IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

By Moshe, 2023

Outline

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

Executive Summary - Methodologies

- Data Collection and Preparation
- Exploratory Data Analysis (EDA) and Visualization
- Data Wrangling and Processing
- Machine Learning Prediction
- Conclusion

Executive Summary - findings

1. **Correlation Between Factors and Landing Success:** Identified relationships between various factors (payload mass, orbit type, etc.) and the successful landing of the Falcon 9 first stage.
2. **Predictive Capability:** Developed a machine learning model capable of predicting first stage landing outcomes with a certain level of accuracy using historical data.
3. **Cost Estimation Impact:** Explored the implications of successful landings on launch costs, showcasing the cost-saving potential derived from SpaceX's first stage reuse.
4. **Spatial Analysis of Launch Sites:** Analyzed the impact of launch site locations on the success rates of launches, potentially uncovering spatial patterns influencing mission outcomes.

Introduction - background and context

- The project aimed to predict the successful landing of the Falcon 9 first stage in SpaceX launches. By leveraging data analysis, machine learning, and visualizations, the goal was to determine if the first stage would land, crucial for estimating launch costs. Insights derived from the analysis could be valuable for competitive bidding scenarios in the rocket launch industry.

Introduction – few problems and answers!

1. What factors influence the successful landing of the Falcon 9 first stage?

1. Through analysis, the project might have identified correlations between certain variables (e.g., payload mass, orbit type, launch site location) and the landing success rate.

2. Can we predict the first stage landing outcome based on historical data?

1. The machine learning model developed in the project could potentially predict whether the first stage would land successfully in future launches by learning from historical data patterns.

3. How does SpaceX's ability to reuse the first stage impact launch costs?

1. By predicting landing outcomes, the project might have provided insights into estimating launch costs, as the ability to reuse the first stage significantly affects cost savings.

4. What role does the launch site location play in the success rate of SpaceX launches?

1. Visualizations and analyses of launch site locations might have revealed spatial patterns or correlations influencing the success rates of launches.

Section 1

Methodology

Methodology

- (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

Data Collection

- The data was collected through a combination of web scraping, API integration, and possibly manual data entry from reliable sources such as SpaceX's records or official databases.

Data Collection – SpaceX API

- **Data Collection Phrases:**
- **Authenticate SpaceX API:** Authenticate access to SpaceX API using necessary credentials.
- **Retrieve Launch Data:** Send API requests to retrieve launch data including launch outcomes, locations, dates, and details.
- **Handle Paginated Responses:** Manage paginated responses from the API to gather comprehensive launch history.
- **Data Structuring:** Organize retrieved data into a structured format suitable for analysis

- [SpaceX API notebook]

Task 1: Request and parse the SpaceX launch data using the GET request

To make the requested JSON results more consistent, we will use the following static response object for this project:

In [10]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS03
```

We should see that the request was successfull with the 200 status response code

In [11]:
```
response.status_code
```

Out[11]: 200

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

In [12]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [13]:
```
# Get the head of the dataframe
data.head()
```

Out[13]:

static_fire_date_utc	static_fire_date_unix	net	window	rocket	success

Data Collection - Scraping

- \- Identify target webpage
- \- Inspect HTML structure
- \- Use scraping libraries (e.g., BeautifulSoup)
- \- Extract relevant data
- \- Perform data cleaning

[Data scraping notebook]

TASK 1: Request the Falcon9 Launch Wiki page from its URL

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

In [6]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

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

In [13]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html5lib')
#soup
```

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

In [14]:
```
# Use soup.title attribute
soup.title.string
```

Out[14]: 'List of Falcon 9 and Falcon Heavy launches - Wikipedia'

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 `BeautifulSoup`, please check the external reference link towards the end of this lab

In [15]:
```
# 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')
```

Starting from the third table is our target table contains the actual launch records.

In [16]:
```
# Let's print the third table and check its content
table1 = html_tables[2]
print(first_launch_table)
```

Data Wrangling

- The data underwent processing involving steps like cleaning for missing values, formatting for consistency, feature engineering to create relevant predictors, and possibly normalization or scaling for model compatibility.

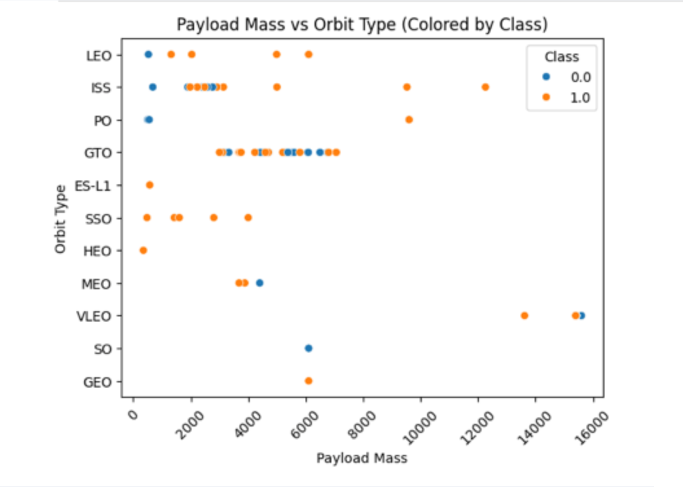
- data wrangling notebook

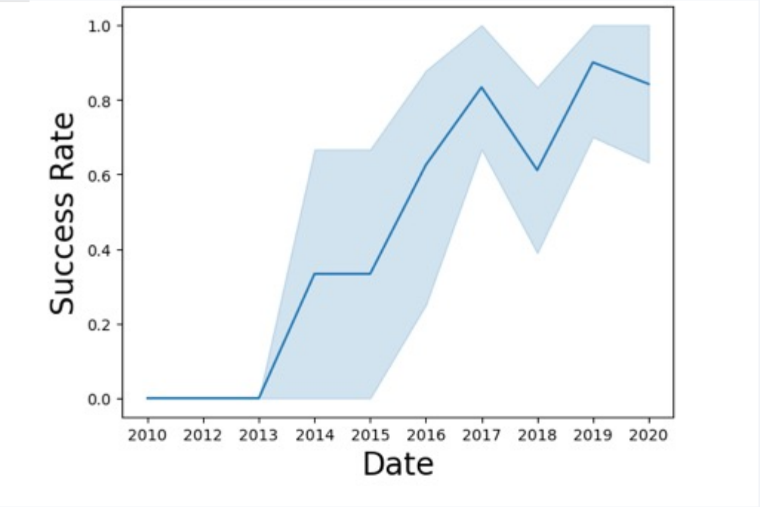
EDA with Data Visualization

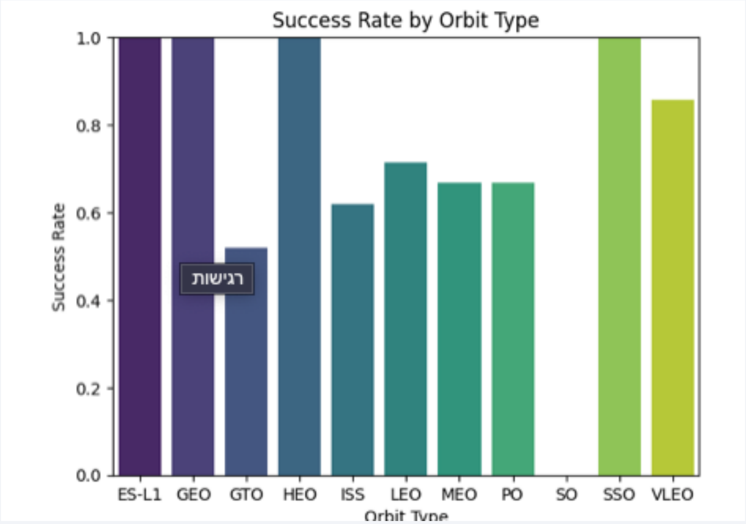
Types of Plots used:

- Table
- Scatter plot
- Linear regression line
- Bar plot
- Line chart

- [EDA with data visualization notebook]





