IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marc Grau
6/9/2024

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction

- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics
  - Predictive Analytics results

Introduction

- SPACEY is here to compete to the space race. To achieve that we need to reuse rockets and avoid landings that end up in crashes.
- What factors determine that the rocket will land successfully?
- Which are the most important features that will help us to classify if a landing will end up in a success?

Section 1

Methodology

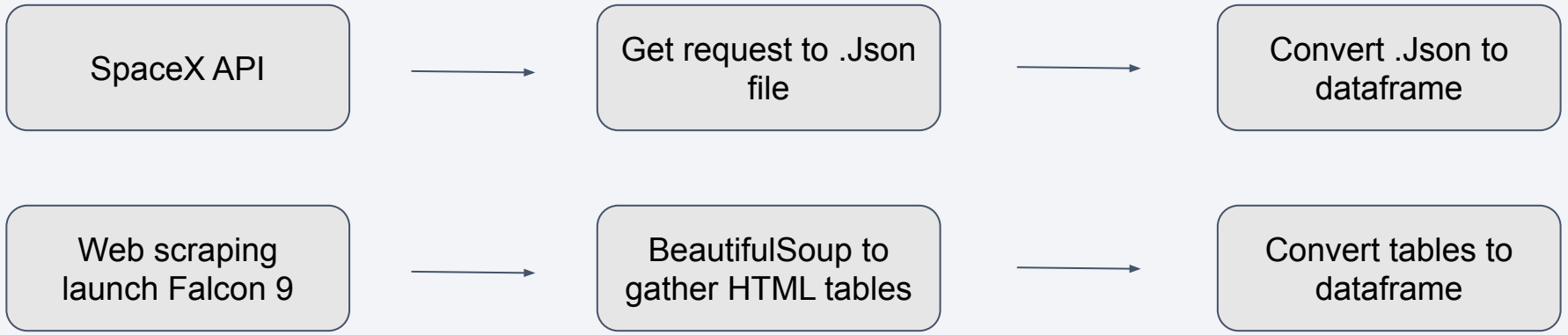
Methodology

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

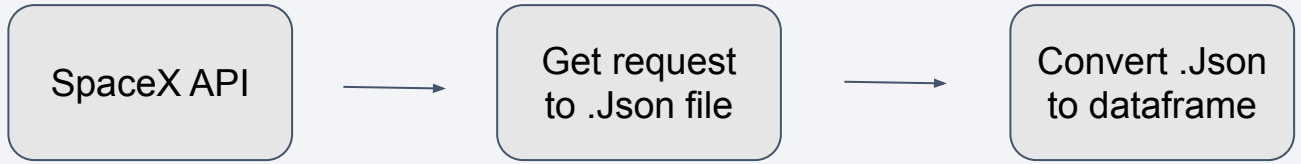
Data Collection

- From the SpaceX API using a request and the .json file was converted to dataframe
- Web scraping from Wikipedia for Falcon 9 launch records and we use the BeautifulSoup framework. We extracted the table and convert it to pandas dataframe.



Data Collection – SpaceX API

- We use get request to the SpaceX API to collect, clean and save to dataframe

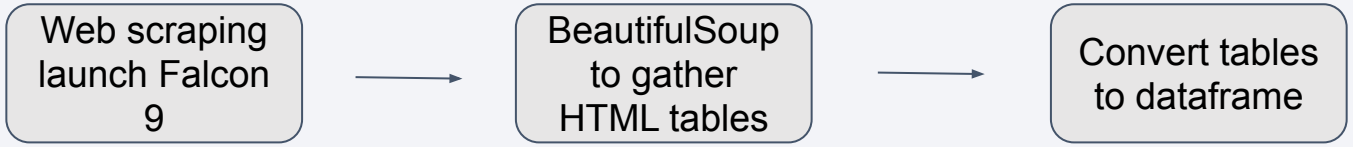


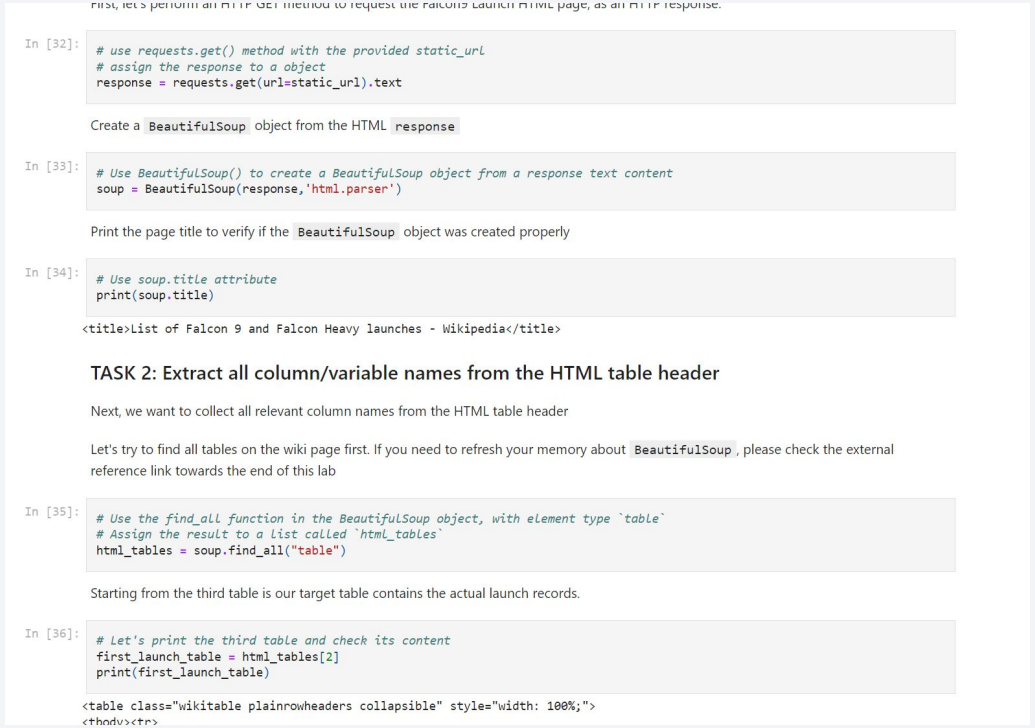
- https://github.com/MGrauLeguia/coursera/blob/main/1_jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

- We used Web scraping from the wikipedia page to extract Falcon9 launches and save the table to dataframe

- https://github.com/MGrauLeguia/coursera/blob/main/2_jupyter-labs-webscraping.ipynb





