Winning Space Race with Data Science

<Name>
<Date>

Outline

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

Executive Summary

Summary of methodologies

- Data Collection API
- Data Collection Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- EDA Visualization
- Interactive Visual Analytics with folium
- Interactive Dashboard with Poly Dash
- Predictive Analyis

Summary of all results

- EDA results
- Interactive Analytics
- Predictive Analytics

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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

Problems you want to find answers

predict if the Falcon 9 first stage will land successfully

Section 1

Methodology

Methodology

1. **Data Collection, Wrangling, and Formatting**
   - Utilized **SpaceX API** for gathering data
   - Employed **web scraping** techniques for additional data extraction
2. **Exploratory Data Analysis (EDA)**
   - Performed analysis with **Pandas** and **NumPy**
   - Utilized **SQL** for querying and data manipulation
3. **Data Visualization**
   - Visualized data using **Matplotlib** and **Seaborn**
   - Created interactive maps with **Folium**
   - Built dynamic dashboards using **Dash**
4. **Machine Learning Prediction**
   - Applied **Logistic Regression** for binary classification
   - Tested with **Support Vector Machine (SVM)**
   - Used **Decision Tree** for model interpretation
   - Explored **K-Nearest Neighbors (KNN)** for comparison

Data Collection

Space X data was gathered from SpaceX REST API

From: api.spacexdata.com/v4/

Data Source: Data was collected from the SpaceX API, specifically focusing on Falcon 9 launches.

Webscraping Wikipedia using Beautiful soup

Data Collection – SpaceX API

GitHub URL of the completed SpaceX API calls notebook:

https://github.com/JMCS111/testrepo/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

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

```
# use requests.get() method with the provided static_url
# assign the response to a object

data  = requests.get(static_url).text
```

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

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text conten

soup = BeautifulSoup(data, 'html.parser')
```

```
# Assign the result to a list called  html_tables
html_tables = soup.find_all('table')
```

Next, we just need to iterate through the `<th>` elements and apply the provided `extract_column_from_header()` to extract colu
name one by one

