IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Hasan Beker
08.01.2024

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 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

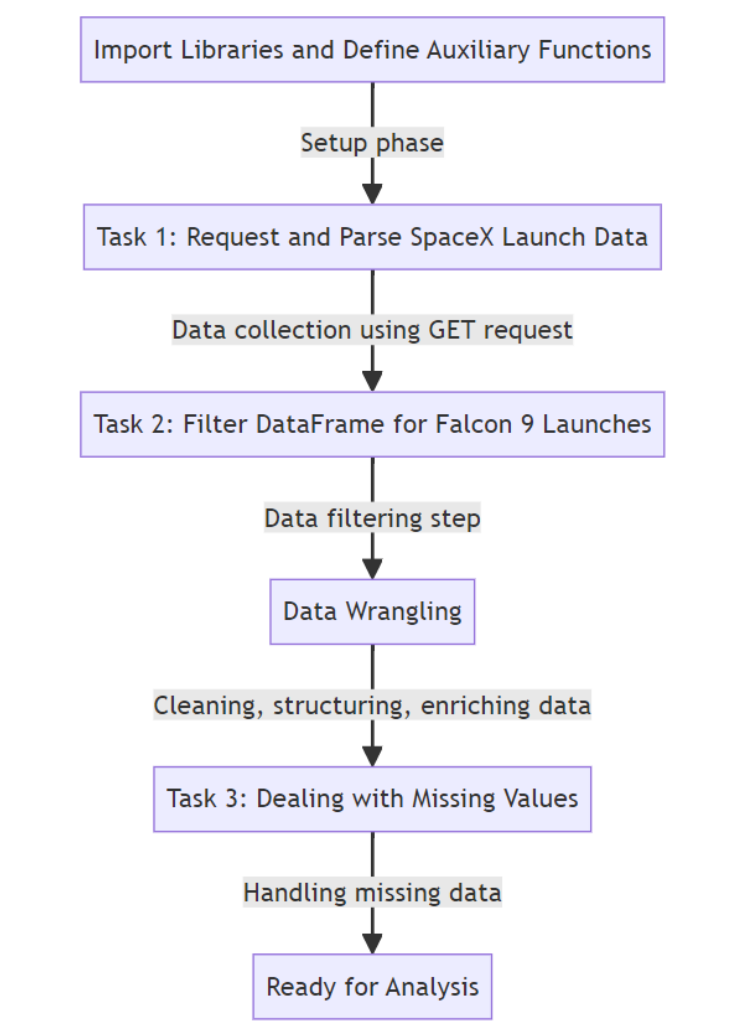
- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- The data was collected using various methods
  - Data collection was done using get request to the SpaceX API.
  - 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.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

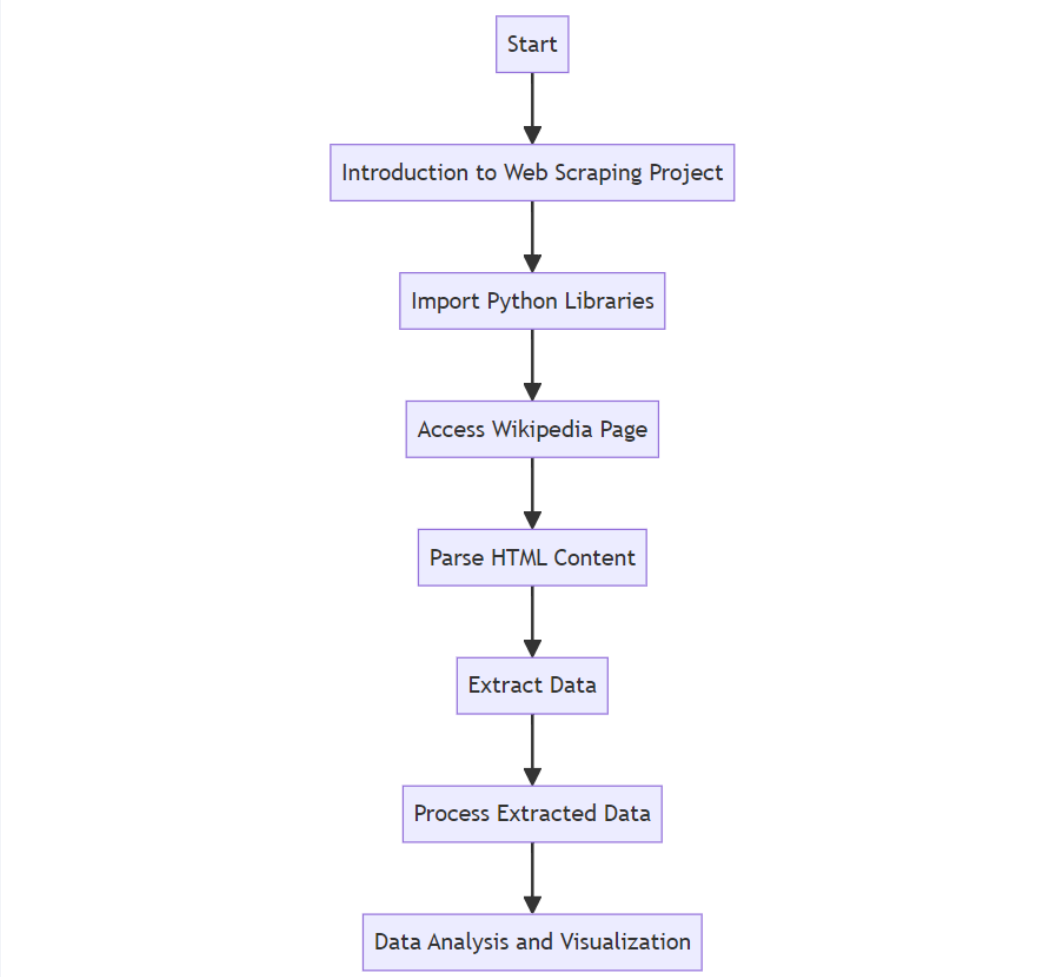
Data Collection – SpaceX API

- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- The link to the notebook is https://github.com/HasanBeker2/Week_1_Introduction_and_Understand_the_Data_Sets/blob/master/jupyter-labs-spacex-data-collection-api.ipynb

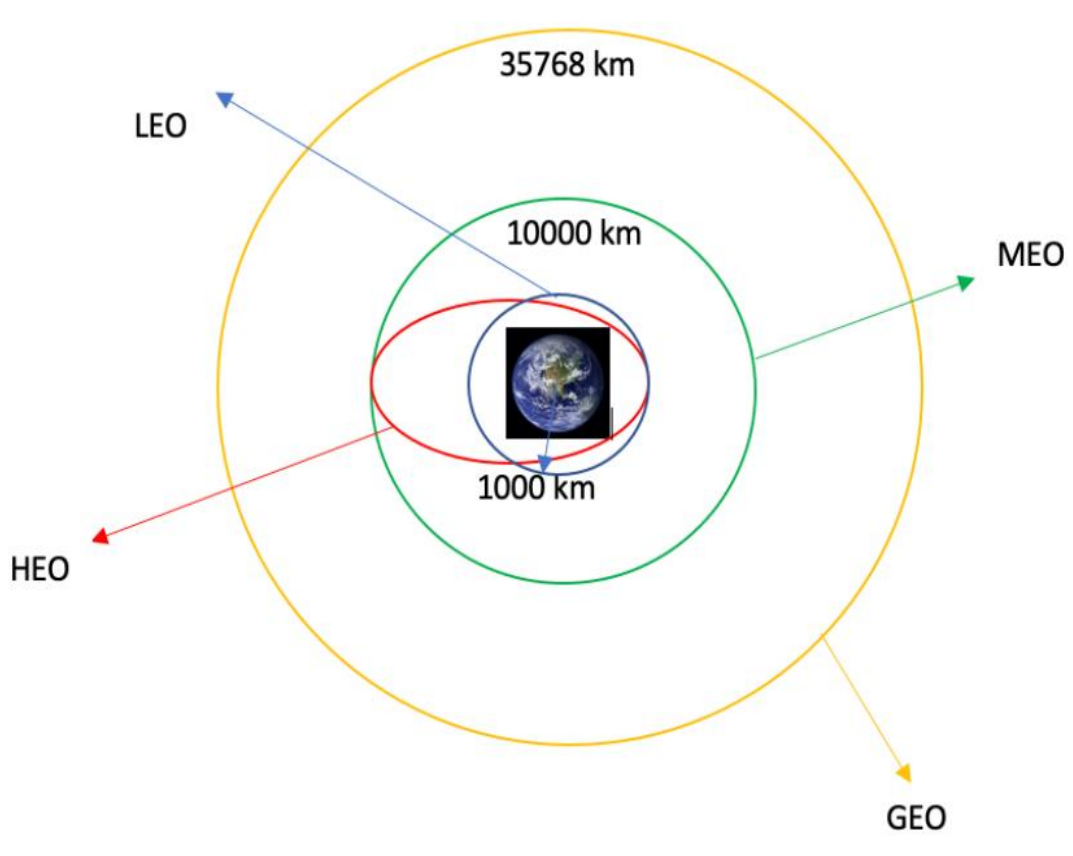


Data Collection - Scraping

- We applied web scrapping to webscrape Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is https://github.com/HasanBeker2/Week_1_Introduction_and_Understand_the_Data_Sets/blob/master/jupyter-labs-webscraping.ipynb



