IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

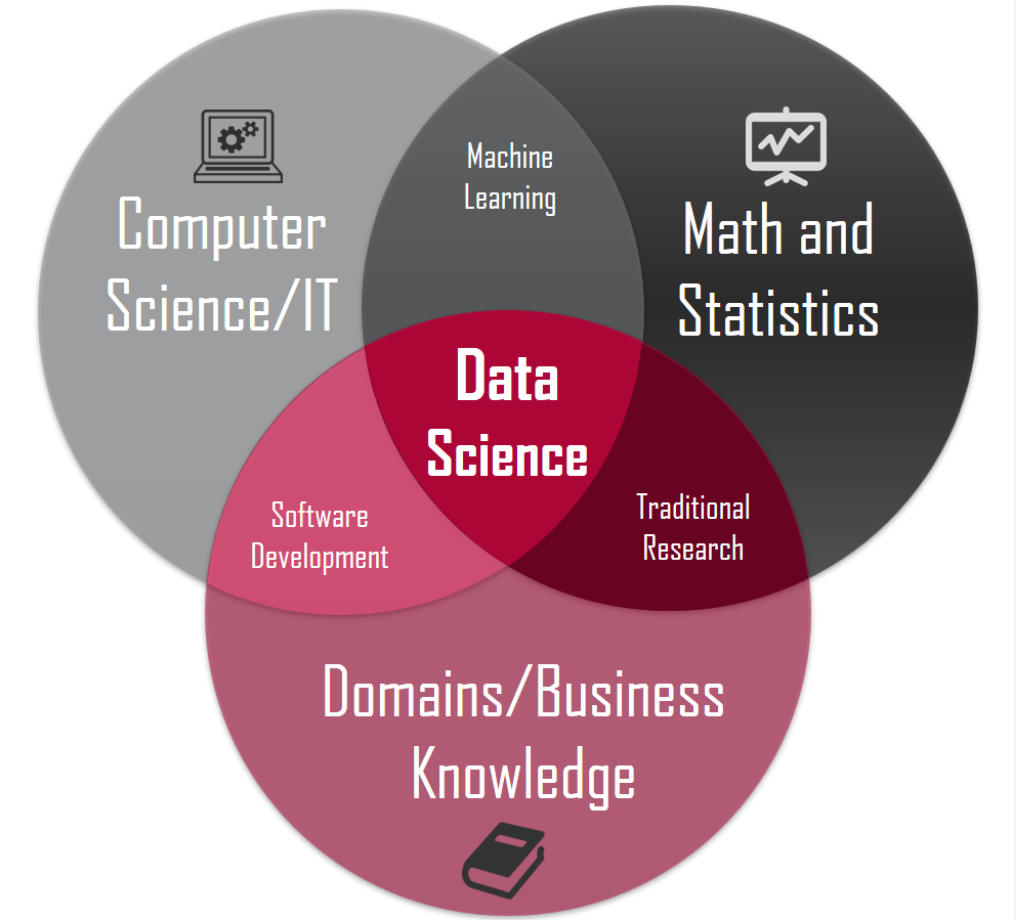
Mahmoud AlFares
10 Feb 2023

Outline

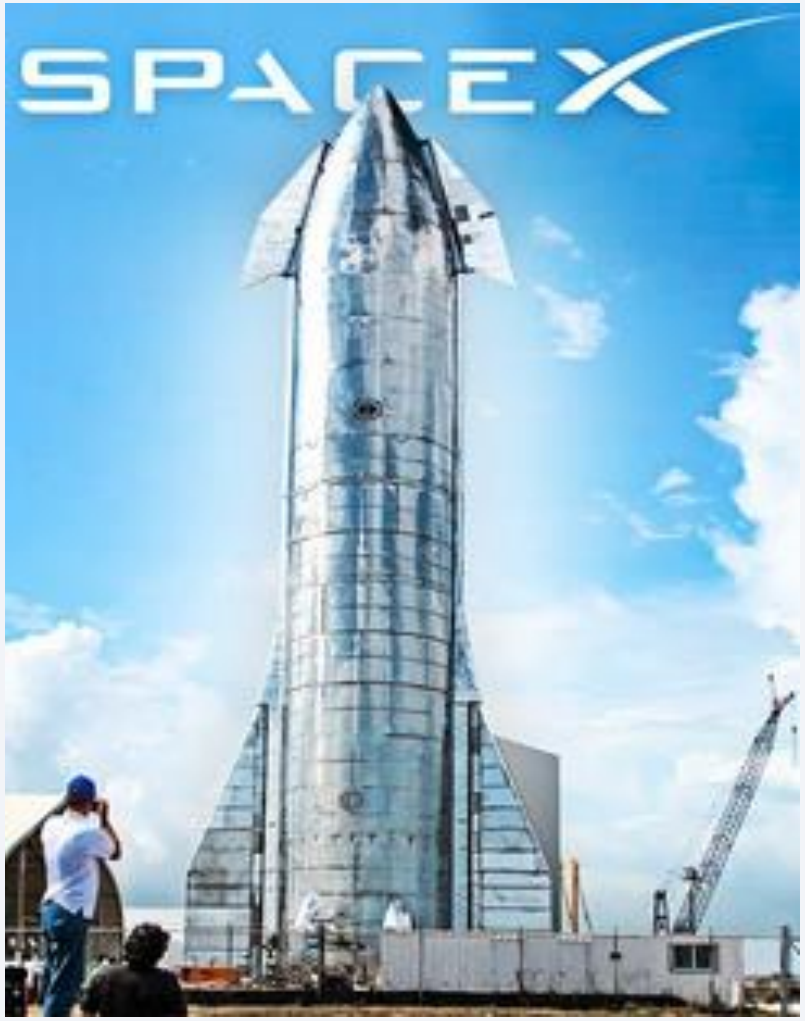
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

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

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. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully

Problems you want to find answers

-What factors determine if the rocket will land successfully?

-The interaction amongst various features that determine the success rate of a successful landing.

-What operating conditions needs to be in place to ensure a successful landing program.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

In this project we working on Data Collection using various methods as following:

1. First, using get request to the SpaceX API.
2. Next, we decoded the response content as a Json using .json().
3. Then turn it into a pandas dataframe using .json_normalize().
4. After that, we cleaned the data, checked for missing values and fill in missing values where necessary.
5. In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
6. Finally, we extracted the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%201A-%20Data%20Collection%20API%20Lab%20.ipynb

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

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

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

Check the content of the response

In [8]:
```
print(response.content)
```

```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},
_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign
l":[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00n
196-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikiped
0Z","static_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda699
reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of
3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","flight_number":1,"name":"Falco
date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"core
legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_typ
ary_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_at
l":"https://images2.imgbox.com/f9/4a/ZboXReNb_o.png","large":"https://images2.imgbox.c
edia":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webca
Q2wP-Nc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbi
_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9
tude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}],"deta
```

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup, then we parsed the table and converted it into a pandas dataframe.
- The link to the notebook is
- https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%201B%20-%20Web%20Scraping.ipynb

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

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

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

```
[n [8]:
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response, 'html.parser')
```

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

```
[n [9]:
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

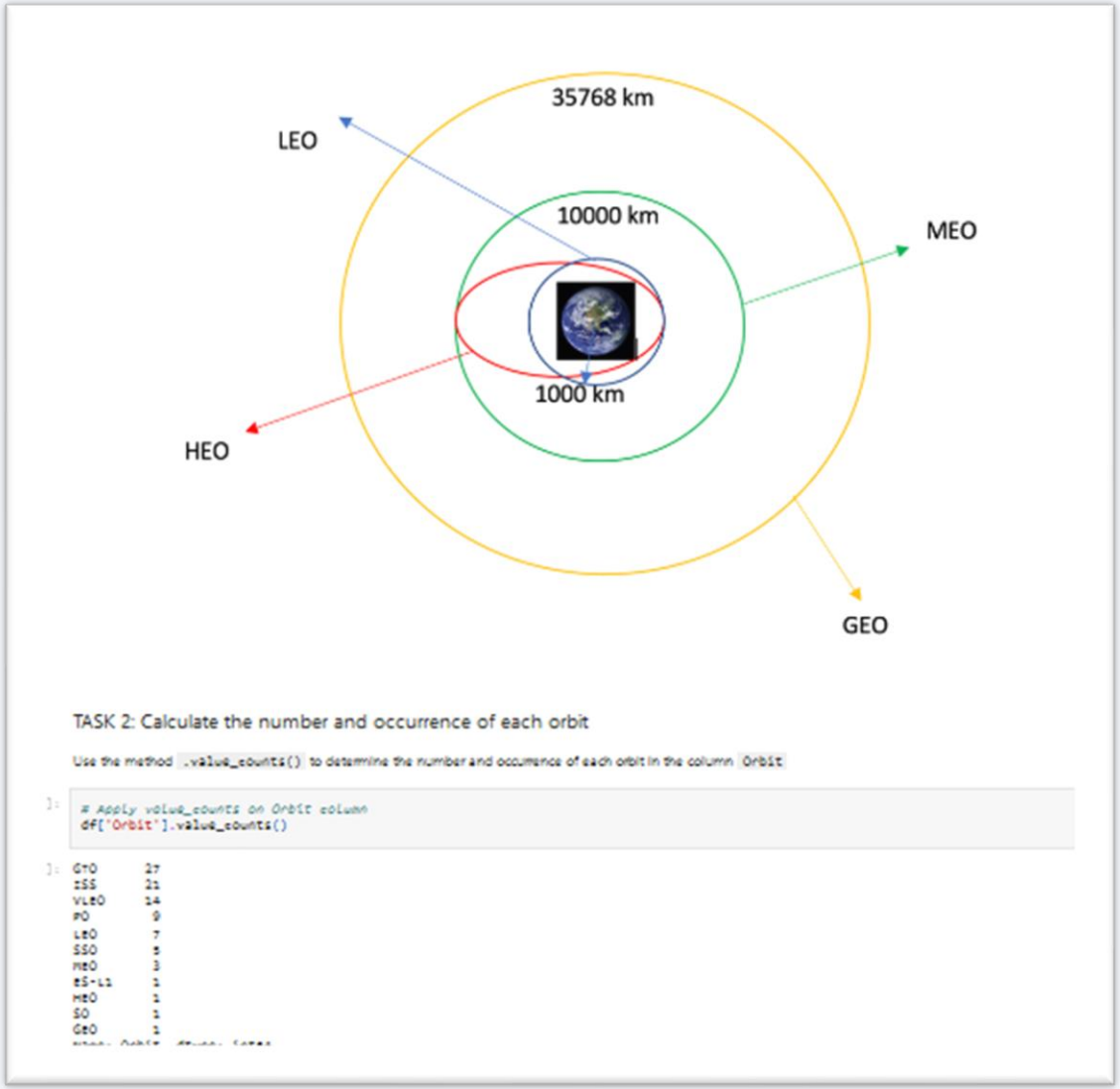
Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSo` lab

