IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Thanh Nguyen Van
24.11.2024

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 Data Visualization
  - Exploratory Data Analysis with SQL
  - Interactive Visual Analytics with Foilum
  - Interactive Visual Analytics with Plotly Dash
  - Machine Leaning prediction (classification)
- **Summary of all results**
  - Exploratory Data Analysis results
  - Interactive Visual Analysis results
  - Predictive Analytic (Classification) results

Introduction

- **The context:** The commercial space age is here, companies are making space travel more affordable for everyone. SpaceX is the most successful in this race. One reason for this success is that SpaceX's rocket launches are relatively inexpensive, SpaceX advertise on Falcon 9 rocket launches on its website with a cost of 62 million dollars, others provider costs upwards of 165 million dollars, much of the saving is because SpaceX can reuse the first stage. Therefore, if we can determine whether the first stage will land, we can determine the cost of the launch. This information can be used if an alternative company wants to bids against SpaceX for a rocket launch.
- **Problems to solve:** To determine whether the SpaceX Falcon 9 first stage can land successfully?
- The GitHub URL of the completed project: https://github.com/Thanhlou1368/DSCapstone

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected by using SpaceX API and Python Web Scraping from Wikipedia
- Perform data wrangling
  - Replacing missing values with mean value
  - 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
  - Using GridSearchCV to tune the hyperparameters and determine the best model among Logistic Regression, Support Vector Machine, Decision Tree and K-Nearest Neighbor

