IBM Developer
SKILLS NETWORK

IBM Data Science Capstone Project

Winning Space Race with Data Science

Muhammad Shahid
2024-02-06

Outline

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

Executive Summary

- Summary of Methodologies:

This project follows these steps:

- Data Collection
- Data Wrangling
- Exploratory Data Analysis
- Interactive Visual Analytics
- Predictive Analysis (Classification)

- Summary of Results:

This project produced the following outputs and visualizations:

1. Exploratory Data Analysis (EDA) results
2. Geospatial analytics
3. Interactive dashboard
4. Predictive analysis of classification models

Introduction

Project background and context

- SpaceX launches Falcon 9 rockets at a cost of around $62m. This is considerably cheaper than other providers (which usually cost upwards of $165m), and much of the savings are because SpaceX can land, and then re-use the first stage of the rocket.

Problems you want to find answers

- If we can make predictions on whether the first stage will land, we can determine the cost of a launch, and use this information to assess whether or not an alternate company should bid and SpaceX for a rocket launch.
- This project will ultimately predict if the Space X Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

1. **Data Collection**
   - Making GET requests to the SpaceX REST API
   - Web Scraping
2. **Data Wrangling**
   - Using the `.fillna()` method to remove NaN values
   - Using the `.value_counts()` method to determine the following:
     - Number of launches on each site
     - Number and occurrence of each orbit
     - Number and occurrence of mission outcome per orbit type
   - Creating a landing outcome label that shows the following:
     - 0 when the booster did not land successfully
     - 1 when the booster did land successfully
3. **Exploratory Data Analysis**
   - Using SQL queries to manipulate and evaluate the SpaceX dataset
   - Using Pandas and Matplotlib to visualize relationships between variables, and determine patterns
4. **Interactive Visual Analytics**
   - Geospatial analytics using Folium
   - Creating an interactive dashboard using Plotly Dash
5. **Data Modelling and Evaluation**
   - Using Scikit-Learn to:
     - Pre-process (standardize) the data
     - Split the data into training and testing data using `train_test_split`
     - Train different classification models
     - Find hyperparameters using `GridSearchCV`
   - Plotting confusion matrices for each classification model
   - Assessing the accuracy of each classification model

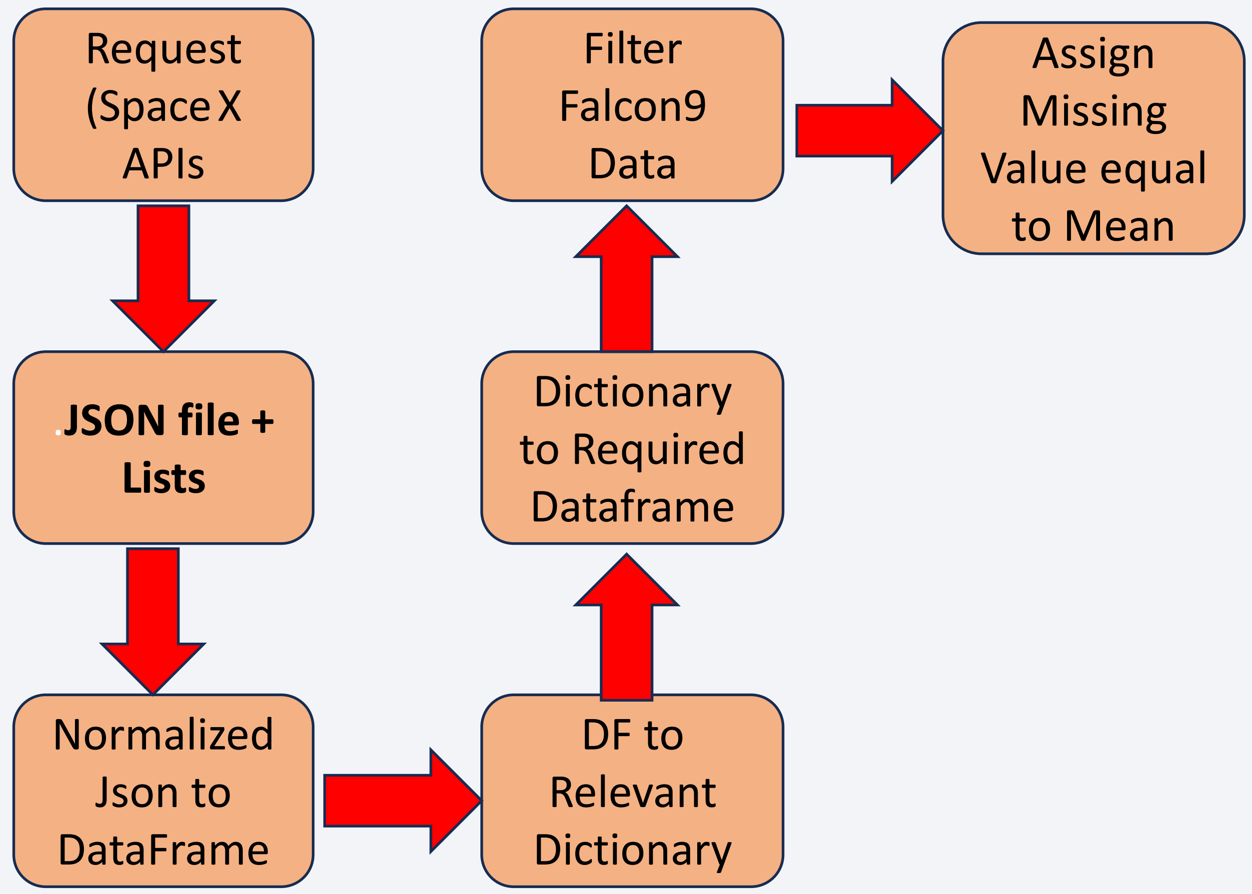
Data Collection

- The data collection process comprised of a hybrid approach involving both API requests from Space X's public API and web scraping of a table within Space X's Wikipedia entry.
- <u>Space X API Data Columns:</u>
- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins,
- Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude
- <u>Wikipedia Webscrape Data Columns:</u>
- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time

Data Collection – SpaceX API

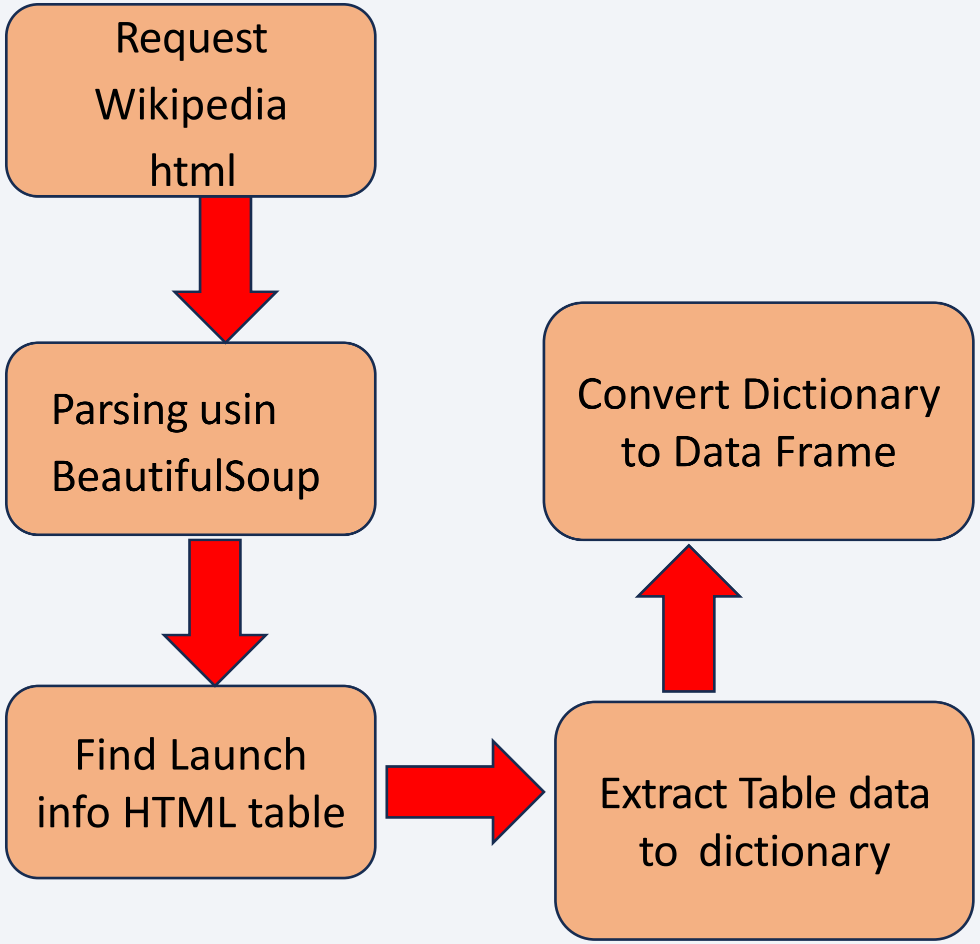
- Data collection with SpaceX REST calls can be shown using flowcharts

