IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Domala Venkata Lakshmi Karthik

26-06-24

Outline

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

Executive Summary

Summary of methodologies

We strategy used two distinct approaches: online scraping tools and API integration to gather data. We used a variety of Python data manipulation techniques to carefully treat and clean the dataset after the acquisition stage. SQL queries were then utilized to retrieve relevant data from the cleaned dataset. Systematic data visualization and trend analysis produced early insights. After completing our analytical framework, we put supervised machine learning models into practice to forecast the likelihood that landing events will be successful. To forecast if the landing event would be successful, we used supervised machine learning models.

Summary of all results

We were able to find clear patterns and connections between the variables that directly affect landing event success by carefully analyzing the data. By utilizing this information, we were able to create and hone a predictive model that was remarkably effective at predicting the likelihood of a successful landing event. The model's noteworthy accuracy rating of 83% highlights its efficacy in providing dependable prognostications in this particular domain.

Introduction

- SpaceX's dedication to reusable rockets has reduced the cost of space travel considerably by carefully concentrating on the first rocket phase's retrieval. Recovering this first stage is critical to maintaining and repurposing costly parts, which directly lowers costs. A thorough examination of the success rate of these retrieval events provides an important indicator for assessing the effectiveness and affordability of SpaceX's innovative strategy. The goal of this specific research is to forecast the first phase retrieval event's success, providing predictive insights that will improve space industry decision-making.

- Predicting the first-phase rocket retrieval's success is our goal, with the ultimate goal being resource allocation optimization. We want to improve mission success rates and make significant cost savings by obtaining this predictive capability.

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

Describe how data sets were collected.

The SpaceX launch data was collected using SpaceX API (REST API) by making a get request to the SpaceX API. This was done in several steps like, first defining a series of helper functions that would be useful to extract information using identification number in the launch data and then requesting rocket launch data from the SpaceX API url.

Finally we will get JSON result, so to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as a Json result which was then converted into a Pandas data frame.

Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in a HTML. Using BeautifulSoup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas DataFrame.

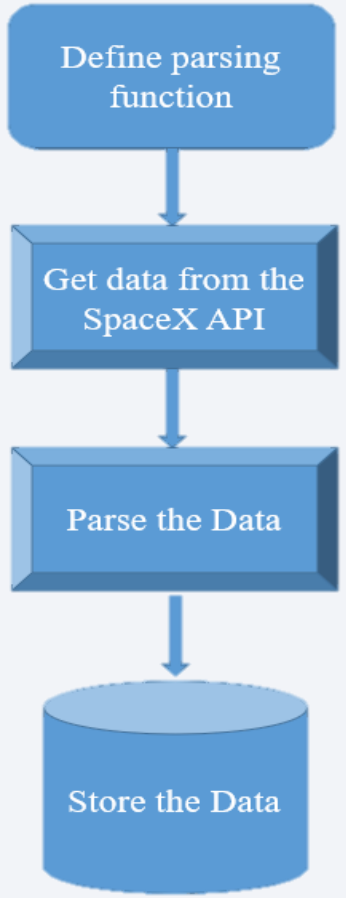
Data Collection – SpaceX API

1) Define auxiliary function to parse the data.

2) Retrieve data from the **REST API** using the method **GET.**

3) Parse the data with the previously built auxiliary functions.

4) Store the data in **PANDAS DataFrame**.



GitHub URL of the completed SpaceX API calls notebook:

- Click Here

Data Collection - Scraping

1) Using the get request.get method to download page code.

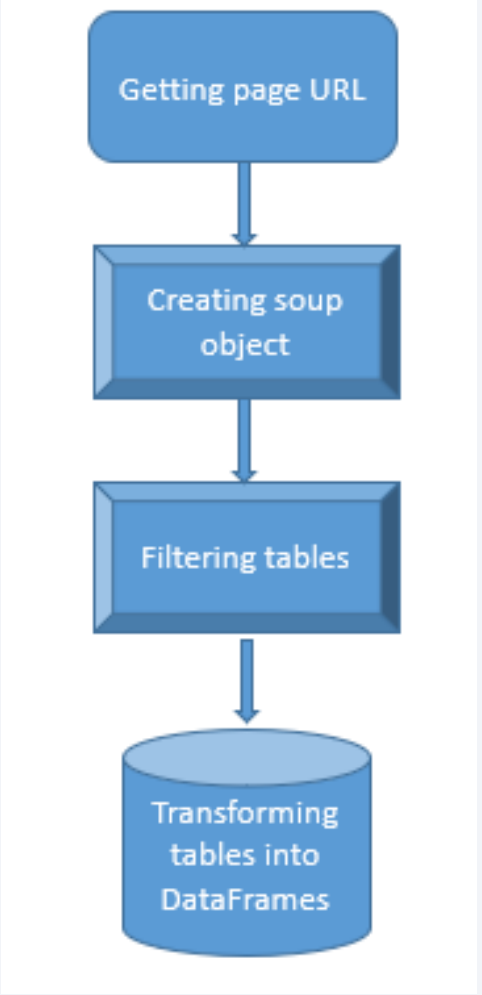
2) Created a BeautifulSoup object to manipulate the html text.

3) Filtered the desired tables using soup manipulation methods.

4) Converted the data from the HTML to pandas DataFrame format.

GitHub URL of the completed web scraping notebook:

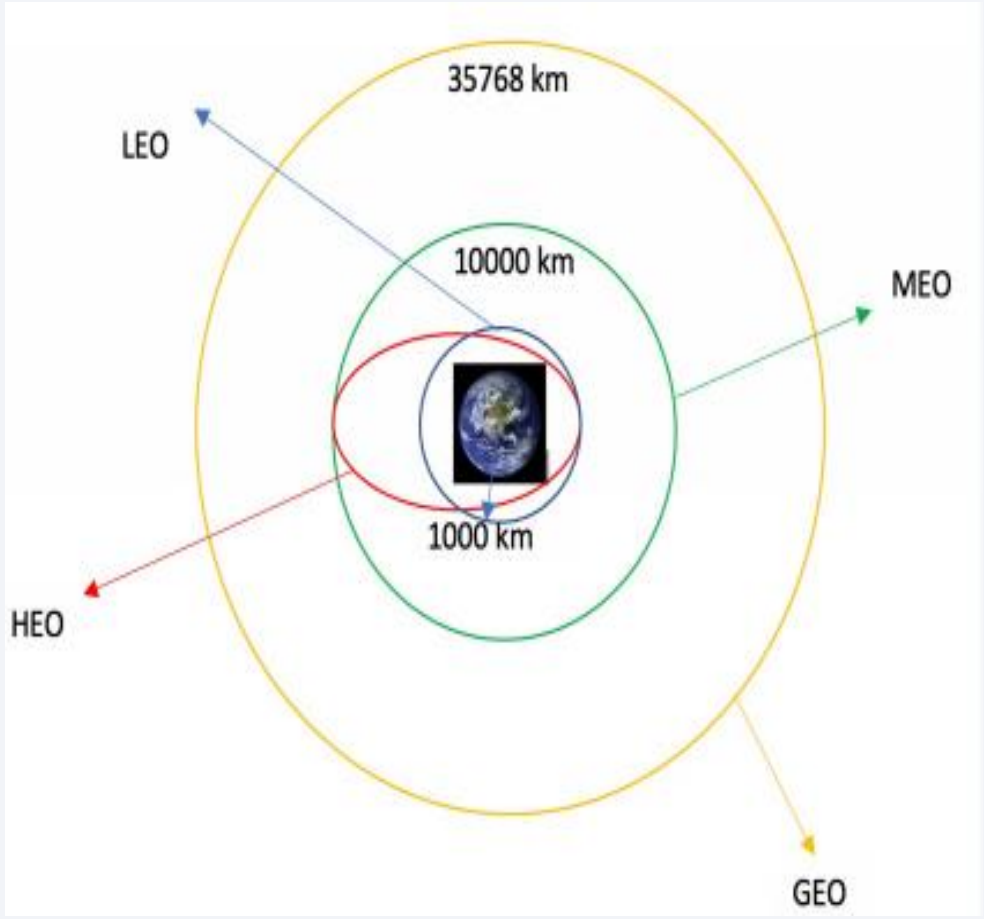
- Jupyter Lab Webscraping



Data Wrangling

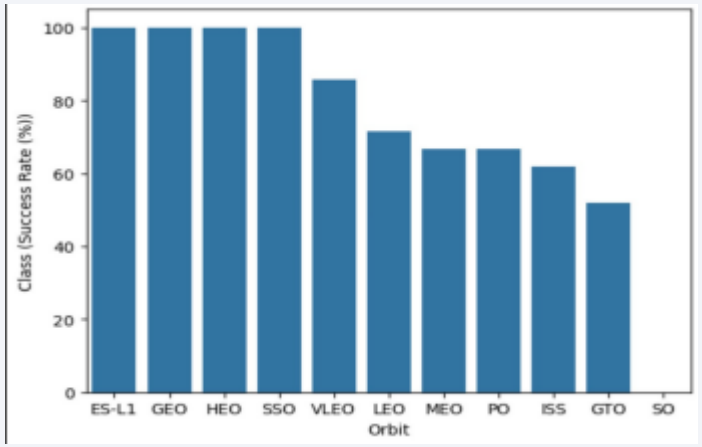
- We identified the training labels by doing an exploratory data analysis.
- We determined the quantity of launches at every location as well as the frequency of each orbit.
- From the outcome column, we generated a landing outcome label and exported the data to CSV.
- Github Notebook file data wrangling

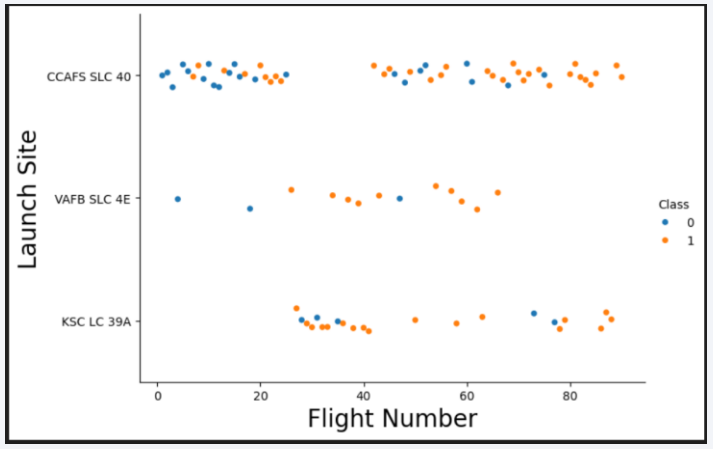
  Click Here

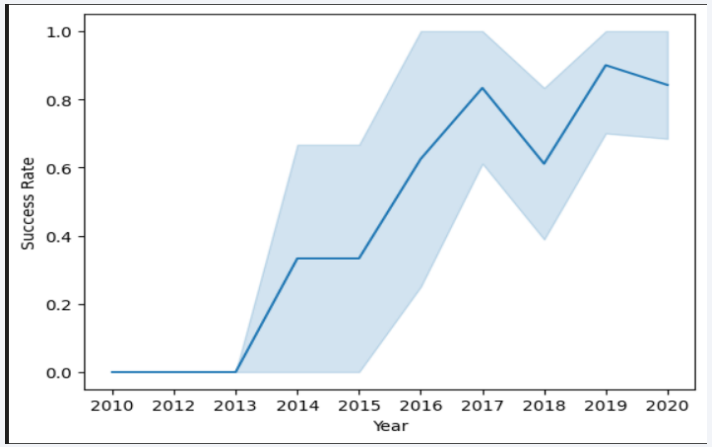


EDA with Data Visualization

- We have explored the data using EDA – Exploratory data analysis with data visualization and fount out the relation between flight number and launch site, payload and launch site, Success rate of each orbit type, flight number and orbit type, payload and orbit type, launch success yearly trend.







For Github Notebook File [Click Here]

EDA with SQL

- Using SQL, we had performed many queries to get better understanding of the dataset.
- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string ‘CCA’.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- 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.
- For Github Link Click Here

Build an Interactive Map with Folium

- On the folium map, we annotated every launch point and added map elements like circles, lines, and markers to indicate the success or failure of launches at each location.
- Class 0 and Class 1 were given the feature launch outcomes (success or failure).i.e., 1 for accomplishment and 0 for failure.
- It was possible to determine which launch sites have a comparatively high success rate by using the color-labeled marker clusters.
- 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.
- For Github link Click Here

Build a Dashboard with Plotly Dash

- We built an interactive dashboard with Plotly dash.
- We plotted pie charts showing the total launches by a certain sites.
- We plotted scatter graph showing the relationship with Outcome and
- PayloadMass (Kg) for the different booster version.

- For Github Notebook Click Here

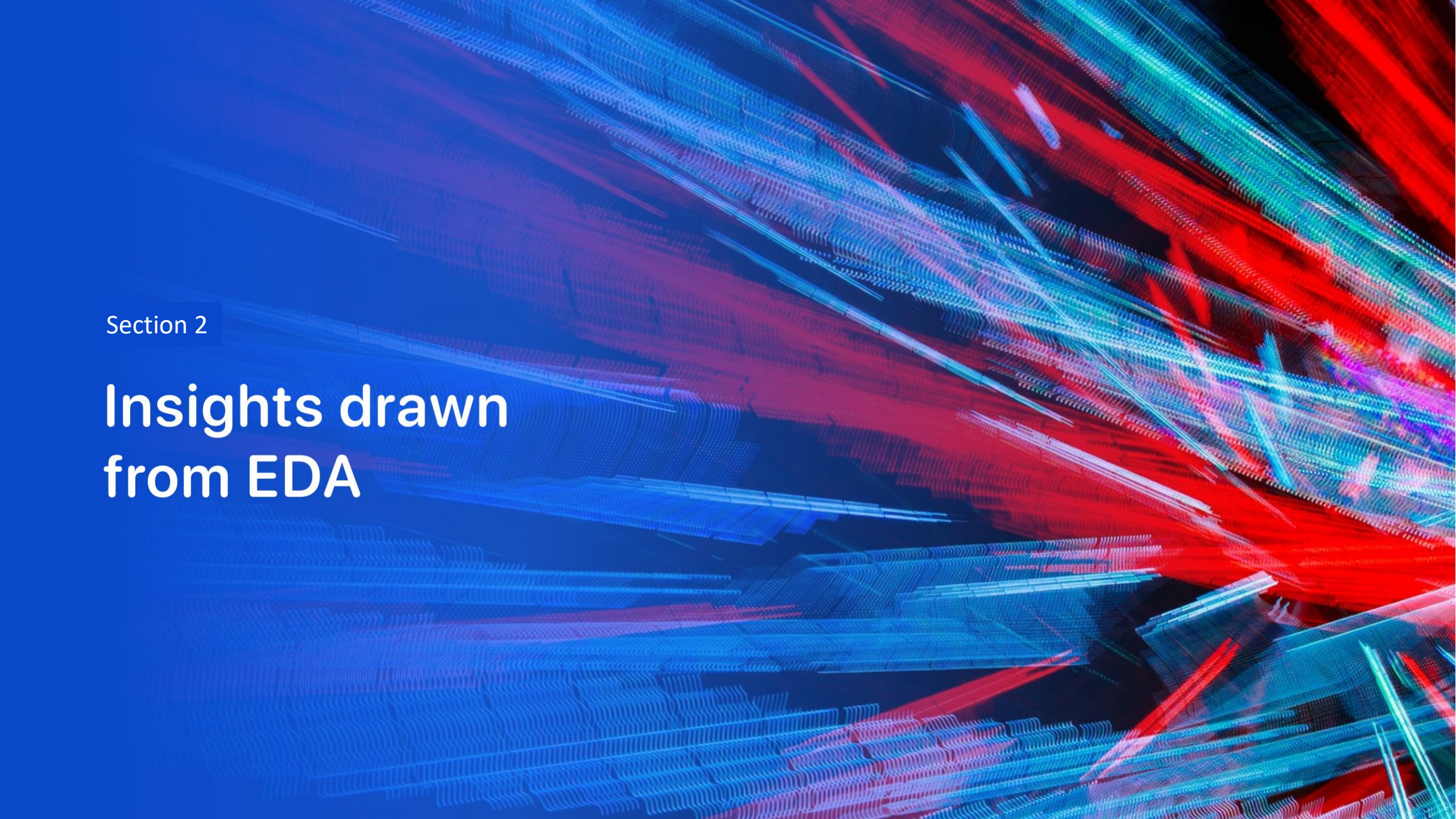
Predictive Analysis (Classification)