Data Collection

- Describe how data sets were collected by using SpaceX API and Web Scraping BS4

SpaceX API

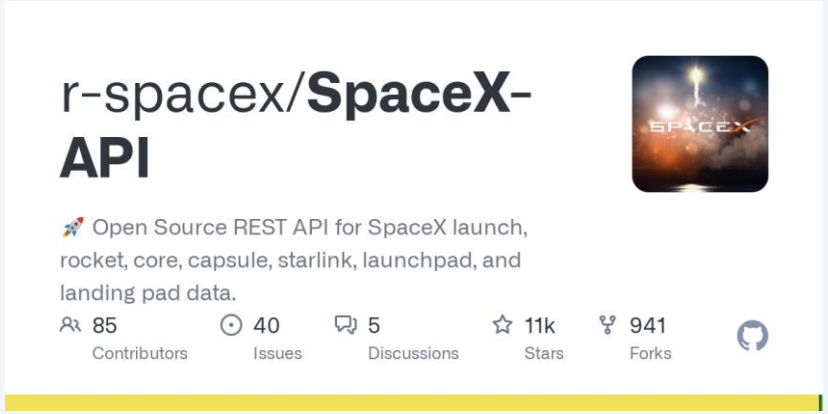
Request to the SpaceX API

Clean the requested data

Web Scraping from Wikipedia

Extract a Falcon 9 launch records HTML table from Wikipedia

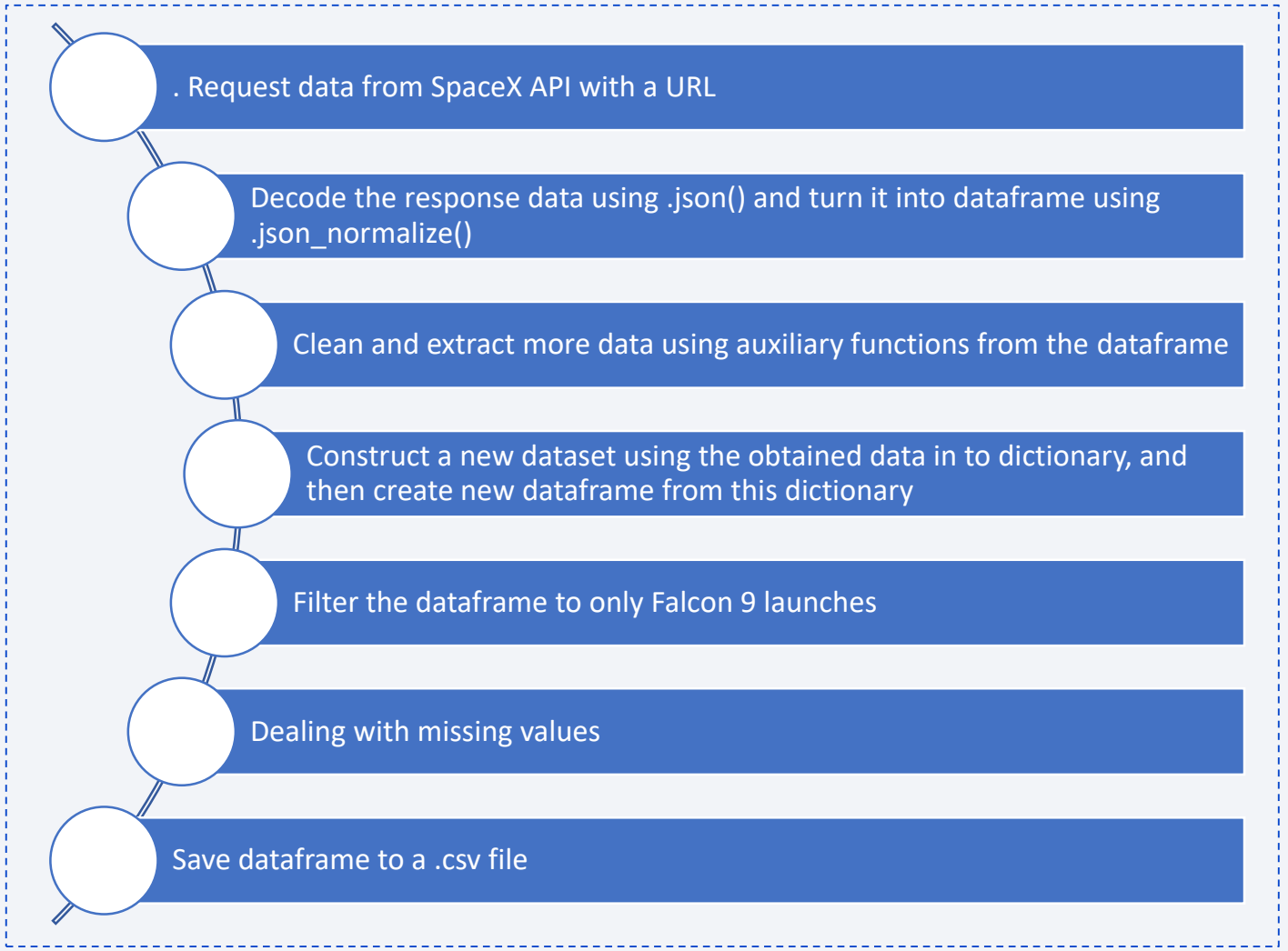
Parse the table and convert it into a pandas dataframe



Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

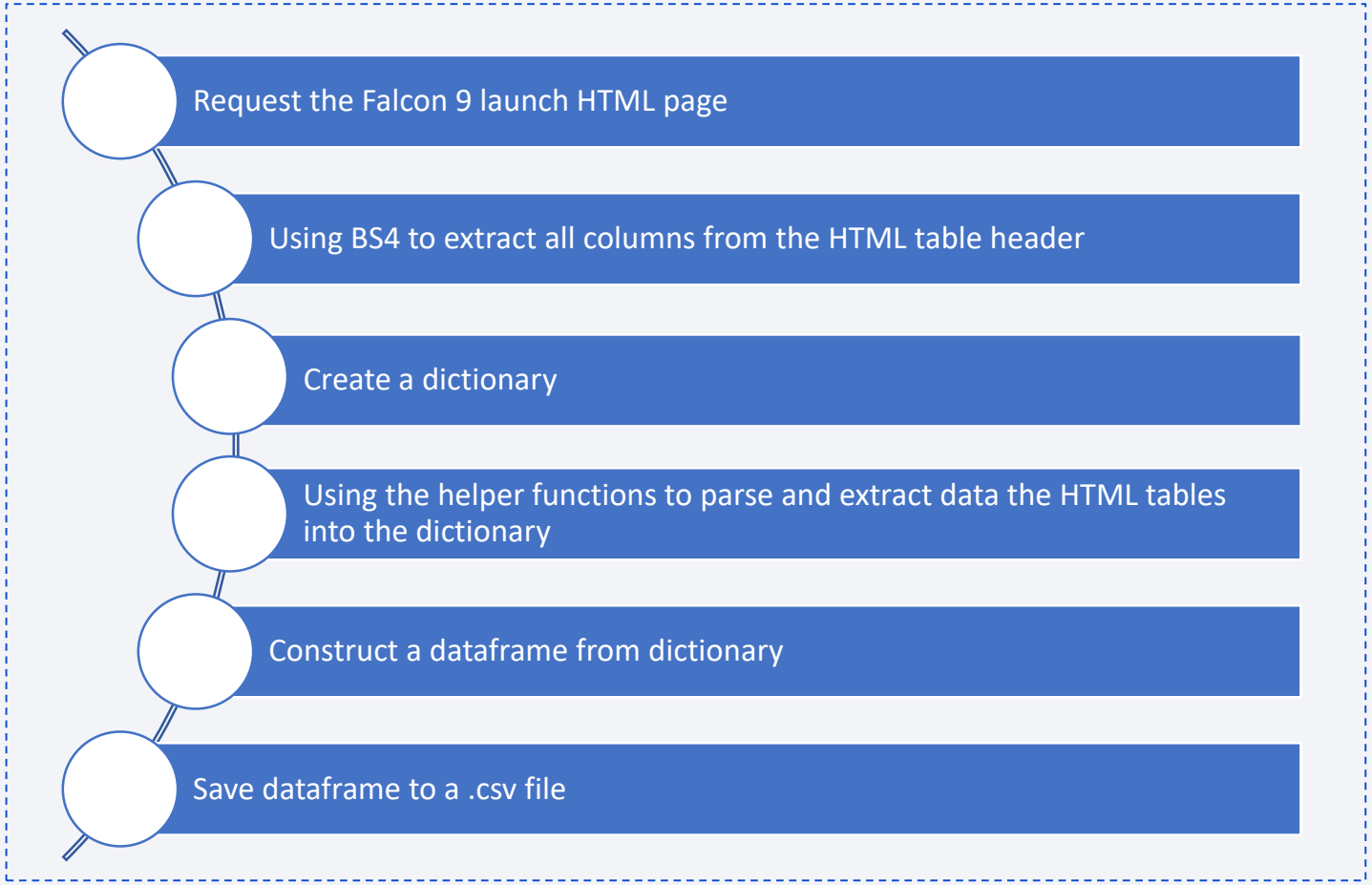
- The GitHub URL of the completed SpaceX API calls notebook: https://github.com/Thanhlou1368/DSCapstone/blob/main/1.%20Data%20Collection%20using%20API%20and%20Wrangling%20Data/jupyter-labs-spacex-data-collection-api.ipynb