```
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
for row in first_launch_table.find_all('th'):
    name = extract_column_from_header(row)
    if (name != None and len(name) > 0):
        column_names.append(name)
```

After you have fill in the parsed launch record values into `launch_dict`, you can create a dataframe from it.

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

We can now export it to a **CSV** for the next section, but to make the answers consistent and in case you have diffic

https://github.com/JMCS111/testrepo/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- **Key Functions:**
  - **Launch Site Analysis:** Determine the **number of launches** per launch site.
  - **Orbit Analysis:** Identify the **number of occurrences** for each type of orbit.
  - **Mission Outcome Analysis:** Count the **number of occurrences** for each mission outcome (success or failure).

- https://github.com/JMCS111/testrepo/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Matplotlib and Seaborn
- • Functions from the Matplotlib and Seaborn libraries are used to visualize the data through
- scatterplots, bar charts, and line charts.
- • The plots and charts are used to understand more about the relationships between several
- features, such as:
- • The relationship between flight number and launch site
- • The relationship between payload mass and launch site
- • The relationship between success rate and orbit type

https://github.com/JMCS111/testrepo/blob/main/edadataviz.ipynb

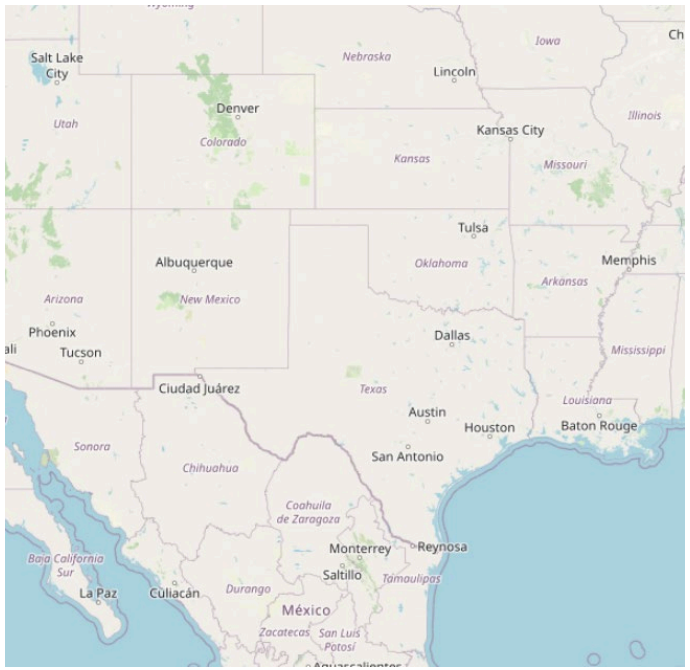
EDA with SQL

- **SQL**