```
[11]:
# 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")
```

Data Wrangling

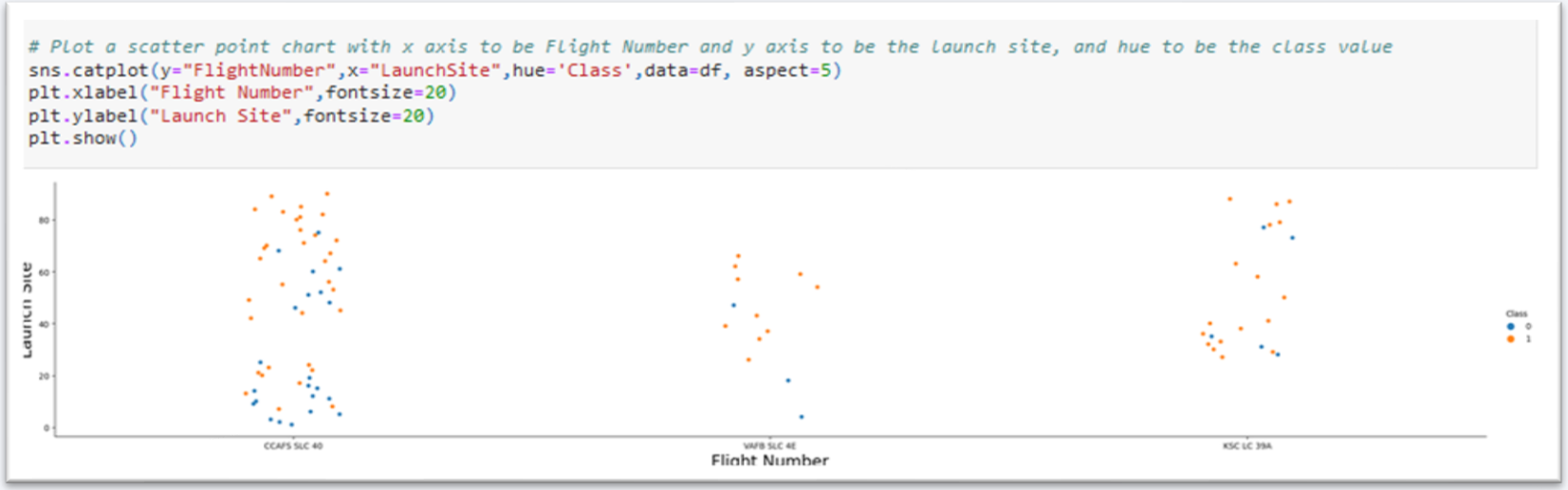
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%201C%20-%20Data%20Wrangling.ipynb

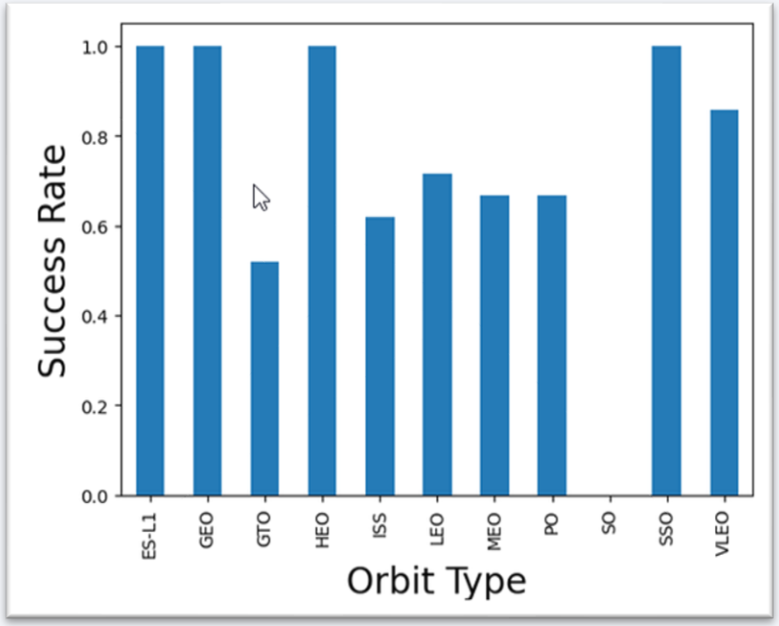


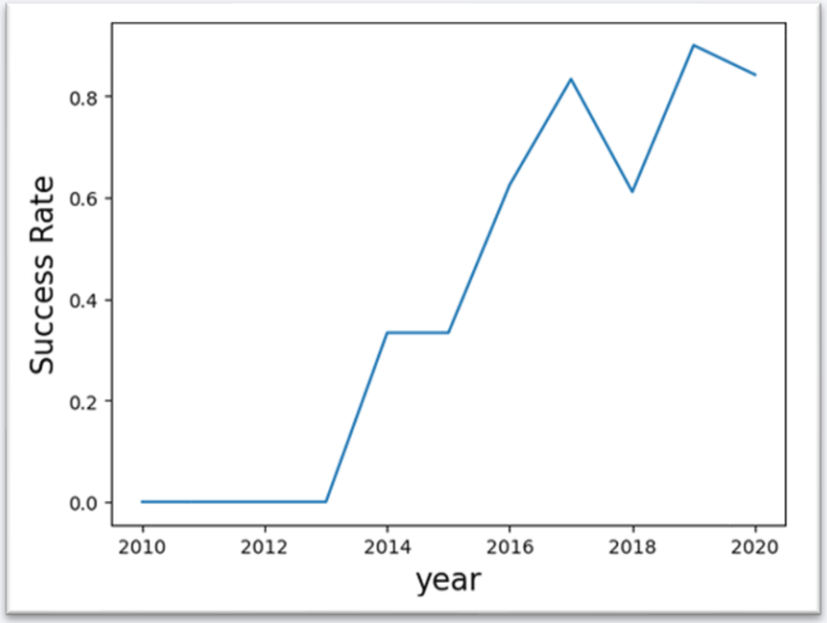
EDA with Data Visualization

- We rendered scatter plots and bar plots to visualize the relationship between pair of features:
  - Payload Mass vs Flight Number
  - Launch Site vs Flight Number
  - Launch Site vs Payload Mass
  - Orbit vs Flight Number
  - Payload vs Orbit
- https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%202B%20-%20EDA%20with%20Visualization%20lab.ipynb.ipynb

```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="FlightNumber",x="LaunchSite",hue='Class',data=df, aspect=5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```





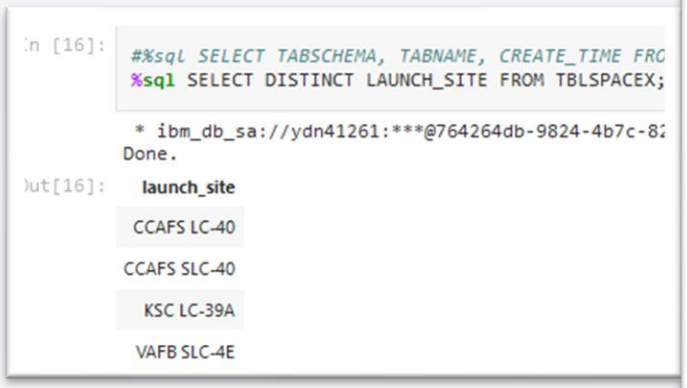


EDA with SQL

We used SQL queries to:

- 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.

https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%202A%20-%20EDA%20with%20SQL.ipynb

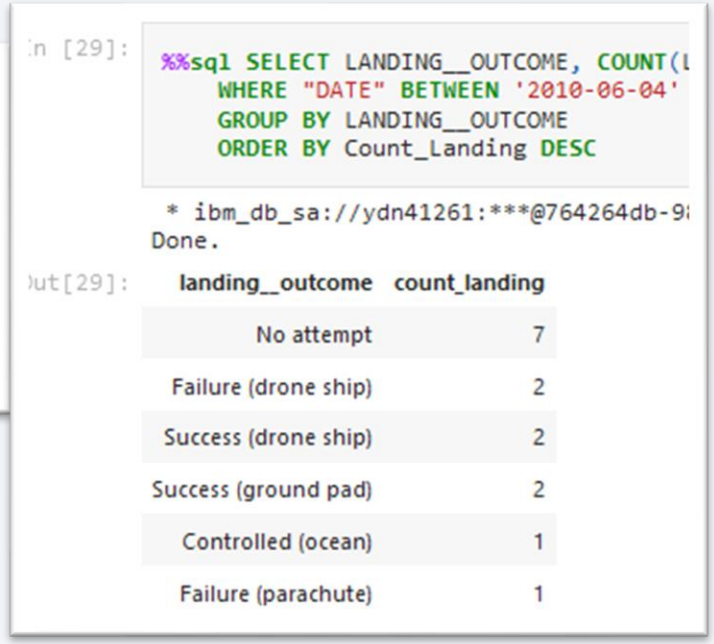


```
In [16]: #%sql SELECT TABSCHEMA, TABNAME, CREATE_TIME FRO
         %sql SELECT DISTINCT LAUNCH_SITE FROM TBLSPACEX;

 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82
Done.
```

Out[16]:

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

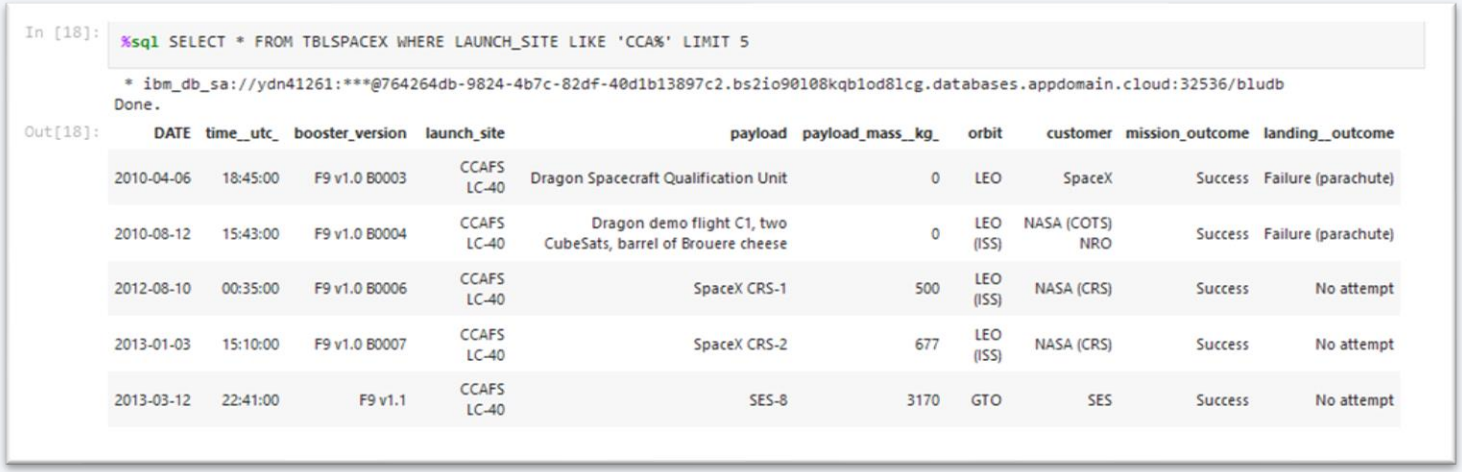


```
In [29]: %%sql SELECT LANDING__OUTCOME, COUNT(L
             WHERE "DATE" BETWEEN '2010-06-04'
             GROUP BY LANDING__OUTCOME
             ORDER BY Count_Landing DESC

 * ibm_db_sa://ydn41261:***@764264db-9
Done.
```

