IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Glexy Sundesha

March 2024

Outline

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

Executive Summary

- Methodologies
  - Data Collection
  - Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- Project background and context
  - SpaceX, a leader in the industry, aims to make space travel affordable . Its accomplishments include sending spacecraft to the international space station, launching a satellite constellation that provides internet access and sending manned missions to space. SpaceX can do this because the rocket launches are relatively inexpensive due to its reuse of the first stage of its Falcon 9 rocket. By determining if the first stage will land, we can determine the price of the launch. To do this, we can use public data and machine learning models to predict whether SpaceX .
- Problems you want to find answers
  - Main characteristics of a successful or failed landing ?
  - Success or failure of a landing ?
  - Conditions that allow SpaceX to achieve the best landing success rate ?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping
- Perform data wrangling
  - Dropping unnecessary columns
- 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

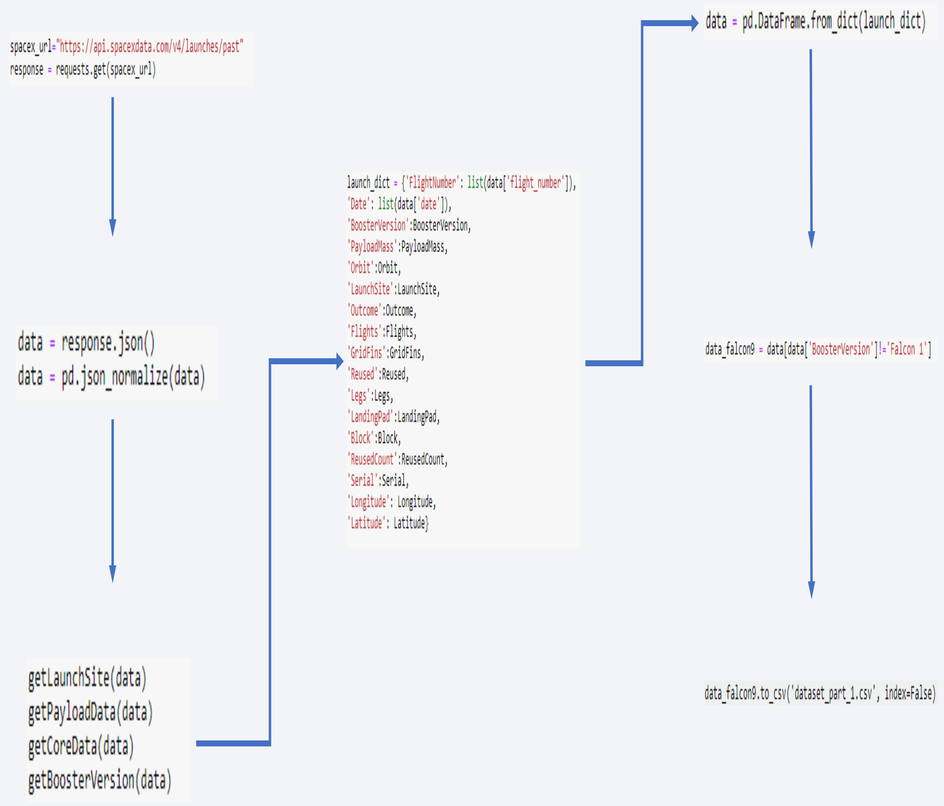
Data Collection

- Request data- from SpaceX API (rocket launch data)
- Decode response- using .json() and convert to a dataframe using .json_normalize()
- Request information- about the launches from SpaceX API using custom functions
- Create dictionary- from the data
- Create dataframe- from the dictionary
- Filter dataframe- to contain only Falcon 9 launches
- Replace missing values- of Payload Mass with calculated .mean()
- Export data- to csv file

Data Collection – SpaceX API

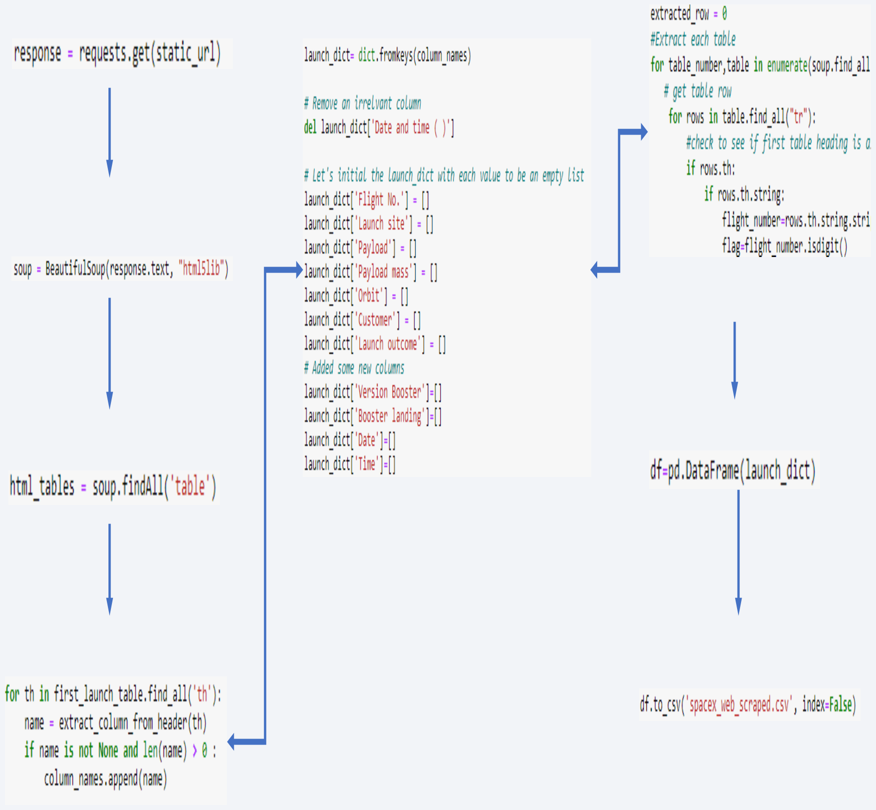
1. Getting Response from API
2. Convert Response to JSON File
3. Transform data
4. Create dictionary with data
5. Create dataframe
6. Filter dataframe
7. Export to file

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/jupyter-labs-spacex-data-collection-api(2).ipynb



Data Collection - Scraping

1. Getting Response from HTML
2. Create BeautifulSoup Object
3. Find all tables
4. Get column names
5. Create dictionary
6. Add data to keys
7. Create dataframe from dictionary
8. Export to file



https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

- In the dataset, there are several cases where the booster did not land successully.
- We need to transform variables into categorical variables where 1 means success and 0 means failure.