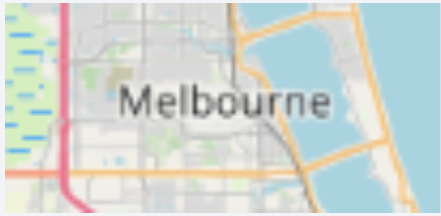


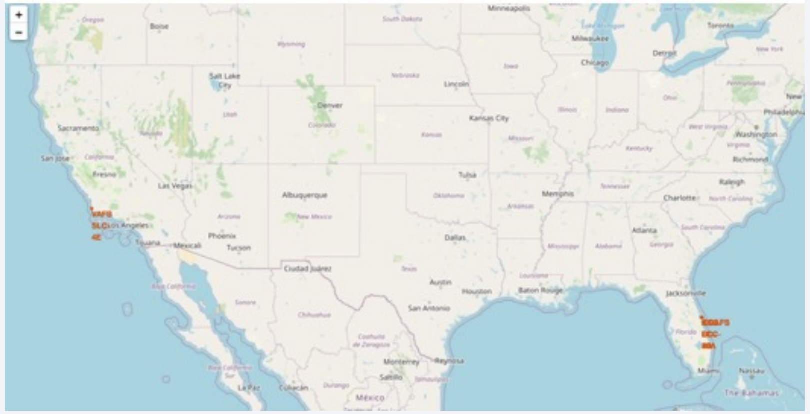
EDA with SQL

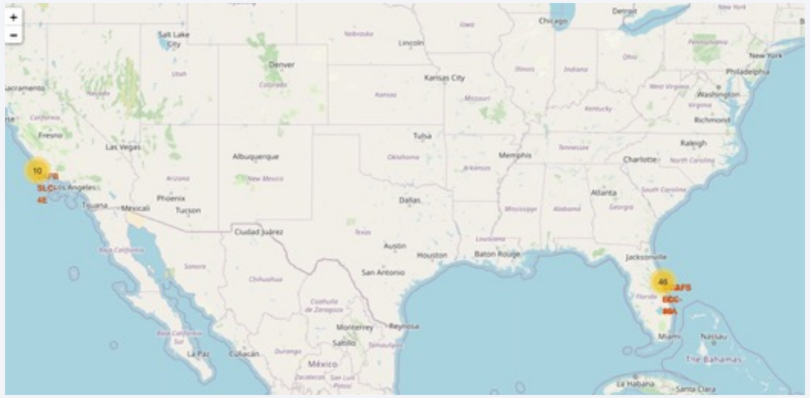
- **%sql** SELECT Distinct LAUNCH_SITE FROM SPACEXTBL
- **%sql** SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
- **%sql** SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
- **%sql** SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'
- **%sql** SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'
- **%sql** SELECT Booster_Version, PAYLOAD_MASS__KG_, Landing_Outcome FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_
- **%sql** SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'
- **%sql** SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_)
- **%sql** SELECT * FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015
- **%sql** SELECT "DATE", COUNT(Landing_Outcome) FROM SPACEXTBL\ WHERE "DATE" BETWEEN '2010-06-04' **and** '2017-03-20' AND Landing_Outcome LIKE '%Success%' \ GROUP BY "DATE" \ ORDER BY DATE
- [EDA with SQL notebook]()

Build an Interactive Map with Folium

- Used objects such as markers, circles, lines, etc.

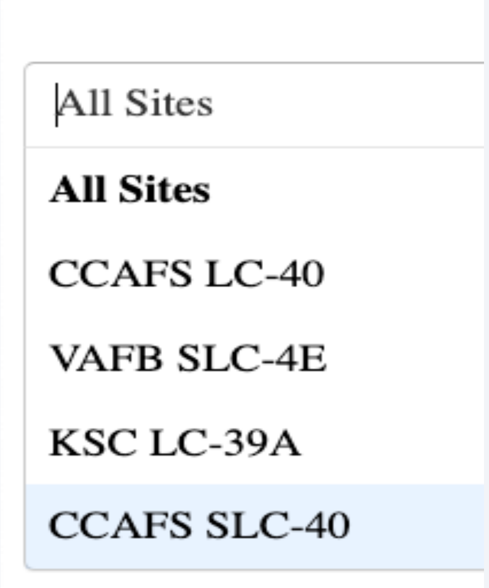


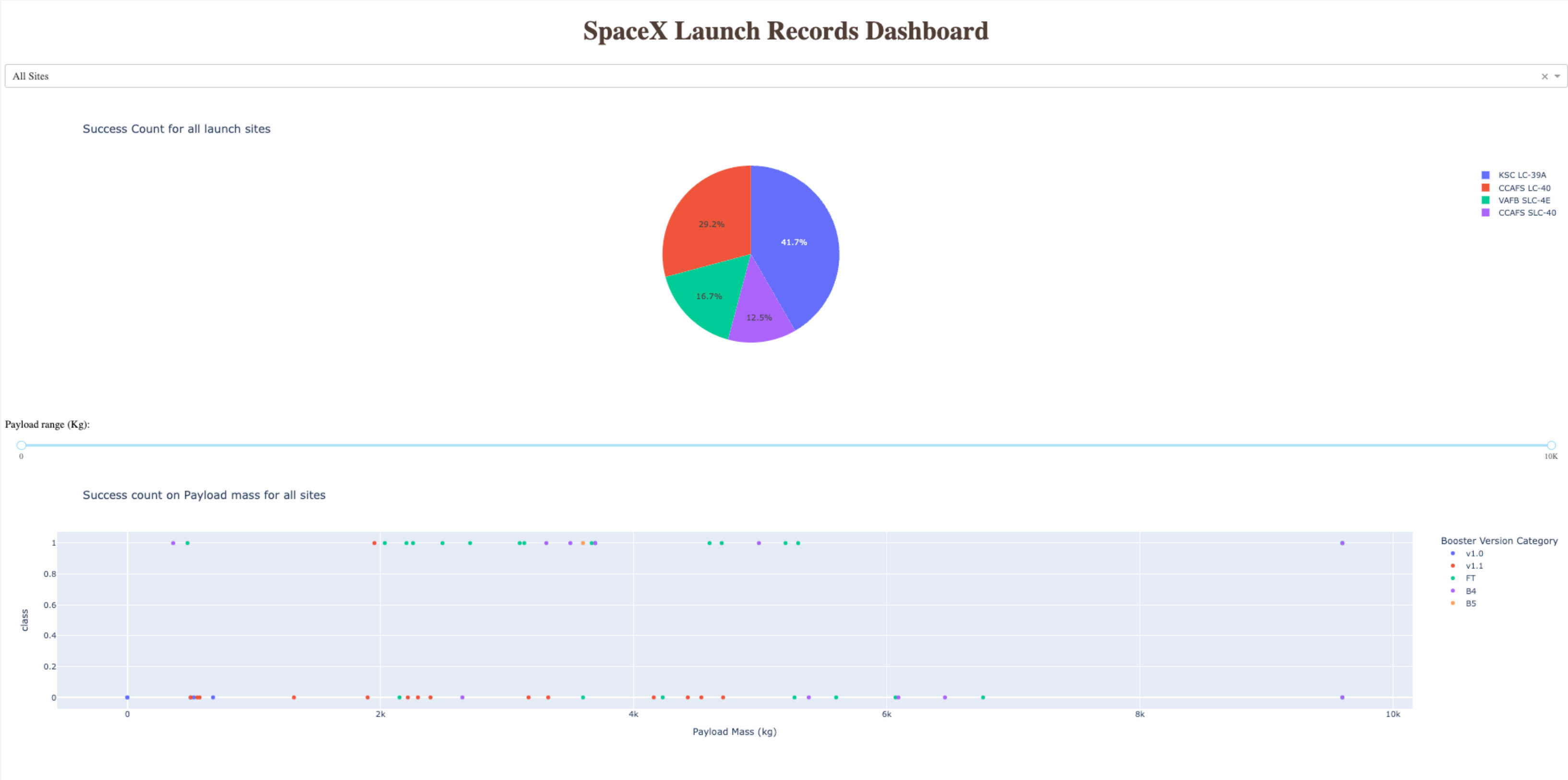




- [interactive map with Folium]()

Build a Dashboard with Plotly Dash





- [Plotly Dash app]()

Predictive Analysis (Classification)

