Winning Space Race with Data Science

Hanscom Ayertey Opey
May 25 2022

Outline

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

Executive Summary

- **Summary of methodologies**
  - ✓ Data Collection through API
  - ✓ Data Collection with Web Scraping
  - ✓ Data Wrangling
  - ✓ Exploratory Data Analysis with SQL
  - ✓ Exploratory Data Analysis with Data Visualization
  - ✓ Interactive Visual Analytics with Folium
  - ✓ Machine Learning Prediction
- **Summary of all results**
  - ✓ Exploratory Data Analysis result
  - ✓ Interactive analytics in screenshots
  - ✓ Predictive Analytics result

Introduction

- Project background and context

  SpaceX 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 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. By using machine learning, we can predict if the first stage will land successfully. This information would be useful to a rival company looking to compete with SpaceX for rocket launches.

- Problems to be answered:
  - ✓ What factors determine if the first stage will land successfully?
  - ✓ What interactions among various factors determine the success rate of a landing?
  - ✓ What operating conditions need to be in place to ensure a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology
- Perform data wrangling
- 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

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

- Used get requests to collect rocket launch data from the SpaceX API, converted json data into a pandas dataframe, cleaned data, and filled in missing values.
- GitHub URL to notebook: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/b7bcc9db82064e211137a5f610ac2d85c2b3e7d88/Spacex-data-collection-api.ipynb

1. Get request for rocket launch data using API

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

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

2. Use json_normalize method to convert json result to dataframe

In [12]:
```
# Use json_normalize method to convert the json result into a dataframe

# decode response content as json
static_json_df = res.json()
```

In [13]:
```
# apply json_normalize
data = pd.json_normalize(static_json_df)
```

3. We then performed data cleaning and filling in the missing values

In [30]:
```
rows = data_falcon9['PayloadMass'].values.tolist()[0]

df_rows = pd.DataFrame(rows)
df_rows = df_rows.replace(np.nan, PayloadMass)

data_falcon9['PayloadMass'][0] = df_rows.values
data_falcon9
```

Data Collection - Scraping

- Using Beautiful Soup, the Falcon9 Launch Wiki page was scraped for data and parsed into a pandas dataframe.
- GitHub URL to notebook: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/b7bcc9db82064e21137a5f610ac2d85c2b3e7d88/Spacex-webscraping.ipynb

1. Apply HTTP Get method to request the Falcon 9 rocket launch page

```
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
        html_data = requests.get(static_url)
        html_data.status_code
Out[5]: 200
```

2. Create a BeautifulSoup object from the HTML response

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

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

```
In [7]: # Use soup.title attribute
        soup.title
Out[7]: <title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

3. Extract all column names from the HTML table header

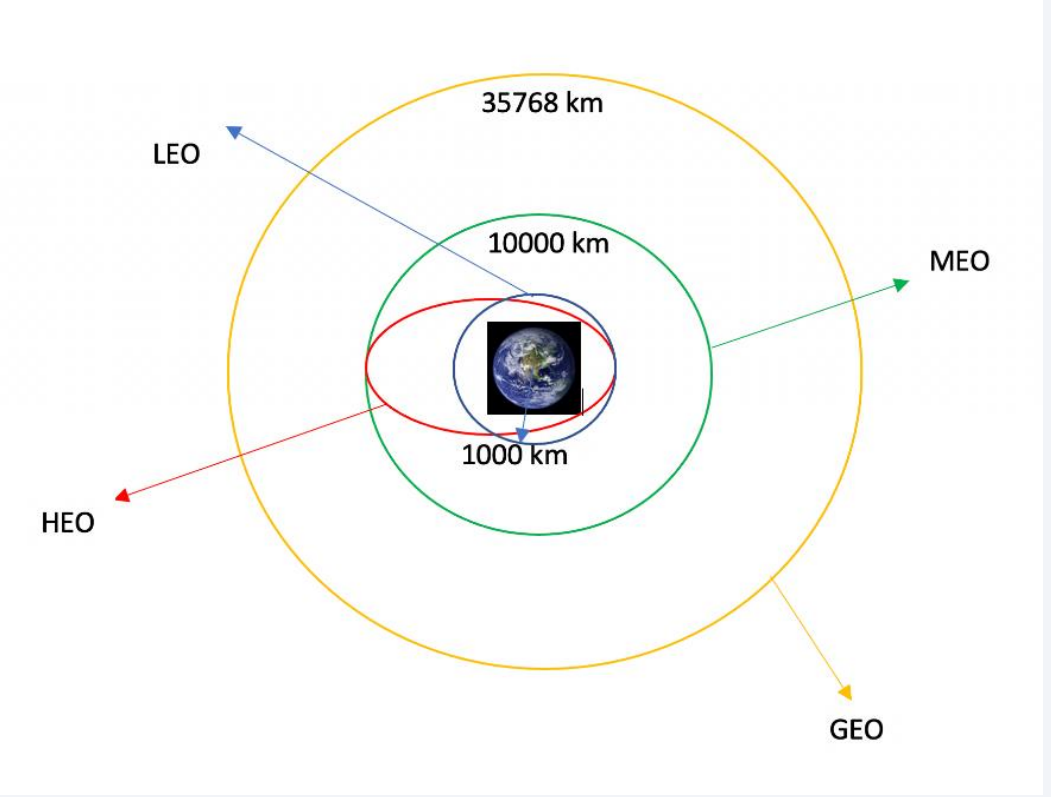
```
In [10]: column_names = []

