IBM Developer
SKILLS NETWORK

# Winning Space Race with Data Science

Bruno Fernández Sánchez-Hermosilla
17-11-2024

# 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 Algorithms

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

# Introduction

- Project background and context

Space X 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 Space X 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 space X for a rocket launch. This goal of the project is to create a machine learning pipeline to predict if the first stage will land successfully.

- Problems you want to find answers
  - What factors determine if the rocket will land successfully?
  - The interaction amongst various features that determine the success rate of a successful landing.
  - What operating conditions needs to be in place to ensure a successful landing program.

Section 1

# Methodology

# Methodology

## Executive Summary

- Data collection methodology:
  - Data was collected using the SpaceX API and web scraping of tables from WikiPedia
- Perform data wrangling
  - To deal with categorical features one-hot encoding was used
- 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.
  - Data collection was done using get request to the SpaceX API and webscraping from wikipedia.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary using different pandas methods.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with the BeautifulSoup library.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

# Data Collection – SpaceX API

- We used the “get” request to the SpaceX API to collect data, clean the requested data and did some basic data formatting.
- The link to the notebook is: https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module1/Module1_1_jupyter-labs-spacex-data-collection-api.ipynb

Now let's start requesting rocket launch data from SpaceX API with the following URL:

```
[6] spacex_url="https://api.spacexdata.com/v4/launches/past"
```

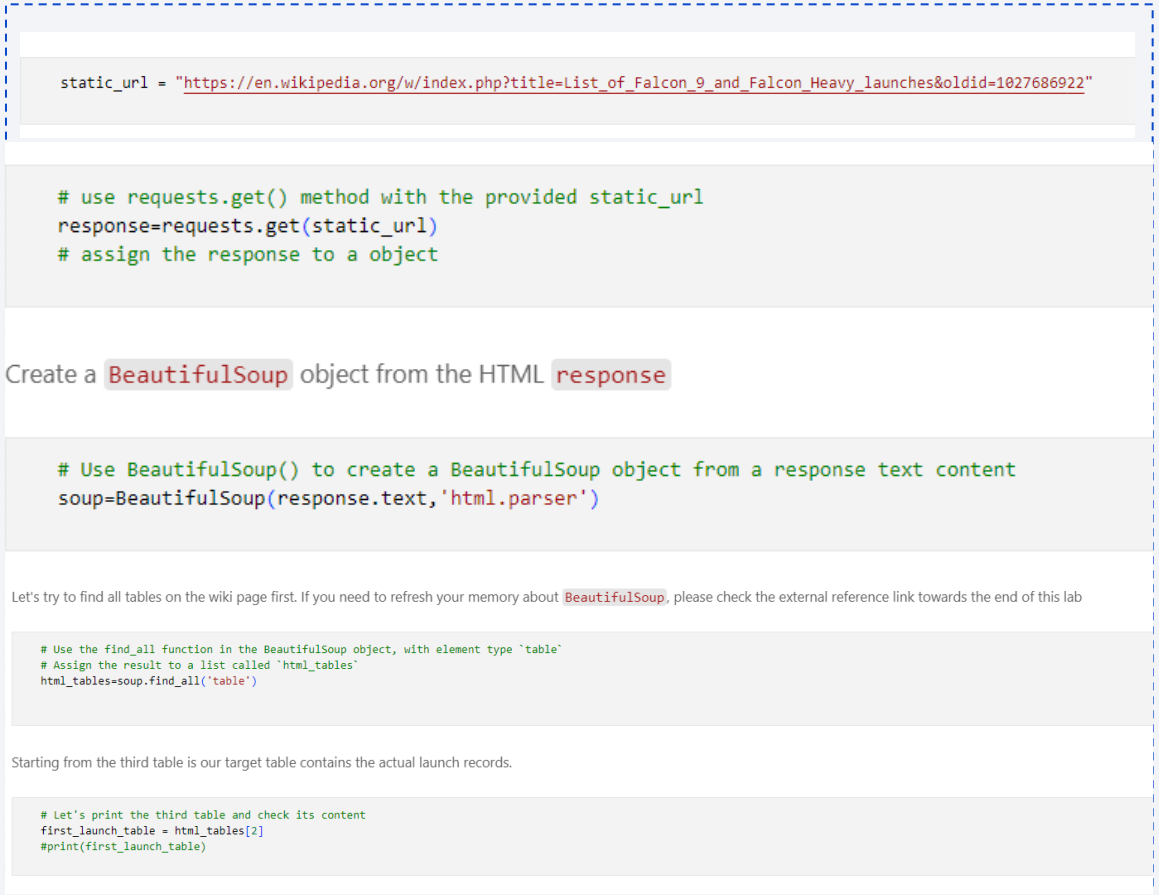
```
[7] response = requests.get(spacex_url)
```

```
# Use json_normalize meethod to convert the json result into a dataframe
response_dec=response.json()
data=pd.json_normalize(response_dec)
```

```
# Calculate the mean value of PayloadMass column
payload_mean=data_falcon9["PayloadMass"].mean()
# Replace the np.nan values with its mean value
data_falcon9["PayloadMass"].replace(np.nan,payload_mean,inplace=True)
data_falcon9.isnull().sum()
```

# Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup on wikipedia
- We converted the table into a pandas dataframe.
- The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module1/Module1_2_jupyter-labs-webscraping.ipynb



```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