- **Purpose:** SQL queries are used to extract specific insights from the dataset by performing structured data queries.
- **Key Queries:**
  - **Unique Launch Sites:** Query to list the **names of unique launch sites** in the SpaceX missions.
  - **Payload Mass by NASA (CRS):** Query to determine the **total payload mass** carried by boosters launched by NASA (CRS missions).
  - **Average Payload Mass for F9 v1.1:** Query to calculate the **average payload mass** carried by the **F9 v1.1** booster version.

- https://github.com/JMCS111/testrepo/blob/main/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities
- https://github.com/JMCS111/testrepo/blob/main/lab_jupyter_launch_site_location%20(1).ipynb



Build a Dashboard with Plotly Dash

- • Functions from Dash are used to generate an interactive site where we can toggle the input
- using a dropdown menu and a range slider.
- • Using a pie chart and a scatterplot, the interactive site shows:
- • The total success launches from each launch site
- • The correlation between payload mass and mission outcome (success or failure) for each launch
- Site
- https://github.com/JMCS111/testrepo/blob/main/spacex_dash_app.py.1

Predictive Analysis (Classification)

Objective: Train multiple machine learning models to predict if the booster will land successfully or not.

Models Used:

Logistic Regression

Support Vector Machine (SVM)Decision Tree

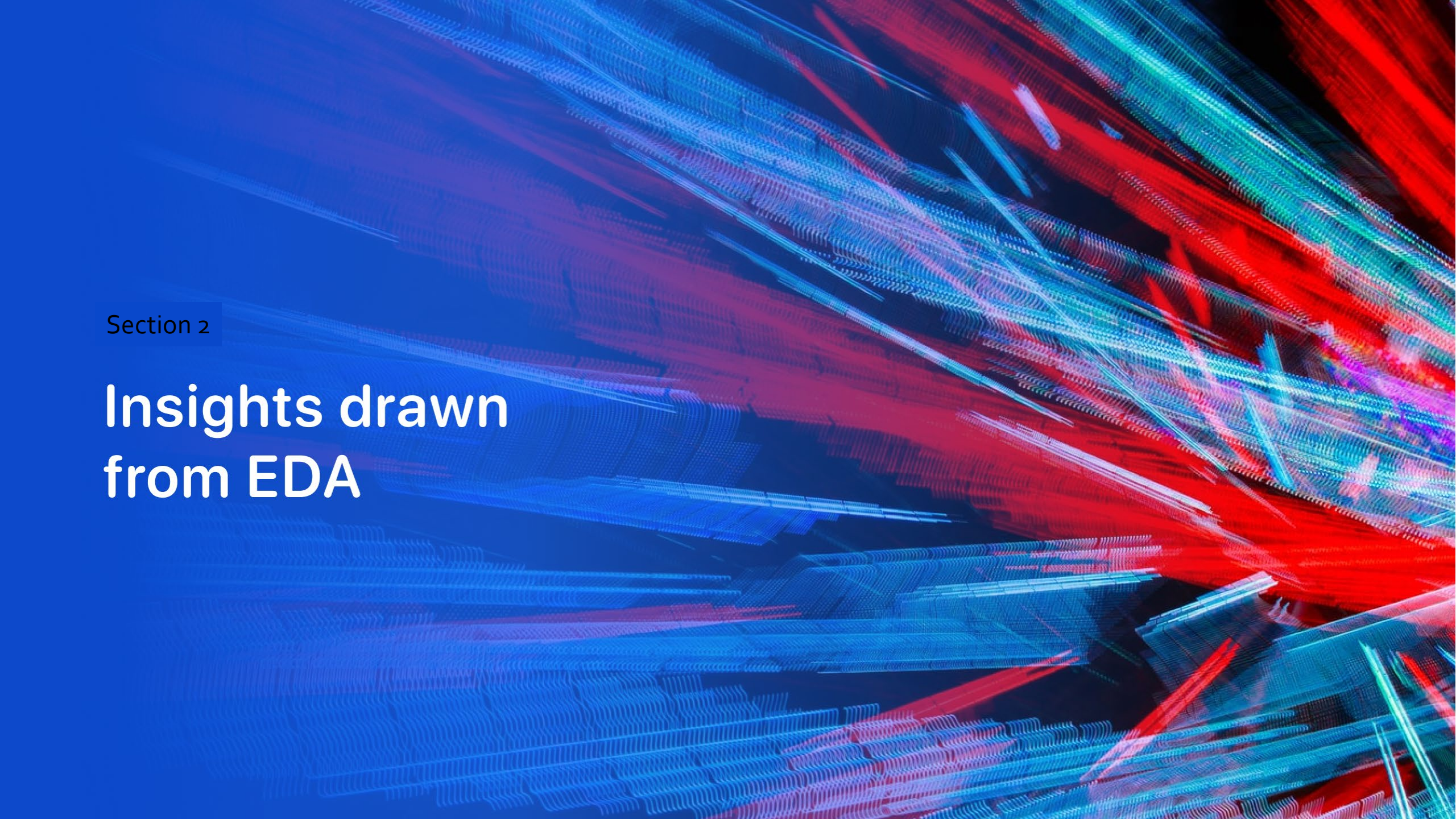
K-Nearest Neighbors (KNN)

Labeling: The launch outcomes were converted into binary labels (1 = success, 0 = failure) for model training.

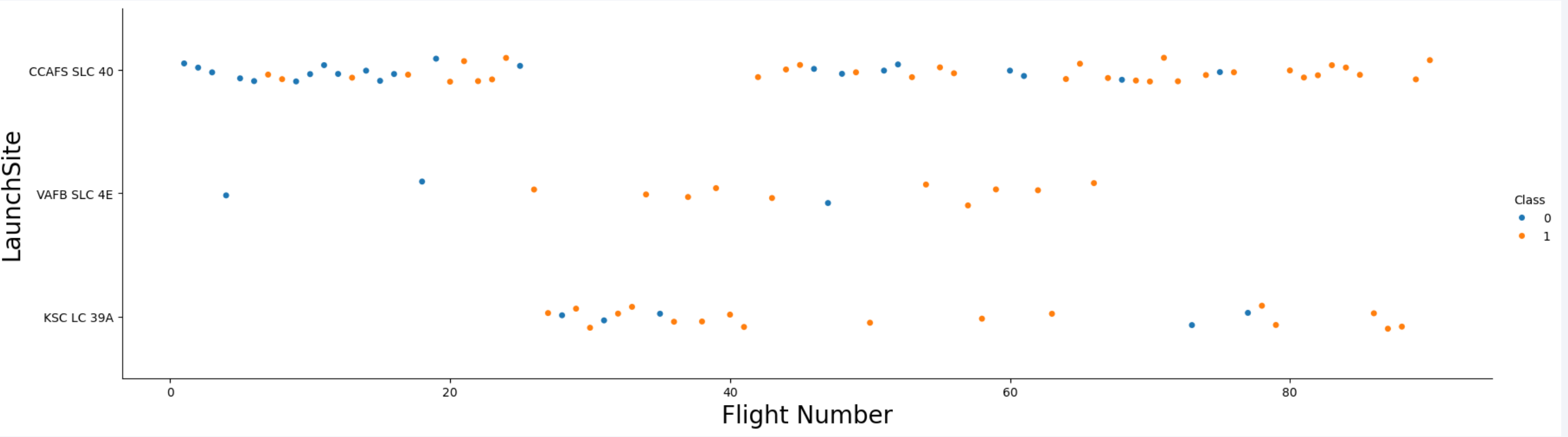
Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

Results

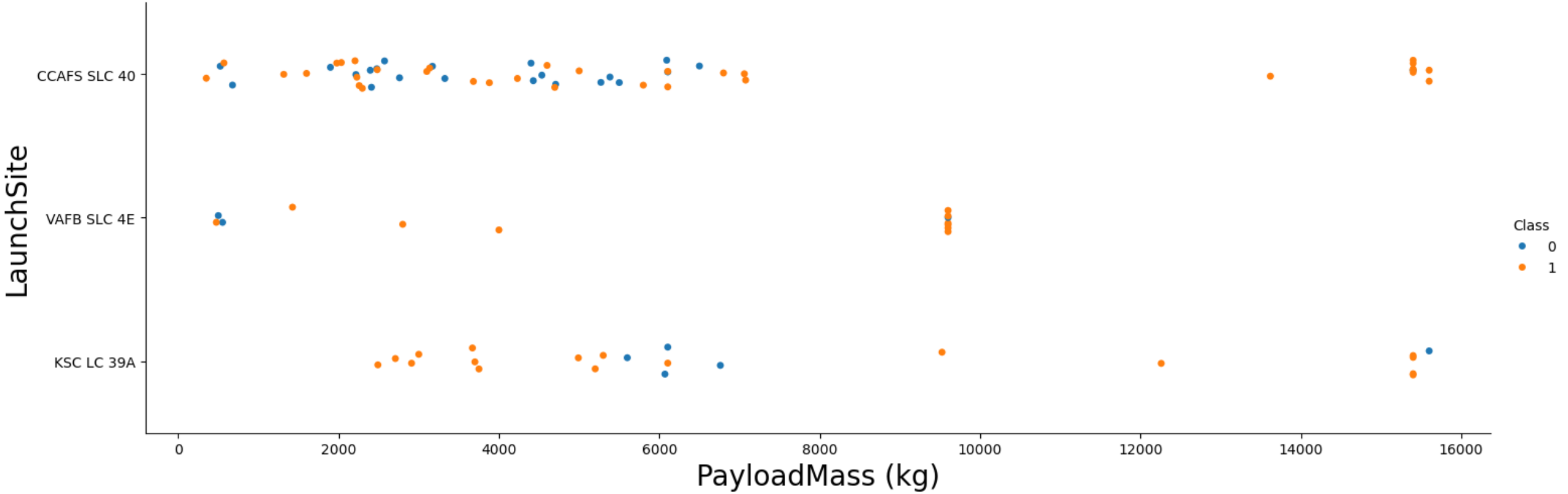
- The SVM, KNN, and Logistic Regression models are the best in terms of prediction accuracy for this dataset.
- Low weighted payloads perform better than the heavier payloads.
- The success rates for SpaceX launches is directly proportional time in years they will eventually perfect the launches.
- KSC LC 39A had the most successful launches from all the sites.
- Orbit GEO,HEO,SSO,ES L1 has the best Success Rate.


Section 2
Insights drawn from EDA

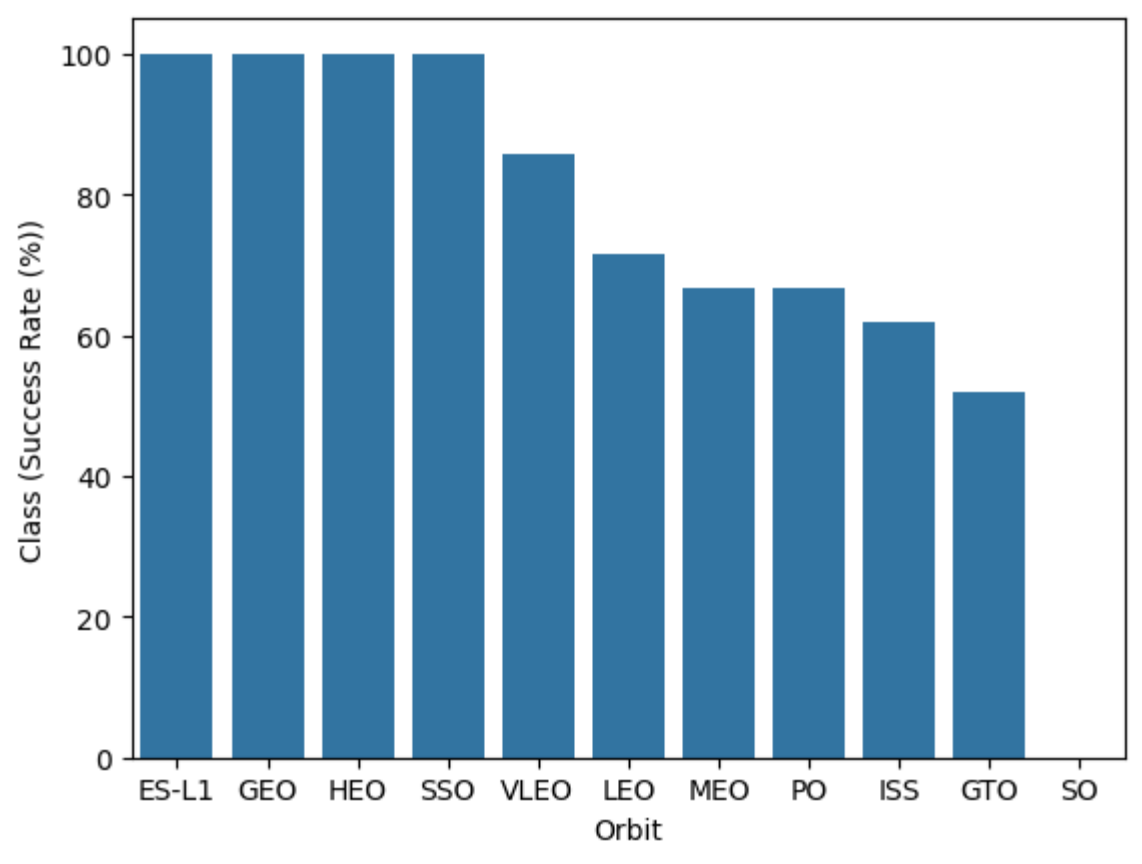
Flight Number vs. Launch Site



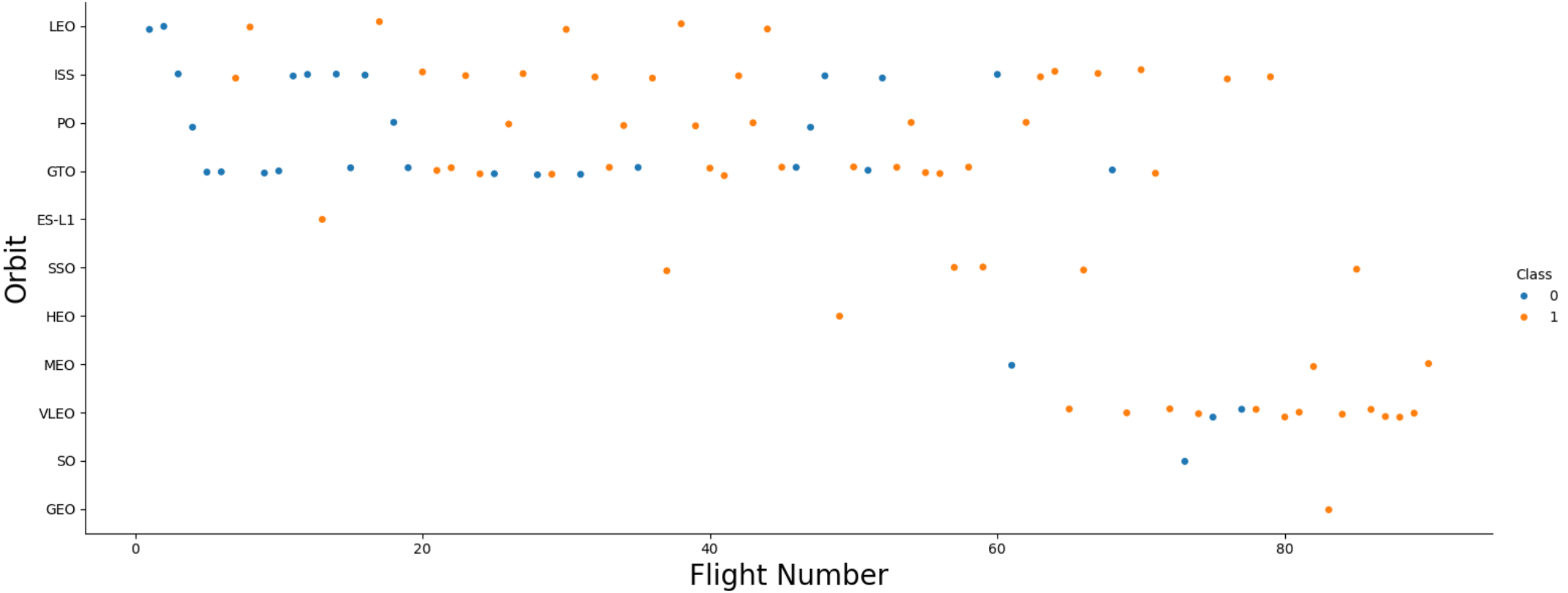
Payload vs. Launch Site



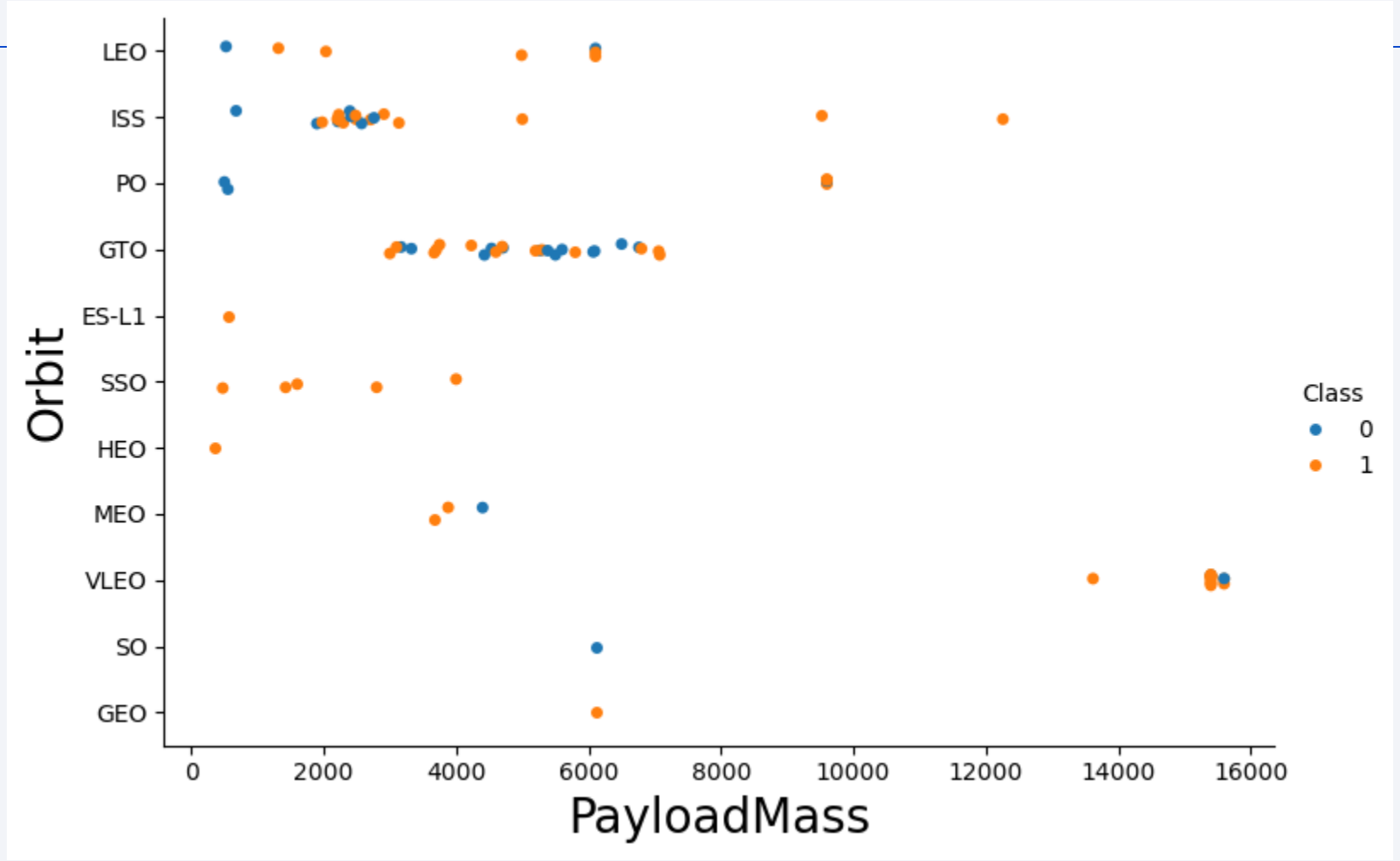
Success Rate vs. Orbit Type



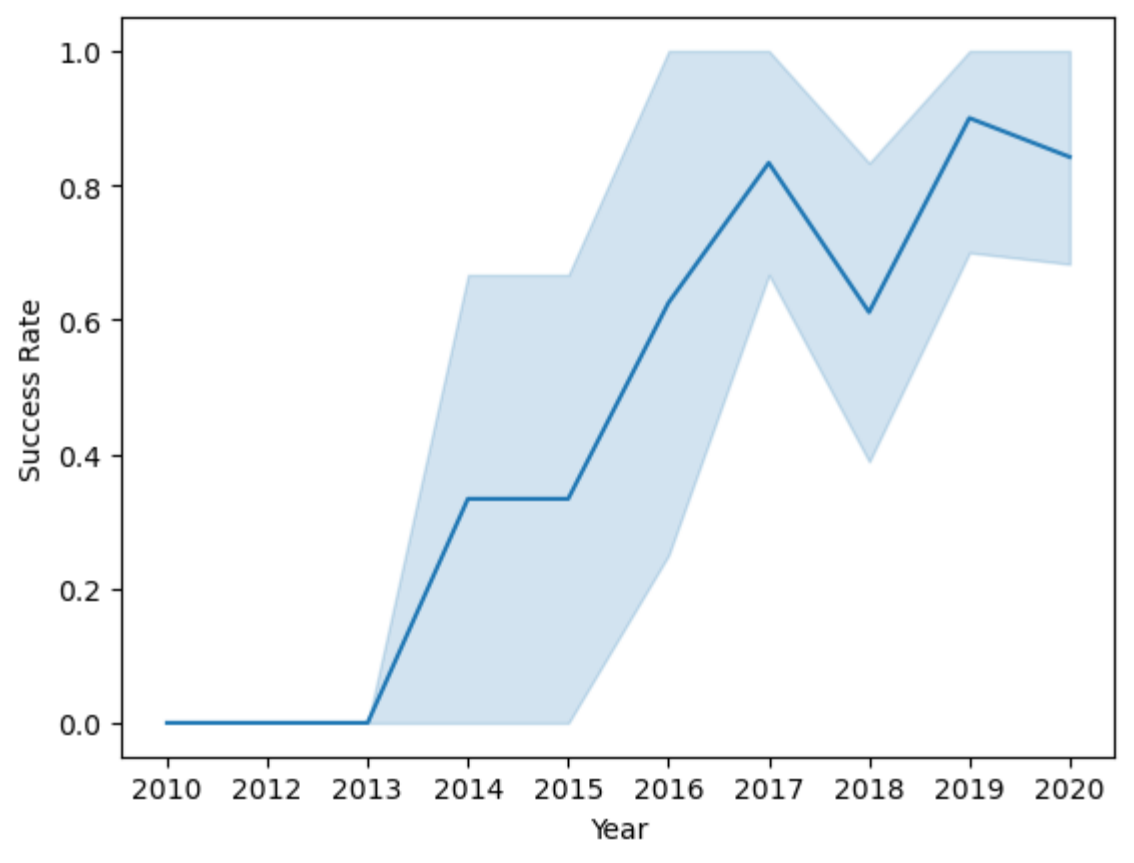
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

LaunchSite	Fli
CCAFS SLC 40	
CCAFS SLC 40	
CCAFS SLC 40	
VAFB SLC 4E	
CCAFS SLC 40	

Launch Site Names Begin with 'CCA'

launch_site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

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

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

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

]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version 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

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT MIN(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (ground pad)']
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

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 than 4000 but less than 6000