- Github Link:
- https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

- Web scraping process using flowcharts can be represented as:
- **Github Link:**
- https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_jupyter-labs-webscraping.ipynb



Data Wrangling

Data Wrangling

- To determine whether a booster will successfully land, it is best to have a binary column, i.e., where the value is 1 or 0, representing the success of the landing.
- This was done as follows:

1. Defining a set of unsuccessful (bad) outcomes,
2. Creating a list, landing_class, where the element is 0 if the corresponding row in Outcome is in the set bad_outcome, otherwise, it's 1.
3. Create a Class column that contains the values from the list landing_class
4. Export the DataFrame as a .csv file.

- **Github Link: https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_jupyter-spacex_Data_wrangling.ipynb**

EDA with Data Visualization

Exploratory Data Analysis performed on several variables such as Launch Site, Flight Number, Payload Mass, Orbit, Class and Year.

- Plots drawn:
  - Flight Number vs. Payload Mass,
  - Flight Number vs. Launch Site,
  - Payload Mass vs. Launch Site,
  - Orbit vs. Success Rate,
  - Flight Number vs. Orbit, Payload vs Orbit, and
  - Success Yearly Trend
- Scatter plots, line charts, and bar plots were used to compare relationships between variables
- GitHub url: https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- Following are the key points to represent EDA using SQL
  - Loading data set into IBM DB2 Database.
  - Establishing connection using SQL Python integration.
  - Querying the dataset.
  - Queries were made to get information about launch site names, mission outcomes, various pay load sizes of customers and booster versions, and landing outcomes

- GitHub url:
- https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium was used to mark the Launch Sites, successful and unsuccessful landings, and a proximity example to key locations: Railway, Highway, Coast, and City on the map.

Why Added: it allows us to understand where launch sites are located. We can also visualize successful landings relative to location.

- GitHub url:
- https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_launch_site_location.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

- **Dashboard Application was built to interactively analyze spaceX visually**.
- It includes a pie chart and a scatter plot.
- Pie chart can be selected to show distribution of successful landings across all launch sites and can be selected to show individual launch site success rates.
- Scatter plot takes two inputs: All sites or individual site and payload mass on a slider between 0 and 10000 kg.
- **GitHub Link:** https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

The following steps were taking to develop, evaluate, and find the best performing classification model:

Model Development

- To prepare the dataset for model development:
  - Load dataset
  - Perform necessary data transformations (standardise and pre-process)
  - Split data into training and test data sets, using `train_test_split()`
  - Decide which type of machine learning algorithms are most appropriate
- For each chosen algorithm:
  - Create a `GridSearchCV` object and a dictionary of parameters
  - Fit the object to the parameters
  - Use the training data set to train the model

Model Evaluation

- For each chosen algorithm:
  - Using the output GridSearchCV object:
    - Check the tuned hyperparameters (`best_params_`)
    - Check the accuracy (`score` and `best_score_`)
  - Plot and examine the Confusion Matrix

Finding the Best Classification Model

- Review the accuracy scores for all chosen algorithms
- The model with the highest accuracy score is determined as the best performing model

- **Github Link:** https://github.com/shahidmohana/IBMDataSciencePublicRepo/blob/main/c10_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

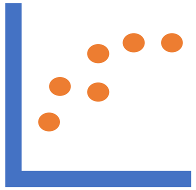
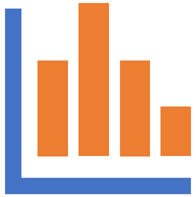
In Next section we will represent results about the following

- Exploratory data analysis
  - EDA using Visualization
  - EDA with SQL
- Interactive analytics
  - Folium
  - Dash App
- Predictive analysis
  - Different Classification Models

