IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Daniel Stephen Wamriew
04 – 03 – 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through SpaceX REST API and Web Scrapping from Wikipedia
  - Data Wrangling
  - Exploratory Data Analysis (EDA)
  - Interactive Visual Analysis (IVA) with Folium maps and Plotly Dashboards
  - Predictive Analysis using Machine Learning
- Summary of all results
  - Results of EDA
  - Results of IVA
  - Results of Predictive Analysis

Introduction

- Project background and context
  - SpaceX advertises Falcon 9 rocket launchers in its website at a cost of US$ 62 million compared to its competitors' upward price of over US$ 165 million. Their secret? They can reuse the first landing
  - The goal of this project is therefore to determine if the first stage of Falcon 9 will land successfully, and thereby make it possible to estimate the cost of a launch.
- Problems you want to find answers
  - Will the first stage of Falcon 9 land successfully?
  - What conditions are necessary for a successful landing of Falcon 9 rocket launchers?
  - What properties contribute to the success or failure in landing?

Section 1

Methodology

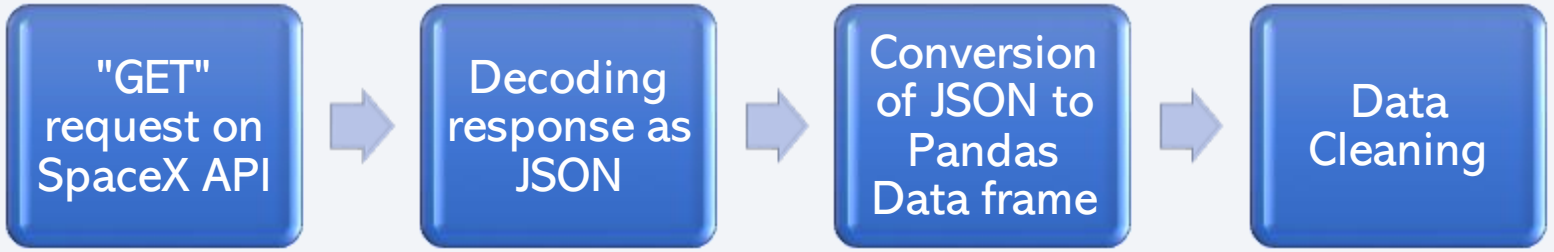
Methodology

Executive Summary

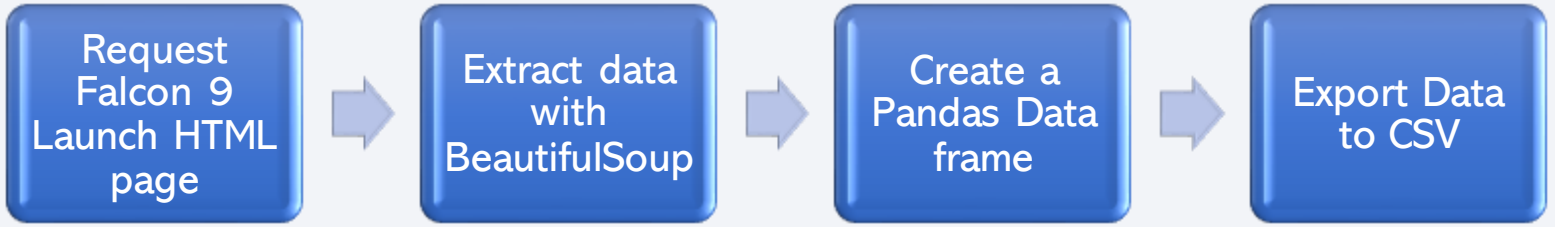
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia.
- Perform data wrangling
  - Filtering to remove unnecessary values/ columns
  - Dealing with missing values
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Building, tuning, and evaluating classification models

Data Collection

1. **SpaceX REST API**
   - Provides launch data, including: rocket types (Booster names), launch pads, payloads and cores.
   - The API URL is : https://api.spacexdata.com/v4/launches/past



2. **Web Scrapping Falcon 9 Launch Records from Wikipedia**
   - Wikipedia provides Falcon 9 and Falcon heavy launches data
   - URL: https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches



Data Collection – SpaceX API

GET request on SpaceX REST API

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

response = requests.get(spacex_url)
```

Decode Response as JSON and convert to Pandas Data Frame

```
response = requests.get(static_json_url).json()
# rjson = response.json()
data = pd.json_normalize(response)
```

Data Cleaning

```
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value ir
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the c
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

Github link:
https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/11_Data-Collection-API.ipynb

Data Collection – Scraping

Request Falcon 9 Launch HTML page

```
response = requests.get(static_url)
```

Extract Data with BeautifulSoup

```
soup = BeautifulSoup(response.text, 'html.parser')
```

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

```
for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Create a Pandas Data Frame

```
df=pd.DataFrame(launch_dict)
```

Github link:

https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/12_Web-Scraping.ipynb

Data Wrangling

- The aim was to identify patterns in the data and determine the labels for training supervised models.
- There were instances of successful, as well as failed landings:
  - True Ocean, True RTLS, True ASDS are successful landings
  - False Ocean, False RTLS, False ASDS are failed landings.

1. Calculate the number of launches in each site

```
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. Calculate the number of occurrences of each orbit

```
df['Orbit'].value_counts()
```

```
GTO      27      MEO      3
ISS      21      ES-L1    1
VLEO     14      HEO      1
PO        9      SO       1
LEO       7      GEO      1
SSO       5
```

3. Calculate number of outcomes per orbit type

```
landing_outcomes = df['Outcome'].value_counts()
```

```
landing_outcomes
```

```
True ASDS      41          True Ocean      5
None None      19          False Ocean     2
True RTLS      14          None ASDS       2
False ASDS      6          False RTLS      1
```

4. Create labels for the outcomes column

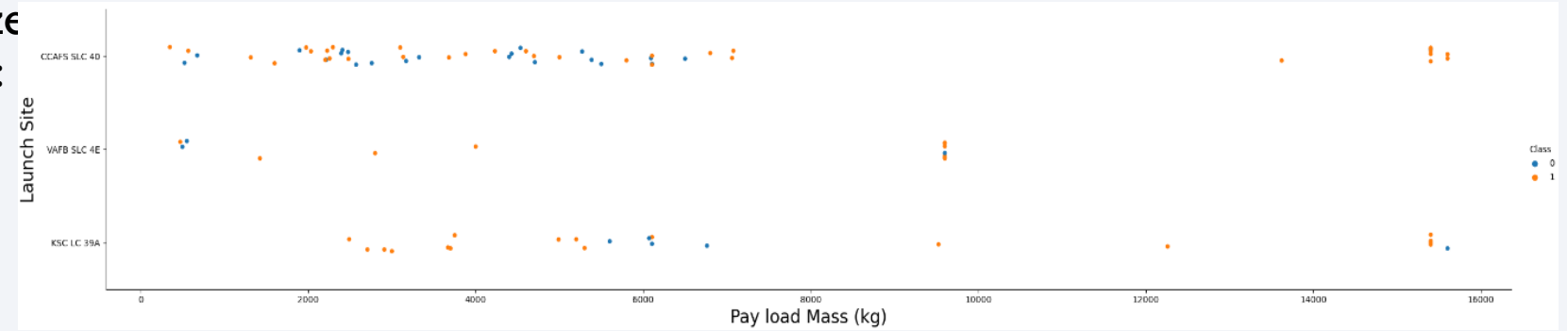
```
landing_class= []
for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
print(landing_class)
```

```
[0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 1,
```

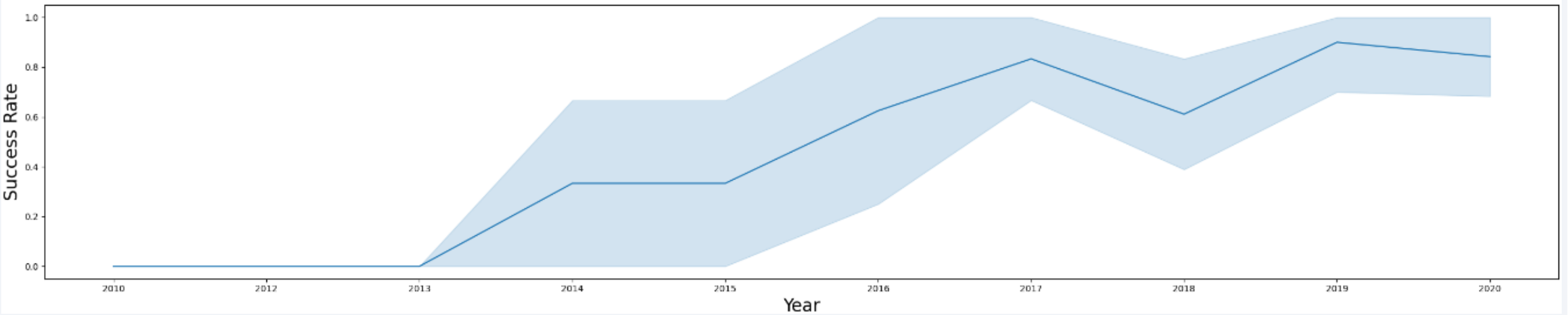

Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/13_Data_Wrangling.ipynb

EDA - Data Visualization

1. 5 scatter plots were used to visualize correlation between two variables, i.e:
    - Flight number vs Launch Site
    - Payload vs Launch Site
    - Success rate vs Orbit type
    - Flight number vs orbit type
    - Payload vs orbit type