```
]:
%sql SELECT PAYLOAD \
FROM SPACEXTBL \
WHERE LANDING__OUTCOME = 'Success (drone ship)' \
AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: LANDING__OUTCOME
[SQL: SELECT PAYLOAD FROM SPACEXTBL WHERE LANDING__OUTCOME = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 60
00;]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

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

Mission_Outcome	total_number
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

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

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

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

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.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

```
%sql SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] \
FROM SPACEXTBL \
where [Landing _Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';
```

```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: Landing _Outcome
[SQL: SELECT substr(Date,4,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing _Outcome] FROM SPACEXTBL where [Landing _
Outcome] = 'Failure (drone ship)' and substr(Date,7,4)='2015';]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

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

Task 10

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.

37]:

```
%sql SELECT [Landing _Outcome], count(*) as count_outcomes \
FROM SPACEXTBL \
WHERE DATE between '04-06-2010' and '20-03-2017' group by [Landing _Outcome] order by count_outcomes DESC;
```

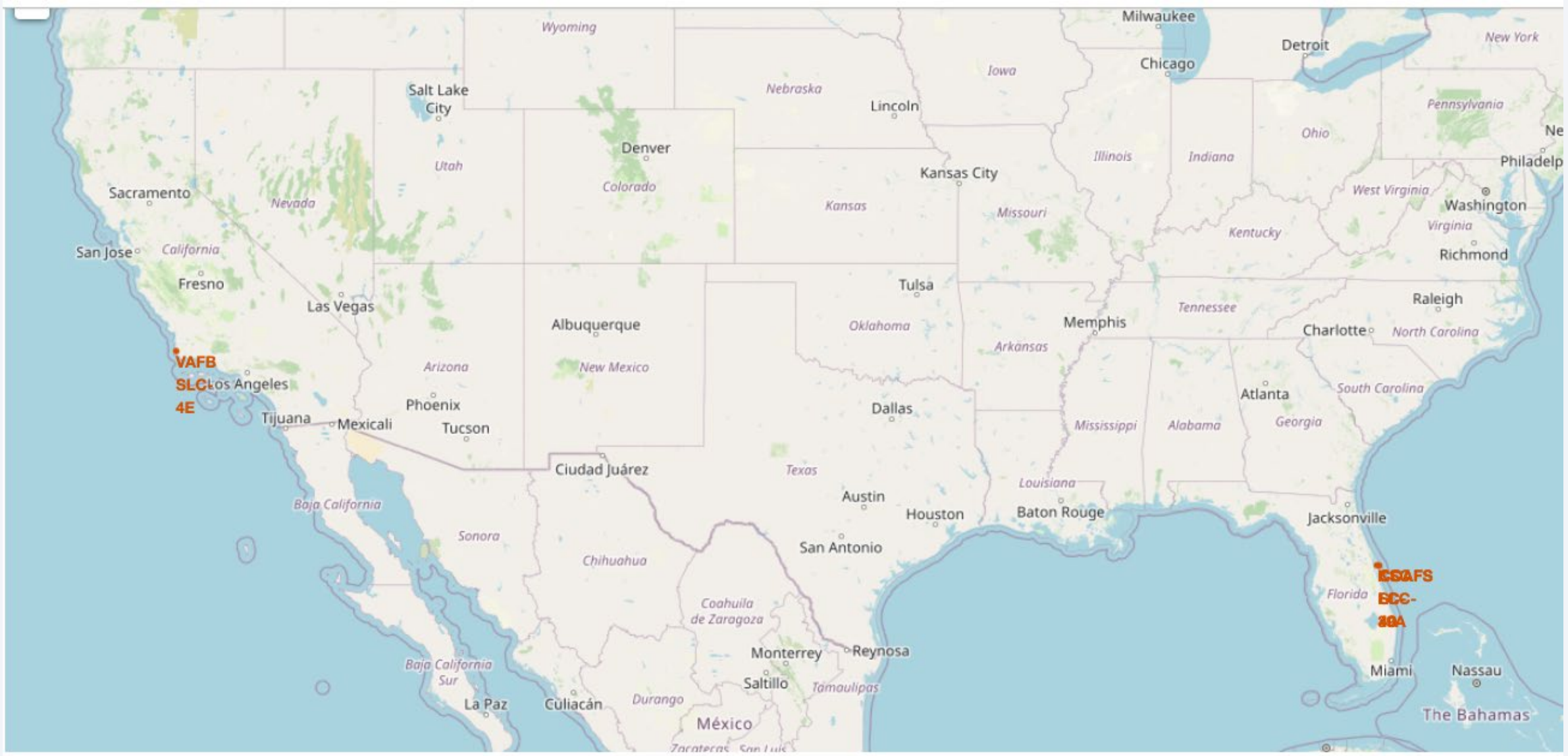
```
 * sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: Landing _Outcome