```
# use requests.get() method with the provided static_url
response=requests.get(static_url)
# assign the response to a object
```

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

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

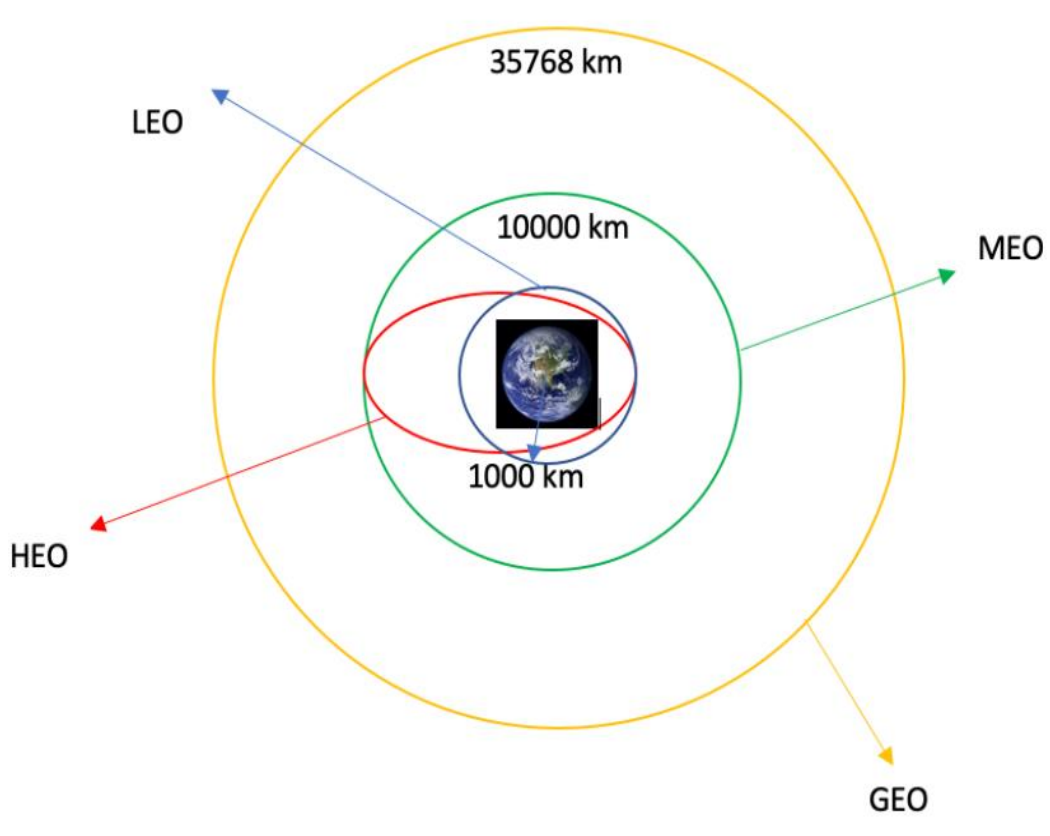
Let's try to find all tables on the wiki page first. If you need to refresh your memory about `BeautifulSoup`, please check the external reference link towards the end of this lab

```
# 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.

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

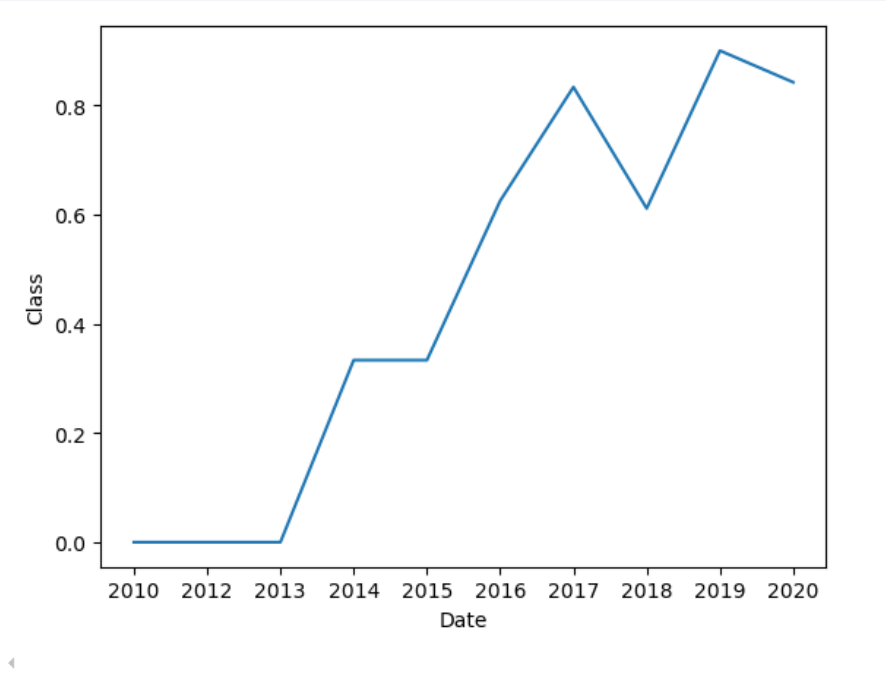
# Data Wrangling

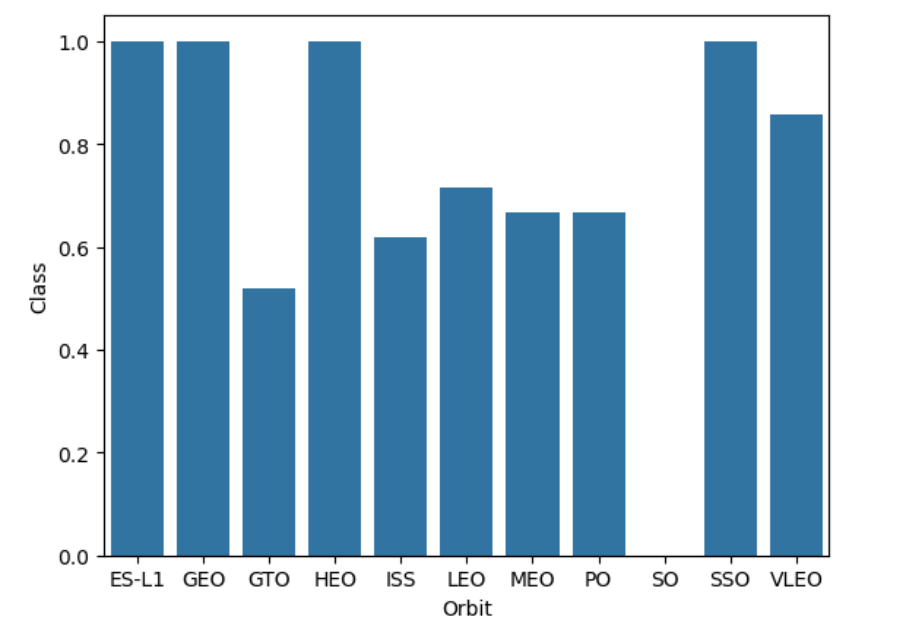



- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module1/Module1_3_labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.




The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module1/Module1_3_labs-jupyter-spacex-Data%20wrangling.ipynb

# EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module2/Module2_1_jupyter-labs-eda-sql-coursera_sqllite.ipynb

# Build an Interactive Map with Folium

We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.

Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

Link to this notebook: https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module3/Module3_1_lab_jupyter_launch_site_location.ipynb

# Build a Dashboard with Plotly Dash

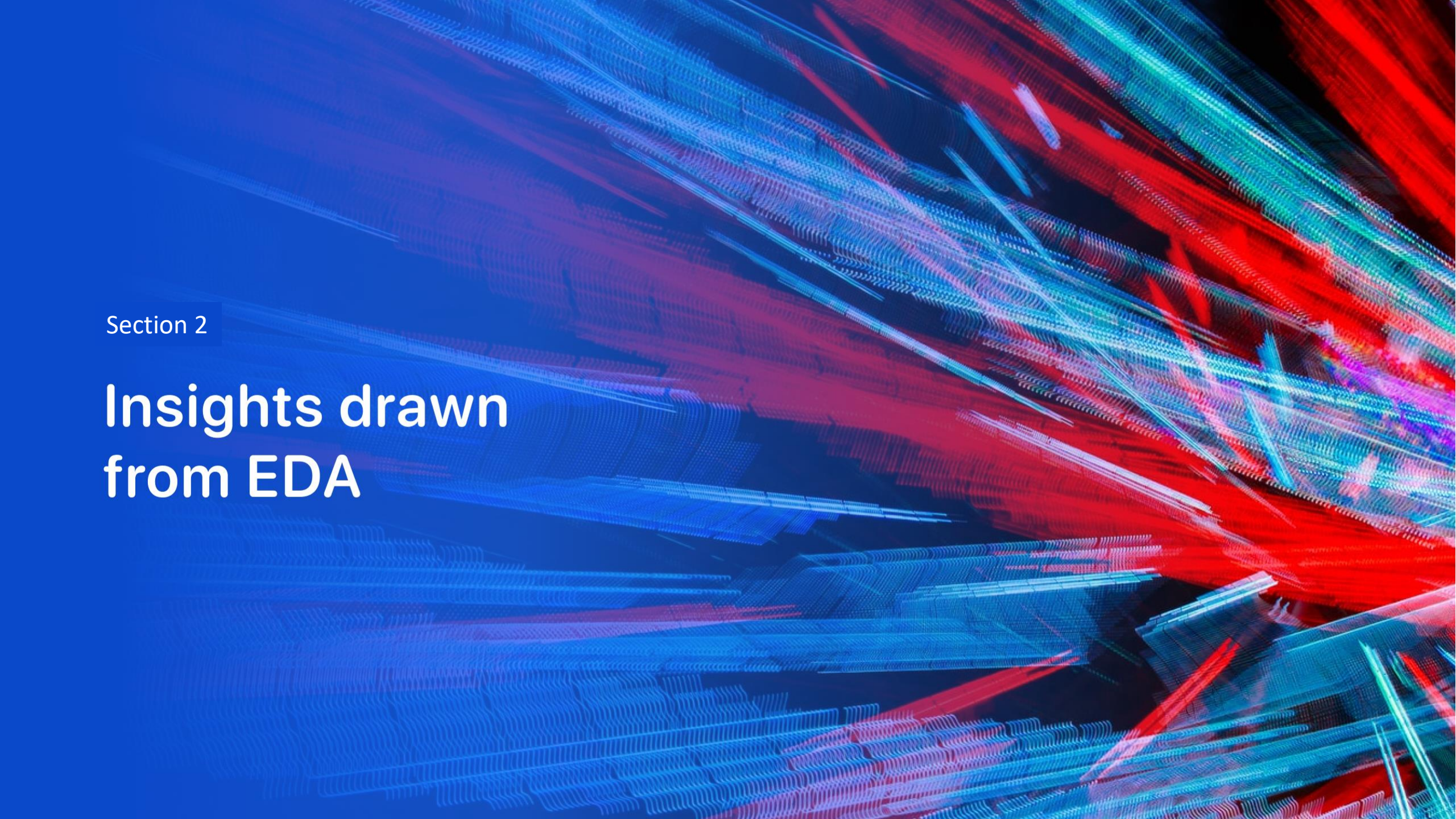
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module3/Module3_2_spacex_dash_app.py

# 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.
- The link to the notebook is https://github.com/Brunofer25/DataScienceIBMcapstone/blob/main/Module5/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