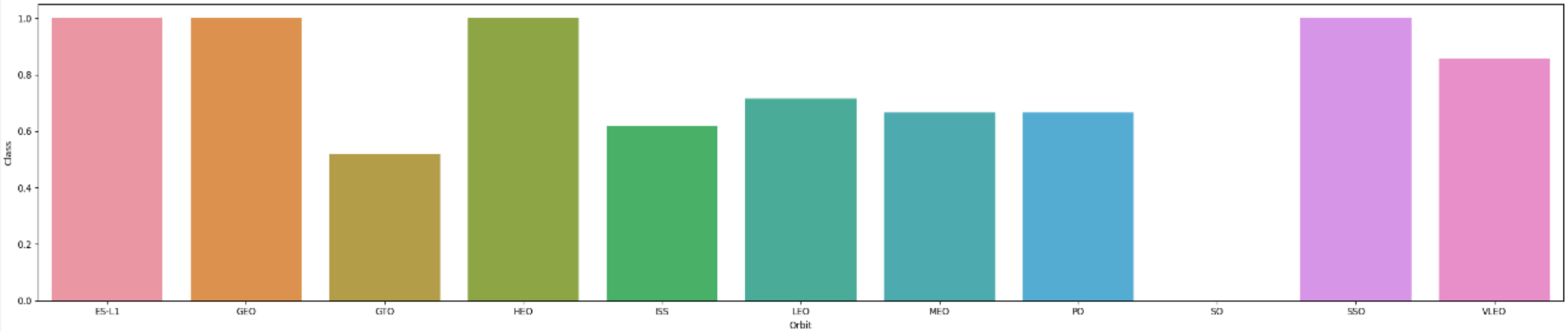
2. Line plot was used to visualize trend of success rate over a period of 11 years from 2010 to 2020



Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/22_EDA-Data-Visualization.ipynb

EDA - Data Visualization

3. A bar chart was plotted to visualize the relationship between success rate of each orbit types. Bar graphs are ideal for comparing categorical and numerical variables.



Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/22_EDA-Data-Visualization.ipynb

EDA with SQL

The following SQL queries were performed to draw insights into the data:

- Display the names of unique launch sites in the space mission
- Display 5 records where the launch site begun with "CCA"
- Display the total payload mass carried by booster 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. Use a subquery
- List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/21_EDA-SQL-sqllite.ipynb

Build an Interactive Map with Folium

Markers, Circles, MarkerCluster and Lines were used to build interactive map:

- **Markers** – used to mark points, e.g:
  - Mark launch sites
  - Mark successful and failed landings: Green for success; Red for Failure
  - Mark points between launch sites and key points such as railway lines, highways, cities, coastlines e.t.c
- **Circles** – used to highlight a specified radius around areas of interest
  - Red circle at each launch site
  - Red circle at NASA John Space Centre
- **MarkerCluster** – used to group events in a cluster at each launch site
- **Lines** – used to show distance between two points, e.g:

  -Distance between launch site and nearest coastline.

Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/31_Interactive_Map_Folium.ipynb

Build a Dashboard with Plotly Dash

- **Built a dashboard comprising of a dropdown menu, range slider, pie chart and scatter plot:**
  - **Dropdown menu** – Allows user to ALL launch sites, or a specific launch site
  - **Range Slider** – Allows user to choose a payload mass in a fixed range interval
  - **Pie Chart** – Shows percentages of successful launches for ALL launch sites or a specific launch site.
  - **Scatter Plot** - Shows the correlation between payload and success.

Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/32_spacex_dash_app.py

Predictive Analysis (Classification)

Models Building

- Create an array of labels
- Standardize the data
- Split the data into training and testing sets
- Create a Classification object
- Create a GridSearchCV object
- Fit the data/ train the model

Models Evaluation

- Calculate accuracy of each model on validation data
- Output best hyperparameter of each model
- Calculate accuracy of each model on test data
- Plot a Confusion Matrix for each model

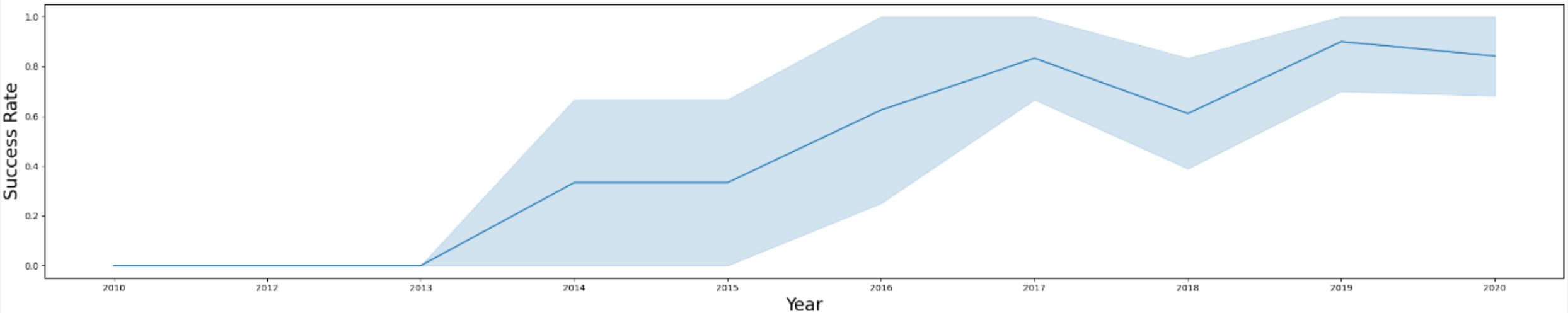
Models Comparison

- Compare the accuracy of each model on validation data
- Compare the accuracy of each model on test data
- Split the data into training and test data
- Calculate the average accuracy on each model
- Choose the best model

Github link: https://github.com/wamriewdan/IBM_Applied_Data_Science_Capstone/blob/main/41_Machine_Learning_Prediction.ipynb

Results

1. Exploratory data analysis results



2. Predictive analysis results

Sample prediction results for Support Vector Machine (SVM)

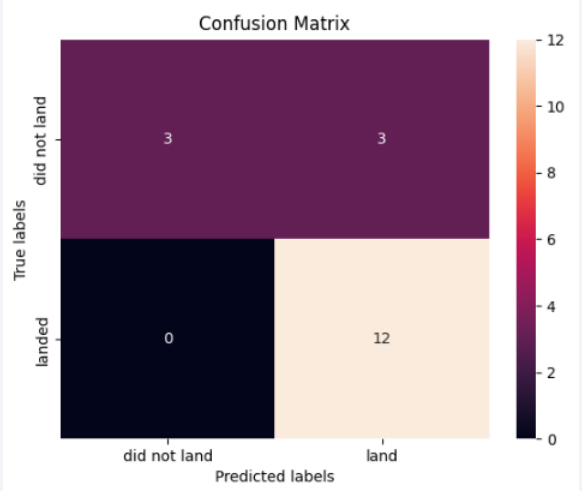
```
print("tuned hpyerparameters :(best parameters) ",svm_cv.best_params_)
print("accuracy :",svm_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'C': 1.0, 'gamma': 0.03162277660168379, 'kernel': 'sigmoid'}
accuracy : 0.8482142857142856
```

Calculate the accuracy on the test data using the method `score`:

```
print("Accuracy on test data :", svm_cv.score(X_test, Y_test))
```

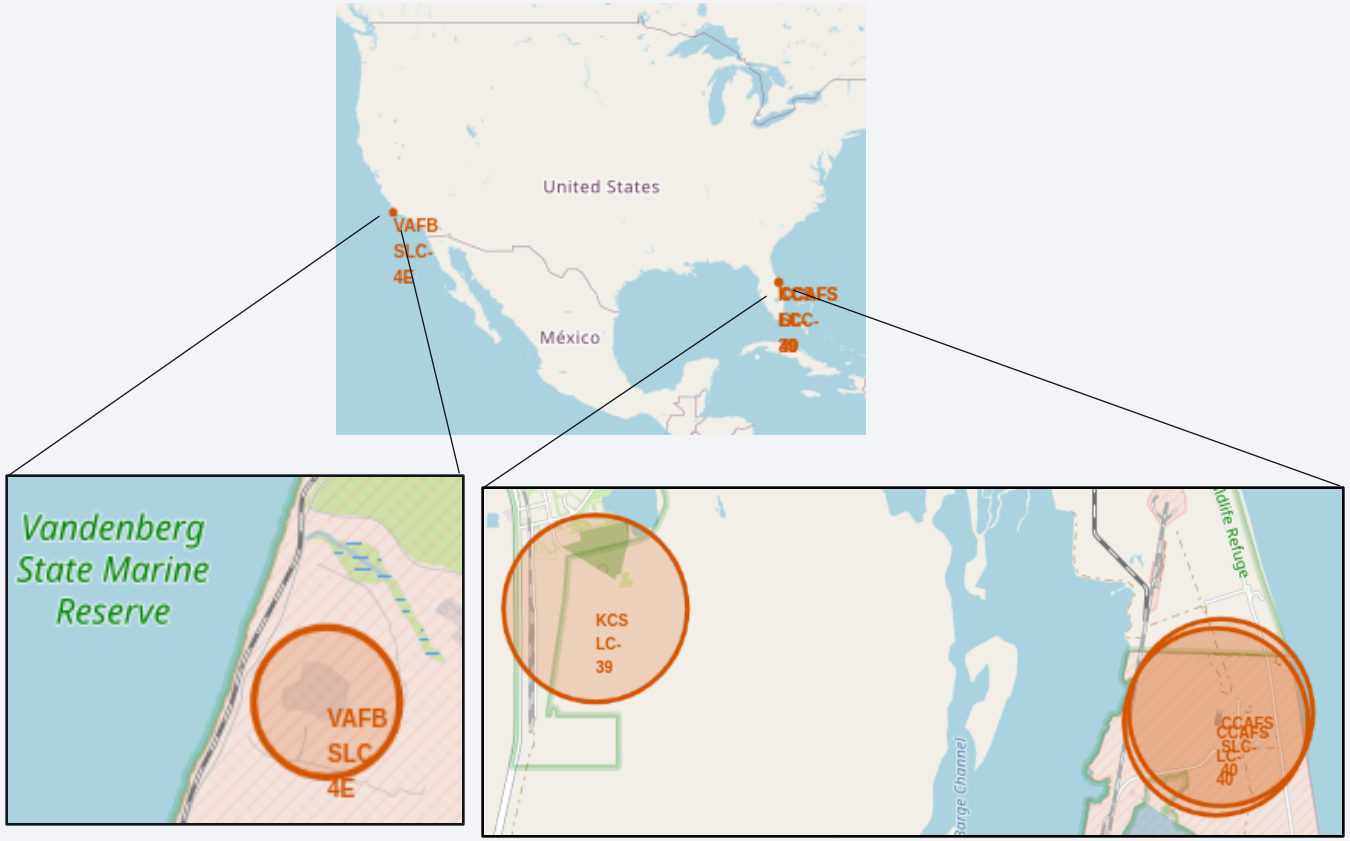
```
Accuracy on test data : 0.8333333333333334
```