[SQL: SELECT [Landing _Outcome], count(*) as count_outcomes FROM SPACEXTBL WHERE DATE between '04-06-2010' and '20-03-2017' g
roup by [Landing _Outcome] order by count_outcomes DESC;]
(Background on this error at: https://sqlalche.me/e/20/e3q8)
```

Reference Links

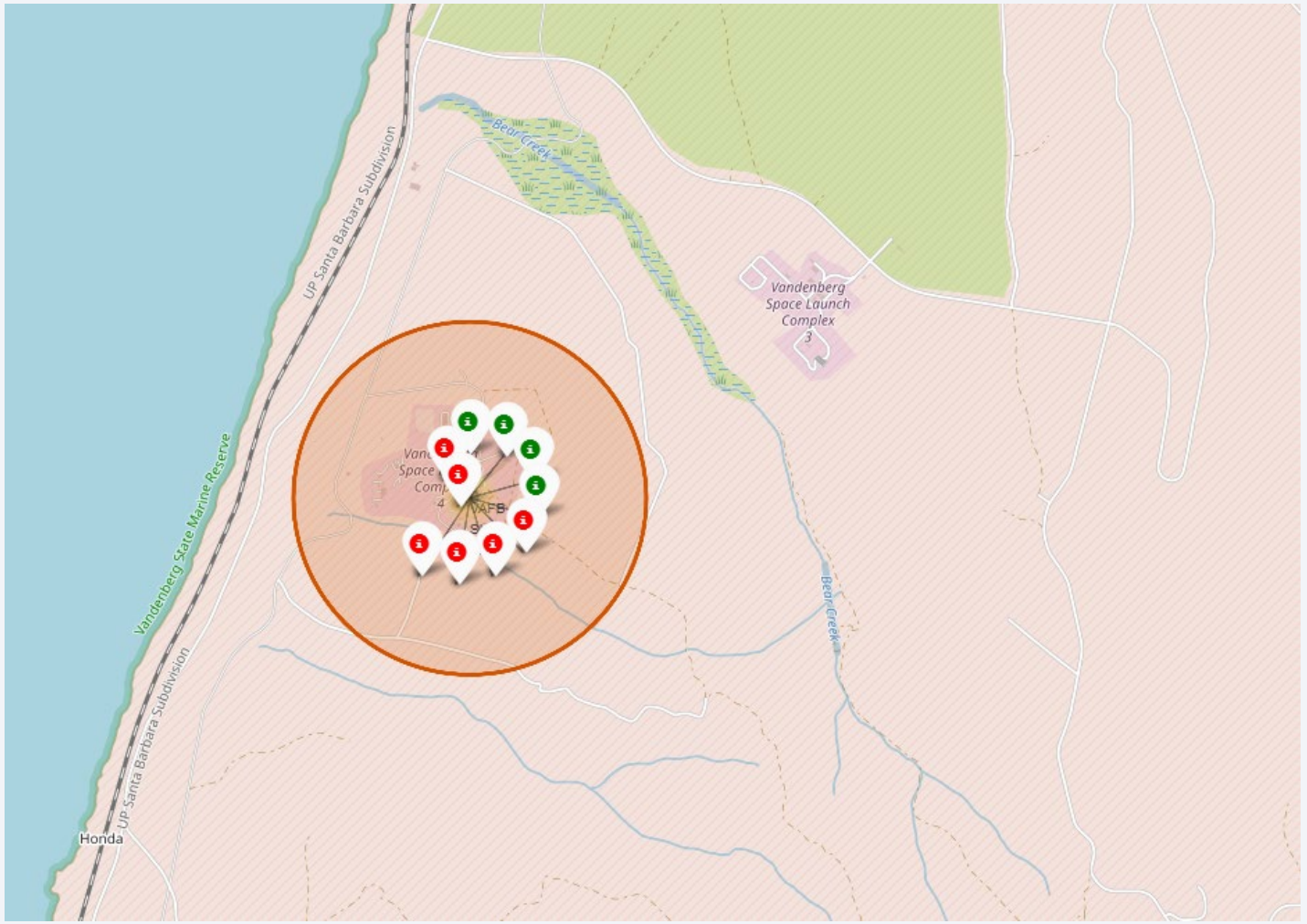
Section 3

Launch Sites Proximities Analysis

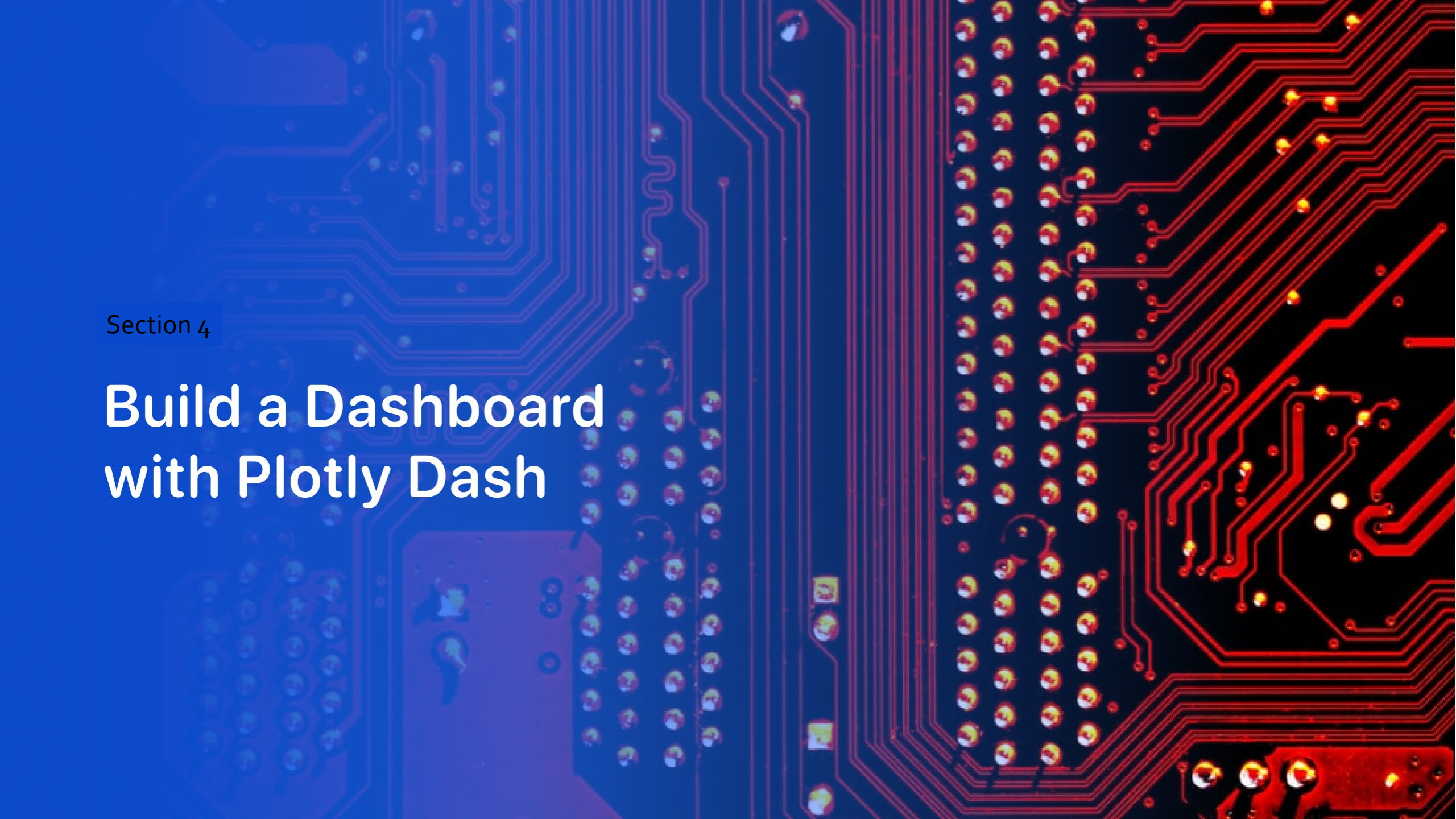
Space X: All launch sites on map



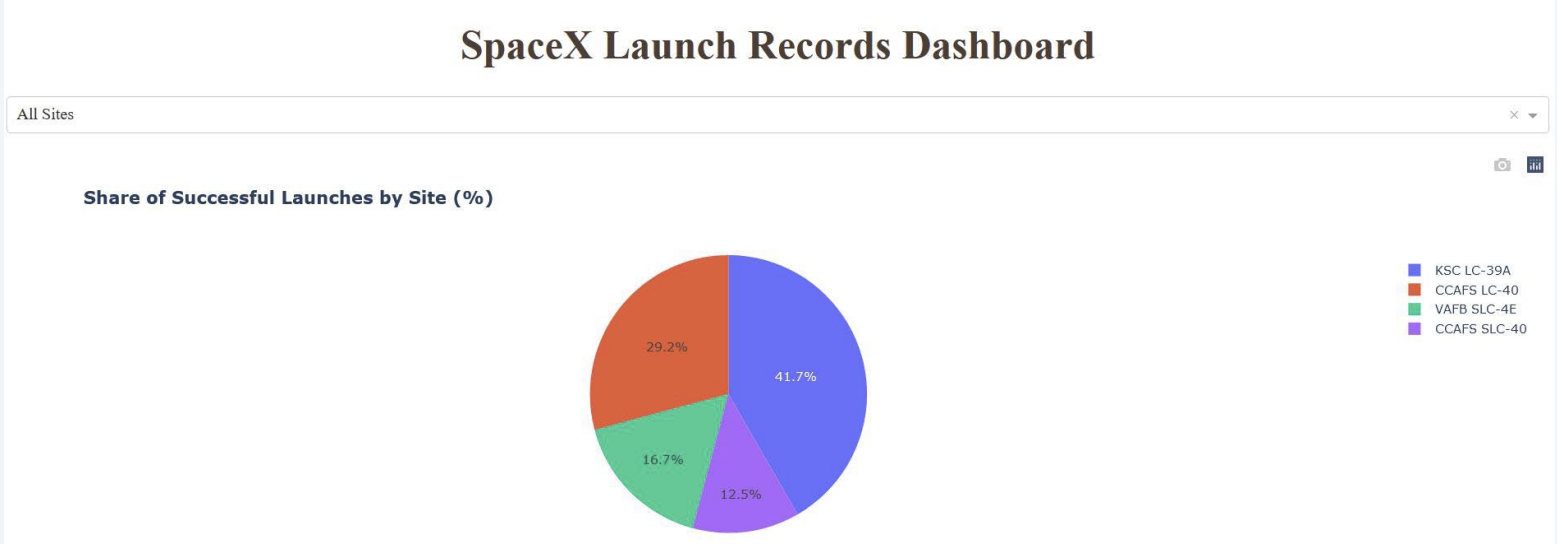
Falcon 9 Success/Failed launches for each site



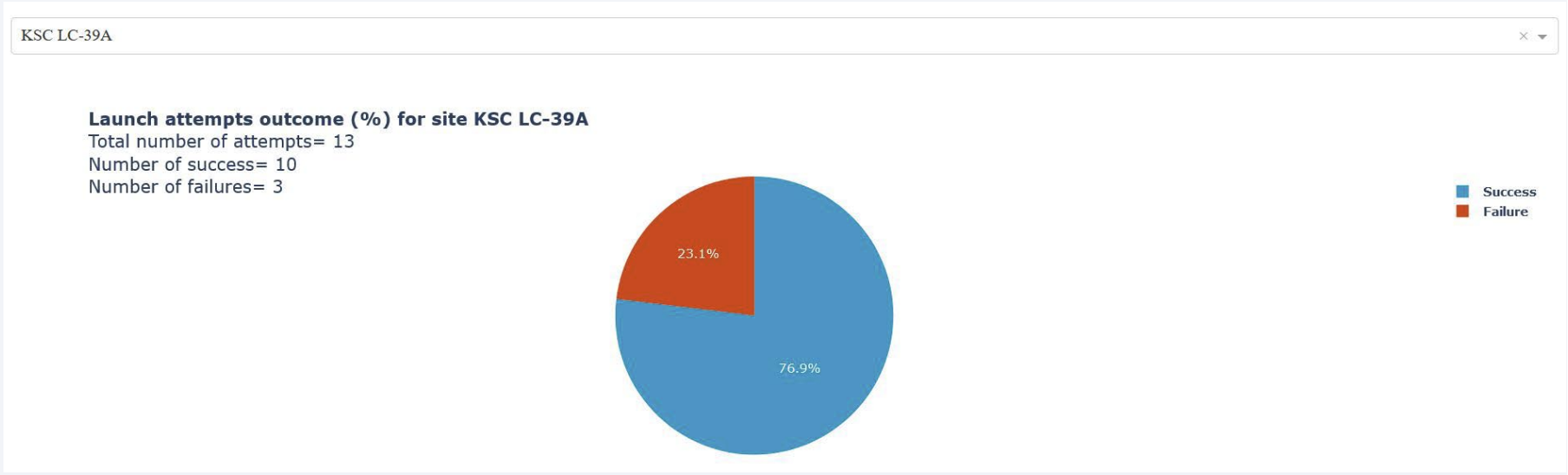
Distances between a launch site to its proximities

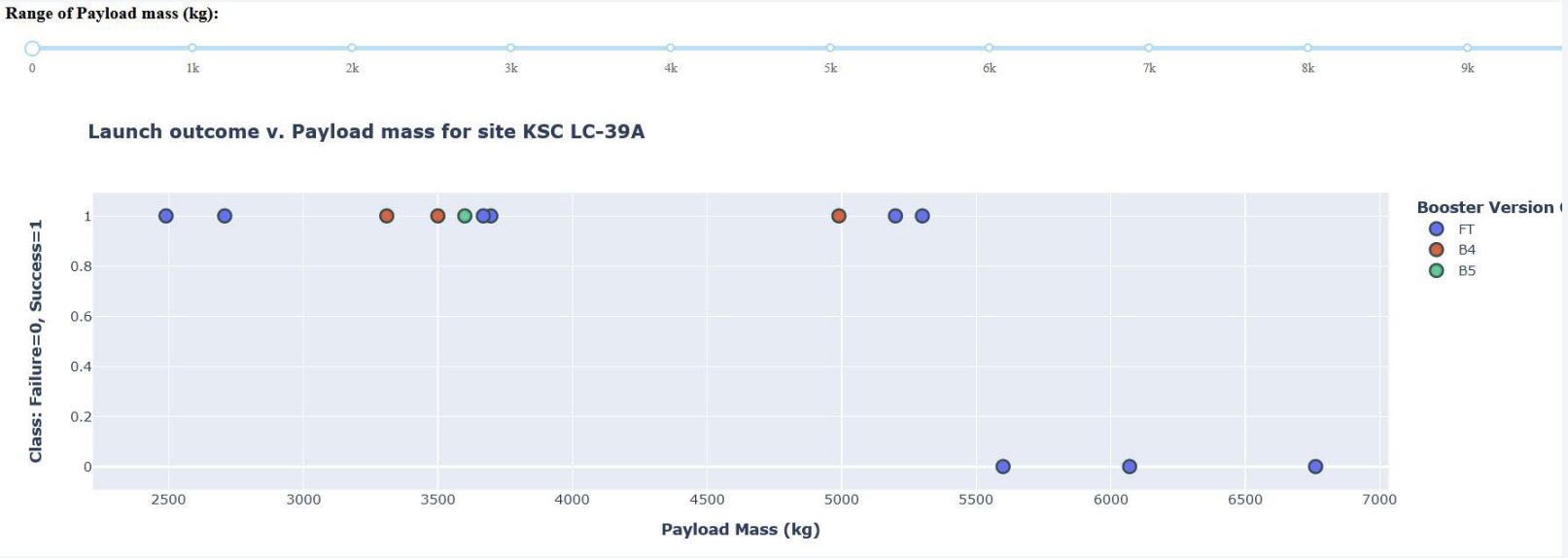

Section 4
Build a Dashboard with Plotly Dash

SpaceX Falcon 9: Launch success count for all sites

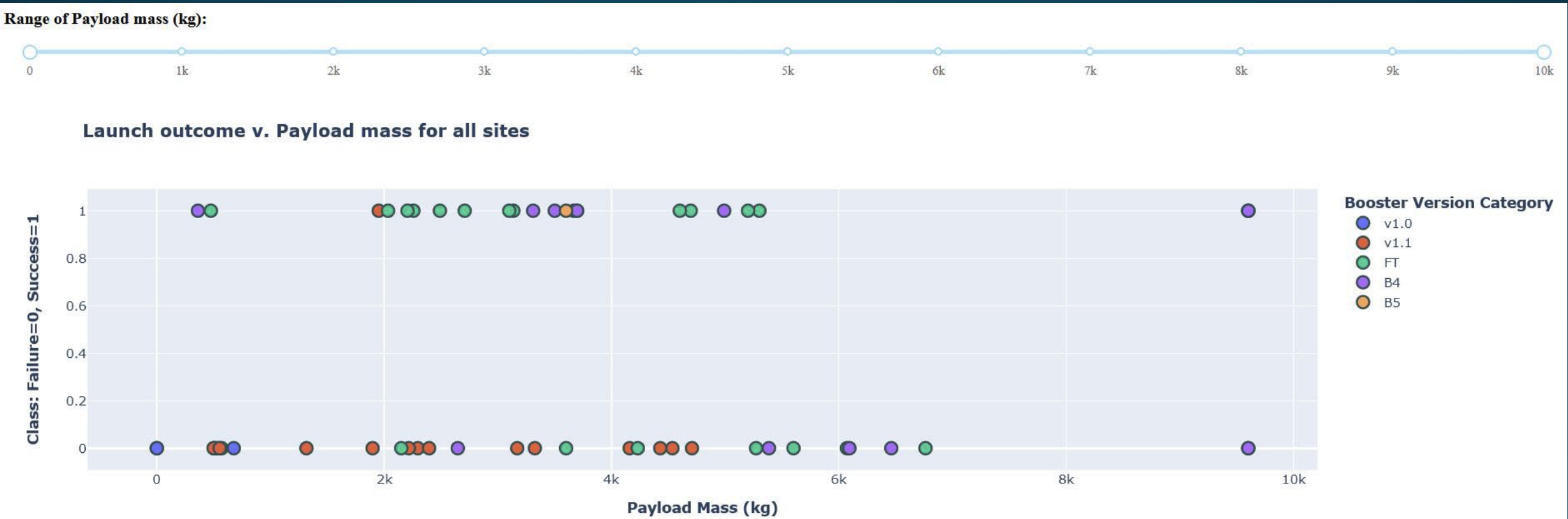


SpaceX Falcon9 Launch site with highest launch success ratio





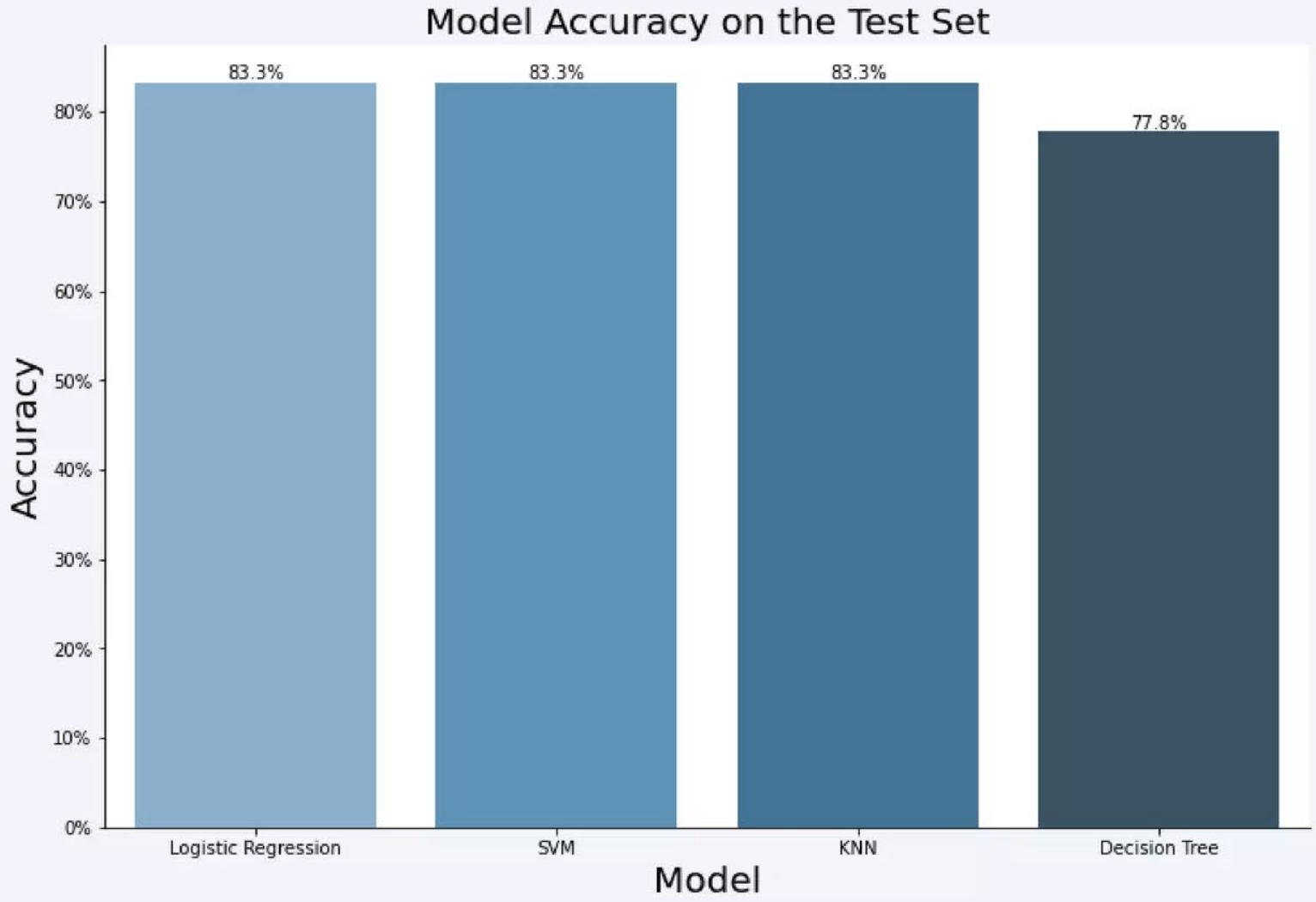
Launch outcome v. Payload mass (all sites)



Section 5

Predictive Analysis (Classification)

Classification Accuracy



Conclusions

•**Objective:**
Predict whether the first stage of a Falcon 9 launch will land successfully, which directly influences the **cost of a launch**.
•**Key Features:**
Features such as **payload mass**, **orbit type**, and other launch parameters may influence the mission outcome (successful or failed landing).
•**Machine Learning Approach:**
Several **machine learning algorithms** were used to identify patterns in historical Falcon 9 launch data, helping to develop predictive models for future launches.
•**Best Performing Model:**
The **decision tree algorithm** yielded the best performance among the four models employed, showcasing the most accurate predictions for the landing outcome.

Appendix

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!