Data Collection - Scraping

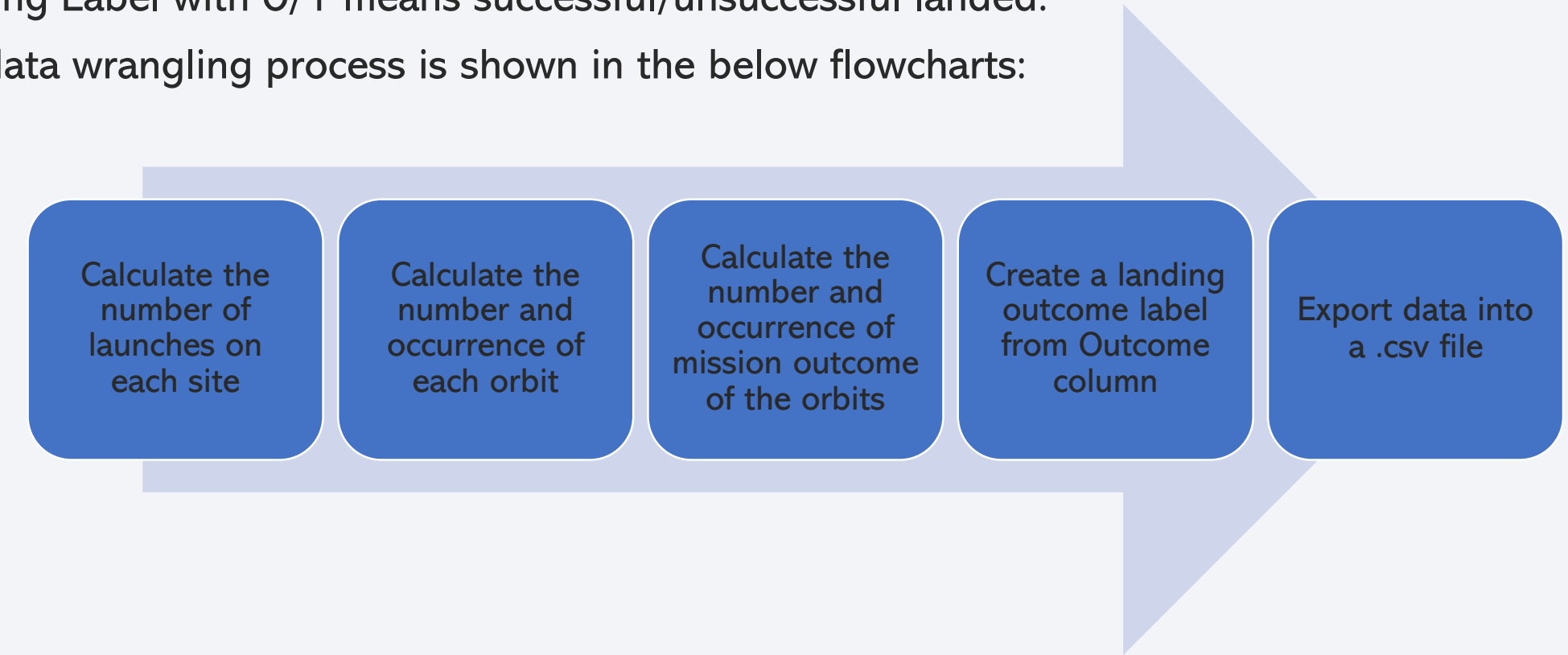
- Present your web scraping process using key phrases and flowcharts

- The GitHub URL of the completed web scraping notebook: https://github.com/Thanhlou1368/DSCapstone/blob/main/2.%20Data%20Collection%20using%20Webscraping%20BS4/jupyter-labs-webscraping%20(3).ipynb