First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

In [32]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(url=static_url).text
```

Create a `BeautifulSoup` object from the HTML `response`

In [33]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response,'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [34]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

TASK 2: Extract all column/variable names from the HTML table header

Next, we want to collect all relevant column names from the HTML table header

Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

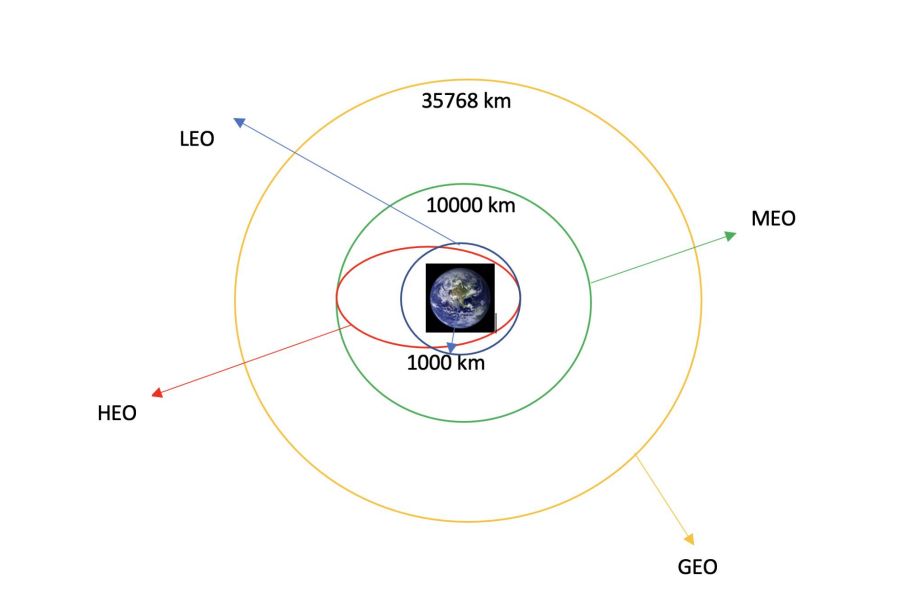
In [35]:
```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.find_all("table")
```

Starting from the third table is our target table contains the actual launch records.

In [36]:
```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
<table class="wikitable plainrowheaders collapsible" style="width: 100%;">
<tbody><tr>
```

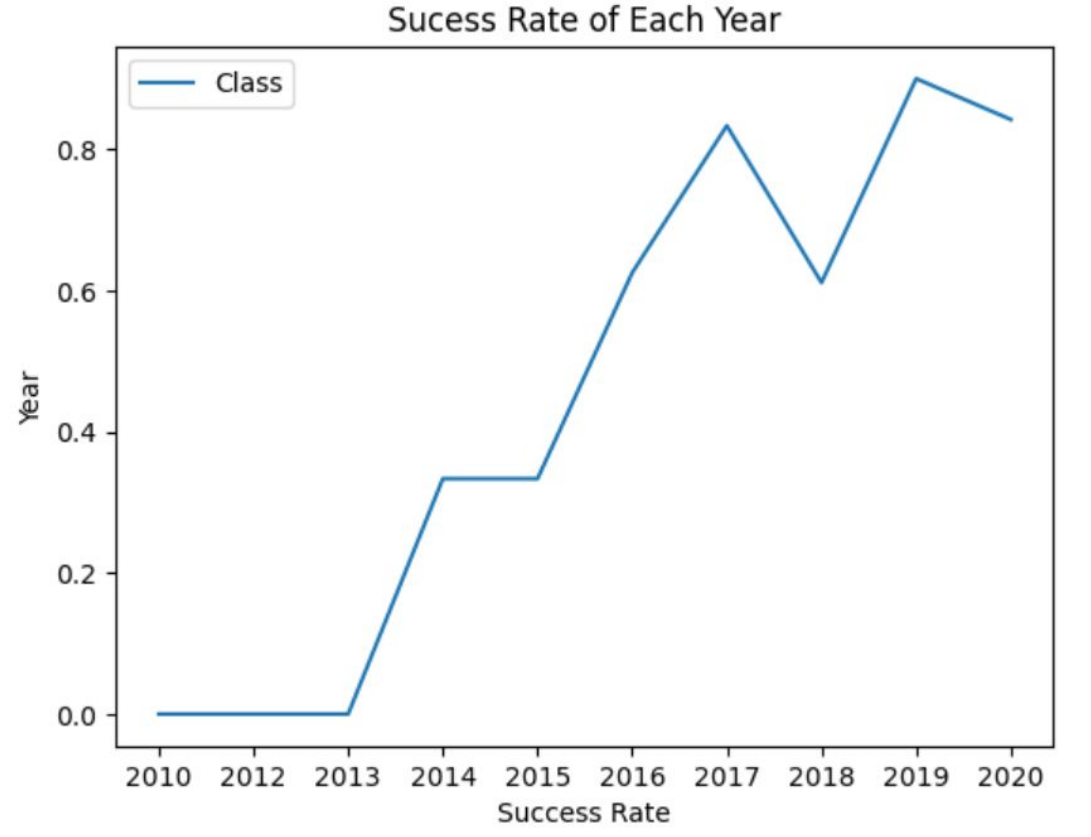
Data Wrangling



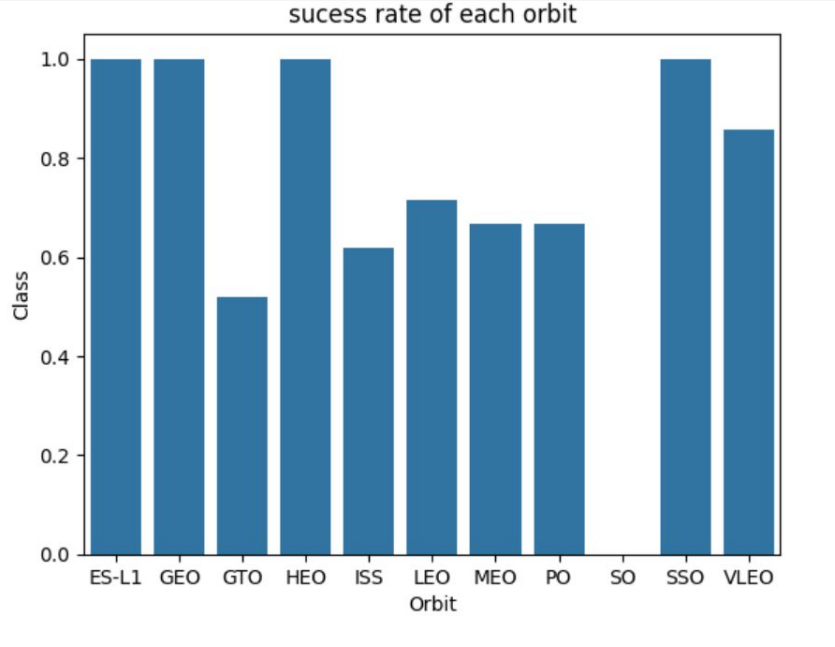
- Exploratory data analysis.
- Check the launching sites
- Created a landing outcome label and save the dataframe.
- https://github.com/MGrauLeguia/coursera/blob/main/3_Iabs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Success rate increased over time!



Some orbits have better success rate



https://github.com/MGrauLeguia/coursera/blob/main/5_edadataviz.ipynb

EDA with SQL

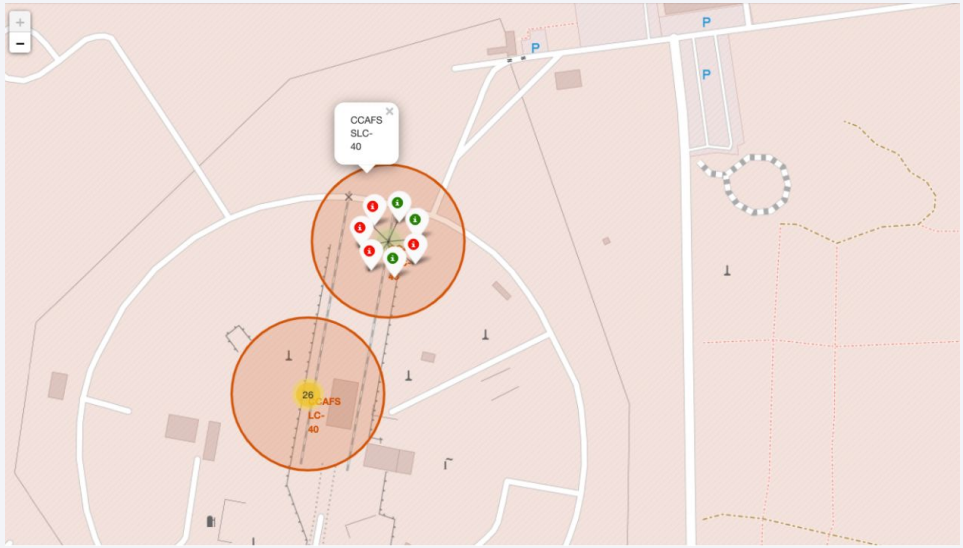
Some of the queries:

- Names of unique launch sites
