IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Anastasia Livio
September 16, 2024

Outline

Executive Summary

- Data collection - SpaceX API
- Data collection - Web Scraping
- Data Wrangling
- Exploratory Data Analysis with SQL
- EDA with Visualization
- Interactive Visual Analytics with Folium
- Interactive Visual Analytics with Ploty Dash
- Predictive Analysis

Introduction

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. The goal of this project is to create a ML model to predict if the first stage will land successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:

  Using SpaceX API and Web Scraping from Wikipedia

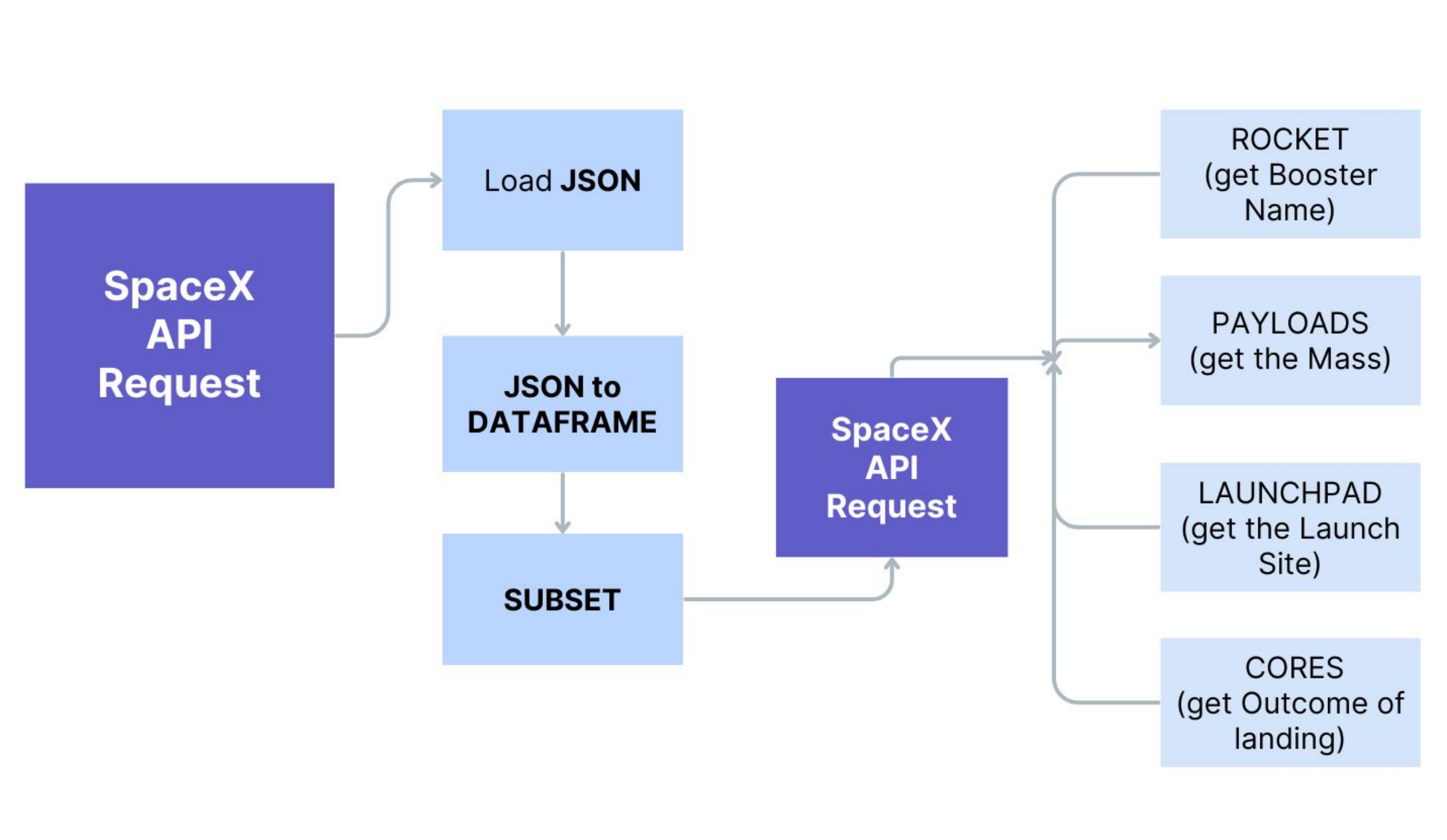
- Perform data wrangling

  Converting the outcomes into training labels with ‘1’ when the booster successfully landed and ‘0’.

- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

  Logistic Regression, SVM, Decision Tree and KNN.

