IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Chukwuemeka Okoli

2nd December 2021

Outline

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

Executive Summary

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

Introduction

- Project background and context

  SpaceX offers Falcon 9 rocket launches on its website for $62 million, whereas other providers charge upwards of $165 million. The primary reason for SpaceX's lower costs is the reusability of the first stage. By predicting the successful landing of the first stage, we can estimate the launch cost. This could be useful for other companies aiming to compete with SpaceX for rocket launches. The objective of this project is to build a machine learning pipeline to predict the successful landing of the first stage.

- Problems you want to find answers

  - What elements influence whether a rocket lands successfully?
  - The interplay among various factors that affect the success rate of a rocket landing.
  - What operating conditions must be in place to guarantee a successful landing program?.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
    - One-hot encoding was applied to categorical features
- 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

- Data was gathered through multiple methods.
- A GET request to the SpaceX API was used for data collection.
- The response content was decoded into JSON format using the `.json()` function.
- The JSON data was converted into a pandas DataFrame with `.json_normalize()`.
- The data was cleaned, checked for missing values, and filled where necessary.
- Web scraping was performed on Wikipedia to obtain Falcon 9 launch records using BeautifulSoup.
- Launch records were extracted from an HTML table, parsed, and converted into a pandas DataFrame for future analysis.

Data Collection – SpaceX API

- We utilized a GET request to the SpaceX API to gather data, then cleaned the retrieved data and performed basic data wrangling and formatting.
- The link to the notebook is Applied-Data-Science-Capstone/Lab 1 Collecting the Data.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)

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 [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_
```

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

In [10]:
```
response.status_code
```

Out[10]: 200

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

In [11]:
```
# 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 [12]:
```
# Get the head of the dataframe
data.head(1)
```

Data Collection - Scraping

- We used web scraping to gather Falcon 9 launch records with BeautifulSoup. The data was parsed from the table and converted into a pandas DataFrame.
- The link to the notebook is <u>Applied-Data-Science-Capstone/Lab 1 Web Scraping.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)</u>

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 [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`

In [6]:
```
# 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

In [7]:
```
# 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 `BeautifulSoup`, please check the external reference link towards the end of this lab

In [8]:
```
# 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")
print(html_tables)
```

Data Wrangling

- We conducted exploratory data analysis and identified the training labels.
- We calculated the number of launches at each site and analyzed the frequency and occurrence of each orbit.
- Additionally, we generated landing outcome labels from the outcome column and exported the results to a CSV file.
- The link to the notebook is Applied-Data-Science-Capstone/Lab 2 Data Wrangling.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)

TASK 1: Calculate the number of launches on each site

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A** .The location of each Launch Is placed in the column `LaunchSite`

Next, let's see the number of launches for each site.

Use the method `value_counts()` on the column `LaunchSite` to determine the number of launches on each site:

In [20]:
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

Out[20]:
```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

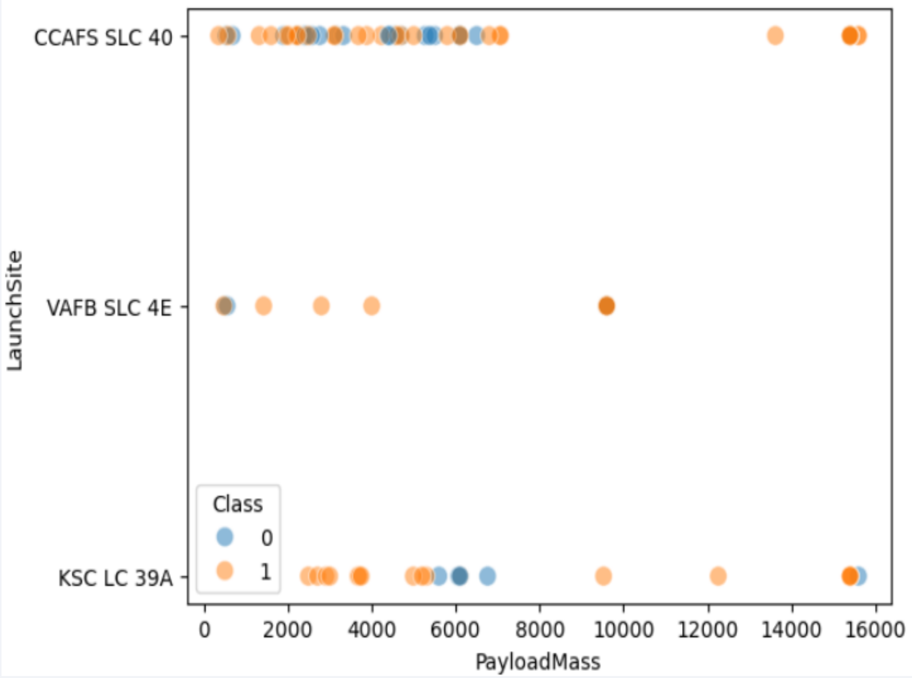
EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.
- The link to the notebook is Applied-Data-Science-Capstone/Assignment Exploring and Preparing Data.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)

TASK 2: Visualize the relationship between Payload Mass and Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the cl
sns.scatterplot(x="PayloadMass", y="LaunchSite", data=df, alpha=0.5, hue="Class", s=80)
plt.xlabel("PayloadMass",fontsize=10)
plt.ylabel("LaunchSite",fontsize=10)
plt.show()
```



EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - Displaying 5 records where launch sites begin with the string ‘CCA’
  - And many more….
- The link to the notebook is Applied-Data-Science-Capstone/Assignment SQL Notebook for Peer Assignment.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)

Task 1

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

```
%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

Task 2

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

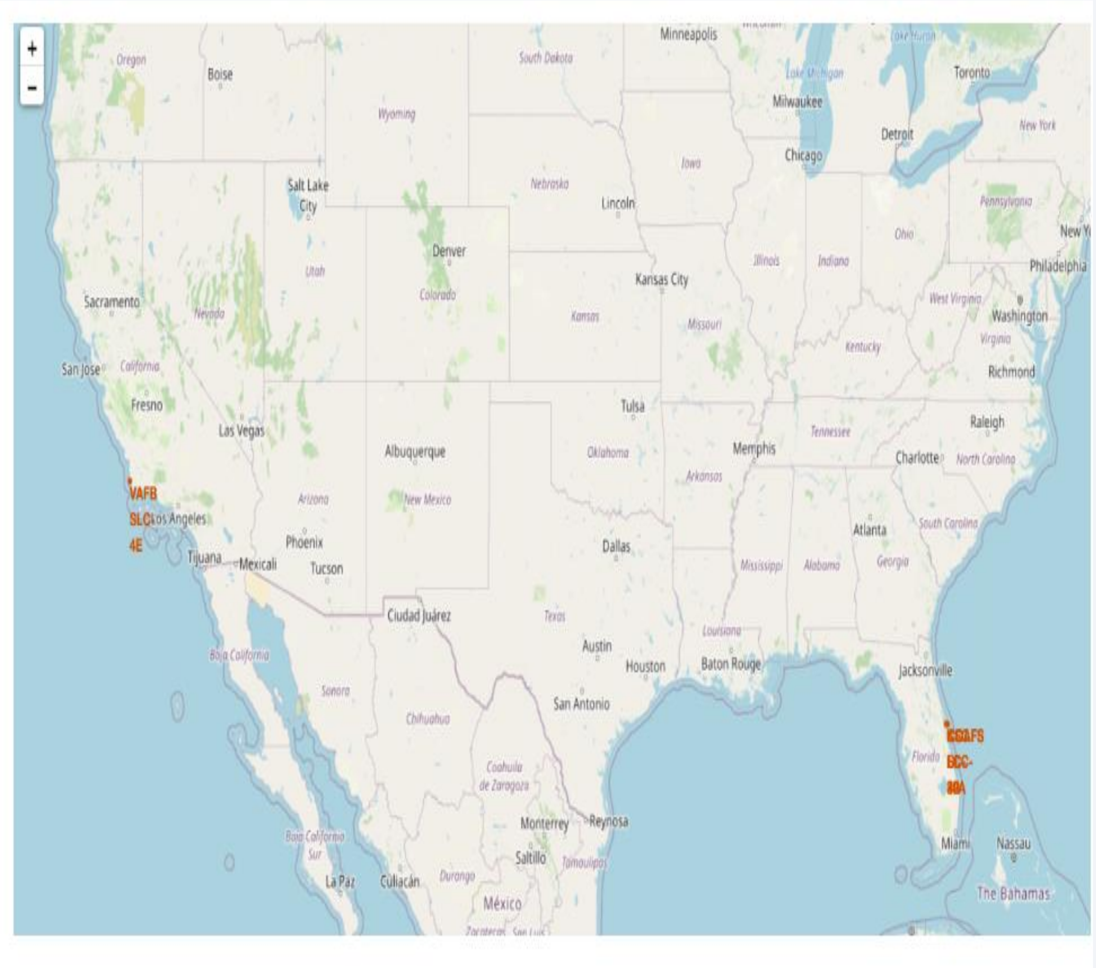
```
%sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)