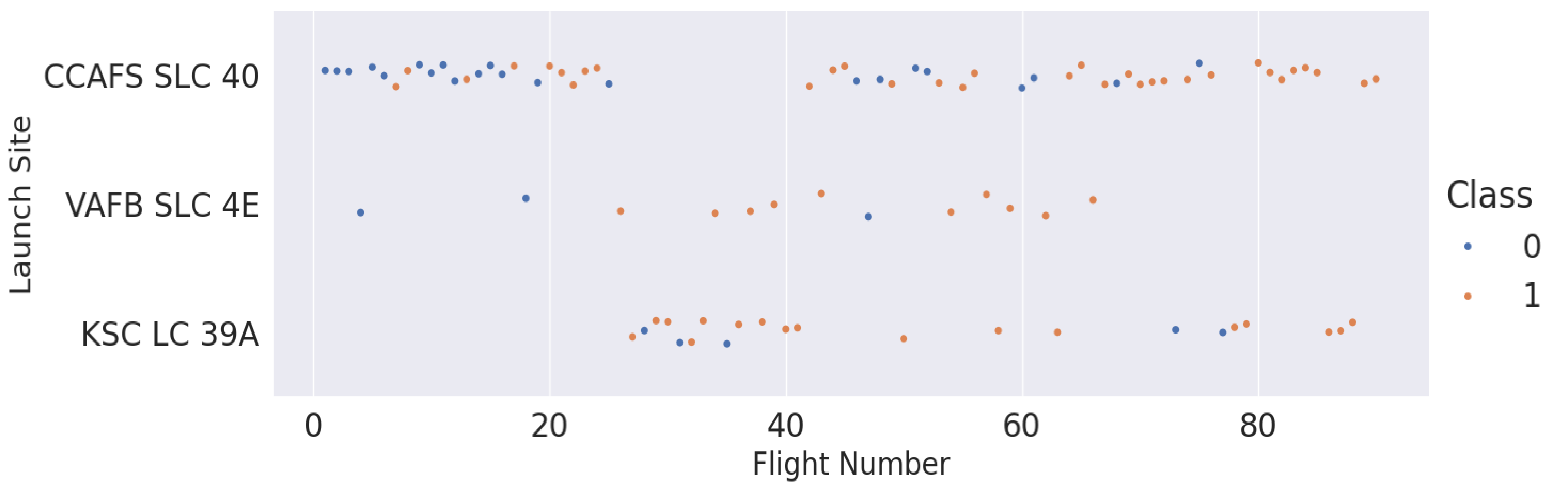
Section 2

Insights drawn from EDA

EDA with Visualization

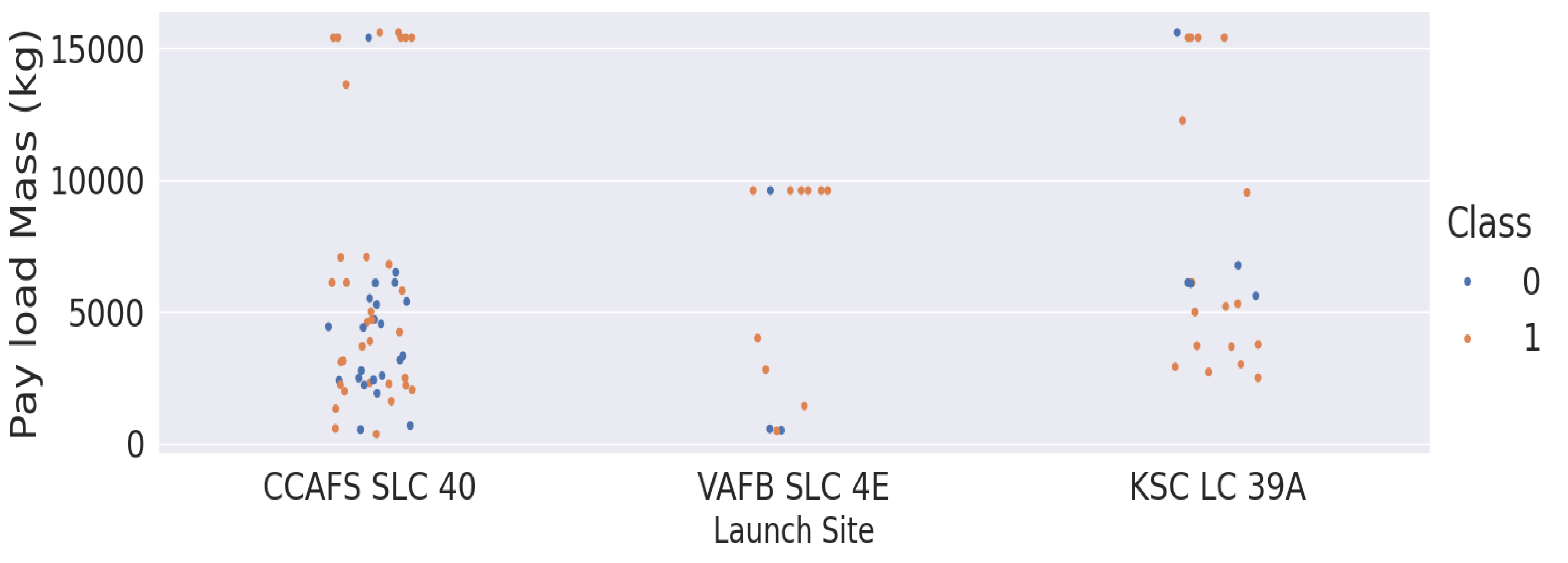

Flight Number vs. Launch Site



This is a scatter plot of Flight Number vs. Launch Site: plot suggests:

- An increase in success rate over time .
- CCAFS SLC 40 seems to be the main launch site as it has the most points whereas VAFB SLC 4E the relatively less flights launched.

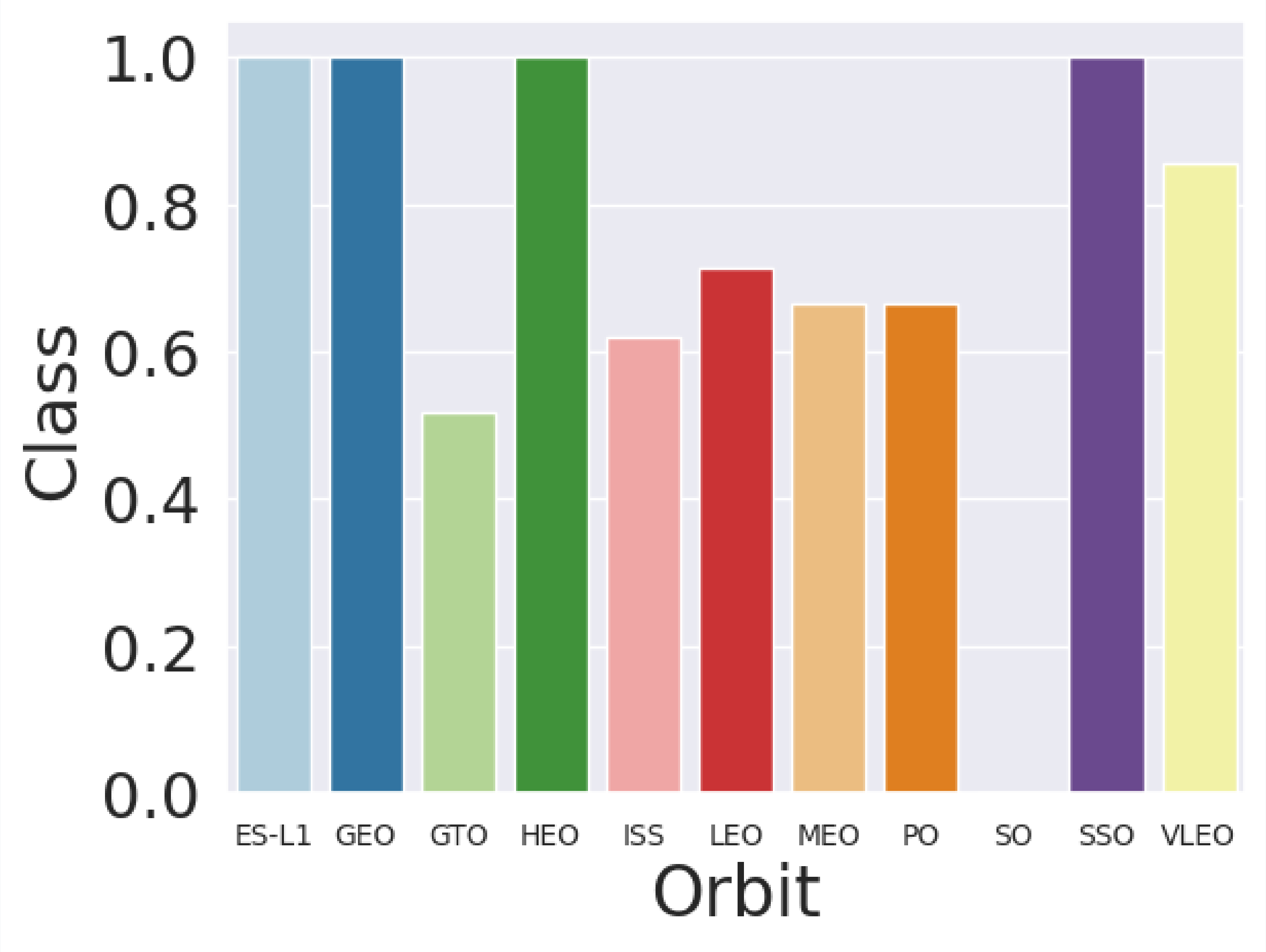
Payload vs. Launch Site



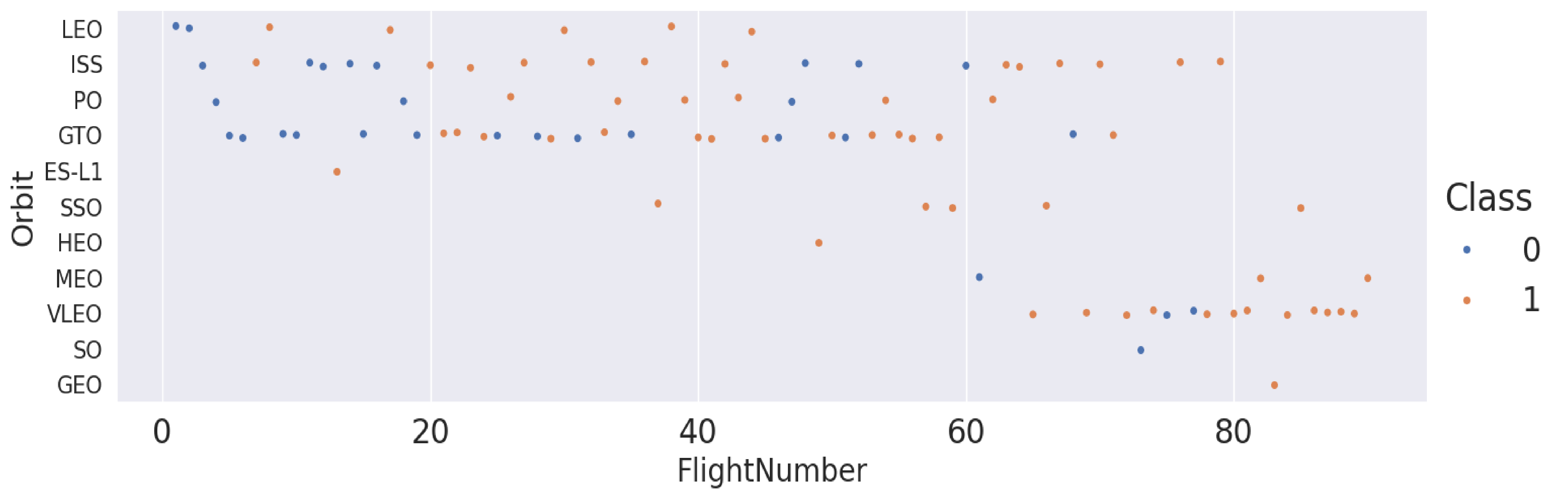
- Payload mass appears to fall mostly between 0-7500 kg in the case of CCAF SLC 40 and KSC LC 39A.
- VAFB SLC 4E has almost 90 % success rate for PLM=1000 kg
- CCAFS SLC 40 has relatively less success rate but most launches