Data Collection – SpaceX API

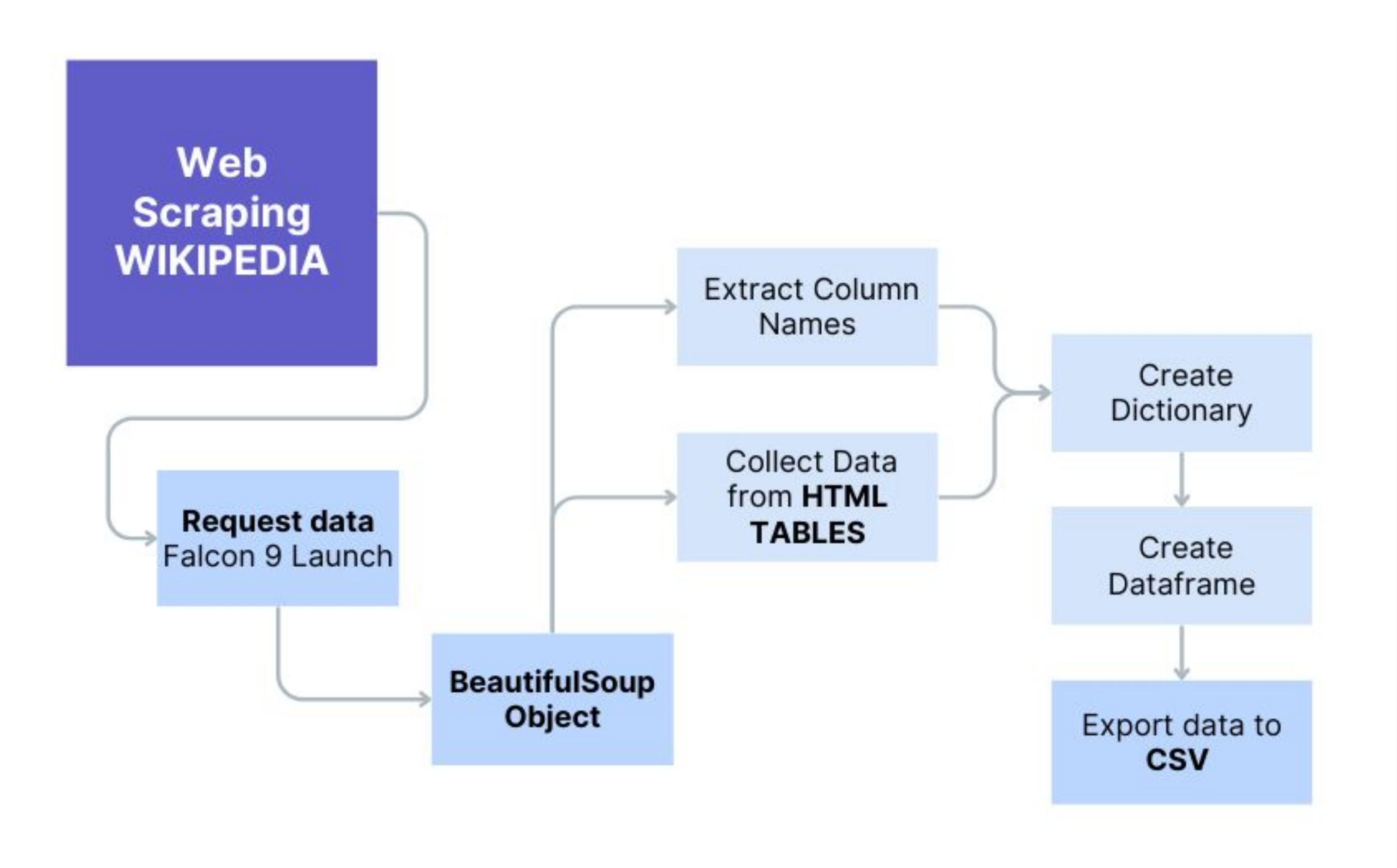


url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/API_data-collection.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/API_data-collection.ipynb

Data Collection - Scraping



url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/SpaceX_Web_Scraping.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/SpaceX_Web_Scraping.ipynb

Data Wrangling

- Load the dataset, identify which columns are numerical and categorical
- Calculate the number of launches on each site.

  Each launch aims to an dedicated orbit, and calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mision outcome of the orbits.
- Create a set of outcomes where the second stage did not land successfully: bad_outcomes
- Create a landing outcome label from Outcome column to generate new column ‘Class’ to represent the success o failure..

url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/Spacex-Data%20wrangling.ipynb)

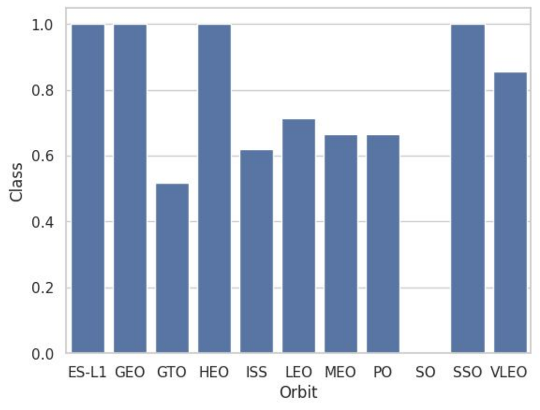
https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/Spacex-Data%20wrangling.ipynb

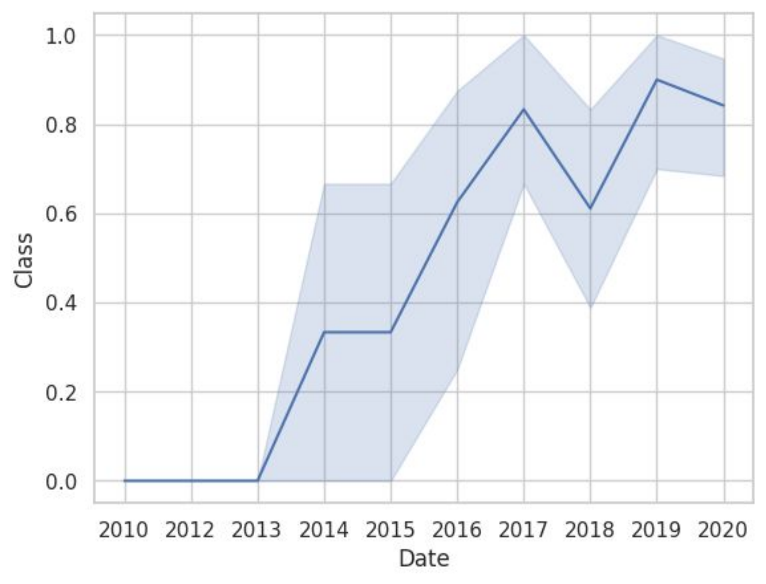
EDA with Data Visualization

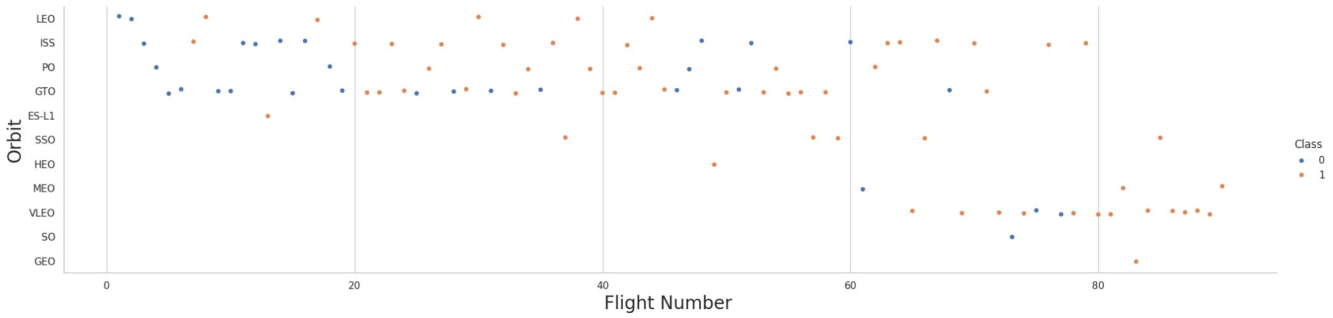
Scatter Plot for relationships between:
“Flight Number” and “Launch Site”
“Payload Mass” and “Launch Site”
“Flight Number” and “Orbit type”
“Payload Mass” and “Orbit type”

Bar Chart
“Success rate” and “Orbit type”

Line Chart to visualize the launch success yearly trend using a







url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/EDA_Data_Vizualization.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/EDA_Data_Vizualization.ipynb

EDA with SQL

- Display the names of unique launch sites
- Display 5 records where launch site begins with 'CCA'
- Display the Total payload mass carried by boosters launched by NASA (CRS)
- Find the Average payload mass carried by booster version F9 v1.1
- Find the Date of first successful landing on ground pad
- Names of boosters with success landing on drone ship and payload mass greater than 4,000 and less than 6,000
- Find the total number of successful and failed missions
- Display the names of booster versions which have carried the max payload
- Find failed landing outcomes on drone ship, booster version and launch site for the months in the year 2015
- Count of landing outcomes between 2010-06-04 and 2017-03-20 (desc).

url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/EDA_SQL_sqllite.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/EDA_SQL_sqllite.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: railway, coastline, city and highway.