1. Calculate launches number for each site

2. Calculate the number and occurence of each orbit

3. Calculate number and occurrence of mission outcome per orbit type

4. Create landing outcome label from Outcome column

5. Export to file

```
df['LaunchSite'].value_counts()

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

```
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
SO        1
ES-L1     1
HEO       1
GEO       1
Name: Orbit, dtype: int64
```

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes

True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
None ASDS       2
False Ocean     2
False RTLS      1
Name: Outcome, dtype: int64
```

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

```
df.to_csv("dataset_part_2.csv", index=False)
```

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- Scatter Graphs
  - Flight Number vs. Payload Mass
  - Flight Number vs. Launch Site
  - Payload vs. Launch Site
  - Orbit vs. Flight Number
  - Payload vs. Orbit Type
  - Orbit vs. Payload Mass
- Bar Graph
  - Success rate vs. Orbit
- Line Graph
  - Success rate vs. Year

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

- We performed SQL queries
  - Displaying the names of the unique lauunch sites in the space mission.
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS).
  - Display average payload mass carried by booster version F9 v1.1.
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
  - List the total number of successful and failure mission outcomes.
  - List the names of the booster_versions which have carried the maximum payload mass.
  - List the records which will display the month names, faiilure landing_ouutcomes in drone ship, booster versions, launch_site for the months in year 2015.
  - 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.

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/jupyter-labs-eda-sql-coursera_sqllite(2).ipynb

Build an Interactive Map with Folium

- Folium map object is a map centered on NASA Johnson Space Center at Houson, Texas
    - Red circle at NASA Johnson Space Center's coordinate with label showing its name
    - Red circles at each launch site coordinates with label showing launch site name The grouping of points in a cluster to display multiple and different information for the same coordinates
    - Markers to show successful and unsuccessful landings. Green for successful landing and Red for unsuccessful landing.
    - Markers to show distance between launch site to key locations

    - We understand the problem and the data. We can easily display all launch sites, the surroundings and successful and unsuccessful landings.

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/lab_jupyter_launch_site_location.jupyterlite%20(3).ipynb

Build a Dashboard with Plotly Dash

- Dashboard has dropdown, pie chart, rangeslider and scatter plot components
  - Dropdown allows a user to choose the launch site or all launch sites
  - Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component Rangeslider allows a user to select a payload mass in a fixed range
  - Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/plotly

14

Predictive Analysis (Classification)

- Data preparation
  - Load dataset
  - Normalize data
  - Split data into training and test sets.
- Model preparation
  - Selection of machine learning algorithms
  - Set parameters for each algorithm to GridSearchCV
  - Training GridSearchModel models with training dataset
- Model evaluation
  - Get best hyperparameters for each type of model
  - Compute accuracy for each model with test dataset
  - Plot Confusion Matrix
- Model comparison
  - Comparison of models according to their accuracy
  - The model with the best accuracy will be chosen (see Notebook for result)

https://github.com/GlexyMS/GLEXY--CAPSTONE-FALCON-9-PROJECT/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite%20(1).ipynb