Out[29]:

landing__outcome	count_landing
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1



```
In [18]: %sql SELECT * FROM TBLSPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5

 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[18]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Build an Interactive Map with Folium

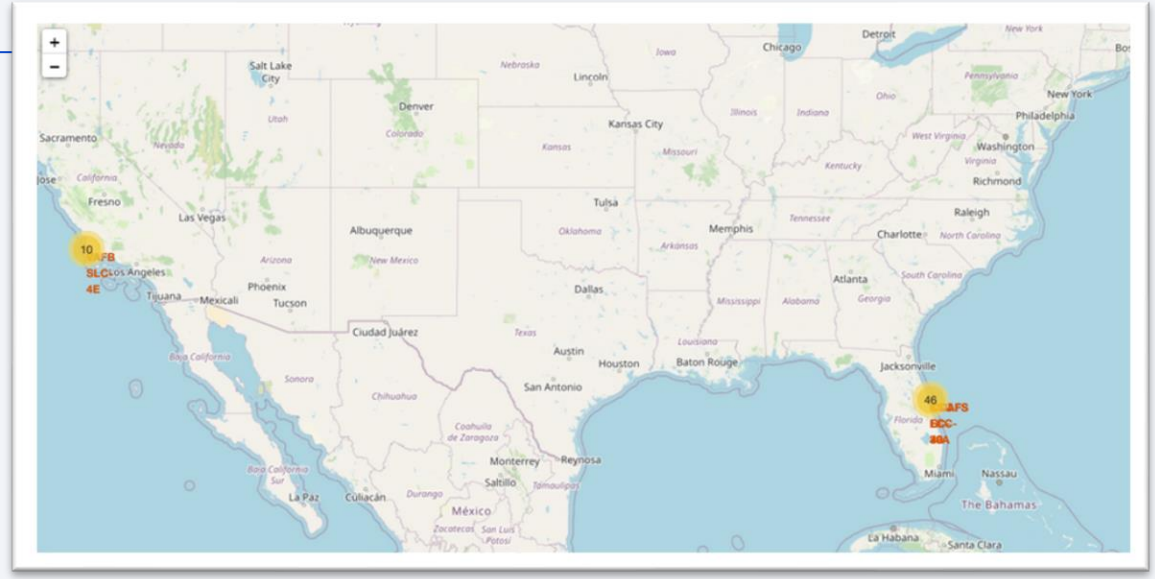
- We used Latitude and Longitude Coordinates at each launch site to Visualizie the Launch Data into an interactive map and added a Circle Marker around each launch site with a label of the name of the launch site.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Then 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. Linesare drawn on the map to measure distance to landmarks
- We calculated the distances between a launch site to its proximities. We answered some question for instance:

  -Are launch sites near railways, highways and coastlines.

  -Do launch sites keep certain distance away from cities.

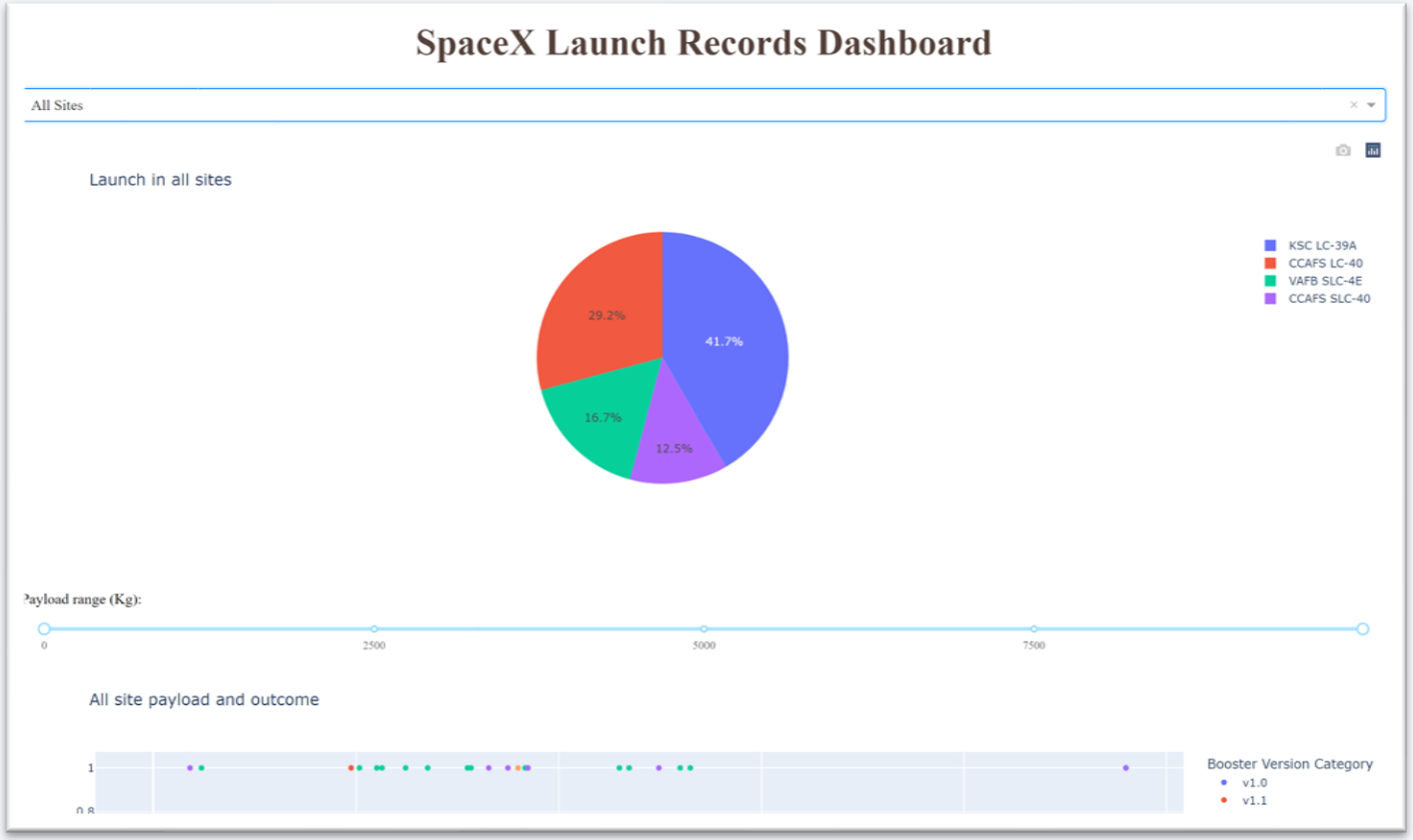
Git URL:

https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%203%20-%20Visual%20Analytics%20with%20Folium%20lab.ipynb



Build a Dashboard with Plotly Dash

- We used Plotly dash to build our Dashboard
- The dashboard allowed us to quickly analyse the relationship between payloads and launch sites, helping us to identify the best launch sites by payload mass
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- https://github.com/Mfaress/IBM_DS_Capstone/blob/main/IBM_Week%203B-Dashboard%20with%20Ploty%20Dash.py



Predictive Analysis (Classification)

Building the Model

- Load the dataset into NumPy and Pandas
- Transform the data and then split into training and test datasets
- Decide which type of ML to use
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Evaluating the Model

- Check the accuracy for each model
- Get tuned hyperparameters for each type of algorithms.