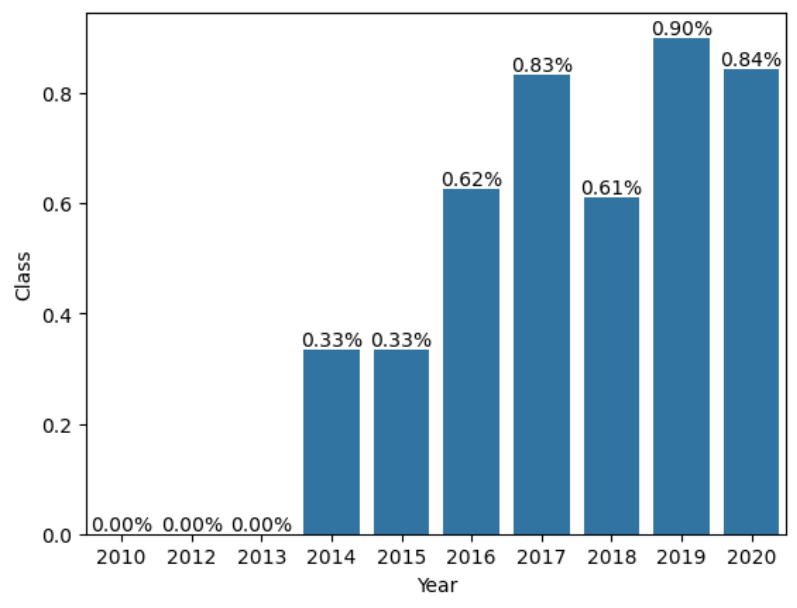
Data Wrangling

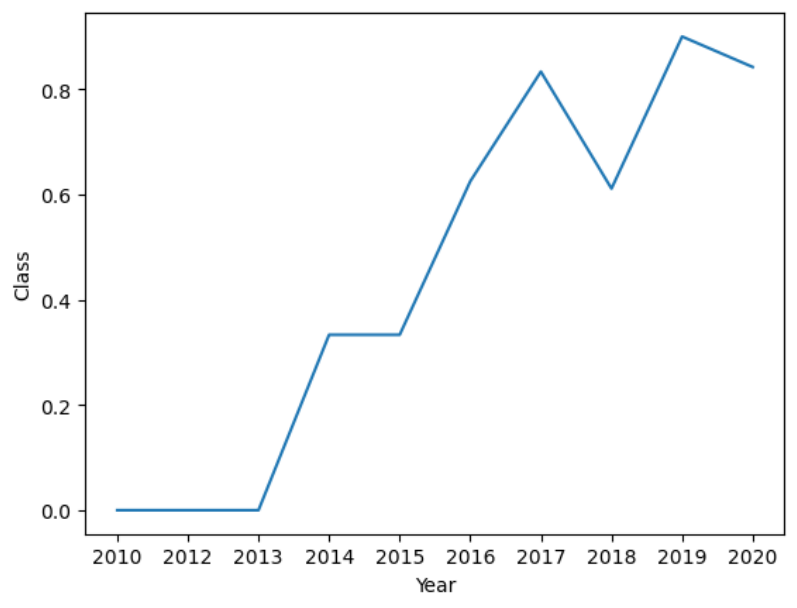
- We used EDA to find some patterns in the data and converted the mission outcome data into Training Label with 0/1 means successful/unsuccessful landed.
- The data wrangling process is shown in the below flowcharts:



- The GitHub URL of the completed data wrangling related notebook: https://github.com/Thanhlou1368/DSCapstone/blob/main/3.%20Data%20Wrangling%20using%20Data%20collected%20from%20API/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb

EDA with Data Visualization





- We explored patterns from the data by visualizing the relationship between the Flight Number and Launch Site (scatter plot), Payload Mass and Launch Site (scatter plot), Success rate and Orbit type (Bar char), Flight Number and Orbit type (scatter plot), Payload Mass and Orbit type (scatter plot), we observed that the success rate since 2013 kept increasing till 2020 by visualizing the launch success yearly trend (bar plot and line plot).
- The GitHub URL of your completed EDA with data visualization notebook: https://github.com/Thanhlou1368/DSCapstone/blob/main/4.%20EDA%20with%20data%20visualization/edadataviz%20(2).ipynb

EDA with SQL

- We loaded the SpaceX dataset into SQLite and applied EDA with SQL to get insight from data.

- Display the names of the unique sites in the space mission
- Display 5 records where launches sites begin with the string 'CCA'
- Display the total Payload Mass carried by booster launched by NASA
- Display average payload mass carried by Booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster versions which have carried the maximum payload mass using subquery
- List the records which display the month names, failure landing_outcomes in drone ship, booster versions, launch_site for the months in year 2015
- Rank the count of the landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order

- The GitHub URL of your completed EDA with SQL notebook: https://github.com/Thanhlou1368/DSCapstone/blob/main/5.%20EDA%20with%20SQL/jupyter-labs-eda-sql-coursera_sqllite%20(1).ipynb