         # 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

         element = soup.find_all('th')
         for row in range(len(element)):
             try:
                 name = extract_column_from_header(element[row])
                 if (name is not None and len(name) > 0):
                     column_names.append(name)
             except:
                 pass
```

4. Create a dataframe by parsing the launch HTML tables
5. Export data to csv

Data Wrangling

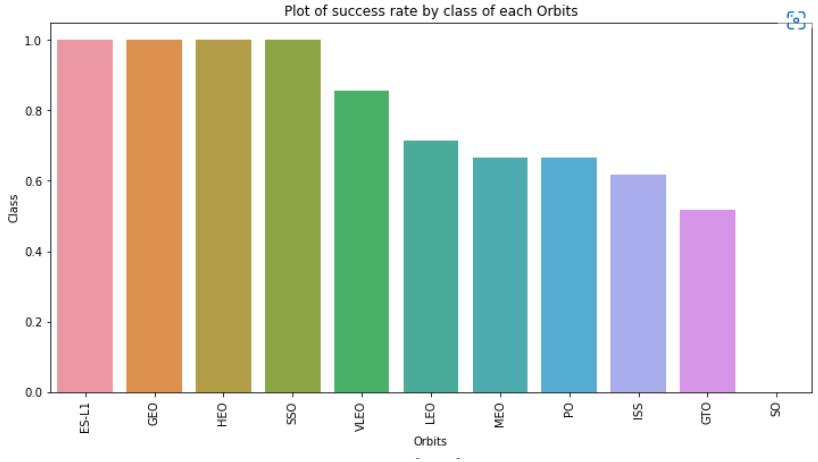
- Exploratory data analysis was performed and the training labels were determined.
- The number of launches at each site, and the number and occurrence of each orbits were calculated.
- Also created a landing outcome label from outcome column and exported the results to a csv file.
- GitHub URL to notebook:

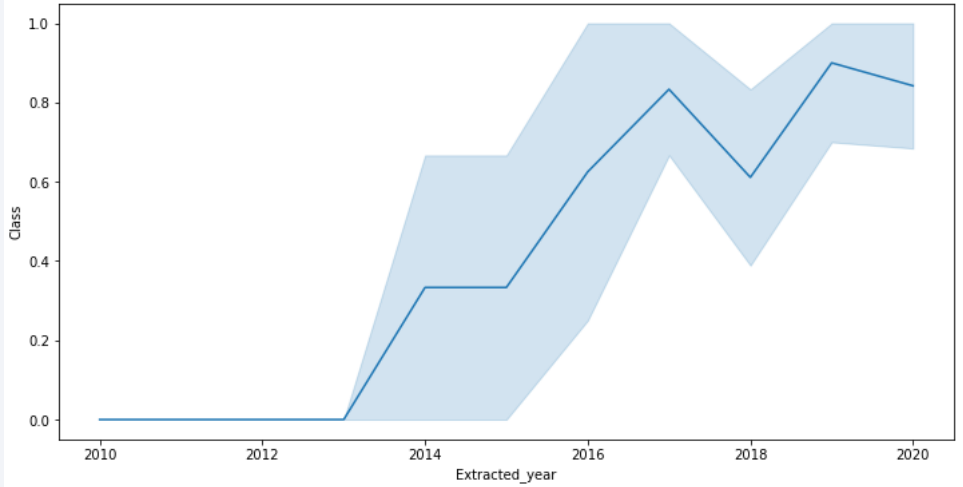
https://github.com/Scomii/IBM-Data-Science-Capstone/blob/b7bcc9db82064e21137a5f610ac2d85c2b3e7d88/Data%20wrangling.ipynb

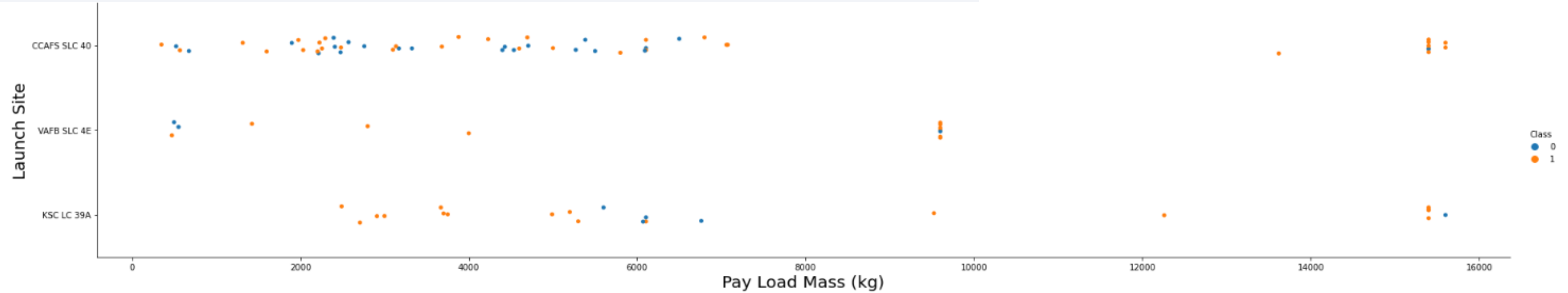


EDA with Data Visualization

- Exploration of the data was done 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, and the launch success yearly trend.
- GitHub URL to notebook: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/b7bcc9db82064e21137a5f610ac2d85c2b3e7d88/Data%20wrangling.ipynb







EDA with SQL

EDA with SQL was applied to get insight from the data. Wrote queries to find out :

✓The names of unique launch sites in the space mission.

✓The total payload mass carried by boosters launched by NASA (CRS)

✓The average payload mass carried by booster version F9 v1.1

✓The total number of successful and failure mission outcomes

✓The failed landing outcomes in drone ship, their booster version and launch site names.