The launch success rate may depend on many factors such as payload mass, orbit type, and so on. It may also depend on the location and proximities of a launch site, the initial position of rocket trajectories. Finding an optimal location for building a launch site certainly involves many factors and hopefully we could discover some of the factors by analyzing the existing launch site locations.

url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/Folium_launch_site_location.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/Folium_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Add a dropdown list to select a specific Launch Site or All.
- Add pie chart to show the total successful launches count for all sites and the success vs. failed counts.
- Add a range slider to select payload
- Add a Plotly Dash Slider to select the Payload Mass Range
- Add a scatter chart of payload mass vs. success rate of different booster versions. Shows the correlation between Payload and Launch Success

url

[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/spacex_DASH_app.py)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/spacex_DASH_app.py

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose

url

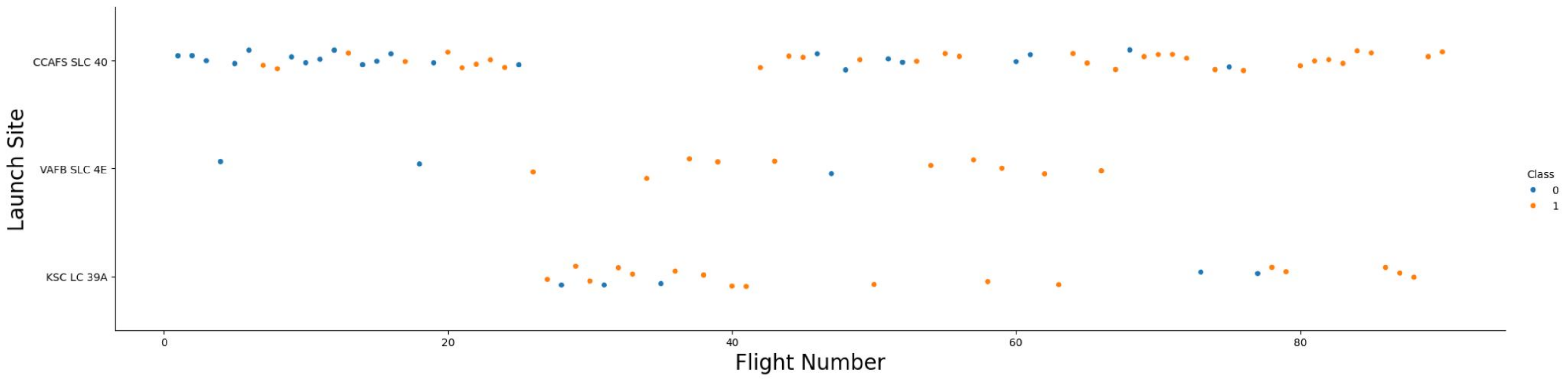
[GitHub notebook](https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb)

https://github.com/AnnLivio/IBM_Data_Science_Capstone/blob/main/SpaceX_Machine%20Learning%20Prediction.ipynb

Section 2

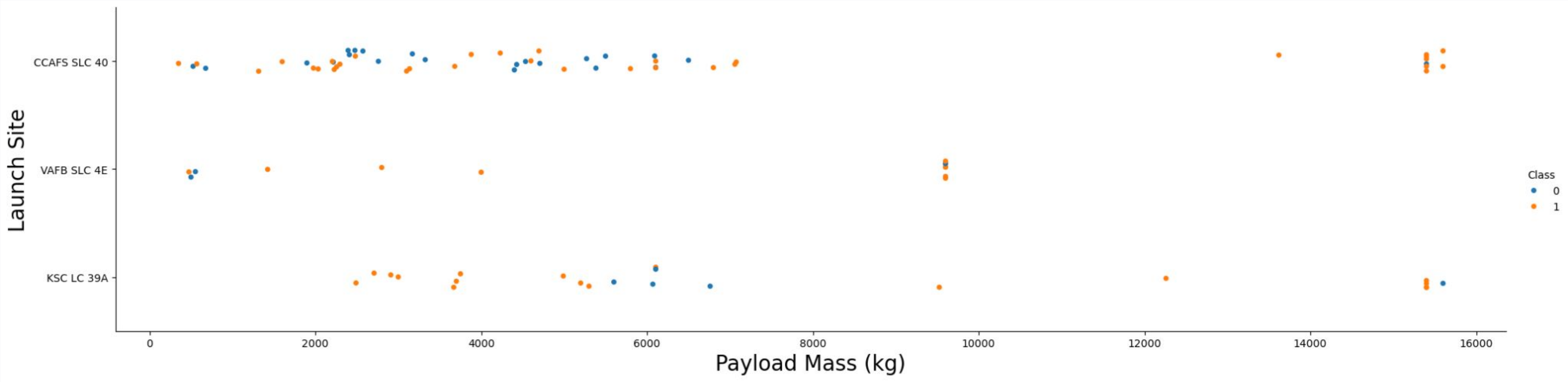
Insights drawn from EDA

Flight Number vs. Launch Site



- We can observe that earlier launches have lower success (blue point, class 0)
- The most recent launches have higher success rate.
- More than half launches were from CCAFS SLC 40 Launch Site.

