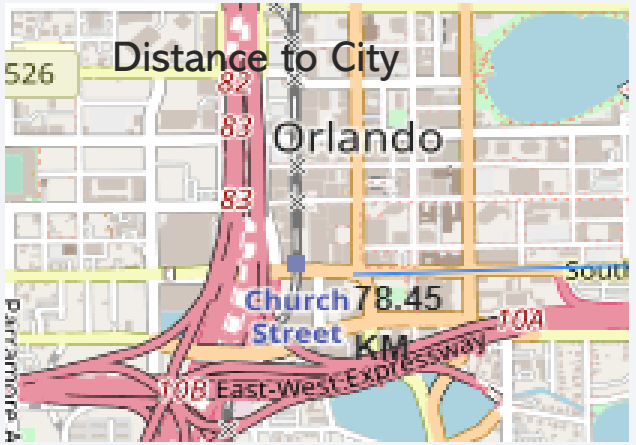


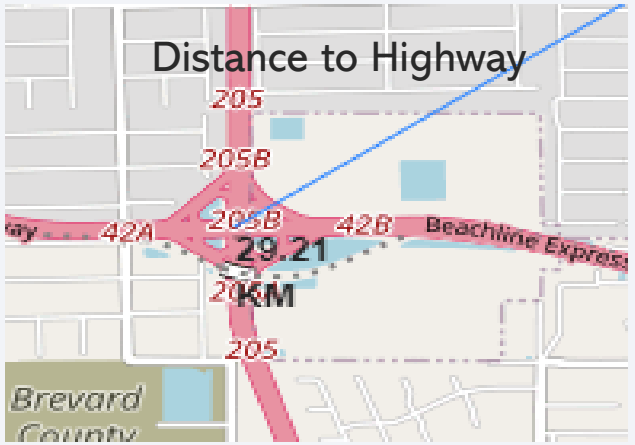
California Launch Site

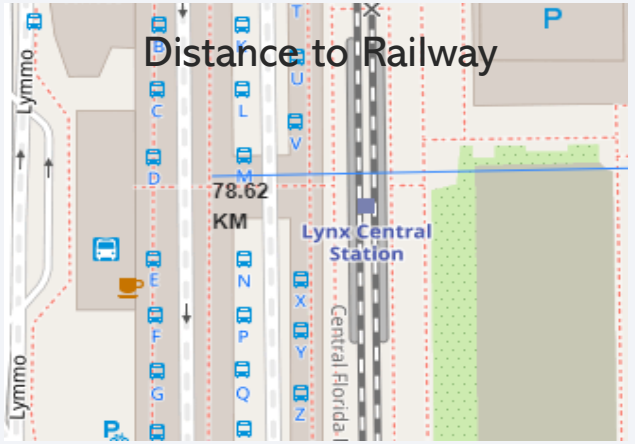


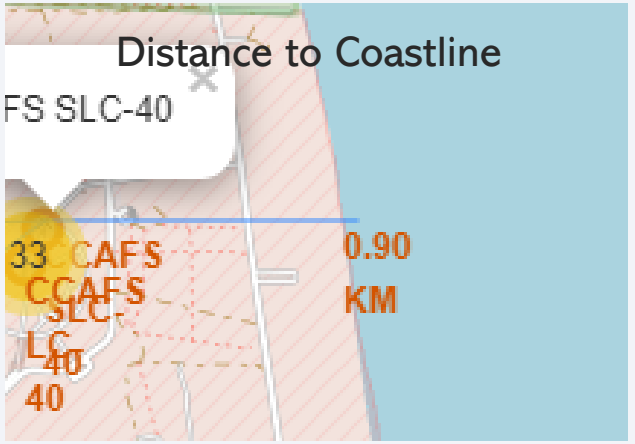
Green markers shows successful launch sites red markers shows failures