Success Rate vs. Orbit Type

- Four orbits i.e. ES-L1, ,HEO, Geo and SSO has highest success rate
- VLEO has about 90% success rate
- Other all range from 50% to 70%

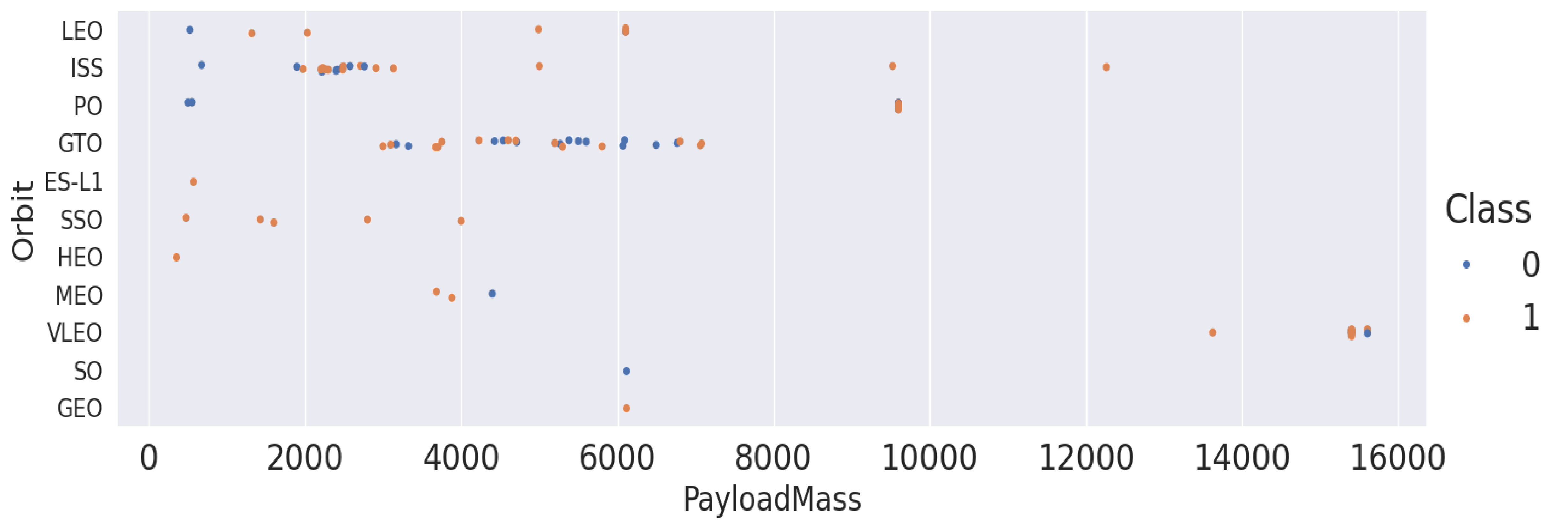


Flight Number vs. Orbit Type



- Up to flights numbers 80 lie in the orbits of LEO, ISS, PO and GTO
- Flight number 75 to 100, lie in orbits ranging from GEO to SSO, more in VLEO and SO

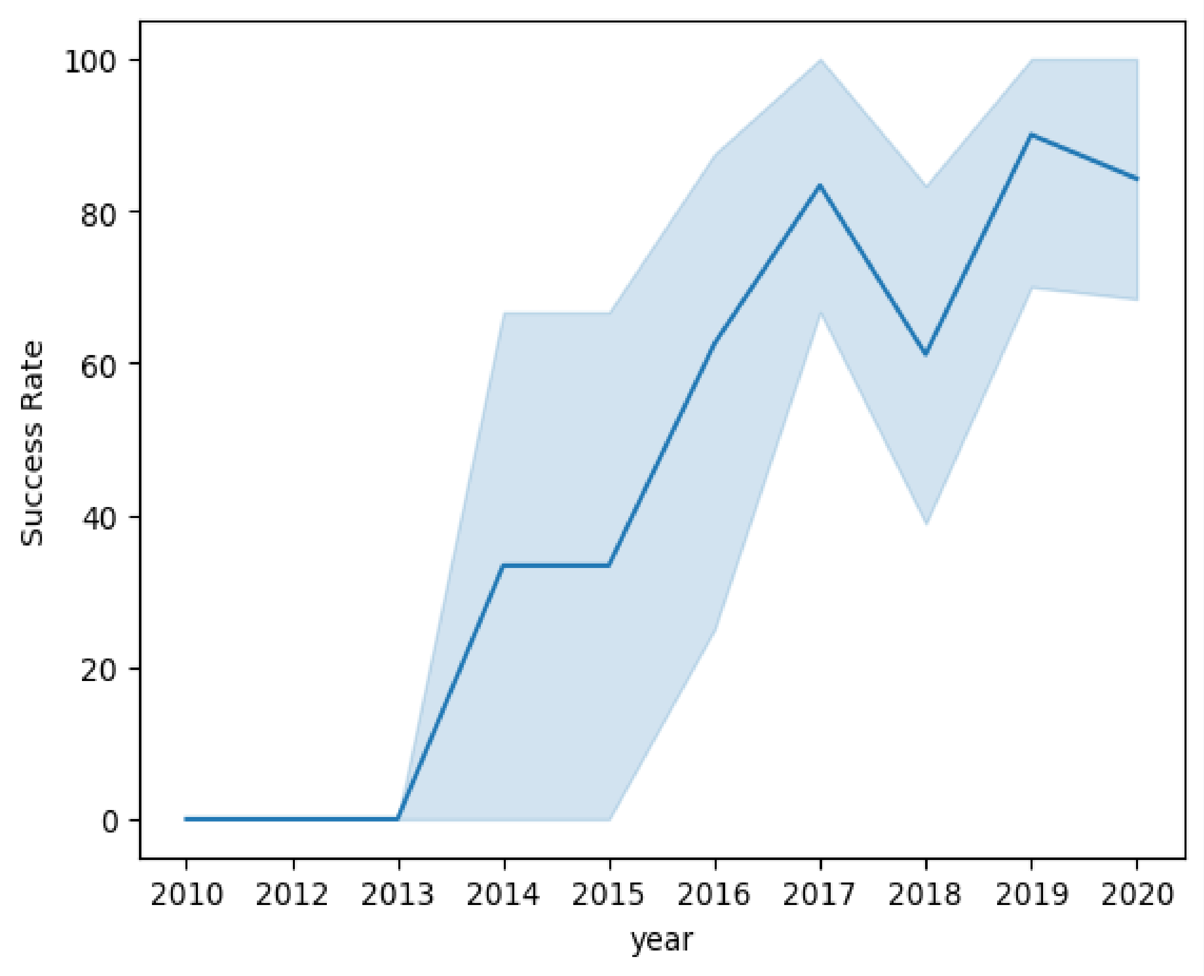
Payload vs. Orbit Type



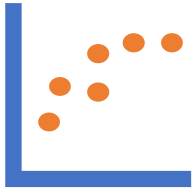
- Payload mass correlate with orbit
- LEO and SSO seem to have relatively low payload mass
- GTO has intermediate payload mass
- The other most successful orbit VLEO only has payload mass values in the higher end of the range