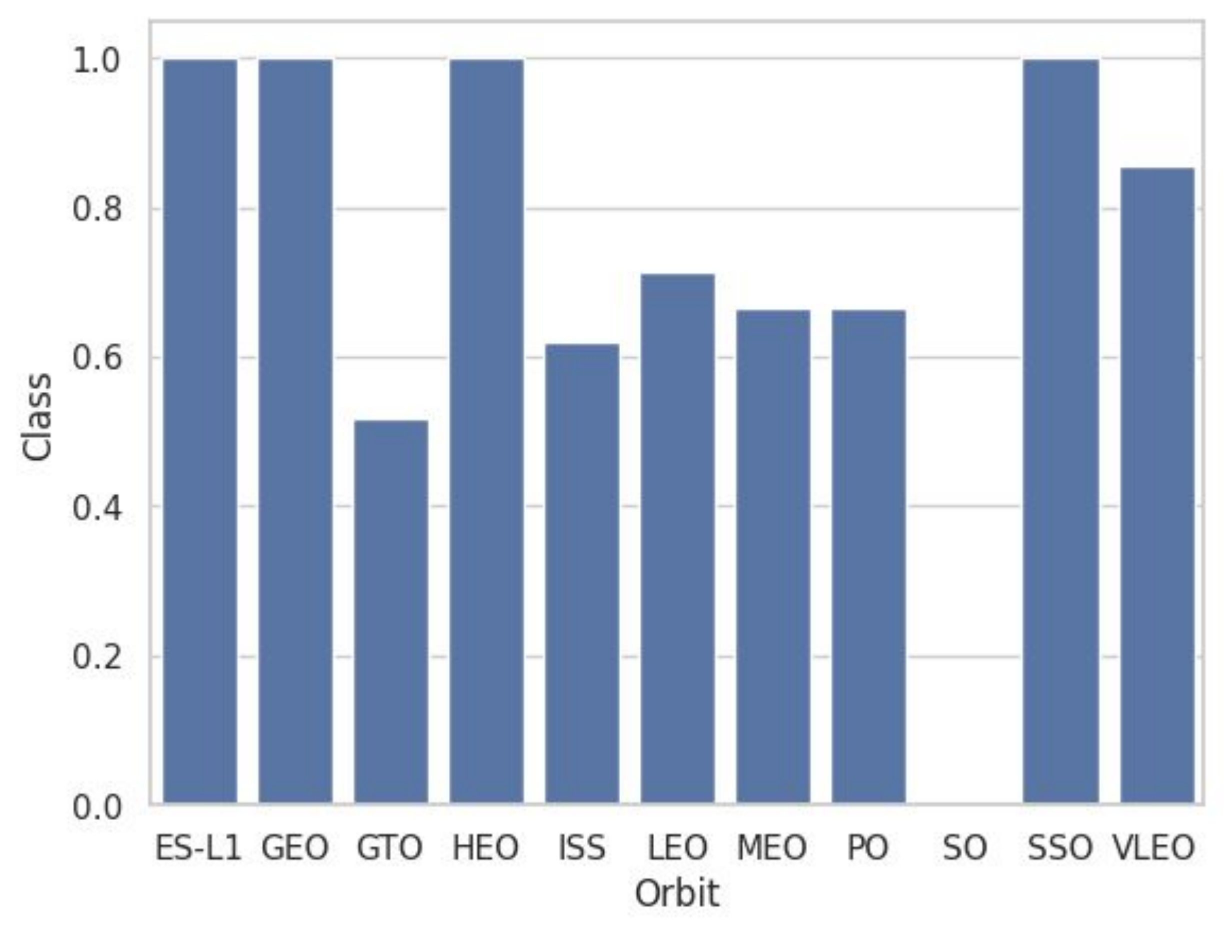
Payload vs. Launch Site



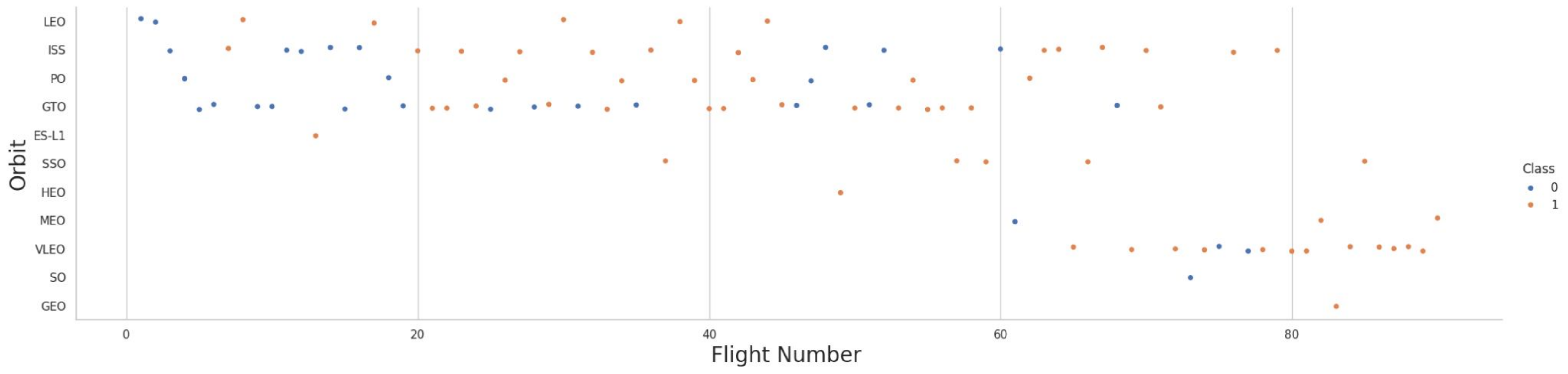
- The higher Payload Mass, higher success rate but, launches from KSC LC 39A have a 100% success rate when payload mass in under 6000 kg.
- VAFB-SLC has no launch over 10000 kg.

Success Rate vs. Orbit Type

- The Orbits ES-L1, GEO, HEO and SSO have the higher success rate.
- SO has a success rate of 0%.
- In the middle we have LEO, MEO, PO, VLEO, GTO and ISS with a success rate over 50%.

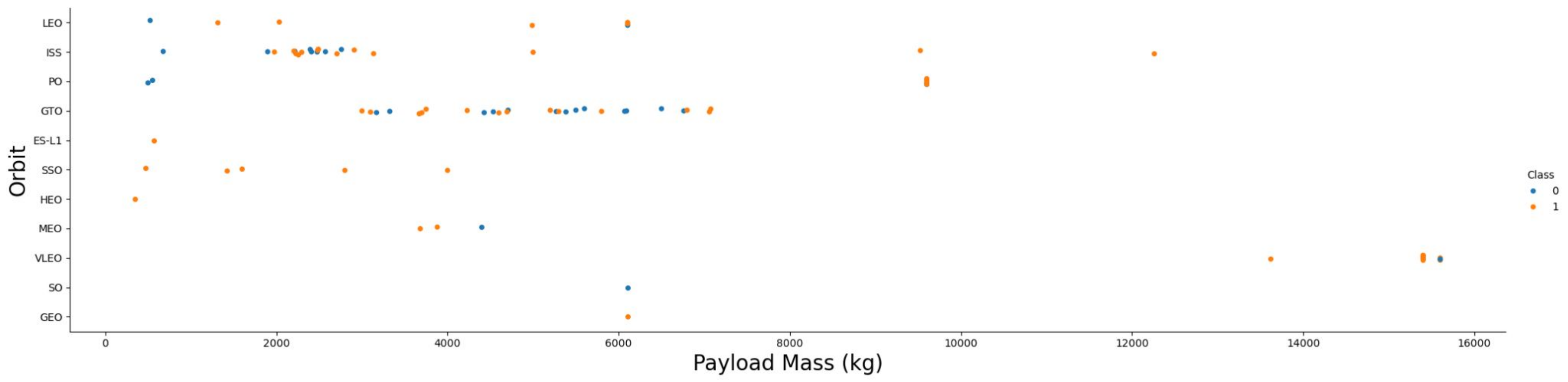


Flight Number vs. Orbit Type



- We can observe again the recent launches have better success rate.
- The most recent launches were landed on VLEO.