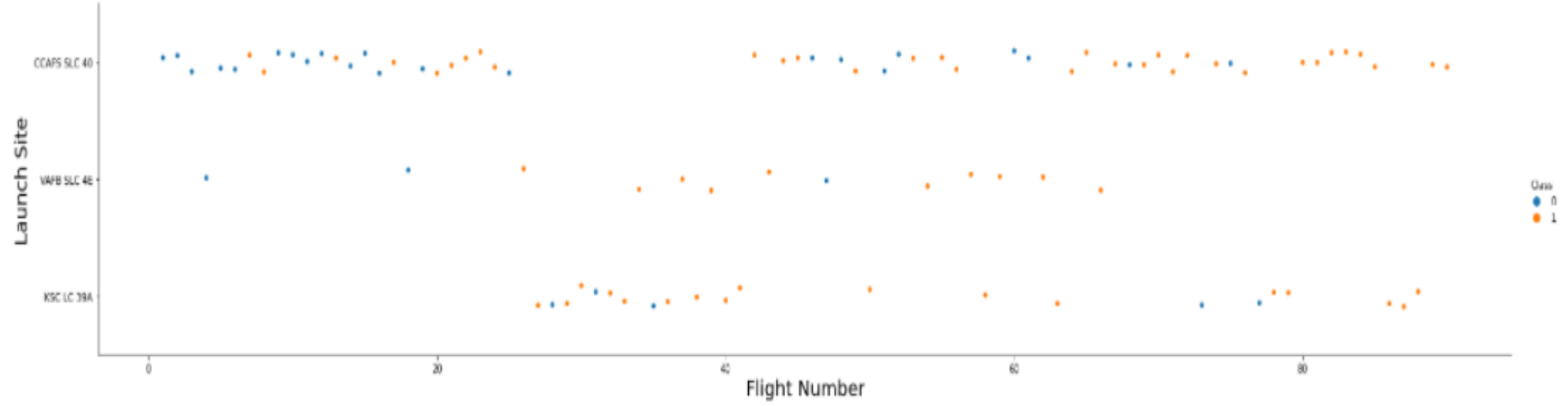
# Results

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


Section 2
Insights drawn from EDA

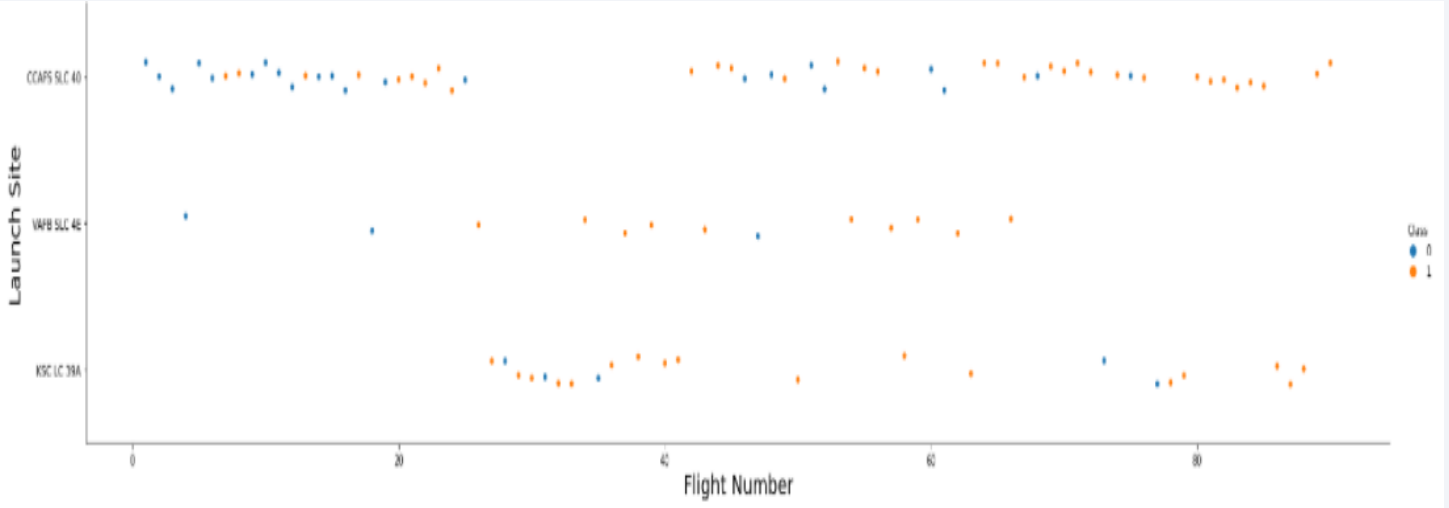
# Flight Number vs. Launch Site

- From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.