- We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.
- We built different machine learning models and tune different hyperparameters using GridSearchCV.
- We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.
- We found the best performing classification model.

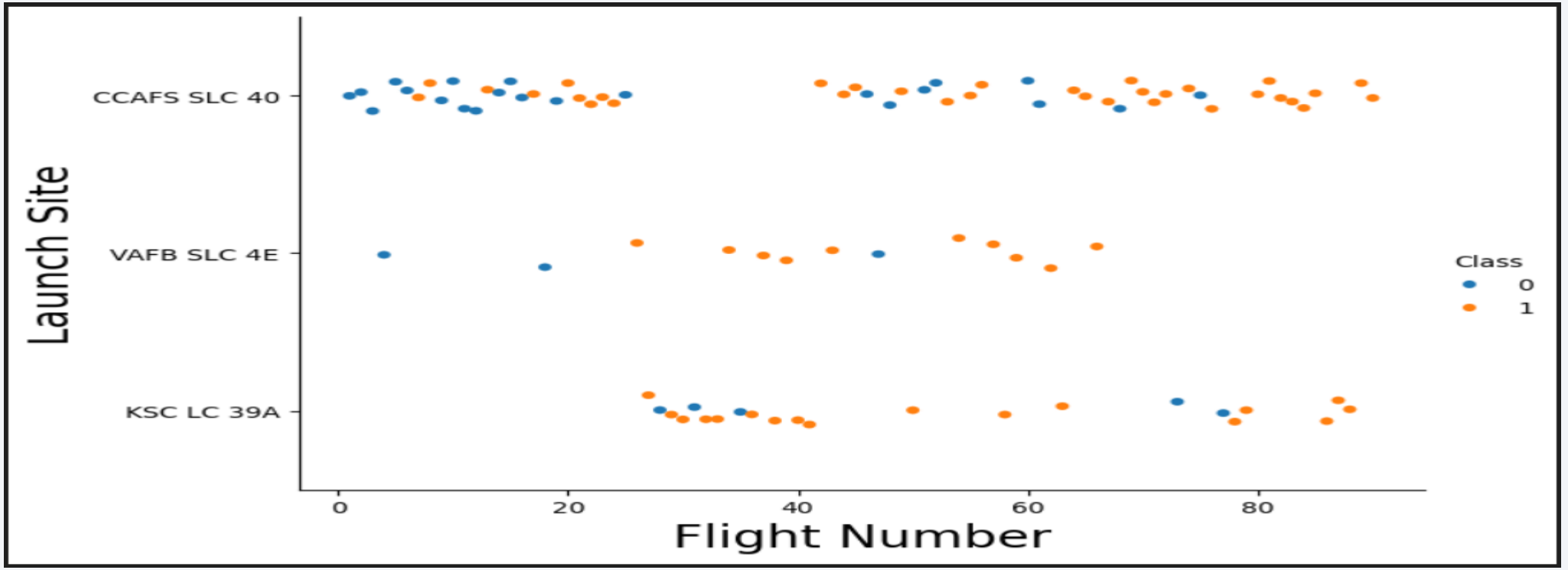
- For Github Notebook Click Here

Results

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

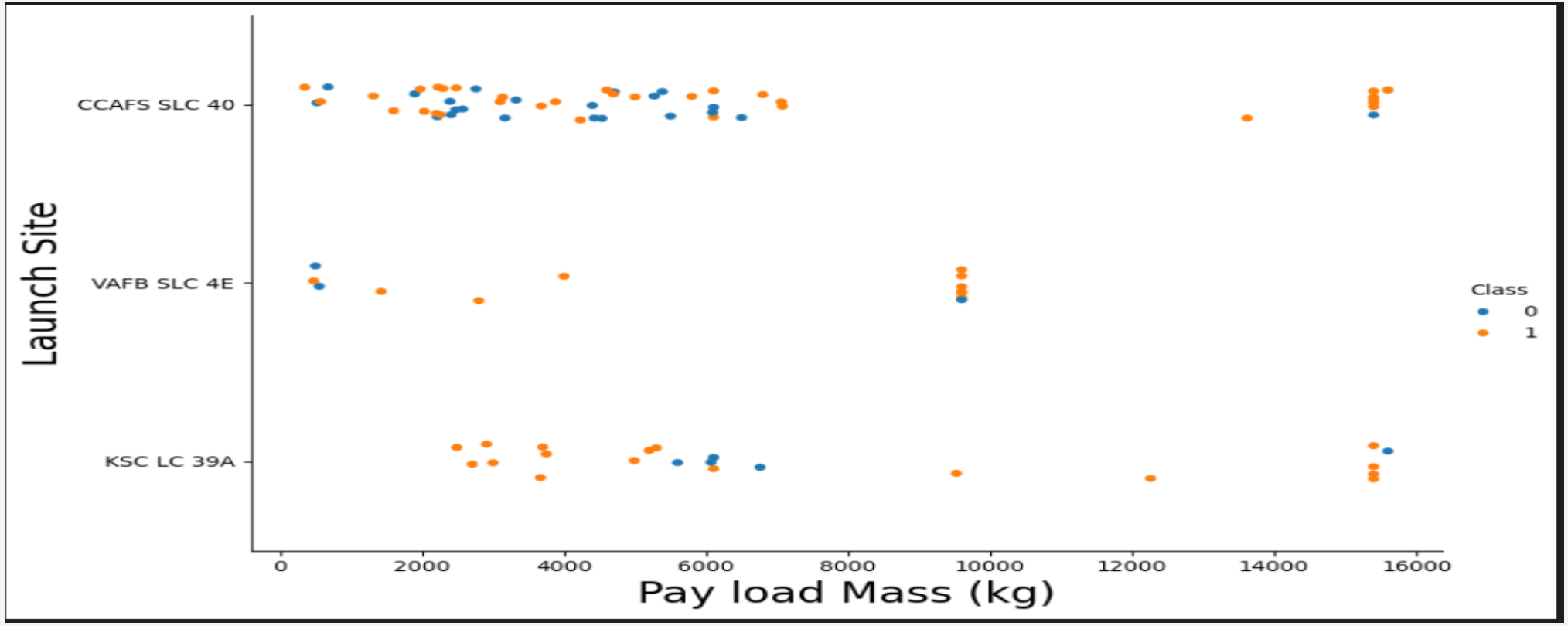

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



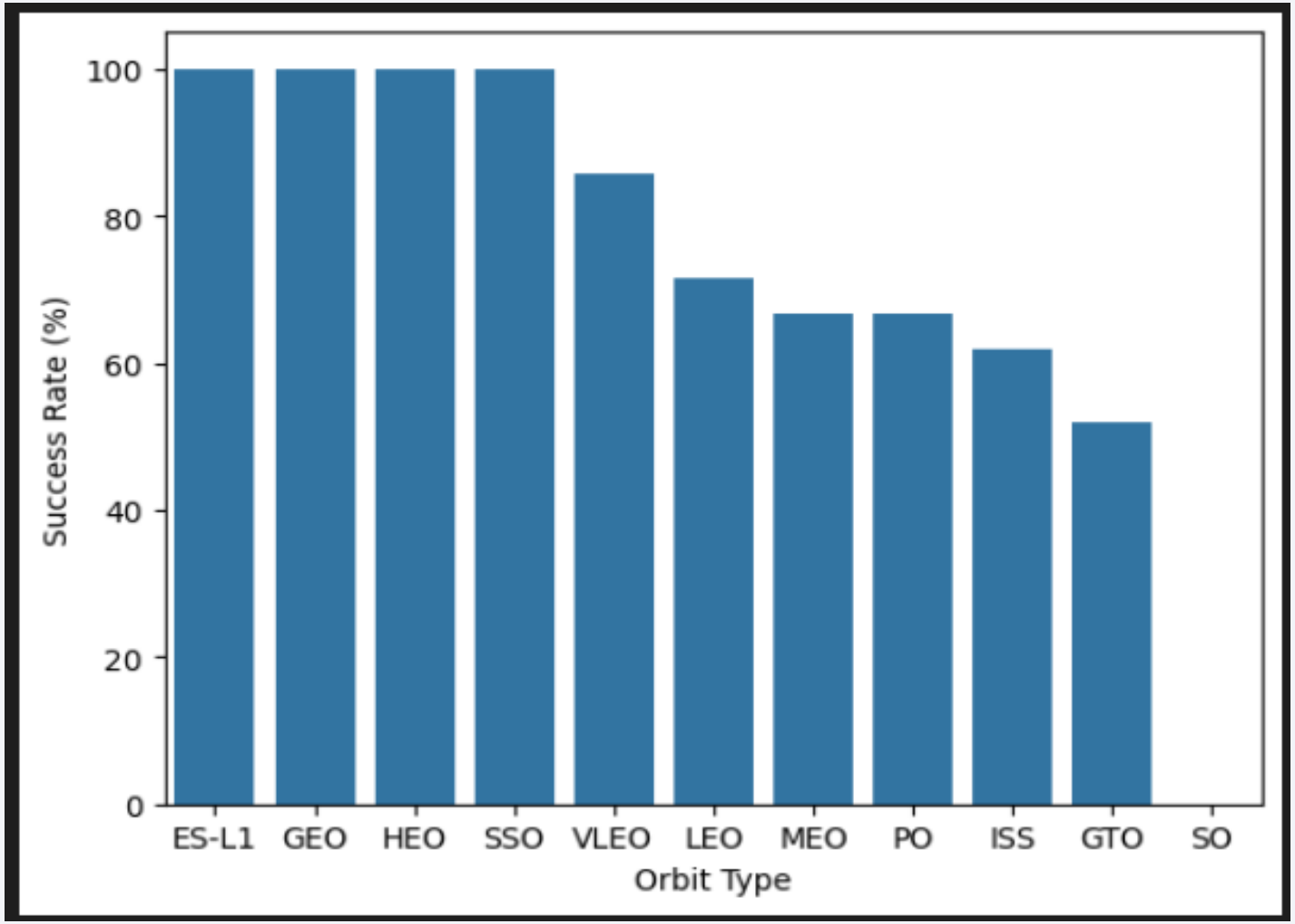
- We deduced from the plot that a launch site's success rate increased with the number of flights conducted there.

Payload vs. Launch Site



- The higher the rocket's success percentage, the larger the payload mass at launch point CCAFS SLC 40.