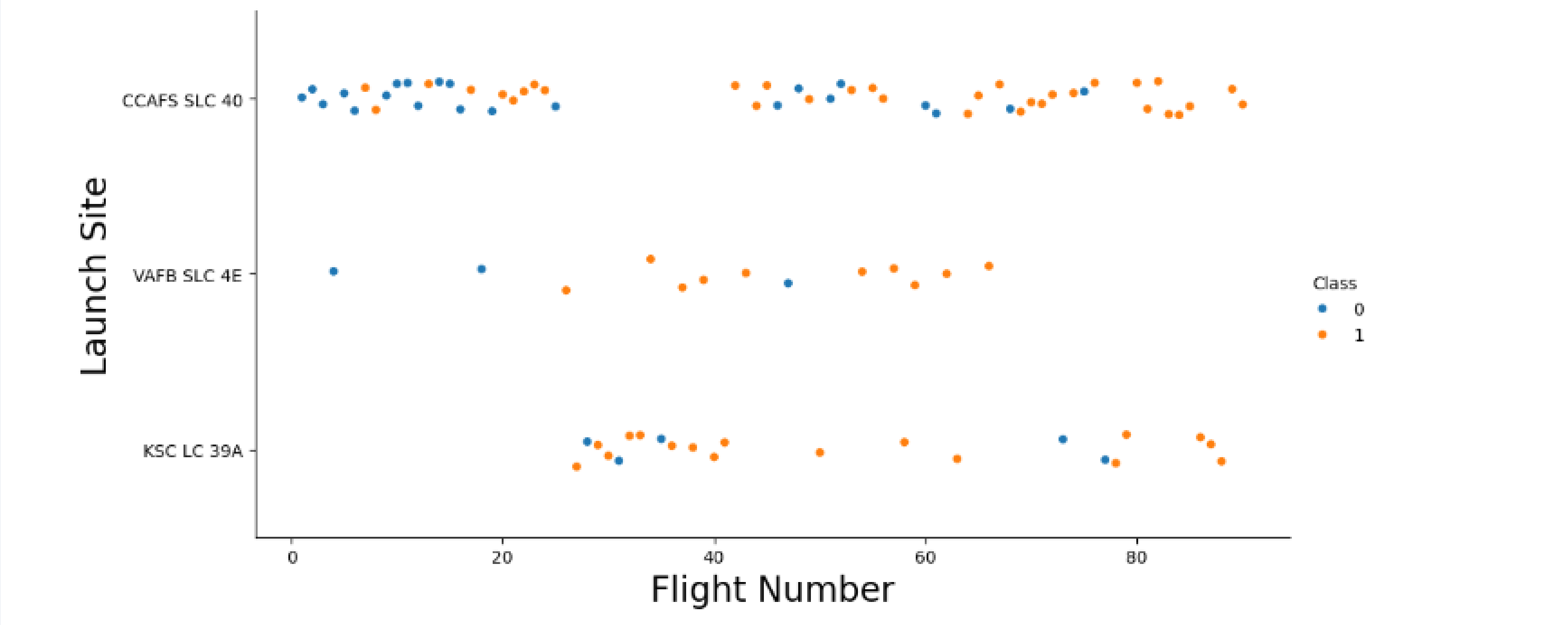
Results

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

Section 2

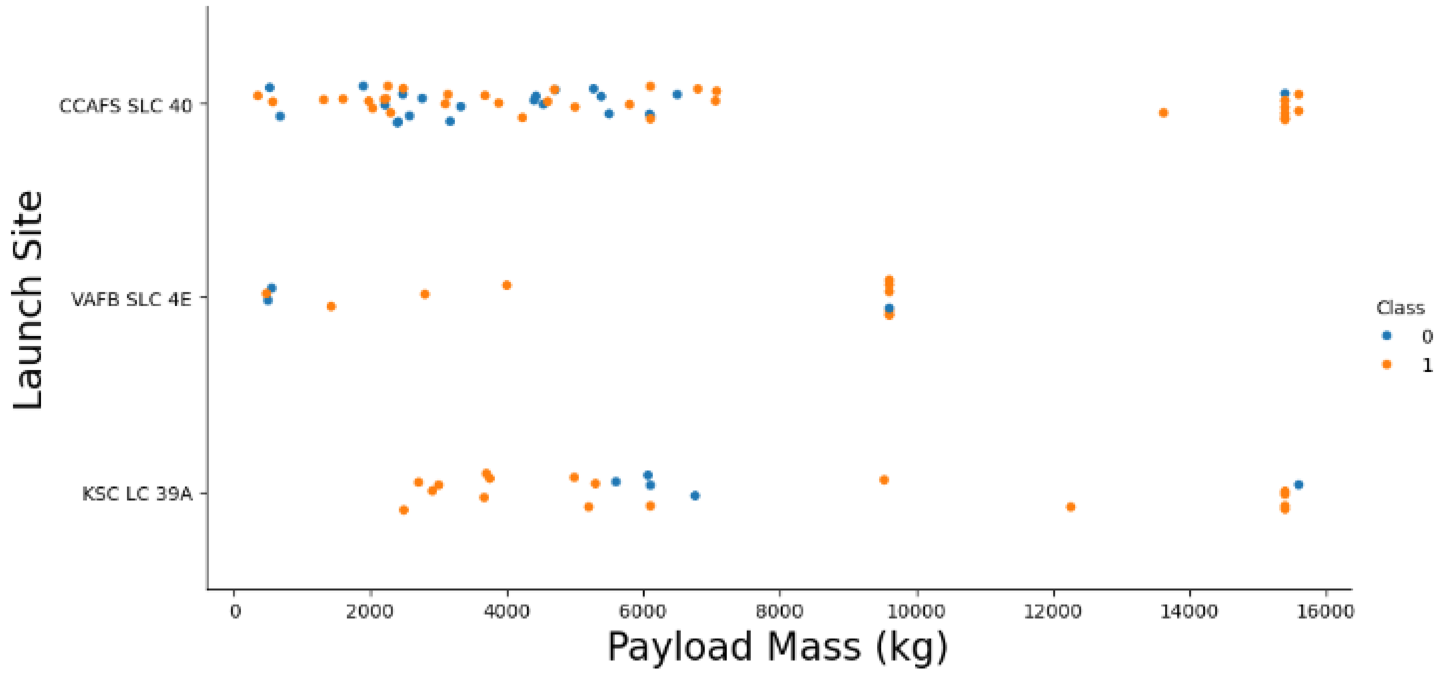
Insights drawn from EDA

Flight Number vs. Launch Site



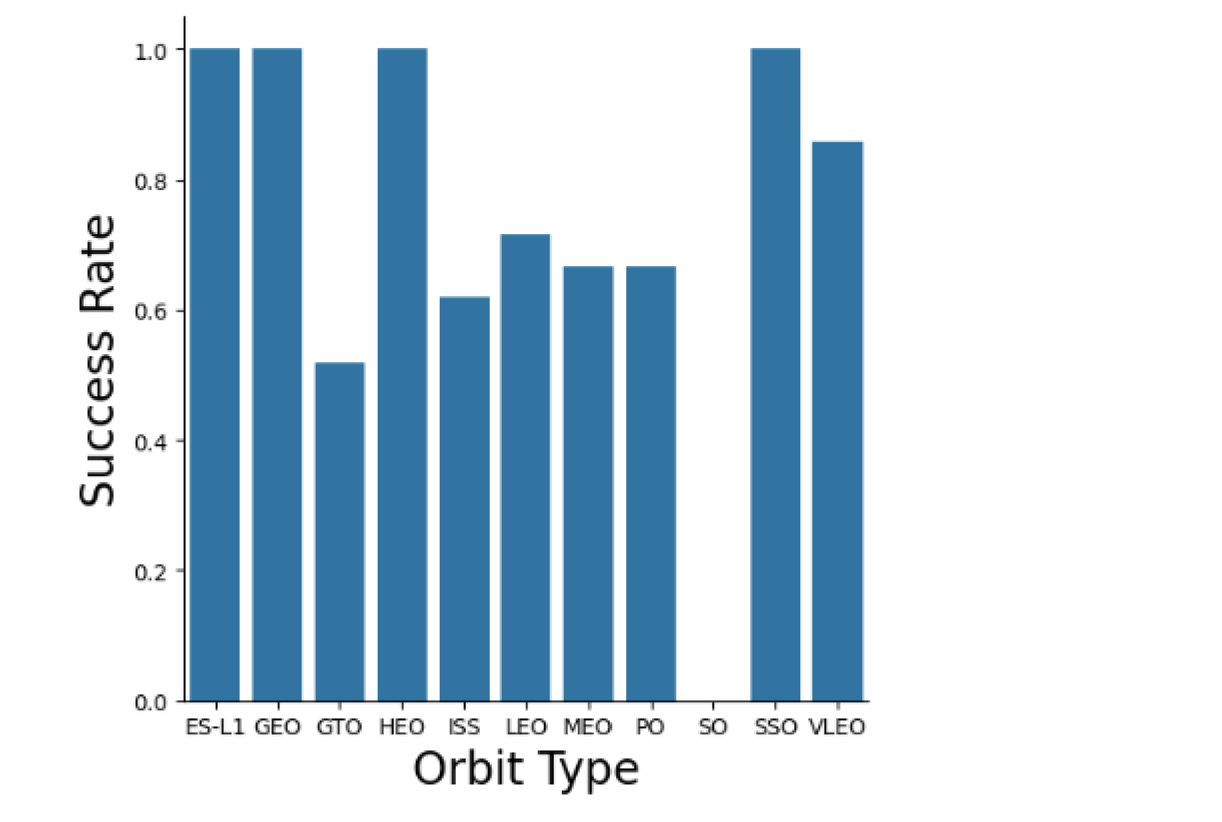
For each site, the success rate is increasing.