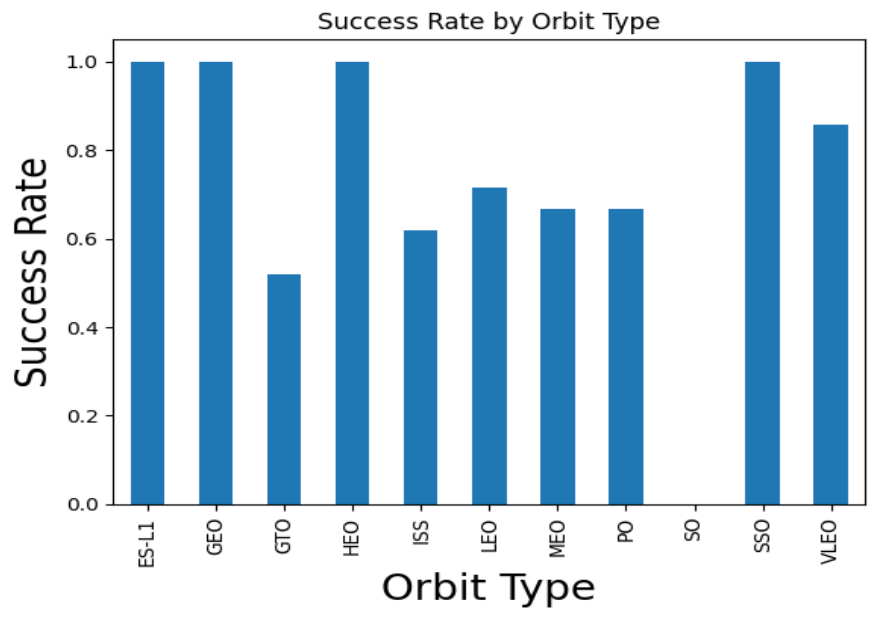
Data Wrangling

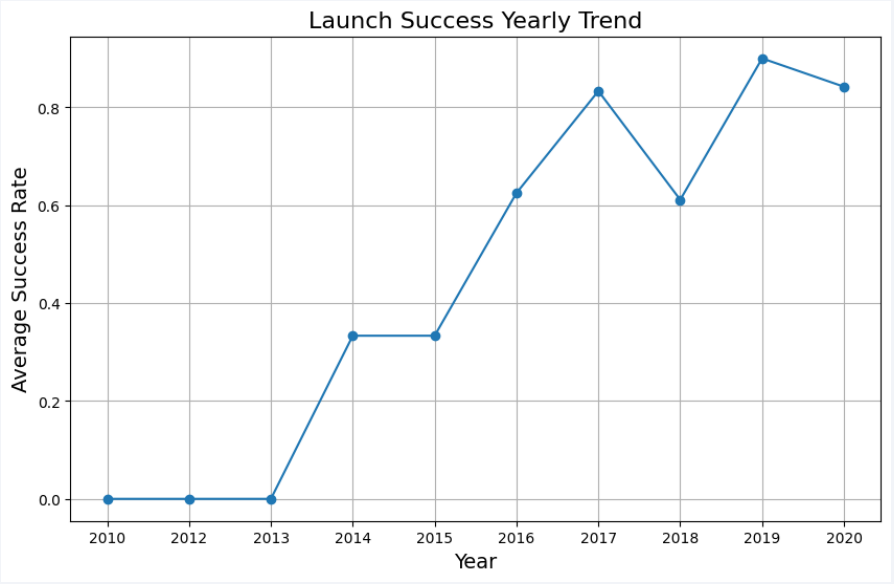


- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is https://github.com/HasanBeker2/Week_1_Introduction_and_Understand_the_Data_Sets/blob/master/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/HasanBeker2/Week_2_Complete-the-EDA-with-SQL/blob/master/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- We loaded the SpaceX dataset into a PostgreSQL database without leaving the jupyter notebook.
- 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/HasanBeker2/Week_2_Complete-the-EDA-with-SQL/blob/master/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.
- We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0 for failure, and 1 for success.
- Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.
- We calculated the distances between a launch site to its proximities. We answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.
- Github link is https://github.com/HasanBeker2/Week_3_Interactive_Visual_Analytics_And_Dashboards/blob/master/dash_interactivity.ipynb

Build a Dashboard with Plotly Dash

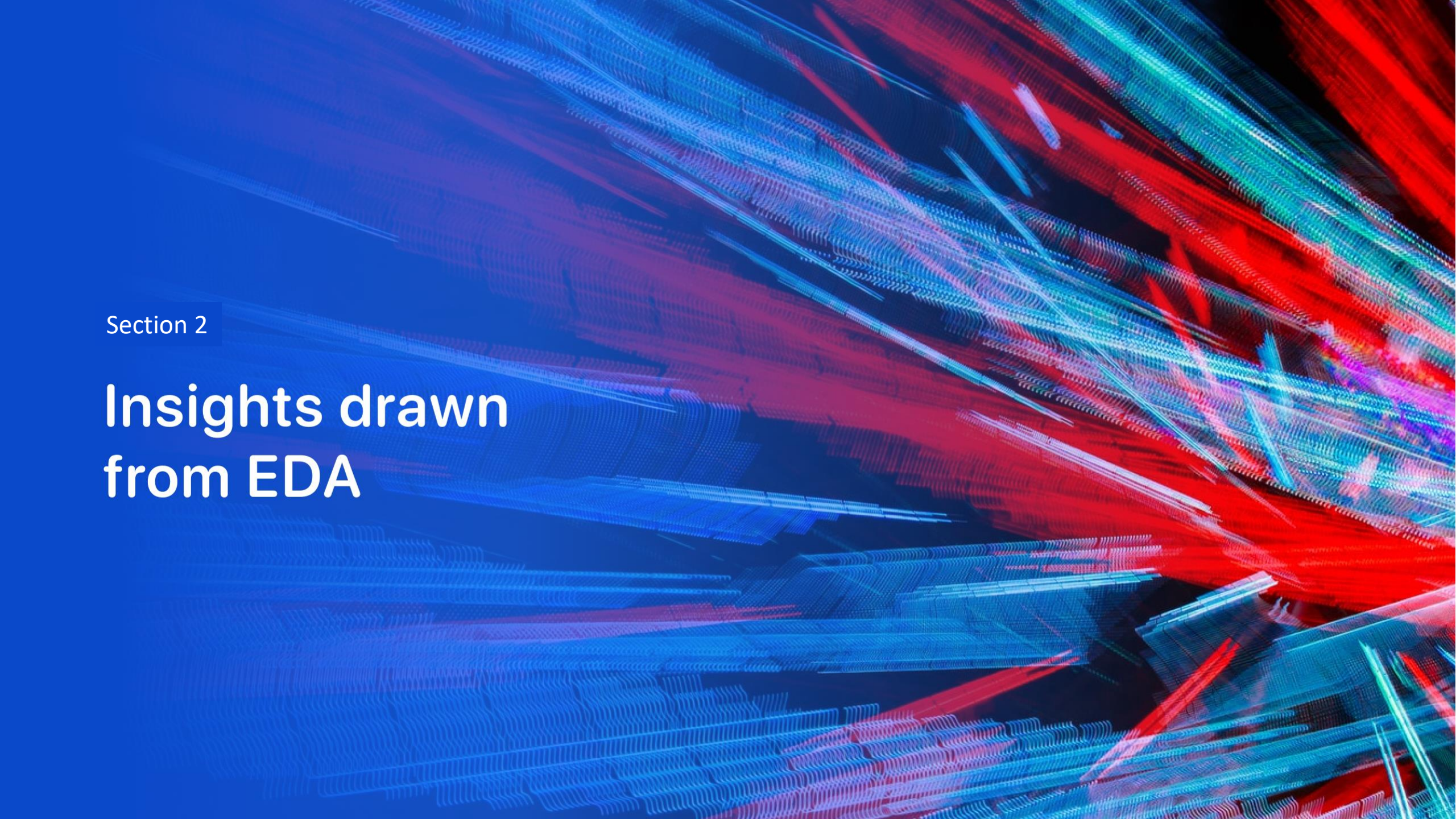
- We built an interactive dashboard with Plotly dash
- We plotted pie charts showing the total launches by a certain sites
- We plotted scatter graph showing the relationship with Outcome and Payload Mass (Kg) for the different booster version.
- The link to the notebook is https://github.com/HasanBeker2/Week_3_Interactive_Visual_Analytics_And_Dashboards/blob/master/dash_interactivity.ipynb