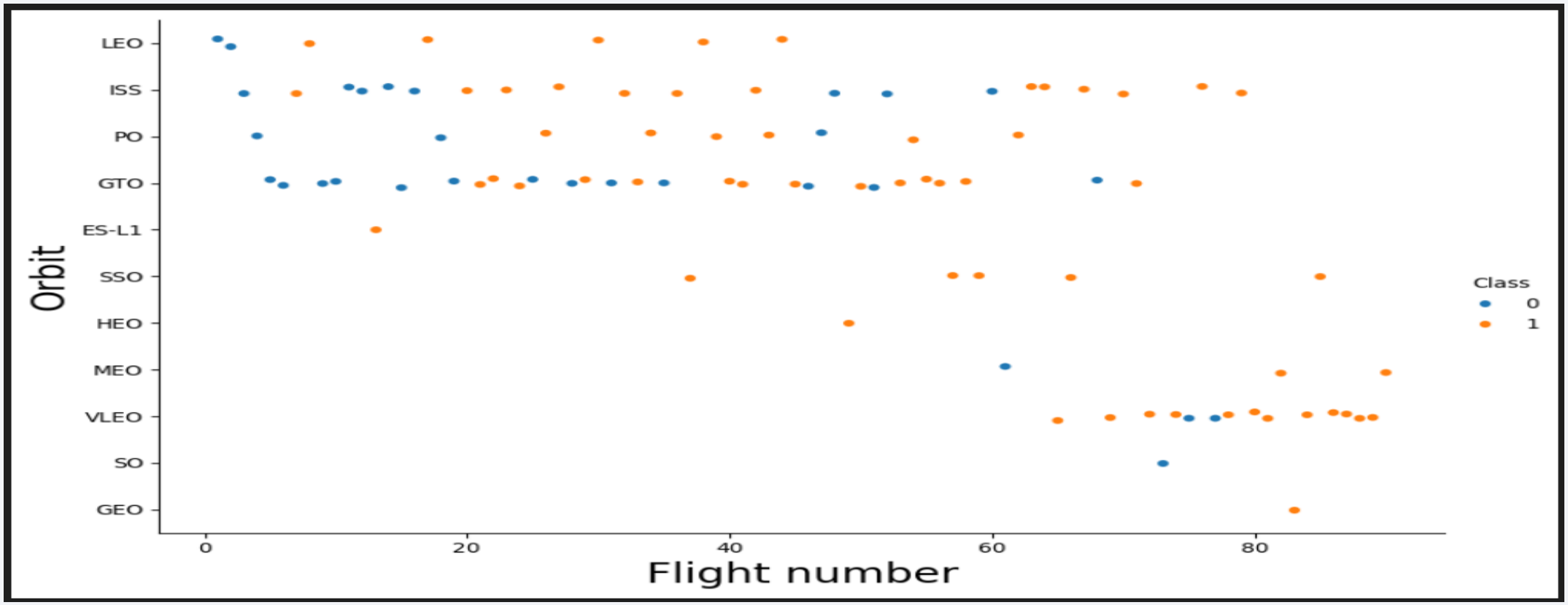
Success Rate vs. Orbit Type



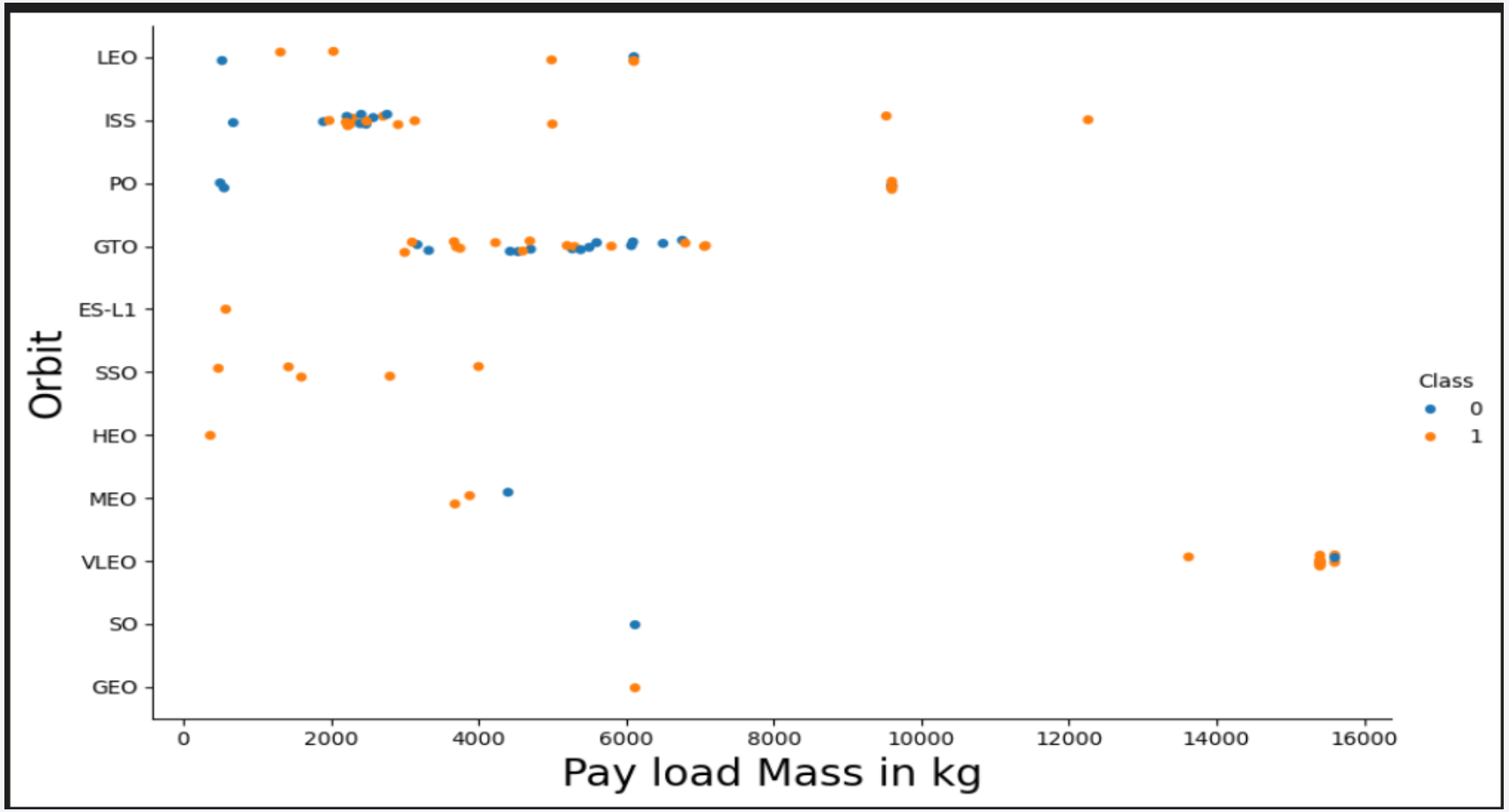
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type

- The Flight Number versus Orbit type is displayed in the plot below. We note that while there is no correlation between trip number and orbit in the GTO orbit, there is in the LEO orbit where success is correlated with the number of flights.

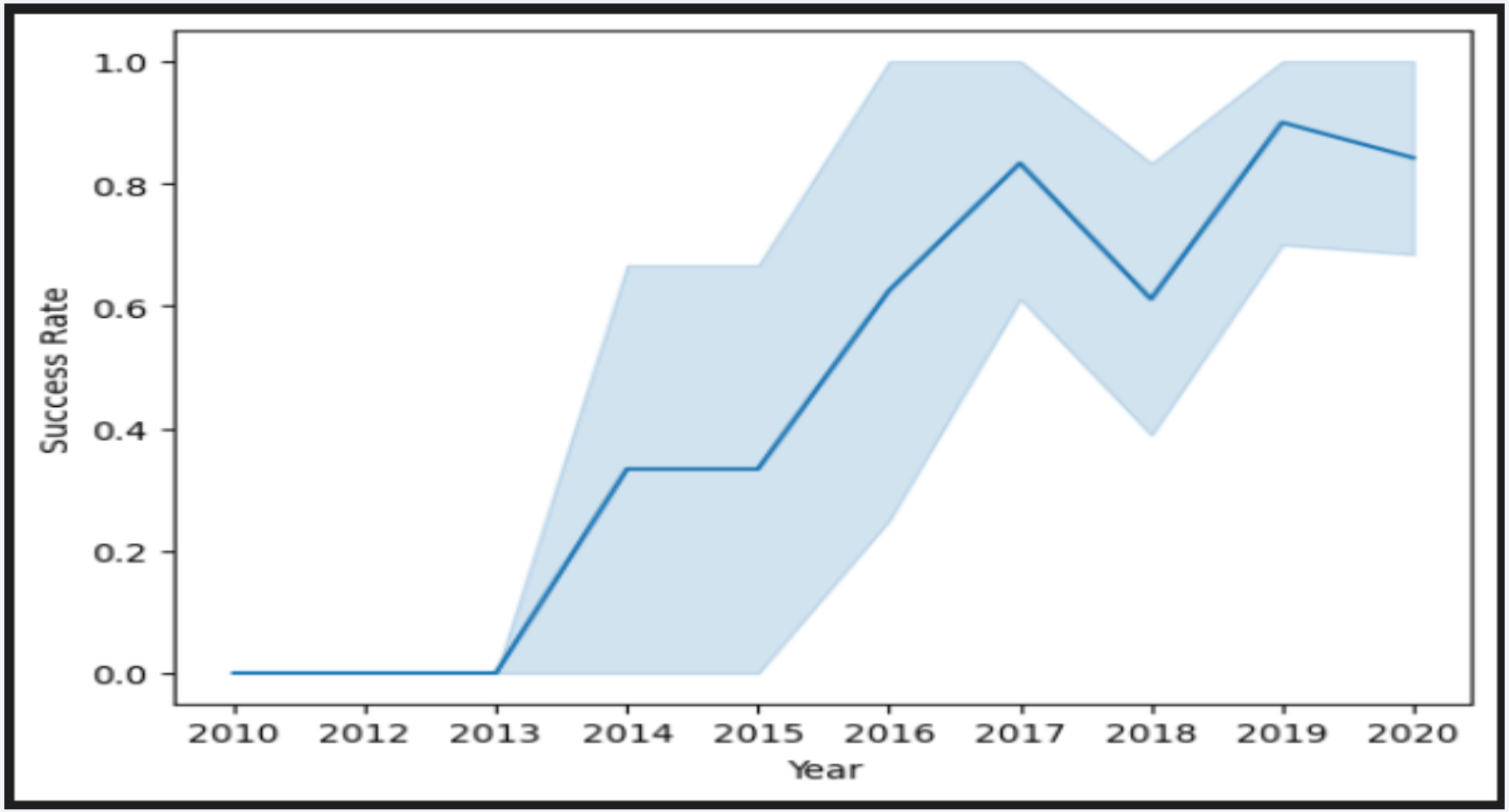


Payload vs. Orbit Type



- We can observe that with heavy payloads, the successful landing are more for PO,LEO and ISS orbits.

