IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Rafael Martínez Sevilla
27th February 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API and with Web Scraping.
  - Data Wrangling.
  - EDA with SQL.
  - Interactive visual analytics with Folium and Plotly Dash.
  - Predictive analysis (Classification Models).
- Summary of all results
  - Exploratory Data Analysis results.
  - Interactive visual analytics results.
  - Predictive analytics results.

Introduction

- Project background and context

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. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. Our mission is creating a machine learning model to predict if the Falcon 9 first stage will land successfully.

- Problems you want to find answers

- Identify the features that make a successfully landing of the first stage.
- Determinate the relationship between these features.
- Find out the conditions that guarantee a successful landing.

Section 1

Methodology

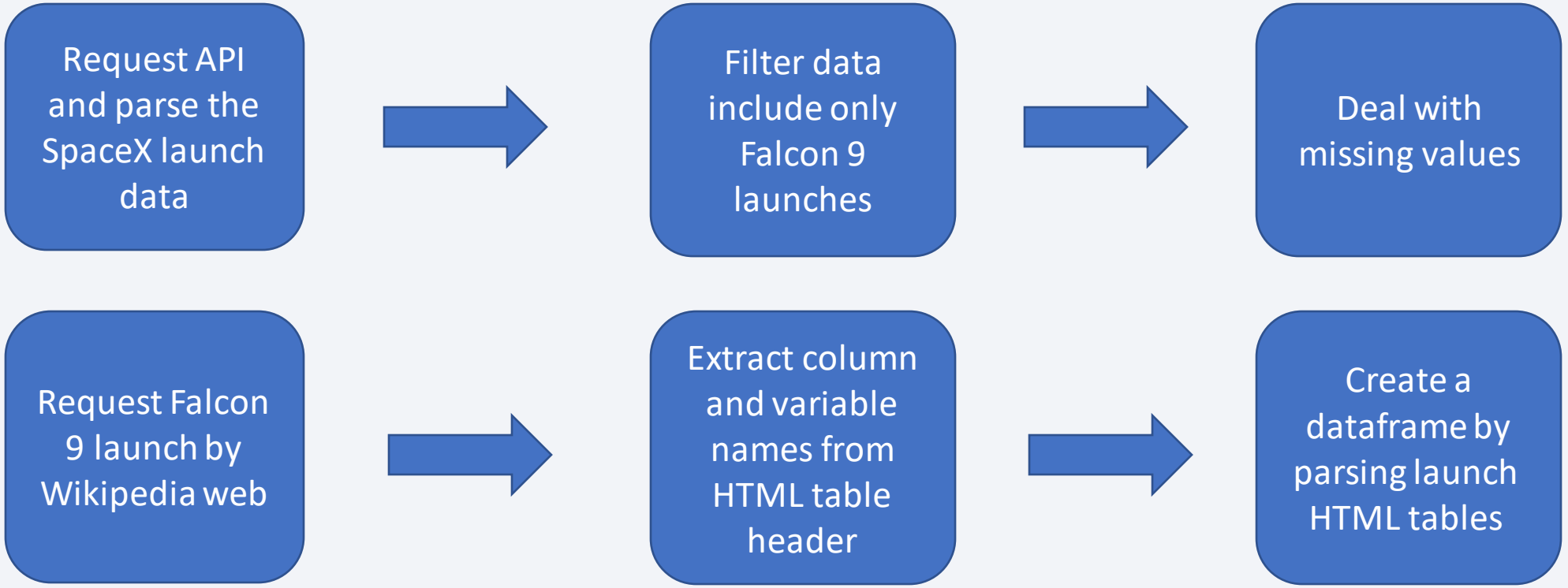
Methodology

Executive Summary

- Data collection methodology:
  - Using GET request in REST API.
  - Web scraping.
- Perform data wrangling
  - Using .fillna(), .value_counts() methods.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Tuned models using GridSearchCV.

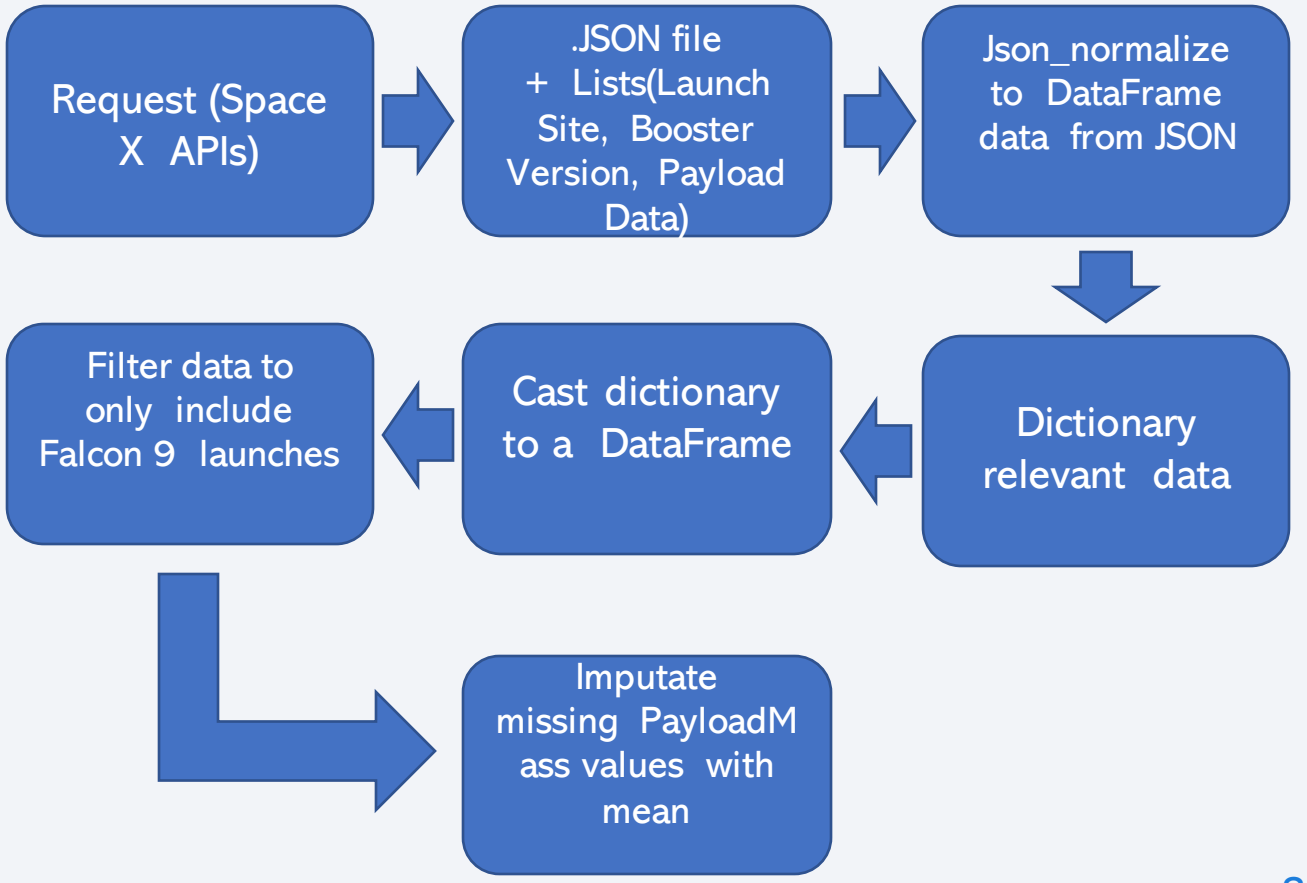
Data Collection

Data collection process involved a combination of API requests from Space X public API and web scraping data from a table in Space X's Wikipedia entry.



Data Collection – SpaceX API

The information obtain by the API is rocket, launches, payload information.