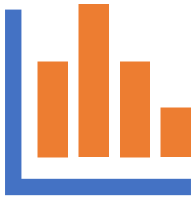
Launch Success Yearly Trend

- **2010-13**: Zero success rate
- **2023-15**: success rises up to 40%
- **2015-17**: success rate rises further up to 80%
- **2018**: falls to 60%
- **2019-20**: rises again up to around 90



EDA with SQL





All Launch Site Names

- Find the names of the unique launch sites

```
%sql select DISTINCT "LAUNCH_SITE" from SPACEXTABLE
```

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

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

- The word UNIQUE returns only unique values from the LAUNCH_SITE column of the SPACEXTBL table.

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
%sql select * from SPACEXTABLE 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)
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

- **LIMIT 5** fetches only 5 records, and the **LIKE** keyword is used with the wild card **'CCA%'** to retrieve string values beginning with 'CCA'.

Total Payload Mass

- Calculate the total payload carried by boosters from NASA

```
%sql select sum(payload_mass__kg_) as sum from SPACEXTABLE where customer like 'NASA (CRS)'
```

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

sum
45596

- Present your query result with a short explanation here

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1

```
%sql select avg(payload_mass__kg_) as Average from SPACEXTABLE where booster_version like 'F9 v1.1%'
```

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

Average
2534.6666666666665

- The SUM keyword is used to calculate the total of the LAUNCH column, and the SUM keyword (and the associated condition) filters the results to only boosters from NASA (CRS).

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

```
%sql select min(date) as Date from SPACEXTABLE where mission_outcome like 'Success'
```

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

Date
2010-06-04

- The MIN is used to calculate the minimum of the DATE column, i.e. the first date.

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 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

- The BETWEEN keyword allows for 4000 < x < 6000 values to be selected.

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
%sql SELECT mission_outcome, count(*) as Count FROM SPACEXTABLE GROUP by mission_outcome ORDER BY mission_outcome
```

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

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

- The COUNT keyword is used to calculate the total number of mission outcomes, and the GROUPBY keyword is also used to group these results by the type of mission outcome.

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

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

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

- A subquery `SELECTS` statement within the brackets finds the **maximum payload**, in `WHERE clause`.
- The `DISTINCT` keyword is then used to retrieve only distinct /unique booster versions

2015 Launch Records

- List the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%sql select substr(Date, 6, 2) AS Month, substr(Date, 0, 5) AS Year, landing_outcome, booster_version,\
launch_site from SPACEXTABLE where DATE like '2015%' AND landing_outcome like 'Failure (drone ship)'
```

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

Month	Year	Landing_Outcome	Booster_Version	Launch_Site
01	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

substr(Date, 6,2) used to get the months and substr(Date,0,5) to get year.

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

- 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

```
%sql select landing_outcome, count(*) as count from SPACEXTABLE where Date >= '2010-06-04' \
AND Date <= '2017-03-20' GROUP by landing_outcome ORDER BY count Desc
```

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

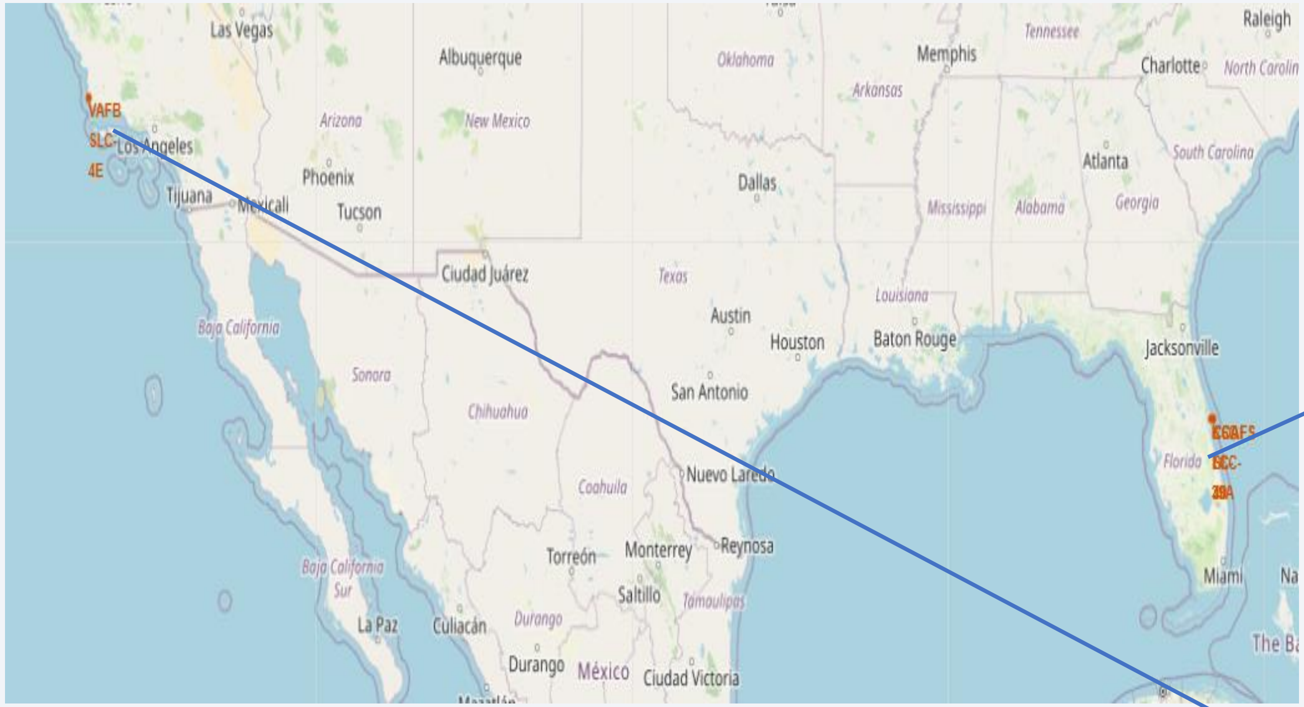
Landing_Outcome	count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

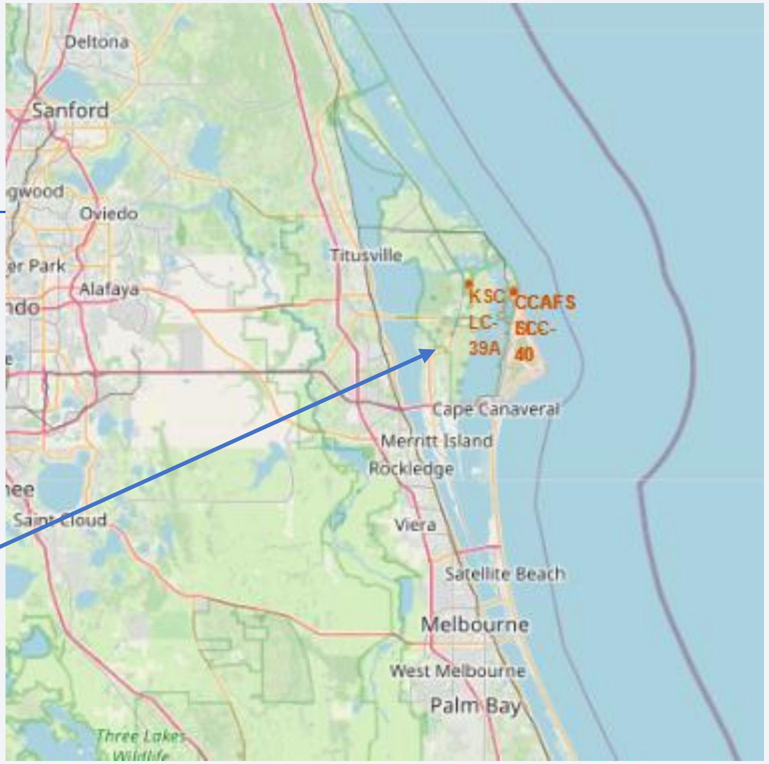
- To get grouped and ordered result, the keywords GROUP BY and ORDER BY, respectively used, where DESC is used to specify the descending order