Launch Site distance to landmarks using CCAFS-SLC-40 as a reference

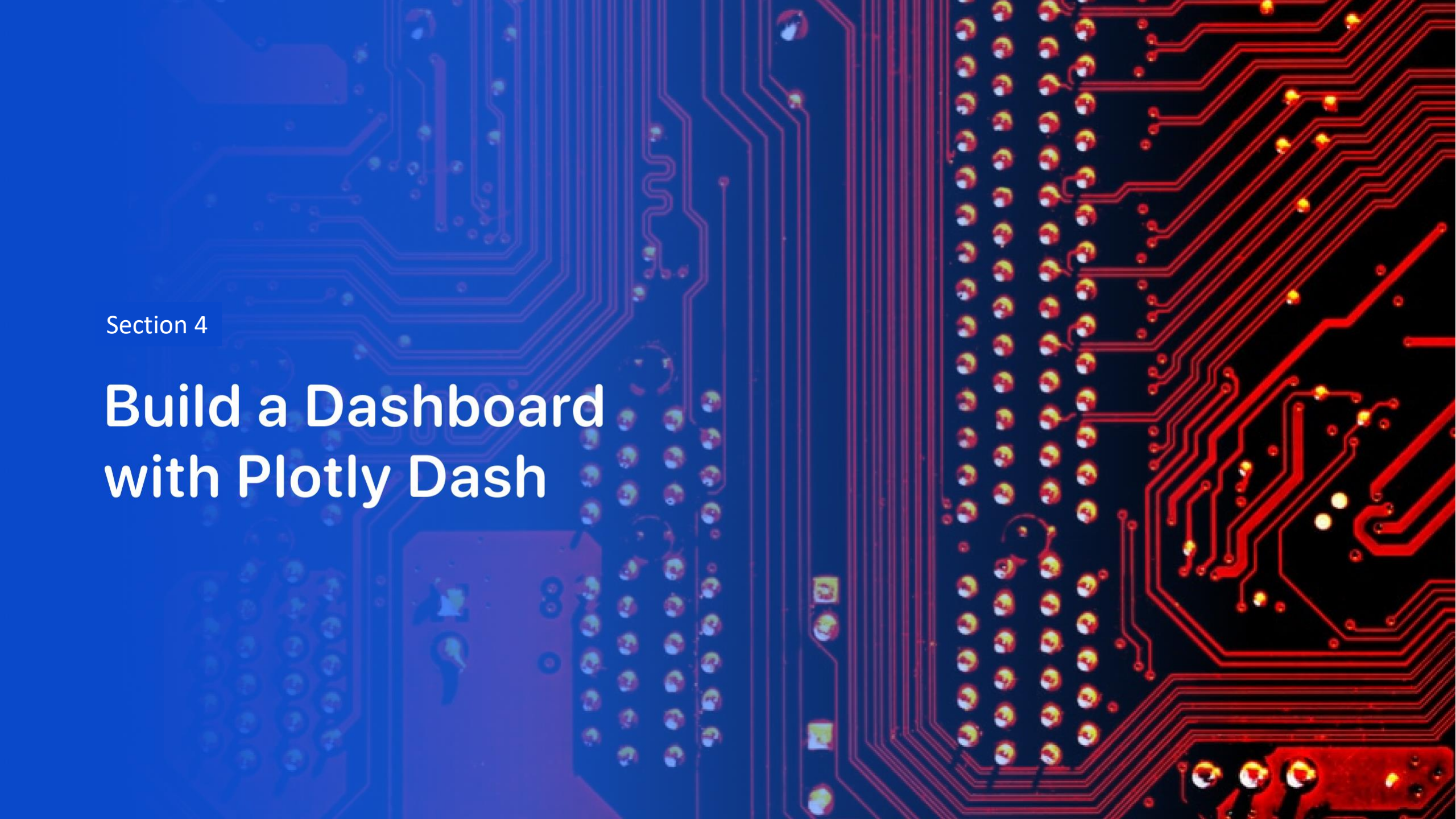




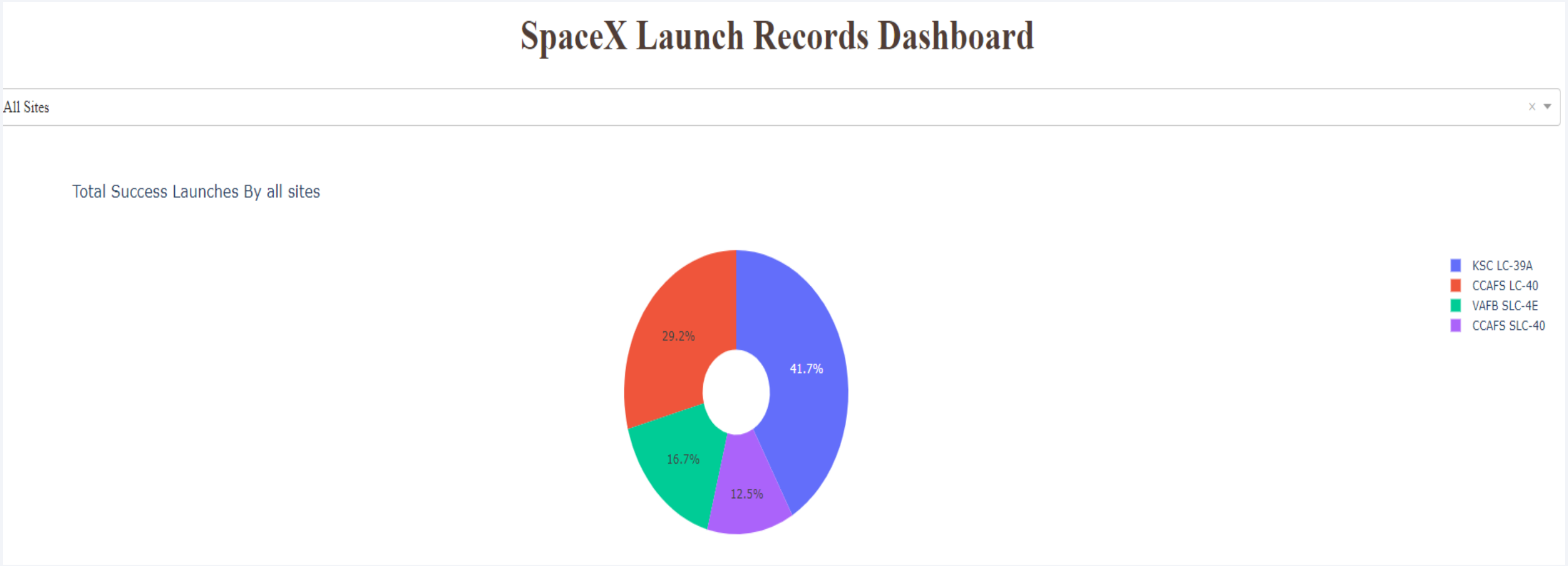