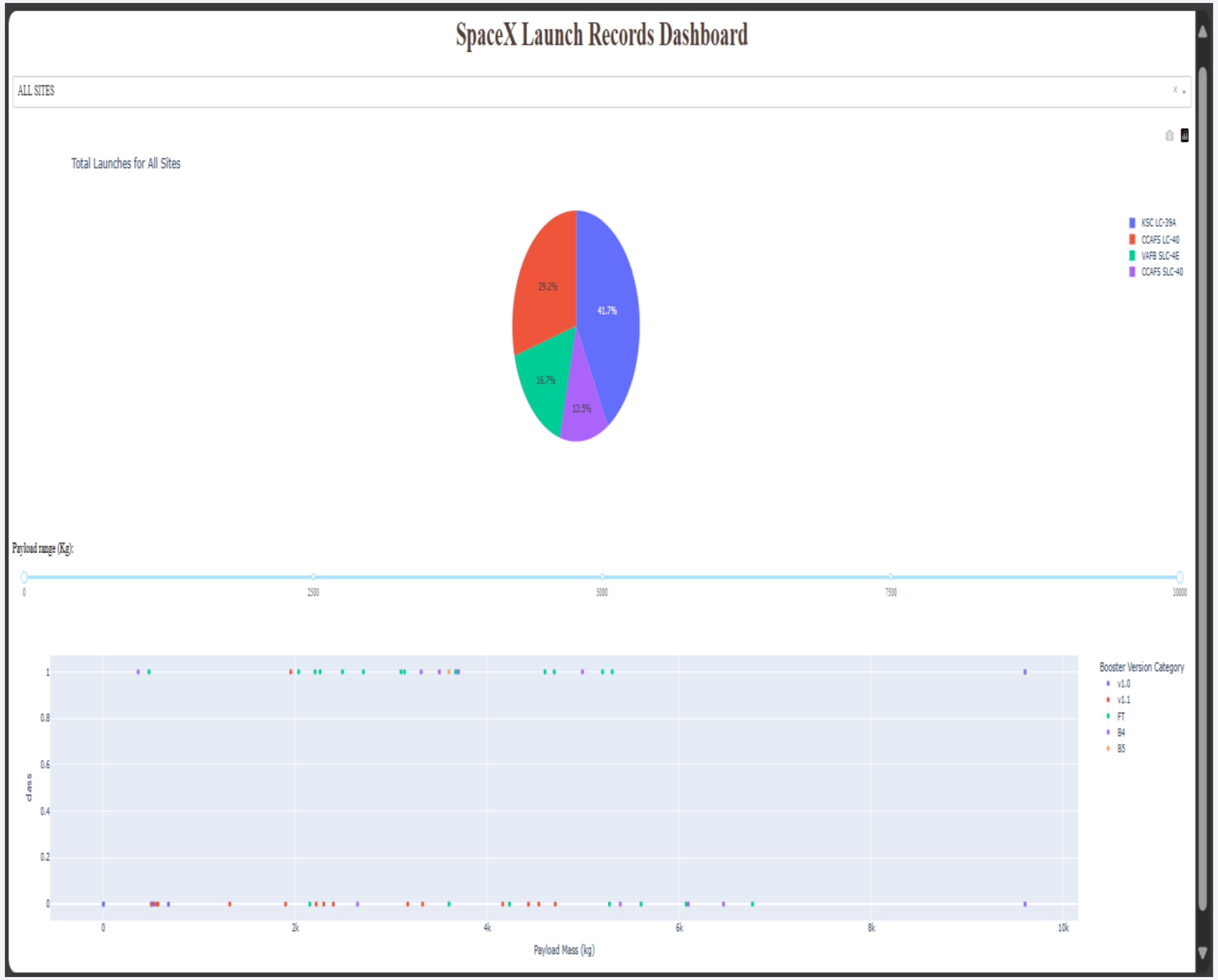
Build an Interactive Map with Folium

- We marked all launch sites and added map elements such as markers, circles, and lines to indicate the success or failure of launches at each site on a Folium map.
- We assigned launch outcomes as class 0 for failure and class 1 for success. By using color-coded marker clusters, we were able to identify launch sites with relatively high success rates.
- The link to the notebook is [Applied-Data-Science-Capstone/Launch Sites Location Analysis with Folium.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)]



Build a Dashboard with Plotly Dash

- We created an interactive dashboard using Plotly Dash.
- We displayed pie charts to show the total number of launches by different sites and plotted a scatter graph to illustrate the relationship between launch outcomes and payload mass (in kilograms) for various booster versions.
- The link to the python file is [Applied-Data-Science-Capstone/Build a Dashboard Application with Plotly Dash.py at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)]



Predictive Analysis (Classification)

- We loaded the data using NumPy and pandas, transformed it, and divided it into training and testing sets. We built various machine learning models and fine-tuned hyperparameters using GridSearchCV.
- Accuracy was used as the evaluation metric, and we enhanced the model through feature engineering and algorithm tuning. We identified the best-performing classification model.
- The link to the notebook is [Applied-Data-Science-Capstone/Assignment Machine Learning Prediction.ipynb at master · aasimshaikh98/Applied-Data-Science-Capstone (github.com)]()

TASK 1

Create a NumPy array from the column `Class` in `data`, by applying the method `to_numpy()` then assign it to the variable `Y`,make sure the output is a Pandas series (only one bracket df['name of column']).

```
Y = data['Class'].to_numpy()
Y
```

```
array([0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1,
       1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 0, 0, 0, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,
       1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1], dtype=int64)
```

TASK 2

Standardize the data in `X` then reassign it to the variable `X` using the transform provided below.

```
# students get this
X= preprocessing.StandardScaler().fit(X).transform(X)
X[0:5]
```