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/HasanBeker2/Week_4_Predictive_Analysis/blob/master/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

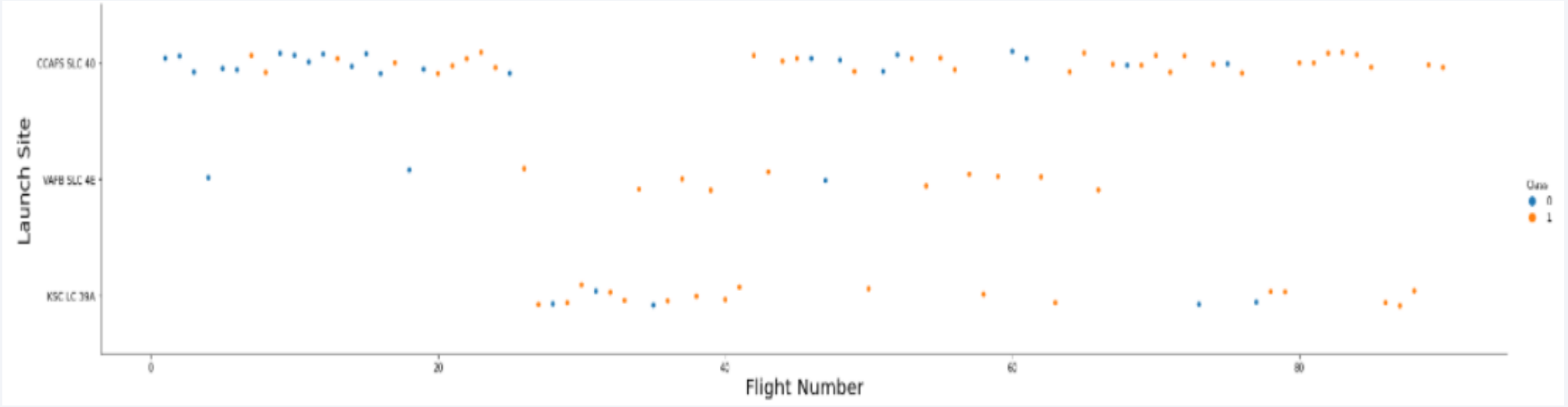
Results

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


Section 2
Insights drawn from EDA

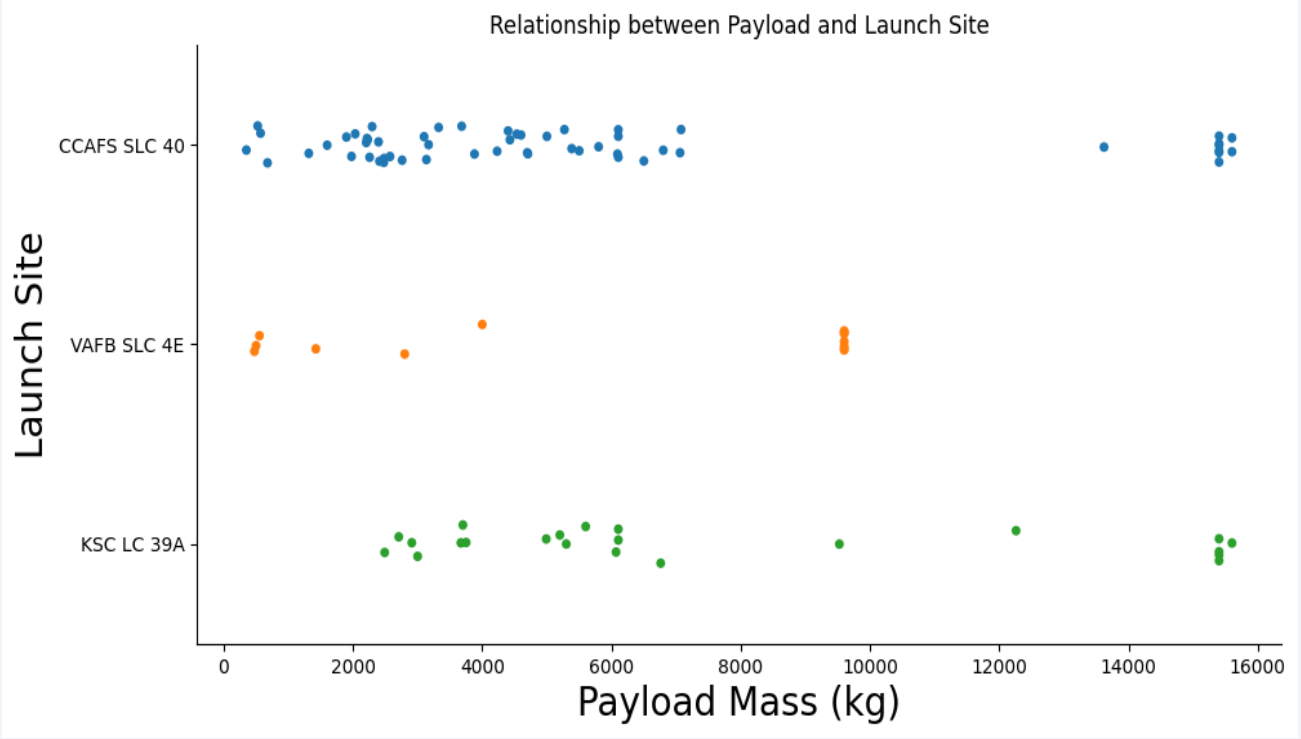
Flight Number vs. Launch Site

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



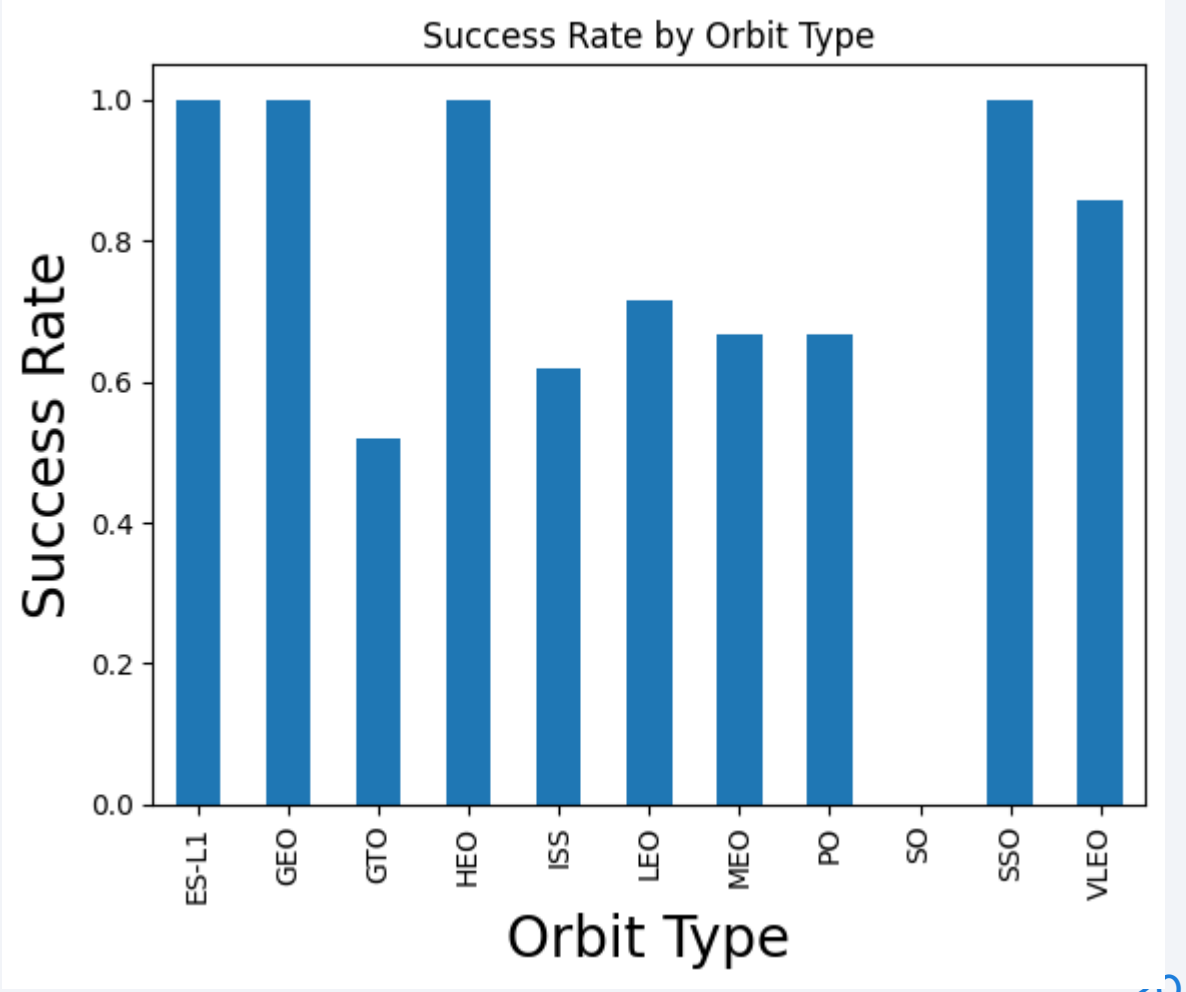
Payload vs. Launch Site

- CCAFS SLC 40 is used for a wide range of payload masses, from small to large.
- VAFB SLC 4E has fewer launches, with the payload masses generally on the lower end compared to the other sites.
- KSC LC 39A is used for launches with a variety of payload masses, though there seems to be a cluster of launches with lower payload masses and a few with very high payload masses.



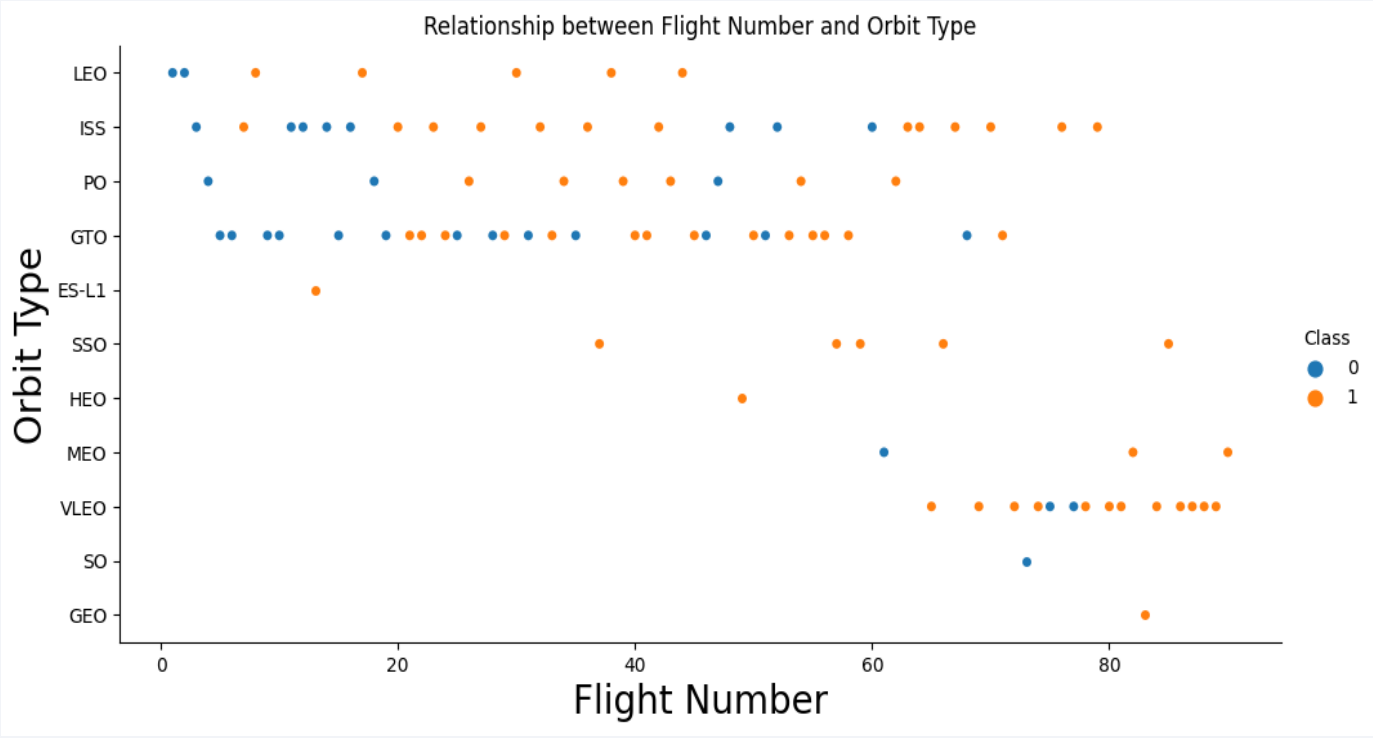
Success Rate vs. Orbit Type