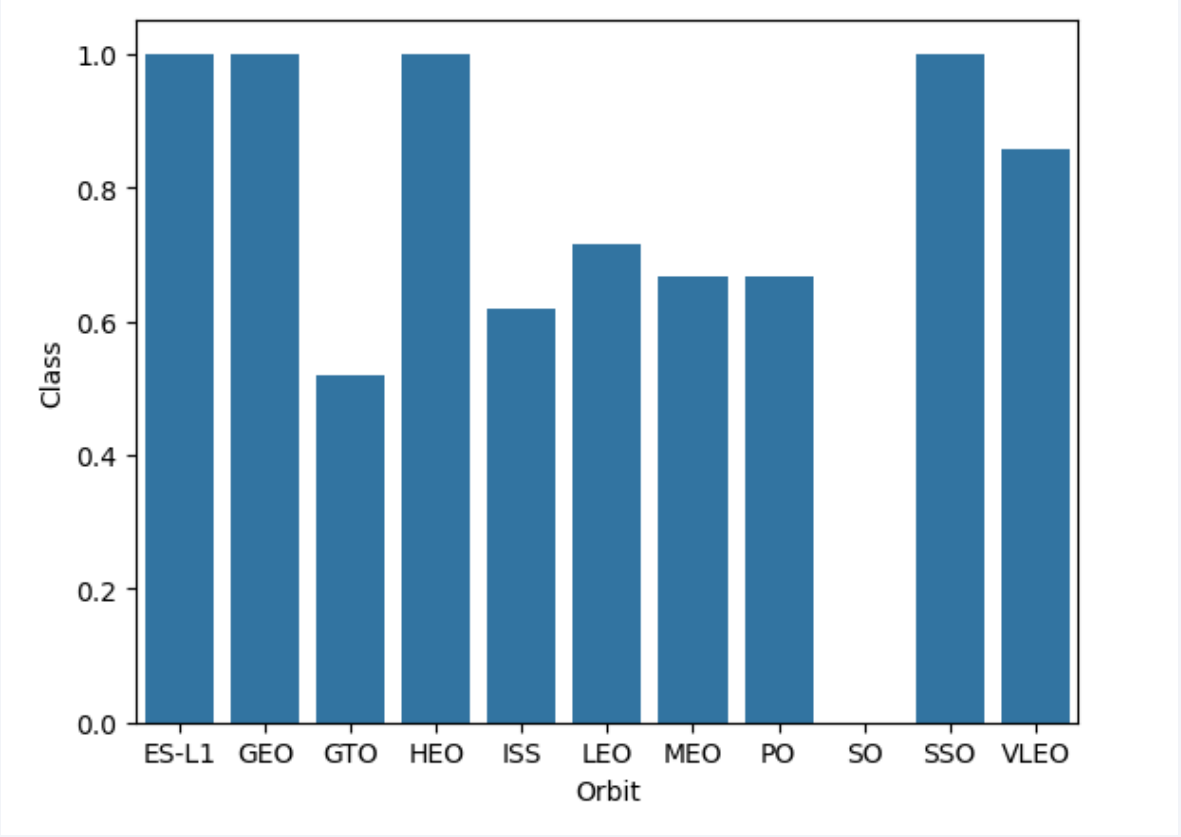
# Payload vs. Launch Site

- The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket



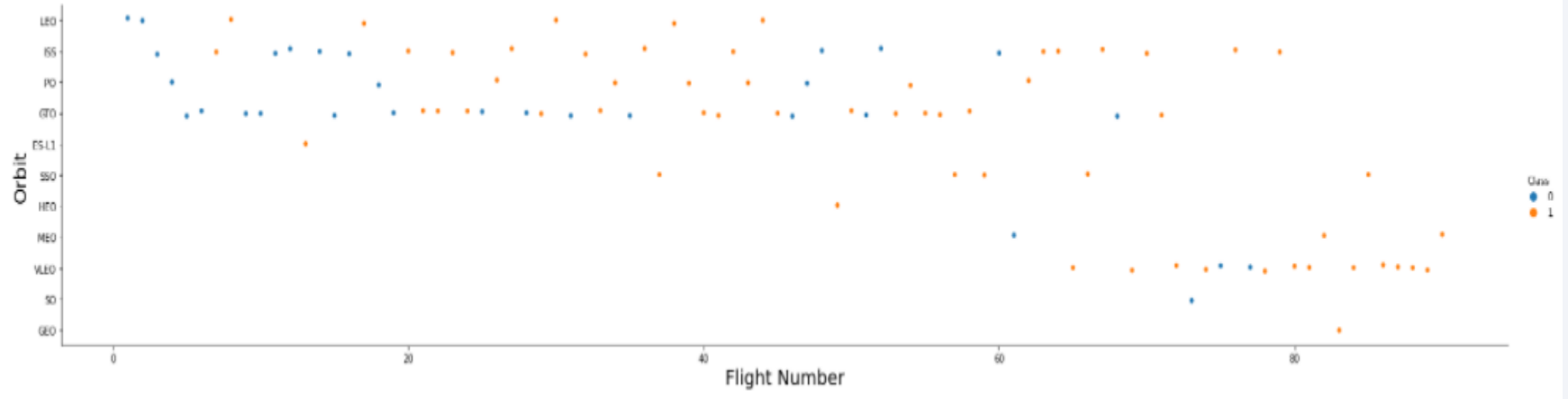
# Success Rate vs. Orbit Type

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



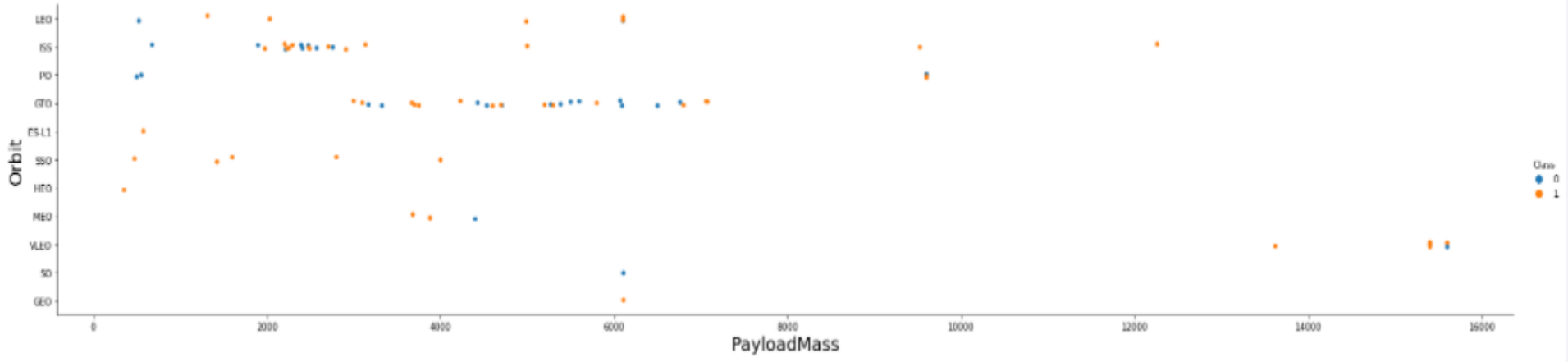
# Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



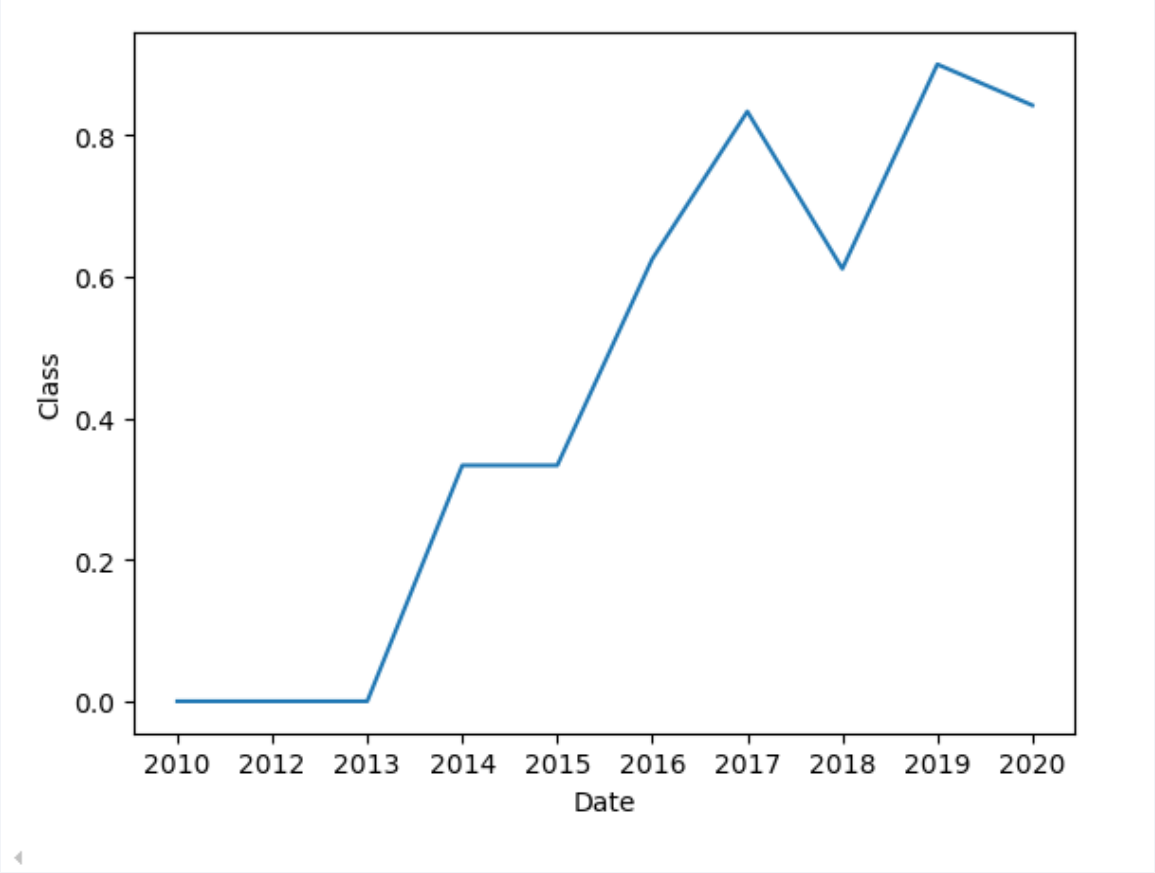
# Payload vs. Orbit Type

- 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

- Using key word DISTINCT we can extract the unique launch sites from the SPACEX table:

```
%%sql
SELECT DISTINCT Launch_Site FROM SPACEXTABLE
```