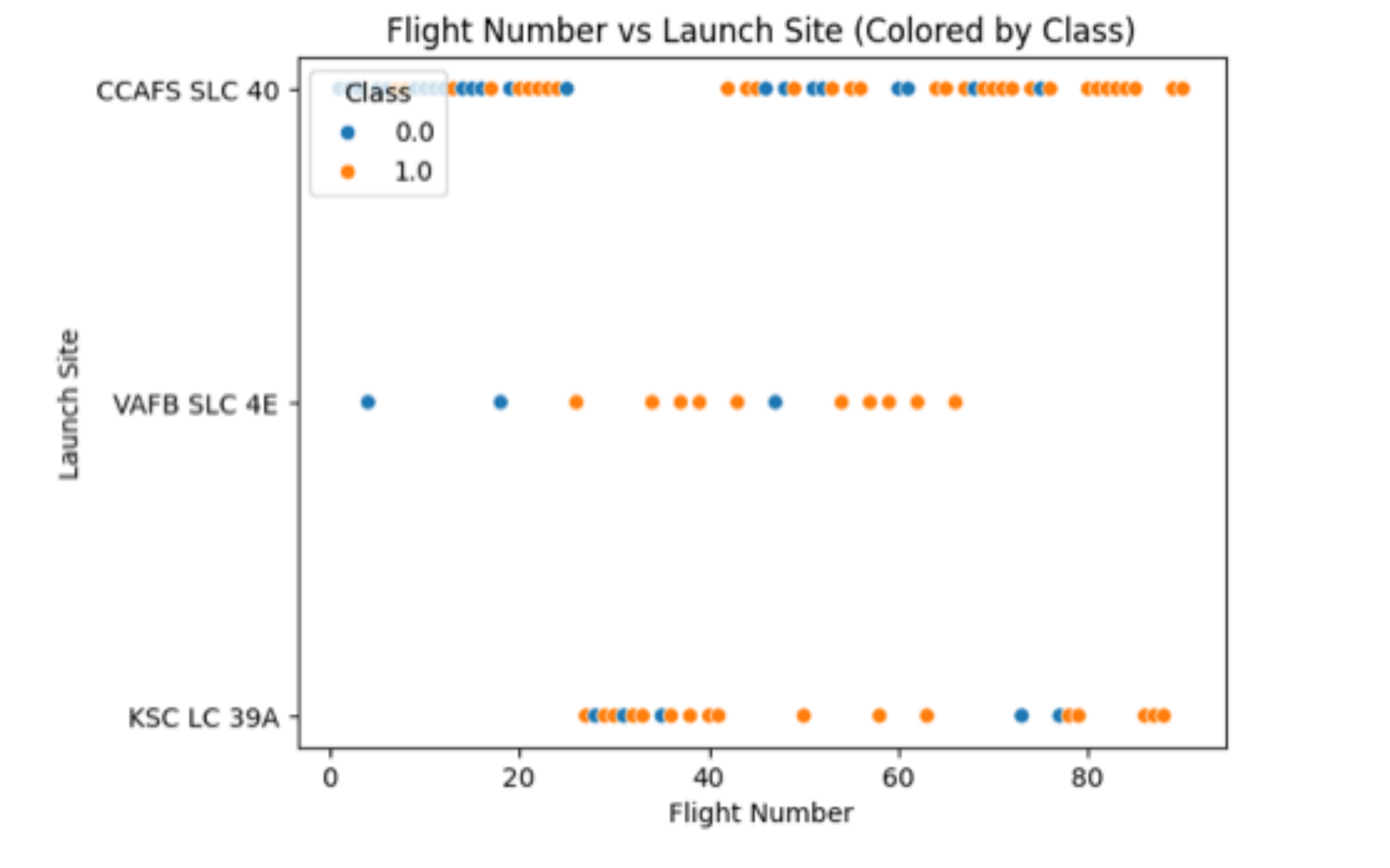
1. **Model Selection:** Choose diverse classification algorithms.
2. **Initial Evaluation:** Assess models using baseline metrics.
3. **Hyperparameter Tuning:** Optimize parameters through grid/random search.
4. **Cross-Validation:** Validate models with k-fold cross-validation.
5. **Model Comparison:** Compare performance metrics to select the best-performing model.
6. **Iterative Improvement:** Iterate on feature engineering or hyperparameter tuning for enhancement.
7. **Final Evaluation:** Evaluate the improved model on a test set for final performance assessment.