- ES-L1, GEO, and VLEO show the highest success rates, close to 1, suggesting a very high rate of successful missions to these orbits.
- GTO and MEO have success rates of around 0.6 to 0.7, which is lower but still suggests a majority of missions are successful.
- HEO, ISS, LEO, PO, and SSO have success rates between approximately 0.8 and 0.9, indicating a relatively high level of success.
- The success rate for SO is the lowest on the chart, significantly below the others, at about 0.2, indicating that only 20% of missions to this orbit type are successful.



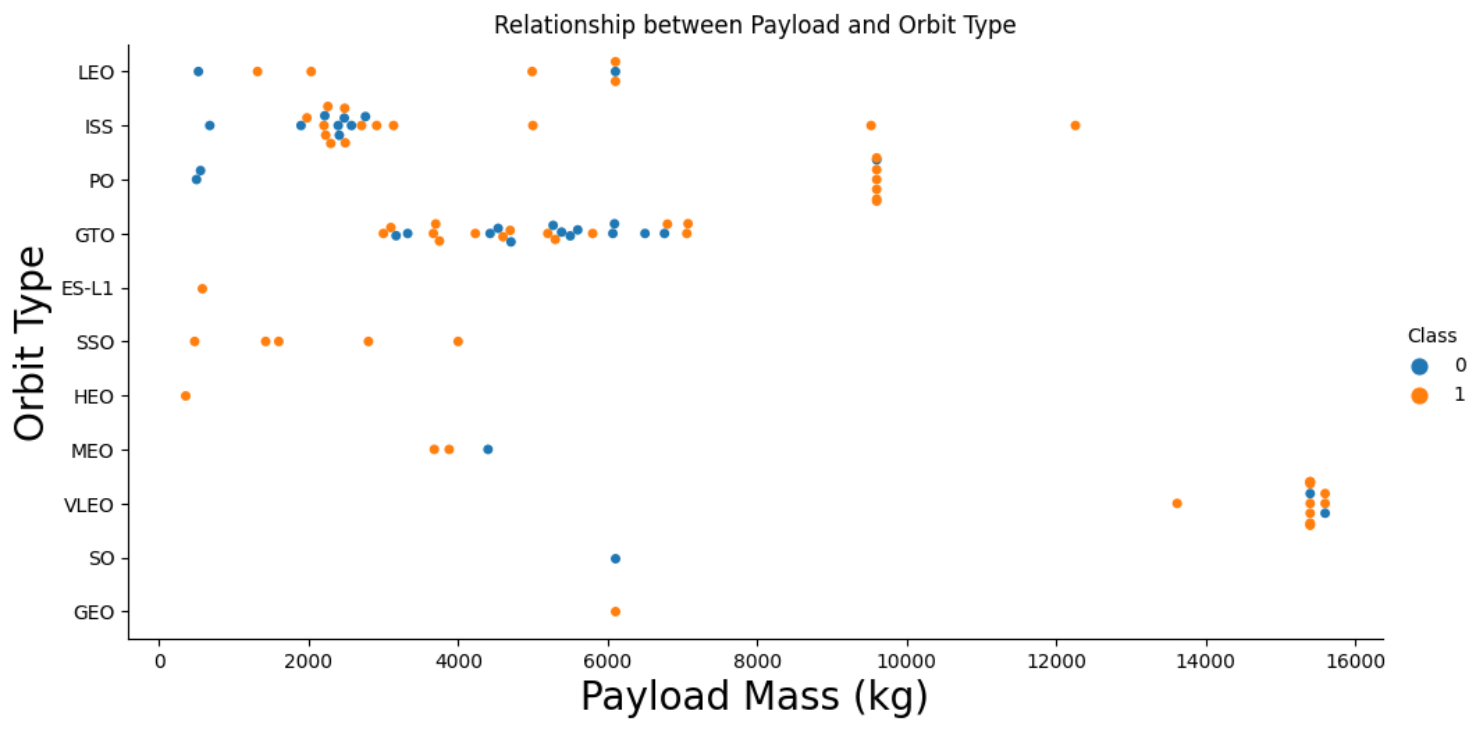
Flight Number vs. Orbit Type

- LEO (Low Earth Orbit) has the highest number of flights spread across the flight numbers, indicating frequent missions to this orbit.
- GEO (Geostationary Orbit) and SO (Suborbital) have the fewest data points, suggesting fewer flights to these orbits.
- Orbits such as ISS (International Space Station), PO (Polar Orbit), and GTO (Geostationary Transfer Orbit) show a moderate number of flights.