Results

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

TASK 12

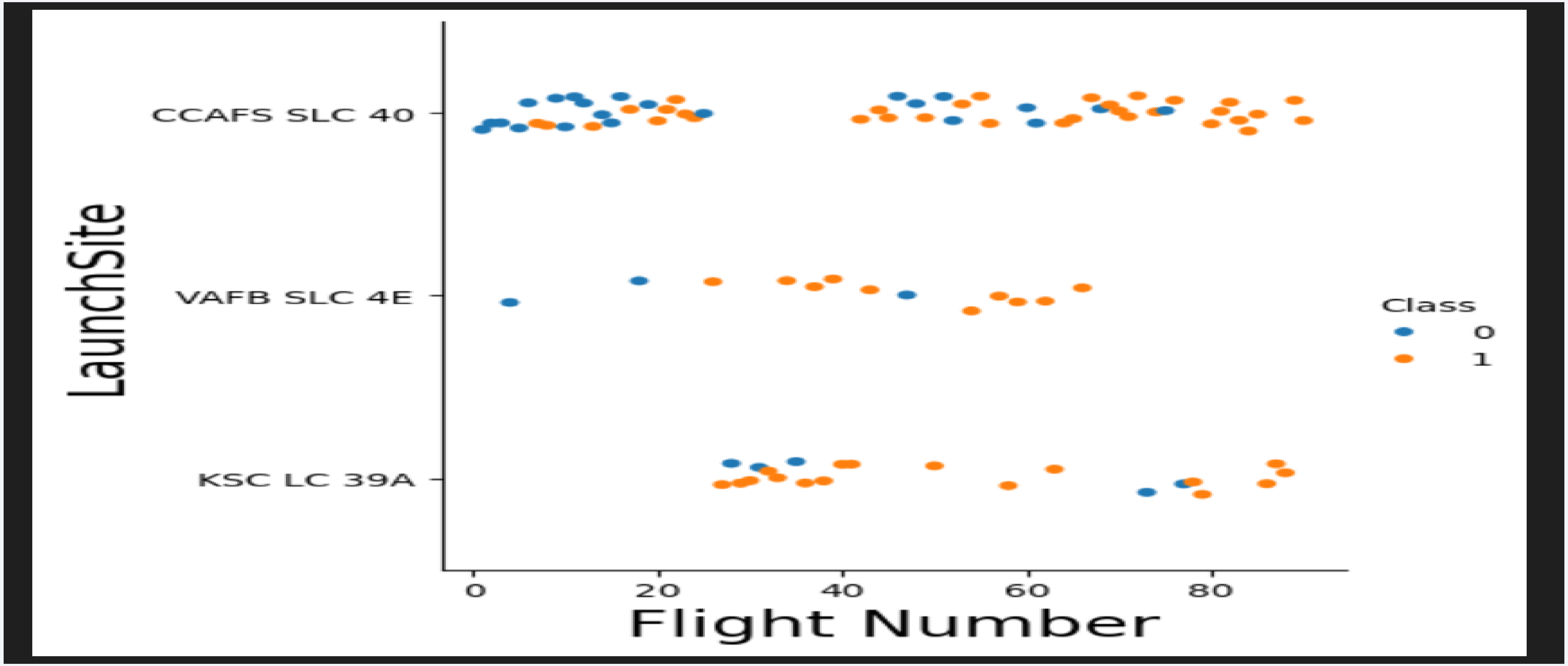
Find the method performs best:

```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision Tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K Nearest Neighbors method:', knn_cv.score(X_test, Y_test))
```