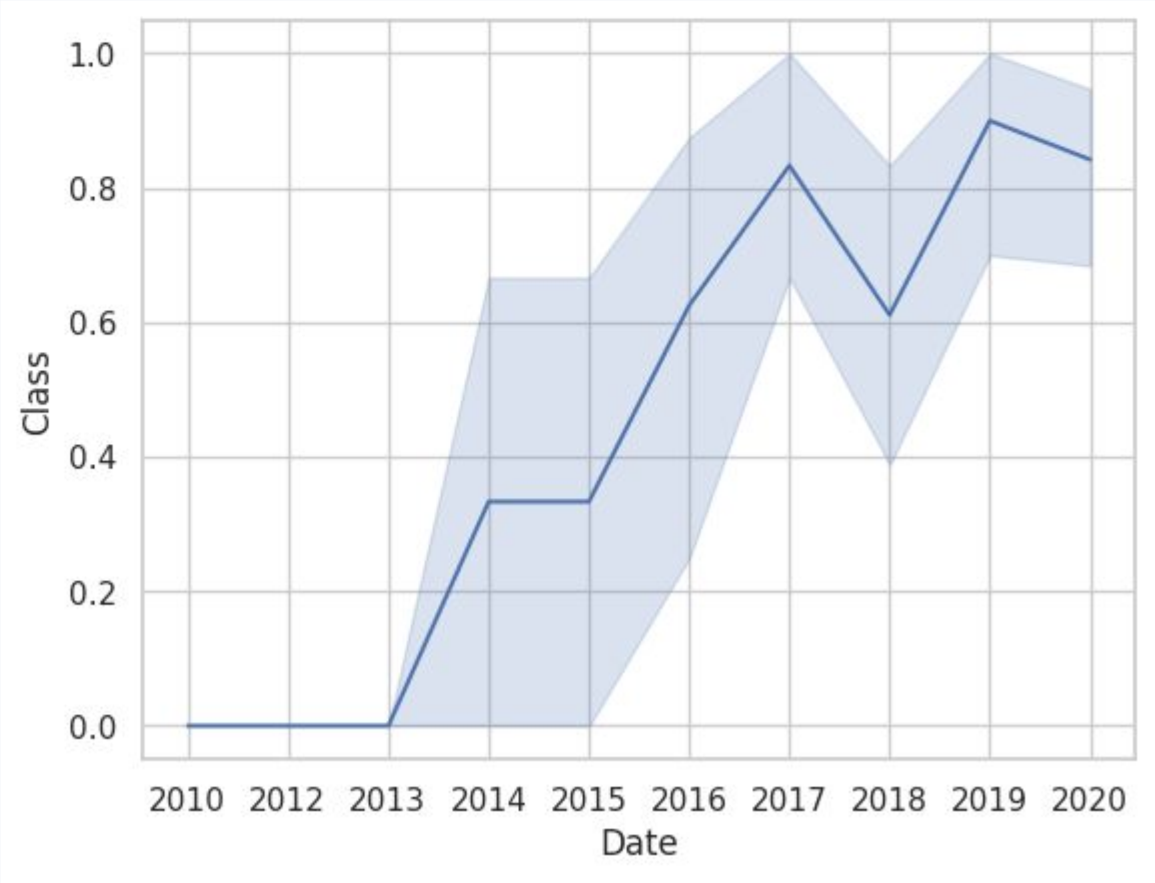
Payload vs. Orbit Type



- Heavy payload mass are better in LEO, PO, ISS.
- Lower payload mass are successfully with SSO, MEO and HEO.

Launch Success Yearly Trend

- The success rate improve after 2013.
- There is a slight decrease from 2017 to 2018.



All Launch Site Names

In [11]:

```
%sql SELECT distinct(Launch_Site) FROM SPACEXTABLE
```

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

Out[11]:

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

Launch Site Names Begin with 'CCA'

```
[16]: %sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
 * sqlite:///my_data1.db
Done.
```

[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

- The first 5 records where Launch Site name starts with ‘CCA’

Total Payload Mass

```
In [17]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Customer='NASA (CRS)'

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

Out[17]:

SUM(PAYLOAD_MASS__KG_)
45596

The total payload mass carried by boosters from NASA is 45,596 kg.

We use the SUM() function to calculate the total amount.

Average Payload Mass by F9 v1.1

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

```
In [18]:  %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version='F9 v1.1'

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

Out[18]:

AVG(PAYLOAD_MASS__KG_)
2928.4

The average payload mass carried by booster F9 v1.1 is 2928.4 kg

We use AVG() over the column PAYLOAD_MASS__KG_ and filter the result by Booster_Version='F9 v1.1'

First Successful Ground Landing Date

```
In [23]: %sql SELECT Date FROM SPACEXTABLE WHERE Landing_Outcome='Success (ground pad)' ORDER BY Date asc LIMIT 1

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

Out[23]:

Date
2015-12-22

The first successful landing outcome on ground pad was in 2015-12-22

We select the column Data for Landing_Outcome equal ‘Success (ground pad)’, we order the results by Date in Ascending order and limit the results to 1. We could use the function MIN() over the Date.

Successful Drone Ship Landing with Payload between 4000 and 6000

[18]:
```
%%sql
    SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
    WHERE Landing_Outcome='Success (drone ship)' AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000
```

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

[18]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Boosters with payload mass between 4000 and 6000, we use Landing_Outcome to check the success landing on drone ship.

Total Number of Successful and Failure Mission Outcomes

```
[28]: %%sql
          SELECT f.failure, s.success FROM
          (SELECT count(Landing_Outcome) AS failure FROM SPACEXTABLE WHERE Landing_OutCome LIKE 'Failure%') f,
          (SELECT count(Landing_Outcome) AS success FROM SPACEXTABLE WHERE Landing_OutCome LIKE 'Success%') s

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

[28]:

failure	success
10	61

There are 10 Failures and 61 success mission outcomes.

We use 2 subqueries. Query 1: count all ‘Failures’ in Landing_Outcome (Failure, Failure (drone ship) and Failure(parachute)). Query 2: count all ‘Success’ (Success, Success (drone ship), Success(ground pad))

Boosters Carried Maximum Payload

In this case, we use a subquery to get the maximum payload mass for each booster.

The MAX() function get those maximum values.

```
[30]: %%sql
      SELECT Booster_Version FROM SPACEXTABLE
      WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)

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

[30]:

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