- plot the confusion matrix

Improving the Model

- Use Feature Engineering
- and Algorithm Tuning

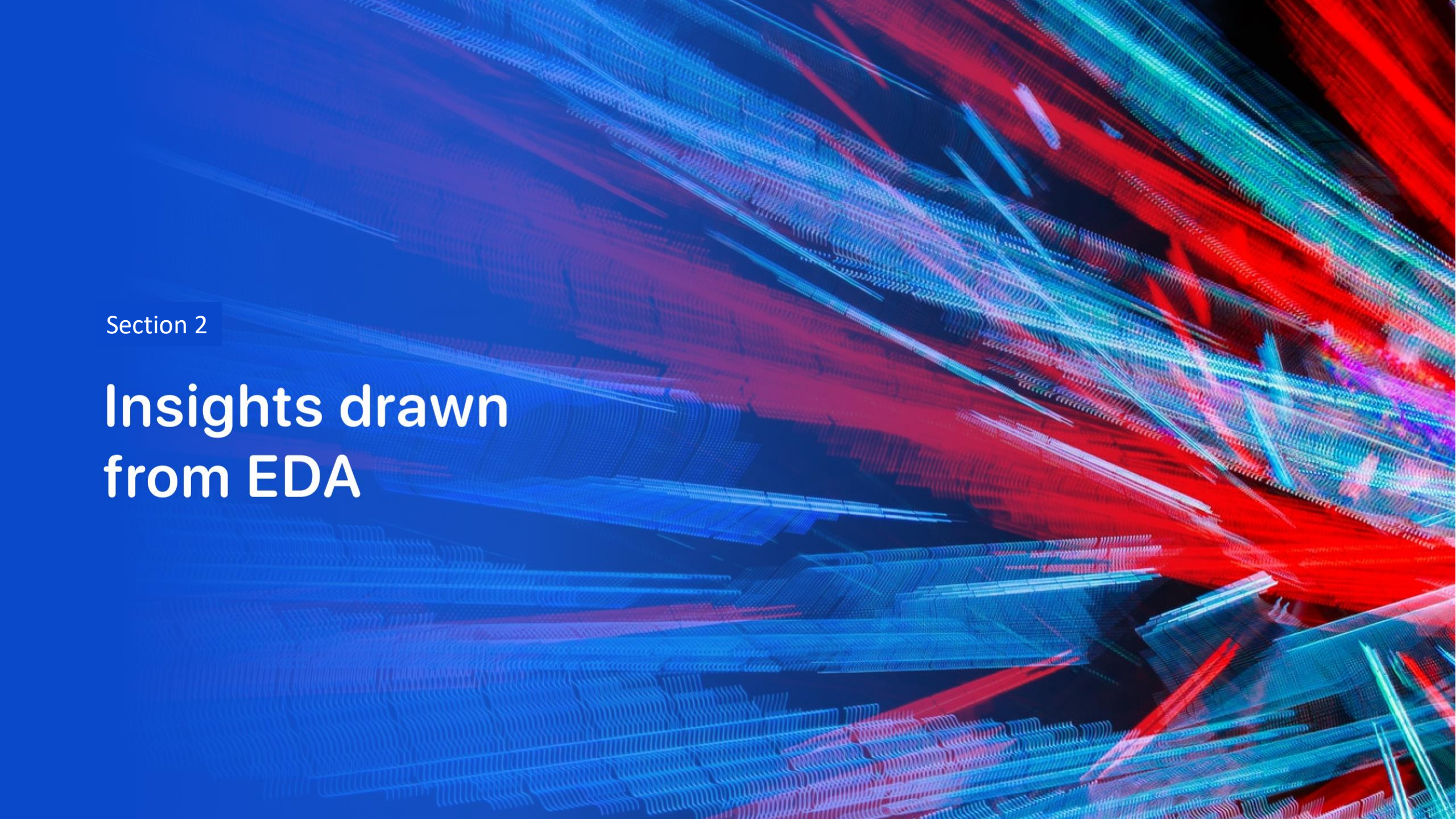
Find the Best Model

- The model with the best accuracy score will be the best performing model.

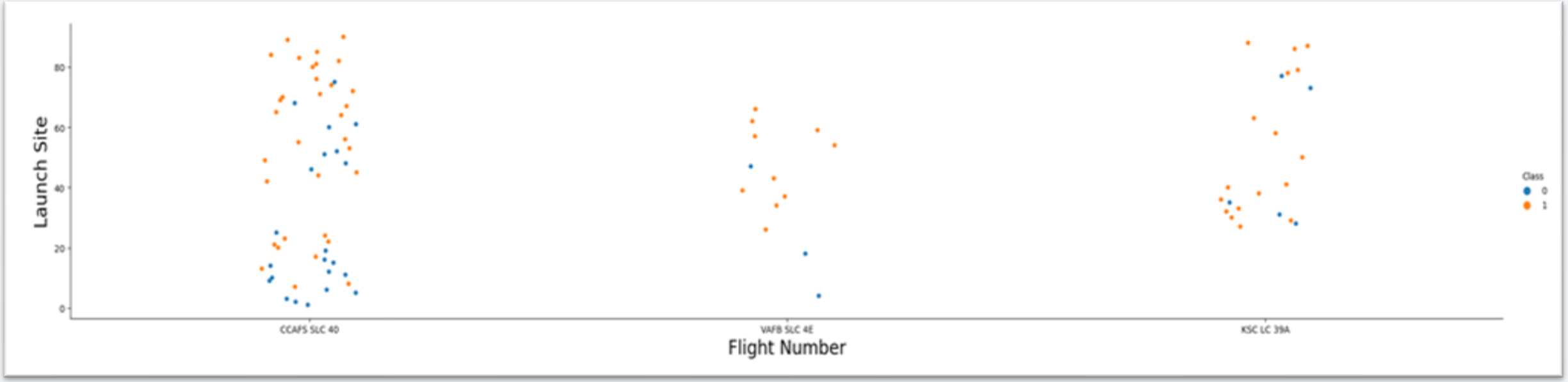
http://localhost:8888/notebooks/Desktop/IBM_DS-Capstone/Python%20code/IBM_Week%204%20-%20Machine%20Learning%20Prediction%20lab.ipynb

Results

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

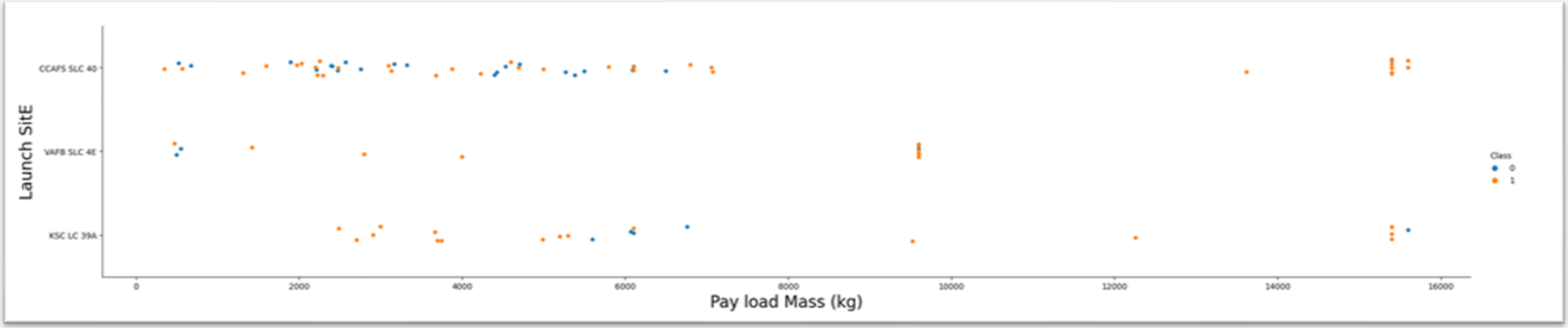

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



we found that the larger the flight amount at a launch site, the greater the success rate at a launch site

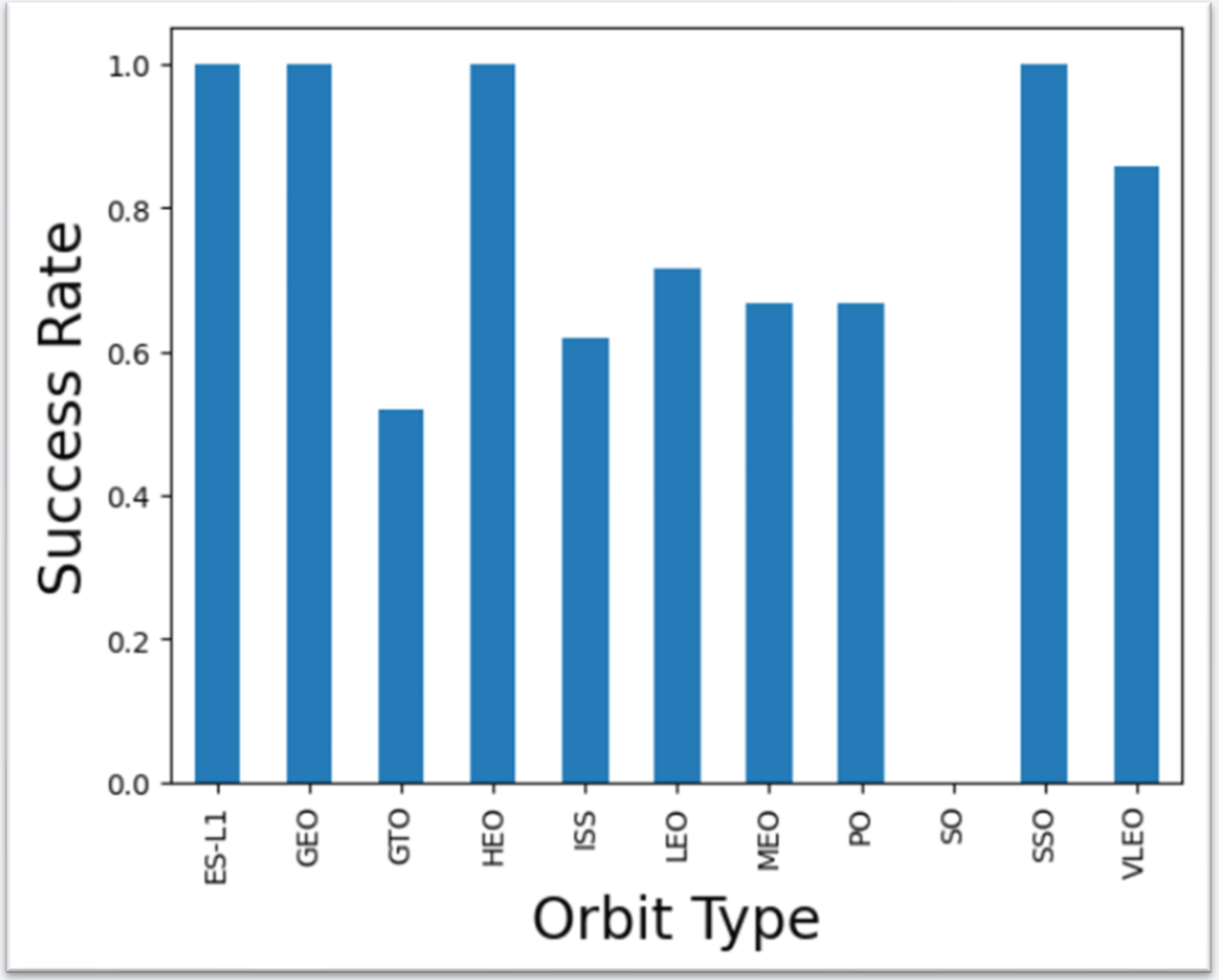
Payload vs. Launch Site



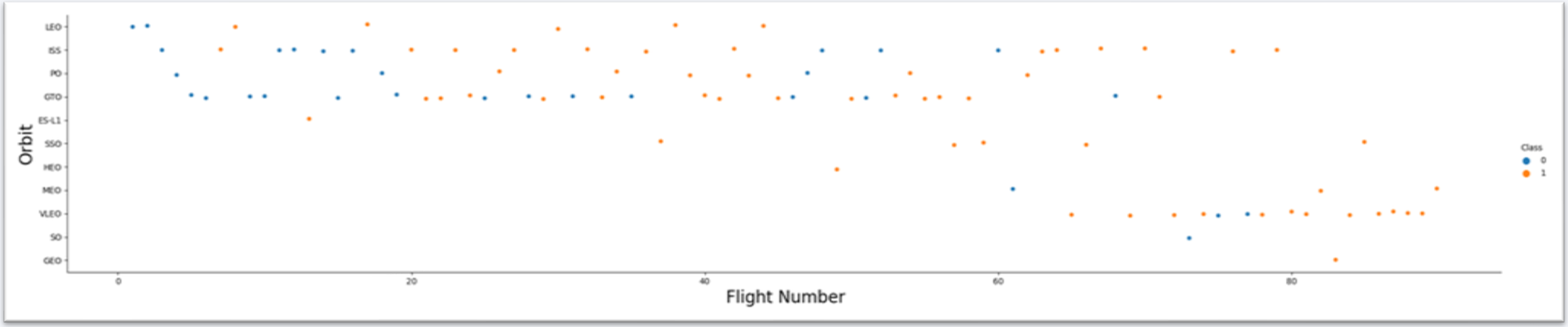
Higher payloads (over 12,000kg) have a better success rate at CCAFS SLC 40 and KSC LC 39A launch sites

Success Rate vs. Orbit Type

From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

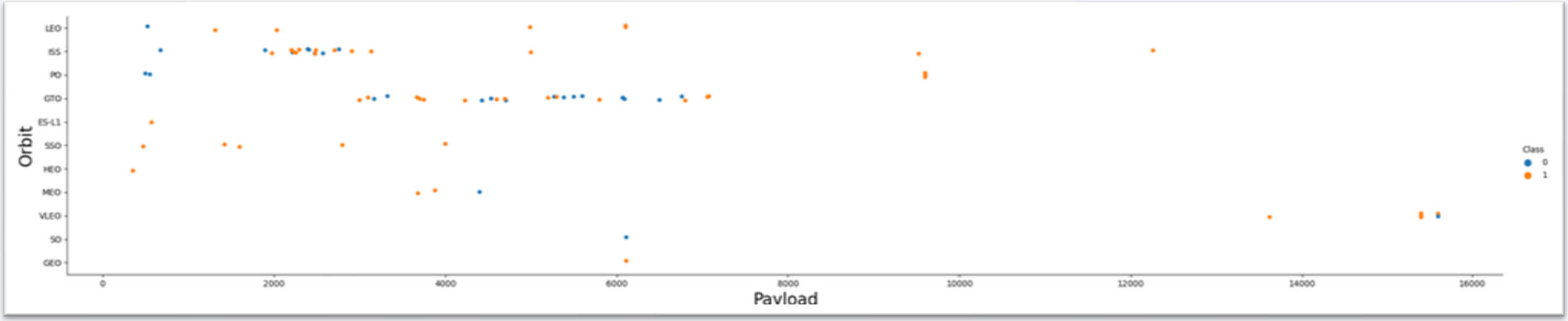


Flight Number vs. Orbit Type



We observe that in the LEO orbit, success is related to the number of flights, on the other hand, there seems to be no relationship between flight number when in GTO orbit.

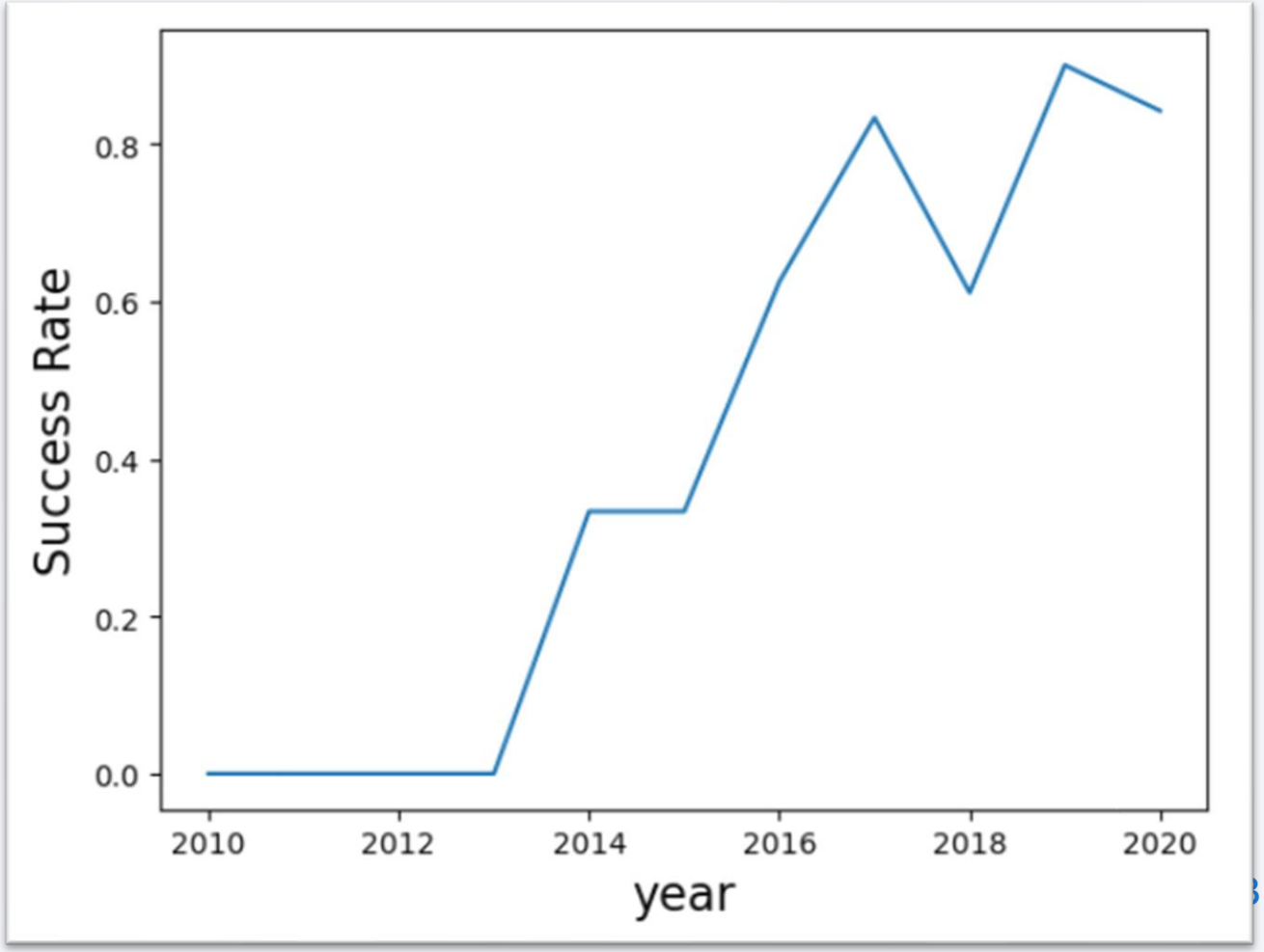
Payload vs. Orbit Type



Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.

Launch Success Yearly Trend

we can observe that success rate since 2013 kept on increasing till 2020



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

In [16]:
```
#%sql SELECT TABSCHEMA, TABNAME, CREATE_TIME FROM
%sql SELECT DISTINCT LAUNCH_SITE FROM TBLSPACEX;
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82
Done.
```

Out[16]:

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

Launch Site Names Begin with 'CCA'

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

In [18]:
```
%sql SELECT * FROM TBLSPACEX WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