- Display average payload mass
- Average payload mass carried by F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List total number of successful lands
- Failed landing outcomes in drone ships, their booster version and launch site

- https://github.com/MGrauLeguia/coursera/blob/main/4_jupyter-labs-eda-sql-coursera_sqllite.ipynb

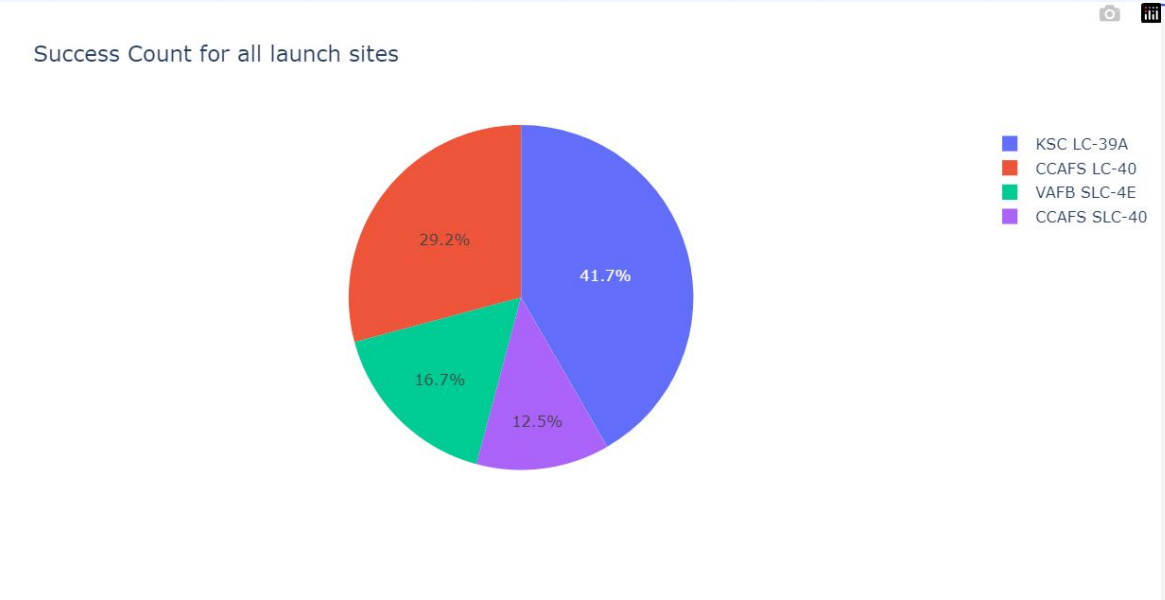
Build an Interactive Map with Folium

- We put markers and circles over the launch sites and added a legend whether a launch at that site was successful or not.
- We add lines to closest populations, roads, trains and coast to determine how close to infrastructure they were.



https://github.com/MGrauLeguia/coursera/blob/main/6_lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash



- We build an interactive dashboard.
- We plotted pie charts with the success rate by different locations
- We plotted a scatter plot between the Outcome and the Payload Mass for different booster versions

https://github.com/MGrauLeguia/coursera/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

- Loaded the data and transformed to be analysed
- Slipt in train/test set with 80% train.
- For the different models we tuned the hyperparameters with GridSearchCV.
- We trained a logistic regression, a SVM, a decision Tree and finally a KNN.

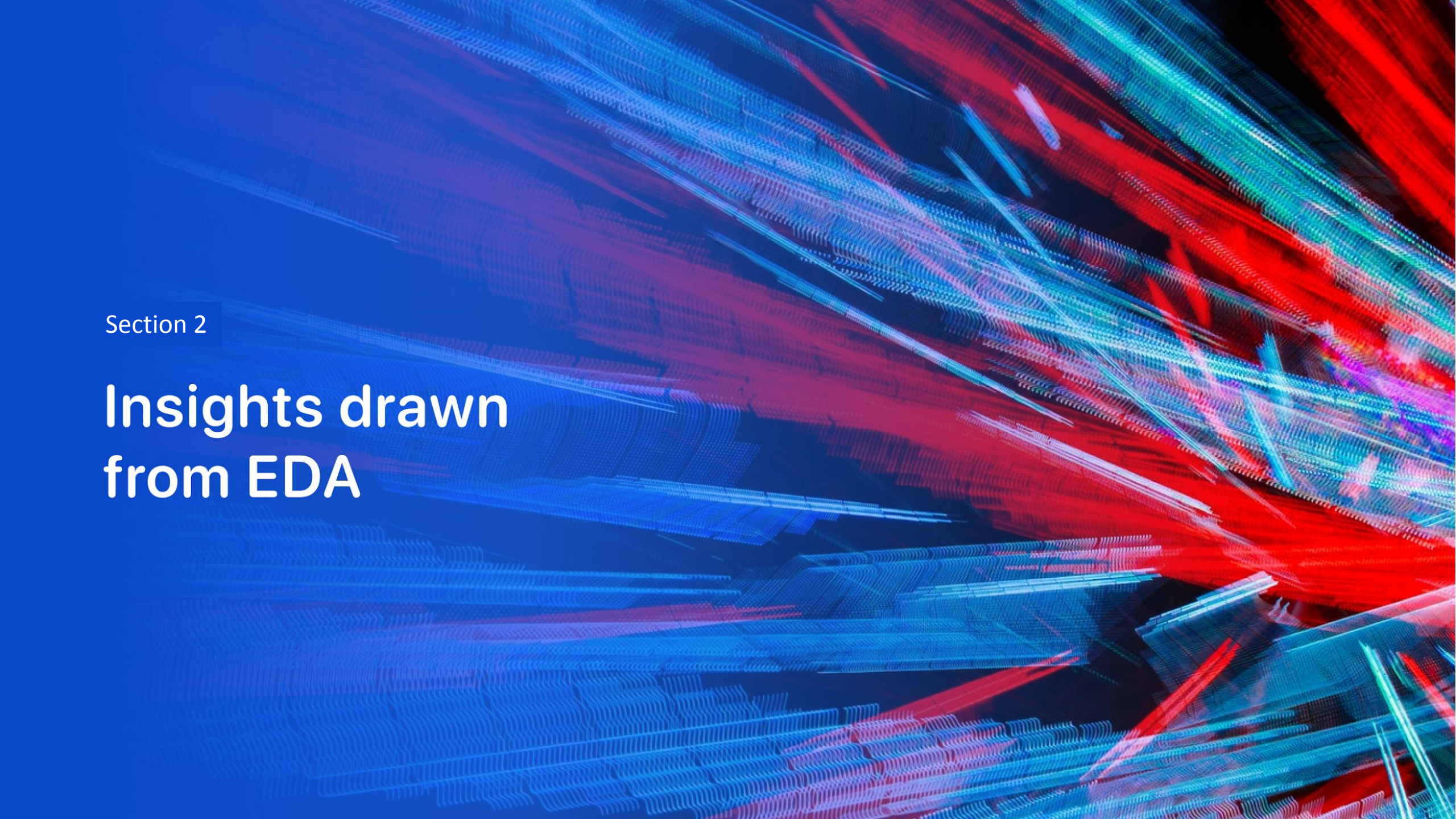
https://github.com/MGrauLeguia/coursera/blob/main/7_SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

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