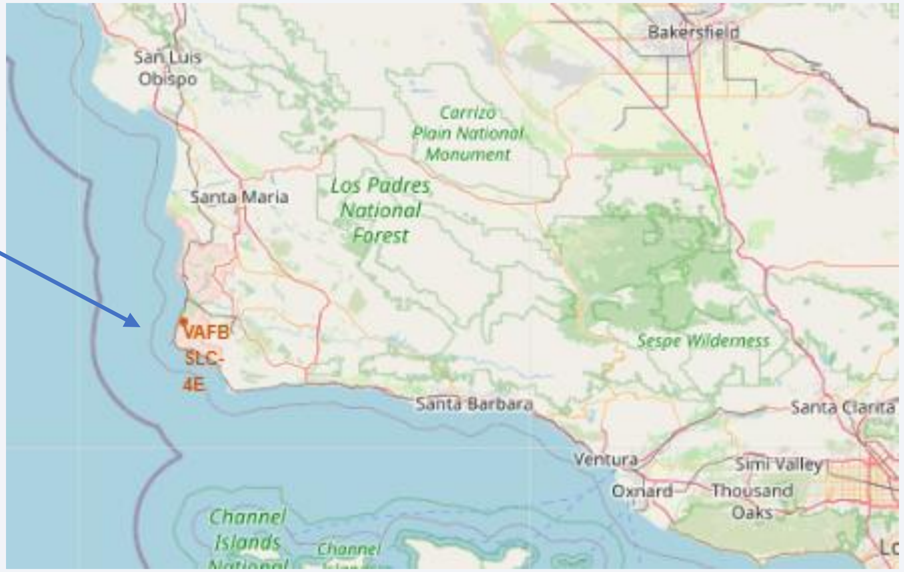
Section 3

Launch Sites Proximities Analysis

Launch site Locations

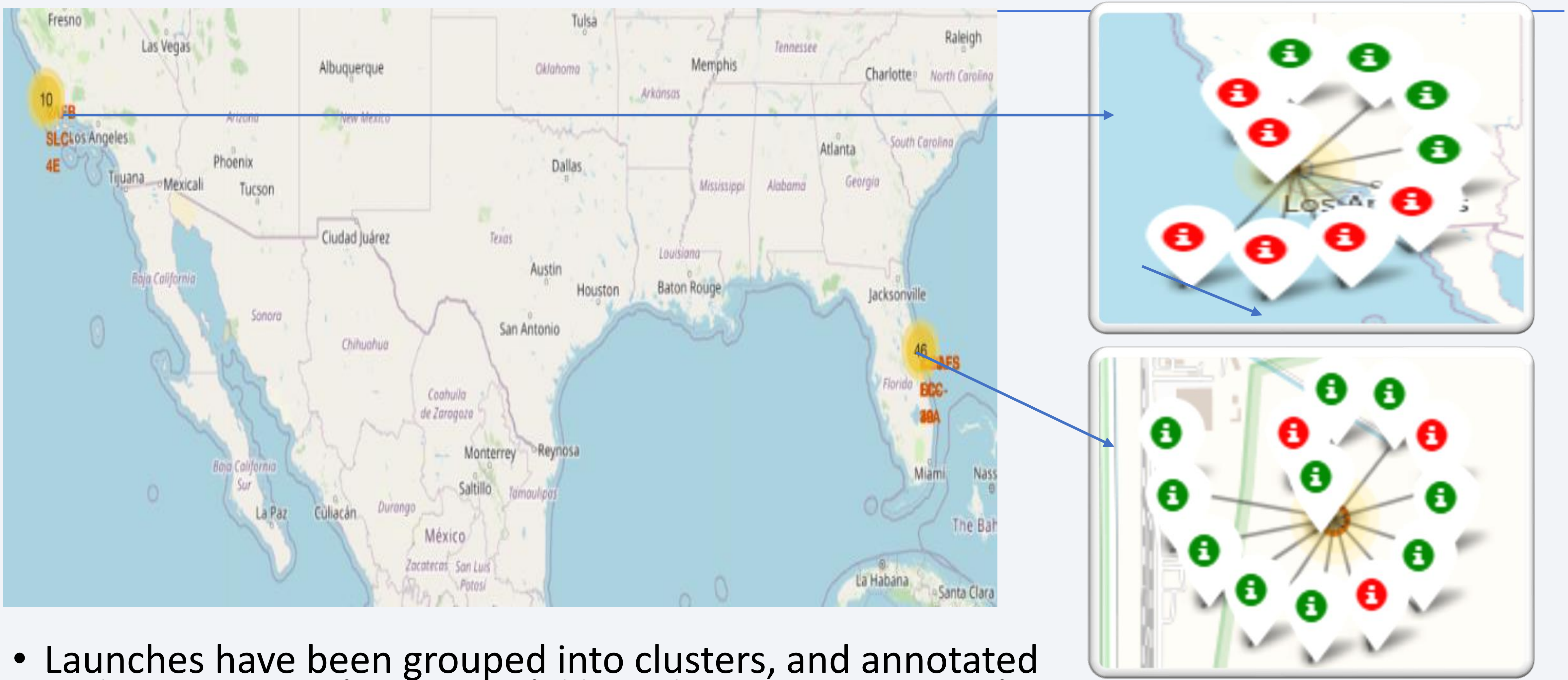






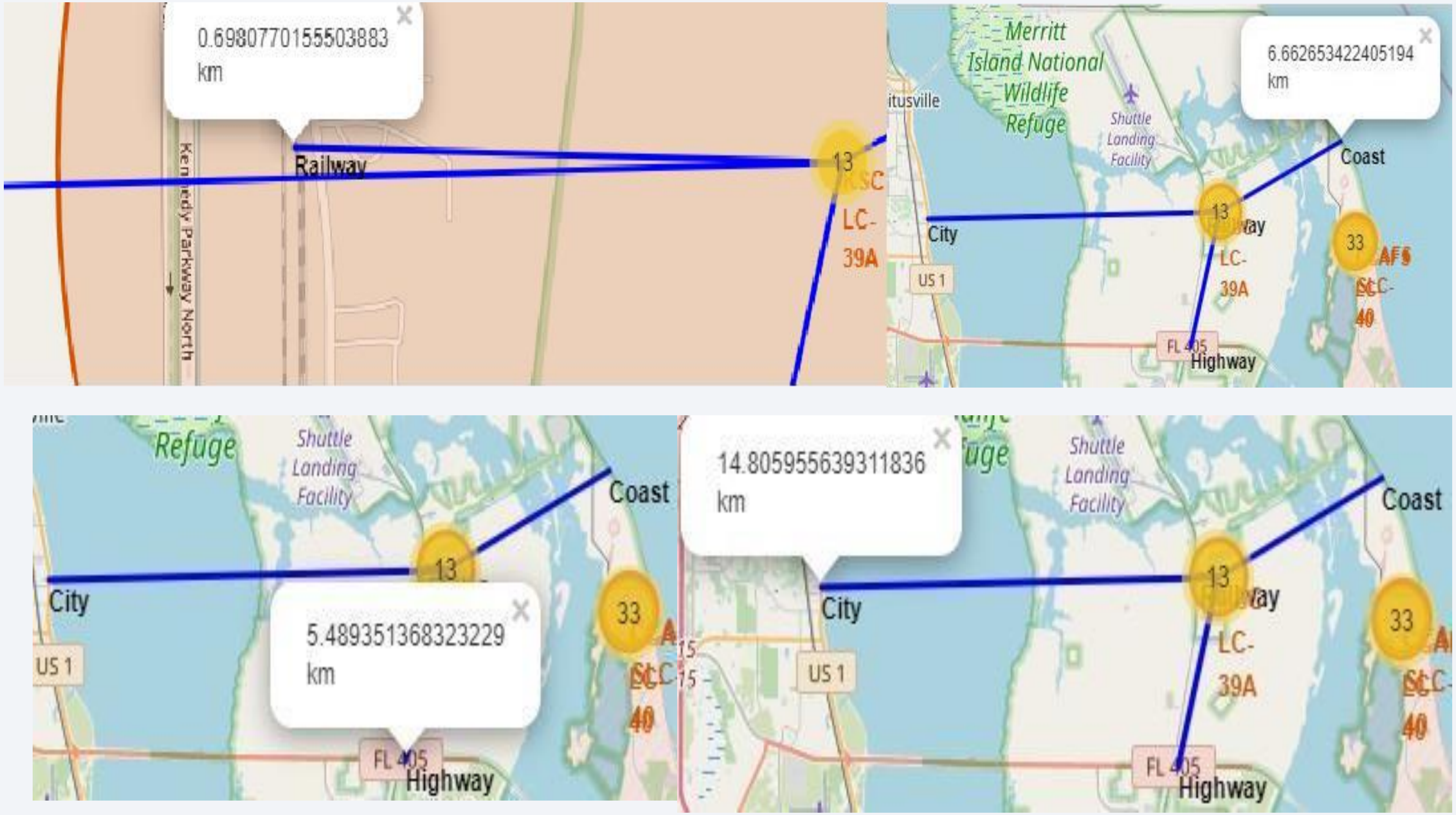
- All SpaceX launch sites are on coasts of the United States of America, specifically Florida and California.
- Each launch site is further shown using arrows.