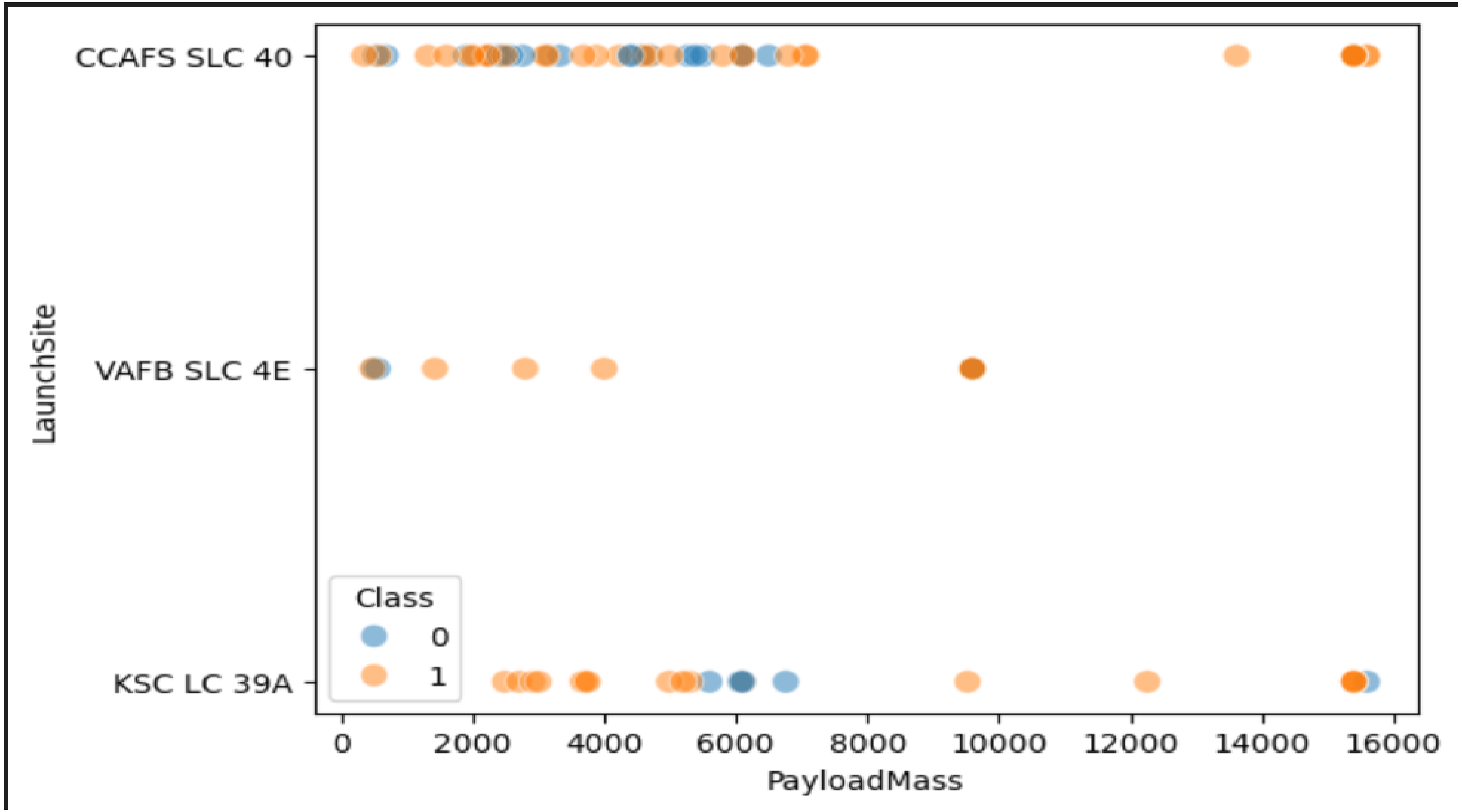
Section 2

Insights drawn from EDA

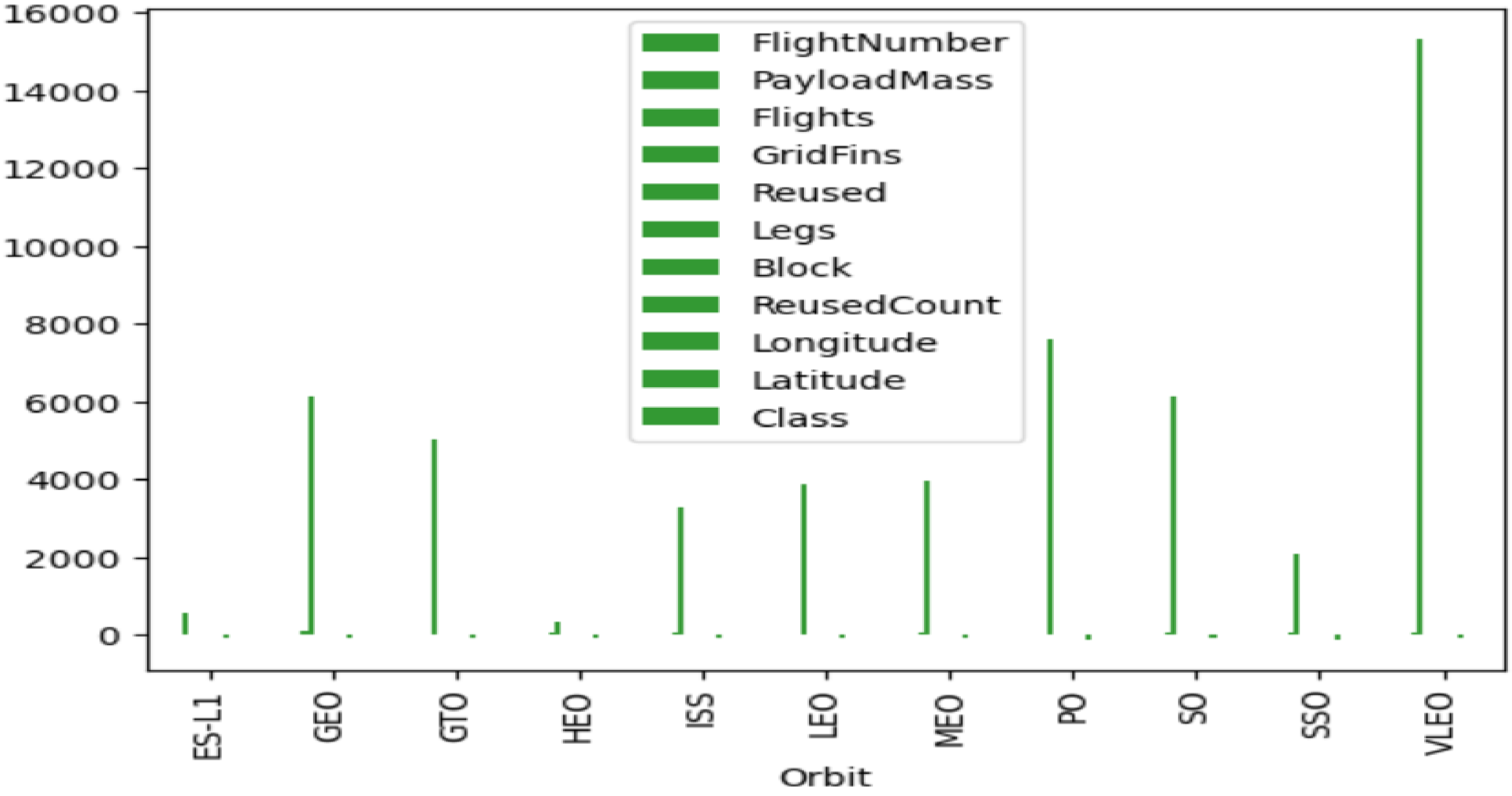
Flight Number vs. Launch Site



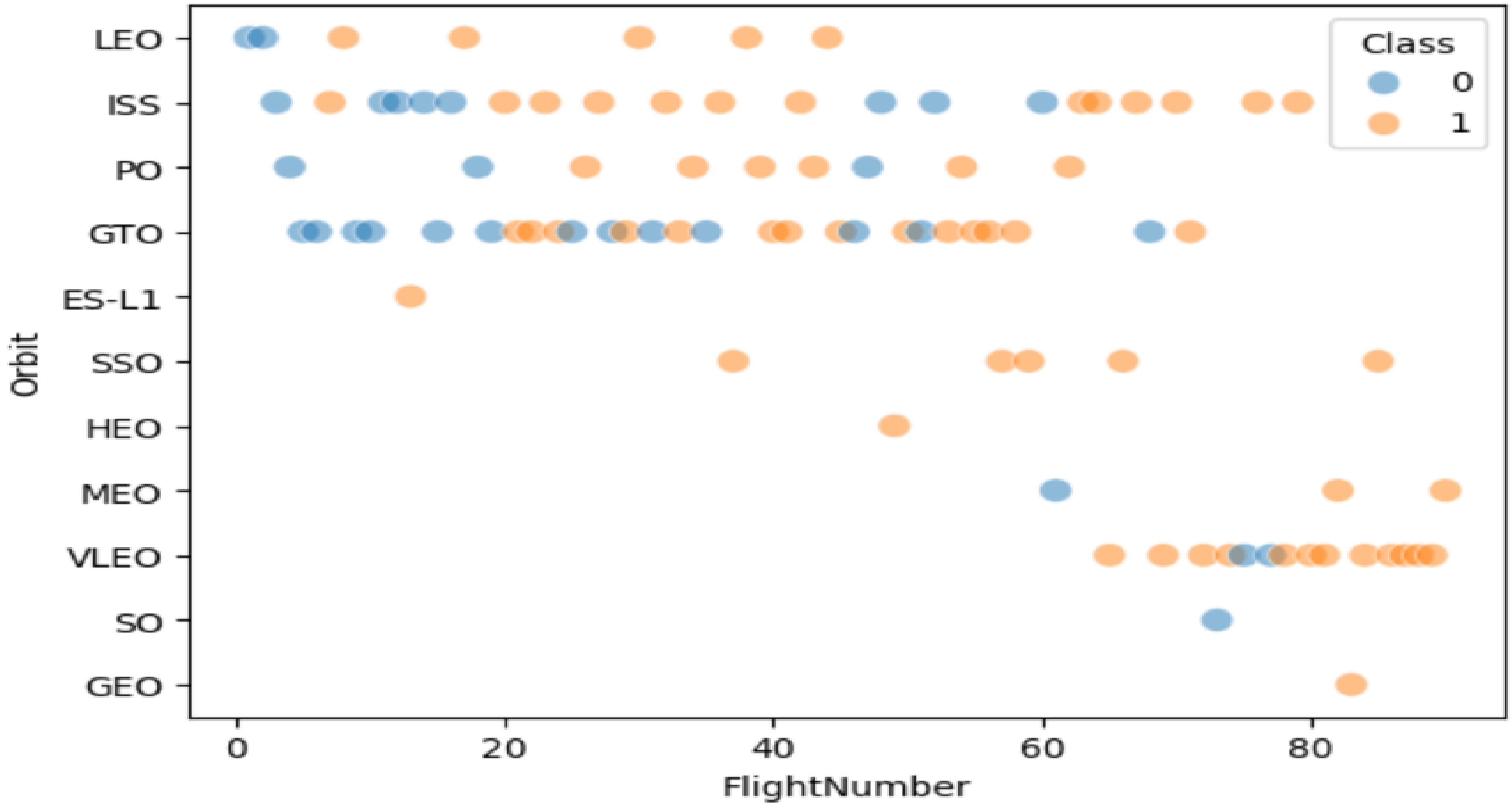
Payload vs. Launch Site



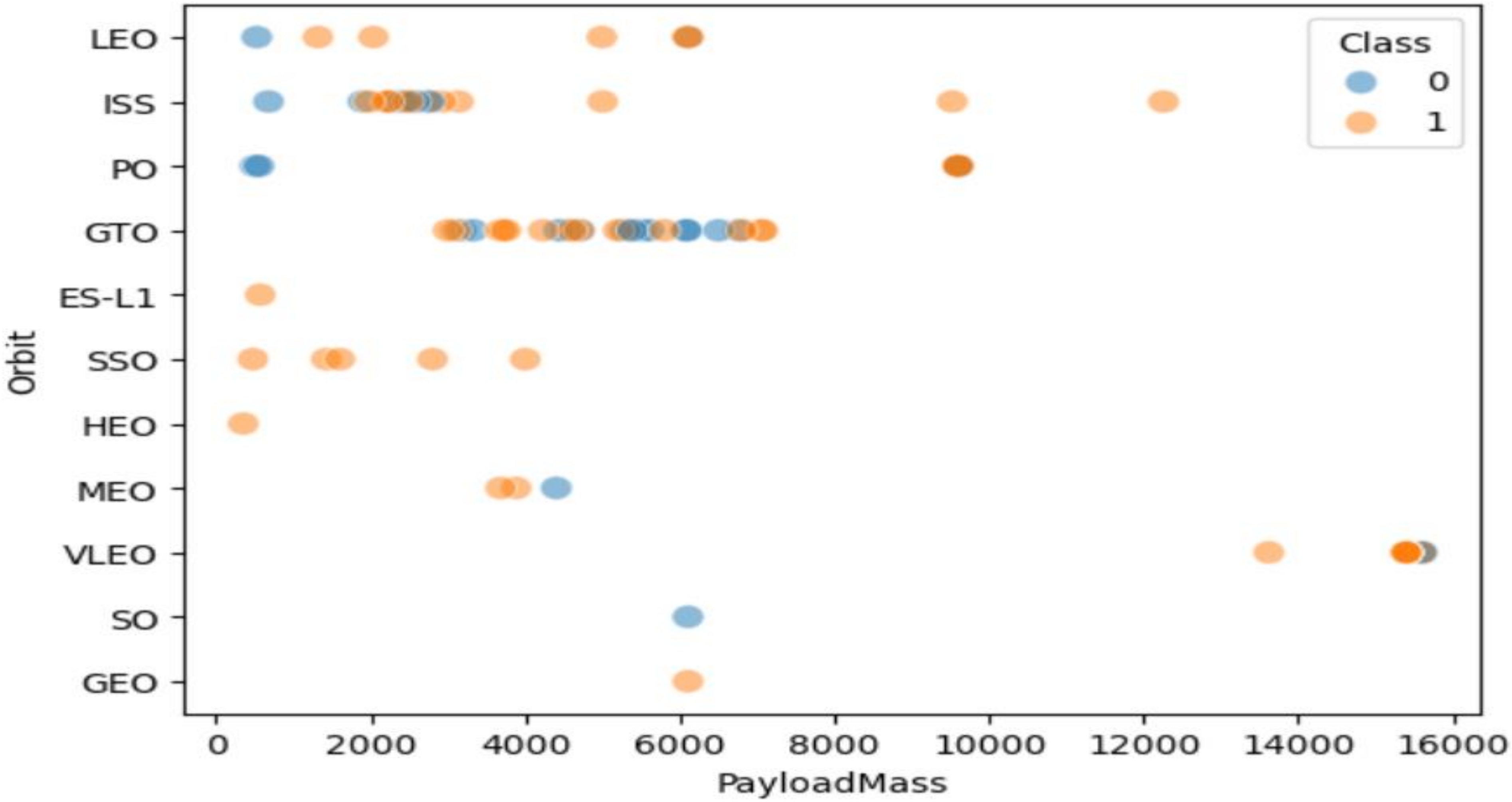
Success Rate vs. Orbit Type



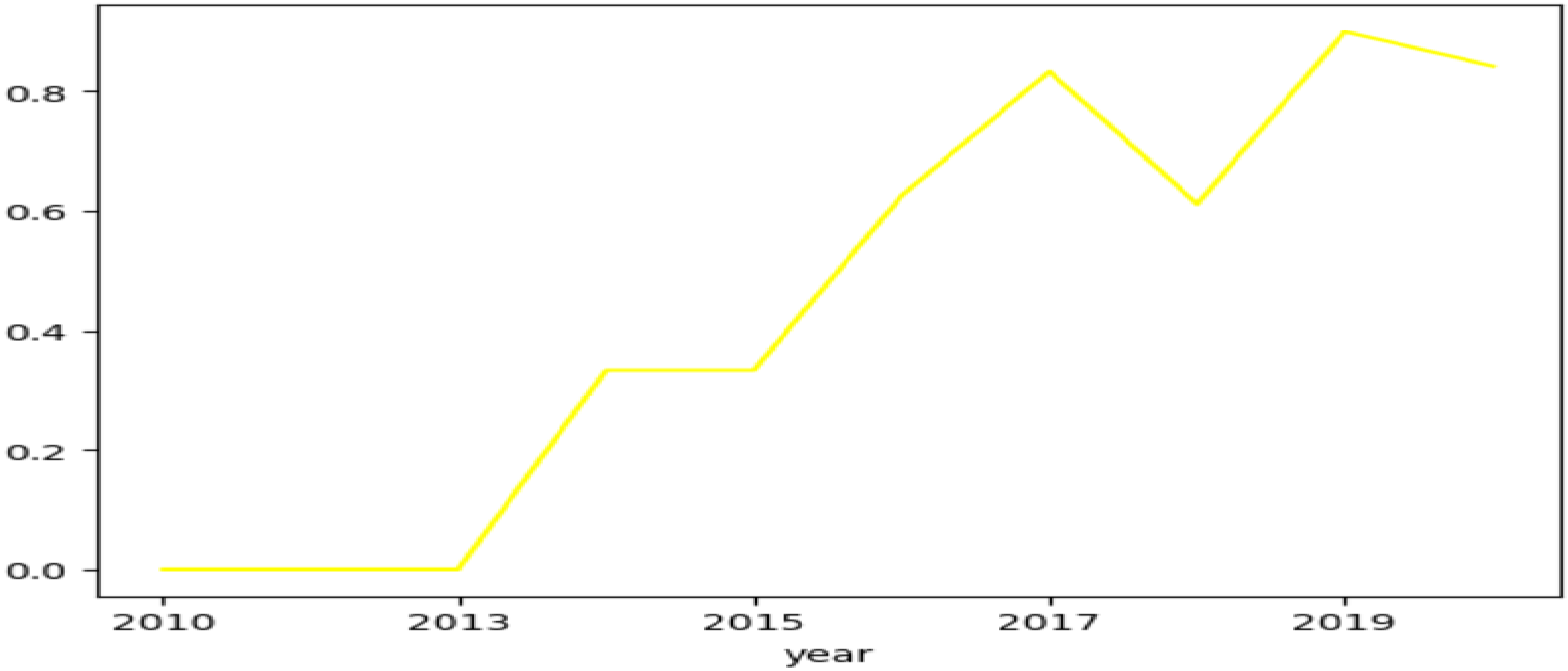
Flight Number vs. Orbit Type



Payload vs. Orbit Type



Launch Success Yearly Trend



All Launch Site Names

```
%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

Launch Site Names Begin with 'CCA'

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

Python

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

```
%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

```
%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 mission_outcome = 'Success';
```

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

MIN(DATE)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT booster_version FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (drone ship)' and payload_mass__kg_ BETWEEN 4000 and 6000;
```

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

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT COUNT(mission_outcome) FROM SPACEXTBL ;
```

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

COUNT(mission_outcome)
101

Boosters Carried Maximum Payload

```
%sql SELECT booster_version FROM SPACEXTBL ORDER BY payload_mass__kg_ LIMIT 10
```

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

Booster_Version
F9 v1.0 B0003
F9 v1.0 B0004
F9 B4 B1045.1
F9 FT B1038.1
F9 v1.0 B0006
F9 v1.1 B1003
F9 v1.0 B0005
F9 v1.1 B1017
F9 v1.1 B1013
F9 v1.0 B0007

2015 Launch Records

```
%sql SELECT booster_version, launch_site FROM (SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Failure%' and DATE LIKE '2015%');
```

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

Booster_Version	Launch_Site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

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

```
%sql SELECT * FROM SPACEXTBL WHERE "Landing_Outcome" = 'Failure (drone ship)' and DATE BETWEEN '2010-06-04' AND '2017-03-20' ORDER BY "Landing_Outc
```