- GitHub URL to completed EDA with SQL notebook, https://github.com/Scomii/IBM-Data-Science-Capstone/blob/9c00b11fb6cd1d77ea0f24f134325b8a653e23f8/EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- Marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.
- Assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, identified which launch sites have relatively high success rate.
- Calculated the distances between a launch site to its proximities and answered some question such as:
  - Are launch sites near railways, highways and coastlines?
  - Do launch sites keep certain distance away from cities?

https://github.com/Scomii/IBM-Data-Science-Capstone/blob/71fac4f533c69edc499b419469fbcb3f9e514aff/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Built an interactive dashboard with Plotly dash.
- Plotted pie charts showing the total launches by a certain sites.
- Plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- GitHub URL to notebook: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/71fac4f533c69edc499b419469fbcb3f9e514aff/dashapp.py

Predictive Analysis (Classification)

- Loaded the data using numpy and pandas, transformed the data, split data into training data and testing data.
- Built different machine learning models and tuned different hyperparameters using GridSearchCV.
- Used accuracy as the metric for the model and improved the model using feature engineering and algorithm tuning.
- Determined the best performing classification model.
- GitHub URL to notebook: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/2e15afdaeaa075ef686e443910282cf0275436a0/Machine%20Learning%20Prediction.ipynb

Results

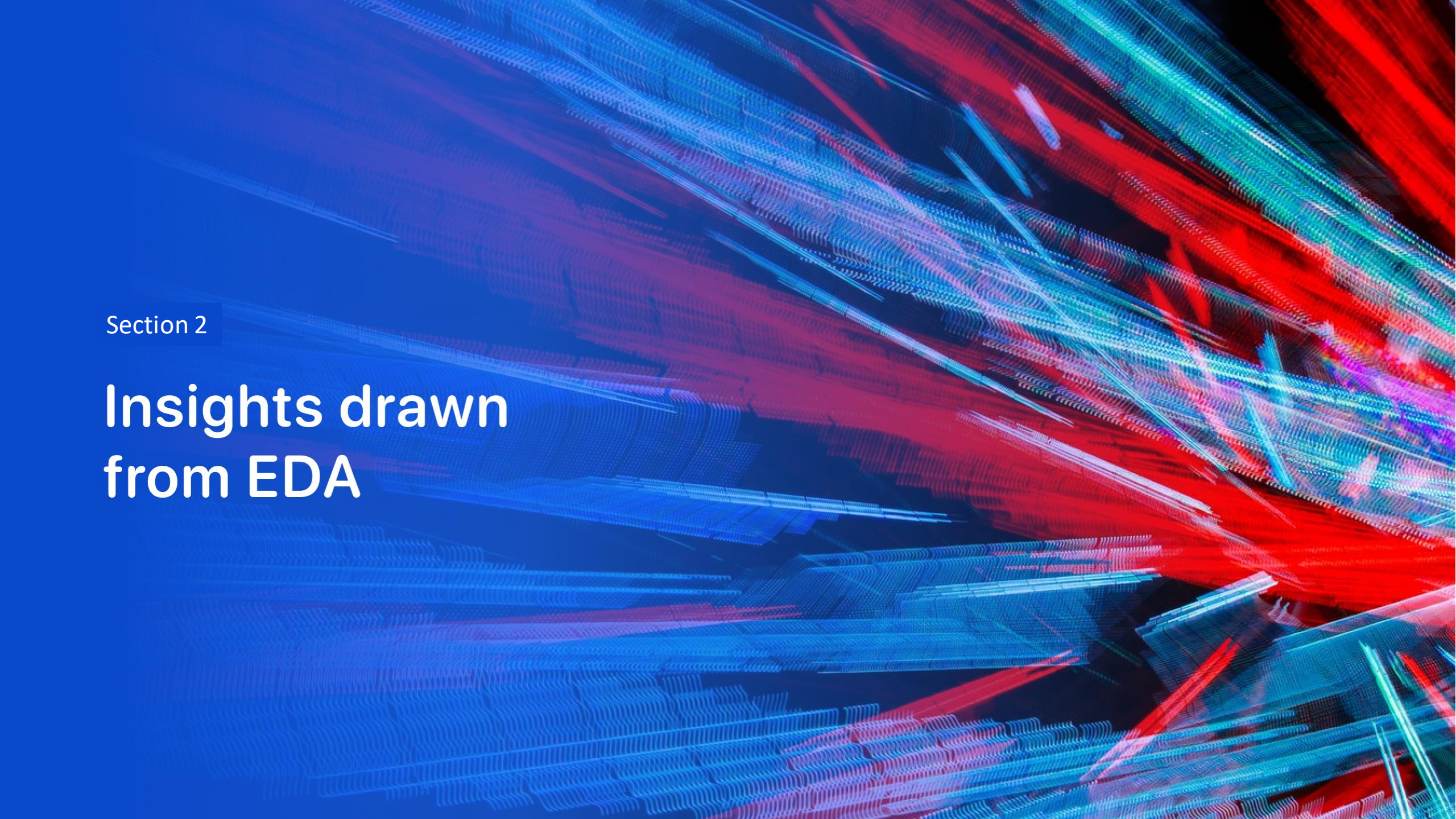
- Exploratory data analysis results: https://github.com/Scomii/IBM-Data-Science-Capstone/blob/b7bcc9db82064e21137a5f610ac2d85c2b3e7d88/Data%20wrangling.ipynb