success/failed launches for each site



- Launches have been grouped into clusters, and annotated with green icons for successful launches, and red icons for failed launches.

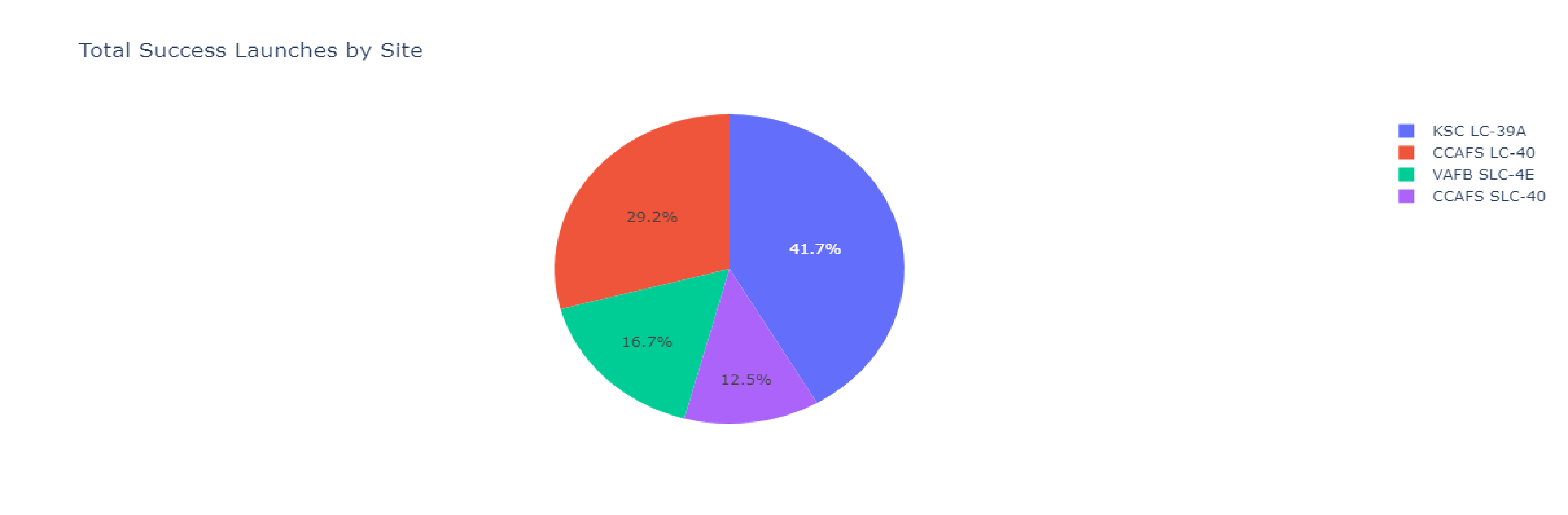
PROXIMITY OF LAUNCH SITES TO OTHERI NTEREST POINTS



Section 4

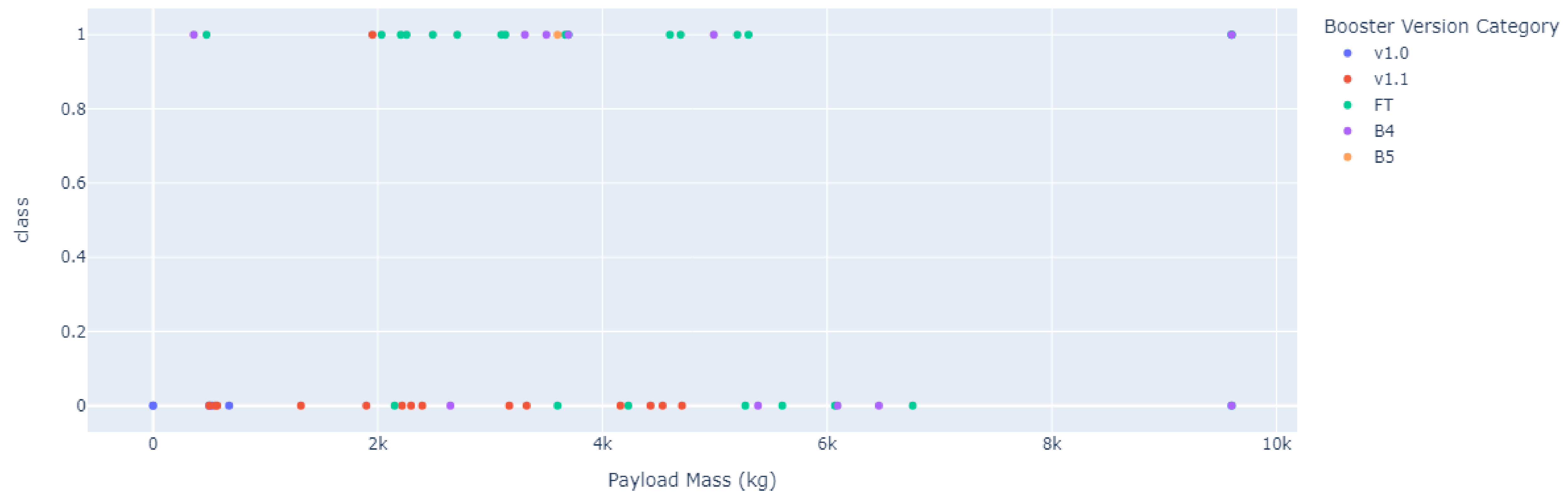
Build a Dashboard with Plotly Dash

Total Success Launches site wise



- The launch site KSC LC-39 is the most successful launches, with 41.7% of the total successful launches.
- The launch site CCAFS SLC 40 with 12.5% is at last number.