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

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

# Launch Site Names Begin with 'CCA'

- With LIMIT we limit the results to 5 instances, and with the keyword LIKE ‘CCA%’ we filter the results that start with ‘CCA%’

```
%%sql
SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

Python

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

| Date | Time (UTC) | Booster_Version | Launch_Site | Payload | PAYLOAD_MASS__KG_ |
|---|---|---|---|---|---|
| 2010-06-04 | 18:45:00 | F9 v1.0 B0003 | CCAFS LC-40 | Dragon Spacecraft Qualification Unit | 0 |
| 2010-12-08 | 15:43:00 | F9 v1.0 B0004 | CCAFS LC-40 | Dragon demo flight C1, two CubeSats, barrel of Brouere cheese | 0 |
| 2012-05-22 | 7:44:00 | F9 v1.0 B0005 | CCAFS LC-40 | Dragon demo flight C2 | 525 |
| 2012-10-08 | 0:35:00 | F9 v1.0 B0006 | CCAFS LC-40 | SpaceX CRS-1 | 500 |
| 2013-03-01 | 15:10:00 | F9 v1.0 B0007 | CCAFS LC-40 | SpaceX CRS-2 | 677 |

# Total Payload Mass

- With SUM we sum all the payload mass in KG filtered by LIKE ‘%CRS’

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Payload LIKE '%CRS%'
```

| | total_payloadmass |
|---|---|
| 0 | 45596 |

# Average Payload Mass by F9 v1.1

- WITH AVG we calculate the average mass of the payload in KG filtered by LIKE ‘F9 v1.1%’:

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTABLE WHERE Booster_Version LIKE 'F9 v1.1%'
```

Python

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

| AVG(PAYLOAD_MASS__KG_) |
| --- |
| 2534.6666666666665 |

# First Successful Ground Landing Date

- We select the minimum Date datapoint with MIN, filtered by LIKE “Success (ground pad)”:

```
%%sql
SELECT MIN(Date) FROM SPACEXTABLE WHERE Landing_Outcome LIKE 'Success (ground pad)'
```

Python

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

| MIN(Date) |
| --- |
| 2015-12-22 |

# Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

- Present your query result with a short explanation here

# Total Number of Successful and Failure Mission Outcomes

- We count all the instances with COUNT and we get the count by groups with GROUP BY Mission_Outcome:

```
%%sql
SELECT Mission_Outcome,COUNT(*) FROM SPACEXTABLE GROUP BY Mission_Outcome;
```

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

| Mission_Outcome | COUNT(*) |
|---|---|
| Failure (in flight) | 1 |
| Success | 98 |
| Success | 1 |
| Success (payload status unclear) | 1 |

# Boosters Carried Maximum Payload

- We get the booster_version and payload mass from those who have the maximum payload mass, for that we use a subquery to calculate this MAX

```
%%sql
SELECT Booster_Version,PAYLOAD_MASS__KG_ FROM SPACEXTABLE WHERE PAYLOAD_MASS__KG_==(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

| Booster_Version | PAYLOAD_MASS__KG_ |
|---|---|
| F9 B5 B1048.4 | 15600 |
| F9 B5 B1049.4 | 15600 |
| F9 B5 B1051.3 | 15600 |
| F9 B5 B1056.4 | 15600 |
| F9 B5 B1048.5 | 15600 |
| F9 B5 B1051.4 | 15600 |
| F9 B5 B1049.5 | 15600 |
| F9 B5 B1060.2 | 15600 |
| F9 B5 B1058.3 | 15600 |
| F9 B5 B1051.6 | 15600 |
| F9 B5 B1060.3 | 15600 |
| F9 B5 B1049.7 | 15600 |

# 2015 Launch Records

- We get the month with substr(Date,6,2) and the year with substr(Date,0,5) and we check the Failure by dron ship landing outcome with LIKE:

```
%%sql
SELECT substr(Date,6,2),Booster_Version,Landing_Outcome,Launch_Site FROM SPACEXTABLE WHERE substr(Date,0,5) LIKE '2015' AND Landing_Outcome LIKE 'Failure (drone ship)'
```

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

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

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

- We get all the rows from between the two dates with BETWEEN and then we group by landing outcome:

```
%%sql
SELECT COUNT(*),Landing_Outcome FROM SPACEXTABLE WHERE Date BETWEEN "2010-06-04" AND "2017-03-20" GROUP BY Landing_Outcome
```