- Interactive analytics results:https://github.com/Scomii/IBM-Data-Science-Capstone/blob/71fac4f533c69edc499b419469fbcb3f9e514aff/Interactive%20Visual%20Analytics%20with%20Folium.ipynb

- Predictive analysis results:https://github.com/Scomii/IBM-Data-Science-Capstone/blob/2e15afdaeaa075ef686e443910282cf0275436a0/Machine%20Learning%20Prediction.ipynb

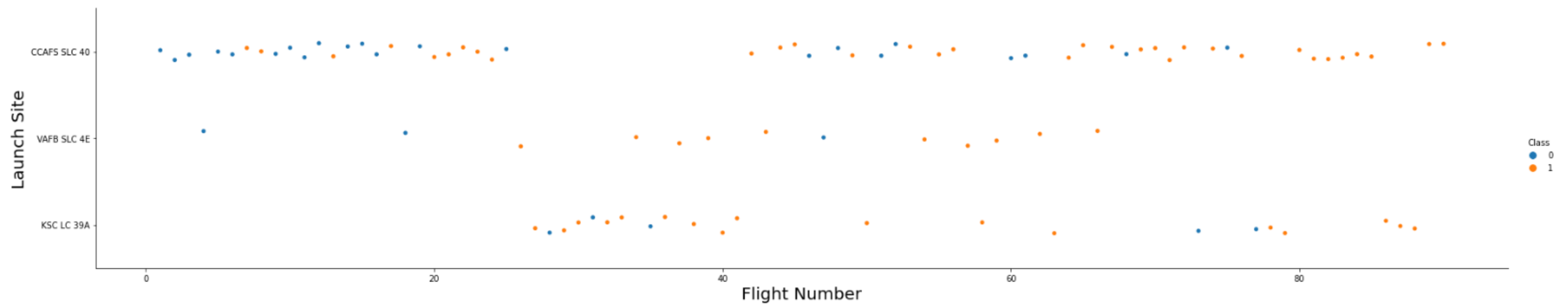
Section 2

Insights drawn from EDA



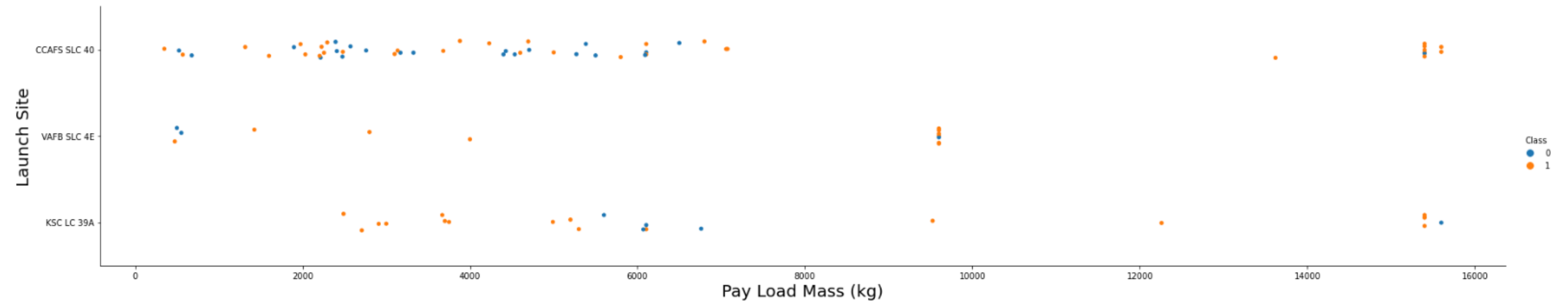
Flight Number vs. Launch Site

From the plot, it can be seen that the larger the flight amount at a launch site, the greater the success rate at a launch site.

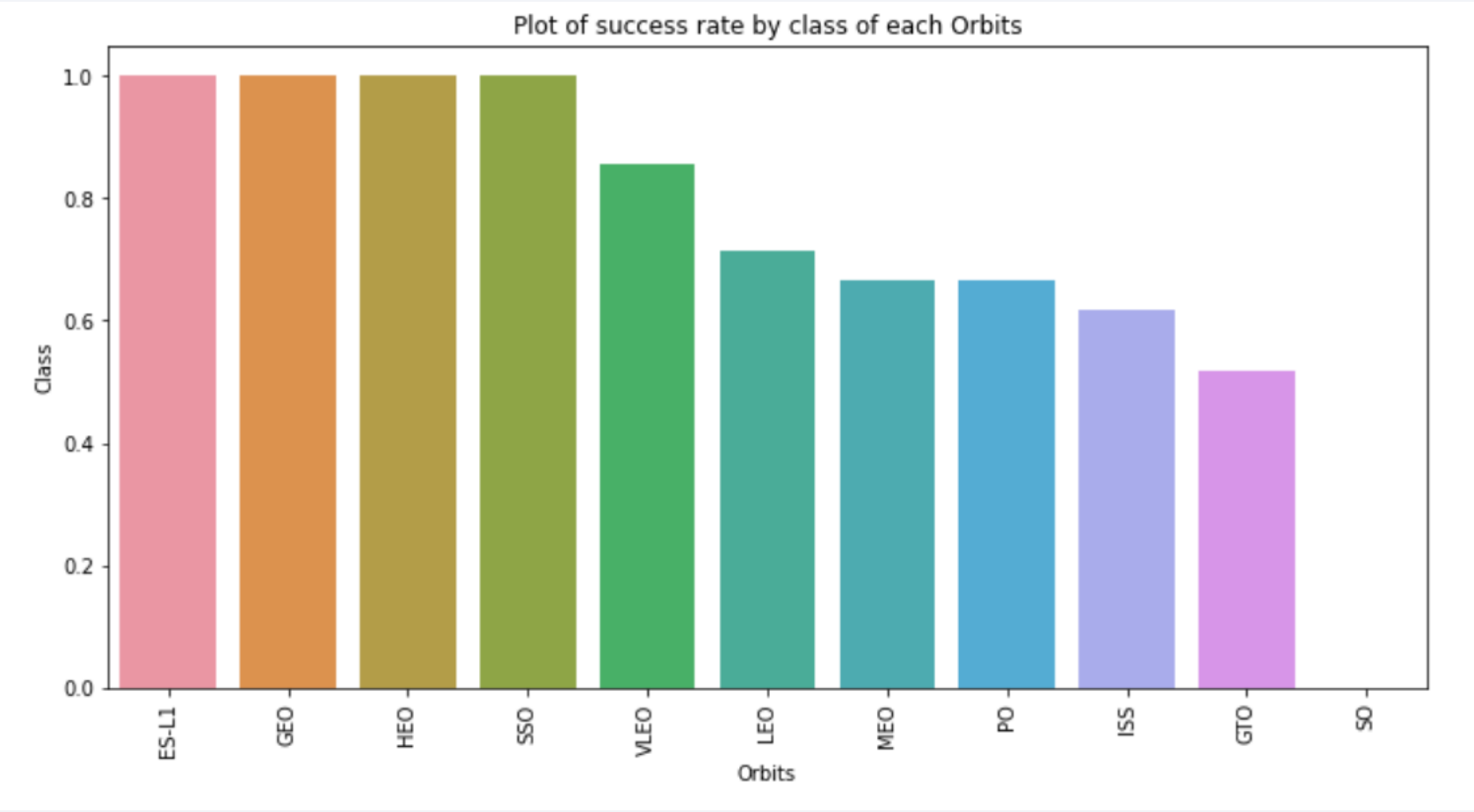


Payload vs. Launch Site

From the plot, it can be seen that bigger pay loads at launch sites, had greater success rates.



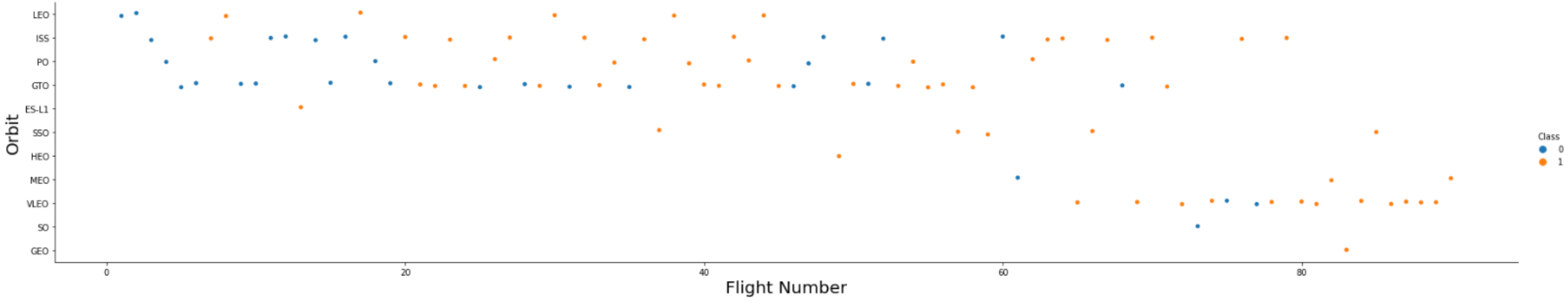
Success Rate vs. Orbit Type



From the chart, it can be seen that ES-L1, GEO, HEO, and SSO, were the most success orbit types
with VLEO closely behind.

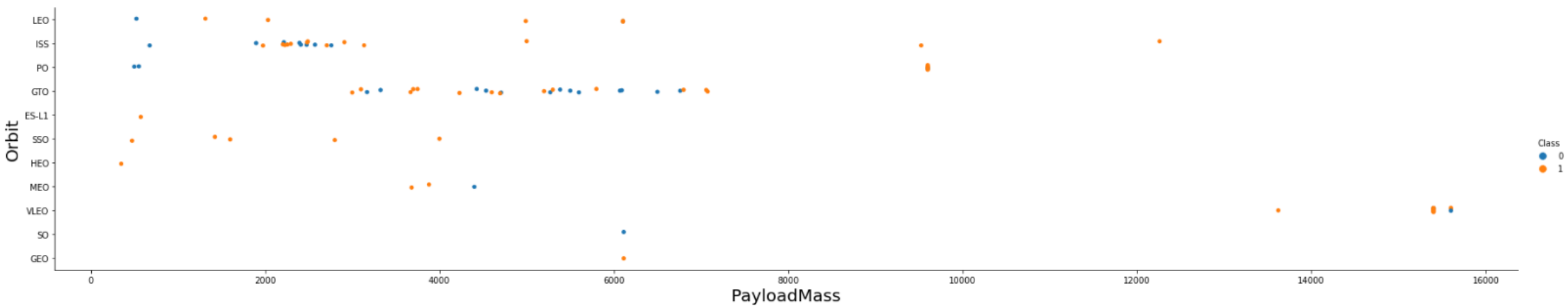
Flight Number vs. Orbit Type

From the plot, it can be seen that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit type.

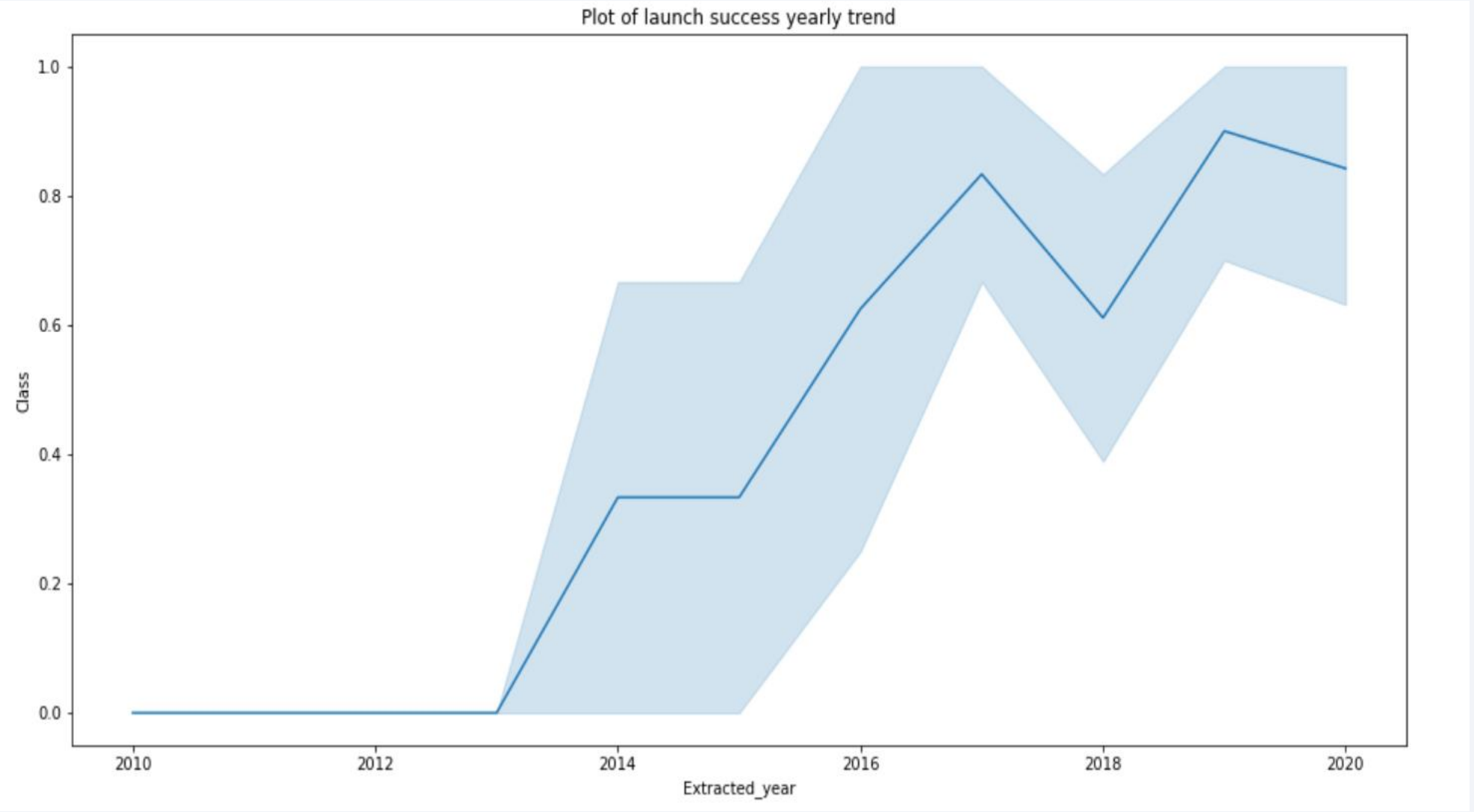


Payload vs. Orbit Type

From the plot, it can be observed that the heavier the payloads, the more the successful landings for PO, LEO and ISS orbits.



Launch Success Yearly Trend



From the plot, it can be seen that success rates increased progressively from 2013 to 2020.

All Launch Site Names

Used the key word DISTINCT to show only unique launch sites from the SpaceX data.

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

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXDATASET;
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f51
Done.
```