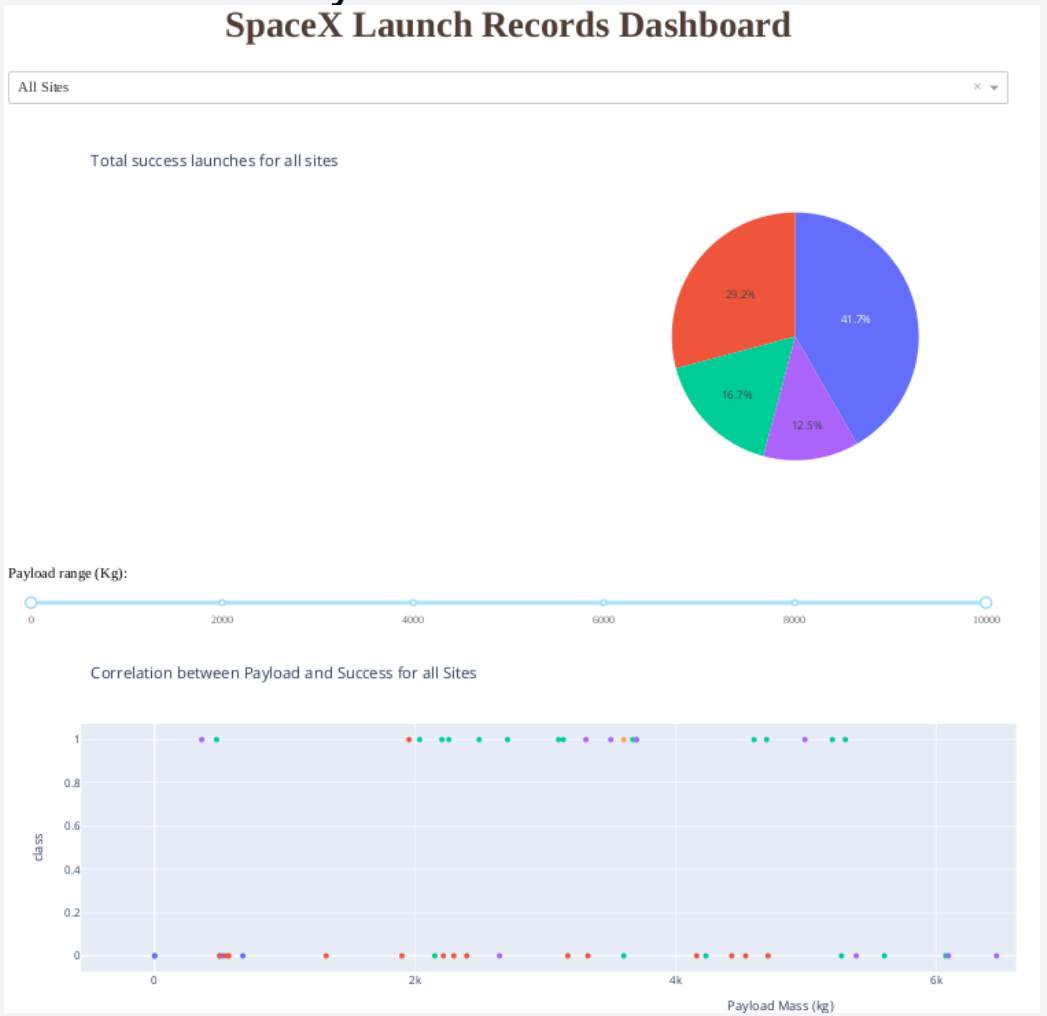
Results

3. Interactive analytics demo in screenshots

3.1 Launch sites on Folium Maps

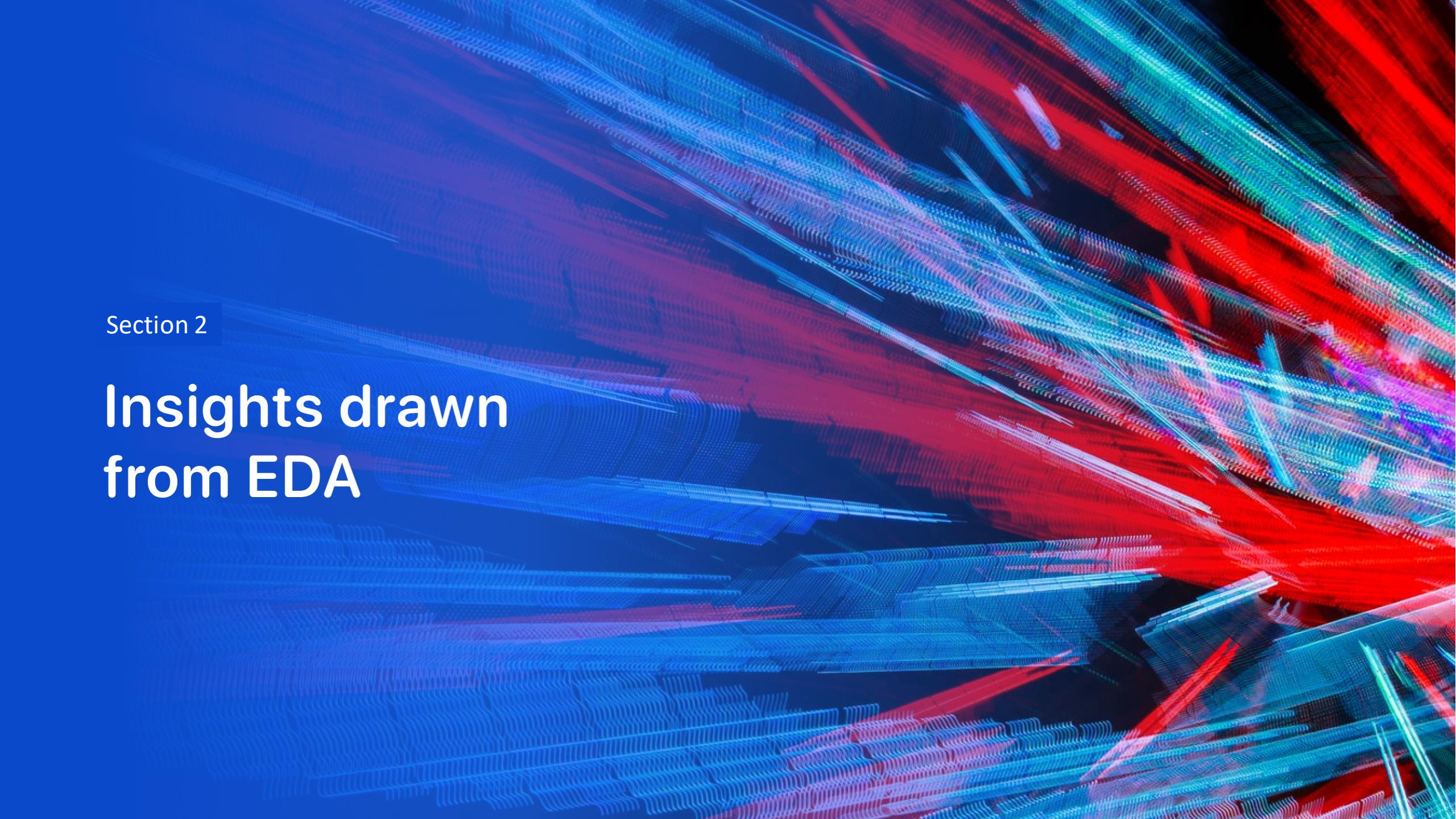


3.2 Plotly Dashboard screenshots

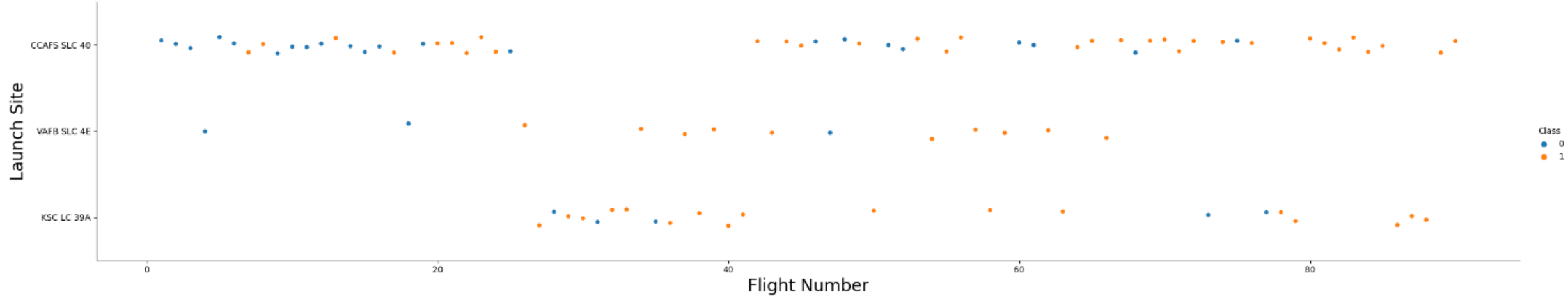


Section 2

Insights drawn from EDA

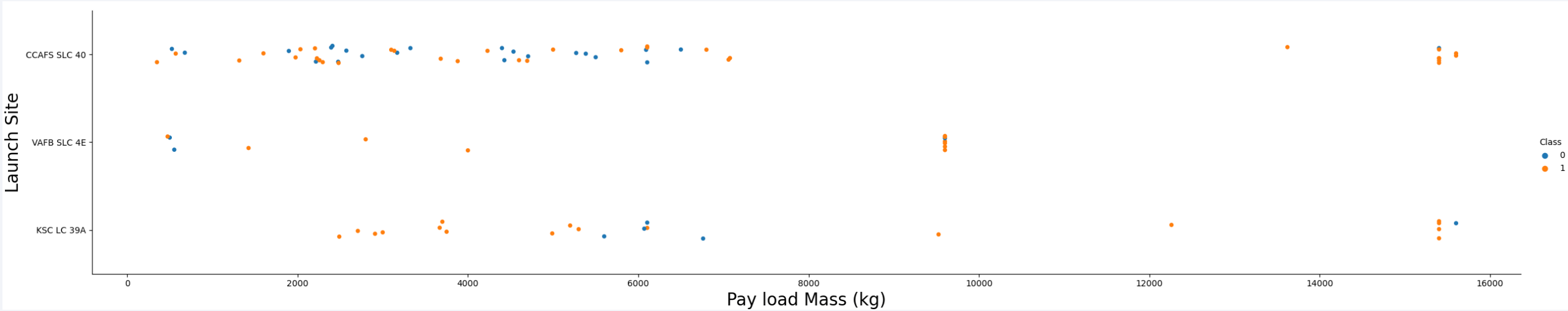


Flight Number vs. Launch Site



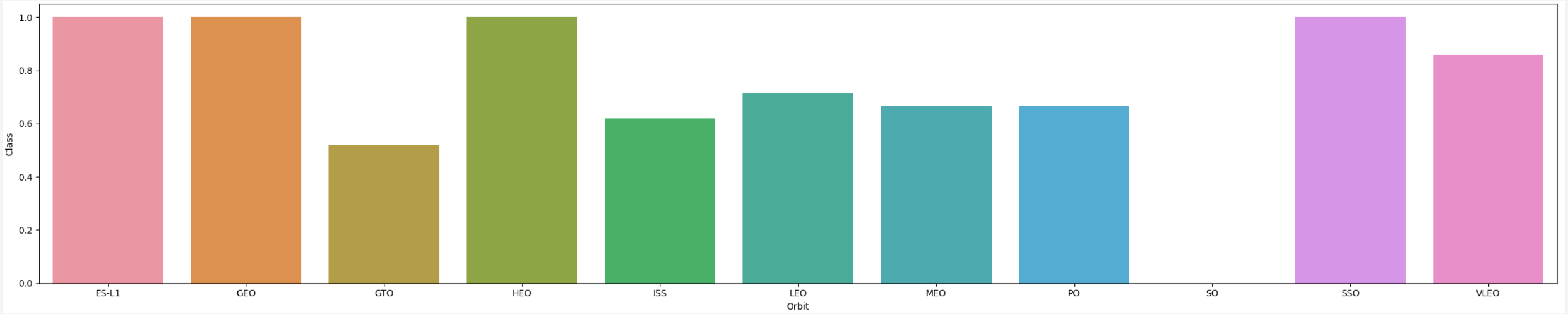
- From the scatter plot, we see that KSC LC-39A launch site has the highest success landing rate.
- We also see that the success rate increases with the increase in flight numbers

Payload vs. Launch Site