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

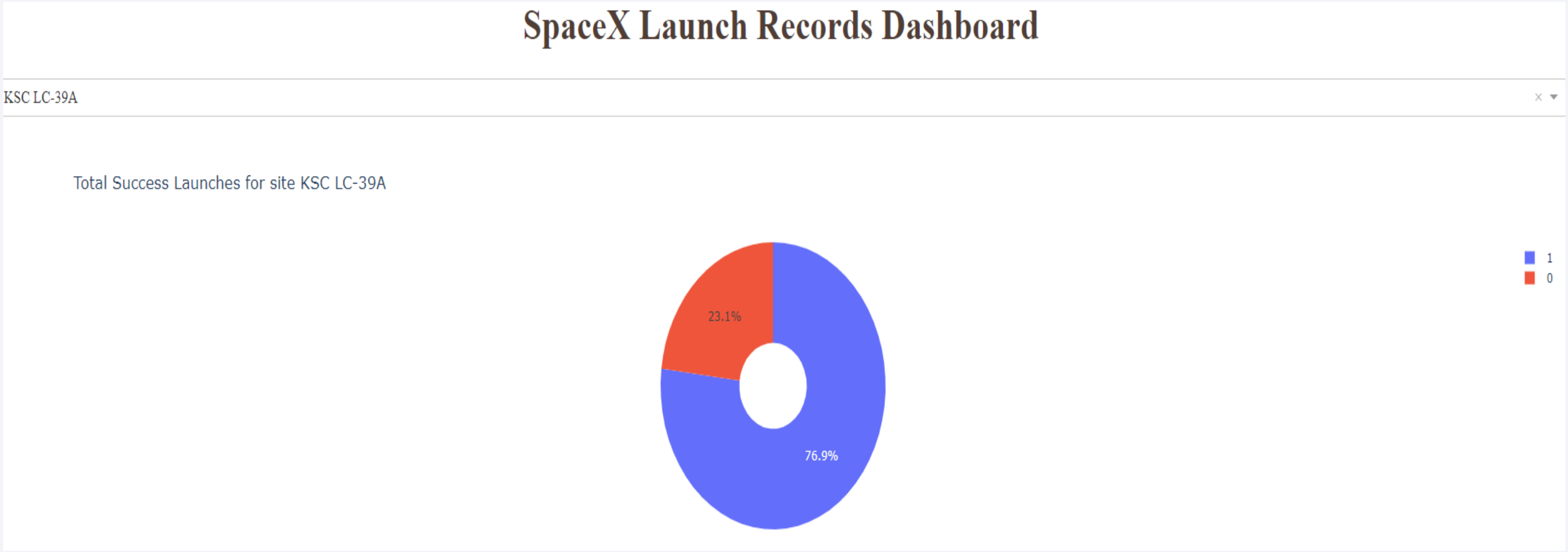

Section 4
Build a Dashboard with Plotly Dash

Dashboard – launch success rates across all sites