Out[18]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

We used the query above to display 5 records where launch sites begin with `CCA`

Total Payload Mass

Task 3

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

In [20]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as Total_Payload FROM TBLSPACEX WHERE CUSTOMER='NASA (CRS)'
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.
Done.
```

Out[20]:

total_payload
22007

We found the total payload carried by boosters from NASA that equal 22007

Average Payload Mass by F9 v1.1

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

```
In [21]: %sql SELECT AVG(PAYLOAD_MASS__KG_) as Average_Payload FROM TBLSPACEX WHERE BOOSTER_VERSION='F9 v1.1'
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.databases.appdo
Done.
```

Out[21]:

average_payload
3676

We found the Average Payload Mass by F9 v1.1 that equal 3676

First Successful Ground Landing Date

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

Hint:Use min function

```
In [22]: %sql SELECT min(DATE) FROM TBLSPACEX WHERE LANDING__OUTCOME='Success (ground pad)'
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8
Done.
Out[22]:
1
2017-01-05
```

We found the First Successful Ground Landing Date is 2017-01-05

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 thar

In [23]:
```
%%sql SELECT BOOSTER_VERSION FROM TBLSPACEX WHERE
    PAYLOAD_MASS__KG_ between 4000 and 6000 AND LANDING__OUTCOME='Success (drone ship)'
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1od8lcg.da
Done.
```

Out[23]:

booster_version
F9 FT B1022

We 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

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
In [24]: %~sql SELECT COUNT(*) as total_number FROM TBLSPACEX
             WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'

 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90108kqb1
Done.
```

Out[24]:

total_number
45

We used wildcard like '%' to filter for **WHERE** Mission_Outcome was a success or a failure.

Boosters Carried Maximum Payload

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

In [25]:
```
!%sql SELECT BOOSTER_VERSION FROM TBLSPACEX
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM TBLSPACEX)
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.datal
Done.
```

Out[25]:

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

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

2015 Launch Records

List the failed landing_outcomes in drone ship, their booster versions, and launch site nan

In [26]:
```
%%sql SELECT LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
        FROM TBLSPACEX
        WHERE Landing__Outcome = 'Failure (drone ship)' AND YEAR(DATE) = 2015;
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb
Done.
```

Out[26]:

landing__outcome	booster_version	launch_site
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40

We 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

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

In [29]:
```
%%sql SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) as Count_Landing FROM TBLSPACEX
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20'
    GROUP BY LANDING__OUTCOME
    ORDER BY Count_Landing DESC
```

```
 * ibm_db_sa://ydn41261:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.c
Done.
```

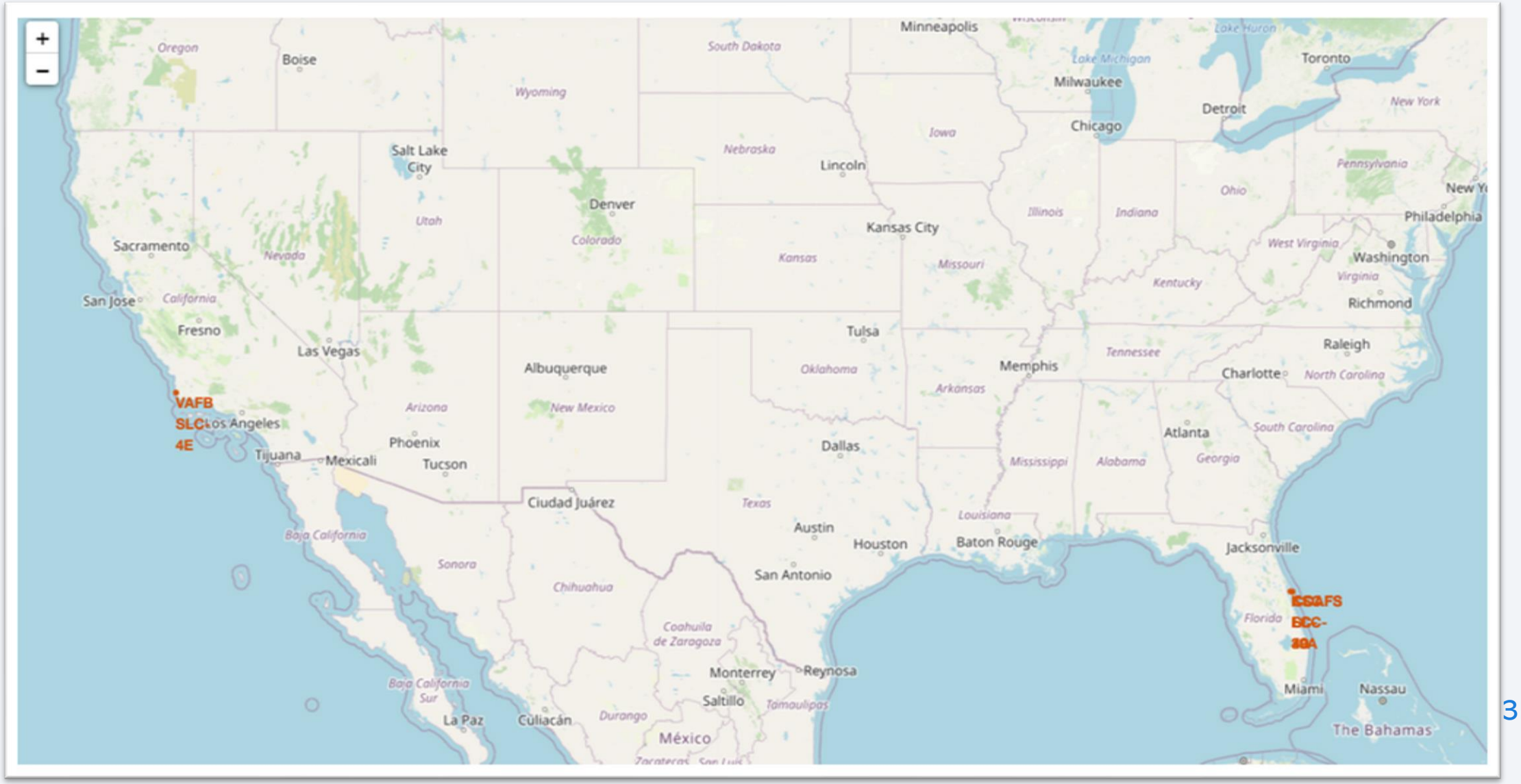
Out[29]:

landing__outcome	count_landing
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