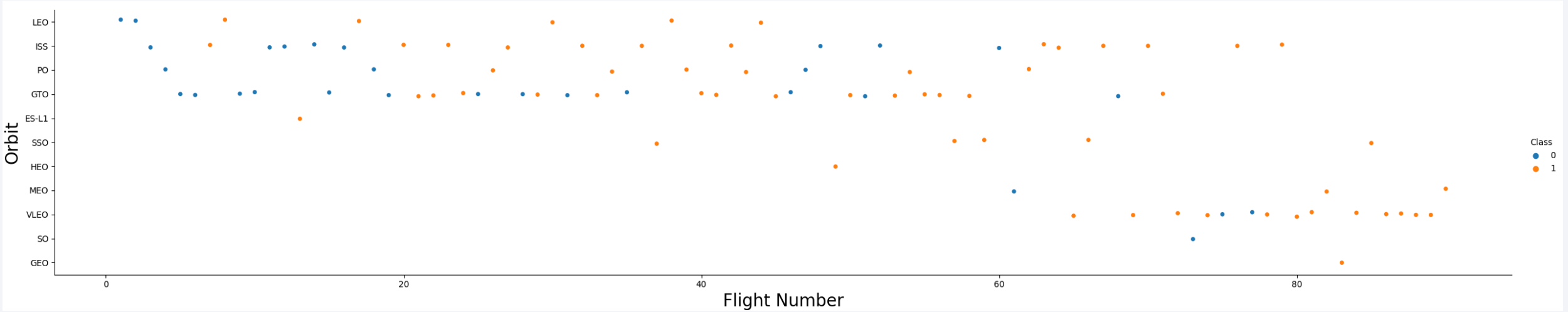
- From the scatter plot, we see that the heavy payloads with mass above 8000 kg have a higher success rate compared to lighter payloads.
- We also see that there are more successful landings from launch site CCAFS SLC-40 with payload mass above 8000 kg.
- There are fewer launches with payload mass above 8000 kg across all launch sites.

Success Rate vs. Orbit Type



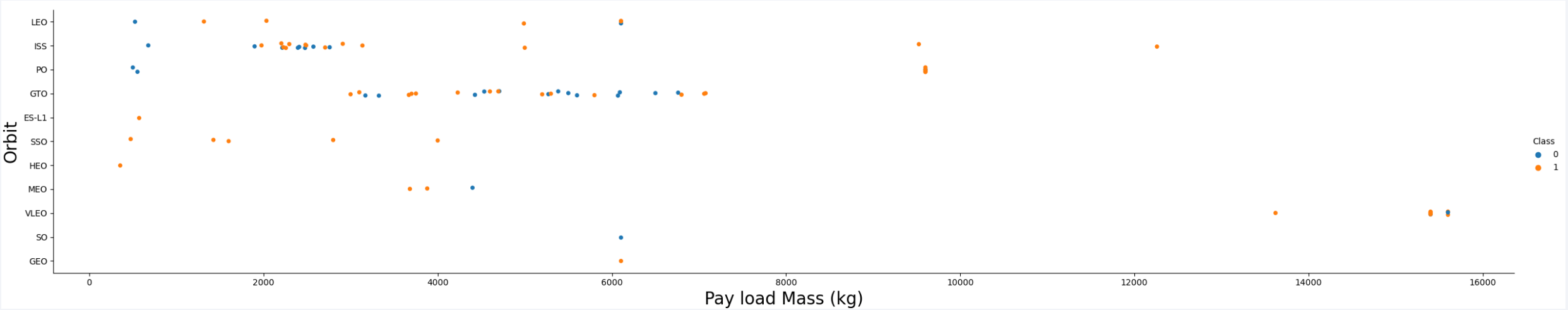
- From the bar chart, we see that ES-L1, GEO, HEO and SSO have the highest success rates
- Orbit SO has no success rate, while MEO and PO have equal success rates.