```
[31]: %%sql SELECT substr(Date, 6,2), Landing_Outcome, Booster_Version, Launch_site
          FROM SPACEXTABLE
          WHERE Landing_Outcome='Failure (drone ship)' and
          substr(Date,0,5) = '2015'
```

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

[31]:

substr(Date, 6,2)	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.

There are 2 matches in this search.

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

[34]:
```
%%sql SELECT COUNT(*) as Rank, Landing_Outcome
    FROM SPACEXTABLE
    WHERE Landing_Outcome IN ('Failure (drone ship)','Success (ground pad)') and
    Date BETWEEN '2010-06-04' and '2017-03-20'
    GROUP BY Landing_Outcome
    ORDER BY Rank desc
```

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

[34]:

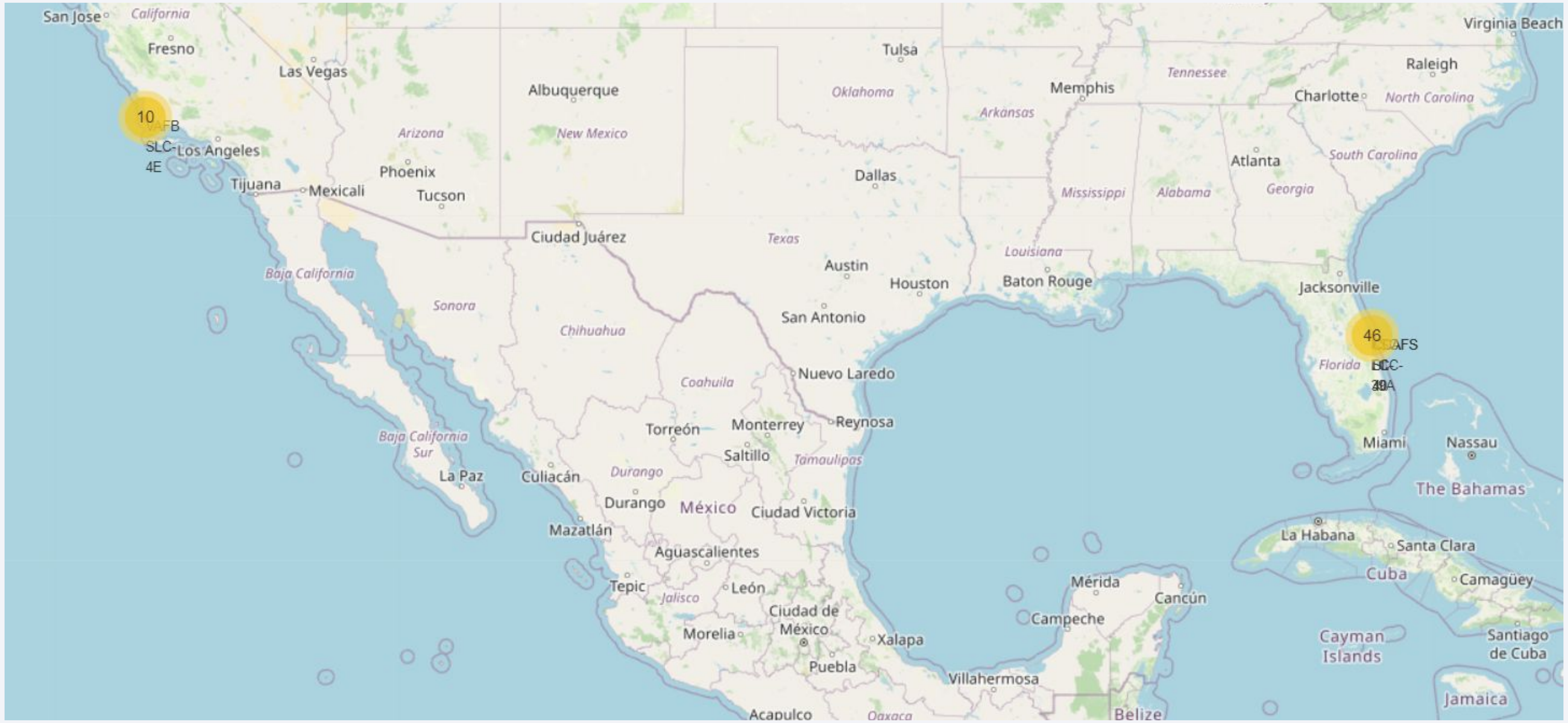
Rank	Landing_Outcome
5	Failure (drone ship)
3	Success (ground pad)

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.
There are 5 results for Failures and 3 for Success.

Section 3

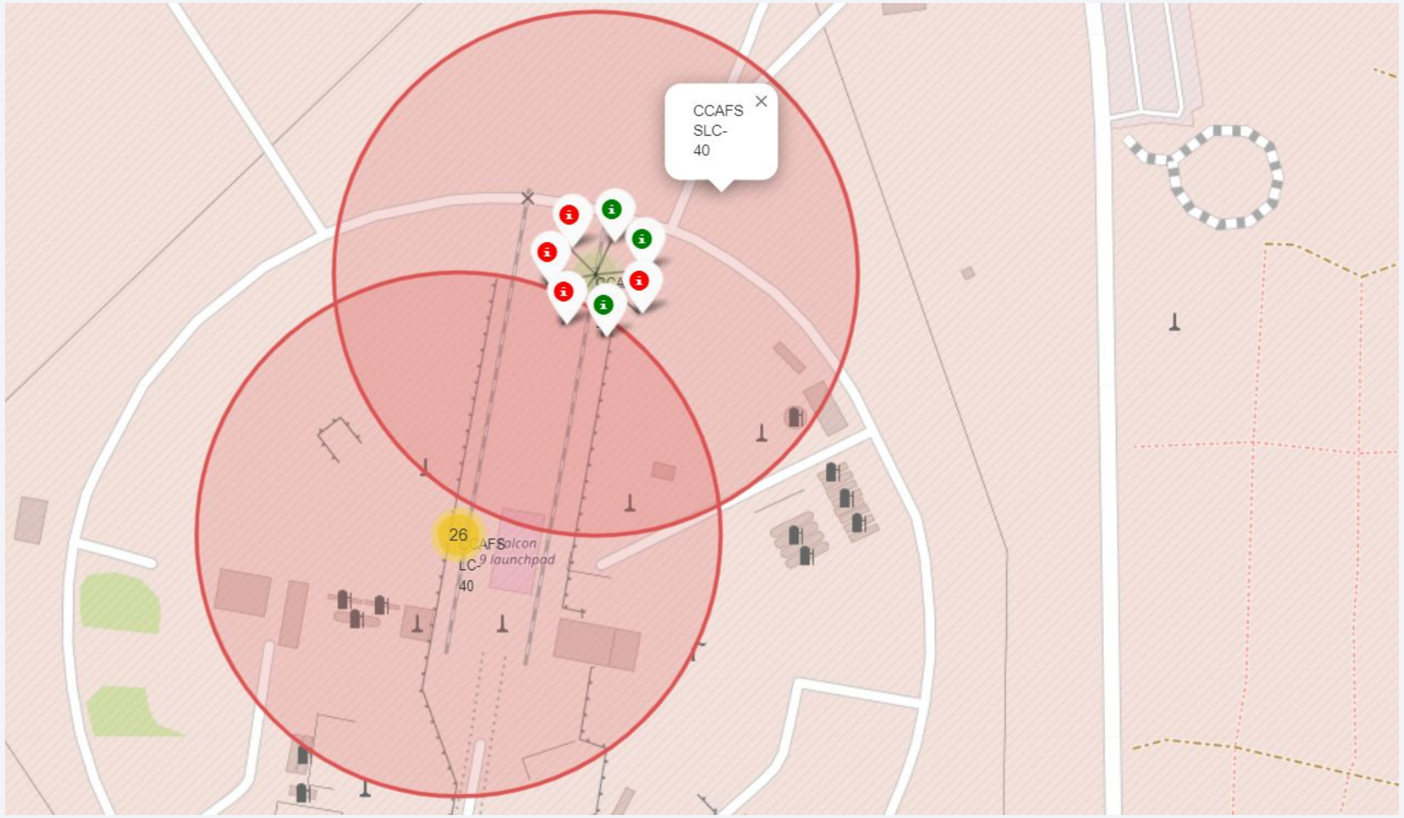
Launch Sites Proximities Analysis

Launch Sites Location



Rockets launched from sites closer to the equator are easier to launch and result in lower launch costs due to fuel savings.

Launch Outcomes for each site