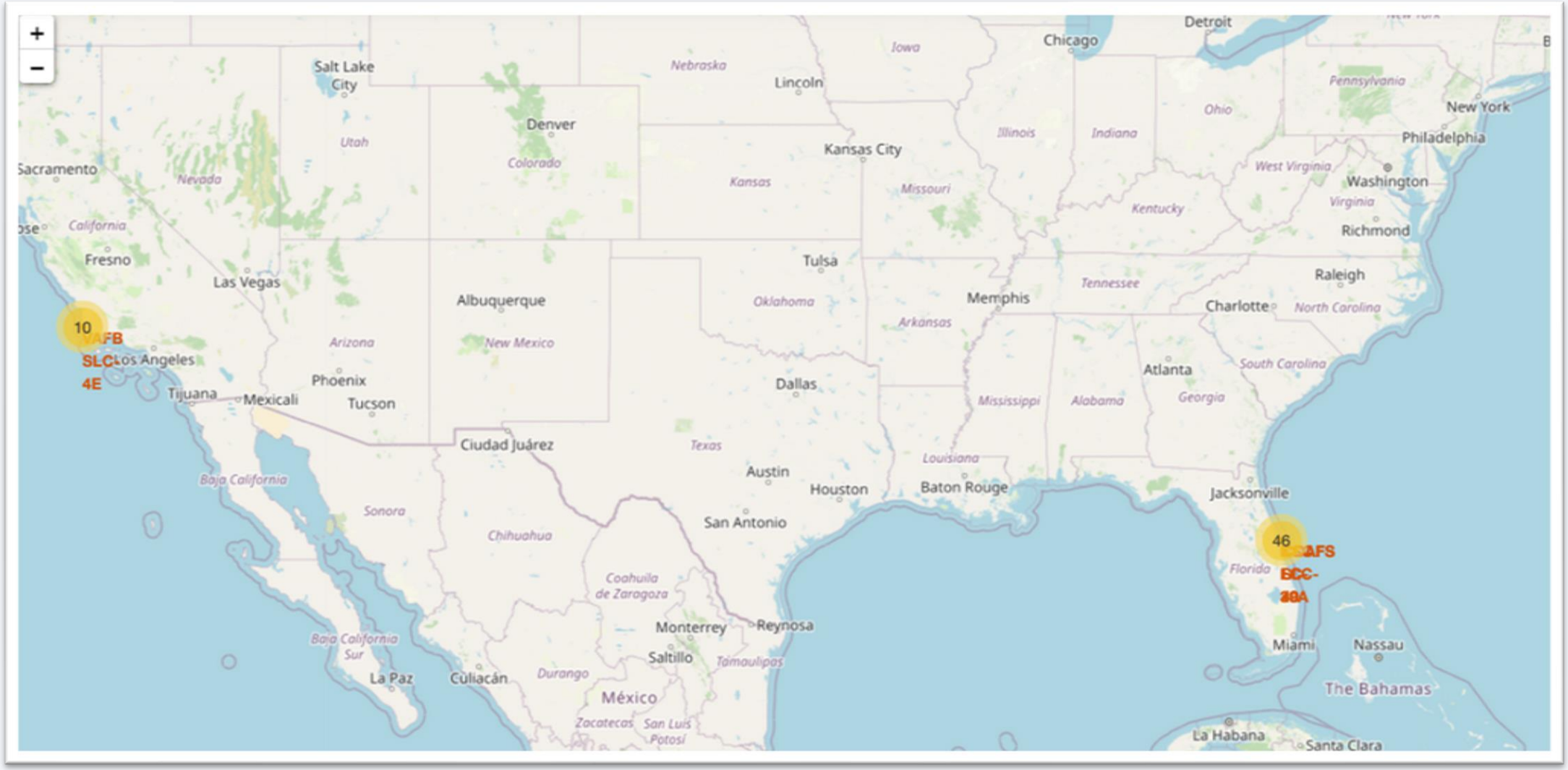
We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.