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

Launch Site Names Begin with 'CCA'

Used the query below to display 5 records where launch sites begin with `CCA`

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

```
%sql SELECT * FROM SPACEXDATASET WHERE LAUNCH_SITE LIKE 'CCA%'LIMIT 5;
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
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	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

Calculated the total payload carried by boosters from NASA using the query below

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE CUSTOMER LIKE 'NASA (CRS)%';
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.data
Done.
```

1
48213

Average Payload Mass by F9 v1.1

Calculated the average payload mass carried by booster version F9 v1.1 with this query

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXDATASET WHERE BOOSTER_VERSION LIKE 'F9 v1.1%';
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases
Done.
```

1
2534

First Successful Ground Landing Date

The first successful landing outcome on ground pad was on 22nd December 2015

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

Hint:Use min function

```
%sql SELECT MIN(DATE) AS FirstSuccessfull_landing_date FROM SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (ground pad)';
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

firstsuccessfull_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

Used the WHERE clause to filter for boosters which have successfully landed on drone ship and applied the AND condition to determine successful landing with payload mass greater than 4000 but less than 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 SPACEXDATASET WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
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

Used the following queries to determine the total number of successful and failure mission outcomes

List the total number of successful and failure mission outcomes

```
task_7a = %sql SELECT COUNT(MISSION_OUTCOME) AS SuccessOutcome FROM SPACEXDATASET WHERE MISSION_OUTCOME LIKE 'Success%';
task_7b = %sql SELECT COUNT(MISSION_OUTCOME) AS FailureOutcome FROM SPACEXDATASET WHERE MISSION_OUTCOME LIKE 'Failure%';
display(task_7a)
display(task_7b)
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

successoutcome
100

failureoutcome
1

Boosters Carried Maximum Payload

Determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

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

```
%sql SELECT BOOSTER_VERSION, PAYLOAD_MASS__KG_ FROM SPACEXDATASET WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXDATASET) ORDER
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

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

2015 Launch Records

Used a combinations of the WHERE clause, LIKE, AND, and BETWEEN conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql SELECT BOOSTER_VERSION,LAUNCH_SITE,LANDING__OUTCOME FROM SPACEXDATASET WHERE LANDING__OUTCOME LIKE 'Failure (drone ship)' AND YEAR(DATE) = 201
```

```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

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

Selected Landing outcomes and the COUNT of landing outcomes from the data and used the WHERE clause to filter for landing outcomes BETWEEN 2010-06-04 to 2010-03-20.

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, in descending order

```
%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) FROM SPACEXDATASET WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'GROUP BY LANDING__OUTCOME
```