We can see that KSC LC-39A had the most successful launches from all the sites

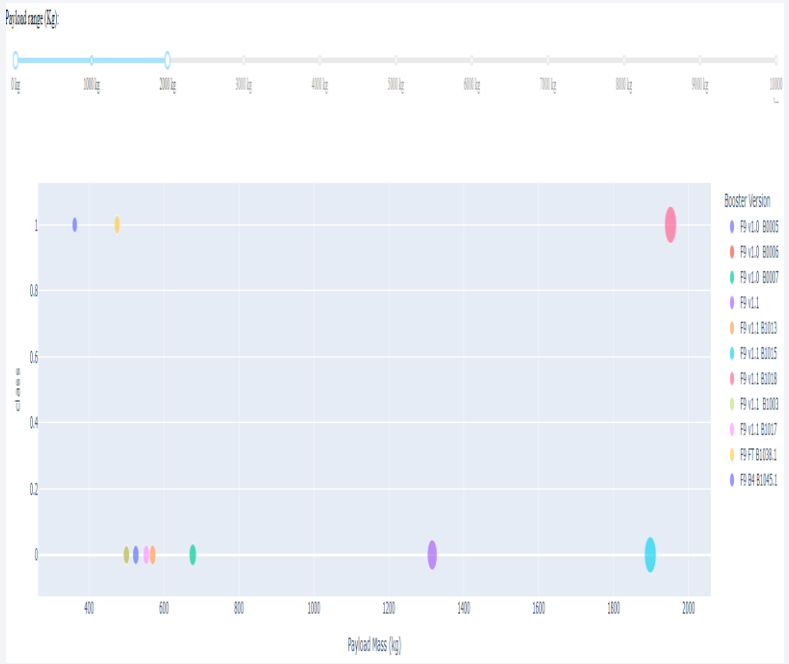
Dashboard –the launch site with highest launch success ratio