Python

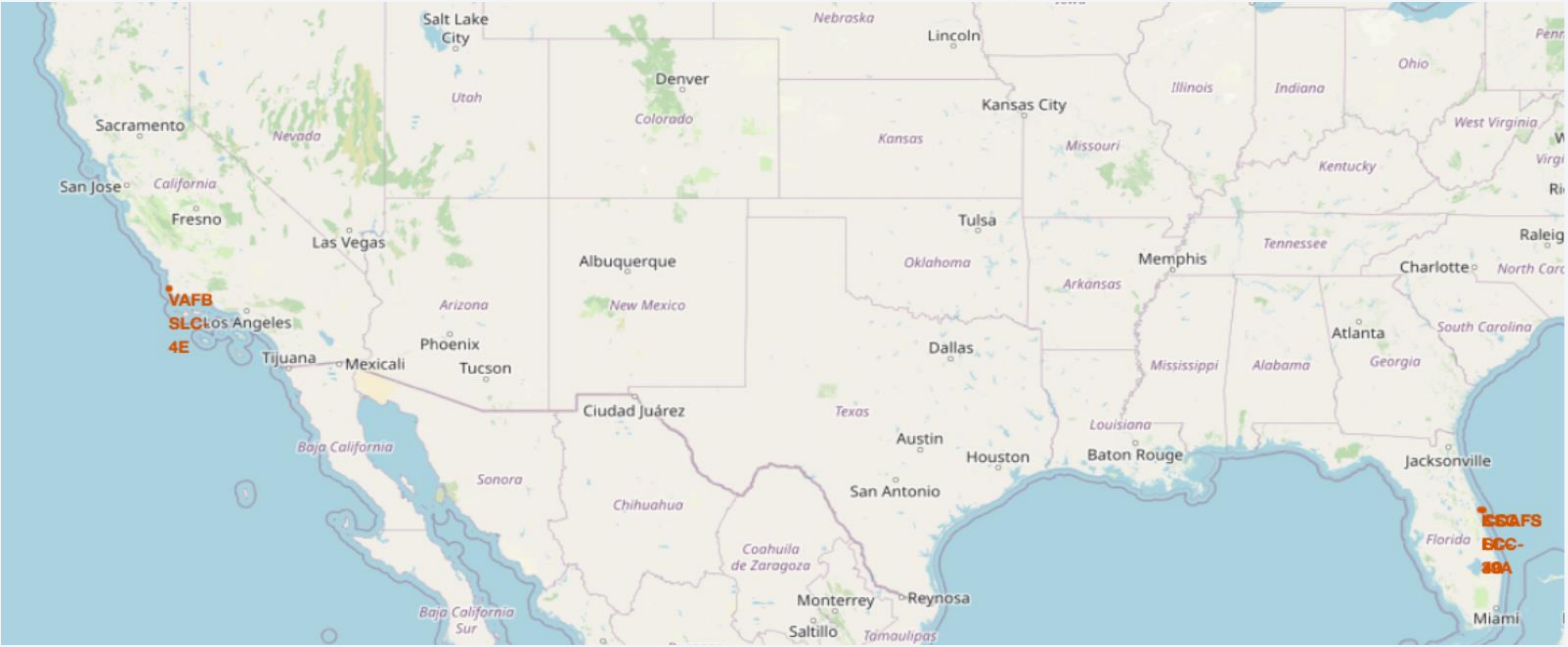
```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2015-01-10	9:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2016-01-17	18:42:00	F9 v1.1 B1017	VAFB SLC-4E	Jason-3	553	LEO	NASA (LSP) NOAA CNES	Success	Failure (drone ship)
2016-03-04	23:35:00	F9 FT B1020	CCAFS LC-40	SES-9	5271	GTO	SES	Success	Failure (drone ship)
2016-06-15	14:29:00	F9 FT B1024	CCAFS LC-40	ABS-2A Eutelsat 117 West B	3600	GTO	ABS Eutelsat	Success	Failure (drone ship)

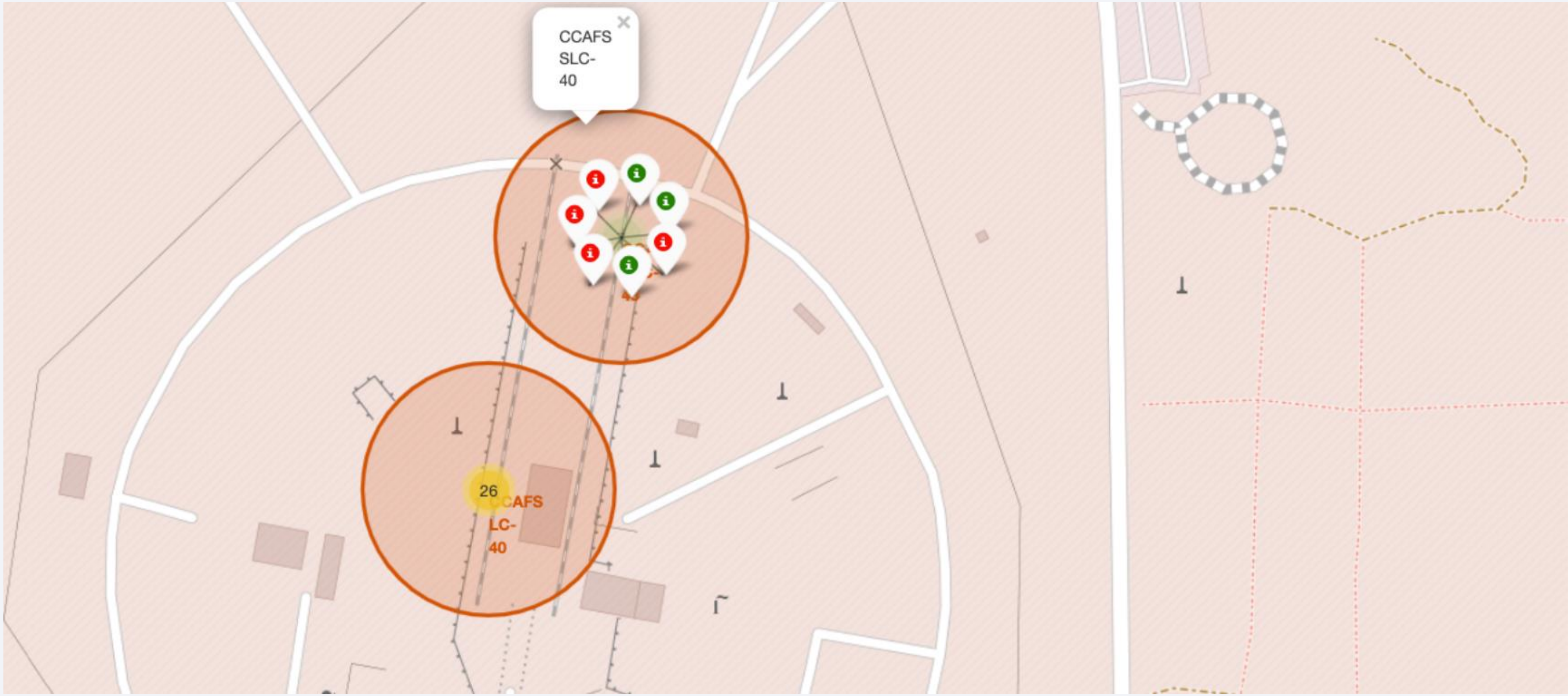
Section 4

Launch Sites Proximities Analysis

All launch sites global map markers



Markers showing launch sites with color labels

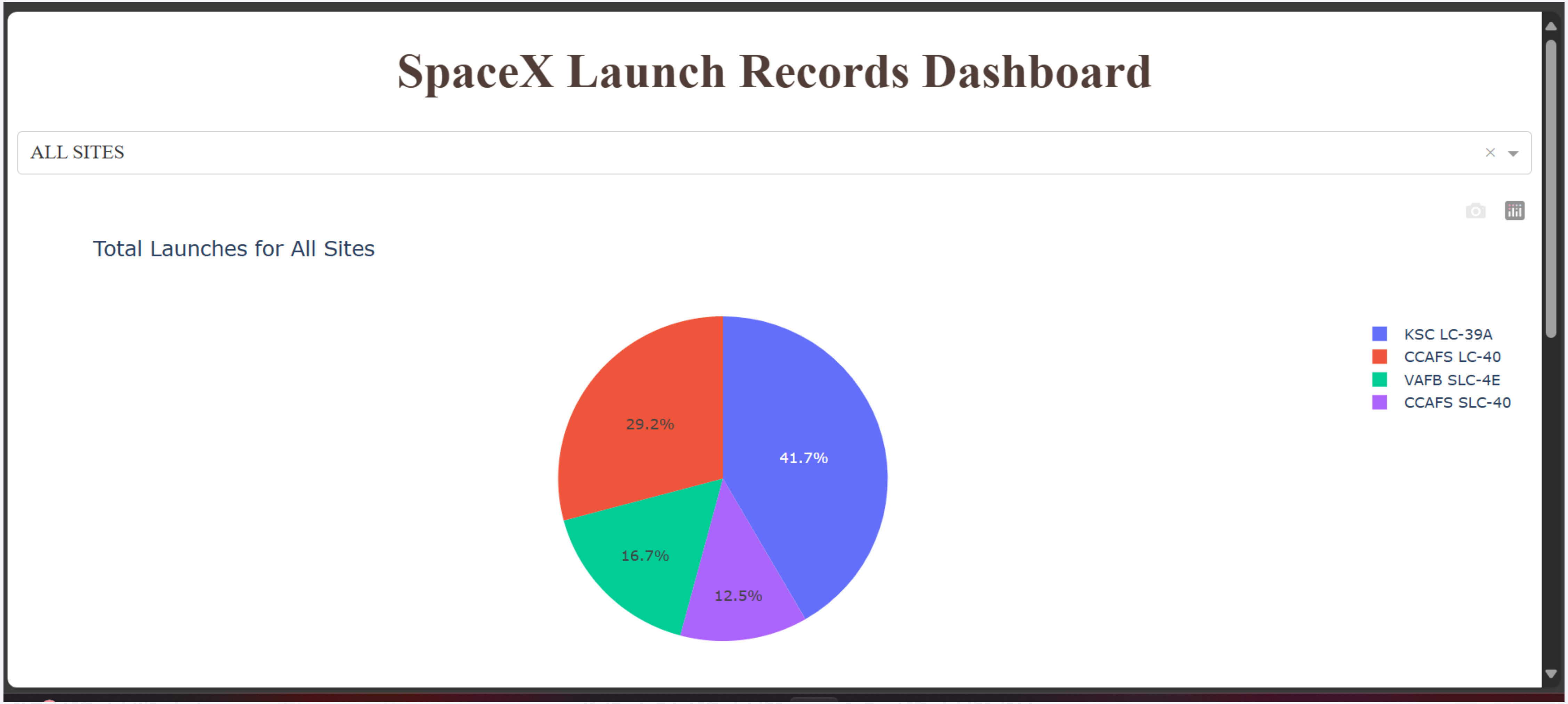


Launch Site distance to landmarks

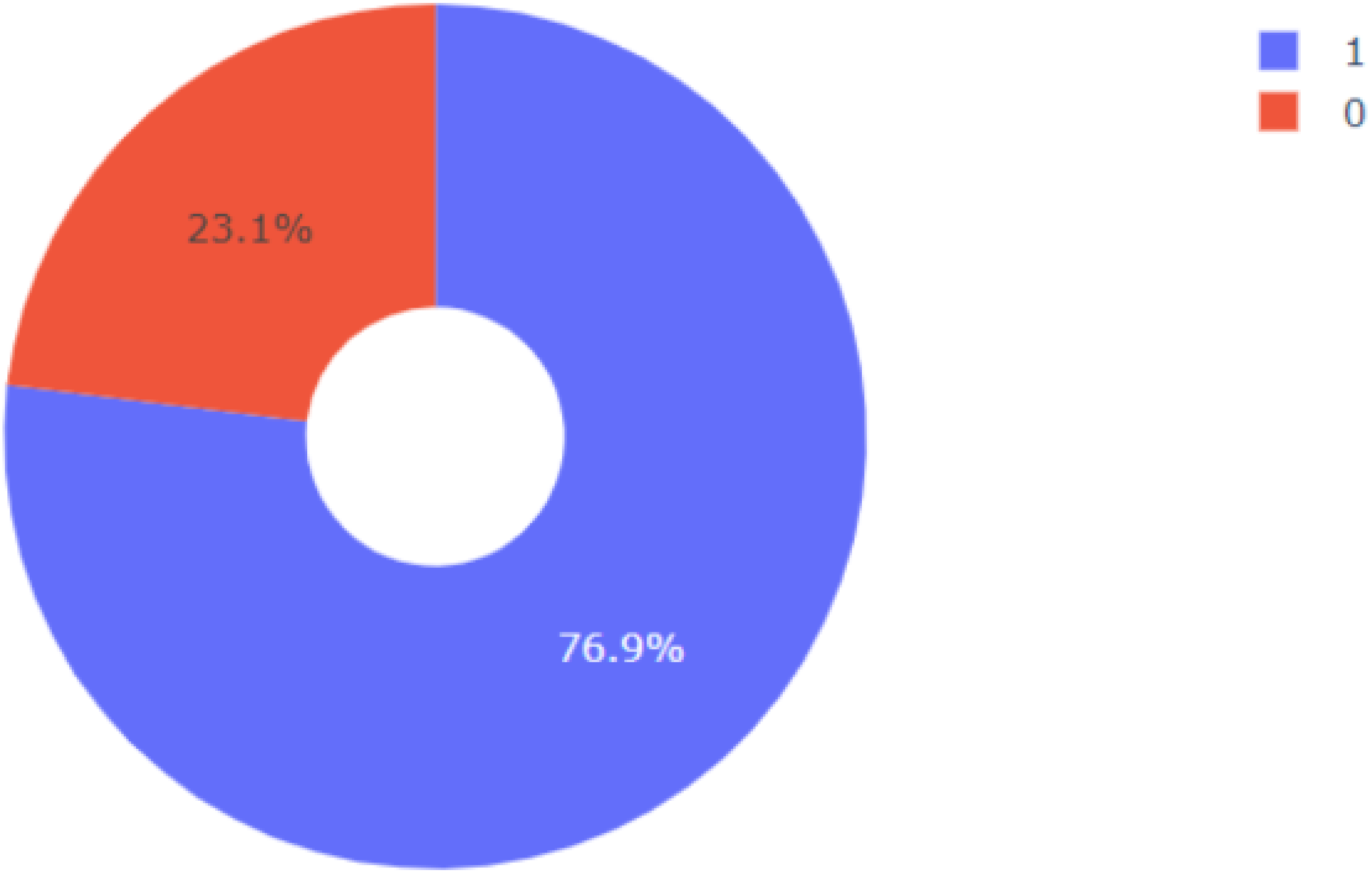