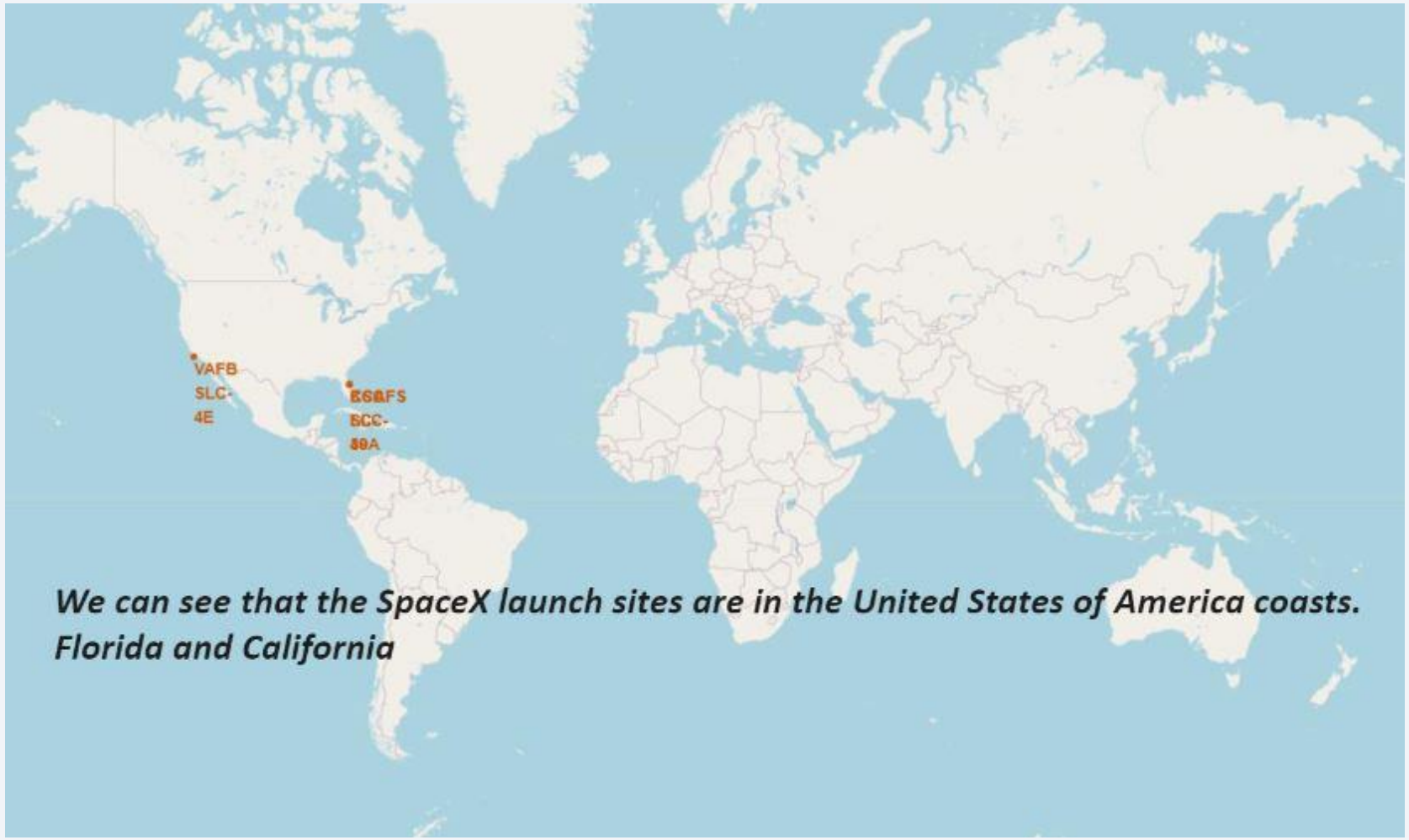
```
 * ibm_db_sa://hby61407:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

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

Applied the GROUP BY clause to group the landing outcomes and the ORDER BY clause to order the grouped landing outcome in descending order.

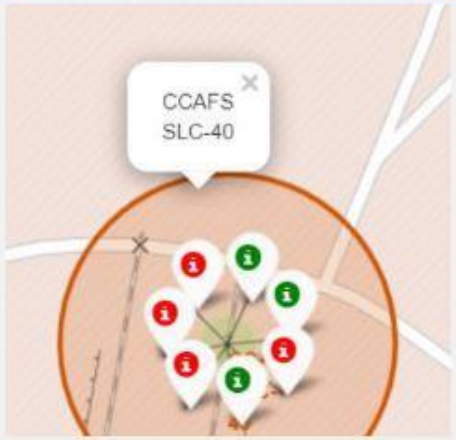

Section 3
Launch Sites
Proximities Analysis

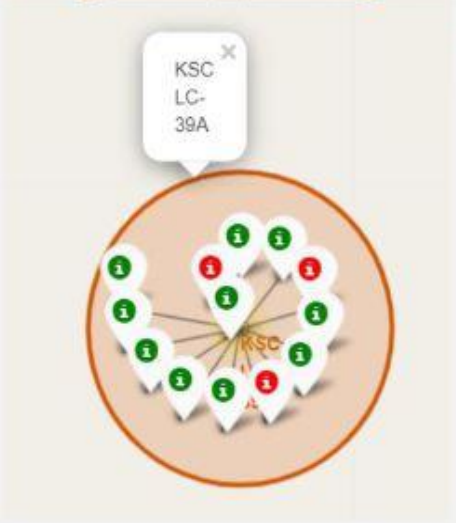
All launch sites global map markers



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





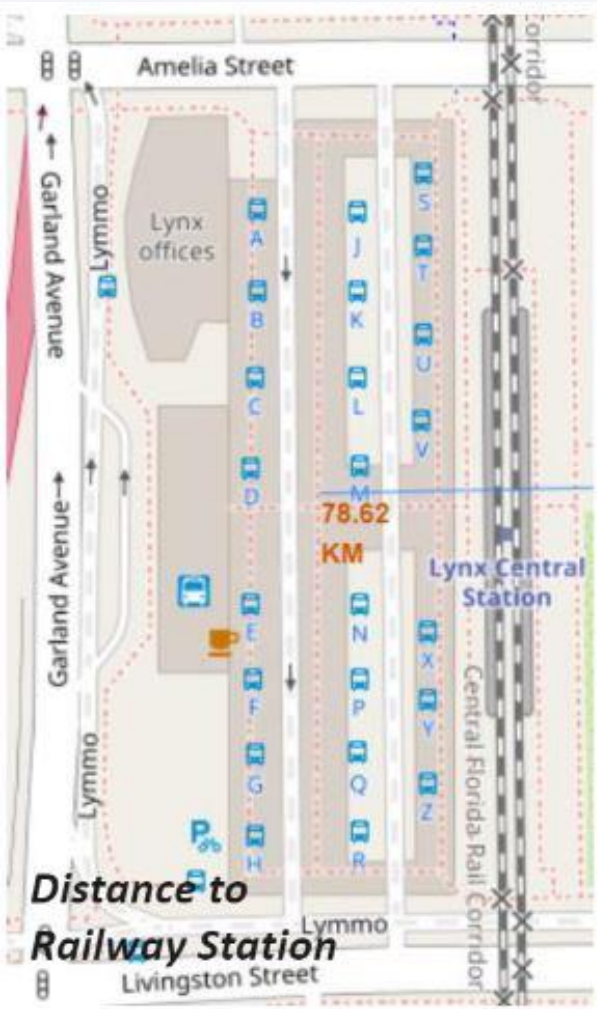
Markers showing launch sites with color labels

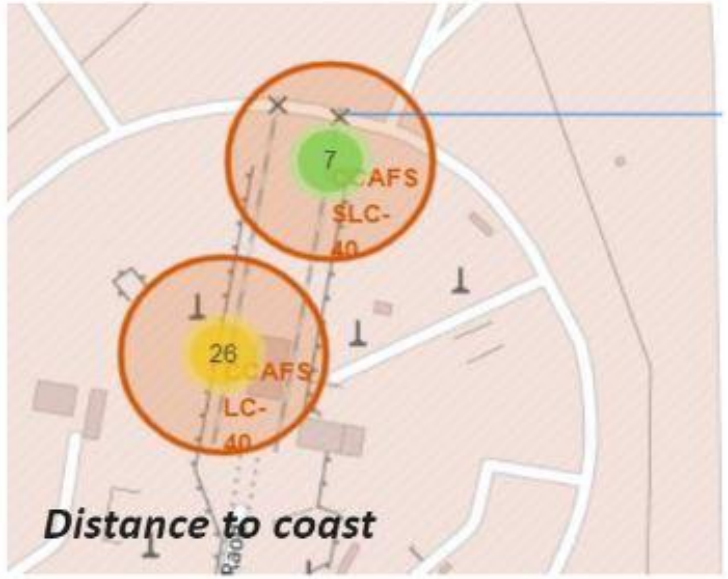
Florida Launch Sites

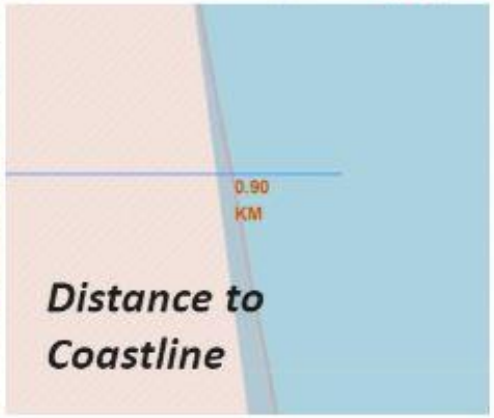