Launch Success Yearly Trend



- From the plot, 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.

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

```
[16] %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string '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

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

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 45596 using the query below
- %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';

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

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
```

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

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 B1003 as 2534.66666666666.

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

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

First Successful Ground Landing Date

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

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

Hint:Use min function

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

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

Date
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

- We used the WHERE clause to filter for boosters which have success fully landed on drone-ship and applied the and condition to determine successful landing with payload mass greater than 4000 but less than 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 DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing_Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

Total Number of Successful and Failure Mission Outcomes

Total Number of failure mission outcome are 1 and

Total number of success mission outcomes are 100

List the total number of successful and failure mission outcomes

```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

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

Boosters Carried Maximum Payload

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

```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

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

2015 Launch Records

- 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.

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,0,5), substr(Date, 6, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing_Outcome" FROM SPACEXTBL WHERE substr(Date,0,5
```

Python

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

substr(Date,0,5)	substr(Date, 6, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing_Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

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 * FROM SPACEXTBL WHERE "Landing_Outcome" LIKE 'Success%' AND (Date BETWEEN '2010-06-04' AND '2017-03-20') ORDER BY Date DESC;
```

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

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Section 3

Launch Sites Proximities Analysis

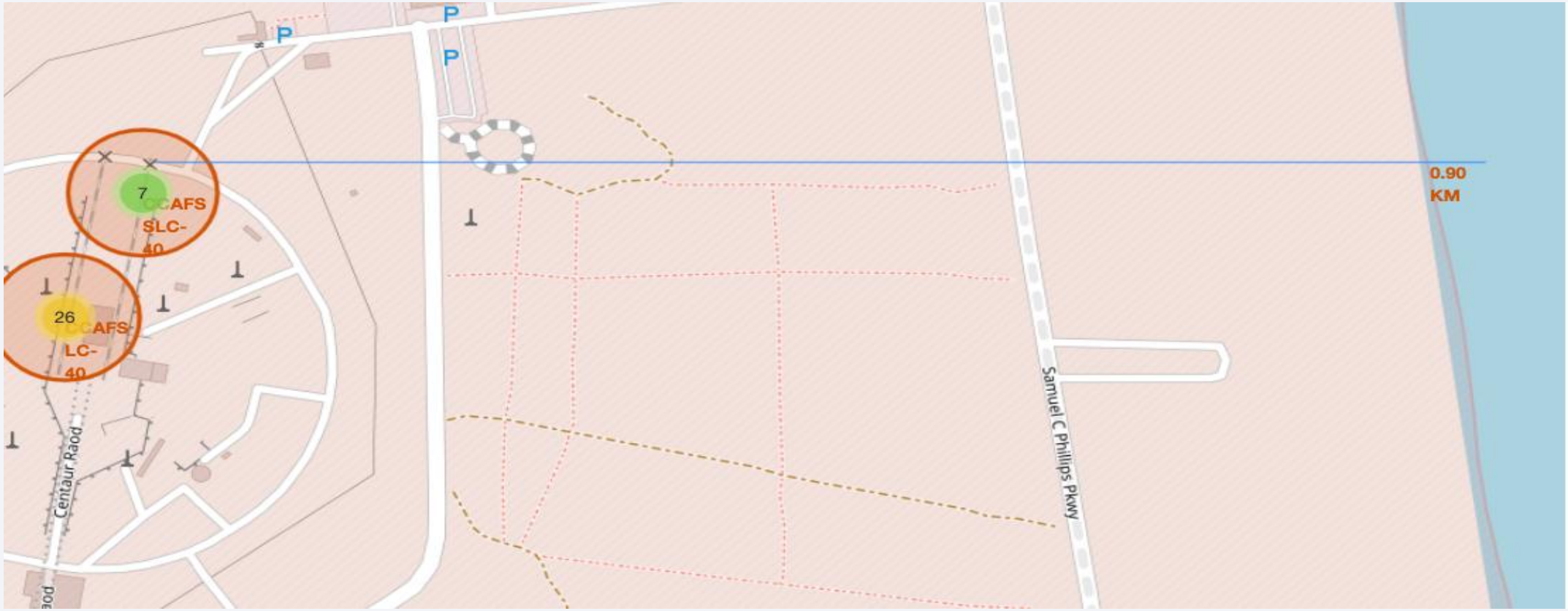
All Launch sites global map markers

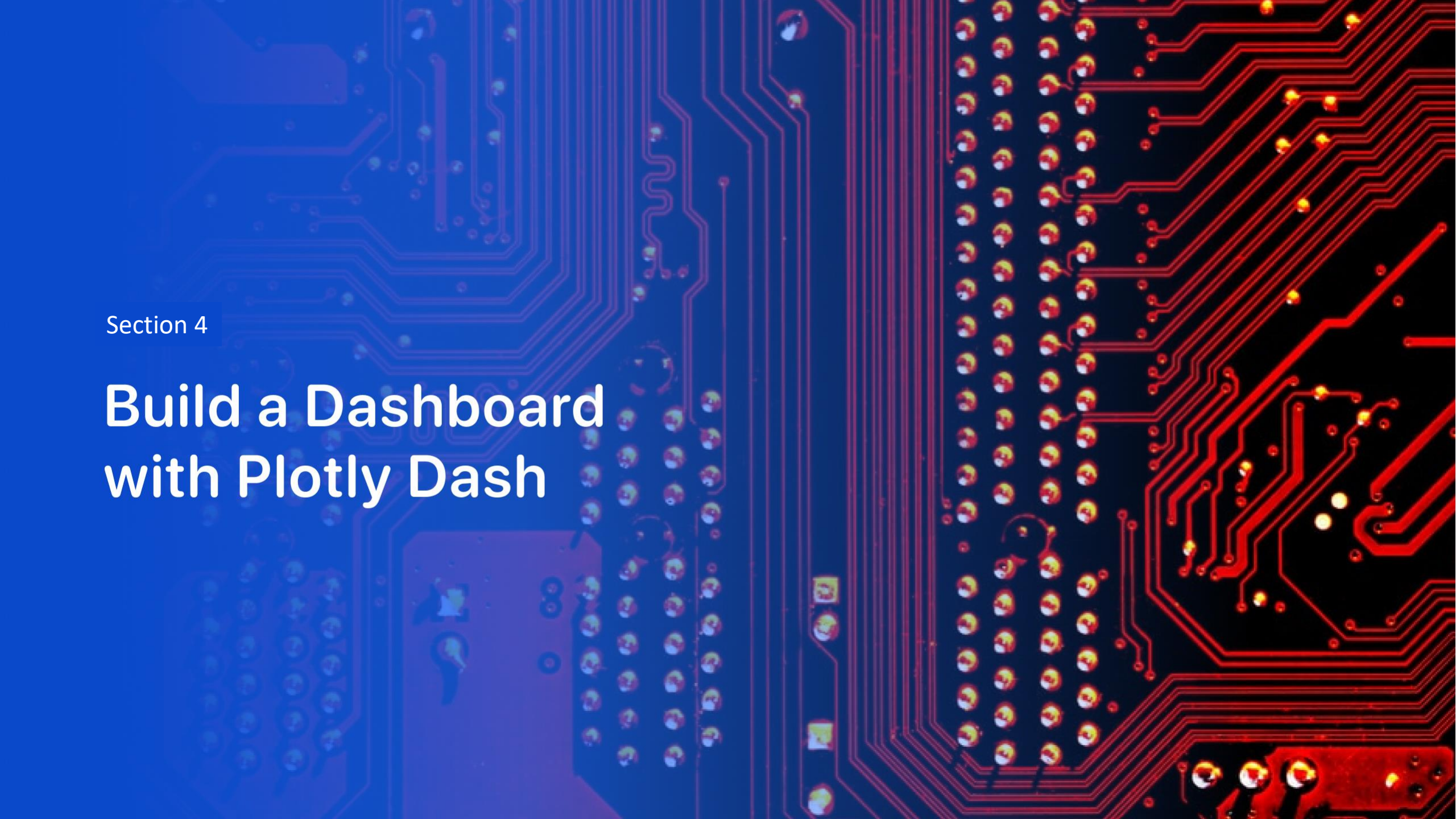
- We can see that the Space launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color lables