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

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

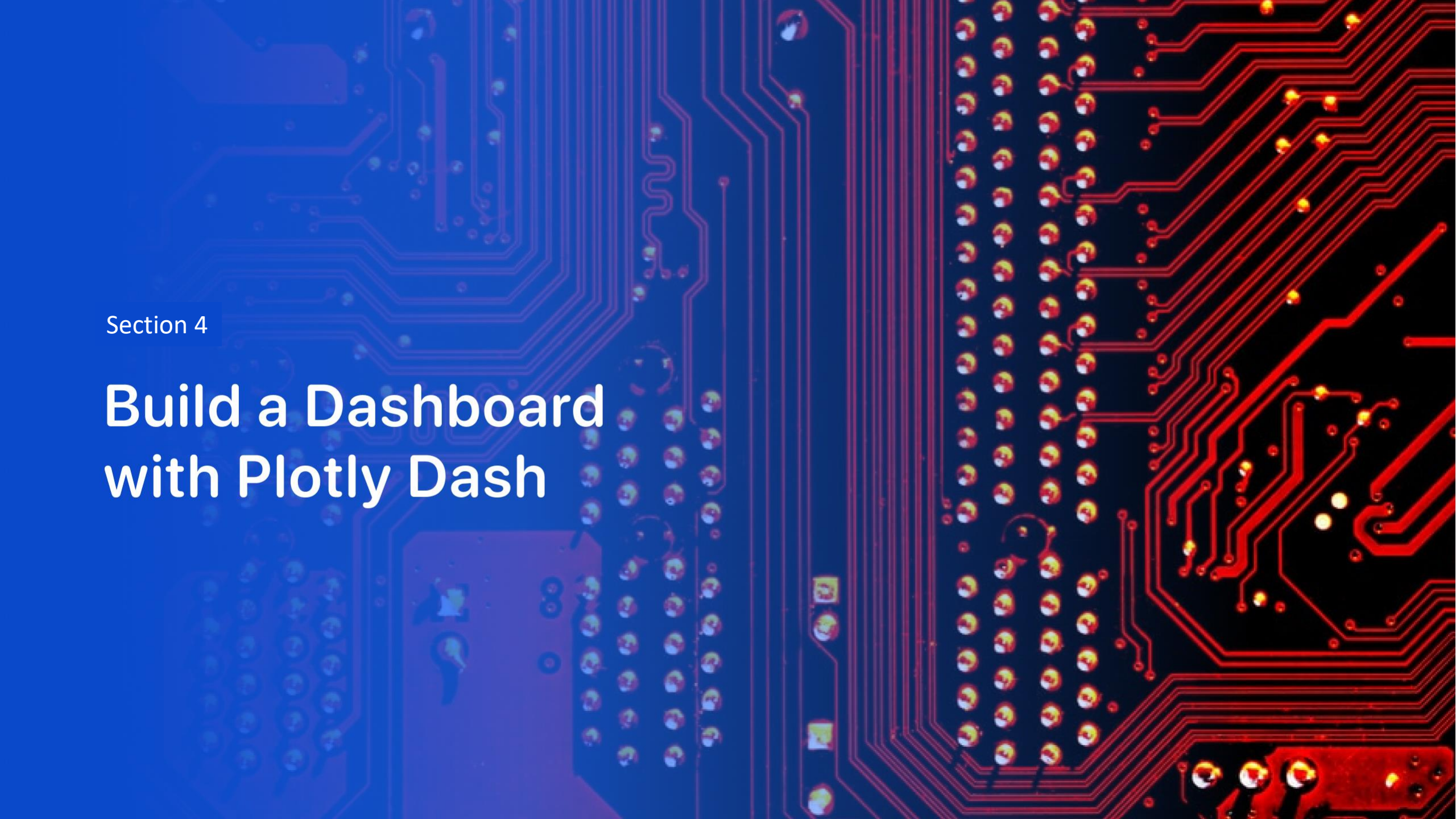
Section 3

# Launch Sites Proximities Analysis

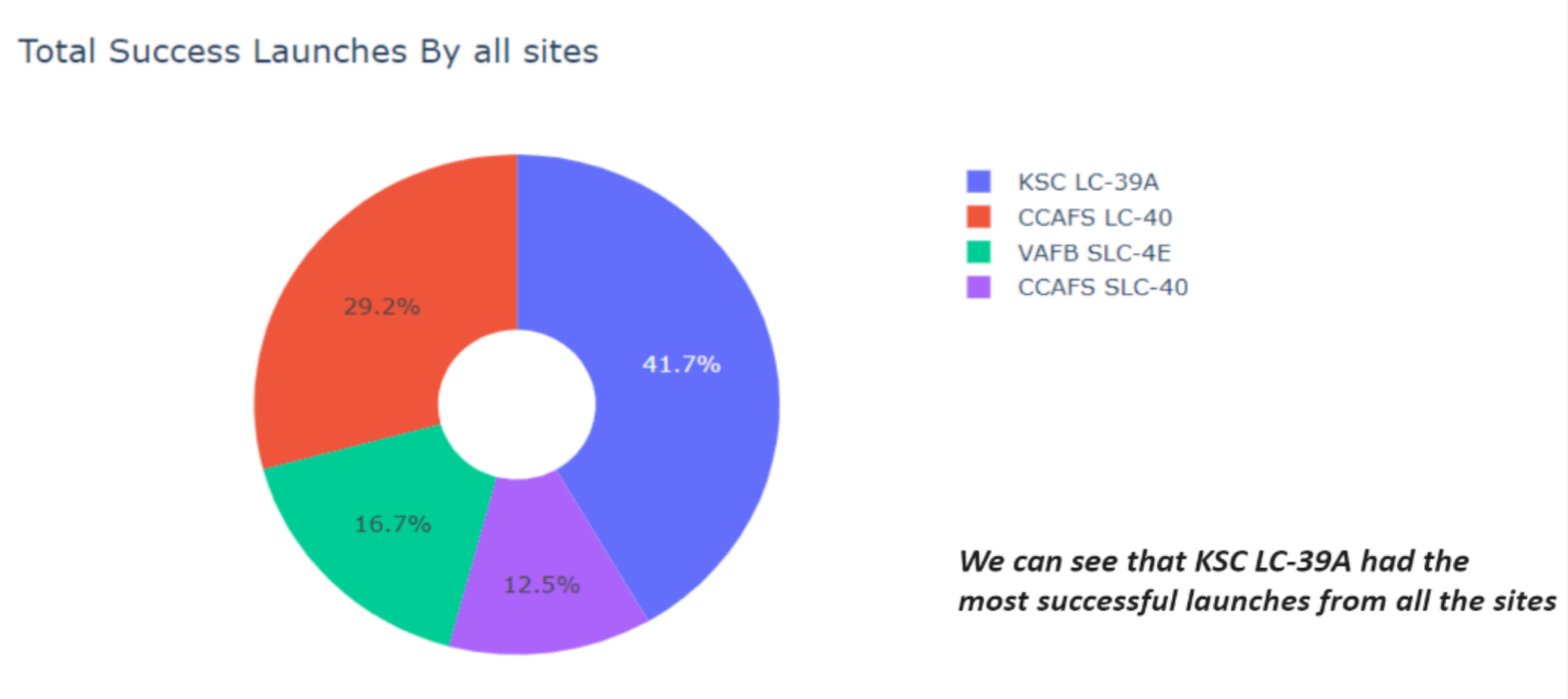
# All launch sites global map markers

# Markers showing launch sites with color labels

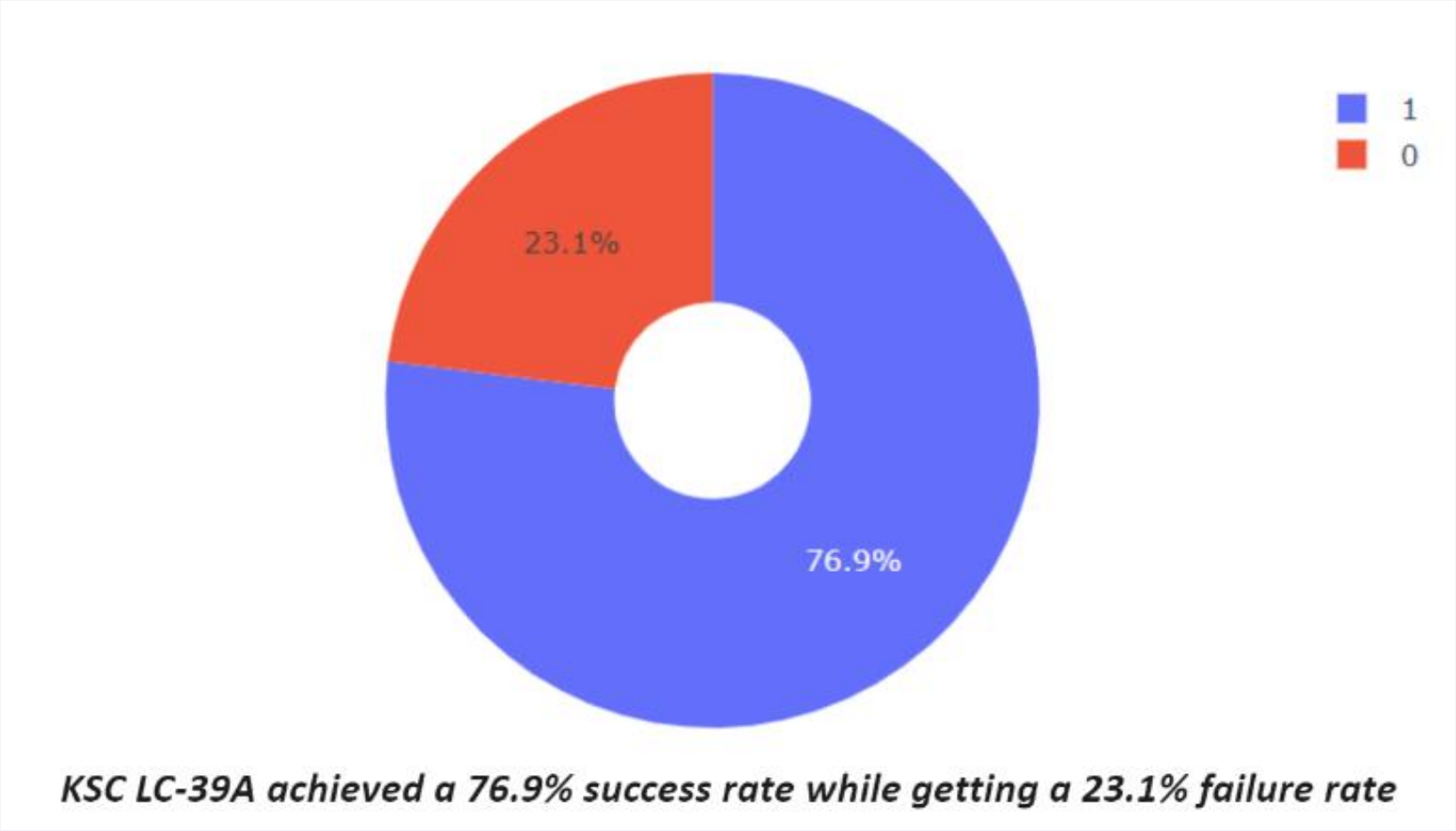
- The green markers represent successful launches and red markers shows failures