Green Marker shows successful Launches and Red Marker shows Failures

California Launch Site

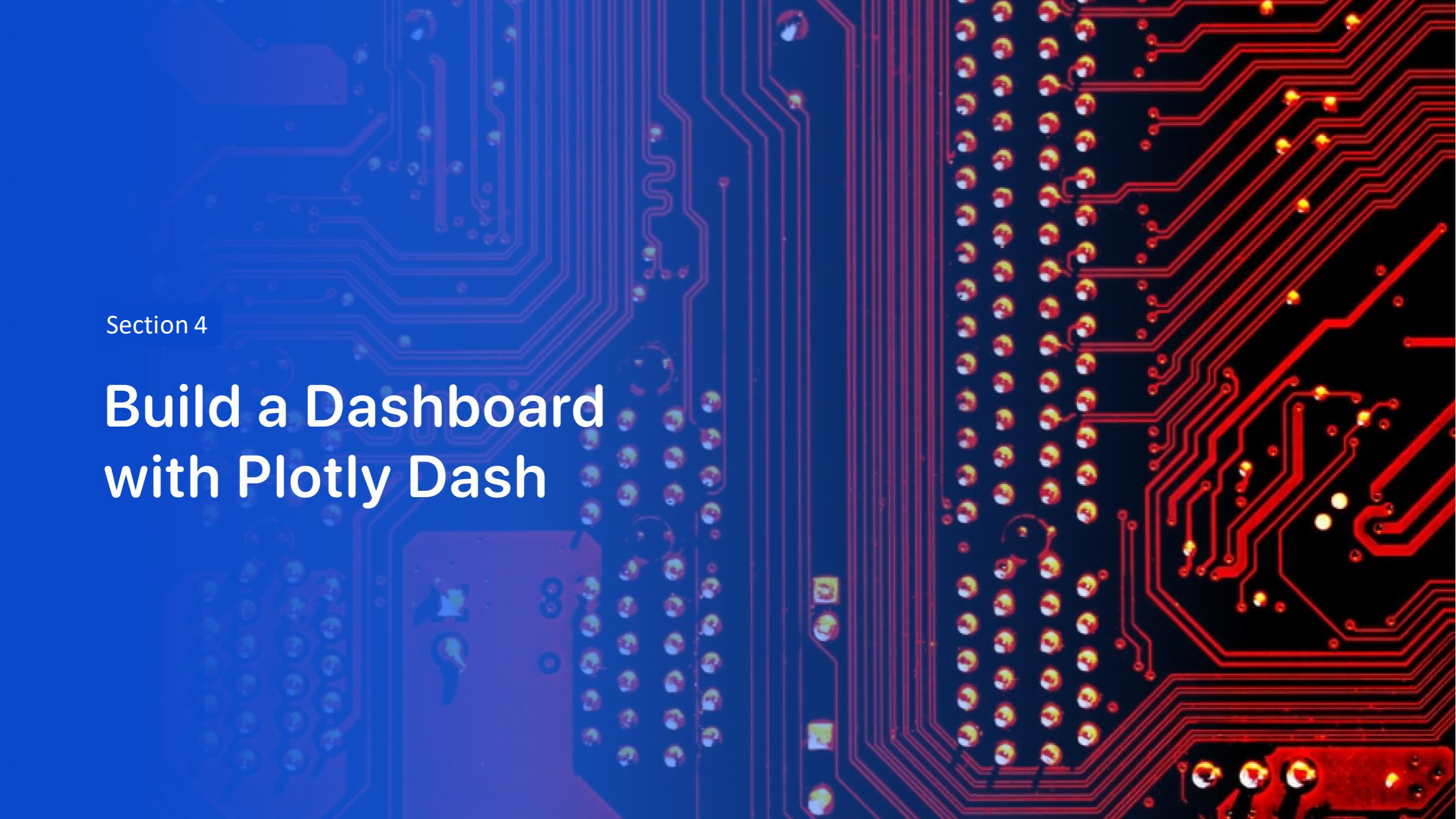
Launch Site distance to landmarks



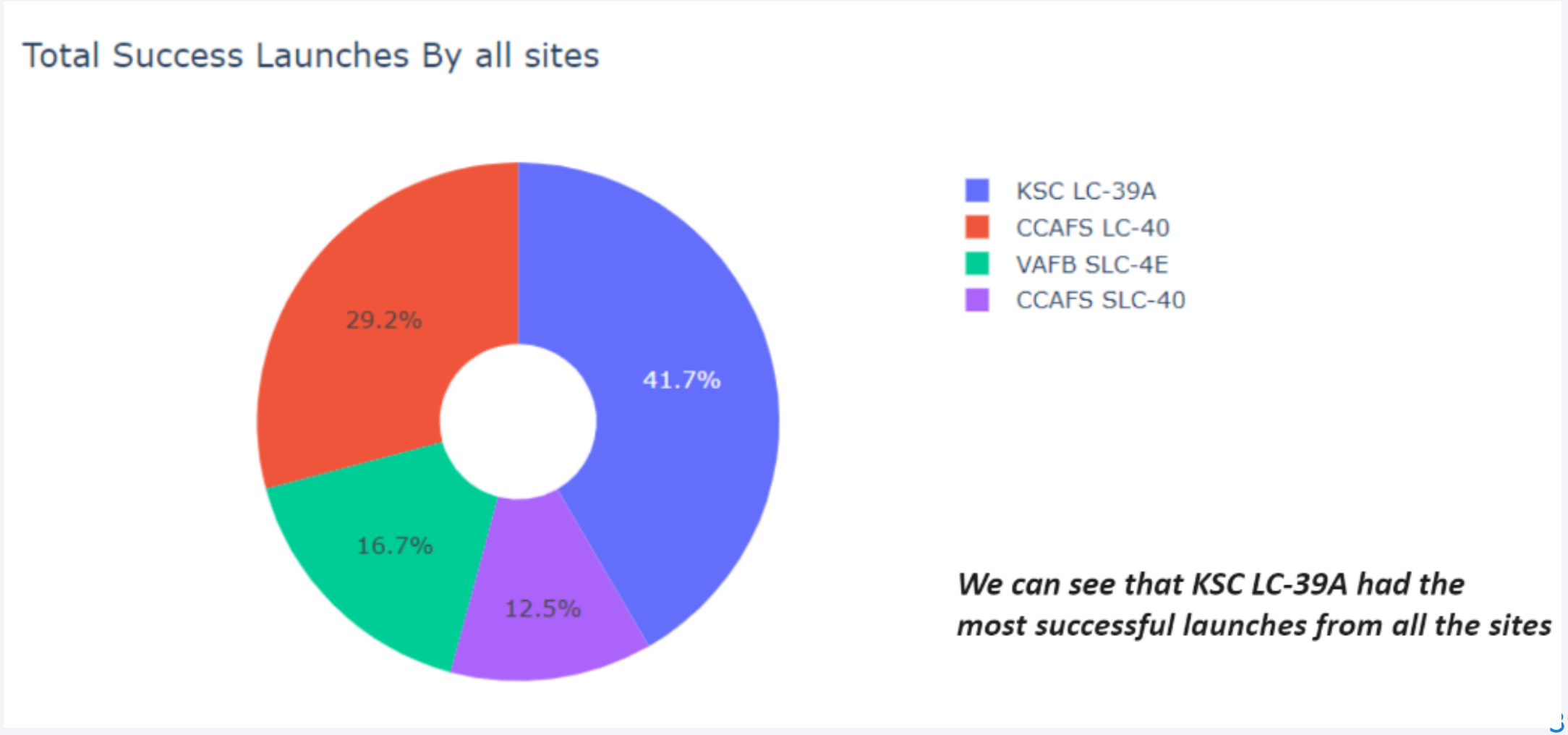




- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

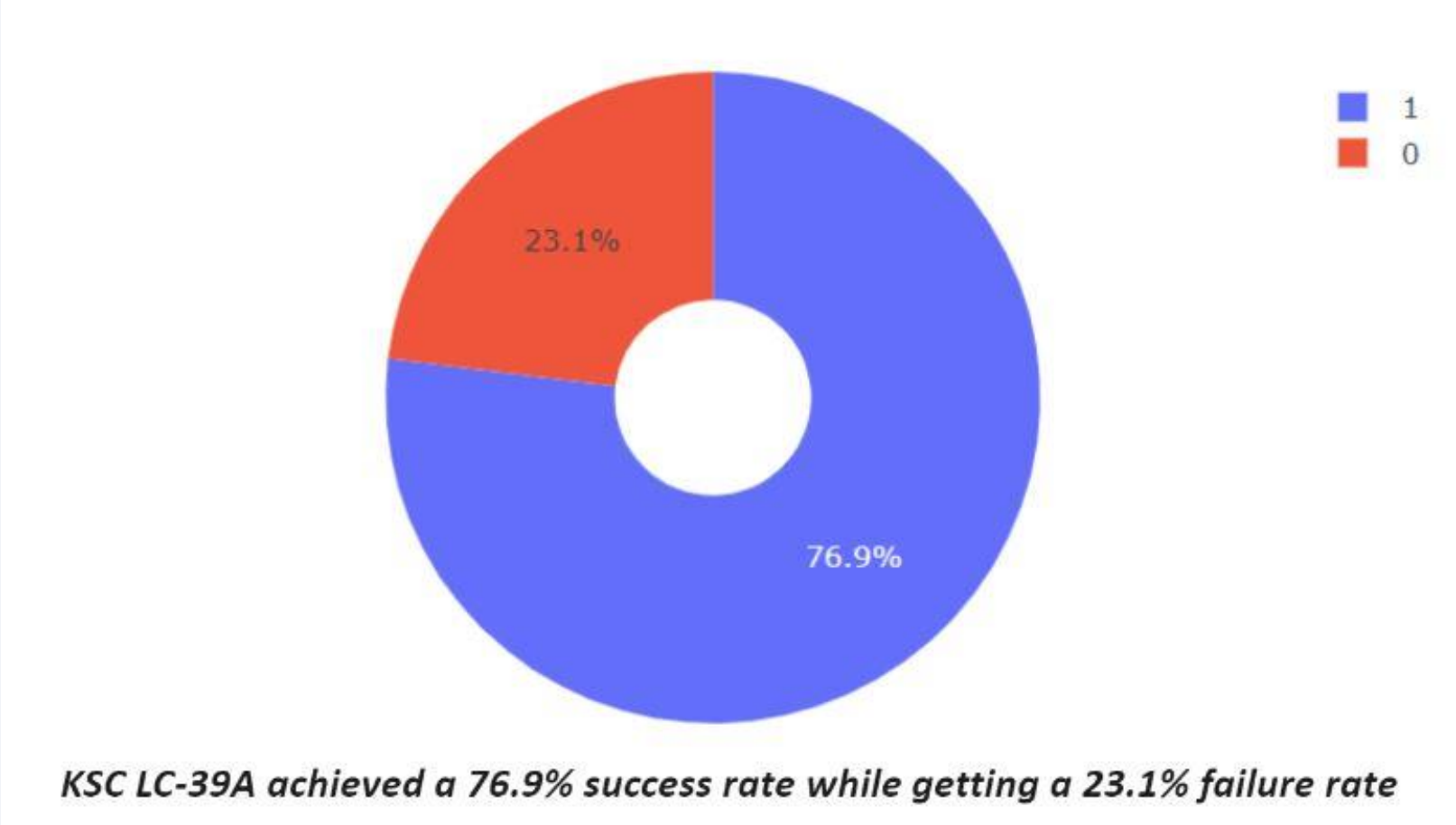

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



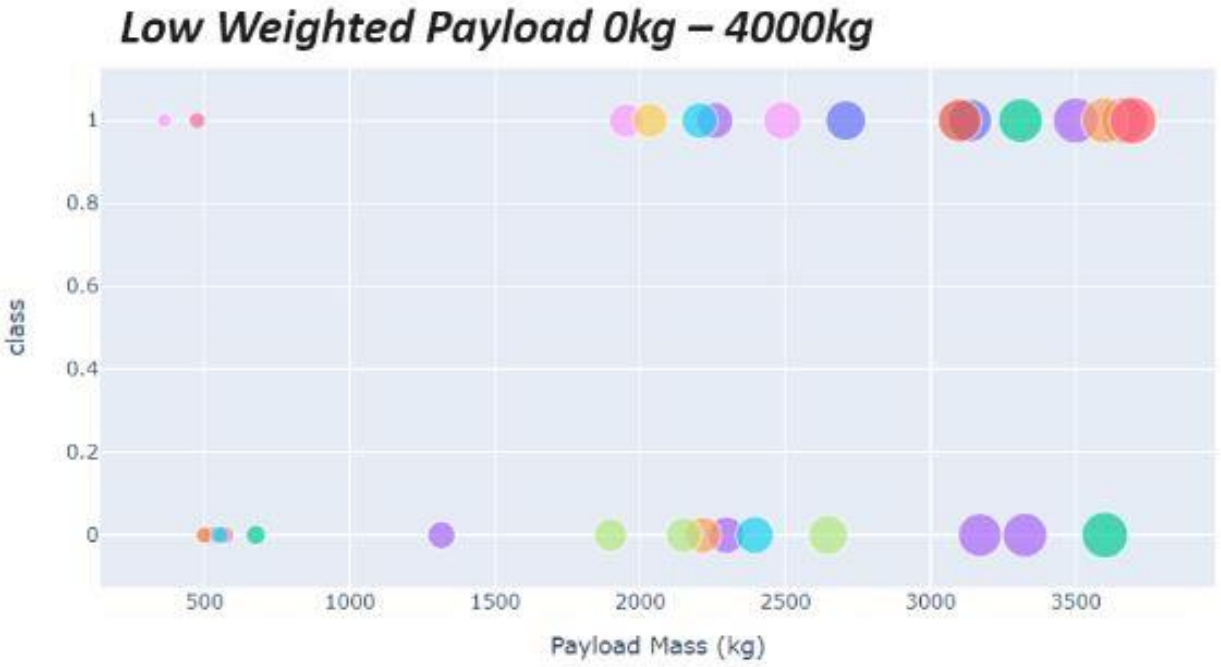
We can see that KSC LC-39A had the most successful launches from all the sites

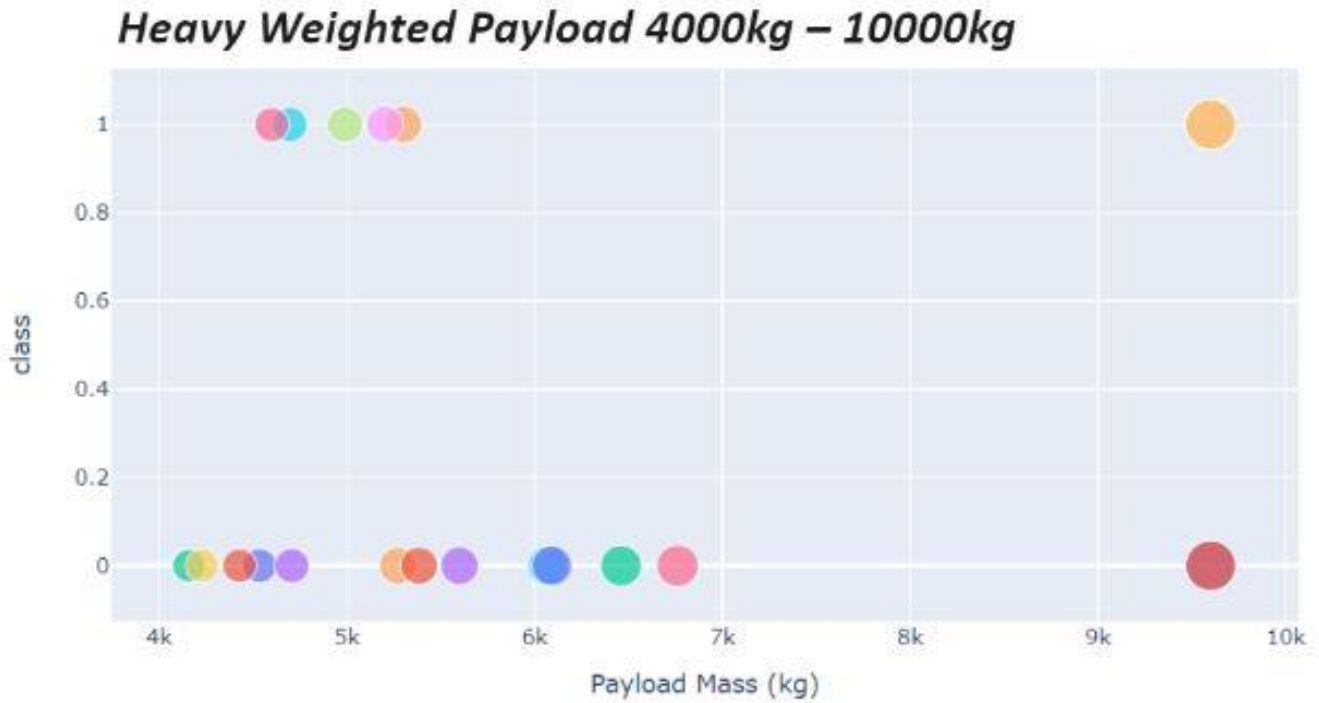
Pie chart showing the Launch site with the highest launch success ratio



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

Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider