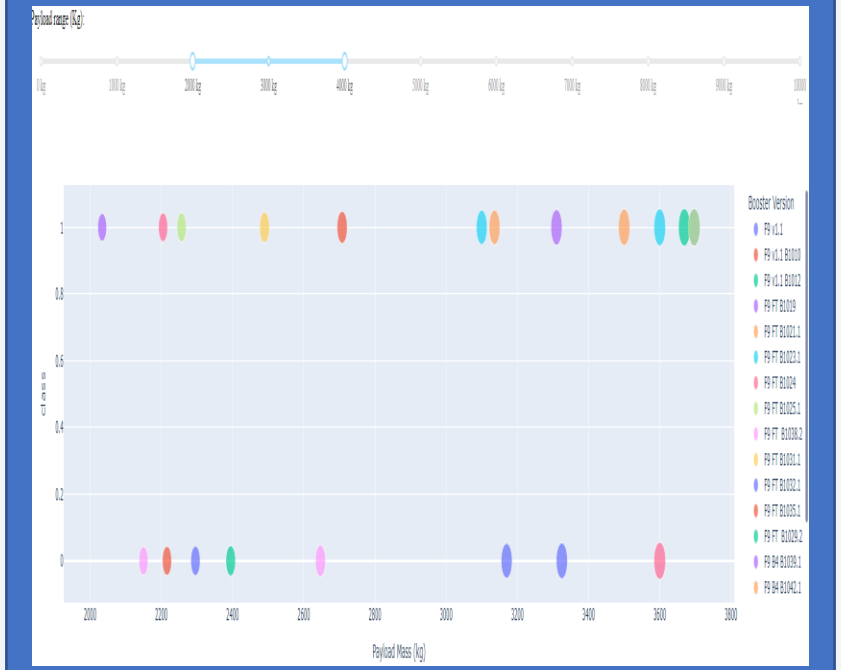
KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Dashboard – Payload vs. Launch Outcome scatter plot for all sites, with different payloads