Payload vs. Launch Site



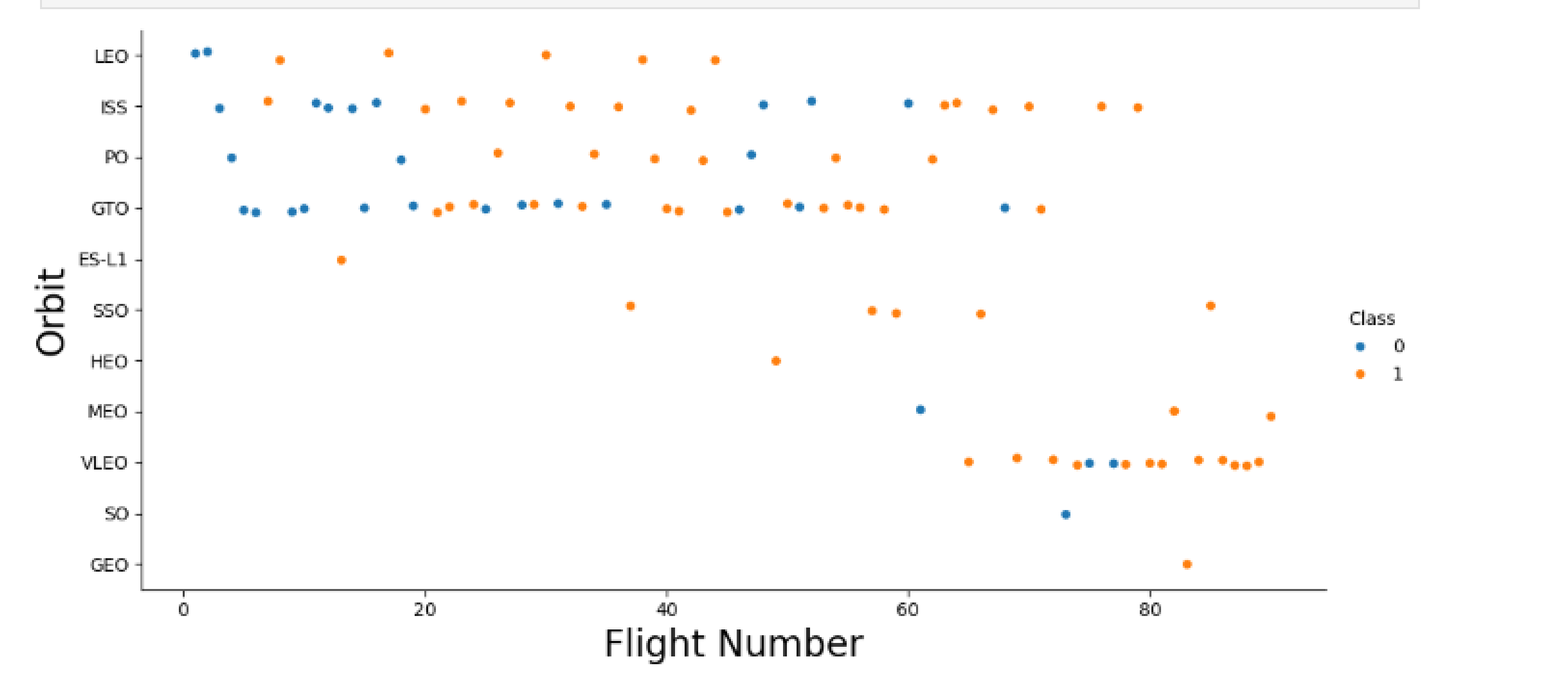
Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launchsite there are no rockets launched for heavypayload mass(greater than 10000).

Success Rate vs. Orbit Type



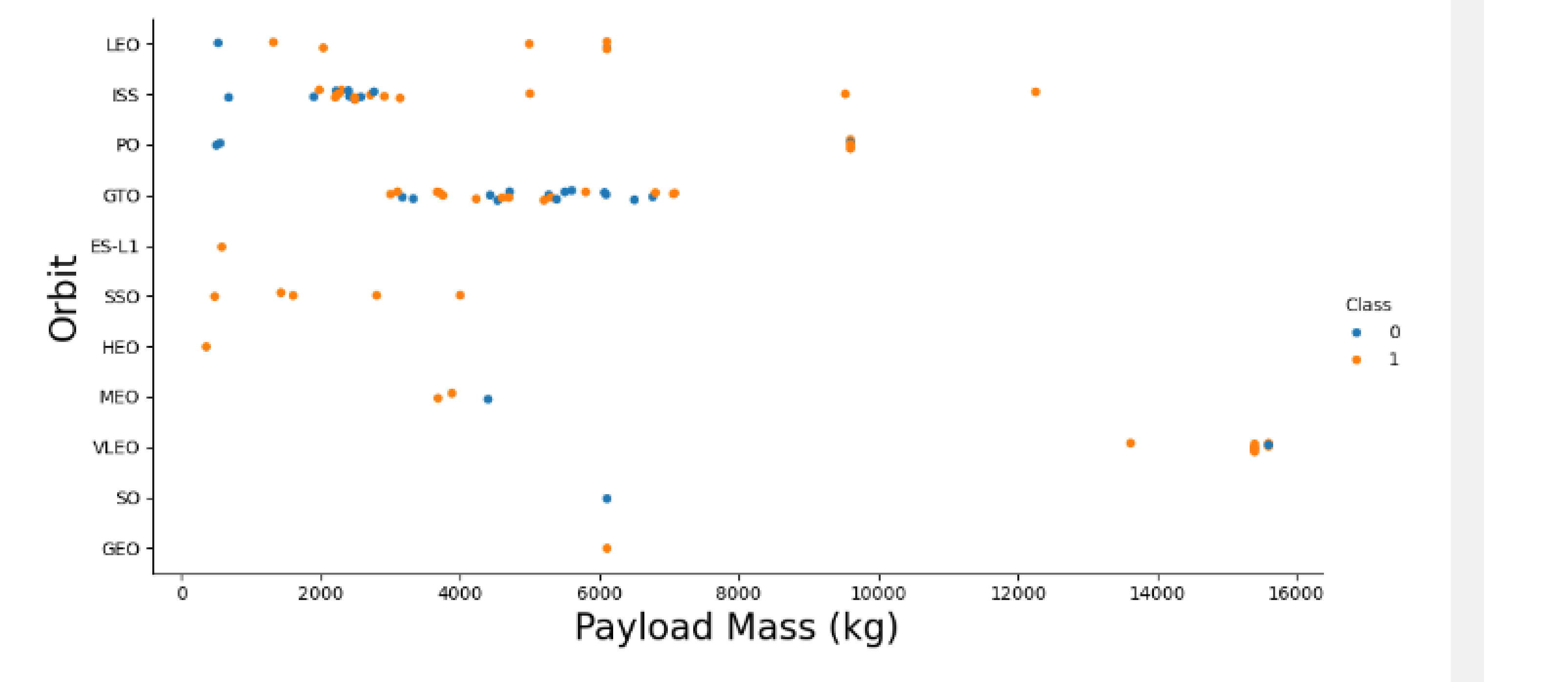
We note that ES-L1, GEO, HEO, SSO have the best success rate.