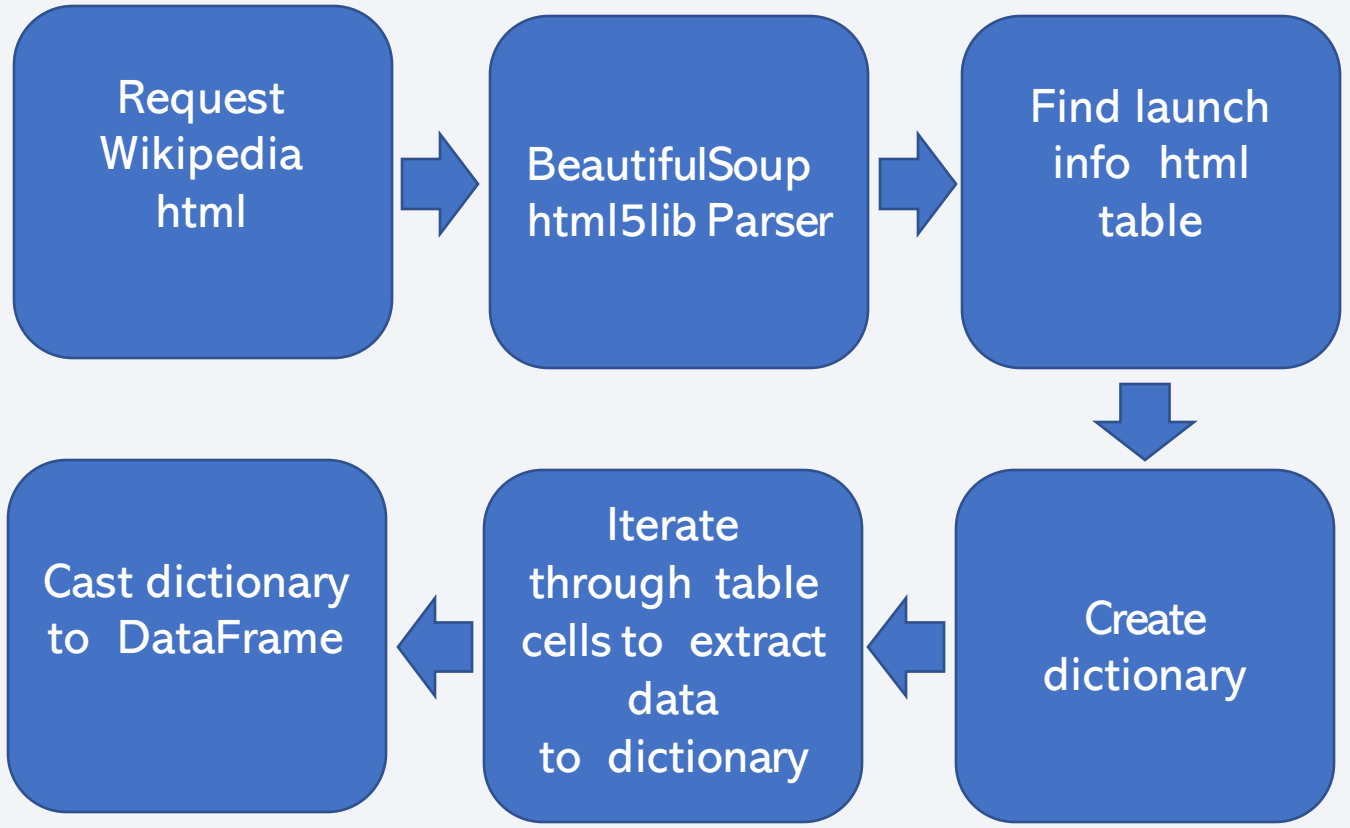


Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Collecting%20the%20Data.ipynb

Data Collection - Scraping

The information obtained by the web scraping of Wikipedia is launches, landing and payload information.



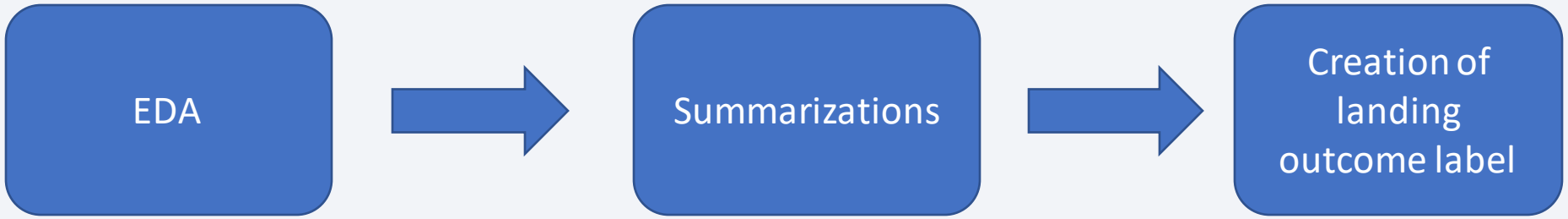
Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Data%20collection%20with%20Webscraping.ipynb

Data Wrangling

The procedure of data wrangling has the next steps:

- To perform exploratory data analysis and determine the training labels.
- To calculate the number of launches at each site, and the number and occurrence of each orbits.
- To create landing outcome label from outcome column and exporte the results to csv.



Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Data%20Wrangling.ipynb

EDA with Data Visualization

- **Scatter charts** are useful to observe relationships, or correlations, between two numeric variables. They were produced to visualize the relationships between:
- Flight Number and Launch Site.
- Payload and Launch Site.
- Orbit Type and Flight Number.
- Payload and Orbit Type.
- **Bar charts** are used to compare a numerical value to a categorical variable and visualize:
- Success Rate and Orbit Type.
- **Line charts** are generally used to show the change of a variable over time. They were produced to visualize the relationships between:
- Success Rate and Year

Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Exploring%20and%20preparing%20data%20with%20data%20visulization.ipynb

EDA with SQL

The SQL queries performed on the data set were used to:

Display the names of the unique launch 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 the average payload mass carried by booster version F9 v1.1.

List the date when the first successful landing outcome on a ground pad was achieved.

List the names of the boosters which had success on a drone ship and a payload mass between 4000 and 6000 kg.

List the total number of successful and failed mission outcomes.

List the names of the booster versions which have carried the maximum payload mass.

List the failed landing outcomes on drone ships, their booster versions, and launch site names for 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.

Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Exploratory%20Data%20Analysis%20(EDA)%20with%20SQL.ipynb

Build an Interactive Map with Folium

Markers, circles, lines and markers clusters were used with Folium maps:

- Markers indicate points like launch sites.
- Circles indicate highlighted areas around specific coordinates like NASA Johnson Space Center.
- Marker clusters indicate groups of events in each coordinate like launches in launche sites.
- Lines are used to indicate distances between two coordinates.

This allow us to understand why launch sites may be located where they are. Also visualizes successful landing relative to location.

Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Launch%20sites%20locations%20analysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

Pie chart showing the total successful launches per site

- This makes it clear to see which sites are most successful
- The chart could also be filtered to see the success/failure ratio for an individual site

Scatter graph to show the correlation between outcome (success or not) and payload mass.

- This could be filtered by ranges of payload masses
- It could also be filtered by booster version

Source code:

https://github.com/RafMartSev/Final_Course_Capstone/blob/main/spacex_dash_app.py

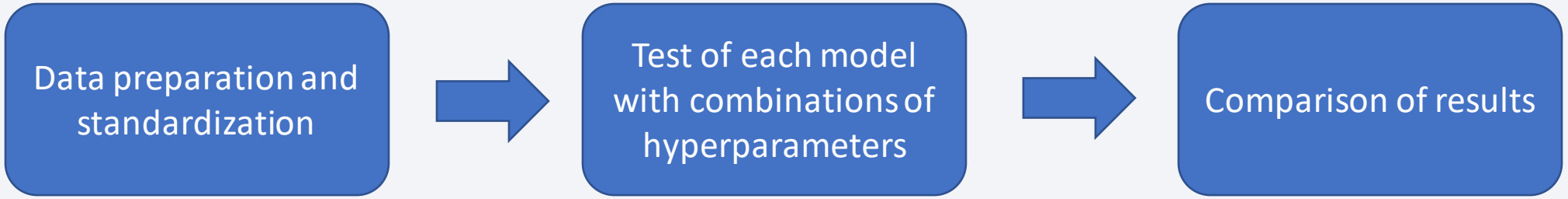
Predictive Analysis (Classification)

To load the data using NumPy and pandas, transformed the data, split our data into training and testing.