- The green points represent the successful launches
- The reg points represent the failure launches.
- CCAFS SLC-40 has a success rate of 42.9%.

CCAFS SLC-40 and proximities

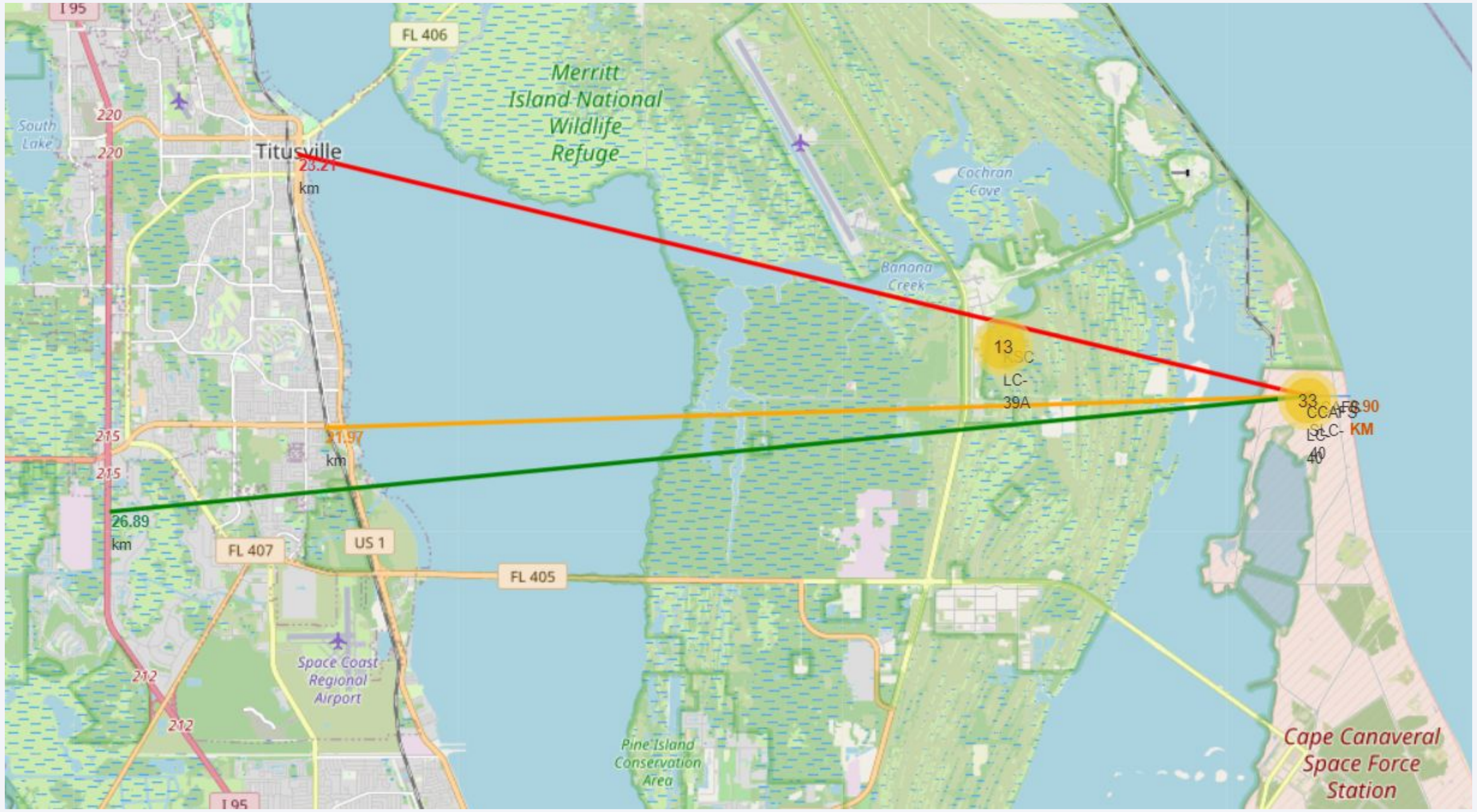
We use Folium map to calculate the distances between CCAFS SLC-40 launch site to its proximities

Railway = 21.97 km

Highway = 26.89 km

Coastline = 0.9 km

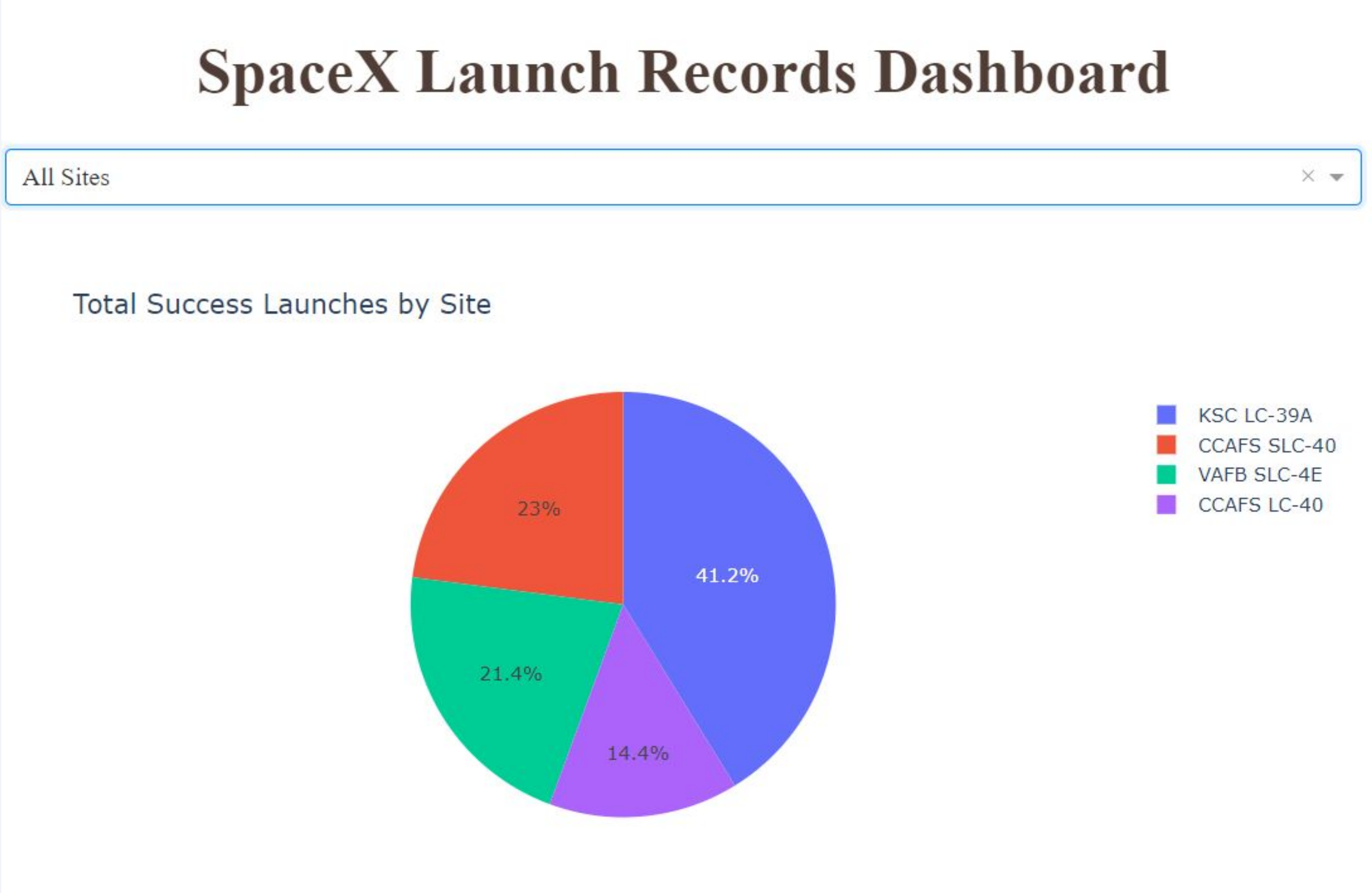
City = 23.21 km



Section 4

Build a Dashboard with Plotly Dash

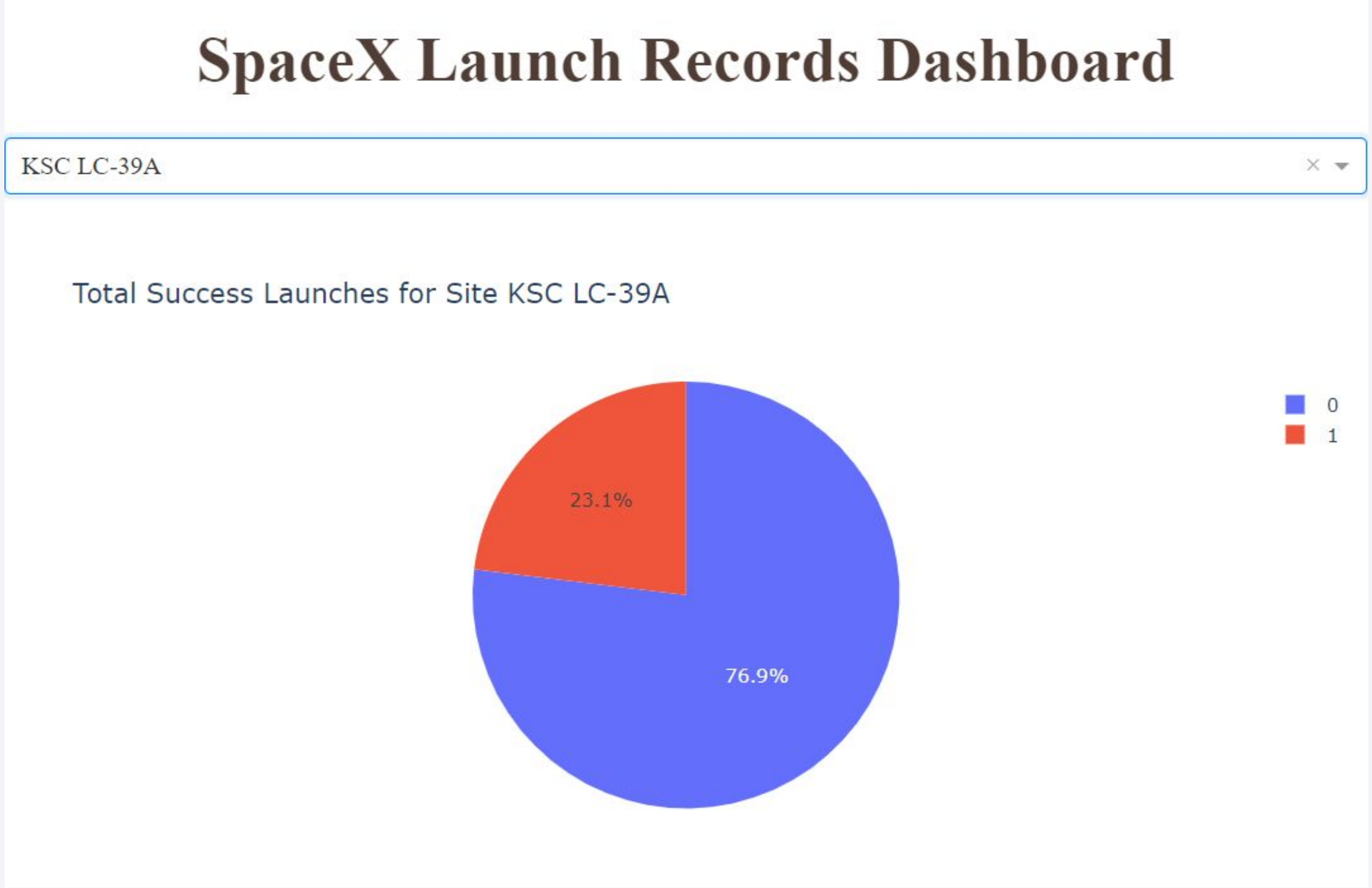
Success Launches by Site



Launch success count for all sites.

We find the **KSC LC-39A** has the best success rate with 41.2%.

Launch Site with highest success rate



KSC LC-39A has the highest success rate of 76.9%

Payload mass vs Success

Correlation between Payload Mass and Sucess group by Booster Version.

1 = success
0 = failure

Payload mass between **2k** and **5.5k** have **better success rate.**

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Out[45]:

	LogReg	SVM	Tree	KNN
F1_Score	0.888889	0.888889	0.916667	0.888889
Accuracy	0.833333	0.833333	0.888889	0.833333

The best model is Decision Tree with a F1 score of 0.92

The model with the best performance is the **Decision Tree** with **0.89 Accuracy and 0.92** F1 score.