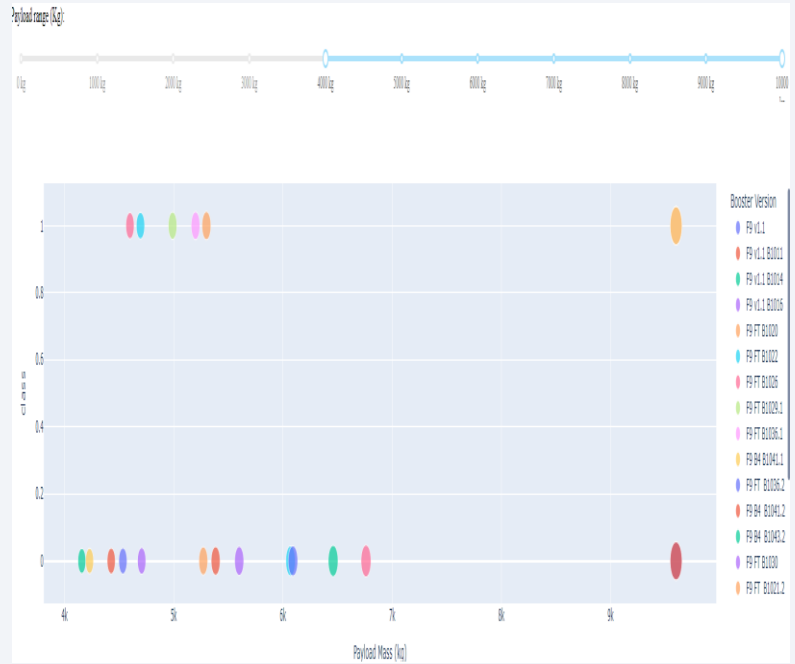
Light weight



Medium weight



Heavy weight

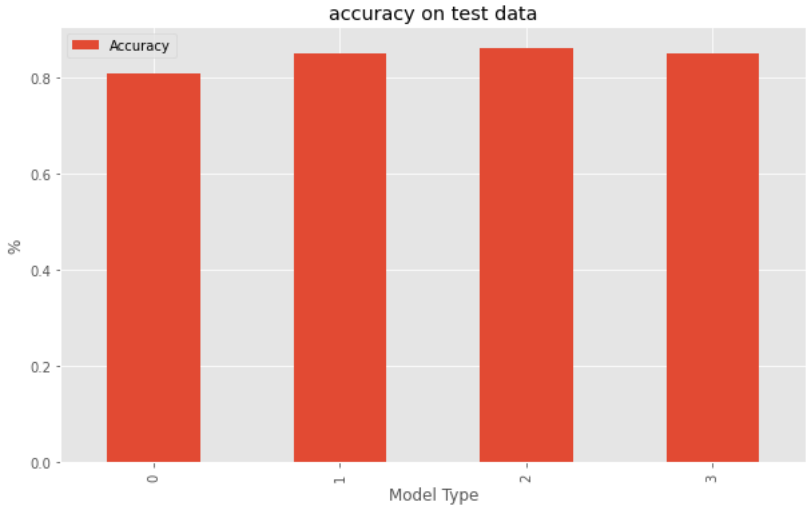


We can see the success rates for payloads between 2000kg and 4000kg have a higher success rate

Section 5

Predictive Analysis (Classification)

Classification Accuracy



0 = Logistic Regression 1 = Support Vector Machine

2 = Decision Tree 3 = K-Nearest Neighbour

Find the method performs best:

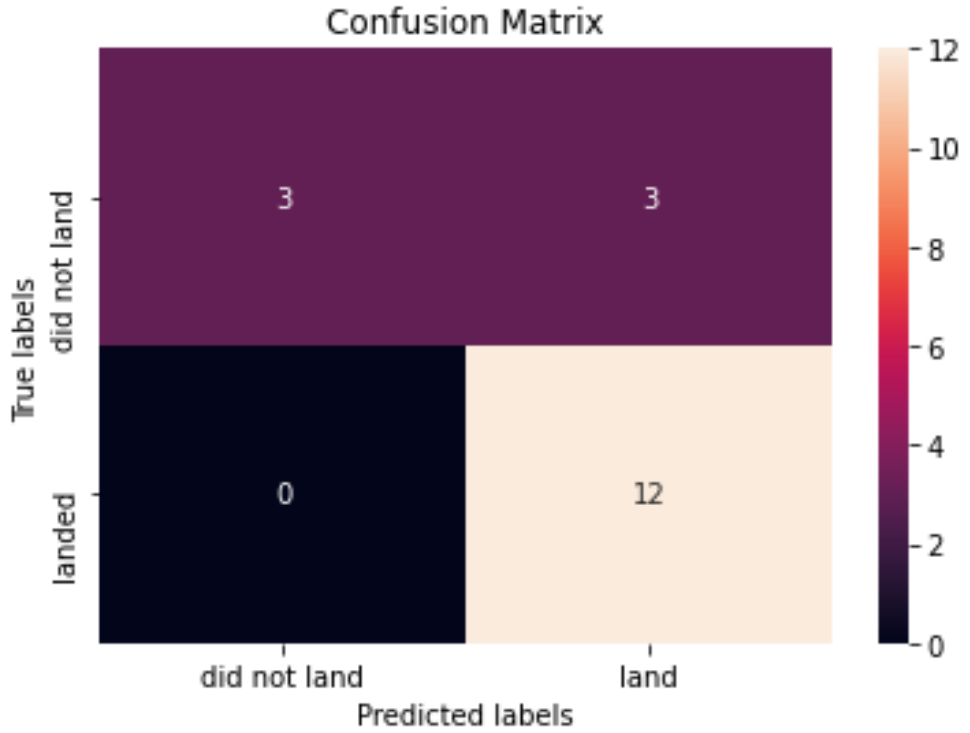
```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8607142857142858
Best Params is : {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 2, 'splitter': 'random'}
```

The Decision Tree algorithm provides the best score of 86.1% after tuning the parameters for each

Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



Conclusions

- The Decision Tree Classifier Algorithm is the best for Machine Learning for this dataset
- We can see that KSC LC-39A had the most successful launches from all the sites
- Orbit GEO,HEO,SSO,ES-L1 has the best Success Rate
- Payloads between 2000kg and 4000kg have a higher success rate
- SpaceX launch success rate started to increase year on year from 2013 through to 2020

Thank you!