Build an Interactive Map with Folium

- We aim to build a interactive map to find some geographical patterns about the launch sites.
  - Mark all the launch sites on the map, we can find out that all the launch sites are in proximity with the Equator line and the coast
  - Mark the success/failed launches for each site on the map. From the color-labeled markers in marker clusters, we could easily identify which launch sites have relatively high success landing rates.
  - Calculate the distances between a launch site to its proximities. By calculating and displaying the distance line on the map, we can observed that the Launch sites are strategically built in close proximity to railways, highways and to the coast line to facilitate transportation and logistics. However, the Launch sites should keep a certain distance away from cities for people's security
- The GitHub URL of the completed interactive map with Folium map: https://github.com/Thanhlou1368/DSCapstone/blob/main/6.%20Build%20a%20interactive%20Map%20with%20Folium/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- We built a dashboard with Plotly Dash enabling users to perform interactive visual analytics on the SpaceX launch dataset. By incorporating dropdown list, range slider to interact with a pie chart and a scatter plot, users can easily identify insights about which aspects affect to the success launch rate

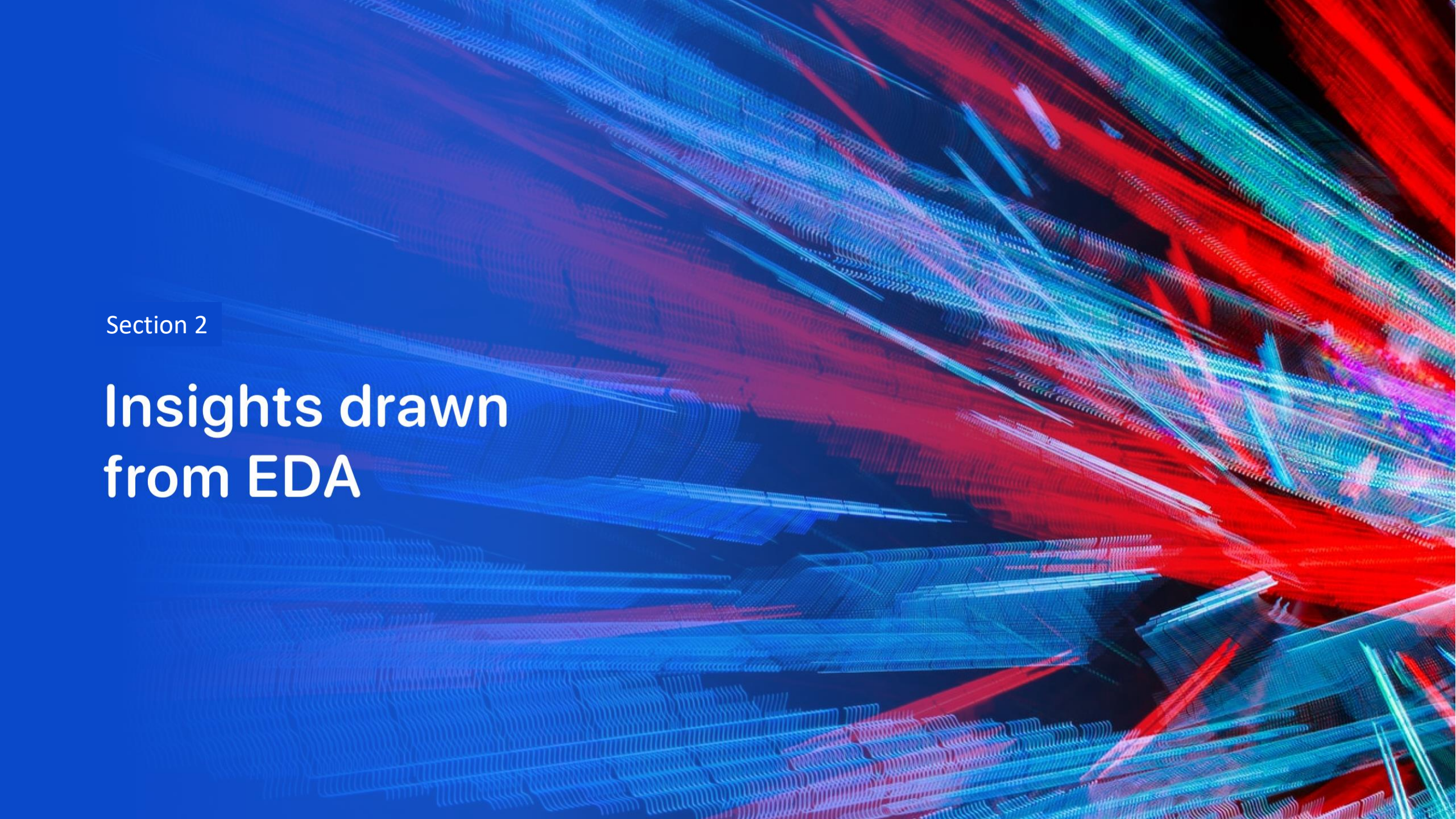
- The GitHub URL of the completed Plotly Dash lab: https://github.com/Thanhlou1368/DSCapstone/blob/main/7.%20Build%20a%20Dashboard%20wth%20Plotly%20Dash/spacex_dash_app_1.py