Flight Number vs. Orbit Type



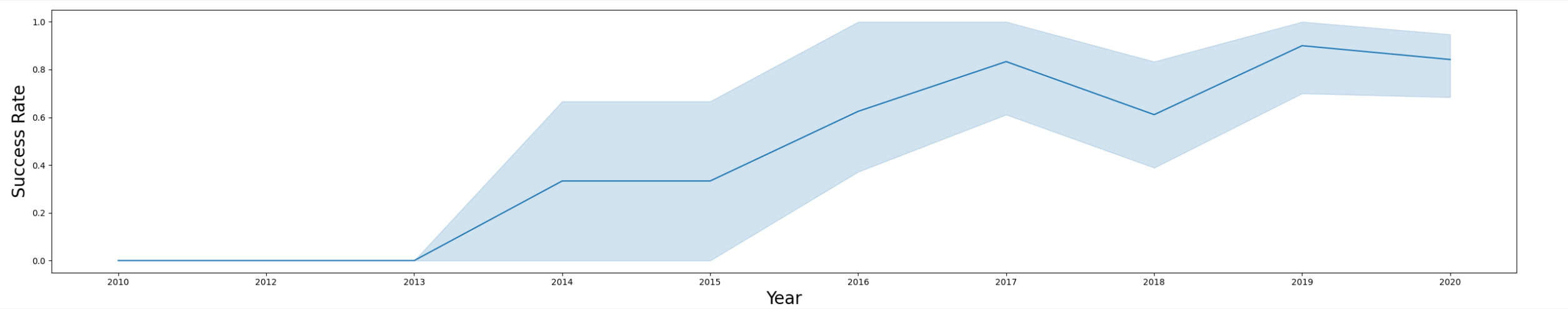
- From the scatter plot, we see that in the LEO orbit the success appears to be related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.
- Only one rocket has been launched to SO (unsuccessful) and GEO (successful) orbits

Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both successful landing rate and unsuccessful landing are both there here.

Launch Success Yearly Trend



- From the line graph, we can observe that the success rate has kept increasing since 2013 till 2020

All Launch Site Names

```
%%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

- DISTINCT was used to remove any Duplicates.
- There are only 4 unique launch sites for SpaceX

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT "Launch_Site"
FROM SPACEXTBL
WHERE "Launch_site" LIKE 'CCA%' LIMIT 5
```

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

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- LIKE clause is used to specify the required substring `CCA`
- LIMIT clause limits the search results to the specified number (5)

Total Payload Mass

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

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

SUM(PAYLOAD_MASS__KG_)
45596

- SUM operator adds all the entries in the column 'PAYLOAD_MASS__KG_'
- WHERE clause filters the rows where customer is NASA (CRS)

Average Payload Mass by F9 v1.1

```
%%sql
SELECT  AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE "Booster_Version" = 'F9 v1.1'
```

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

AVG(PAYLOAD_MASS__KG_)
2928.4

- AVG function calculates the arithmetic mean of the column PAYLOAD_MASS__KG_
- WHERE clause filters the booster version to F9 v1.1

First Successful Ground Landing Date

```
%%sql
SELECT "Date" FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (ground pad)"LIMIT 1
```

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

Date
22-12-2015

- WHERE clause only returns the substring 'Success' from the column 'Landing_Outcme'.
- LIMIT clause limits the output to 1.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT "Booster_Version", PAYLOAD_MASS__KG_ AS "Payload Mass"
FROM SPACEXTBL
WHERE "Landing _Outcome" = "Success (drone ship)" AND (PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000)
```

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

Booster_Version	Payload Mass
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

- WHERE and AND clauses have been used to constrain the search to the defined conditions.
- There are four boosters in the defined range.

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT (SELECT COUNT("Mission_Outcome") FROM SPACEXTBL WHERE "Mission_Outcome" LIKE '%Success%') AS 'SUCCESS',
(SELECT COUNT( "Mission_Outcome") FROM SPACEXTBL WHERE "Mission_Outcome" LIKE '%Failure%') AS 'FAILURE'
```

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

SUCCESS	FAILURE
100	1

- Subqueries are used to select to total 'Success' and 'Failure' from 'Mission_Outcome' column
- AS clause is used to alias the result to 'SUCCESS' and 'FAILURE' respectively.

Boosters Carried Maximum Payload

```
%%sql
SELECT "Booster_Version", PAYLOAD_MASS__KG_ AS "Maximum Payload Mass (Kg)"
FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
```

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

Booster_Version	Maximum 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

- Subquery is used to select the maximum payload mass from PAYLOAD_MASS__KG_ column using the MAX function
- There are 12 different booster version with the maximum mass of 15600 kg

2015 Launch Records

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

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

substr("Date", 4, 2)	Landing _Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Clauses **substr(Date, 4, 2)** and **substr(Date,7,4)='2015'** have been used to select the month and year respectively, since sqlite does not support the date function.
- There are two failure landings in the drone ship in the specified time period.

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

```
%%sql
SELECT "Date","Landing _Outcome"
FROM SPACEXTBL
WHERE "Landing _Outcome" LIKE '%Success%' AND (substr("Date",7,4)||substr("Date",4,2)||substr("Date",1,2) BETWEEN '20100604' AND '20170320')
ORDER BY substr("Date",7,4)
```

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

Date	Landing _Outcome
22-12-2015	Success (ground pad)
08-04-2016	Success (drone ship)
06-05-2016	Success (drone ship)
27-05-2016	Success (drone ship)
18-07-2016	Success (ground pad)
14-08-2016	Success (drone ship)
14-01-2017	Success (drone ship)
19-02-2017	Success (ground pad)

- The pipes (||) have been used to reverse the date format from dd-mm-yyyy to yyyy-mm-dd to make it readable for the query.
- There are 8 records in the specified period of time.


Section 3
Launch Sites
Proximities Analysis

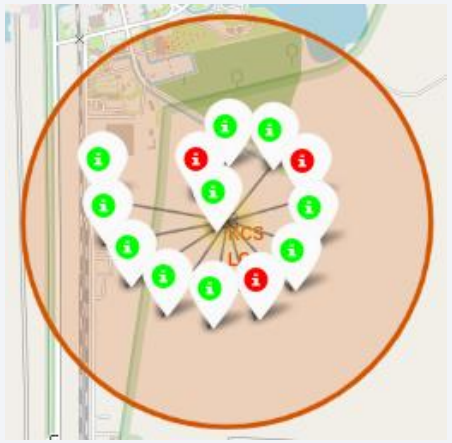
Map of All Launch Sites

- All the four launch site are located within the United States of America
- All the launch sites are located near the coastline for safety measures.

Color-labeled launch outcome for each site

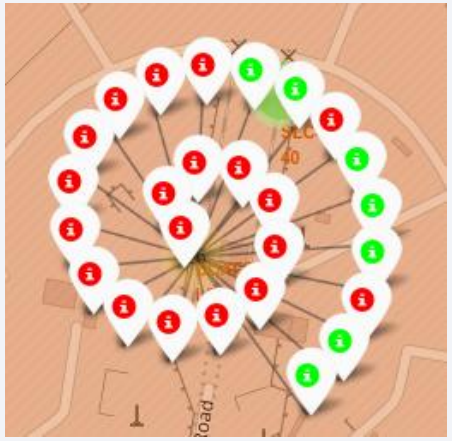
VAFB SLC-4E

KSC LC-39

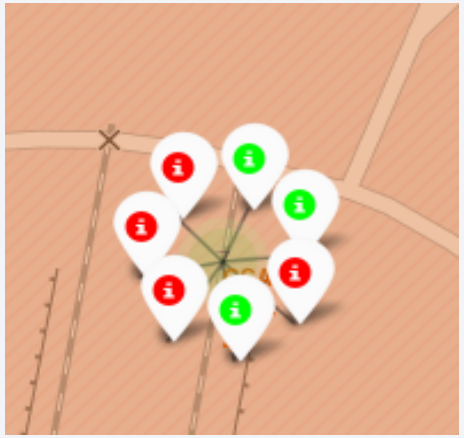


- Green marker – indicates successful landing
- Red marker – indicates failed landing