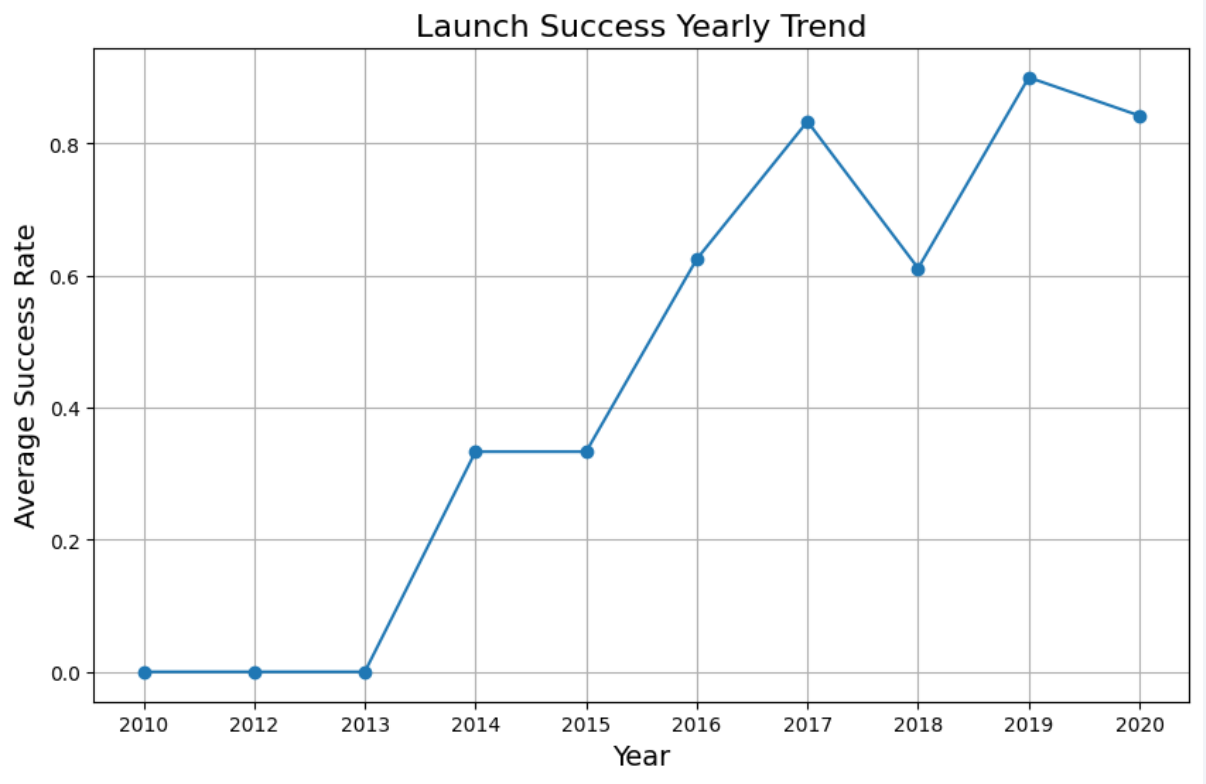
Payload vs. Orbit Type

- Show a scatter point of payloLEO and GTO are the most frequently used orbits for a wide range of payload masses.
- The GEO orbit shows only two instances with heavy payloads, which may indicate its specialized use for larger satellites.ad vs. orbit type
- There are a few instances of very heavy payloads exceeding 10,000 kg, specifically in orbits like GTO and LEO.



Launch Success Yearly Trend

- **Overall Trend**: The graph shows a strong improvement in launch success from nearly 0 in 2010 to over 80% by 2020.
- **Yearly Variability**: Success rates fluctuate notably, especially with a dip after 2016, indicating variable performance across years.
- **Recent Stability**: Success rates level out above 80% from 2018 onwards, suggesting a period of consistent launch reliability.



All Launch Site Names

- We used the key word **DISTINCT** to show only unique launch sites from the SpaceX data.

```
%sql SELECT DISTINCT "Launch_Site" FROM SPACEXTBL;
```

```
 * 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'

- The query was designed to fetch 5 records from a database table named SPACEXTBL where the "Launch_Site" begins with the string 'CCA'.

Task 2

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

```
%sql SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

[12] Python

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

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

- This query sums up the payload mass for all entries in the SPACEXTBL table where the customer is listed as 'NASA (CRS)'. The result of the query is a single value: 45596 kg, which represents the total payload mass carried by all NASA (CRS) missions in the dataset.

Task 3

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

```
[14] %sql SELECT SUM(PAYLOAD_MASS__KG_) AS total_payload_mass FROM SPACEXTBL WHERE Customer = 'NASA (CRS)';
```

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

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- This query determines the average (mean) value of payload mass in kilograms for all launches that used the specific booster version 'F9 v1.1'. According to the result displayed, the average payload mass for this booster version is 2928.4 kg.

Task 4

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

```
[15] %sql SELECT AVG(PAYLOAD_MASS__KG_) AS average_payload_mass FROM SPACEXTBL WHERE Booster_Version = 'F9 v1.1';
```

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

average_payload_mass
2928.4

First Successful Ground Landing Date

- The MIN(Date) function is applied to select the earliest date from the table SPACEXTBL where the landing outcome was a success on a ground pad. The result of this query is the date 2015-12-22, indicating that the first successful ground pad landing occurred on December 22, 2015.

Task 5

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

Hint:Use min function

```
%sql SELECT MIN(Date) AS first_successful_landing_date FROM SPACEXTBL WHERE Landing_Outcome = 'Success (ground pad)';
```

16]

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

first_successful_landing_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- This indicates that these four booster versions successfully landed on a drone ship and were carrying payloads within the specified mass range.

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

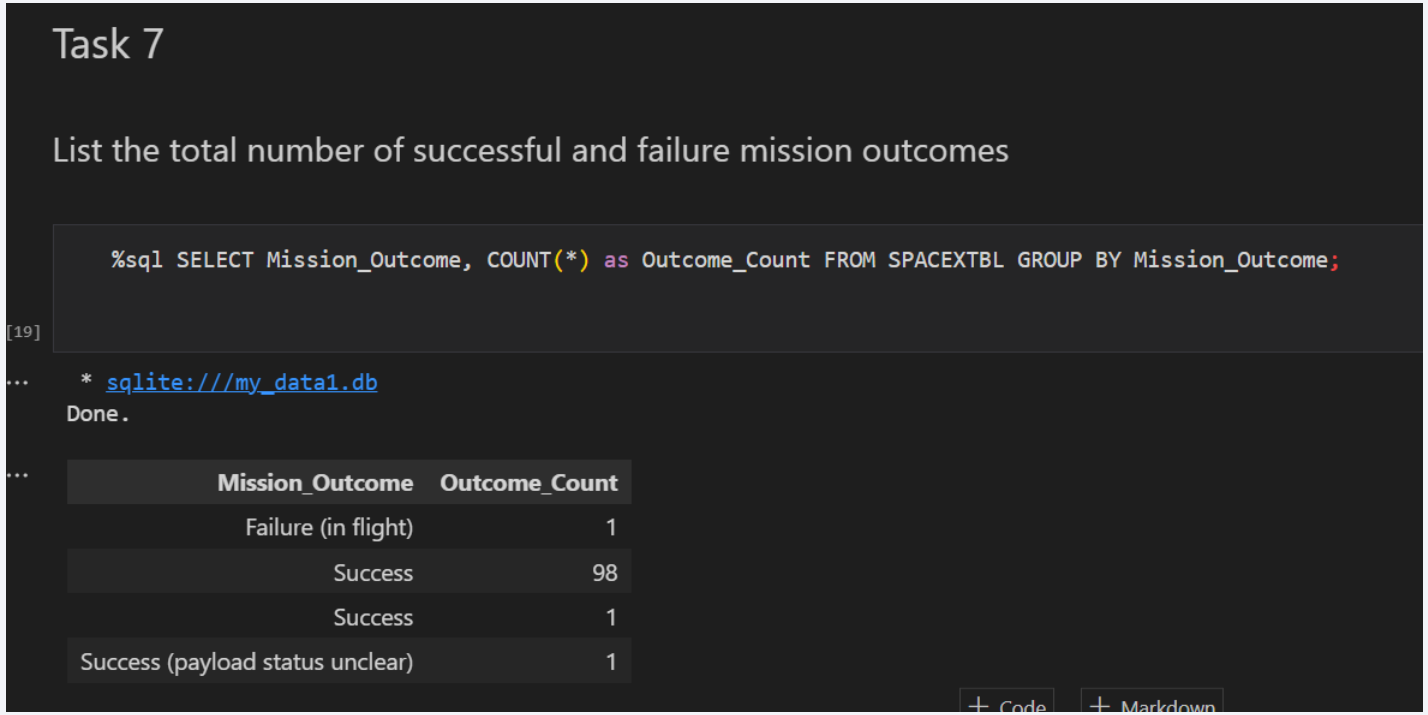
```
[18] %sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

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

Total Number of Successful and Failure Mission Outcomes

- This SQL statement classifies each mission outcome as 'Success' if the Mission_Outcome column contains the word "Success", as 'Failure' if it contains the word "Failure", and as 'Other' for all other outcomes. Then, it counts the number of occurrences for each category.



Task 7

List the total number of successful and failure mission outcomes