Section 3
Launch Sites
Proximities Analysis

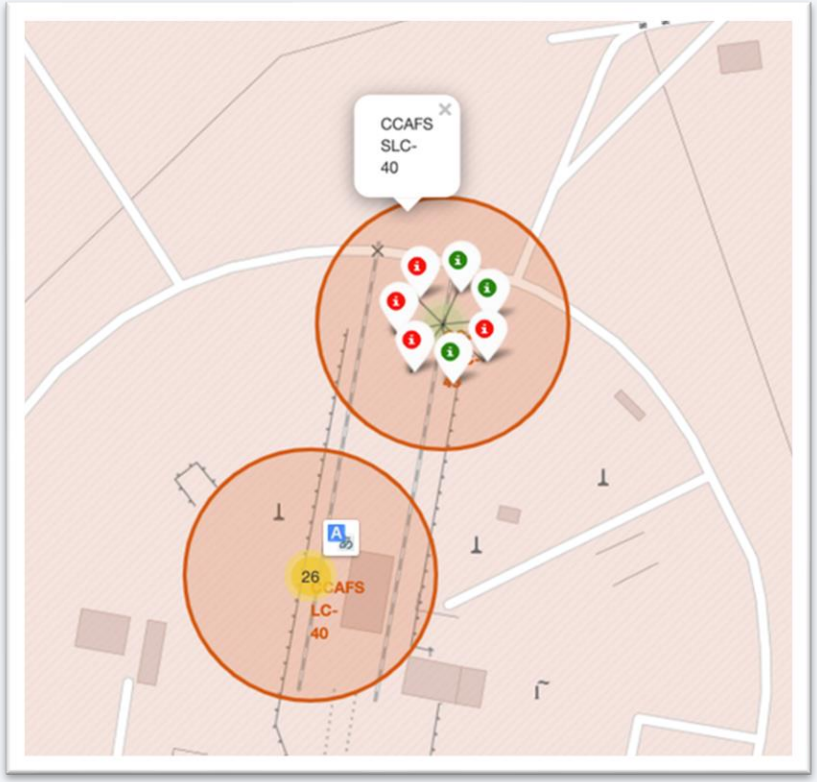
All launch sites

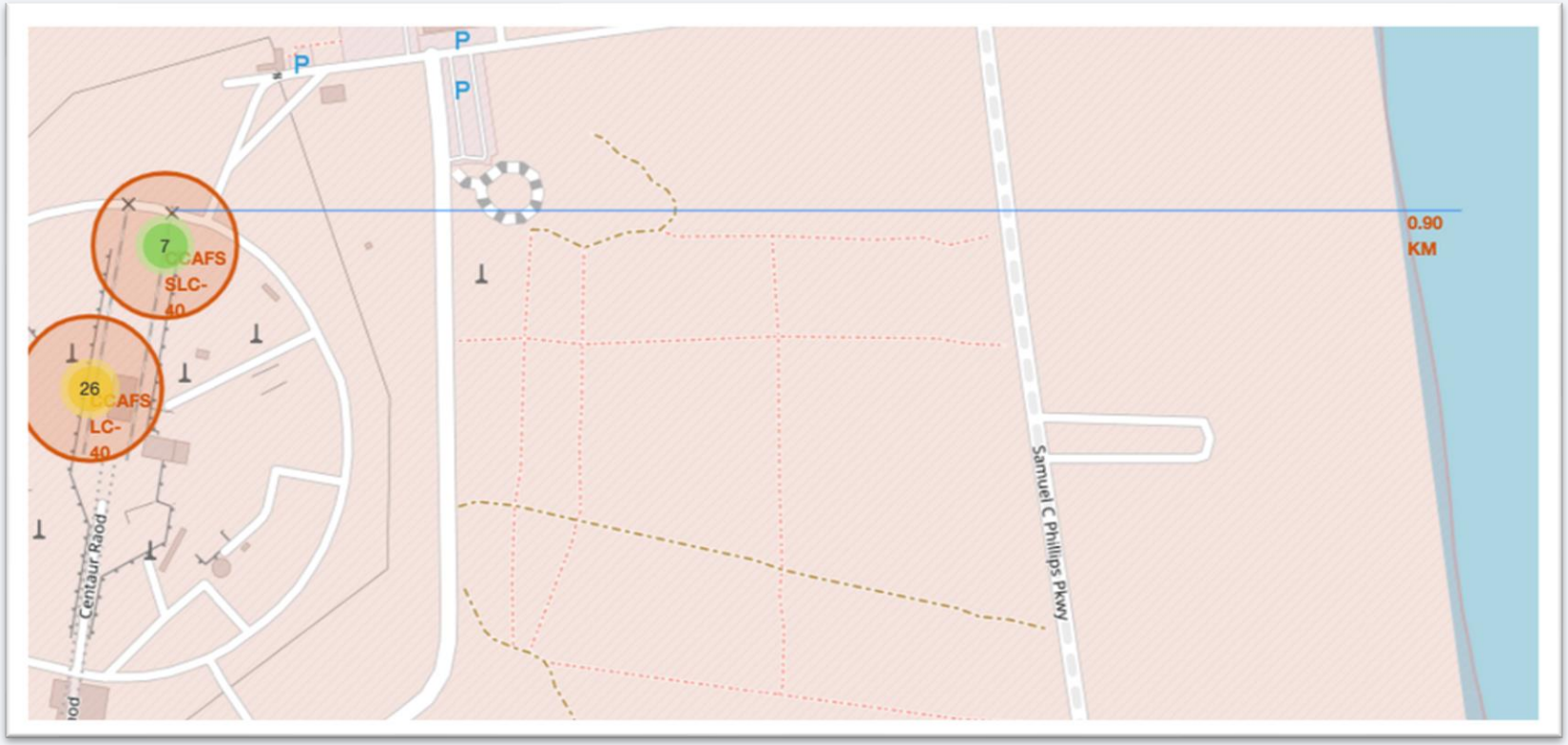


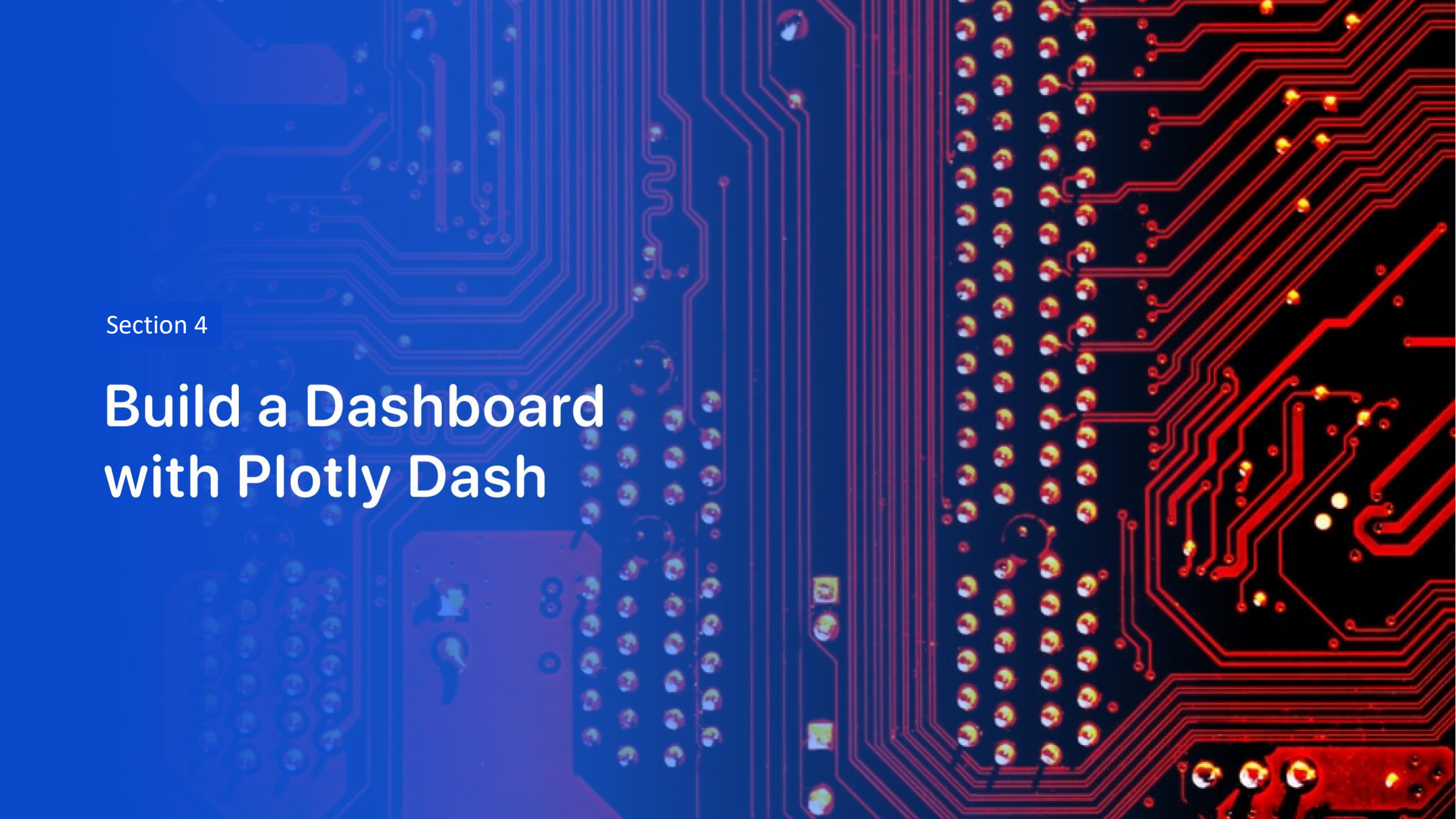
Launch outcome by launch site



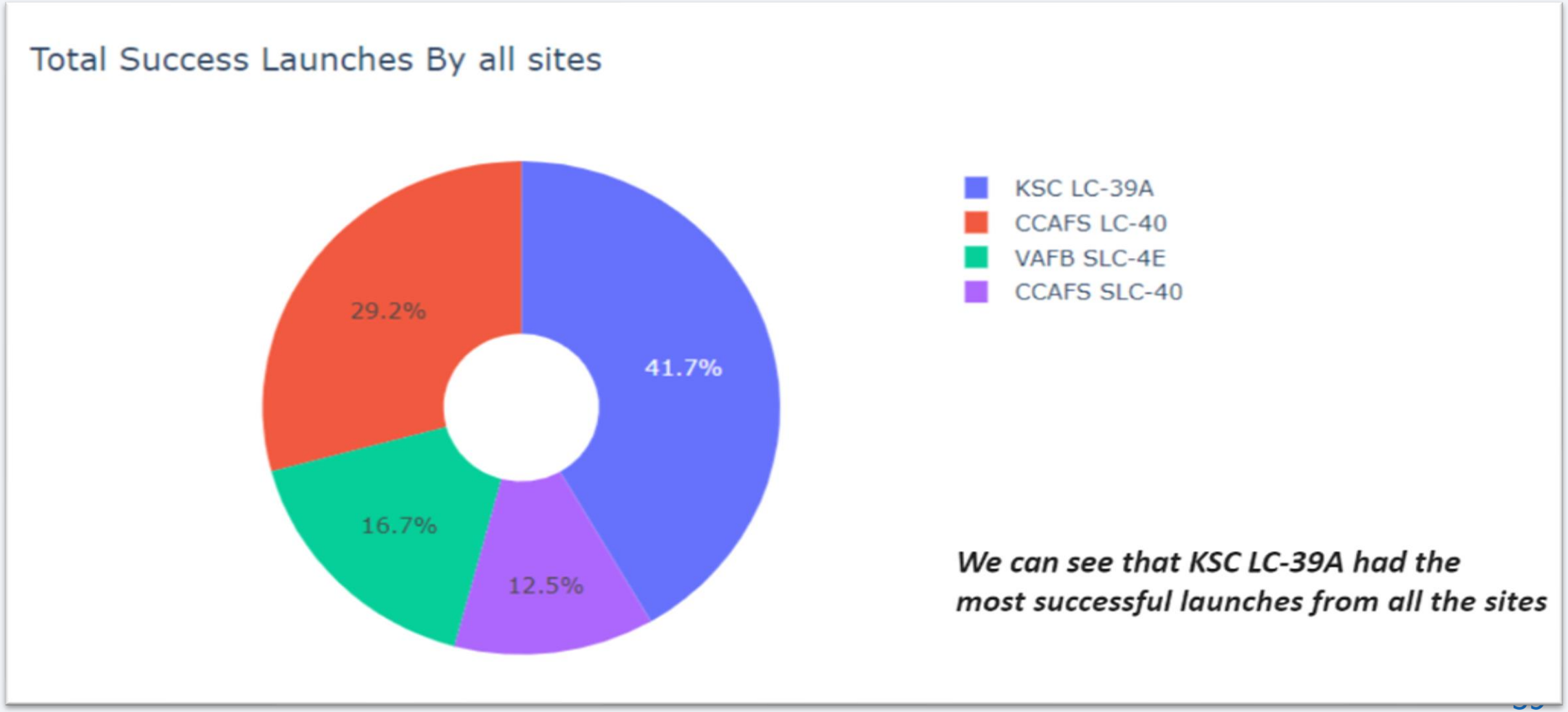
Launch Site distance to landmarks



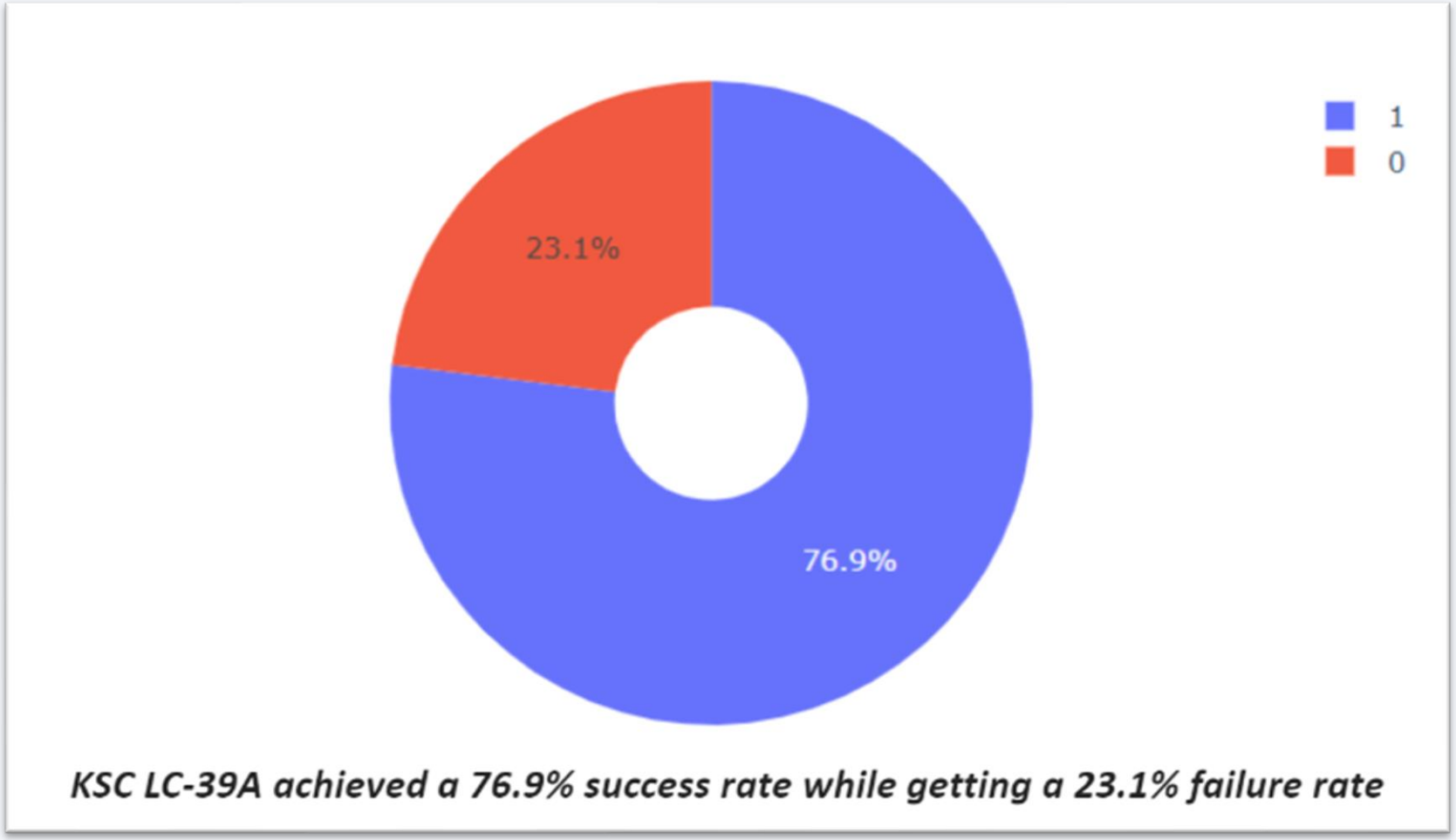



Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site

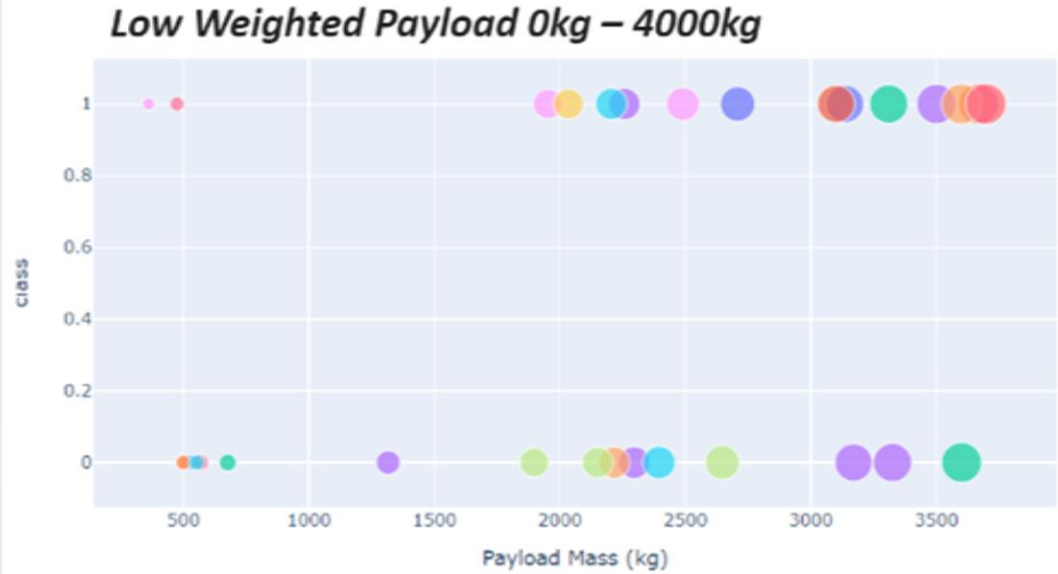


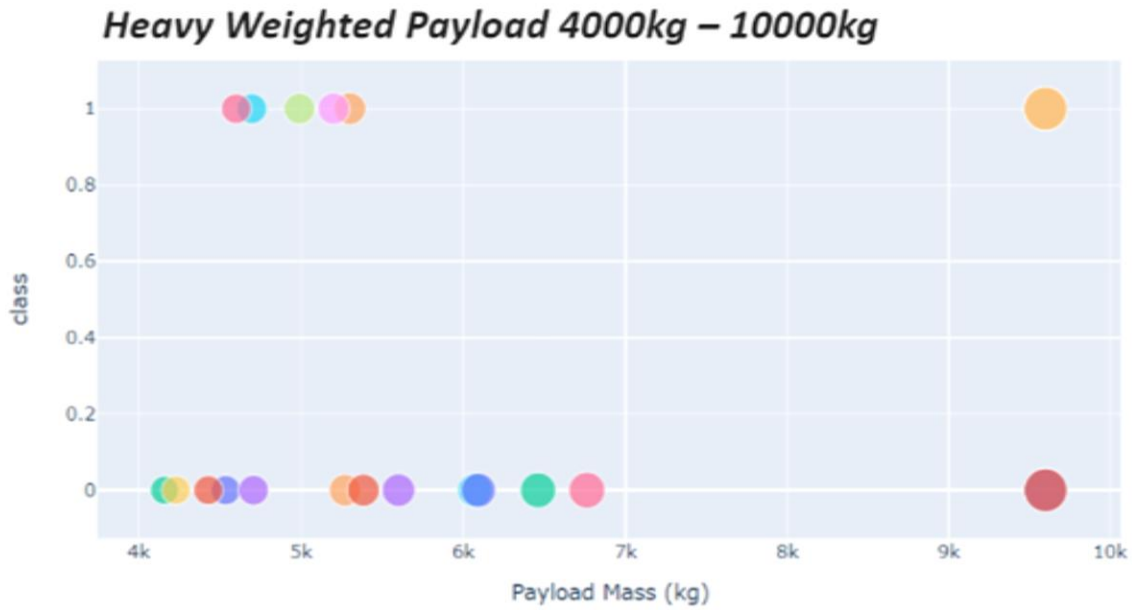
Launch Success Ratio for KSC LC-39A



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

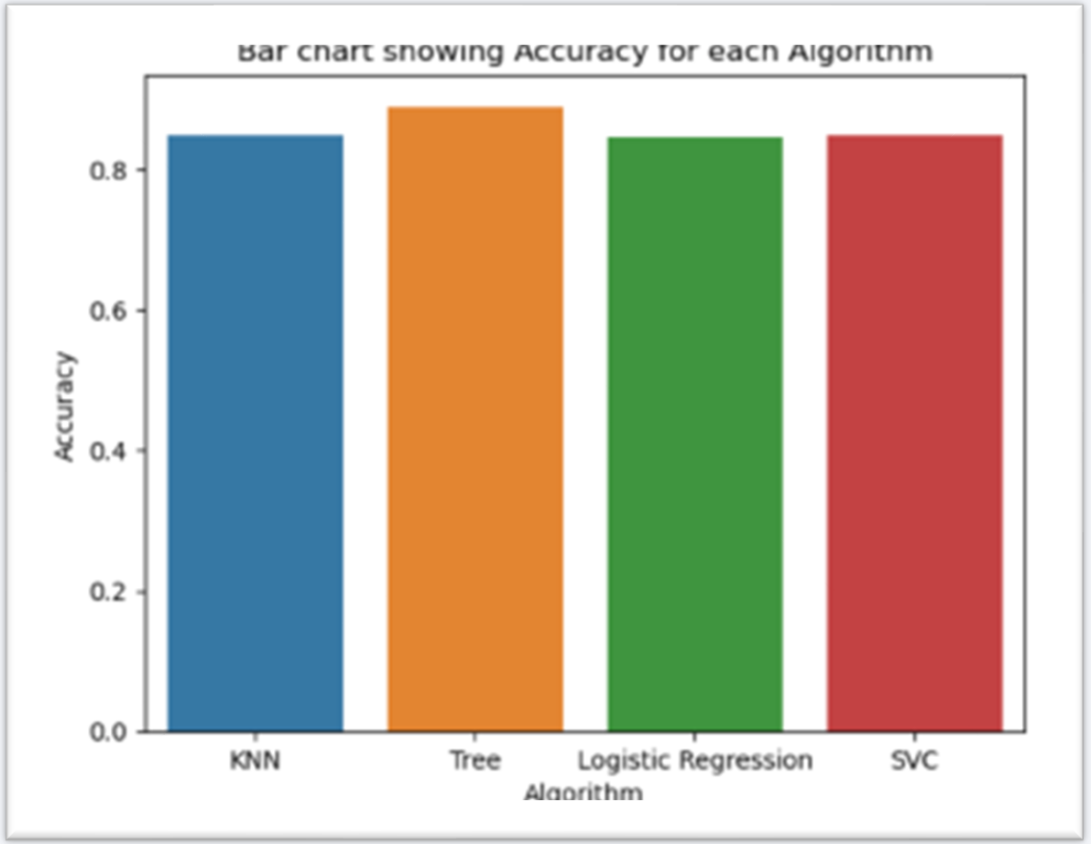
Predictive Analysis (Classification)

Classification Accuracy

```
bestalgorithm = max(algorithms, key=algorithms.get)
```

5]:

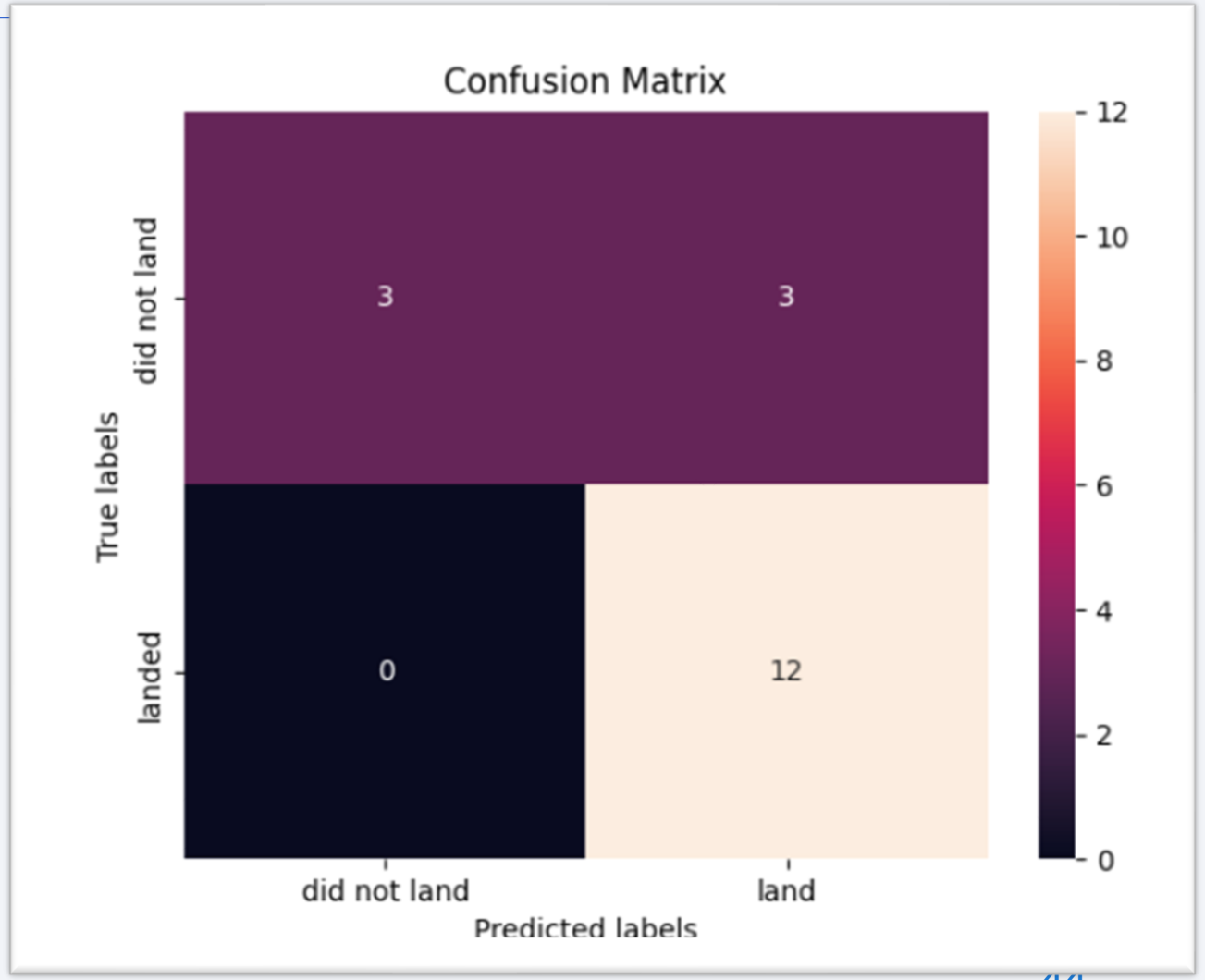
	Algorithm	Accuracy
0	KNN	0.848214
1	Tree	0.889286
2	Logistic Regression	0.846429
3	SVC	0.848214



Confusion Matrix

The Decision Tree confusion matrix shows the largest number of true positive and true negative compared to other models

```
In [40]: yhat = knn_cv.predict(X_test)
         plot_confusion_matrix(Y_test,yhat)
```



Conclusions

- The larger the flight amount 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 Tree Classifier Algorithm is the best for Machine Learning for this dataset

Appendix

- https://github.com/Mfaress/IBM_DS_Capstone
- https://www.coursera.org/learn/applied-data-science-capstone
- https://www.coursera.org/professional-certificates/ibm-data-science
- https://cloud.ibm.com/

Thank you!