CCAFS LC-40

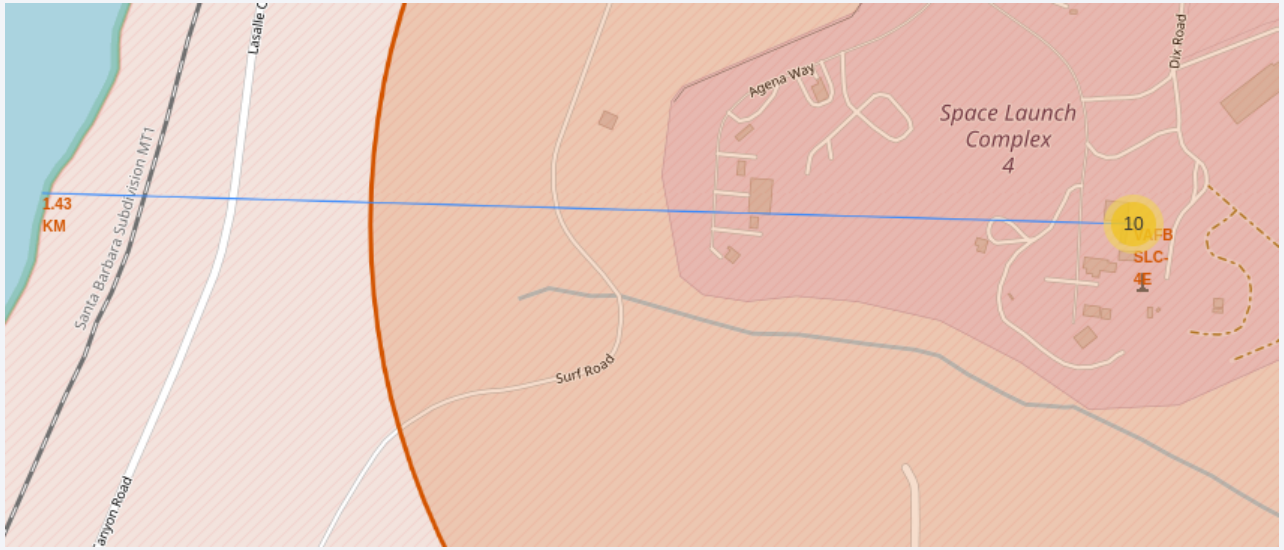


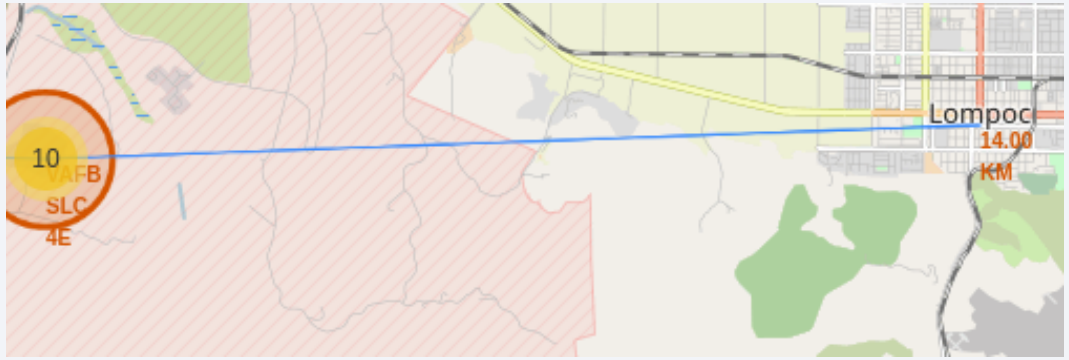
CCAFS SLC-40

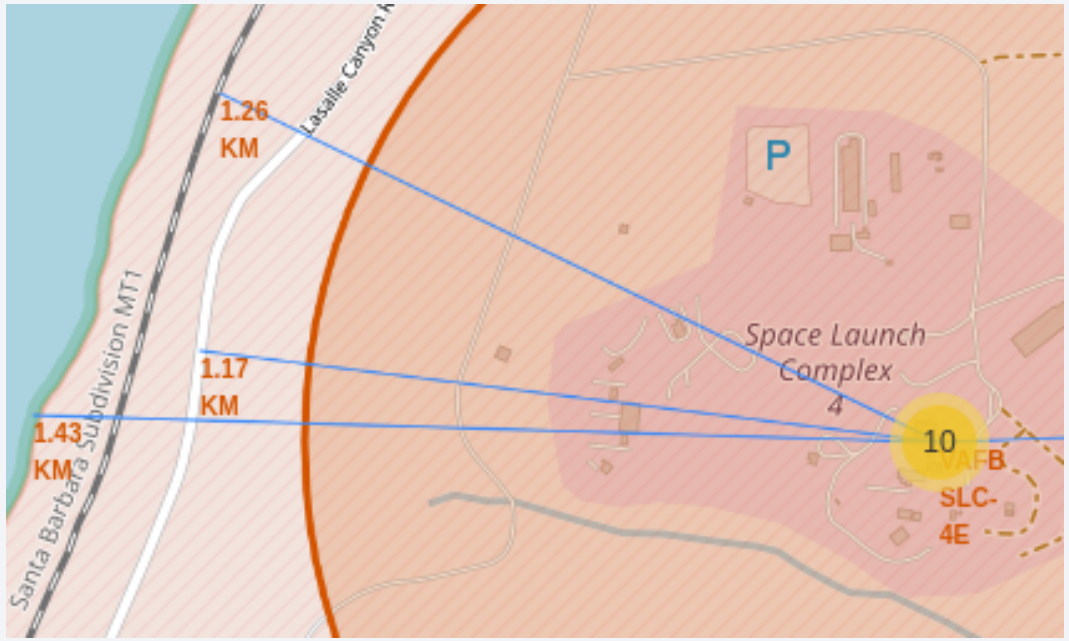


- KSC LC-39 has the highest number of successful launches
- CAFS LC-40 has the most number of launches and the least success rate.

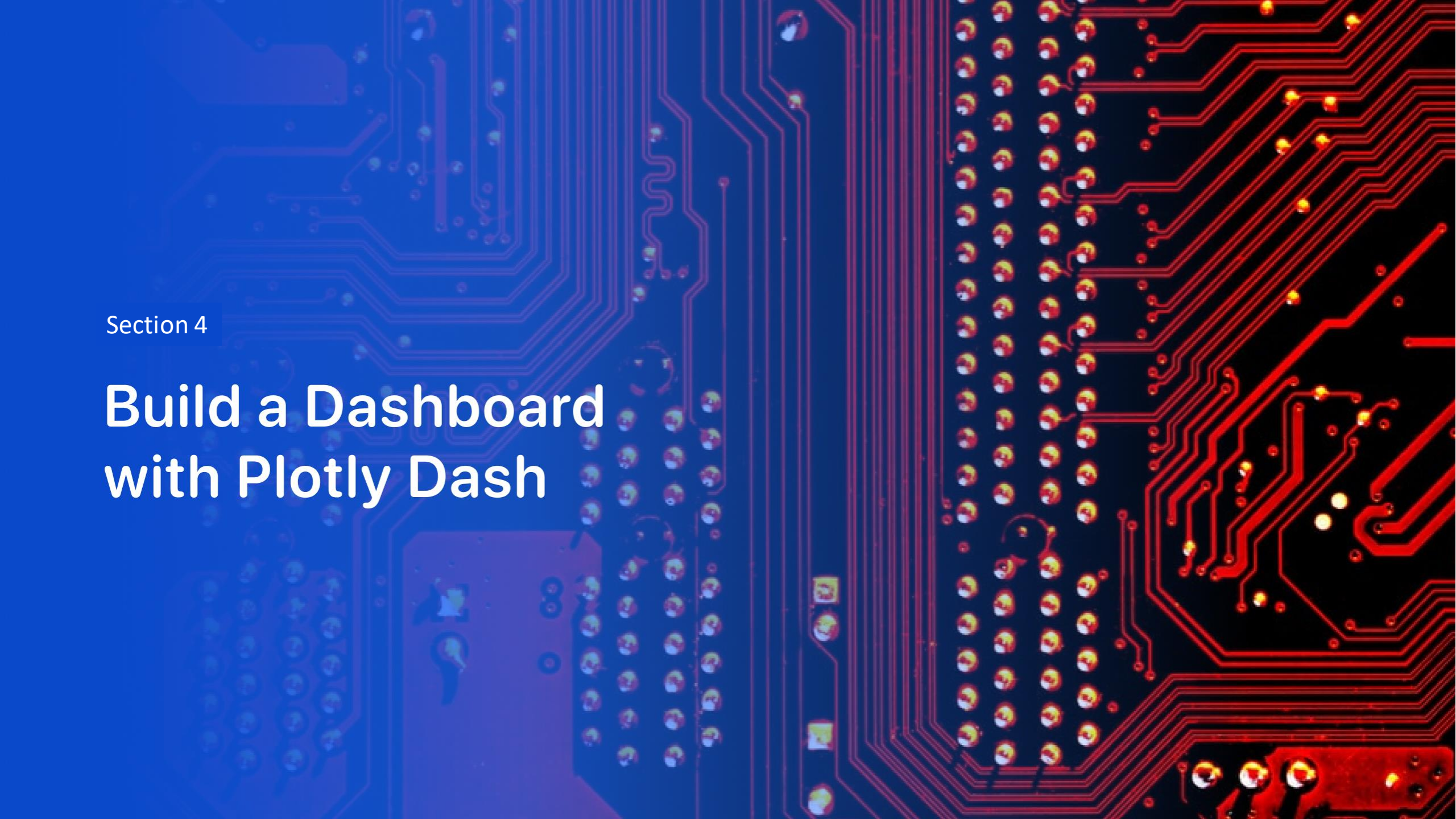
Launch site and its proximities



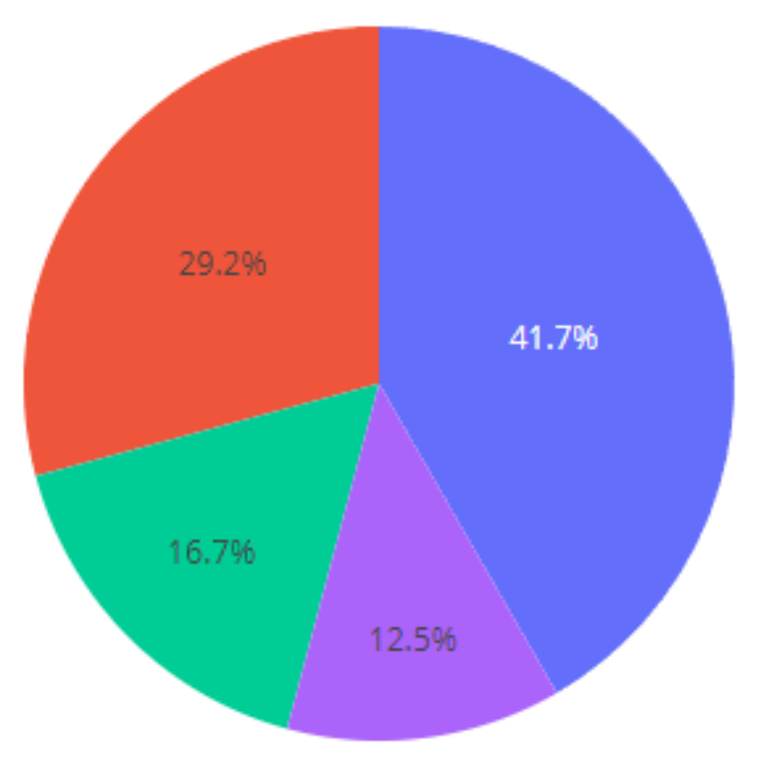




- Launch sites are close to coastlines for safety reasons
- Launch sites are close to railways, and highways for ease of transport
- Launch sites tend to be far from Cities