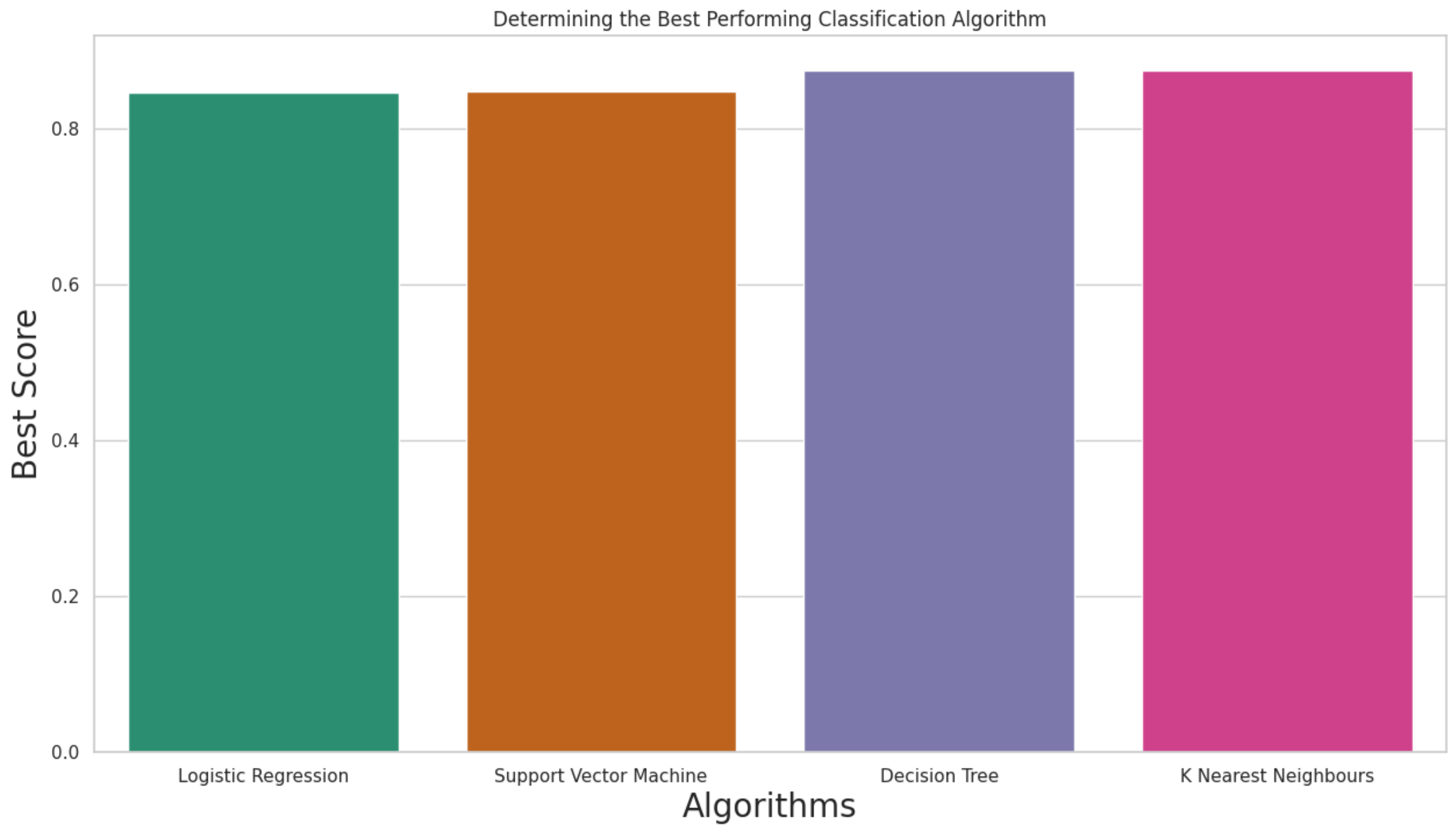
Booster Version Category: Scatter plot for payload mass vs Class



Section 5

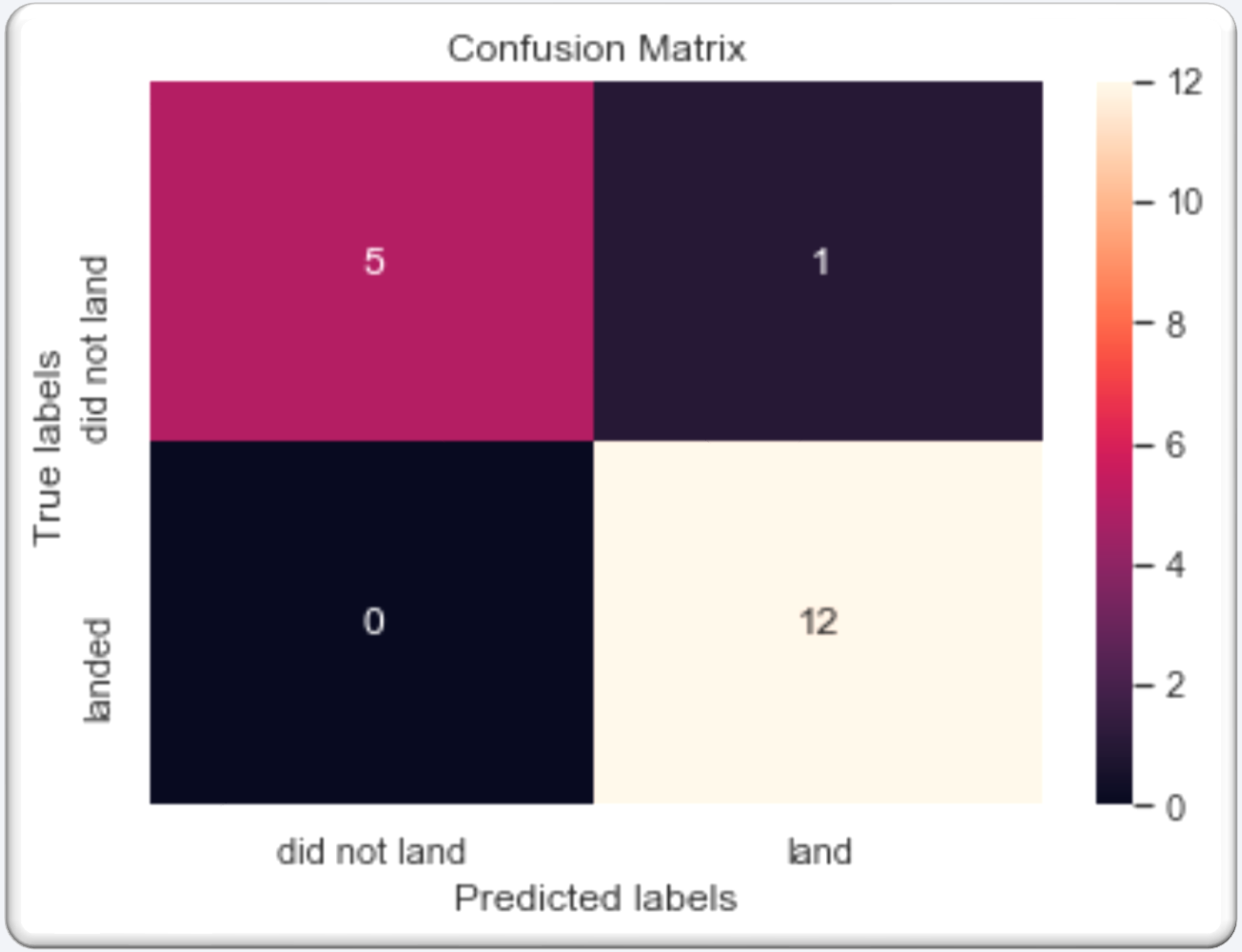
Predictive Analysis (Classification)

Classification Accuracy



	Algorithm	Accuracy Score	Best Score
0	Logistic Regression	0.833333	0.846429
1	Support Vector Machine	0.833333	0.848214
2	Decision Tree	0.722222	0.875000
3	K Nearest Neighbours	0.722222	0.875000

Confusion Matrix



- The best performing classification model is the Decision Tree model, with an accuracy of 94.44%.
- 1 out of 18 total results classified incorrectly
- The other 17 results are correctly classified:
  - 5 not landed
  - 12 landed