To build different machine learning models and tune different hyperparameters using GridSearchCV.

To use accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.

To found the best performing classification model.



Source code:

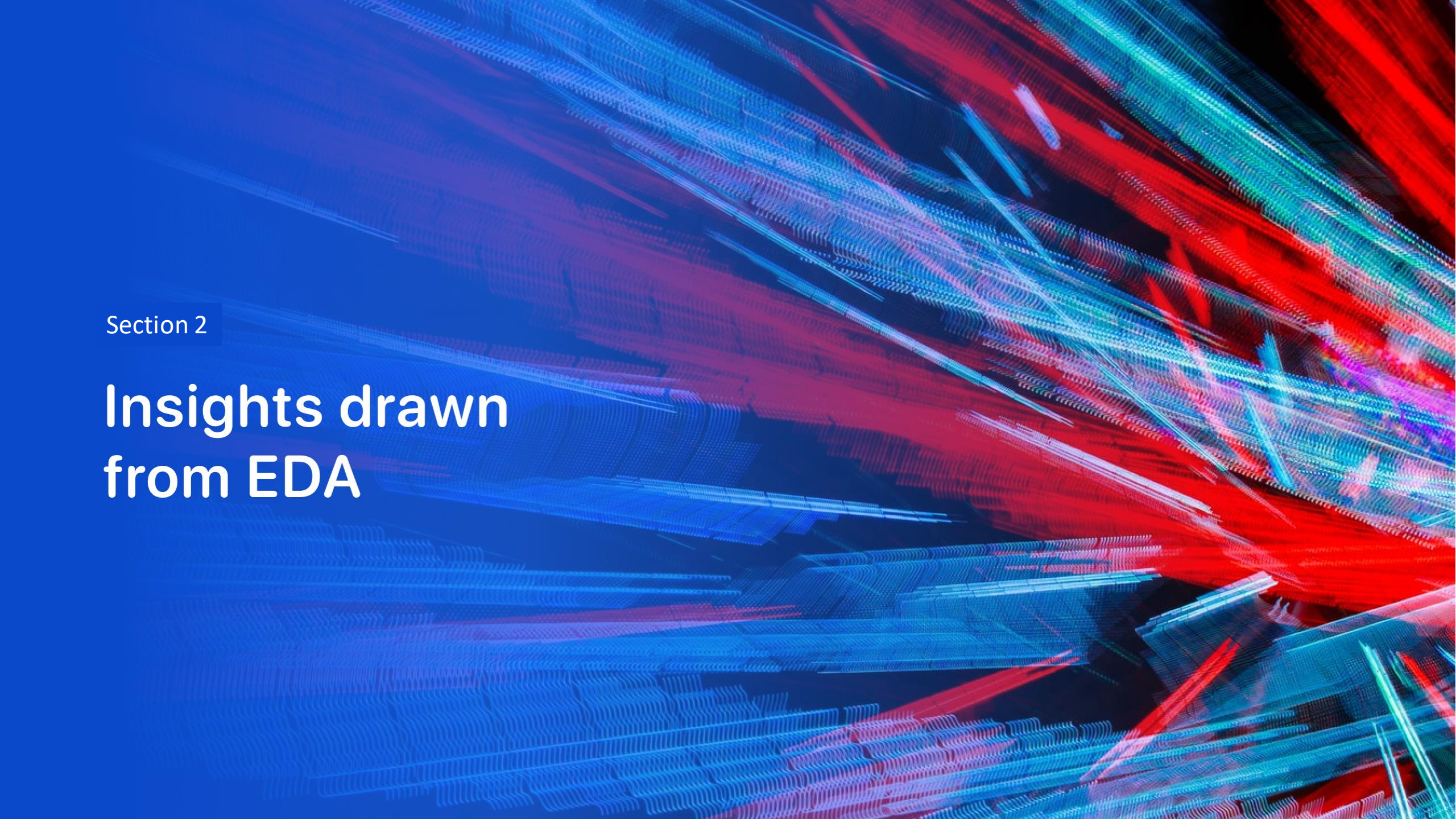
https://github.com/RafMartSev/Final_Course_Capstone/blob/main/Machine%20Learning%20Prediction.ipynb

Results

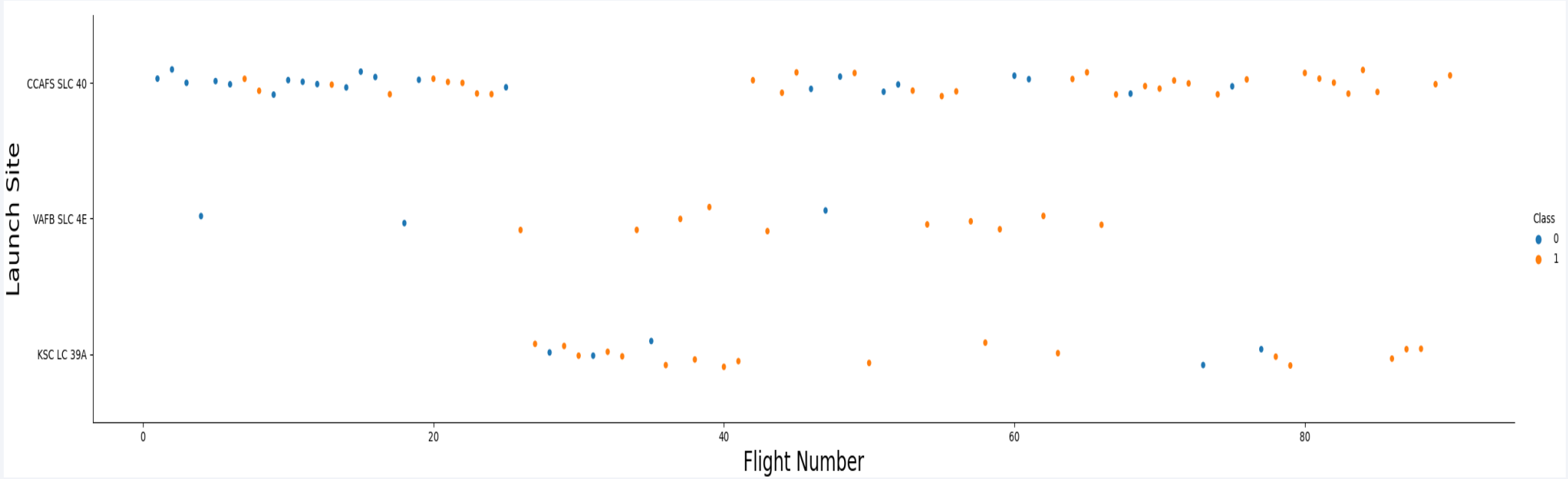
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA



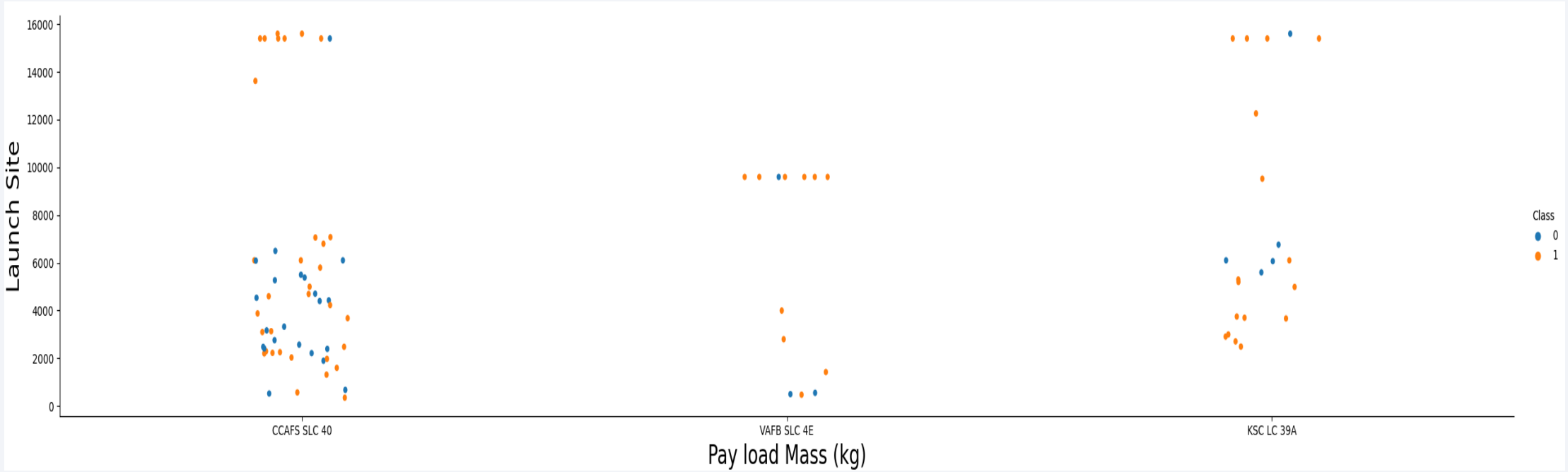
Flight Number vs. Launch Site



As the number of flights increases, the rate of success at a launch site increases.

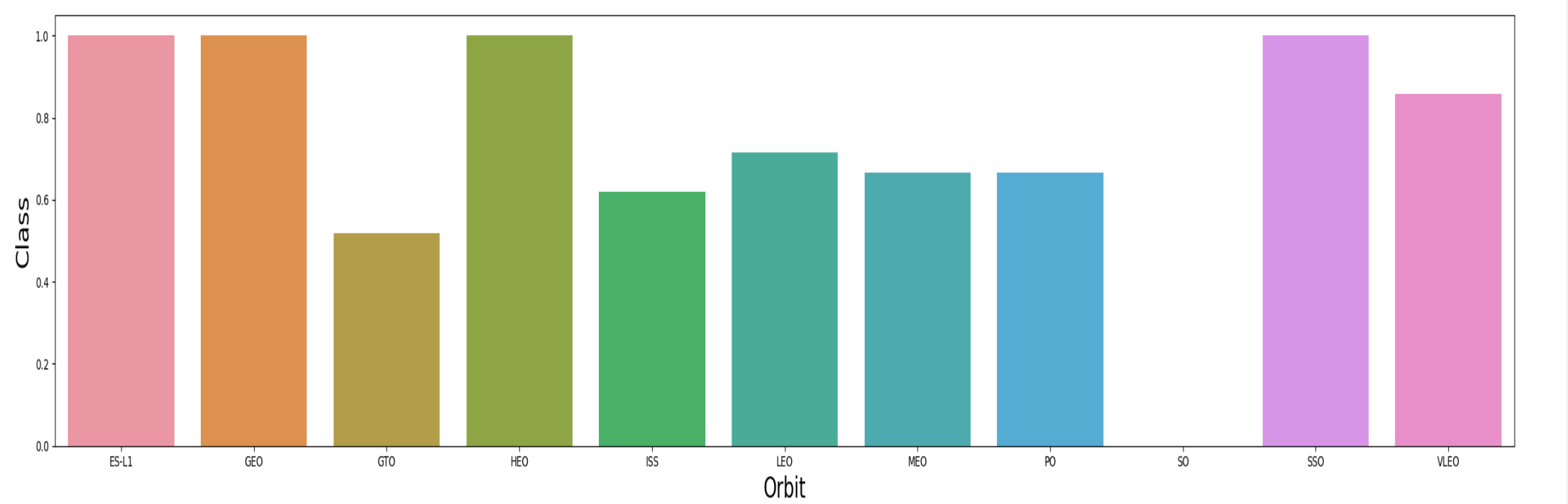
Also, according to the plot above, the launch site most successfully is CCAF5 SCL 40.

Payload vs. Launch Site



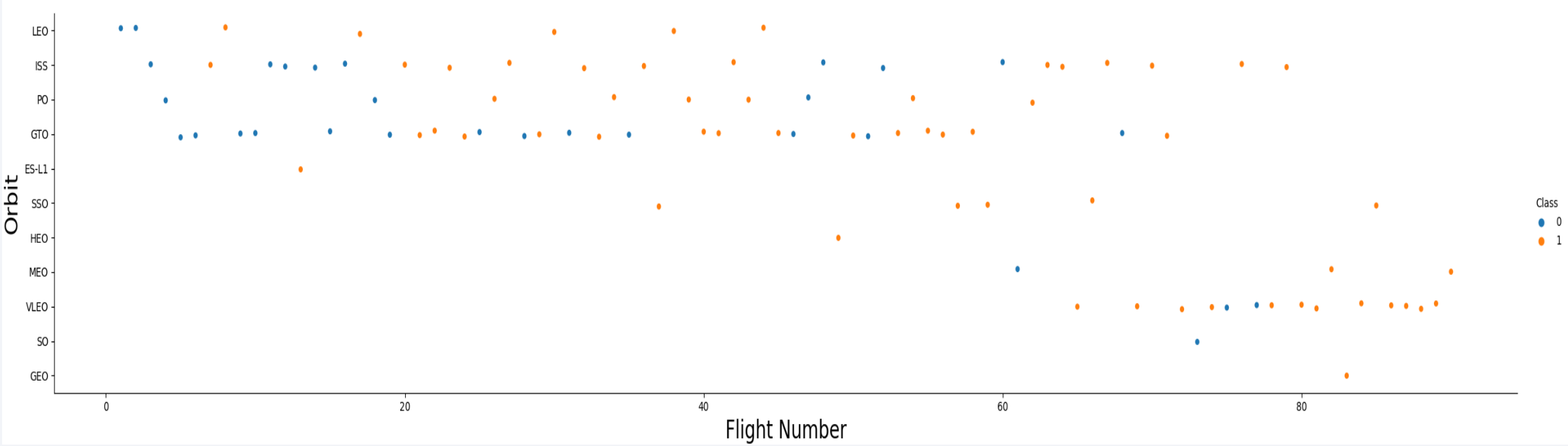
There is no clear correlation between payload mass and success rate for a given launch site.

Success Rate vs. Orbit Type



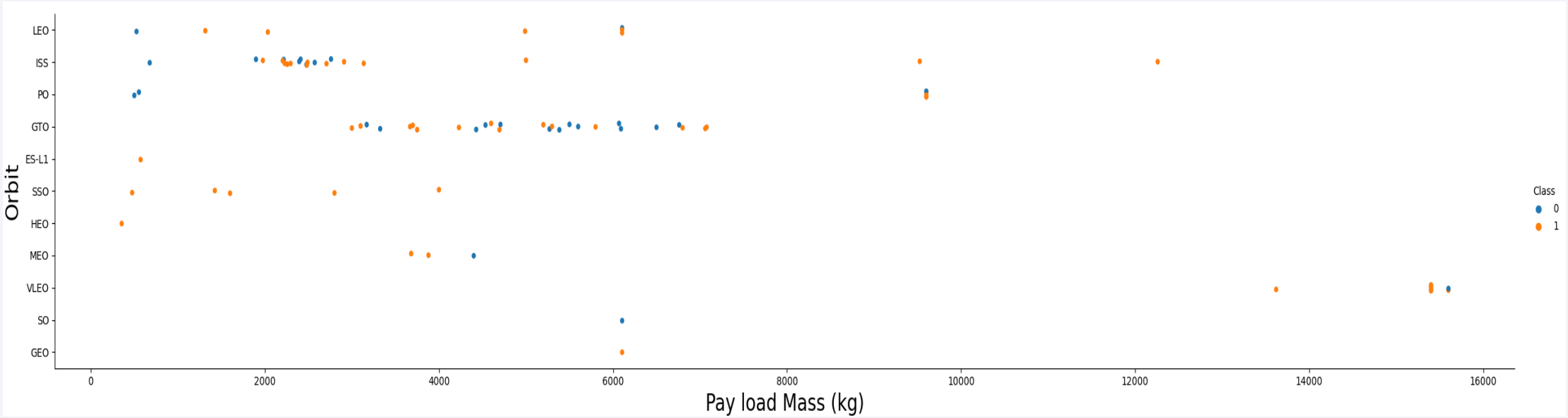
ES-L1, GEO, HEO and SSO have the most successful rate.