Section 4
Build a Dashboard with Plotly Dash

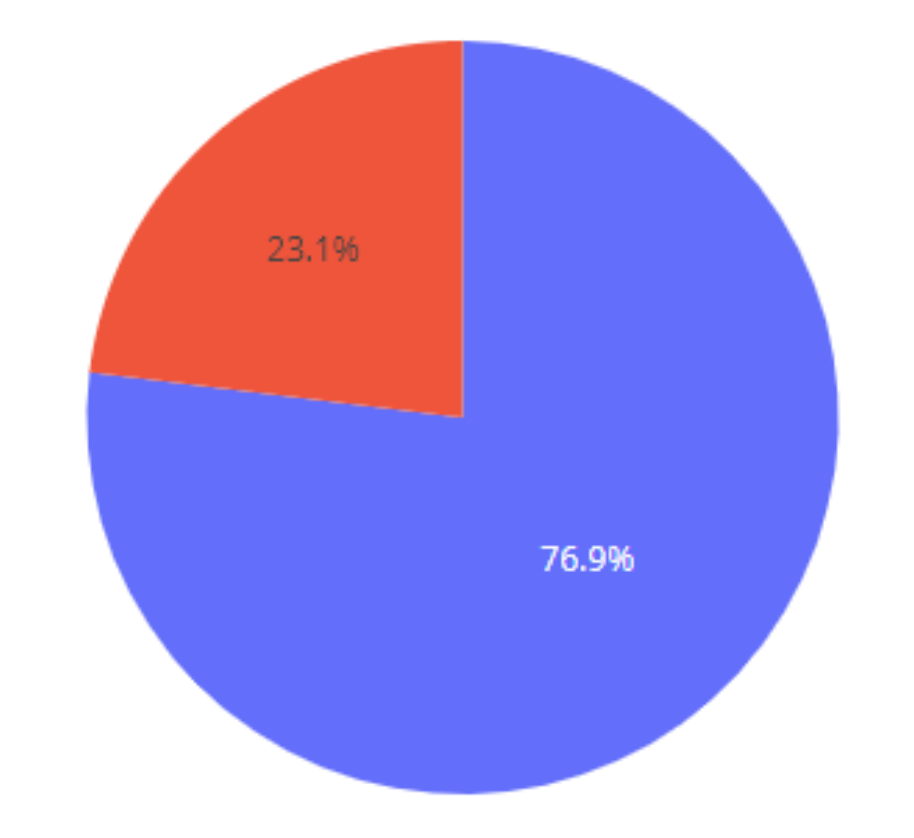
Launch Success Count for All Sites



- Each site is represented by a sector and color coded as shown in the key
- The percentages in each sector of the pie chart shows the successful launch for the site
- From the pie chart, we can see that the site KSC LC-39A has the highest launch success count, followed by CCAFS LC-40, VAFB SLC-4E and CCAFS SLC-40 in that order

Lauch Site With Highest Launch Success Ratio

Total success launches for KSC LC-39A

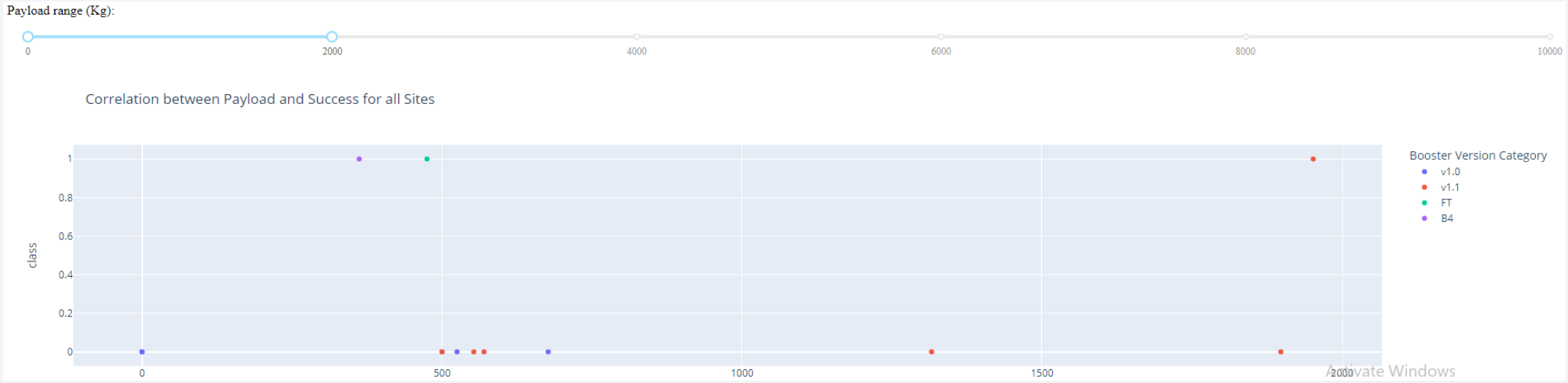


Key:

- **Blue** – Success Launches
- **Red** – Failed Launches

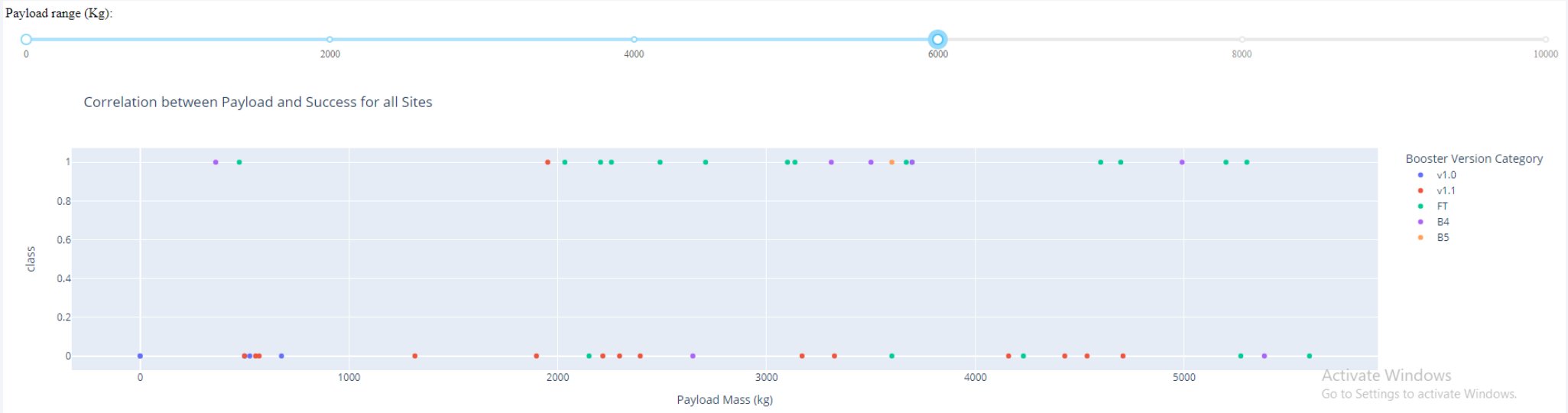
- From the pie chart, we can see that KSC LC_39A launch site has an overall success rate of 76.9% and only 23.1% failed launches.

Scatter Plots of Payload vs Launch Outcome – All Sites



Range glider: 0 – 2000 kg

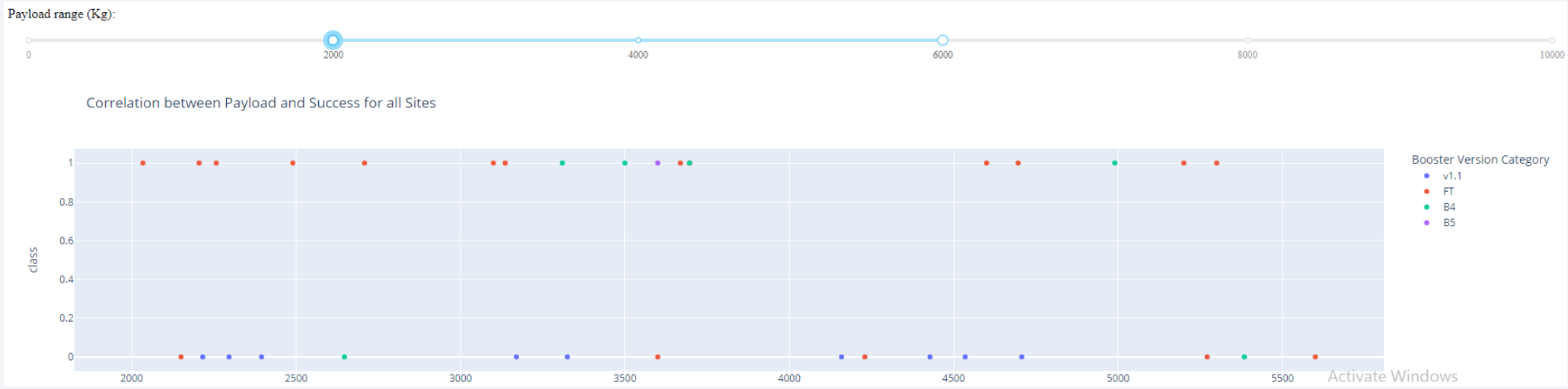
- Fewer boosters were launched mostly Falcon 9 v1.1
- Low success rate



Range glider: 0 – 6000 kg

- Most boosters have a payload between 2000 – 6000 kg
- Booster FT had the highest success landing rate
- Booster v1.1 had the least success rate

Scatter Plots of Payload vs Launch Outcome – Cont'd



Range glider: 2k – 6k kg

- Booster FT had a high success rate in the range
- Booster v1.1 had the least success rate in this range
- Most boosters had payloads in this range.