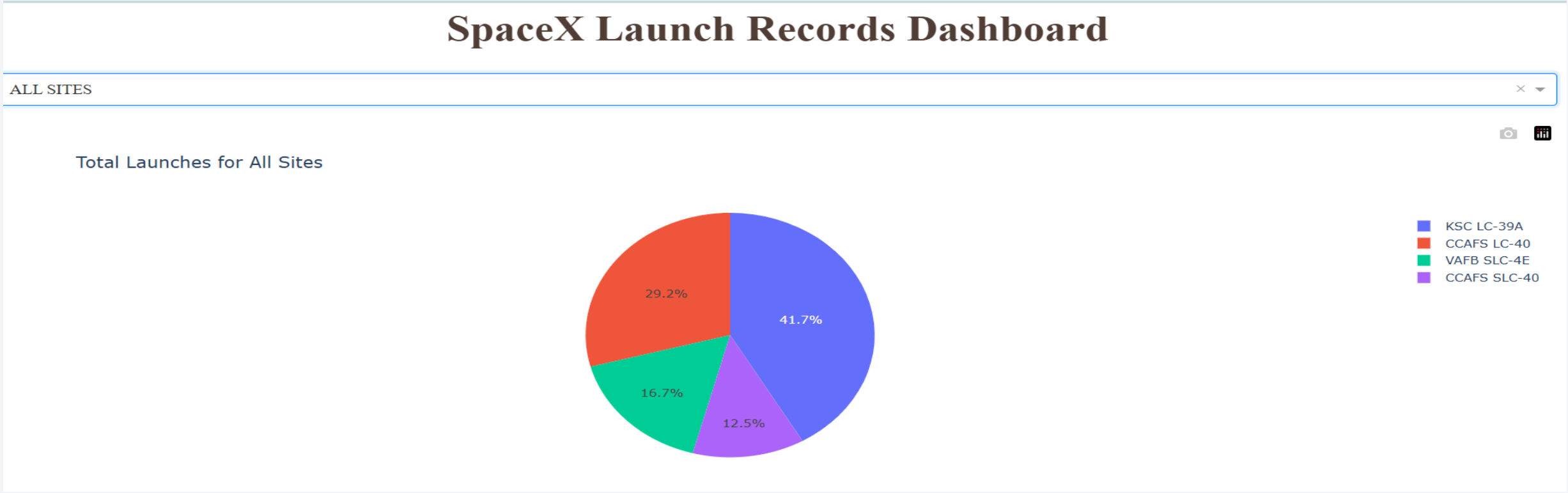
- Green marker -> successful launches.
- Red marker showing unsuccessful launches,

Launch sites distance to landmarks



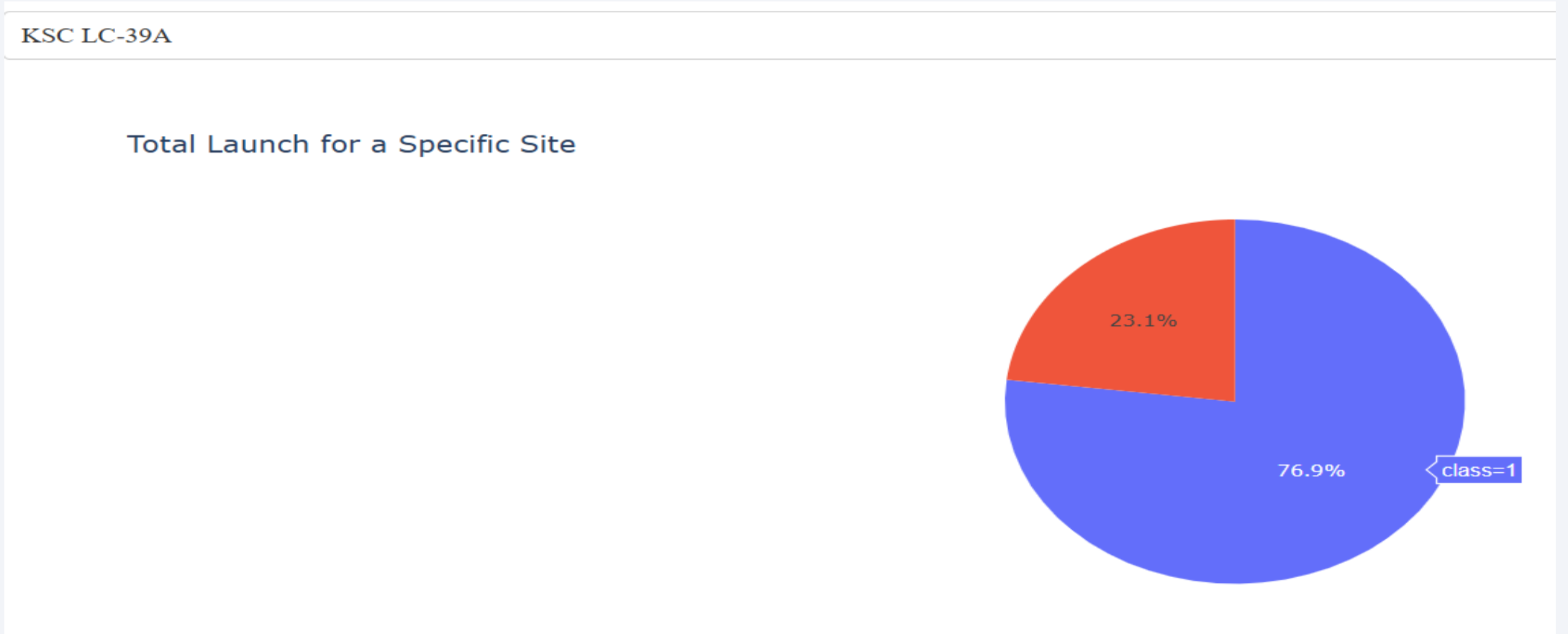

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



- We can see that KSC LC-39A had the most successful launches from all the sities