- predictive analysis notebook

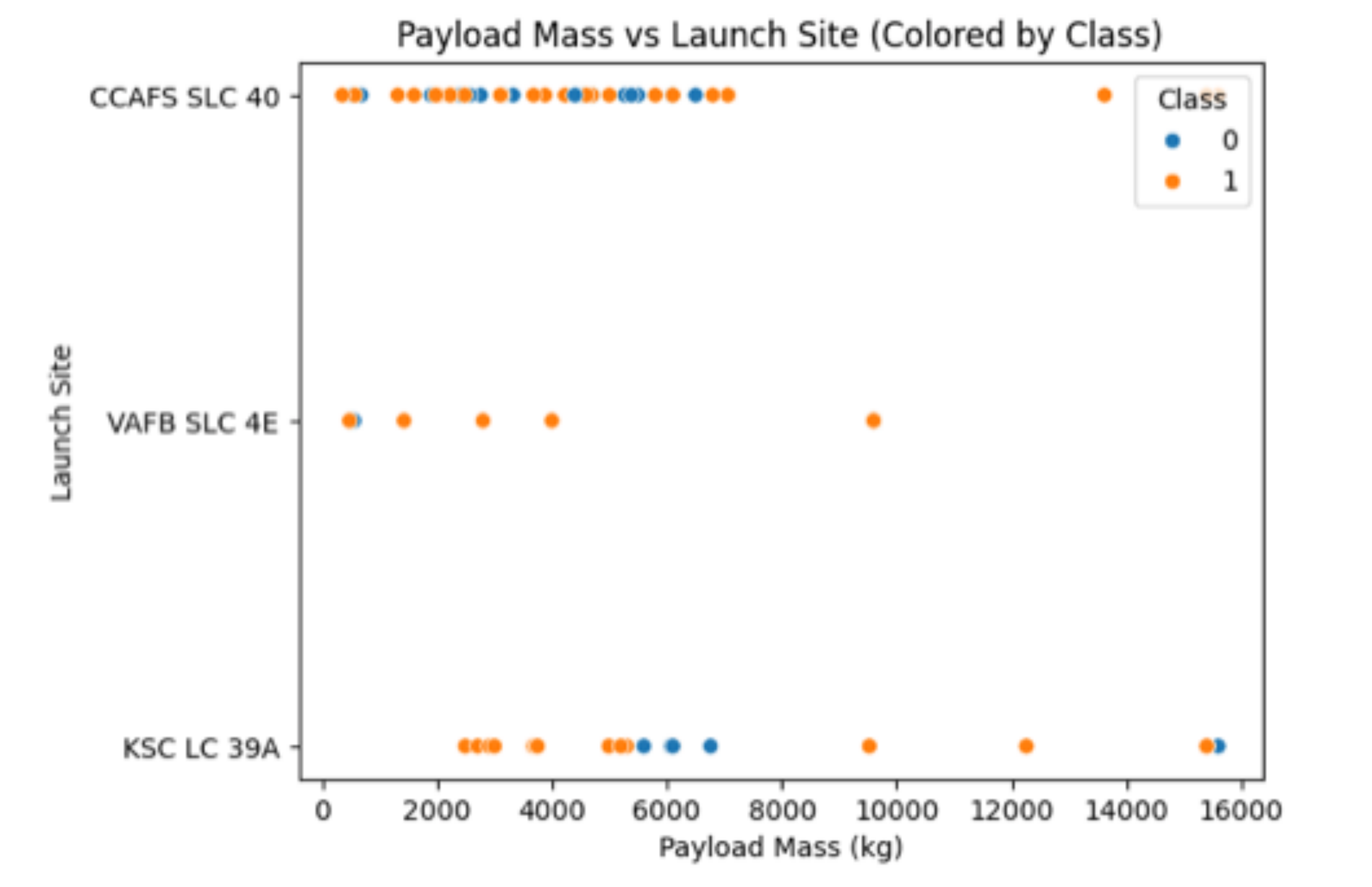
Section 2

Insights drawn from EDA

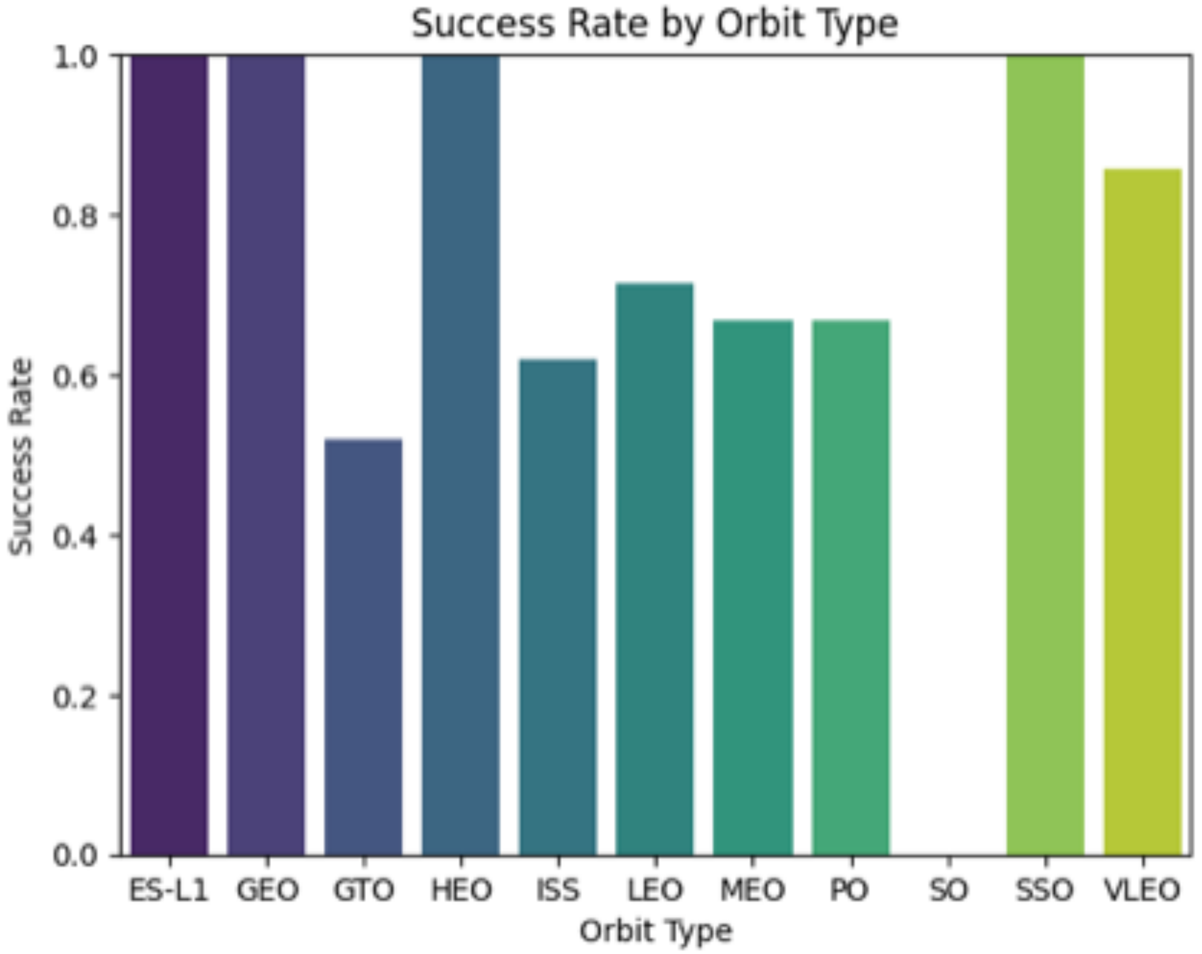
Flight Number vs. Launch Site



Payload vs. Launch Site

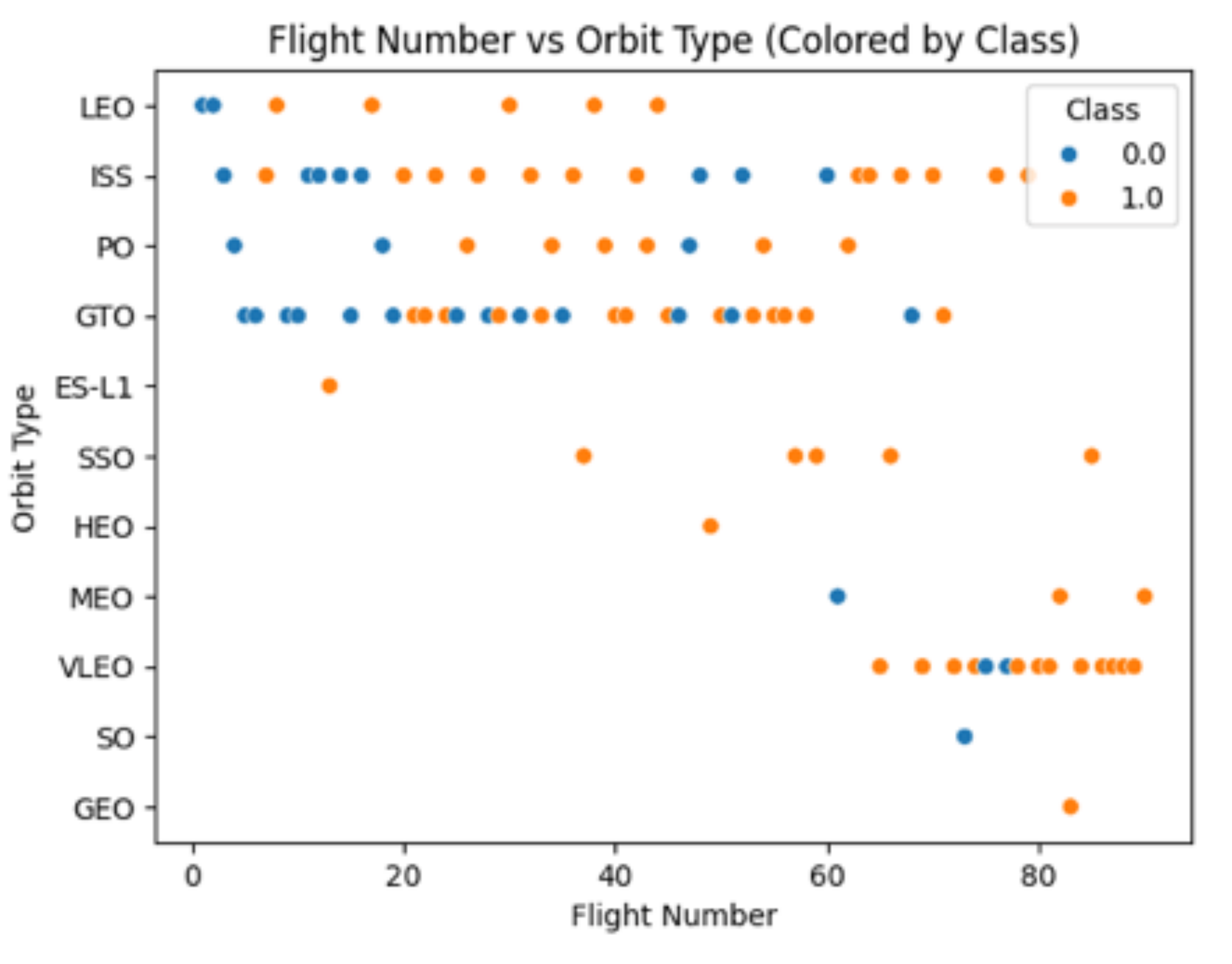


Success Rate vs. Orbit Type

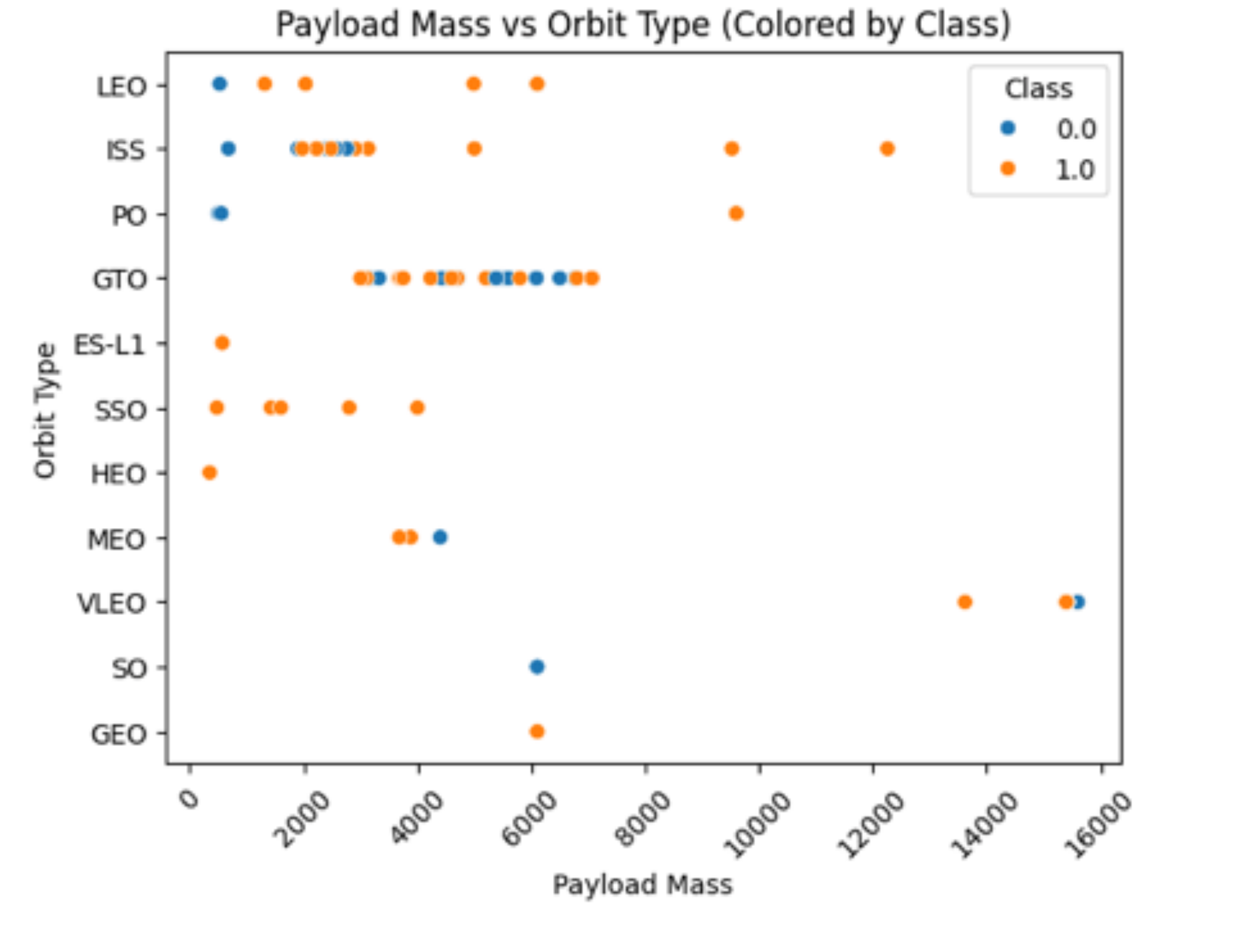


Analyze the ploted bar chart try to find which orbits have high sucess rate.

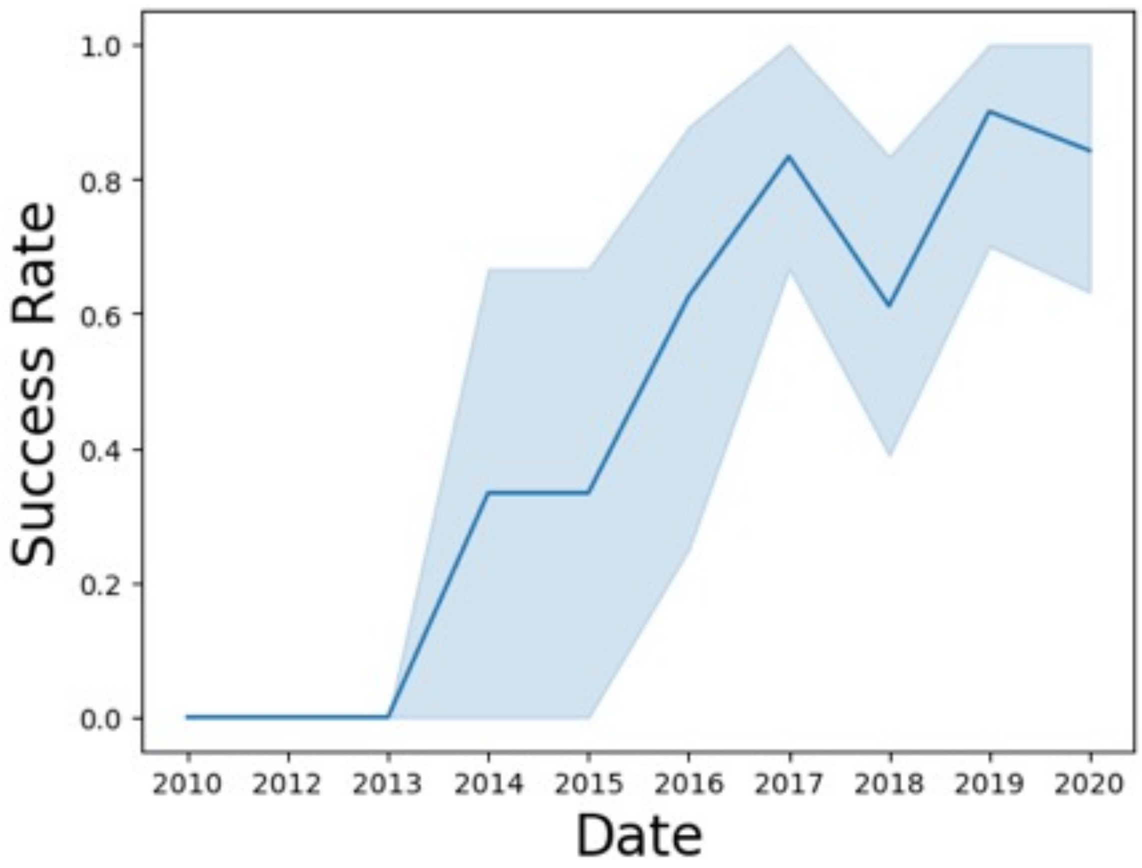
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

```
In [ ]: %sql SELECT Distinct LAUNCH_SITE FROM SPACEXTBL

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

Out[ ]:

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

Launch Site Names Begin with 'CCA'

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

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

Out[ ]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success

Total Payload Mass

```
In [ ]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'

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

Out[ ]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

```
In [ ]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE BOOSTER_VERSION='F9 v1.1'

 * sqlite:///my_data1.db
Done.

Out[ ]: AVG(PAYLOAD_MASS__KG_)
                        2928.4
```

First Successful Ground Landing Date

```
In [ ]: %sql SELECT min(DATE) FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)'

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

Out[ ]:

min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [ ]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_, Landing_Outcome FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_

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

Out[ ]:

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 B5 B1046.3	4000	Success
F9 B5 B1051.2	4200	Success
F9 B5B1060.1	4311	Success
F9 B5B1062.1	4311	Success
F9 B5 B1048.3	4850	Success
F9 B5 B1047.2	5300	Success
F9 B5 B1058.2	5500	Success
F9 B5 B1046.2	5800	Success

Total Number of Successful and Failure Mission Outcomes

```
In [ ]: %sql SELECT COUNT(*) FROM SPACEXTBL WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME LIKE '%Failure%'

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

Out[ ]:

COUNT(*)
101

Boosters Carried Maximum Payload

In [ ]:

```
%sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTB
```

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

Out[ ]:

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records

```
In [ ]: %sql SELECT  * FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%'

 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT * FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Failure (drone ship)' AND "DATE" LIKE '%2015%']
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

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

In [ ]:

```
%sql SELECT "DATE", COUNT(Landing_Outcome) FROM SPACEXTBL\
    WHERE "DATE" BETWEEN '2010-06-04' and '2017-03-20' AND Landing_Outcome LIKE '%Success%' \
    GROUP BY "DATE" \
    ORDER BY DATE
```

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

Out[ ]:

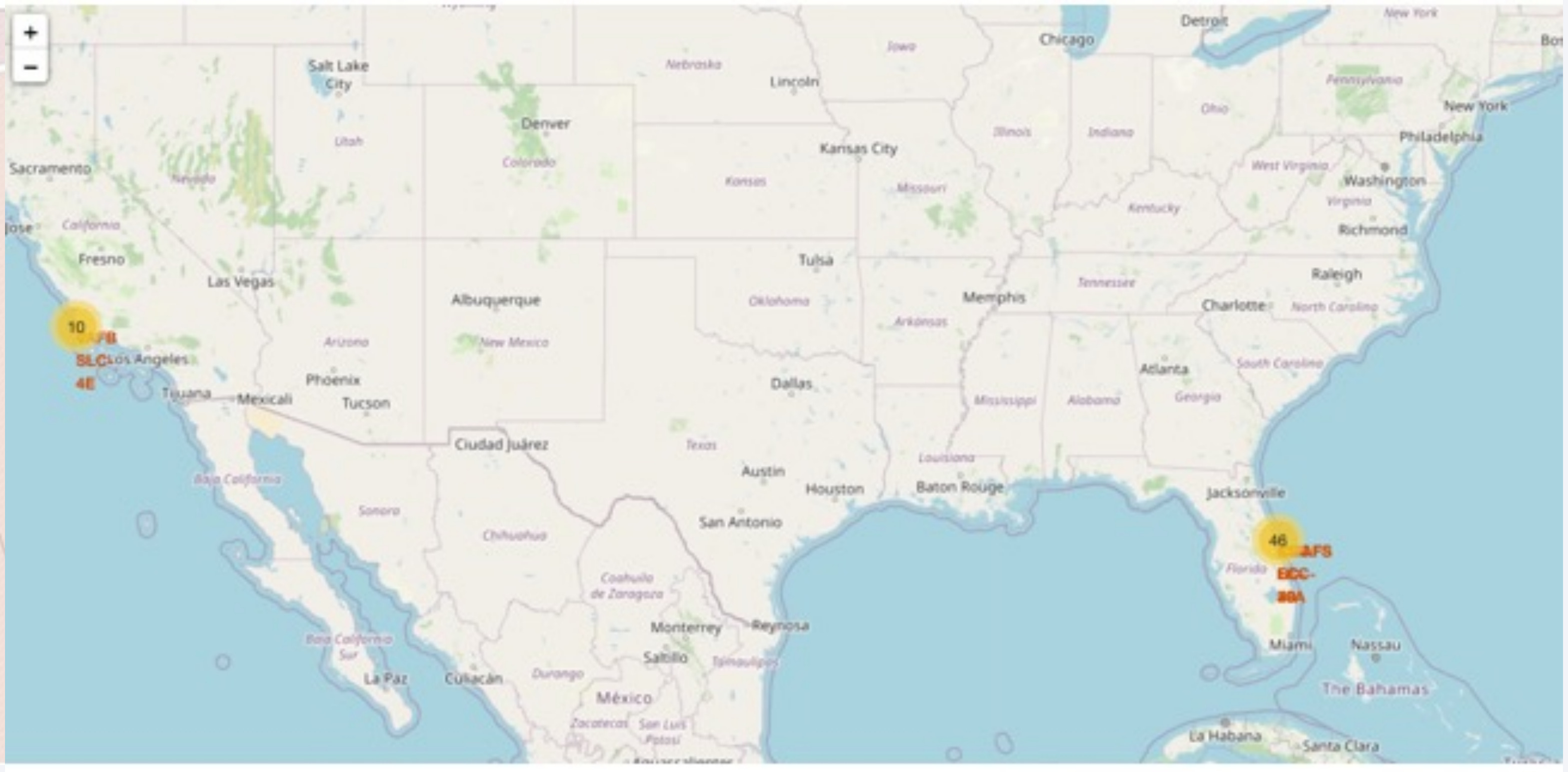
Date	COUNT(Landing_Outcome)
2015-12-22	1
2016-04-08	1
2016-05-06	1
2016-05-27	1
2016-07-18	1
2016-08-14	1
2017-01-14	1
2017-02-19	1

Section 3

Launch Sites Proximities Analysis

all launch sites

the success/failed launches for each site

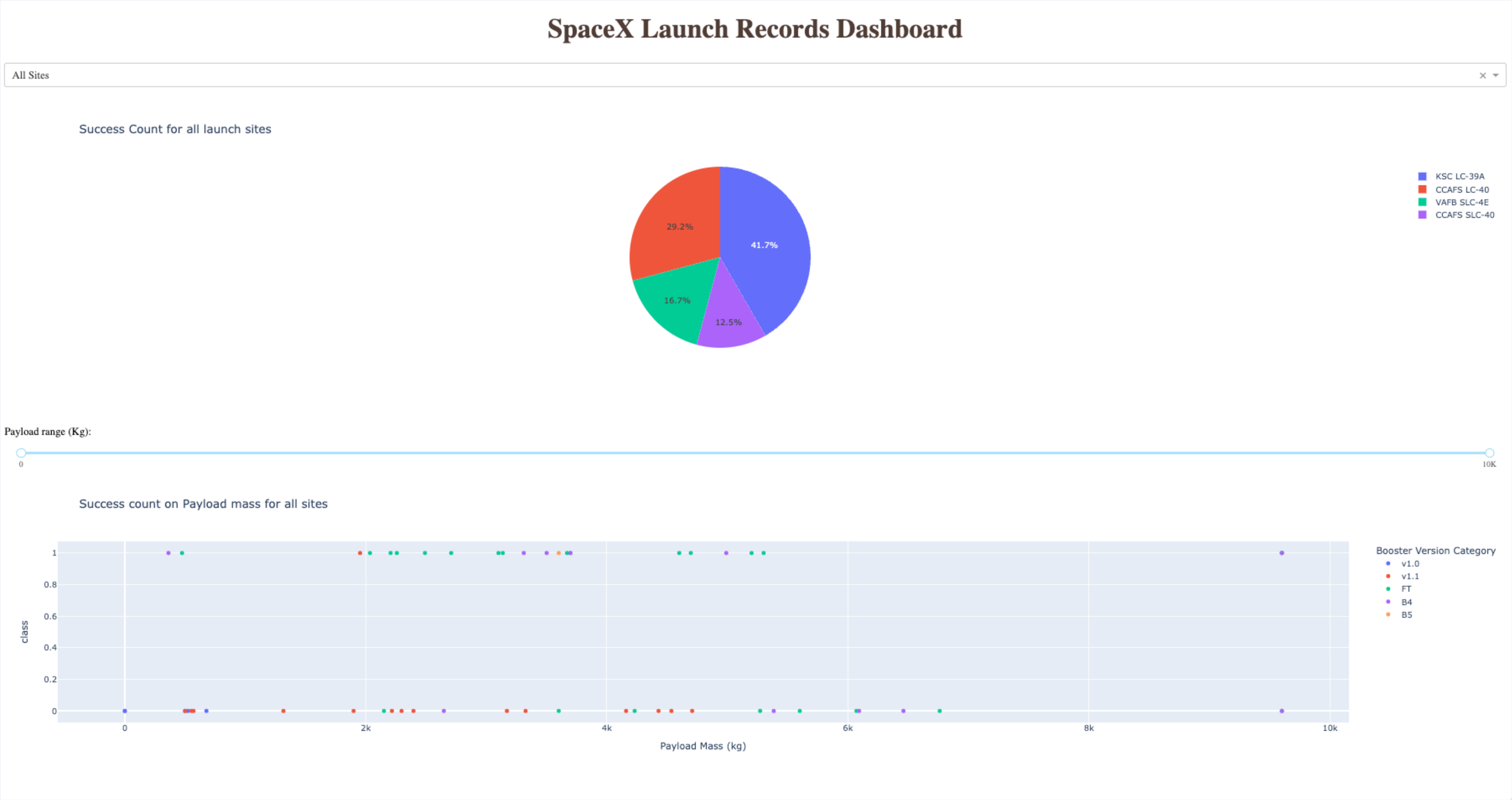


Calculate the distances between a launch site to its proximities

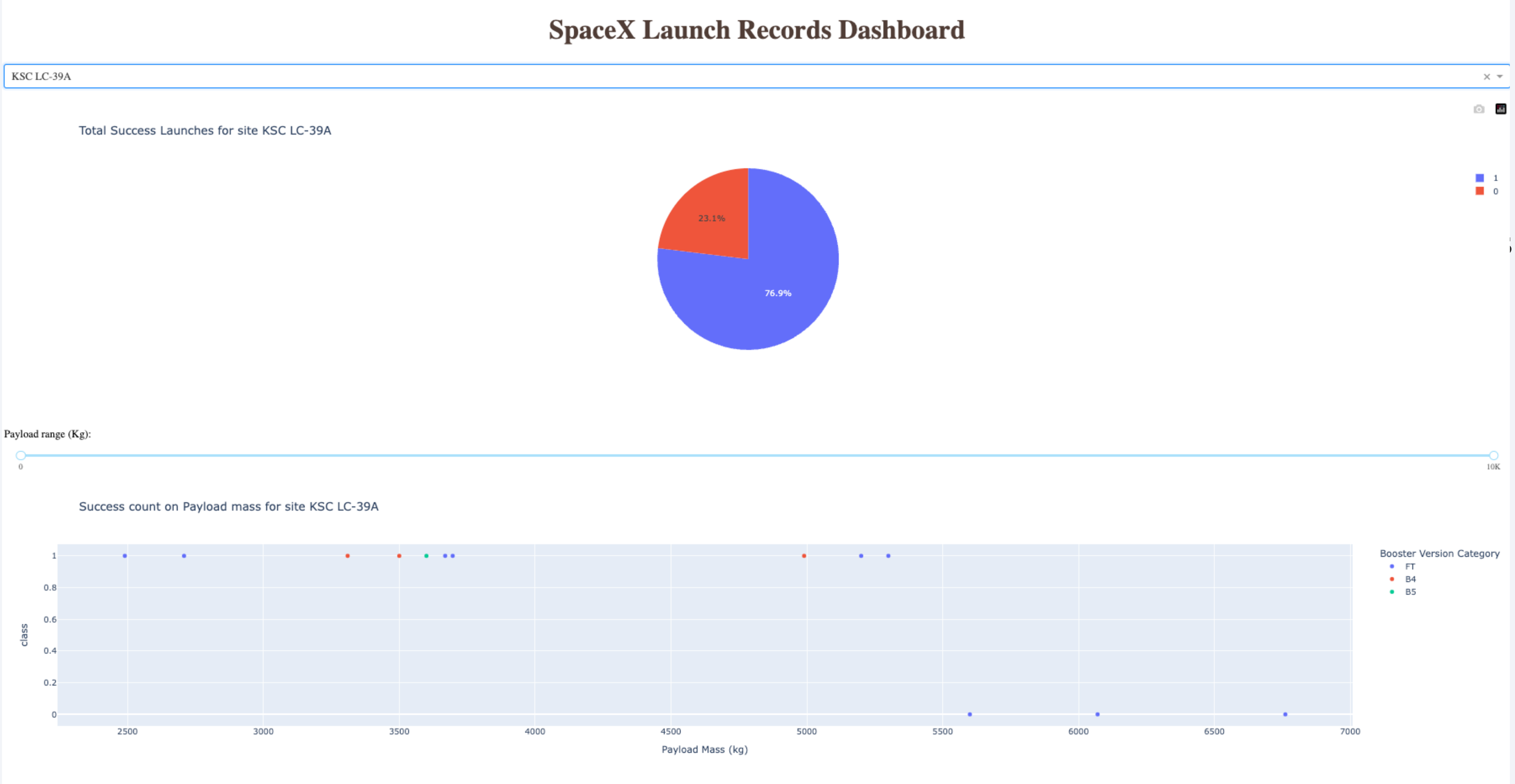
Section 4