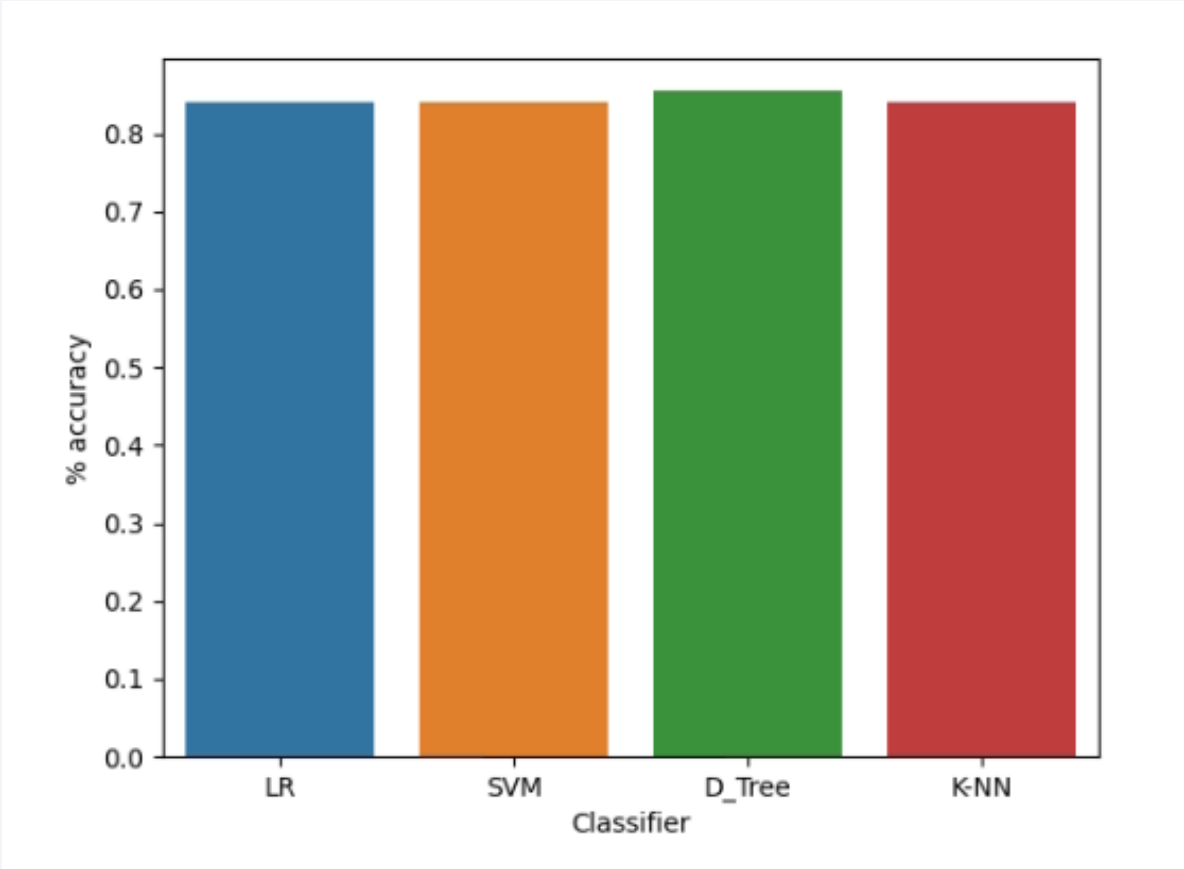
Range glider: 5k – 10k kg

- Only boosters FT and B4 version were launched with payload between 5k and 6.5k kg
- No success for booster FT beyond payload mass of 5.5k kg
- B4 had the heaviest payload mass beyond 9.5k kg

Section 5

Predictive Analysis (Classification)

Classification Accuracy

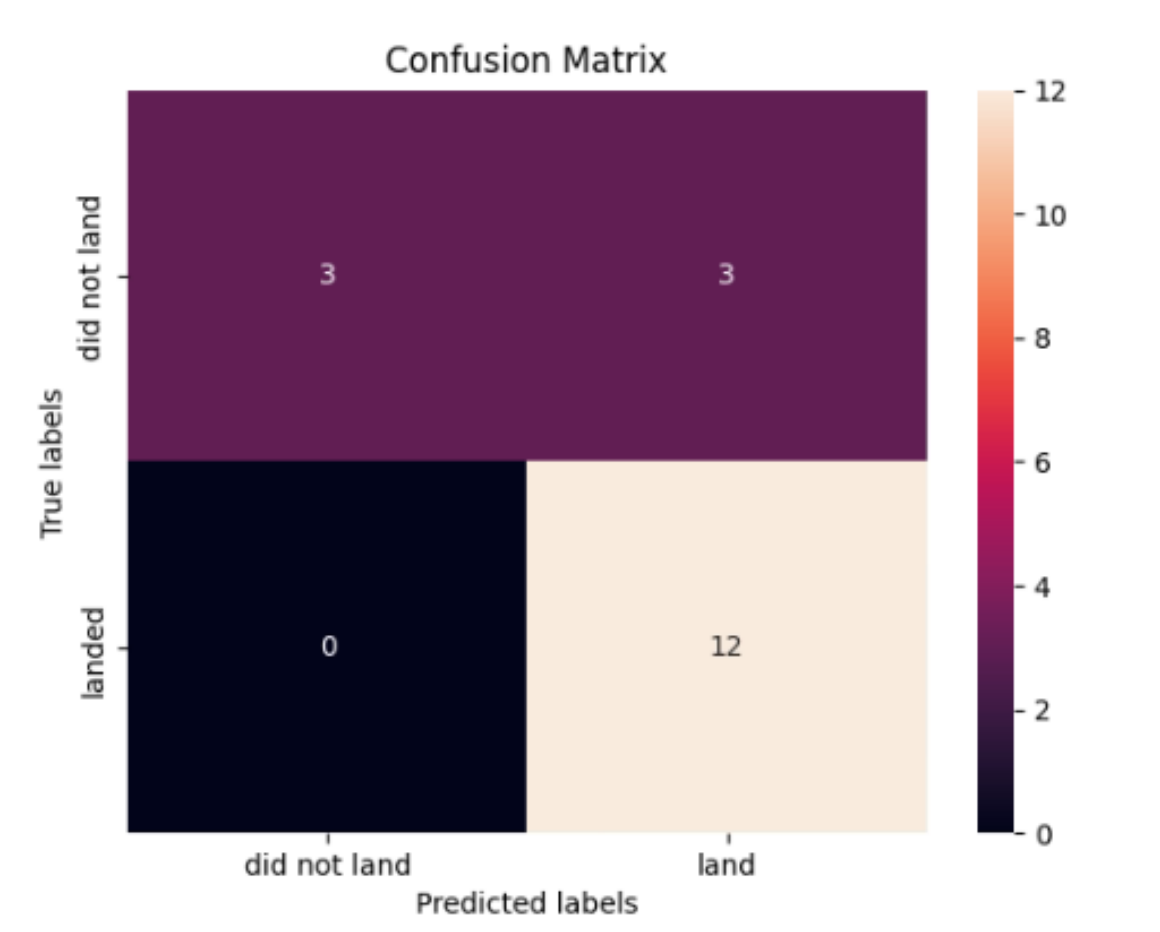


Key:

- LR – Linear Regression classifier
- SVM – Support Vector Machine Classifier
- D_Tree – Decision Trees Classifier
- K-NN – K-Nearest Neighbour Classifier

- From the bar chart, we can see that the **Decision Tree Classifier (D_Tree) has the best average percentage accuracy**
- The average percentage accuracy is computed over the validation and test data set

Confusion Matrix – Decision Tree Model



- From the confusion matrix, we can see that there were a total of 6 failed landings and 12 successful landings in the test dataset.
- The model correctly classified all the successful landings
- The model correctly classified 3 failed landings but misclassified 3.

Conclusions

- Successful landing is dependent on a number of factors including: the payload, flight number, launch site and target orbit.
- Four orbits: ES-L1, GEO, HEO, and SSO have the highest success landing rates.
- KSC LC-39A launch site has the highest success rate compared to others sites.
- Booster version FT has the highest success rate compared to other boosters
- Low payload mass, up to 6000 kg has higher success rate than heavier payloads
- Decision Tree is the best classification model with higher average accuracy over both validation and test data.

Thank you!