Flight Number vs. Orbit Type



In the LEO orbit, success is related to the number of flights whereas in the rest orbits, there is no relationship between flight number and each orbit.

Payload vs. Orbit Type

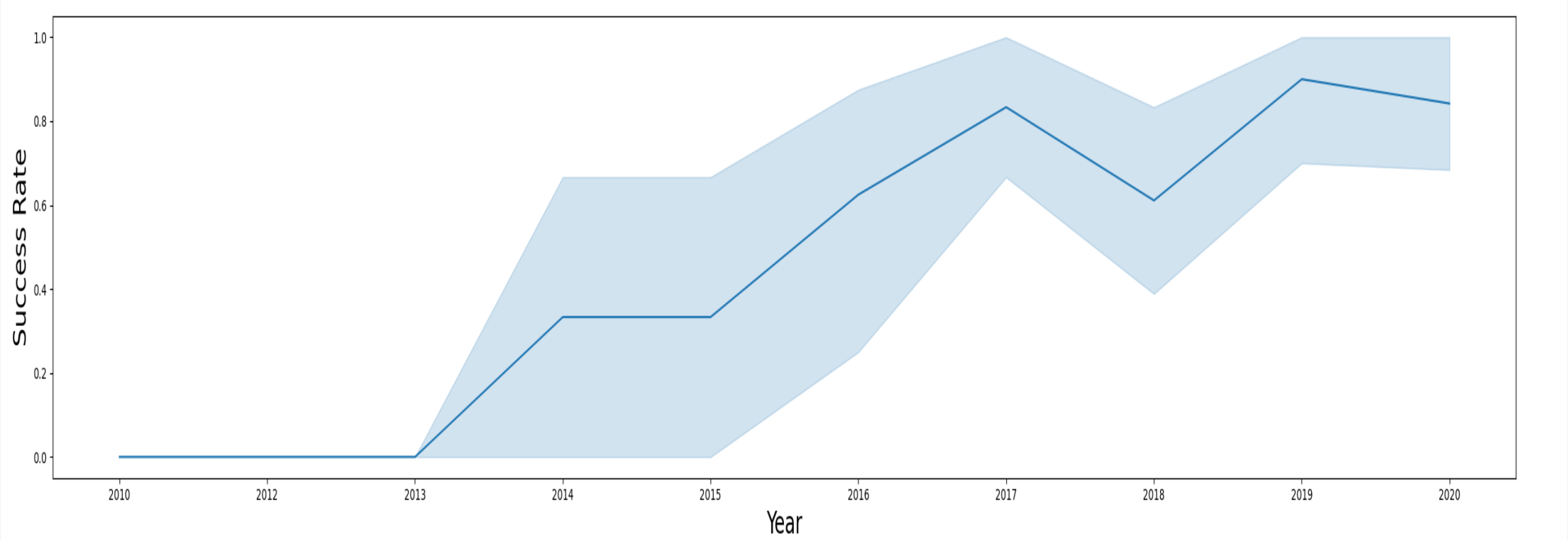


Payload mass seems to correlate with orbit

LEO and SSO seem to have relatively low payload mass

The other most successful orbit VLEO only has payload mass values in the higher end of the range

Launch Success Yearly Trend



The success rate since 2013 kept on increasing till 2020.

All Launch Site Names

```
In [8]: %%sql
        SELECT DISTINCT launch_site from SPACEXTBL

 * sqlite:///my_data1.db
Done.
Out[8]:
```

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

The use of the keyword DISTINCT in the query allows to remove duplicates in the column LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

In [10]:
```
%%sql
SELECT * from SPACEXTBL where launch_site LIKE 'CCA%' LIMIT 5;
```

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

Out[10]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	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

```
In [15]: %%sql
         select SUM (PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = "NASA (CRS)";

          * sqlite:///my_data1.db
         Done.

Out[15]: SUM(PAYLOAD_MASS__KG_)
         ---------------------
                           45596
```

This query sums the total payload mass in kg .

Average Payload Mass by F9 v1.1

```
In [18]: %%sql
         select AVG(PAYLOAD_MASS__KG_) as Average from SPACEXTBL where Booster_Version LIKE "F9 v1.1%";

          * sqlite:///my_data1.db
         Done.

Out[18]:        Average
        ---------------
        2534.6666666666665
```

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

First Successful Ground Landing Date

```
In [71]: %%sql
         select min(DATE) from SPACEXTBL where "Landing _Outcome" = "Success (ground pad)";

          * sqlite:///my_data1.db
         Done.

Out[71]: min(DATE)
         ----------
         01-05-2017
```

The MIN keyword is used to calculate the minimum of the DATE column and the WHERE keyword filters the results to only the successful ground pad landings.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [75]:
%%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.
```

Out[75]:

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

Using the **WHERE** clause to filter for boosters which have successfully landed on drone ship and applied the **AND** condition to determine successful landing with payload mass greater than 4000 but less than 6000

Total Number of Successful and Failure Mission Outcomes

In [74]:
```
%%sql
select count(Mission_Outcome) as Total_Missions from SPACEXTBL where Mission_Outcome like "Success%" or
Mission_Outcome like "Failure%";
```

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

Out[74]:

Total_Missions
101

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