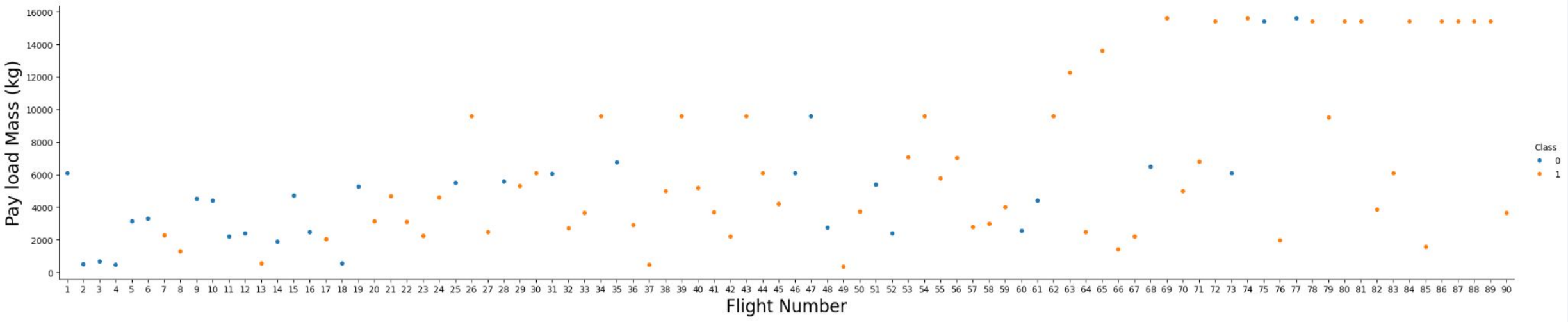
Section 2

Insights drawn from EDA



Flight Number vs. Launch Site

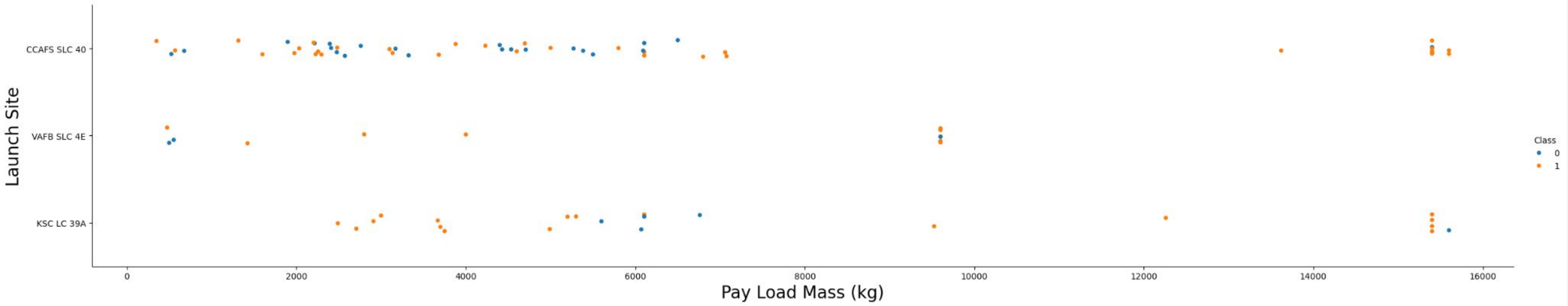
- Flight Number vs. Launch Site



As more flights were launched Payload mass also increased and there were more class 0 outcomes at the beginning.

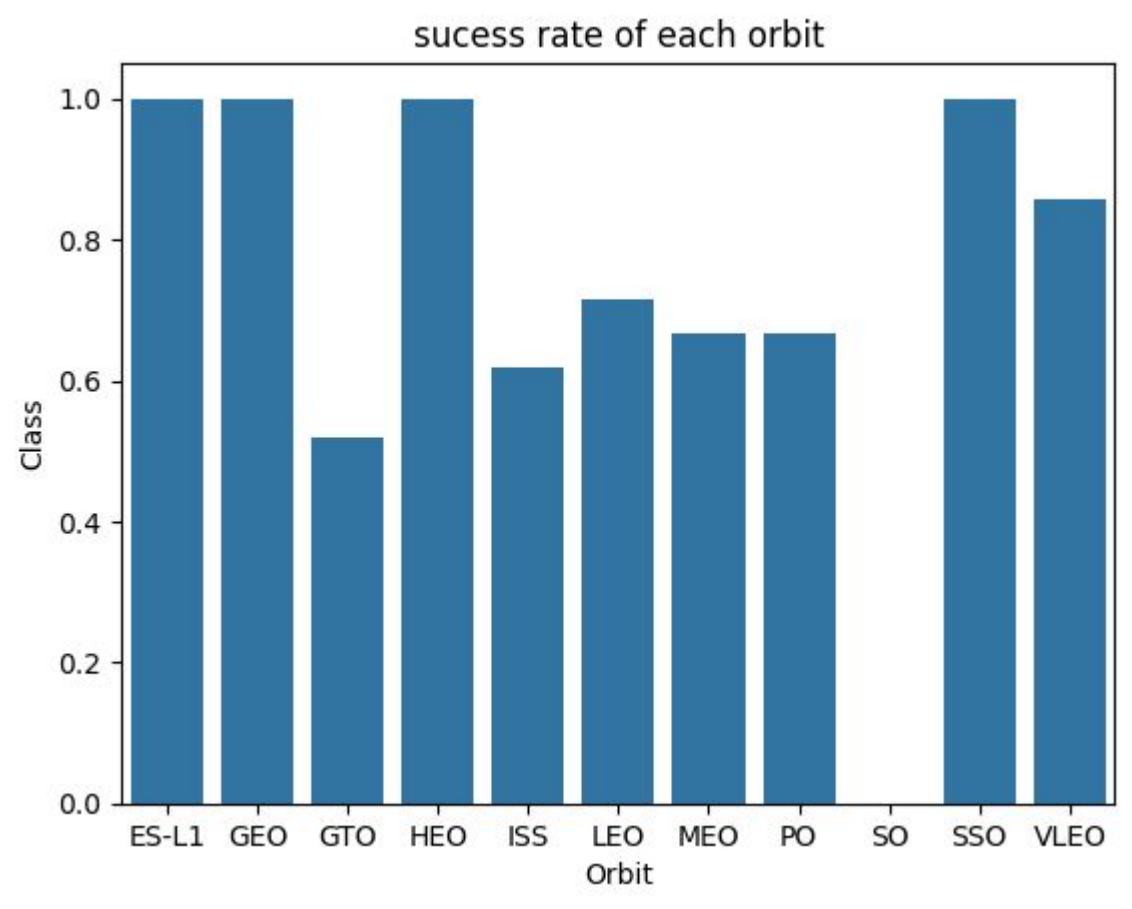
Payload vs. Launch Site

- Payload vs. Launch Site



No Launches over 10000kg in VAFB

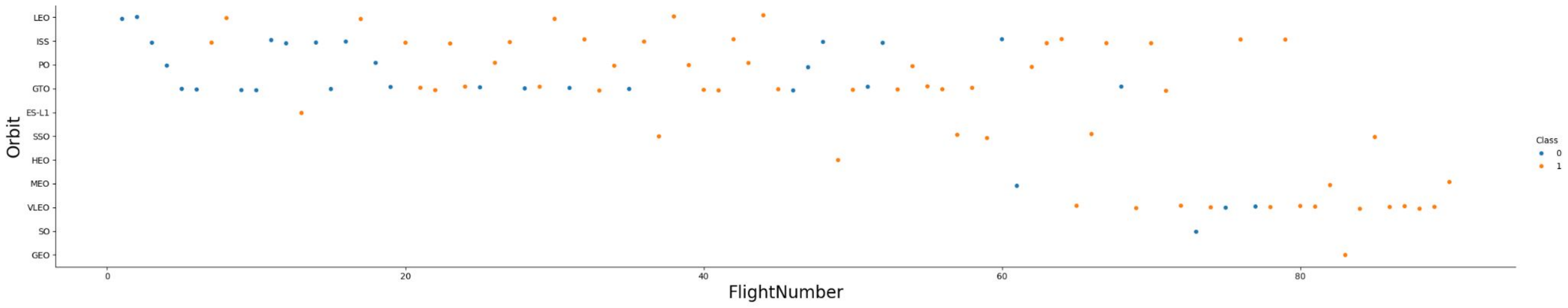
Success Rate vs. Orbit Type



Some orbits have 100% success while others way lower.

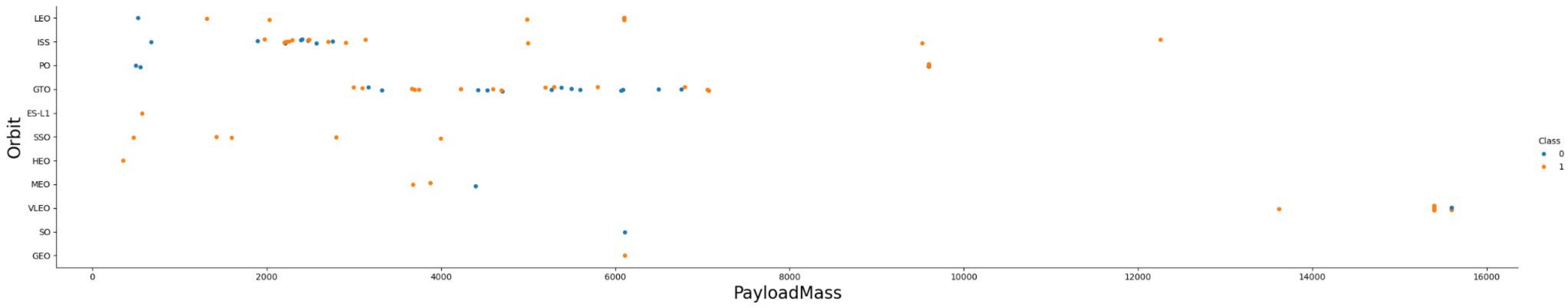
Flight Number vs. Orbit Type

- Flight number vs. Orbit type



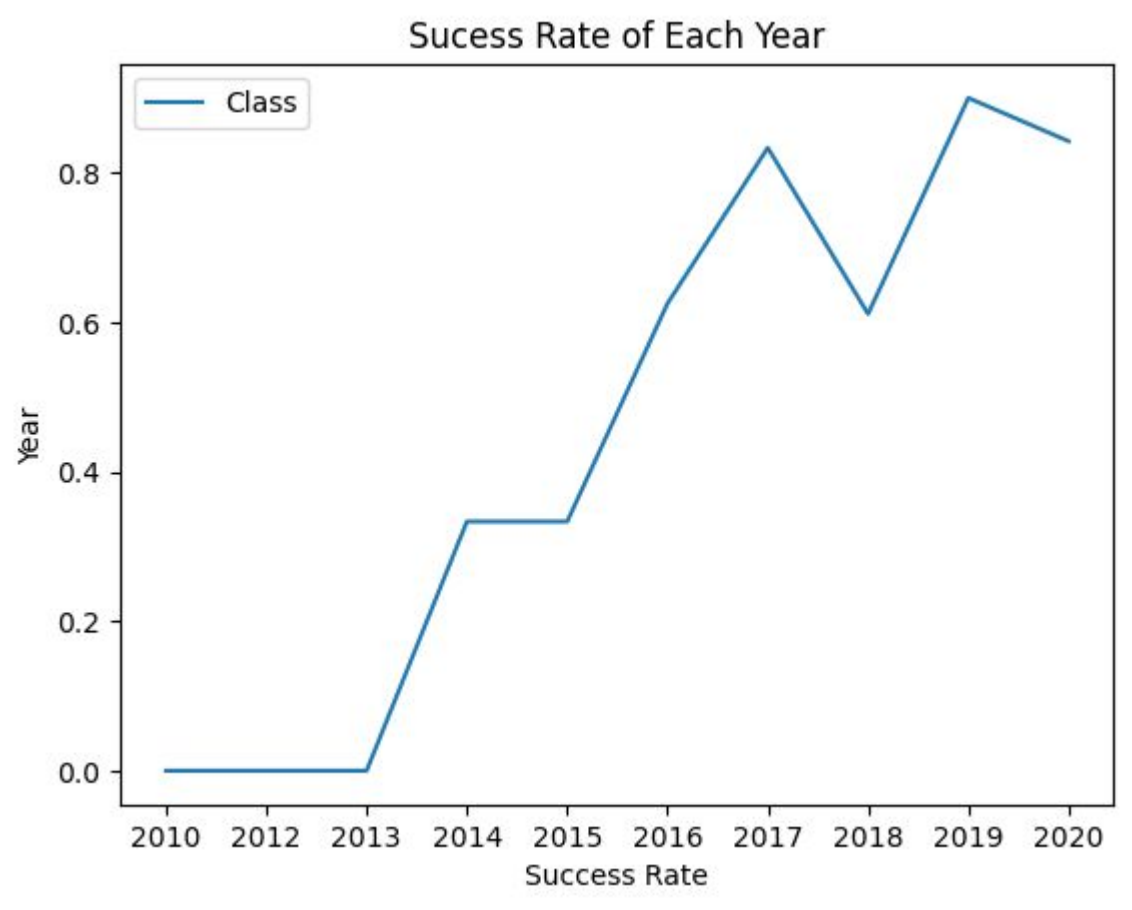
They shifted the orbit as they did more flights

Payload vs. Orbit Type



Some orbits require less payload mass

Launch Success Yearly Trend



Success rate increase with the experience untial a point.

All Launch Site Names

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

4 launch sites names in total

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [38]:
```
%sql Select * from SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' limit 5
```

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

Out[38]:

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

All them did not properly land.

Total Payload Mass

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

In [26]:
```
%sql Select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Customer" LIKE 'NASA (CRS)%'
```