Section 4
Build a Dashboard with Plotly Dash

# Pie chart showing the success percentage achieved by each launch site

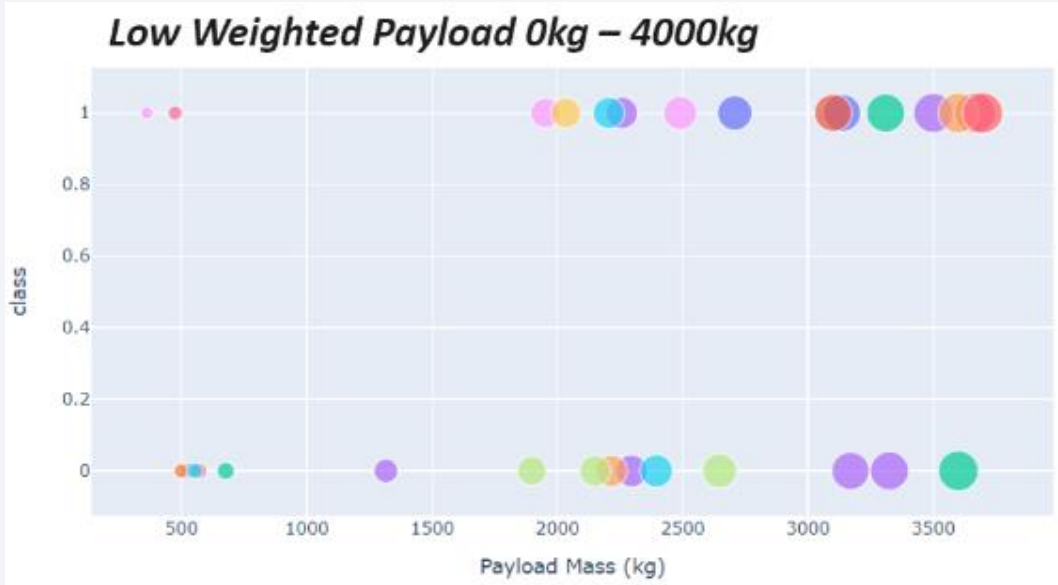


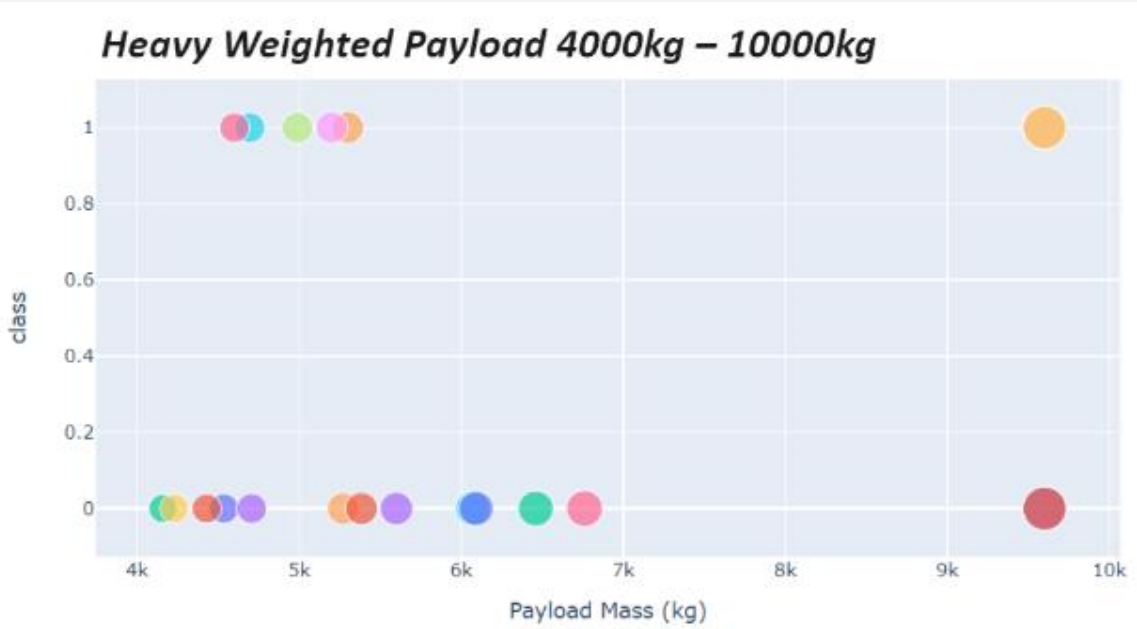
# Pie chart showing the Launch site with the highest launch success ratio




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

# Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in 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

- I could not get to work the Jupyter notebook for this part properly, so I cannot make a thorough analysis.

# Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation

# 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.

# Thank you!