```
%sql SELECT Mission_Outcome, COUNT(*) as Outcome_Count FROM SPACEXTBL GROUP BY Mission_Outcome;
```

[19]

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

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

+ Code + Markdown

Boosters Carried Maximum Payload

- The result is a list of booster versions that have carried payloads equal to the maximum payload mass recorded in the database. The list includes several versions of the Falcon 9 Block 5 booster, indicating that these specific versions have been used to launch the heaviest payloads.

Task 8

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

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

[21]

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

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

2015 Launch Records

- The query then filters for landing outcomes that were failures on a drone ship. The results shown include two records from January and April of 2015, both with failures in drone ship landings.

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.

```
%sql SELECT SUBSTR(Date, 6, 2) AS month, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE SUBSTR(Date, 1, 4) = '2015' AND Landing_Outcome
= 'Failure (drone ship)';
```

[25]

Pytho

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

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

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

- The resulting data shows that 'No attempt' at landing is the most common outcome, with 10 occurrences. This is followed by an equal number of 'Success (drone ship)' and 'Failure (drone ship)' outcomes, each with 5 occurrences. There are also instances of successful ground pad landings, controlled and uncontrolled ocean landings, and a couple of failures related to parachutes, with a single instance of a 'Precluded (drone ship)' outcome.

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.