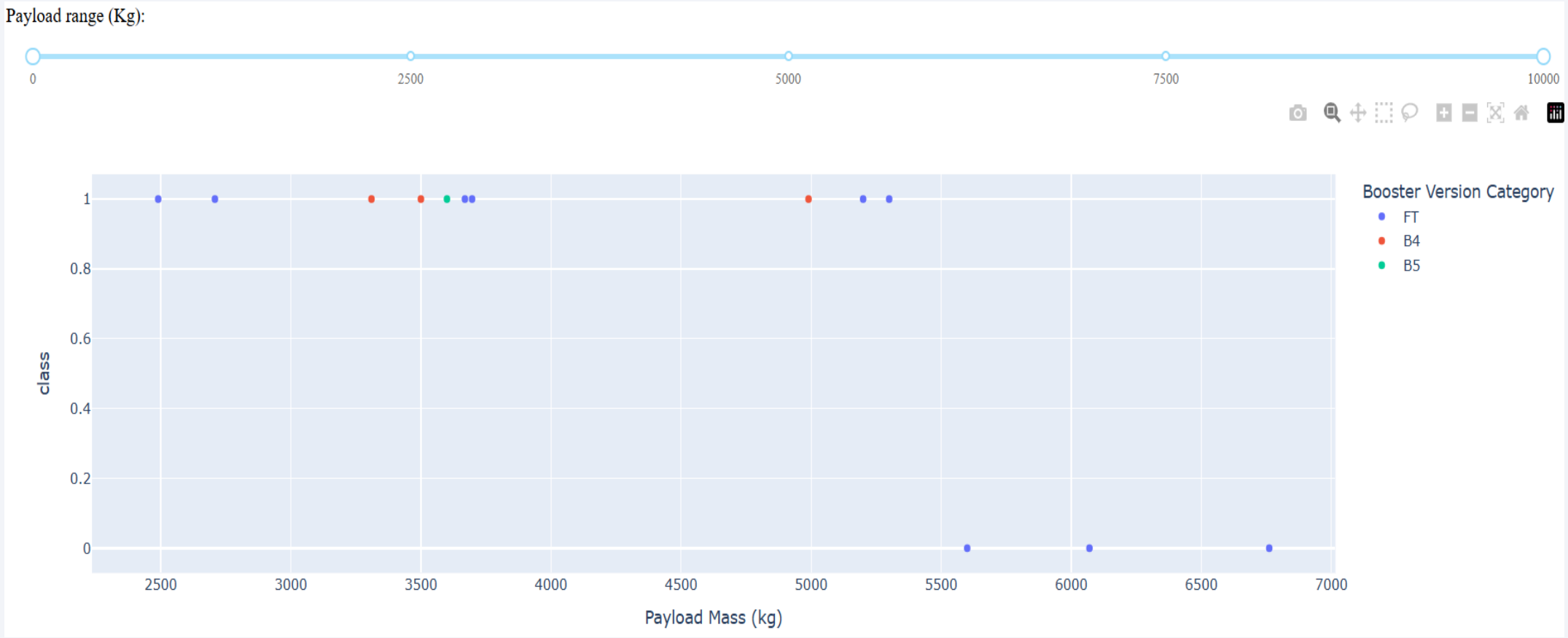
Pie chart showing the launch site with the highest launches success ratio



KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

Scatter plot of the payload vs launch outcome for all sites, with different payload selected int 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

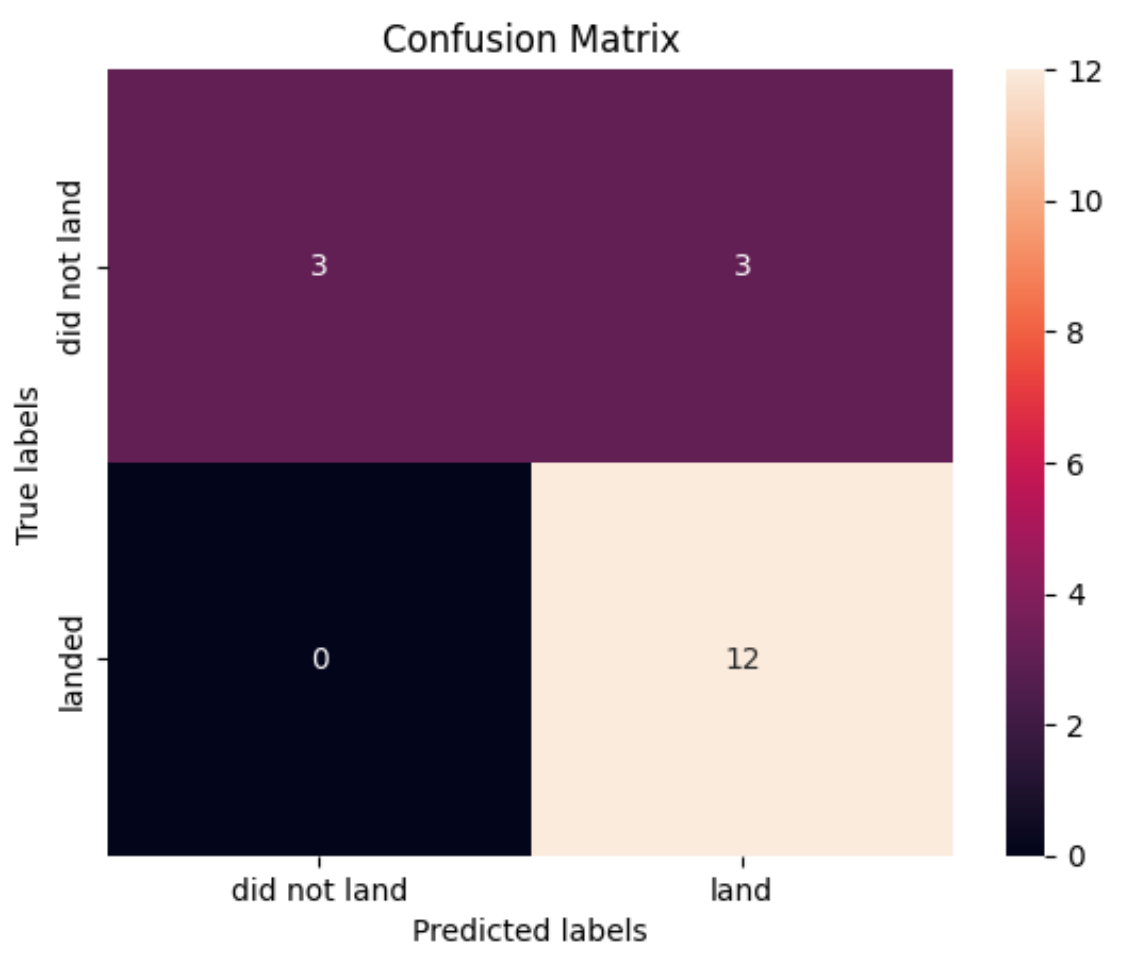
- The decision tree classifier is the model with the highest classification accuracy

Find the method performs best:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
best_alg = max(algorithms, key=algorithms.get)
print('Best Algorithm is',best_alg,'with a score of',algorithms[best_alg])
if best_alg == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if best_alg == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if best_alg == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8892857142857142
Best Params is : {'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix



- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.The major problem is the false positives .i.e.,unsuccessful landing marked as successful landing by the classifier.

Conclusions

We can conclude that:

- 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 Decision tree classifier is the best machine learning algorithm for this task.

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

- Github Link Click Here

Thank you!