Section 5

Build a Dashboard with Plotly Dash

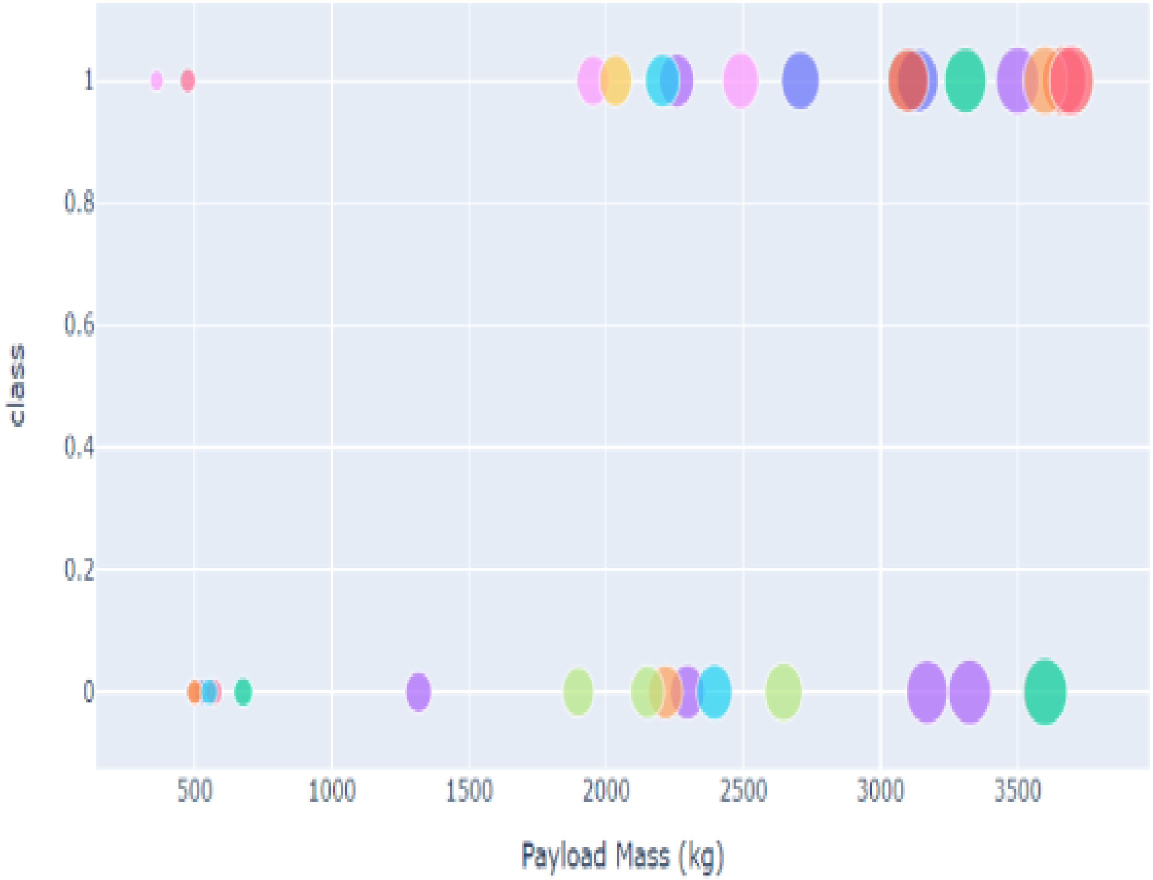
Pie chart showing the success percentage achieved by each launch site

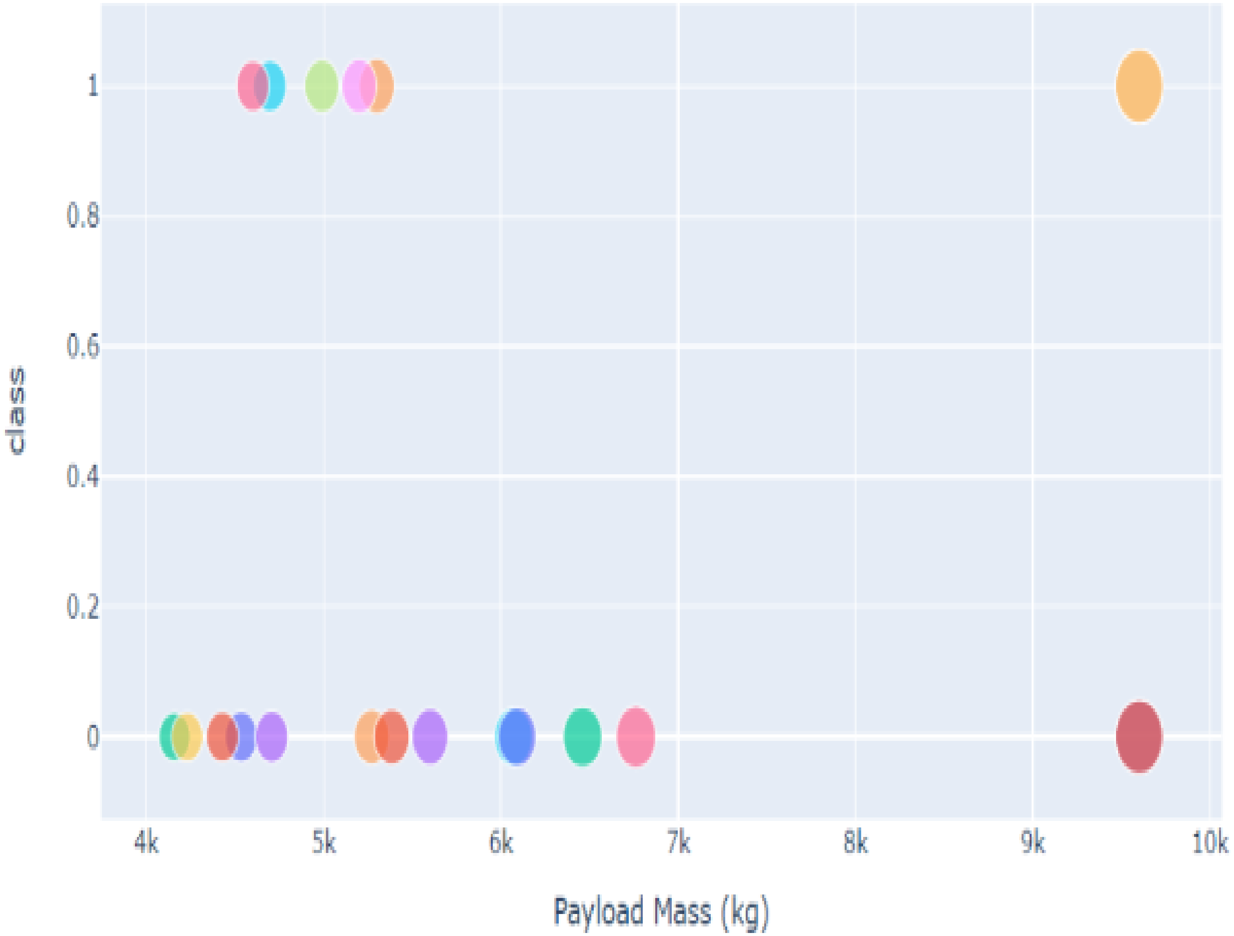


Pie chart showing the Launch site with the highest launch success ratio



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





Section 6

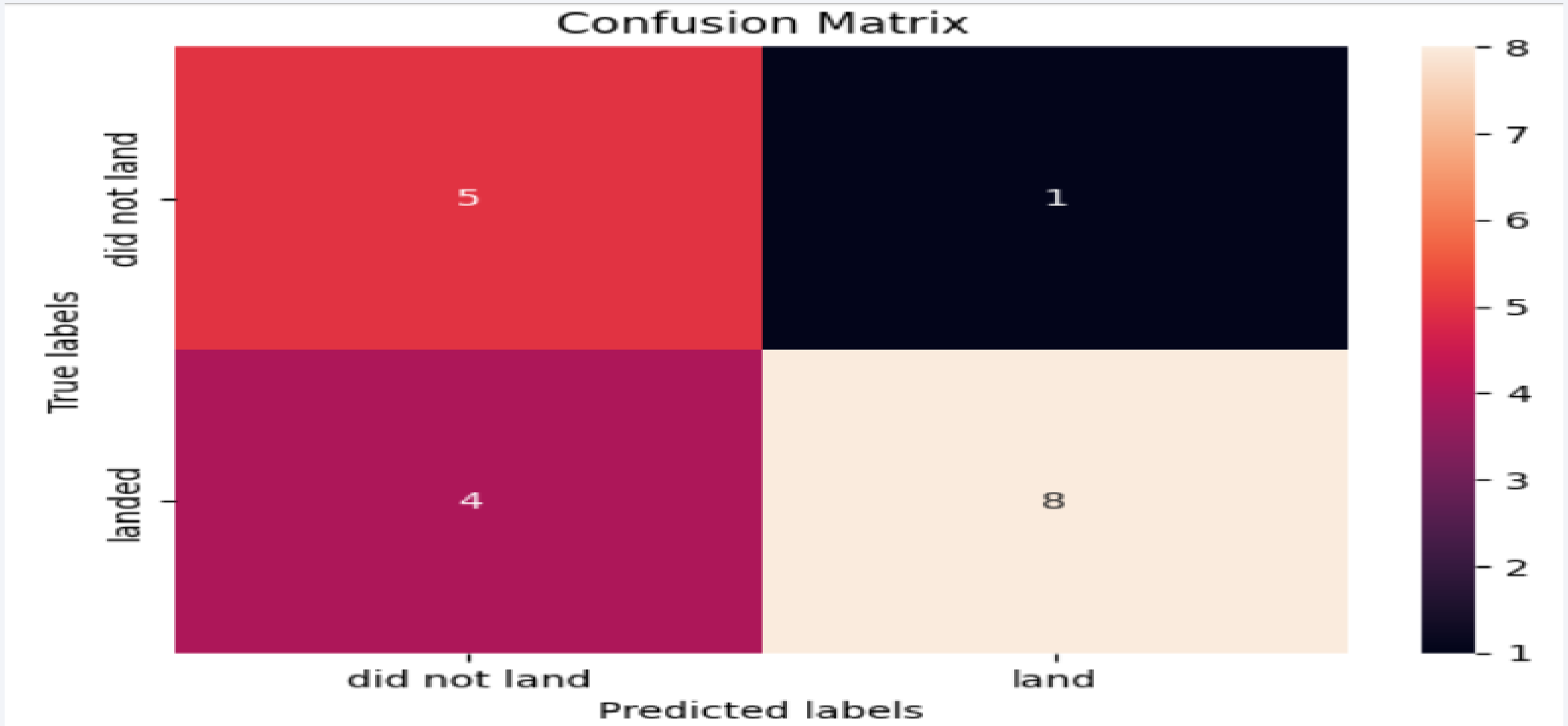
Predictive Analysis (Classification)

Classification Accuracy

```
[33]: print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
      print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
      print('Accuracy for Decision Tree method:', tree_cv.score(X_test, Y_test))
      print('Accuracy for K Nearest Neighbors method:', knn_cv.score(X_test, Y_test))

      Accuracy for Logistics Regression method: 0.8333333333333334
      Accuracy for Support Vector Machine method: 0.8333333333333334
      Accuracy for Decision Tree method: 0.7222222222222222
      Accuracy for K Nearest Neighbors method: 0.8333333333333334
```

Confusion Matrix



- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

Conclusions

We can conclude that:

- A higher number of flights at a launch site correlates with a greater success rate at that site.
- The launch success rate began to rise from 2013 and continued to improve through 2020.
- Orbits such as ES-L1, GEO, HEO, SSO, and VLEO showed the highest success rates.
- KSC LC-39A recorded the most successful launches among all sites.
- The Decision Tree classifier emerged as the most effective machine learning algorithm for this task.

Thank you!