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

Out[26]:

SUM(PAYLOAD_MASS__KG_)
48213

Check the total payload mass. Used SQL with select sum.

Average Payload Mass by F9 v1.1

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

In [30]:
```
%sql Select AVG(PAYLOAD_MASS__KG_) from SPACEXTBL WHERE "Booster_Version" LIKE 'F9 v1.1%'
```

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

Out[30]:

AVG(PAYLOAD_MASS__KG_)
2534.6666666666665

We select the average payload mass
and we only wanted the F9 v1.1

First Successful Ground Landing Date

In [39]:
```
%sql Select MIN("Date") from SPACEXTBL WHERE "Landing_Outcome"="Success (ground pad)"
```

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

Out[39]:

MIN("Date")
2015-12-22

We select the Minimum date where the landing was a success

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [44]:
```
%sql Select Booster_Version from SPACEXTBL where "Landing_Outcome"="Success (drone ship)" AND "PAYLOAD_MASS__KG_">4000 AND '
```

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

Out[44]:

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

Had to specify in the query the mass and that it was a success in drone ship

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [45]:
```
%sql Select "Mission_Outcome", count("Mission_Outcome") from SPACEXTABLE GROUP BY "Mission_Outcome"
```

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

Out[45]:

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

100 successes and 1 failure

Boosters Carried Maximum Payload