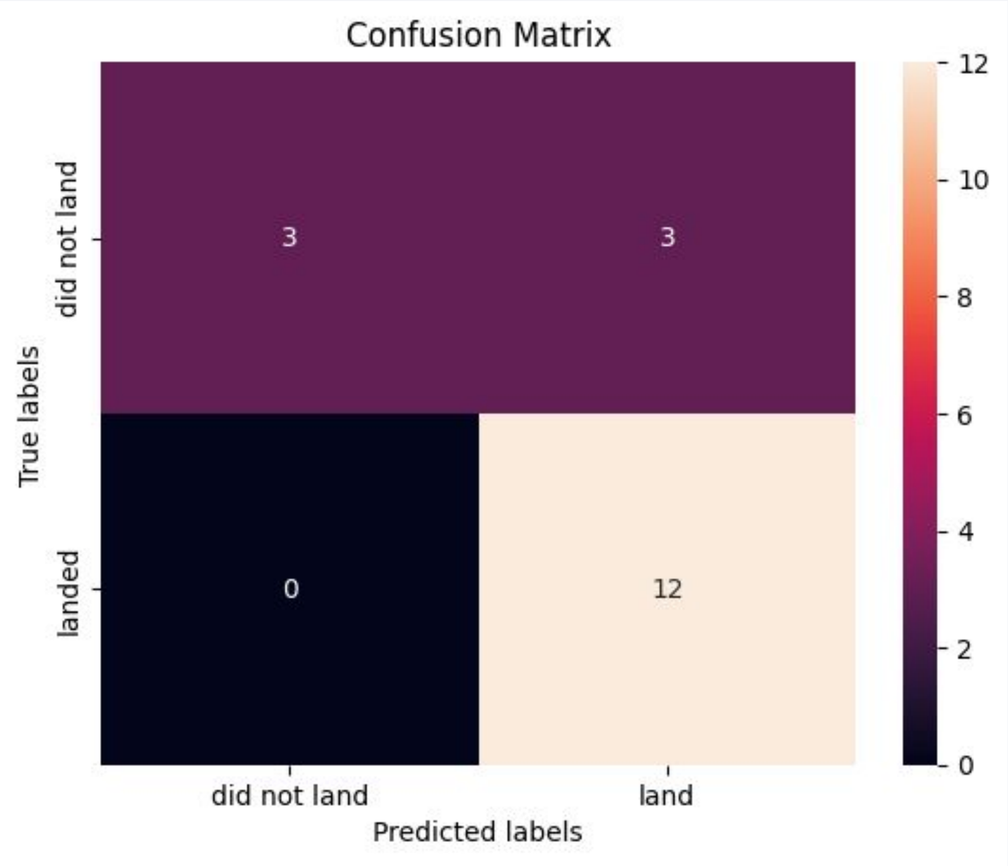
I know we are taking the accuracy as reference, I consider, in this case, the f1 score is the best score to make conclusions because it combine the precision and the recall.

The best params for **Decision Tree**

```
tree_cv.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 10,
 'max_features': 'sqrt',
 'min_samples_leaf': 4,
 'min_samples_split': 5,
 'splitter': 'best'}
```

Confusion Matrix



Decision Tree Model

True positives: 12
True negatives: 3
False positives: 3
False negatives: 0

Precision = TP / (TP + FP)
Recall = TP / (TP + FN)

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)
2 * (.8 * 1) / (.8 + 1) = .89

Accuracy = (TP + TN) / (TP + TN + FP + FN) = .833

Conclusions

- **Launch success** increase over the time, particularly **after 2013**.
- The launch sites closer to the **equator** represent a better choice to save cost.
- **KSC LC-39A** has the highest success rate and a 100% of success for launches with a payload mass between 2k and 5,5k kg.
- Models KNN, LR, SVM and **Decision Tree** perform similarly, the last one could be the best algorithm.

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!