Build a Dashboard with Plotly Dash

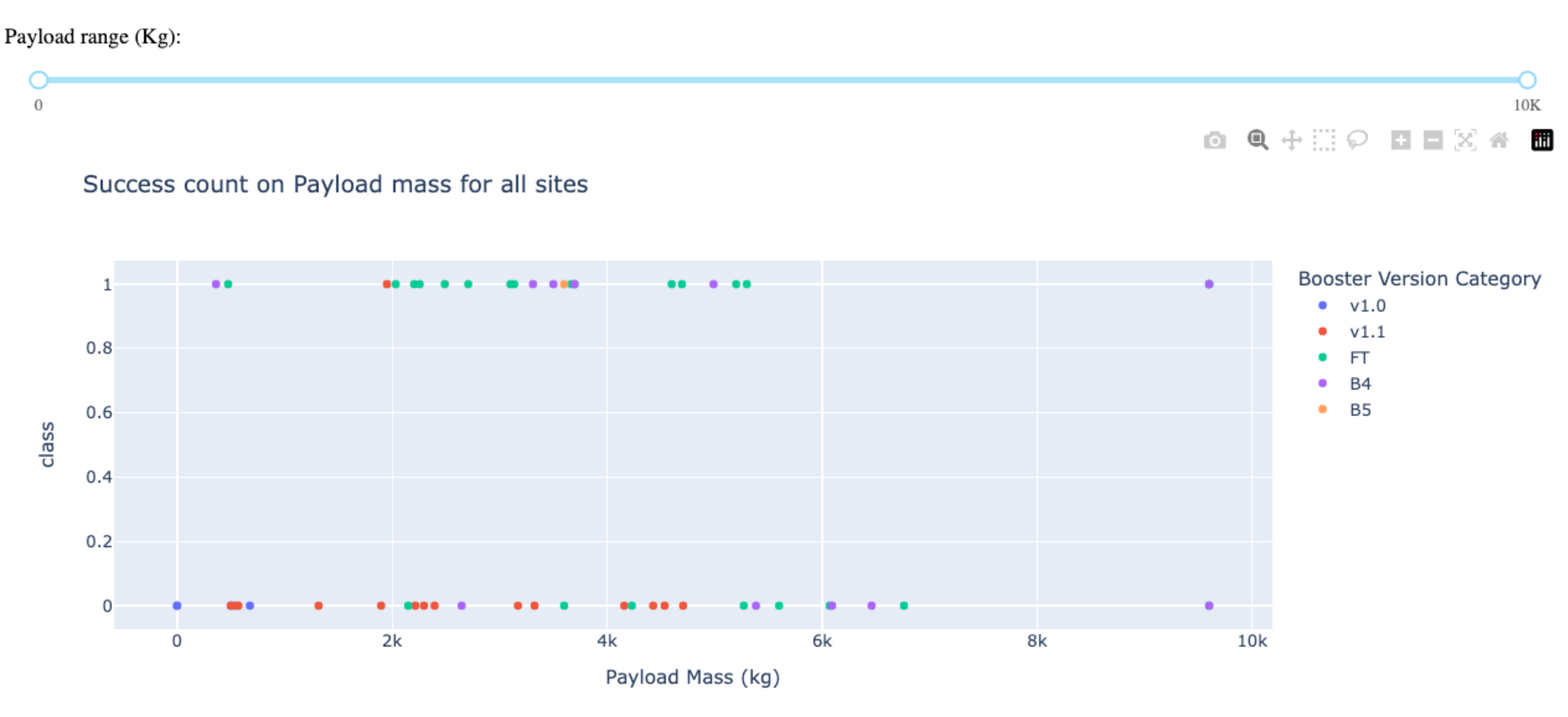
All sites


SpaceX Launch Records Dashboard
All Sites
Success Count for all launch sites
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40
Payload range (Kg):
0
10K
Success count on Payload mass for all sites
class
1
0.8
0.6
0.4
0.2
0
2k
4k
6k
8k
10k
Payload Mass (kg)
Booster Version Category
v1.0
v1.1
FT
B4
B5

Dashboard for KSC LC-39A



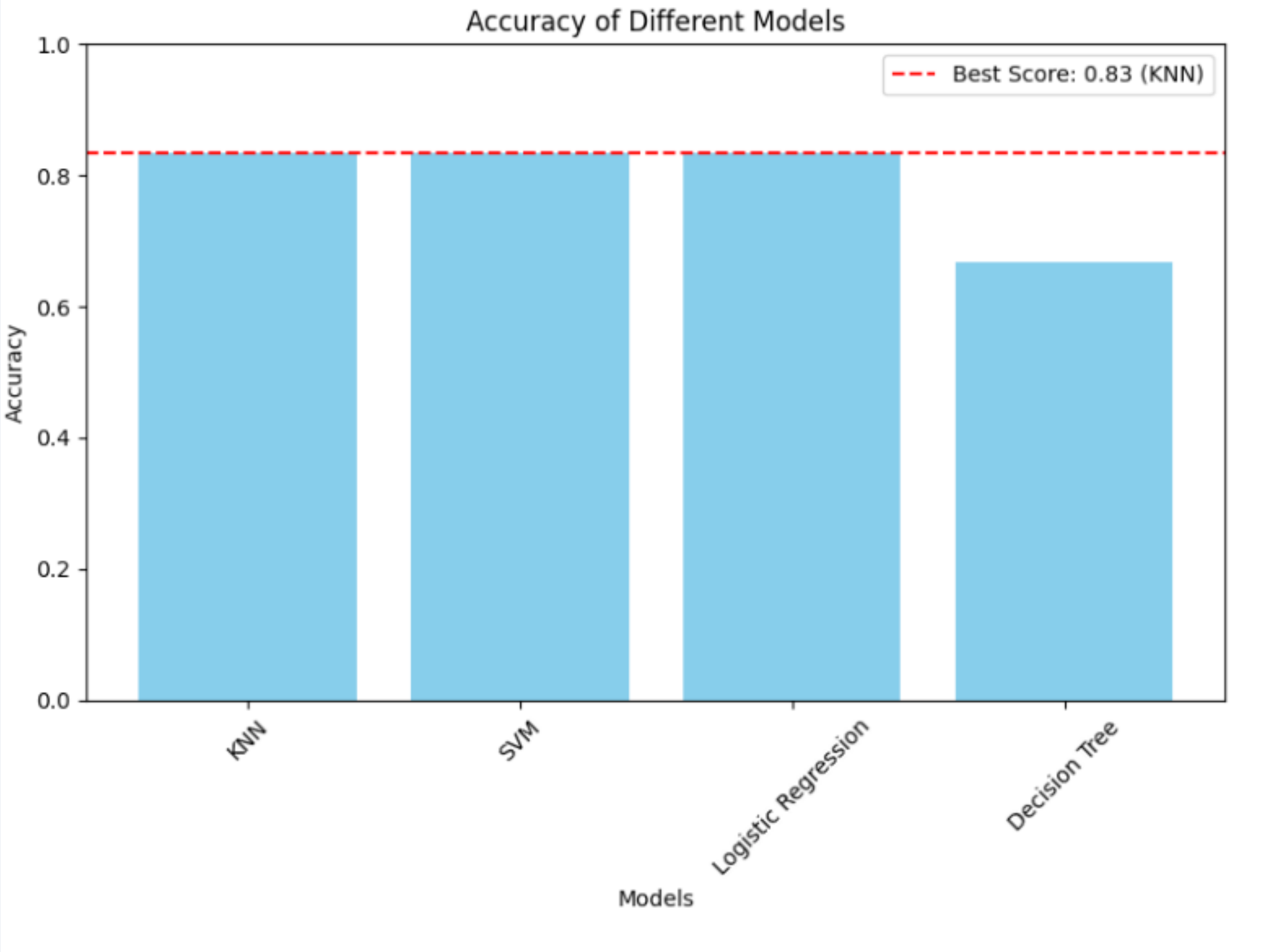
Payload vs. Launch Outcome for all sites



Section 5

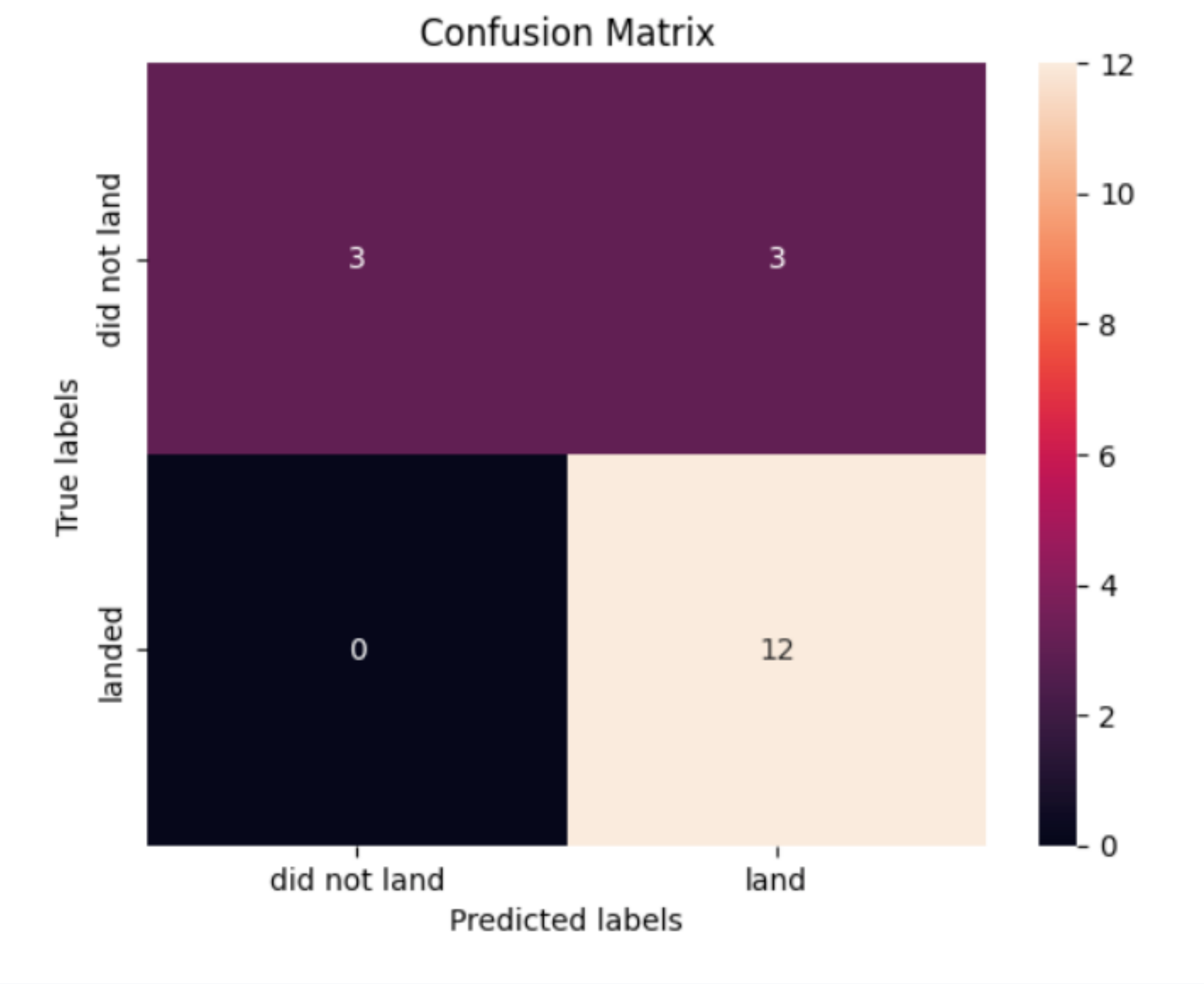
Predictive Analysis (Classification)

Classification Accuracy




```
Best Score: 0.8333333333333334
[33]:
▸ GridSearchCV
▸ estimator: KNeighborsClassifier
▸ KNeighborsClassifier
```

Confusion Matrix




```
Best Score: 0.8333333333333334
[33]:
▸ GridSearchCV
▸ estimator: KNeighborsClassifier
▸ KNeighborsClassifier
```

Conclusions

1. **Key Predictors:** Identification of crucial factors influencing the successful landing of Falcon 9 first stages.
2. **Predictive Capability:** Confirmation of the model's ability to reasonably predict landing outcomes based on historical data.
3. **Cost Implications:** Understanding the cost impact associated with successful versus unsuccessful landings, aiding in estimating launch costs.
4. **Spatial Analysis:** Insights into the significance of launch site locations and their influence on mission success rates.

Thank you!