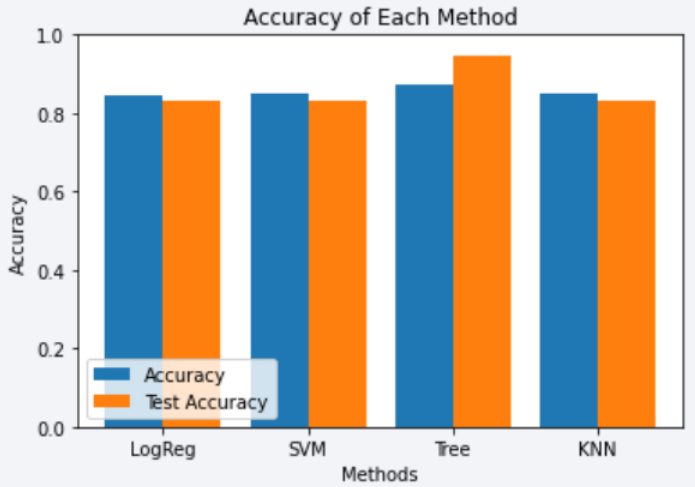
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

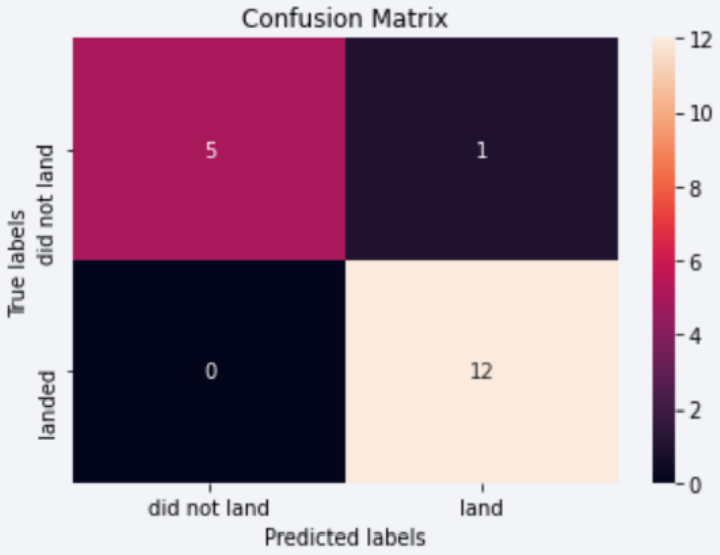
Predictive Analysis (Classification)

Classification Accuracy

- Four classification models were tested, and their accuracies are plotted beside;
- The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%.



Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. Confusion matrix of Decision Tree Classifier proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.

Conclusions

- The larger the number of launches at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.
- Decision Tree Classifier can be used to predict successful landings and increase profits.

Thank you!