```
%sql SELECT Landing_Outcome, COUNT(*) as Outcome_Count FROM SPACEXTBL WHERE Date BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY Landing_Outcome ORDER BY
Outcome_Count DESC;
```

[26] Python

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

Landing_Outcome	Outcome_Count
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

Section 3

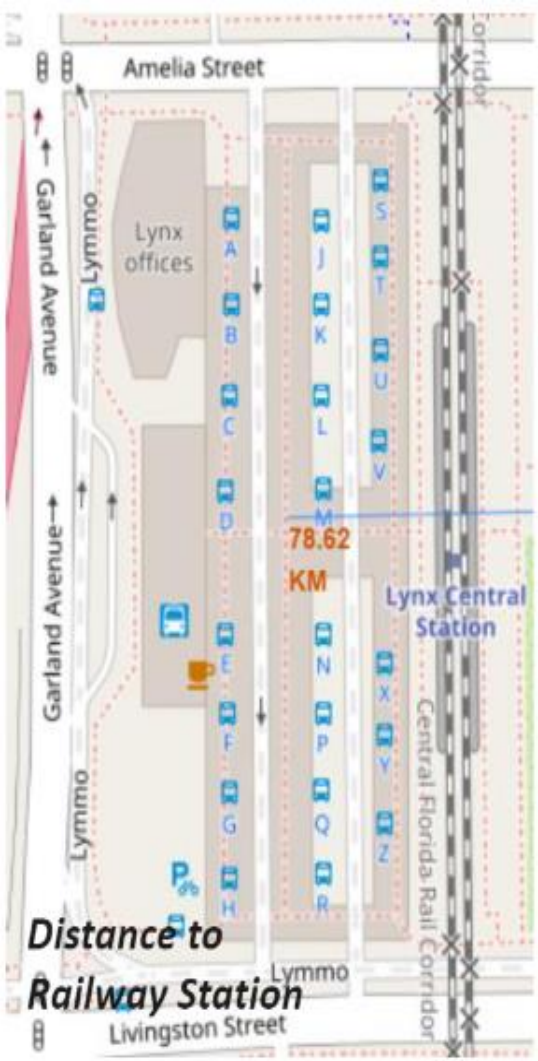
Launch Sites Proximities Analysis

All launch sites global map markers

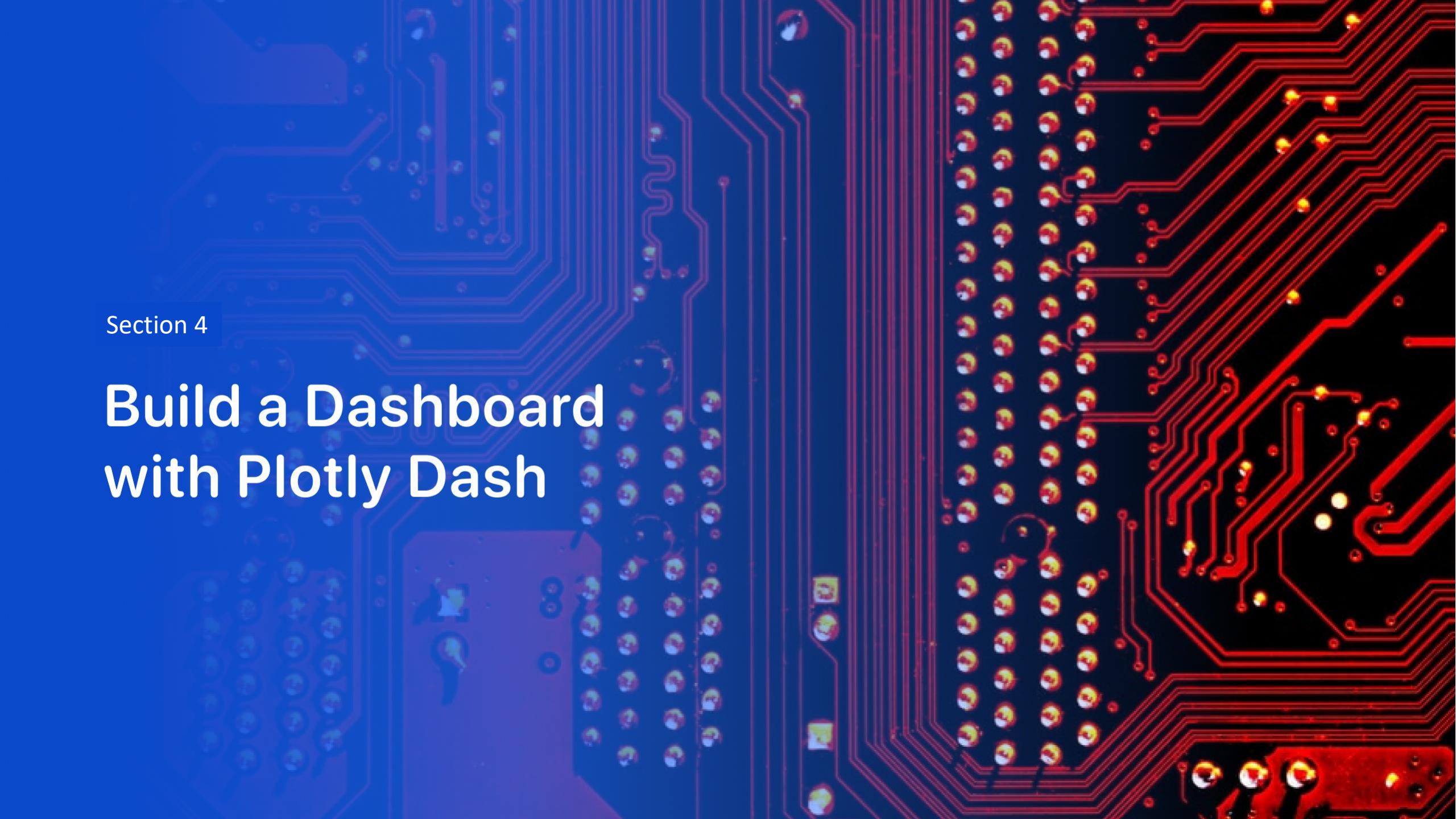
We can see that the SpaceX launch sites are in the United States of America coasts. Florida and California

Markers showing launch sites with color labels

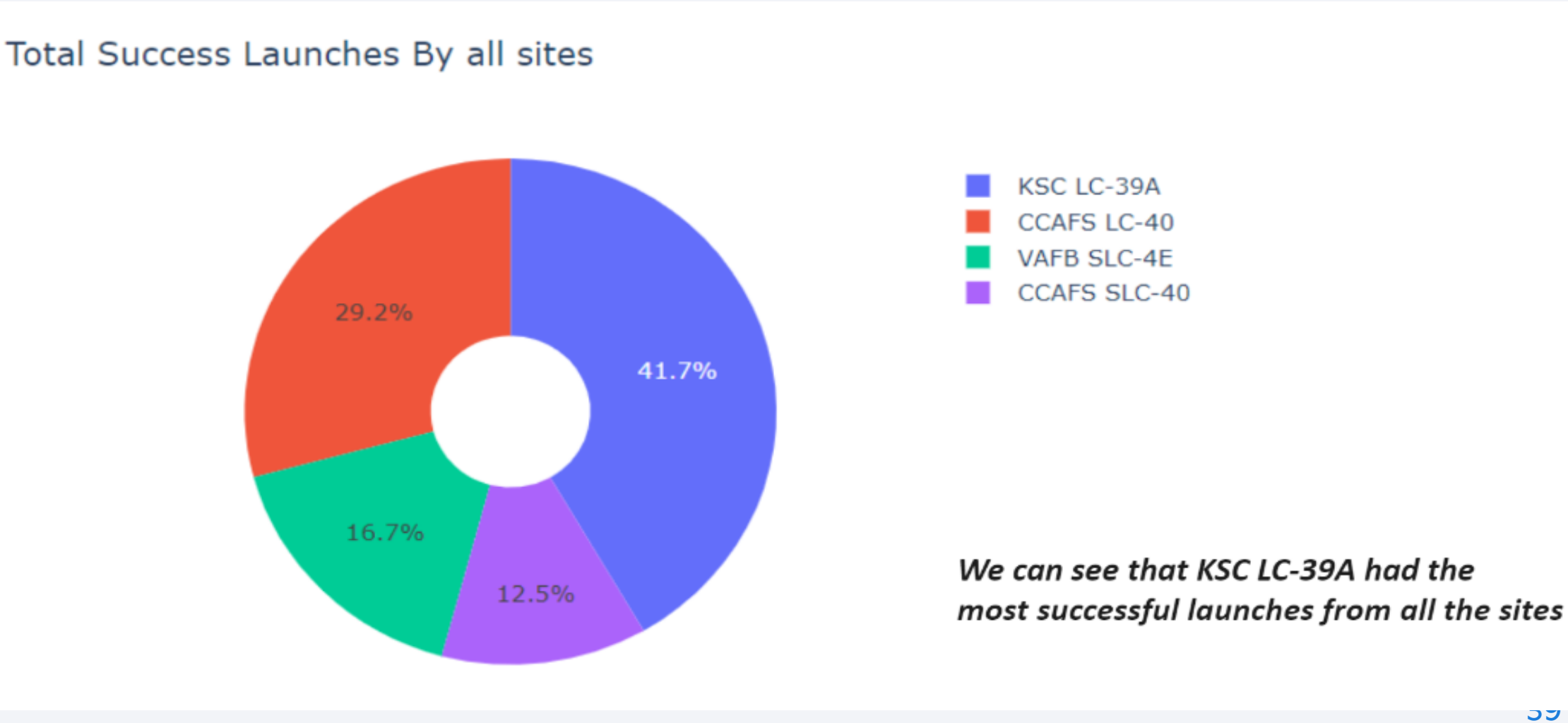
Launch Site distance to landmarks



- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

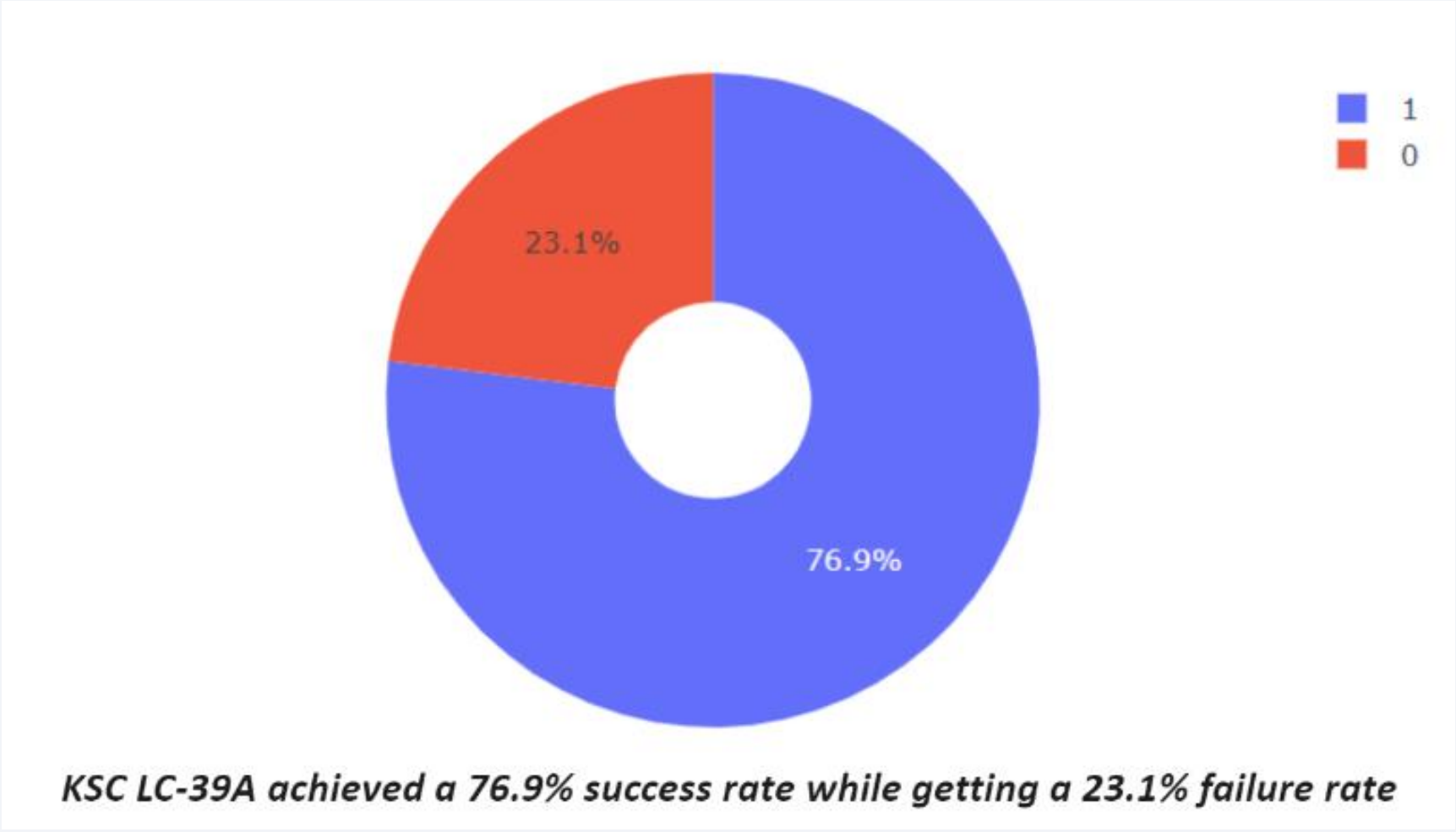

Section 4
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