Flight Number vs. Orbit Type



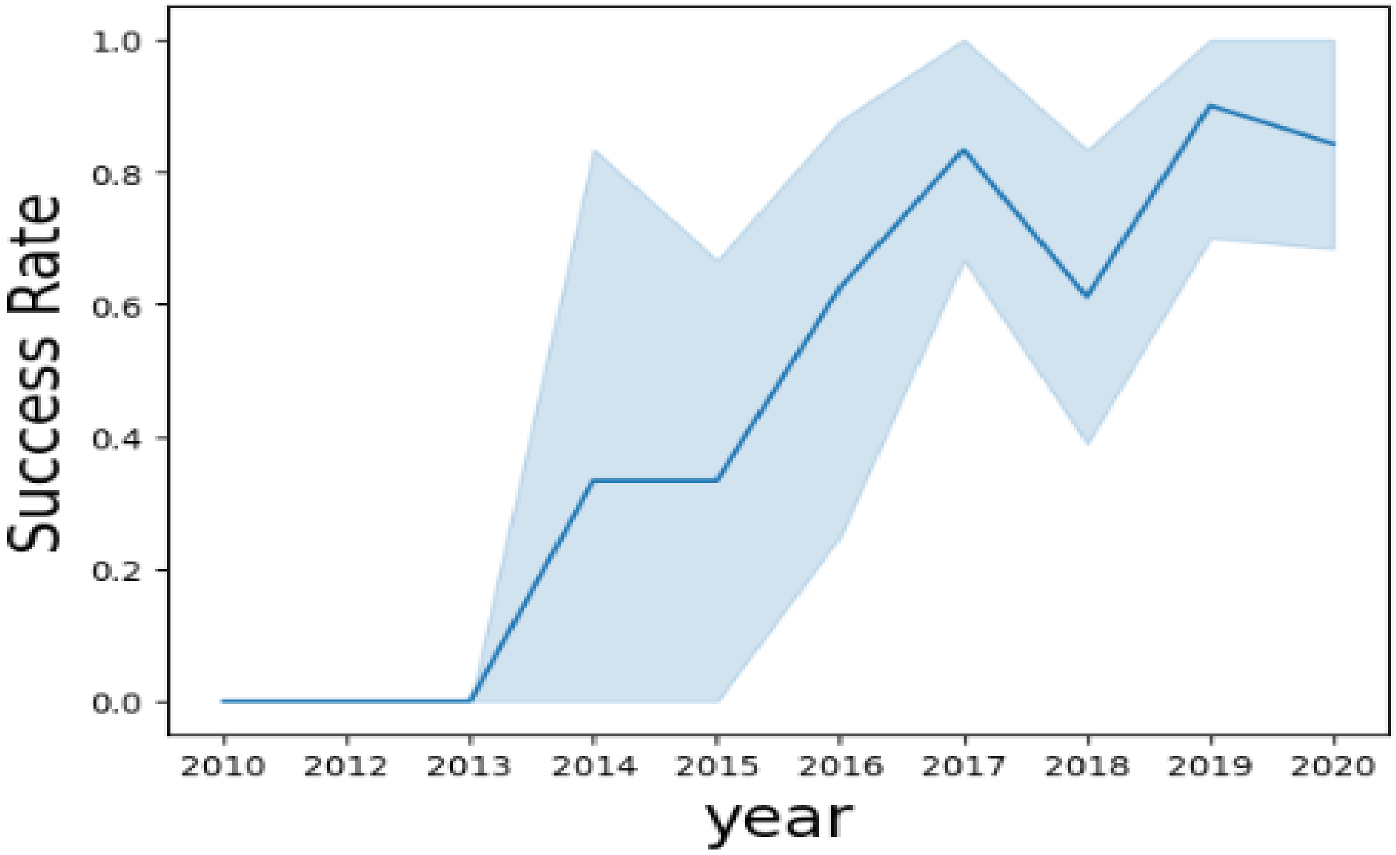
The success rate increases with the number of flights for the LEO orbit. For some orbits like GTO, there is no relation between the success rate and the number of flights.

Payload vs. Orbit Type



The weight of the payloads influence the success rate of the launches in certain orbits. For example, heavier payloads improve the success rate for the LEO orbit.

Launch Success Yearly Trend



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

All Launch Site Names

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

Task 1

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

In [15]:

```
%sql SELECT DISTINCT (LAUNCH_SITE) FROM SPACEXTBL;
```

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

Out[15]:

Launch_Site
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'

In [16]:
```
%sql SELECT * \
    FROM SPACEXTBL \
    WHERE LAUNCH_SITE LIKE'CCA%' LIMIT 5;
```

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

Out[16]:

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

SUM(PAYLOAD_MASS__KG_)
45596

Task 3

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

In [17]:

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

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

Out[17]:

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

AVG(PAYLOAD_MASS__KG_)

2928.4

Task 4

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

In [18]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) \
    FROM SPACEXTBL \
    WHERE BOOSTER_VERSION = 'F9 v1.1';
```

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

Out[18]:

AVG(PAYLOAD_MASS__KG_)
2928.4

This query returns the average of all payload masses where the booster version contains the substring F9 v1.1.

First Successful Ground Landing Date

MIN(DATE)
2015-12-22

Task 5

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

Hint:Use min function

In [20]:

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

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

Out[20]:

MIN(DATE)
2015-12-22

We select the oldest successful landing.
WHERE clause filters dataset to keep only records where landing was successful. MIN function to select the oldest date

Successful Drone Ship Landing with Payload between 4000 and 6000

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [23]:

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

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

Out[23]:

Payload
JCSAT-14
JCSAT-16
SES-10
SES-11 / EchoStar 105

The booster version where landing was successful oad mass is between 4000 and 6000 kg. The WHERE and AND filter the dataset.

Total Number of Successful and Failure Mission Outcomes

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

Task 7

List the total number of successful and failure mission outcomes

In [24]:
```
%sql SELECT MISSION_OUTCOME, COUNT(*) as total_number \
FROM SPACEXTBL \
GROUP BY MISSION_OUTCOME;
```

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

Out[24]:

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

With SELECT, we show the subqueries. The first subquery counts the successful mission. The second subquery counts the unsuccessful mission. The WHERE and LIKE clause filters mission outcome. The COUNT function counts records filtered.

Boosters Carried Maximum Payload

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

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

In [25]:

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

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

Out[25]:

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

Filter data by returning only the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass

2015 Launch Records

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Task 9

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.

In [29]:

```
%sql SELECT substr(Date, 6,2) as month, DATE,BOOSTER_VERSION, LAUNCH_SITE, [Landing_Outcome] \
FROM SPACEXTBL \
where [Landing_Outcome] = 'Failure (drone ship)' and substr(Date,0,5)='2015';
```

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

Out[29]:

month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

Substr to take month or year. Substr(DATE, 6, 2) shows month. Substr(DATE,0, 5) shows year.

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

Landing_Outcome	count_outcomes
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

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.

In [32]:

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

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

Out[32]:

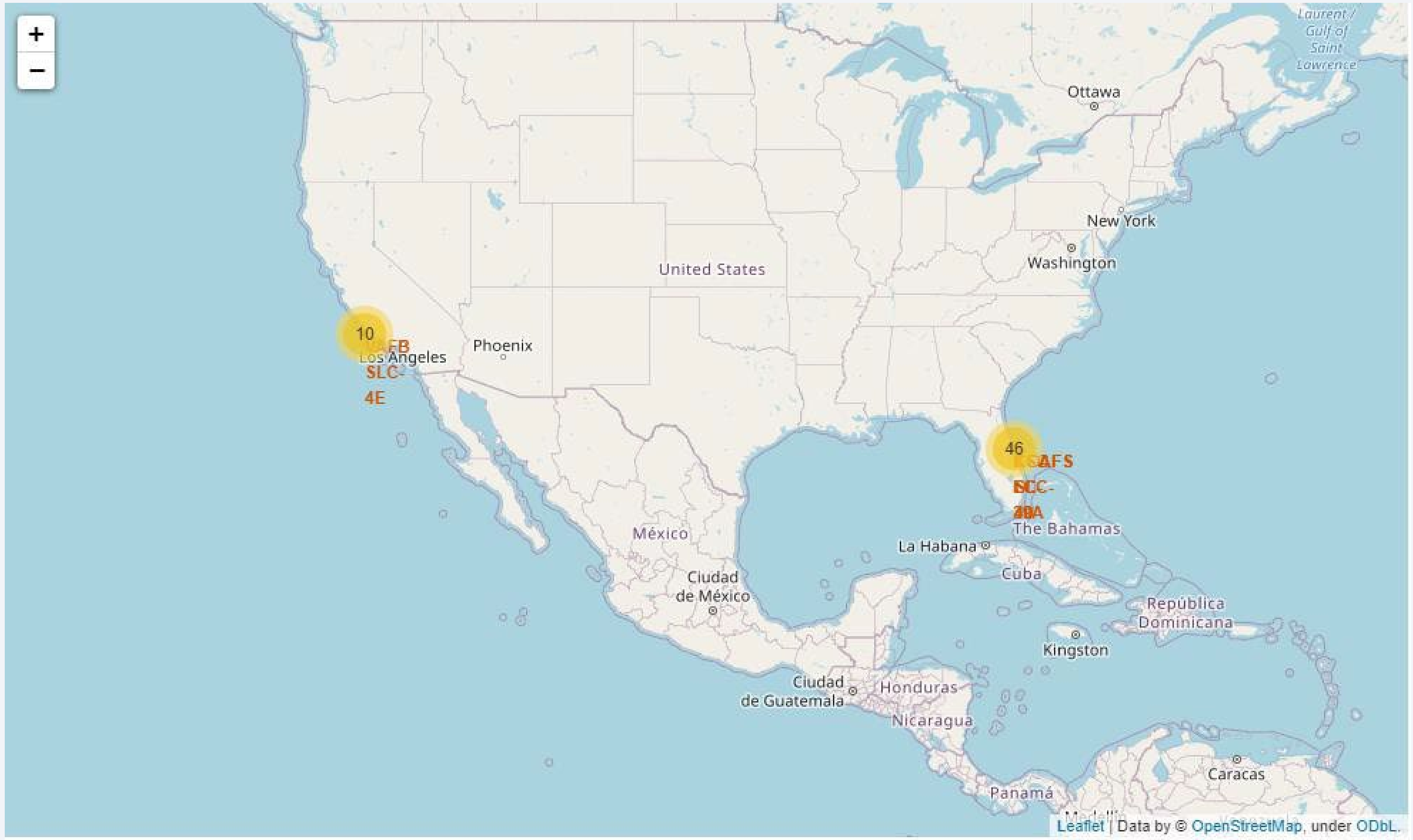
Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Landing outcome and count between 2010-06-04 and 2017-03-20. The GROUP BY clause groups results by landing outcome and ORDER BY COUNT DESC demonstrates results in decreasing order.

Section 3

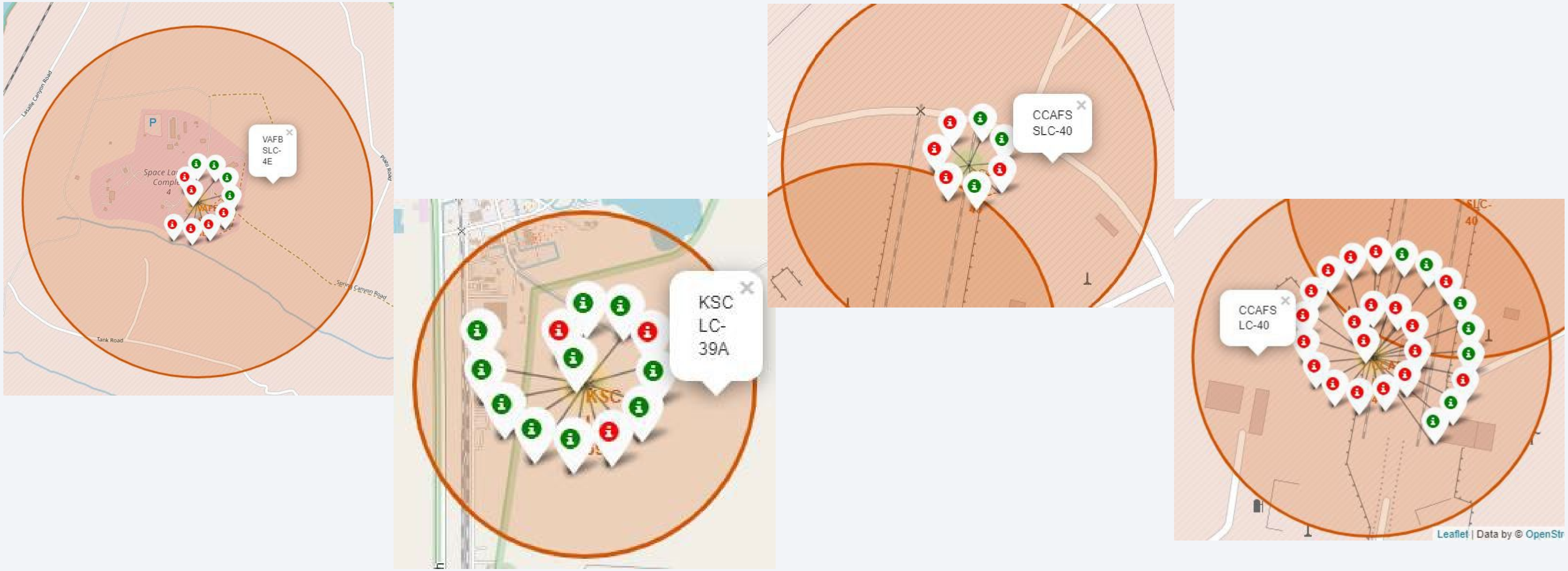
Launch Sites Proximities Analysis

Folium map

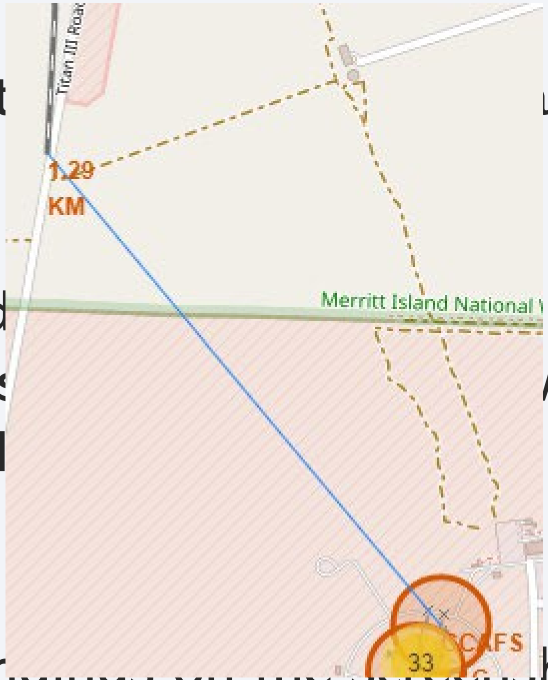


+
−
10
Los Angeles
SLC-
4E
Phoenix
United States