```
In [60]:
%%sql
select Booster_Version from SPACEXTBL where PAYLOAD_MASS__KG_=(select max(PAYLOAD_MASS__KG_) from SPACEXTBL);
```

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

Out[60]:

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

To determinate the booster that have carried the maximum payload using a subquery in the **WHERE** clause and the **MAX()** function.

2015 Launch Records

In [79]:
```
%%sql
select (substr(Date,4,2)), Booster_Version, Launch_Site from SPACEXTBL where DATE like "%2015" and
"Landing _Outcome" = "Failure (drone ship)";
```

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

Out[79]:

(substr(Date,4,2))	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

The WHERE keyword is used to filter the results for only failed landing outcomes, AND only for the year of 2015.

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

In [90]:
```
%%sql
select "Landing _Outcome", count(*) as count from SPACEXTBL where "Landing _Outcome" like "Success%" and
date between '04-06-2010' and '20-03-2017'
group by "Landing _Outcome"
order by count DESC;
```

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

Out[90]:

Landing _Outcome	count
Success	20
Success (drone ship)	8
Success (ground pad)	6

Selecting Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2017-03-20 and applying the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.


Section 3
Launch Sites
Proximities Analysis

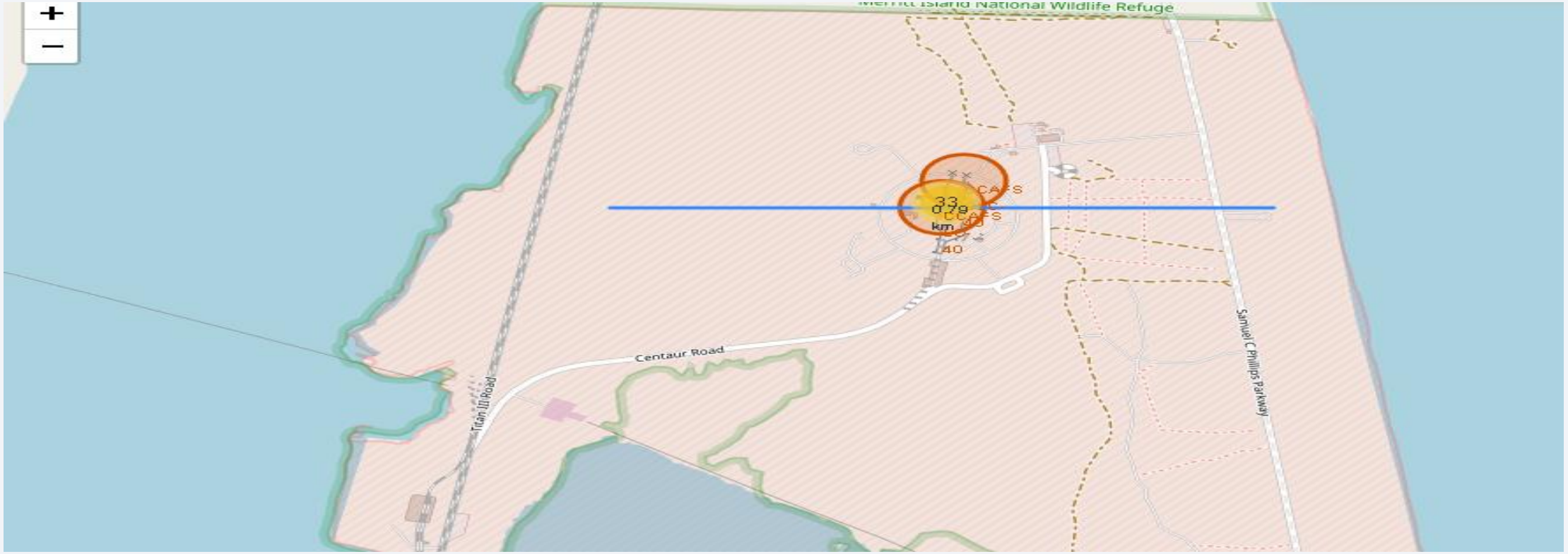
Grounds site locations for launch

This are the SpaceX locations for launch on the coast of United States.

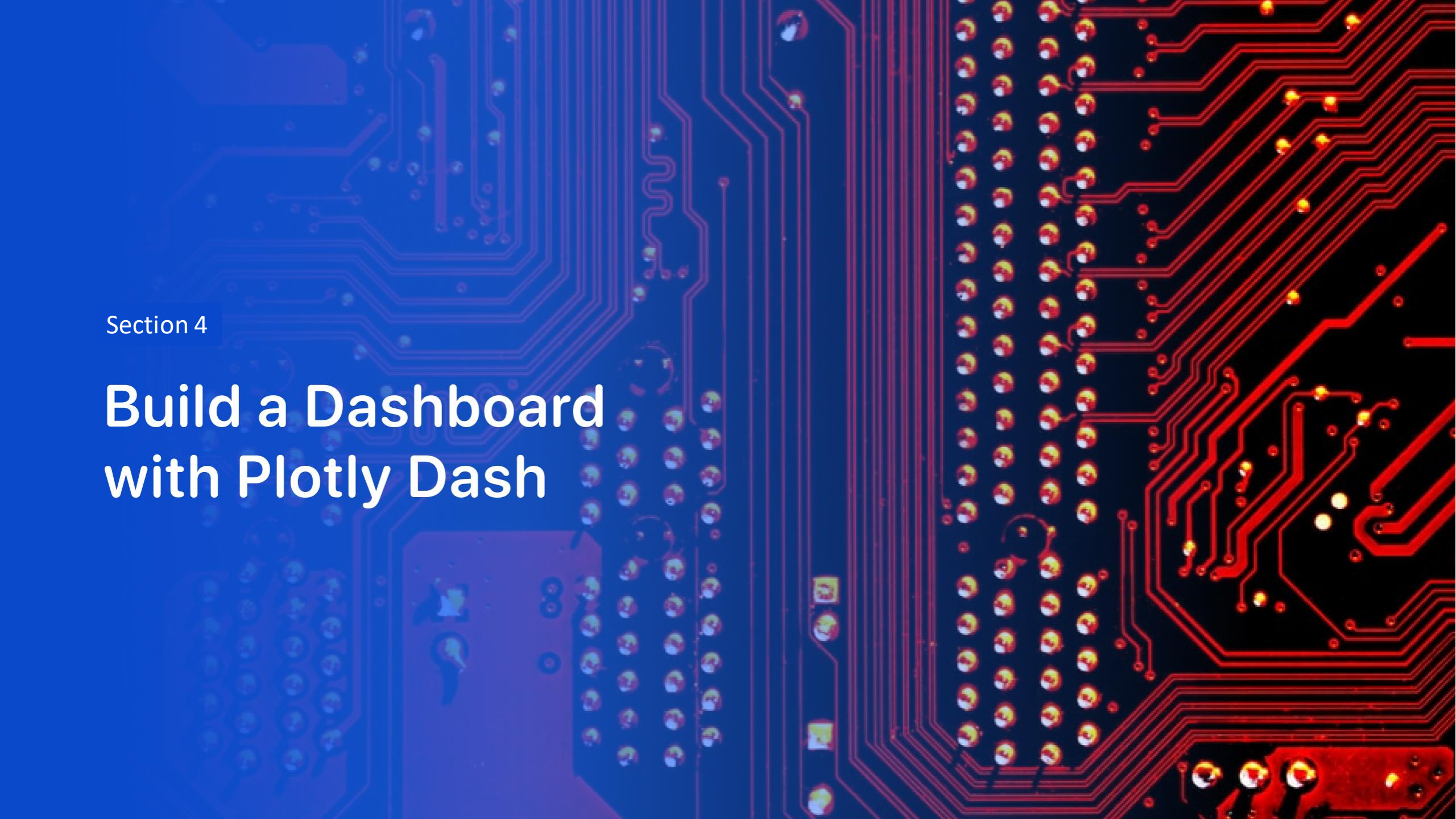
Successful or failed launches in the launch sites.

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

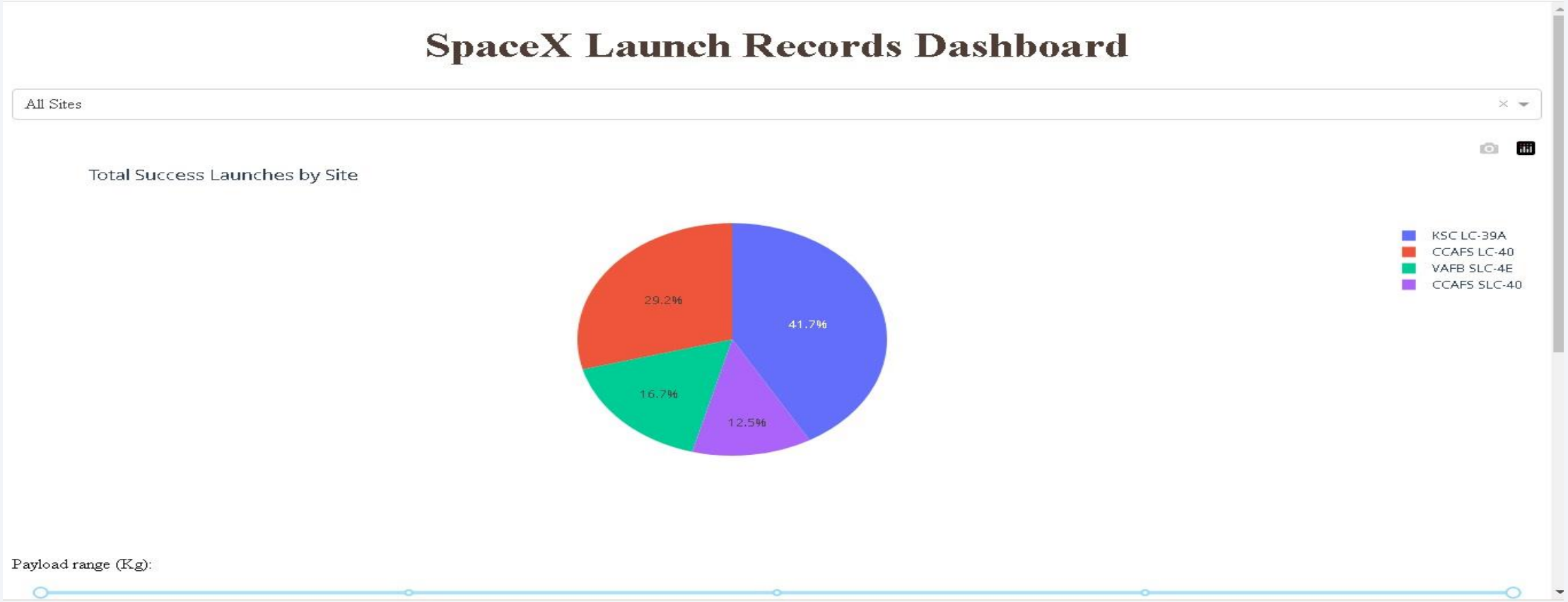
Distance to the coast



Launch sites are also close to coasts

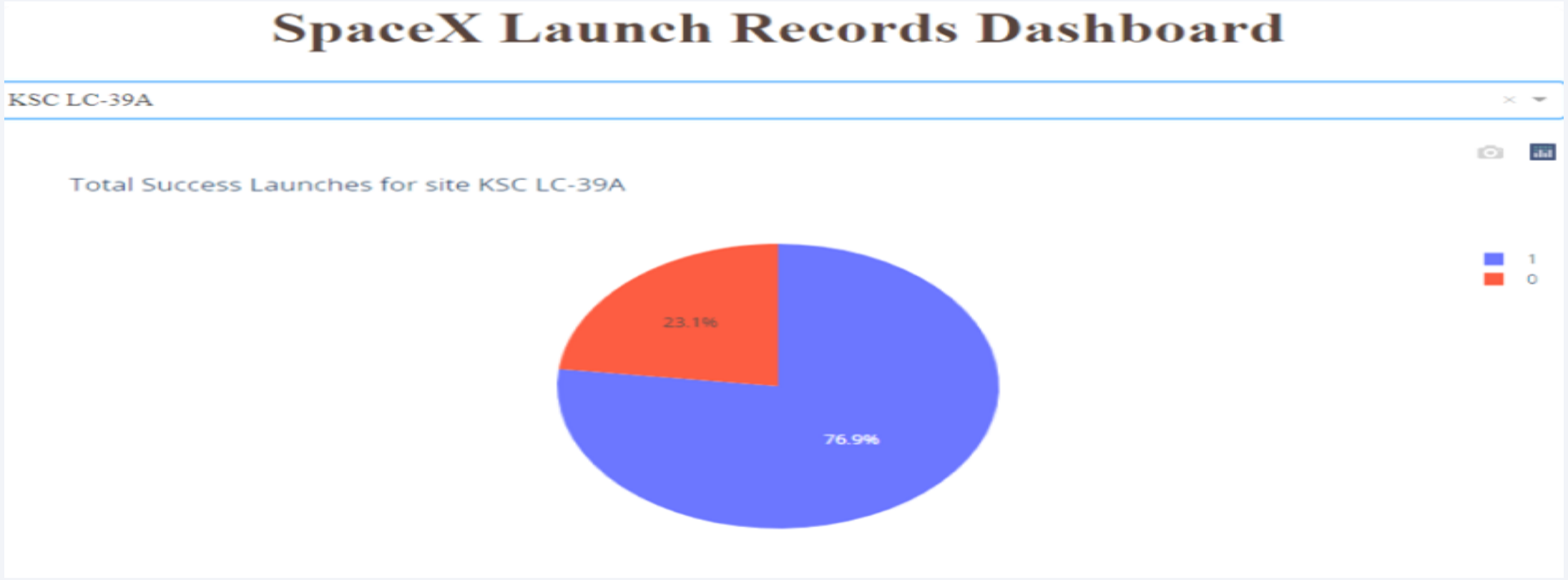

Section 4
Build a Dashboard with Plotly Dash

Successful launch rate in different launch sites



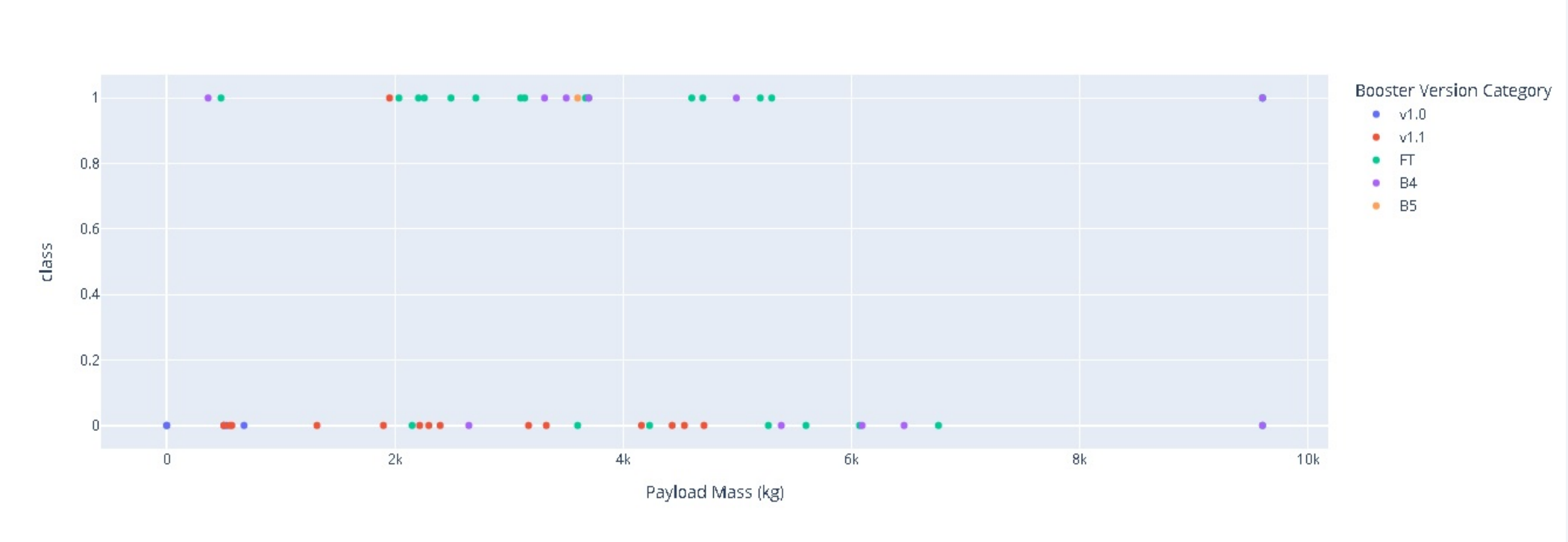
KSC LC-39A has the best success rate of launches.

Launch success ratio for KSC LC-39A



The successful rate in KSC LC-39A is 76'9%.

Successful rate vs Payload Mass

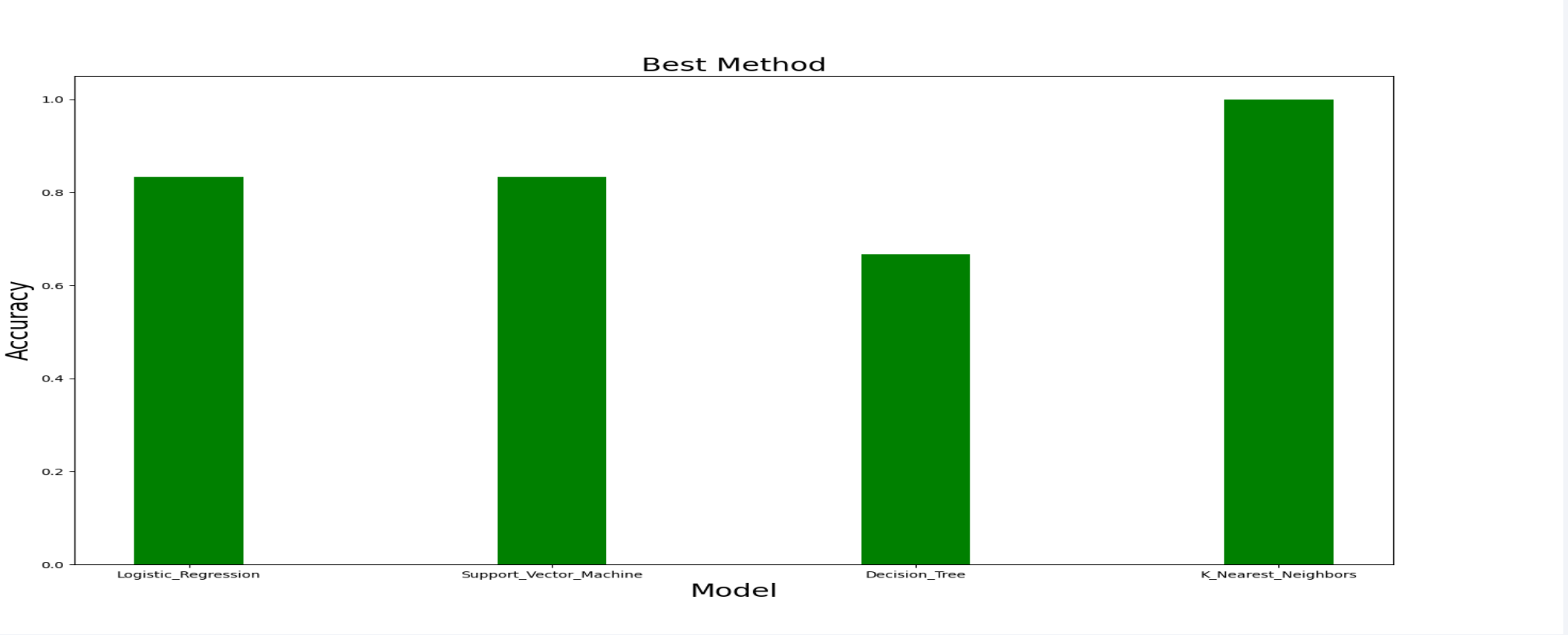


Class indicates 1 for successful landing and 0 for failure. Scatter plot also accounts for booster version category in color and number of launches in point size.

Section 5

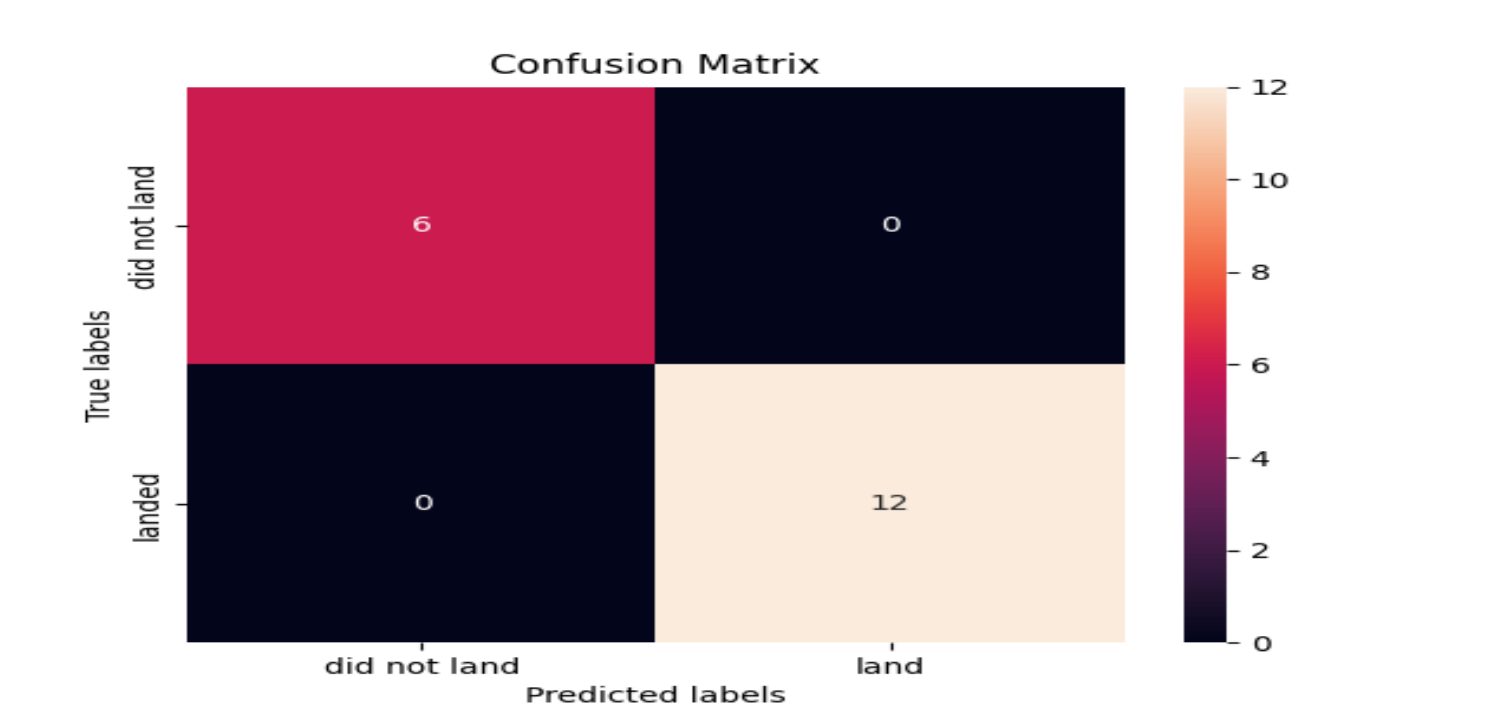
Predictive Analysis (Classification)

Classification Accuracy



The model with the best accuracy is KNN.

Confusion Matrix



KNN is the best model and its accuracy is 100%.

Conclusions

- The larger the flight amount at a launch site, the greater the success rate at a launch site
- The launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches, and also the highest rate of successful launches, with a 76.9% success rate.
- It is impossible to explain why certain launch sites are better than other with the current data.
- KNN is the best model and can be used to predict the next launch.

Appendix

The next python functions were needed to resolve different problematic situations.

```
def plot_confusion_matrix(y,y_predict):
    "this function plots the confusion matrix"
    from sklearn.metrics import confusion_matrix

    cm = confusion_matrix(y, y_predict)
    ax= plt.subplot()
    sns.heatmap(cm, annot=True, ax = ax); #annot=True to annotate cells
    ax.set_xlabel('Predicted labels')
    ax.set_ylabel('True labels')
    ax.set_title('Confusion Matrix');
    ax.xaxis.set_ticklabels(['did not land', 'land']); ax.yaxis.set_ticklabels(['did not land', 'landed'])
    plt.show()
```

```
from js import fetch
import io

URL1 = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_2.csv"
resp1 = await fetch(URL1)
text1 = io.BytesIO((await resp1.arrayBuffer()).to_py())
data = pd.read_csv(text1)
```

Thank you!