```
In [47]: %sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTA
 * sqlite:///my_data1.db
Done.
```

Out[47]:

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 lot of different versions

2015 Launch Records

In [51]:
```
%sql Select substr(Date, 6,2) as Month,MISSION_OUTCOME,BOOSTER_VERSION,LAUNCH_SITE FROM SPACEXTBL where substr(Date,0,5)='20
```

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

Out[51]:

Month	Mission_Outcome	Booster_Version	Launch_Site
01	Success	F9 v1.1 B1012	CCAFS LC-40
02	Success	F9 v1.1 B1013	CCAFS LC-40
03	Success	F9 v1.1 B1014	CCAFS LC-40
04	Success	F9 v1.1 B1015	CCAFS LC-40
04	Success	F9 v1.1 B1016	CCAFS LC-40
06	Failure (in flight)	F9 v1.1 B1018	CCAFS LC-40
12	Success	F9 FT B1019	CCAFS LC-40

We had to subtract the month and the year for it to work

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

descending order.

```
In [20]: %sql Select Landing_Outcome, count(*) from SPACEXTBL where Date between '2010-06-04' AND '2017-03-20' group by Landing_Outcome OR
```

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

Out[20]:

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

Important say the date between.


Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

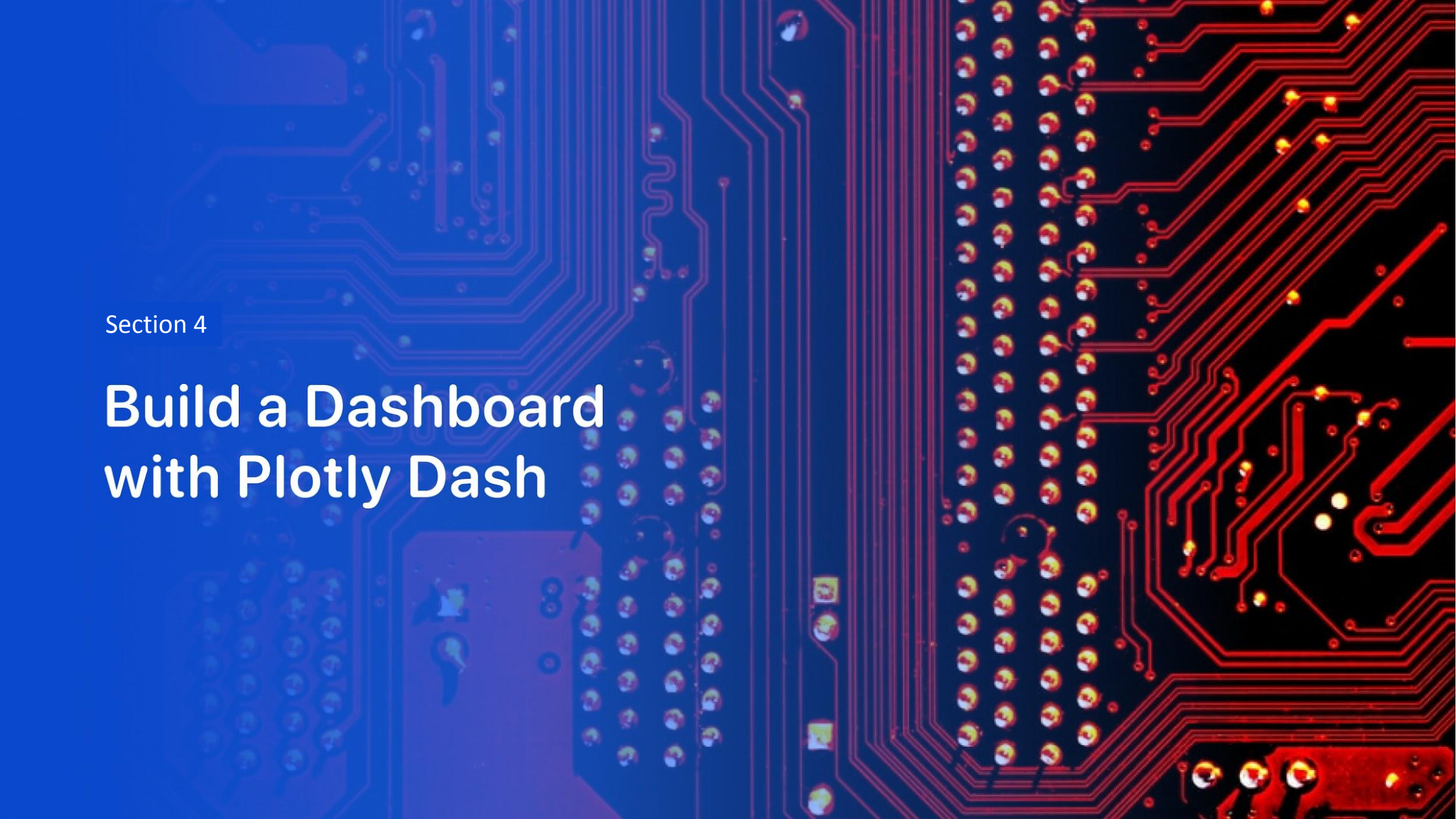
- Replace <Folium map screenshot 1> title with an appropriate title
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 2>

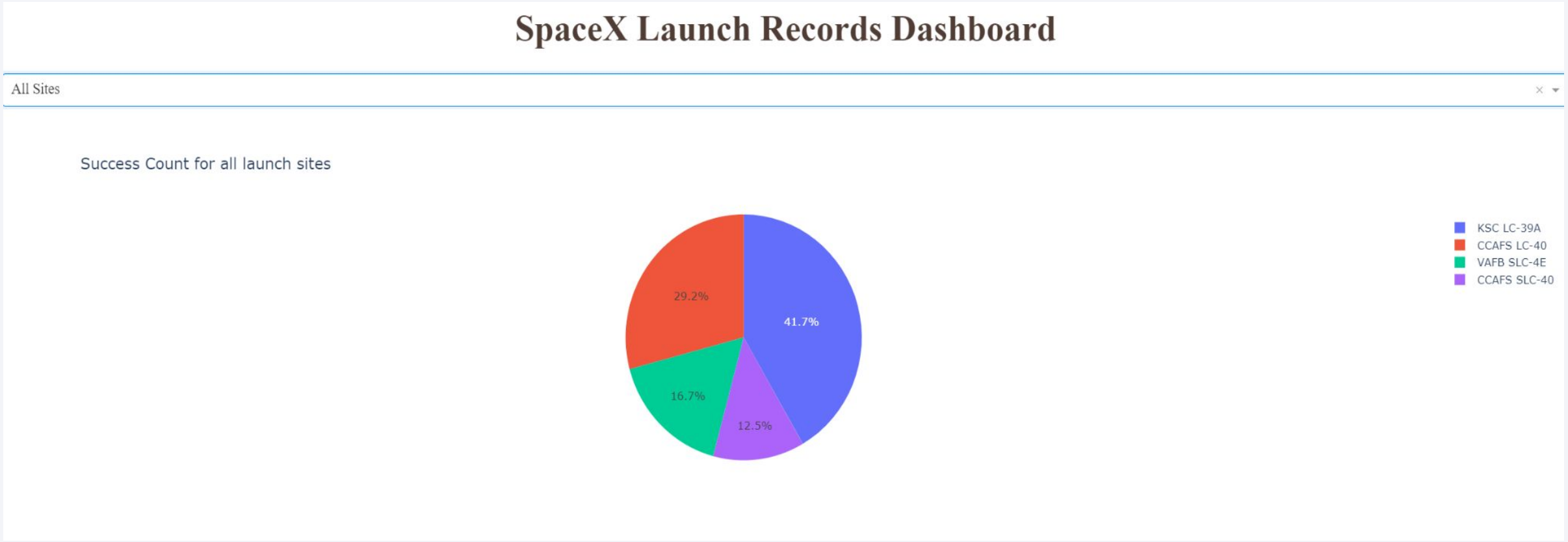
- Replace <Folium map screenshot 2> title with an appropriate title
- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map
- Explain the important elements and findings on the screenshot

<Folium Map Screenshot 3>

- Replace <Folium map screenshot 3> title with an appropriate title
- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed
- Explain the important elements and findings on the screenshot

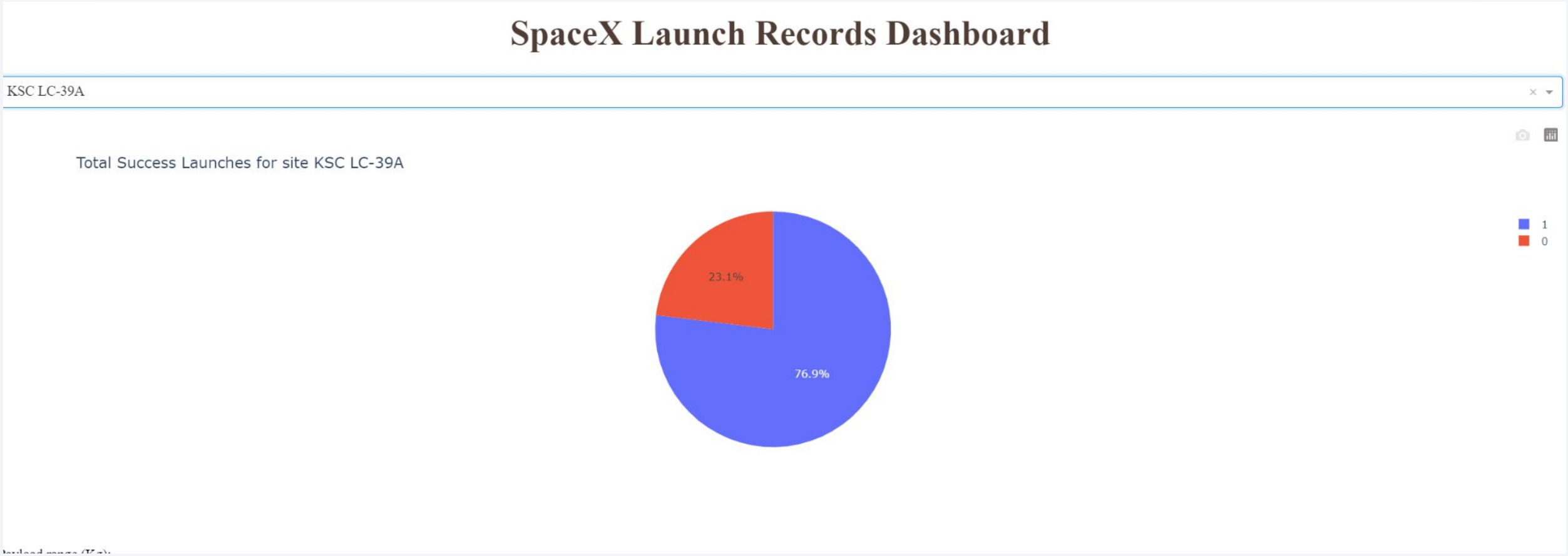

Section 4
Build a Dashboard with Plotly Dash

Success Launch per site