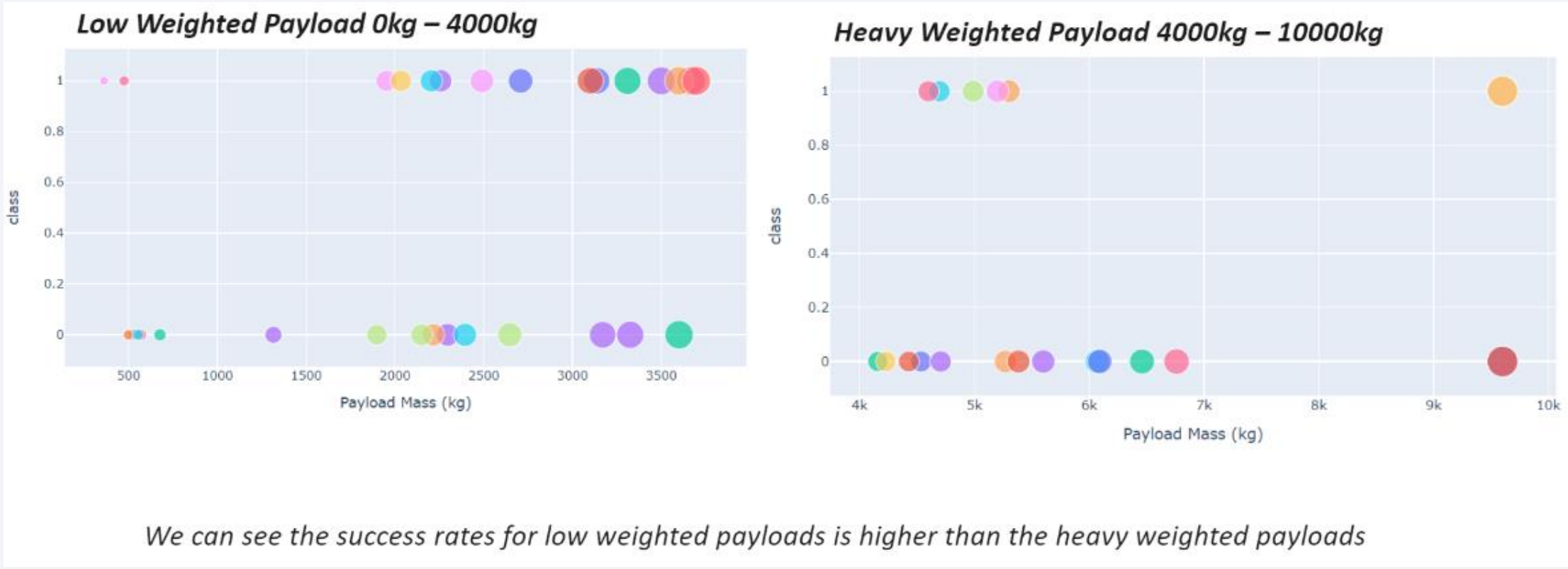
We can see that KSC LC-39A had the most successful launches from all the sites

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



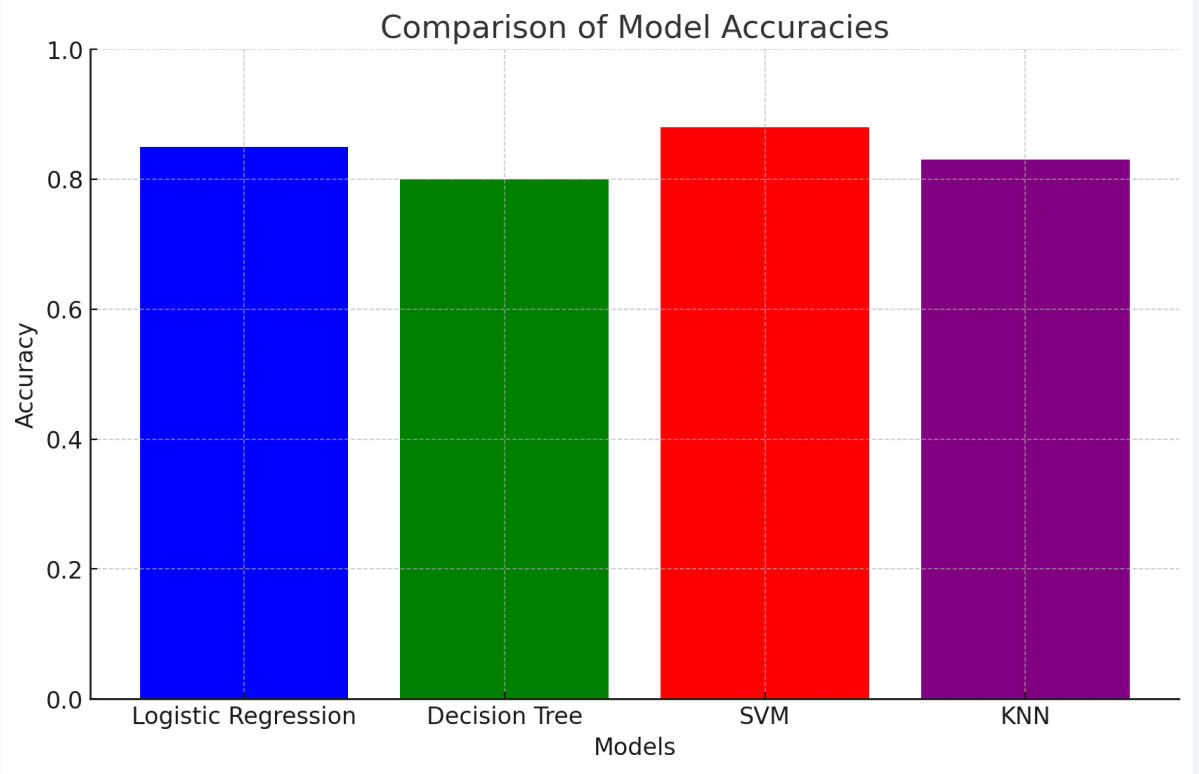
We can see the success rates for low weighted payloads is higher than the heavy weighted payloads

Section 5

Predictive Analysis (Classification)

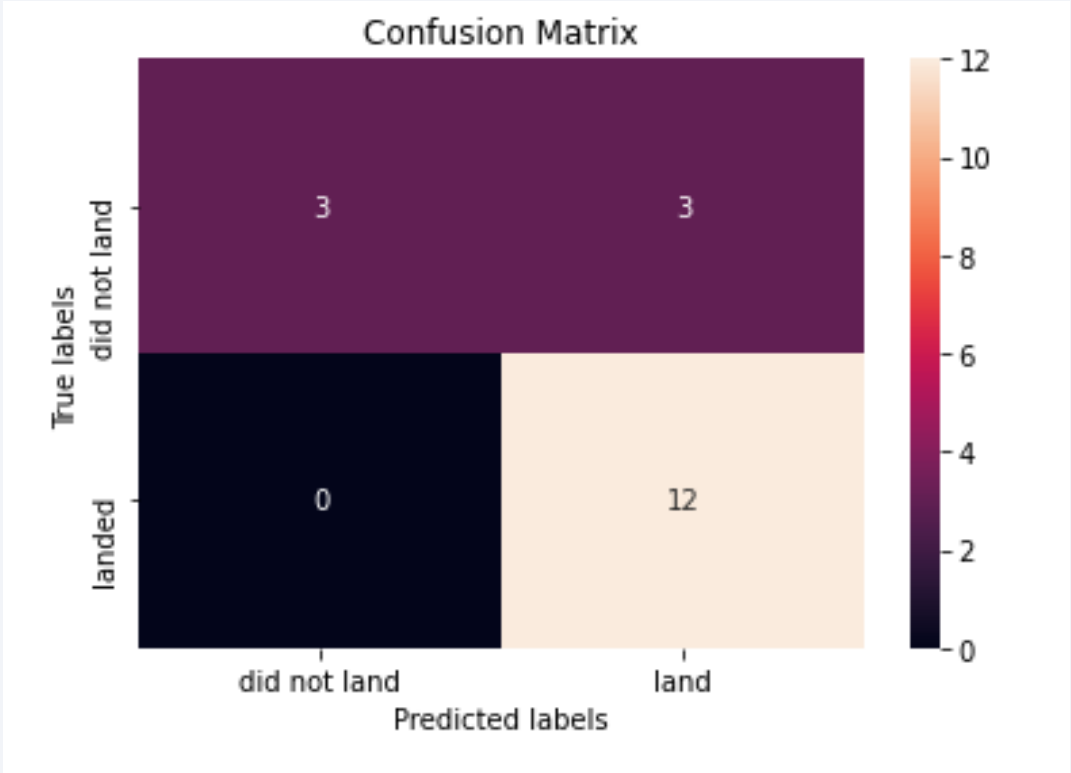
Classification Accuracy

- Here is the bar chart showing the accuracy of each model: Logistic Regression, Decision Tree, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). The chart visualizes their respective accuracies, providing a clear comparison of their performance



Confusion Matrix

- The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes. The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.



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!