Ottawa
New York
Washington
46
The Bahamas
La Habana
Cuba
México
Ciudad
de México
República
Dominicana
Kingston
Ciudad
de Guatemala
Honduras
Nicaragua
Panamá
Caracas
Leaflet | Data by © OpenStreetMap, under ODbL.

Folium map – Color Labeled Markers

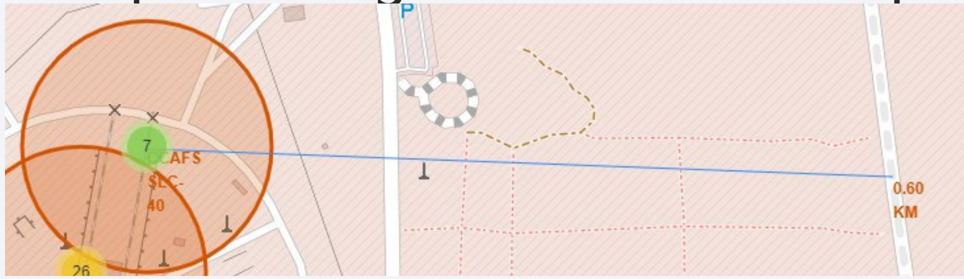


Folium Map – Proximities



> t ate title

- Explore the generated folium map and of



es s way
nd

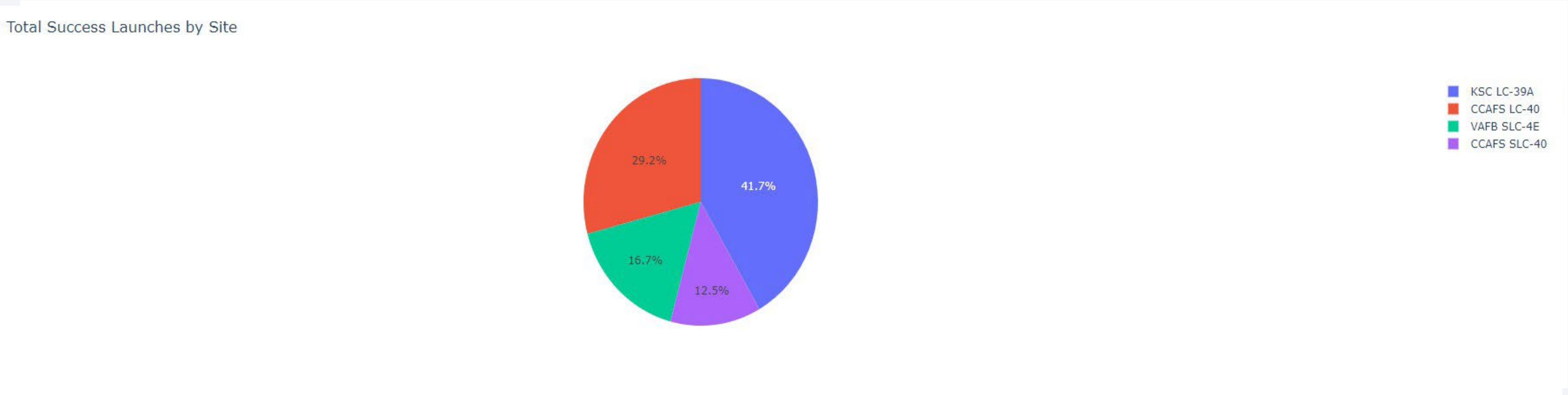
- Explain the important elements and fi shot

Launchsite in close proximity to railways ? Yes Is
Launchsite in close proximity to highways ? Yes Is in
Launchsite in close proximity to coastline ? Yes
Launchsite keeps certain distance away from cities ? No

Section 4

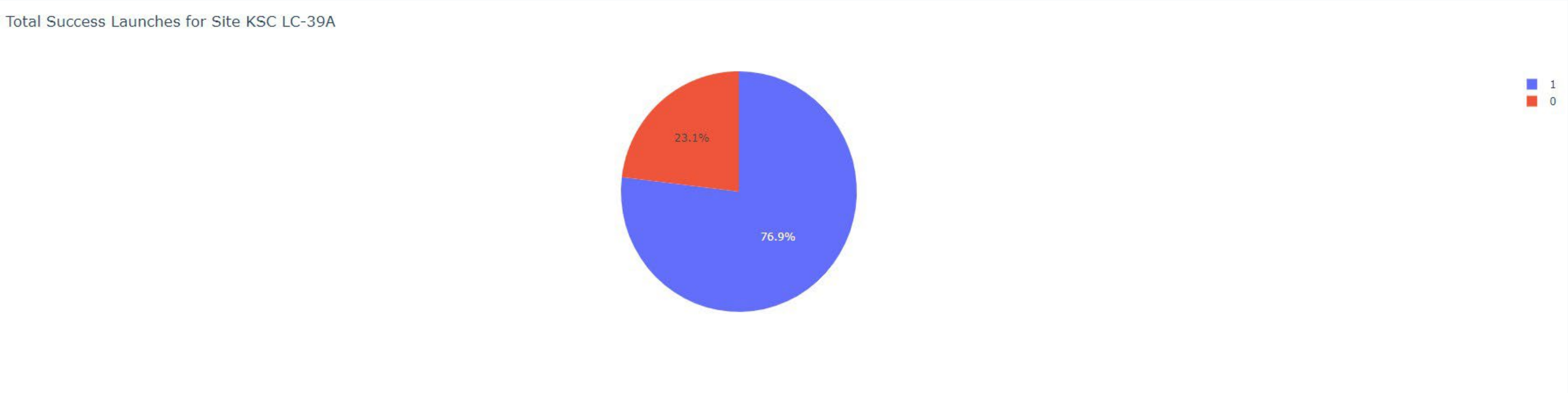
Build a Dashboard with Plotly Dash

Dashboard

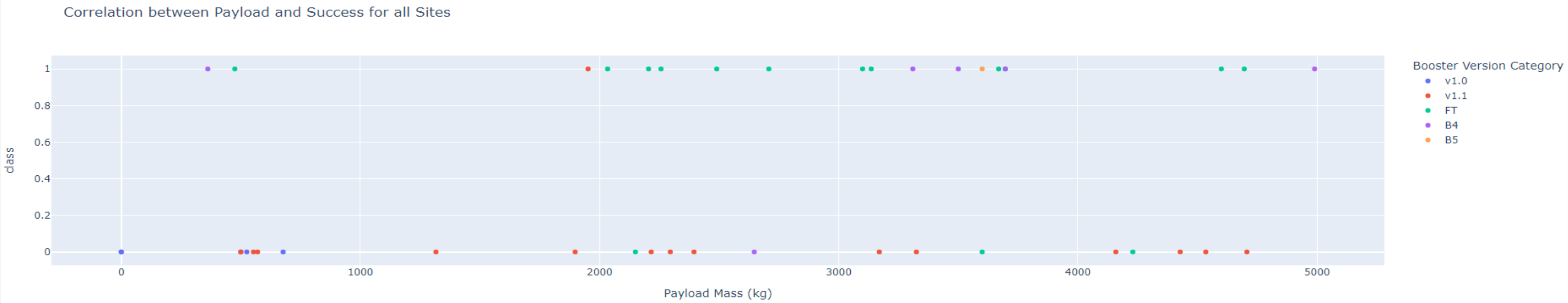


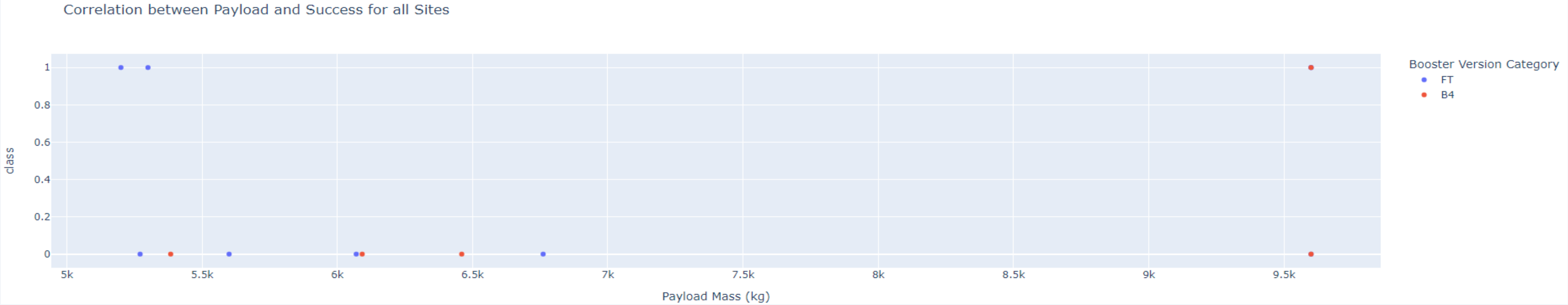
Total Success Launches by Site
29.2%
41.7%
16.7%
12.5%
KSC LC-39A
CCAFS LC-40
VAFB SLC-4E
CCAFS SLC-40

Dashboard – Success launches



Dashboard – Payload mass vs Outcome





Section 5

Predictive Analysis (Classification)

Classification Accuracy

TASK 12

Find the method performs best:

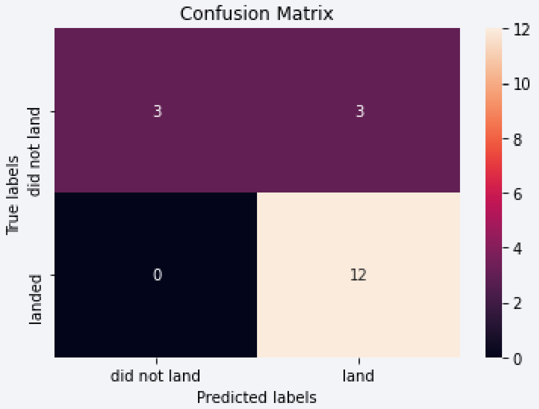
In [36]:

```
accuracy = [svm_cv_score, logreg_score, knn_cv_score, tree_cv_score]
accuracy = [i * 100 for i in accuracy]

method = ['Support Vector Machine', 'Logistic Regression', 'K Nearest Neighbour', 'Decision Tree']
models = {'ML Method':method, 'Accuracy Score (%)':accuracy}

ML_df = pd.DataFrame(models)
ML_df
```

Out[36]:

	ML Method	Accuracy Score (%)
0	Support Vector Machine	83.333333
1	Logistic Regression	83.333333
2	K Nearest Neighbour	83.333333
3	Decision Tree	83.333333

Confusion Matrix



As the test accuracy are all equal, the confusion matrices are also identical.

		Actual values	
		1	**0**
Predicted values	**1**	**TP**	**FP**
	0	**FN**	**TN**

Conclusions

- **Model Performance:** Similar performance on the test set
- **Equator:** Most of the launch sites are near the equator for an additional natural boost
- **Coast:** All the launch sites are close to the coast
- **Launch Success:** Increases over time
- **KSC LC-39A:** Has the highest success rate among launch sites. Has a 100% success rate for launches less than 5,500 kg
- **Orbits:** ES-L1, GEO, HEO, and SSO have a 100% success rate
- **Payload Mass:** Across all launch sites, the higher the payload mass (kg), the higher the success rate


Thank you!