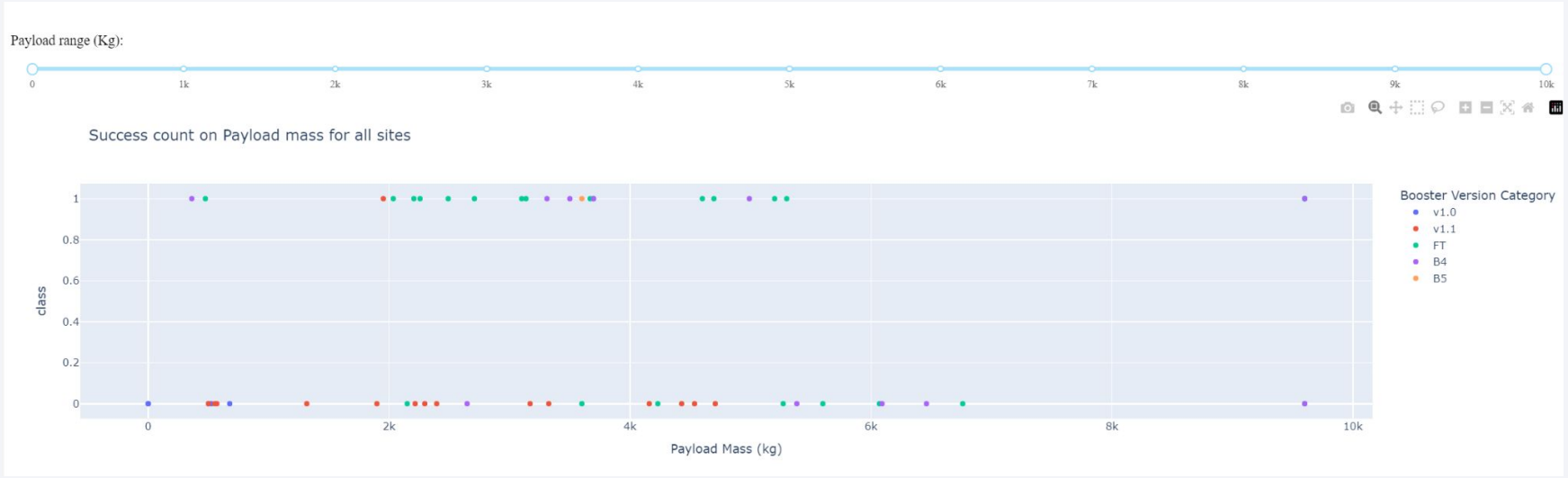
Most of the Launches are in KSC

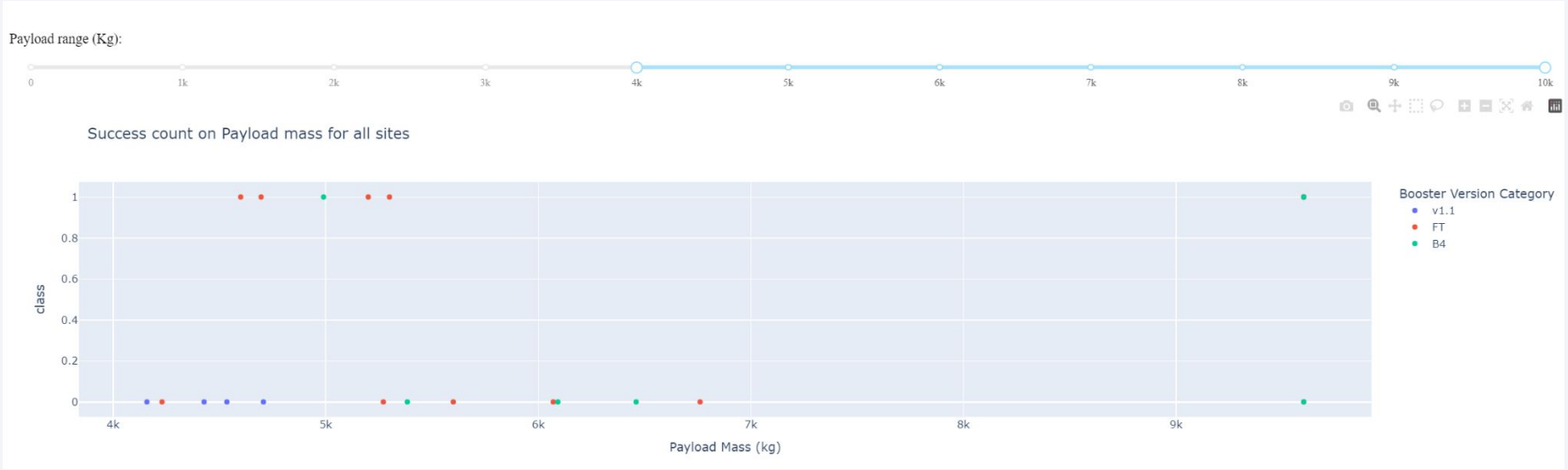
Highest success launch rate



In fact, it is also the one with highest success rate

Payload vs Launch outcome for different boosters



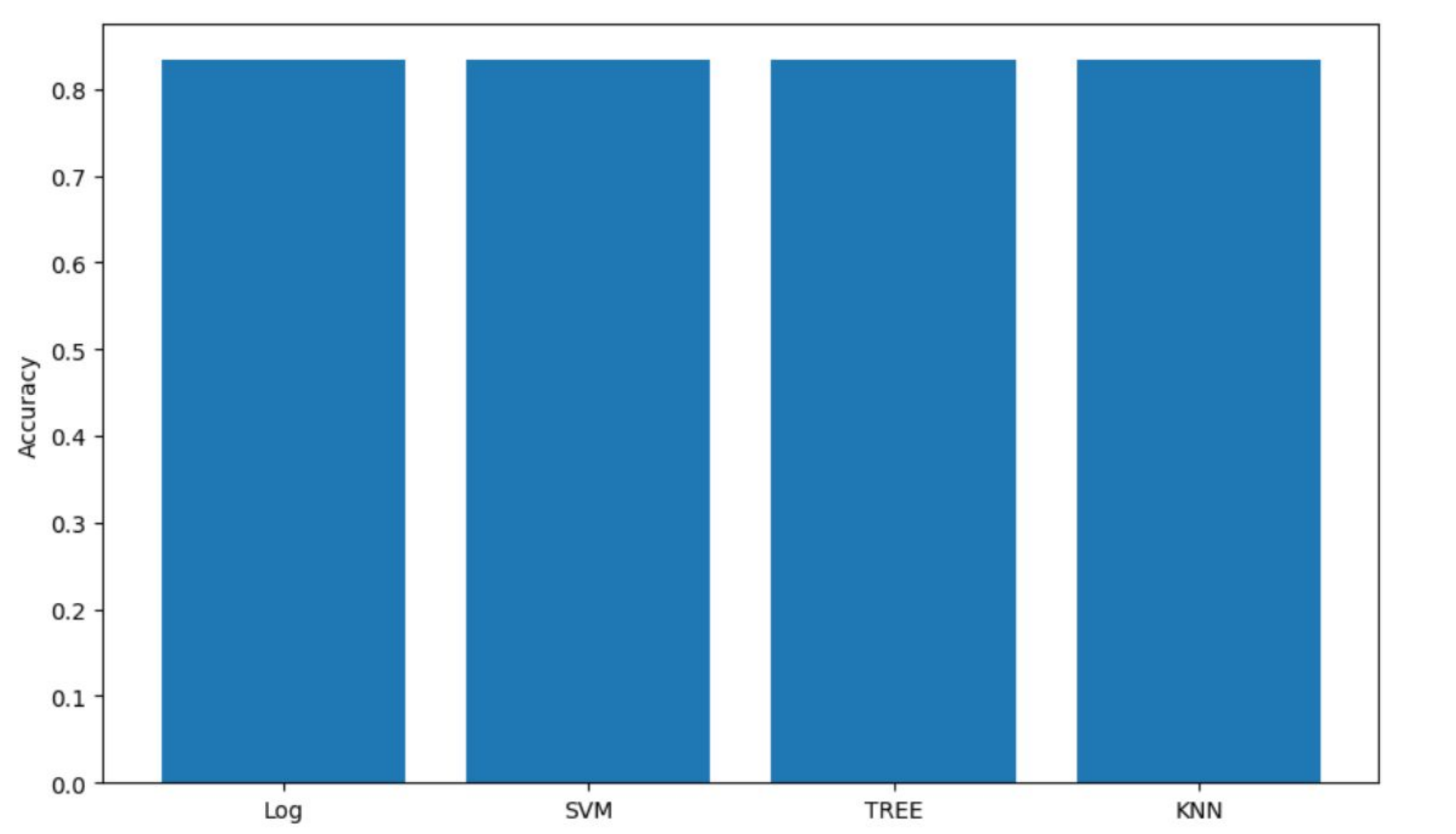


Range between 2500 and 4000 kg is the one with more launches and with high success

Section 5

Predictive Analysis (Classification)

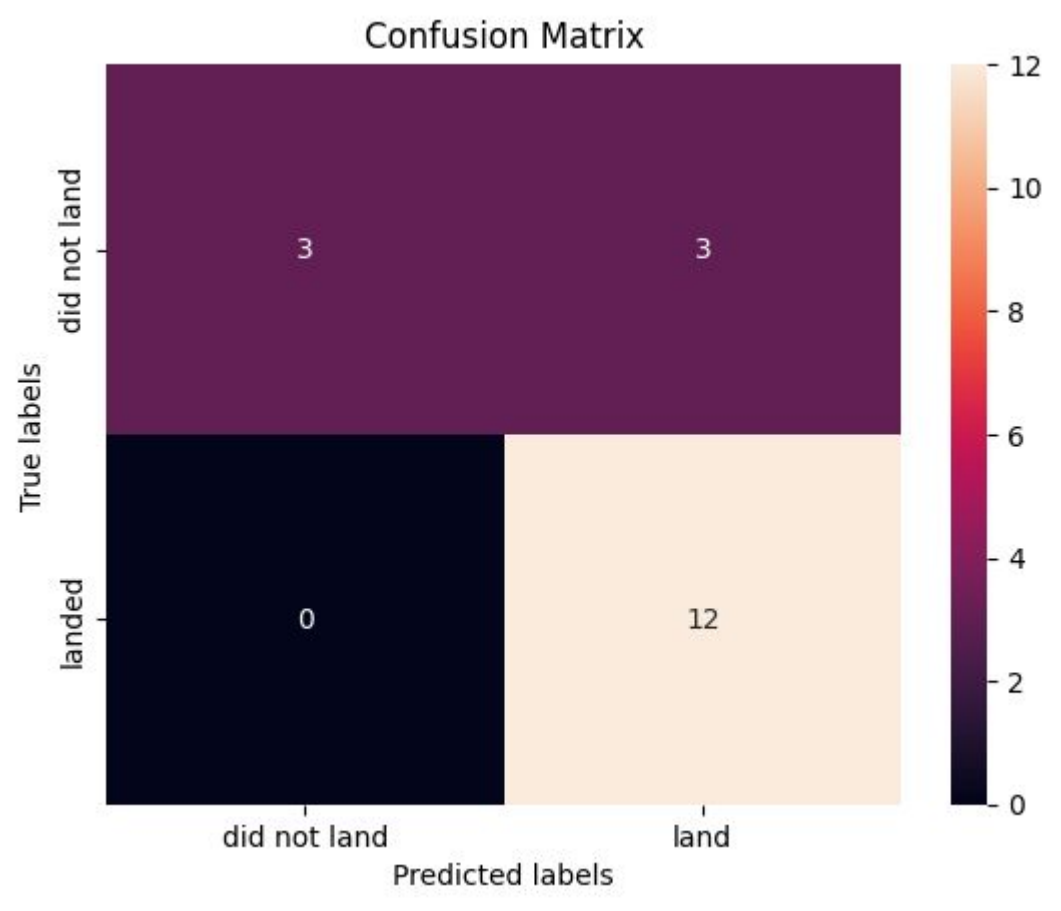
Classification Accuracy



All models have the same accuracy which is quite high.

Confusion Matrix

- All the models performed the same and had the same confusion matrix



Conclusions

- All algorithms have the same accuracy of 0.83.
- We have not such of a big dataset that is why all models perform the same
- The machine learning models have high accuracy and can be use to predict the outcome of the landing. However we have few false positives.

Appendix

https://github.com/MGrauLeguia/coursera/tree/main

Thank you!