Predictive Analysis (Classification)

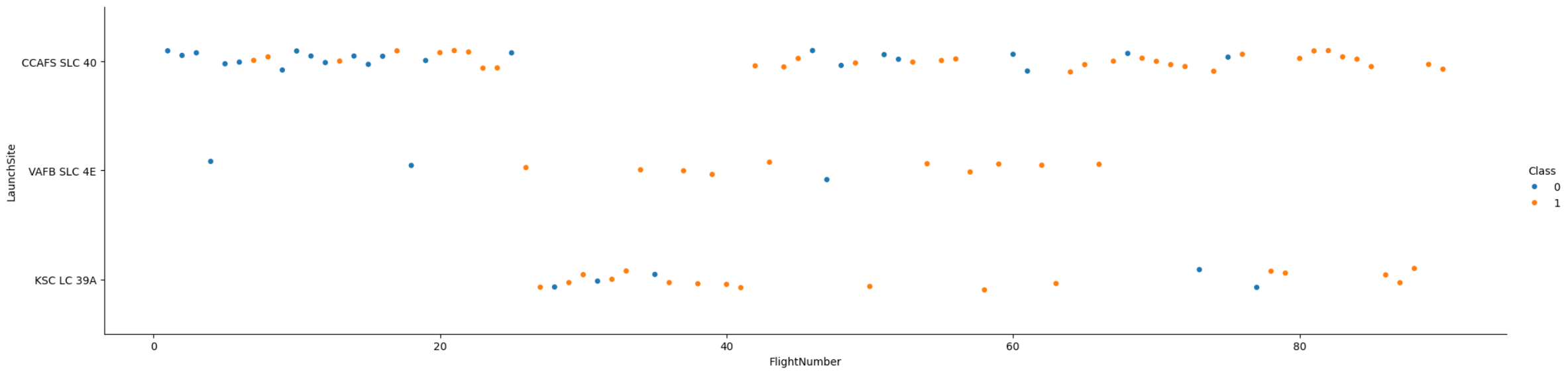
- We imported necessary libraries and downloaded the cleaned dataset.
- After standardizing the dataset, we create X and Y variables for the predictive analysis task
- Next, we used the train_test_split function to split the data into training and test sets.
- We then created a GridSearchCV object using logistic regression, trained it with training data and evaluated the model using the test data.
- The process was repeated with support vector machine, decision tree classifier, K nearest neighbor models
- The GitHub URL of your completed predictive analysis lab: https://github.com/Thanhlou1368/DSCapstone/blob/main/8.%20Predictive%20analysis/SpaceX_Machine%20Learning%20Prediction_Part_5%20(1).ipynb

Results

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

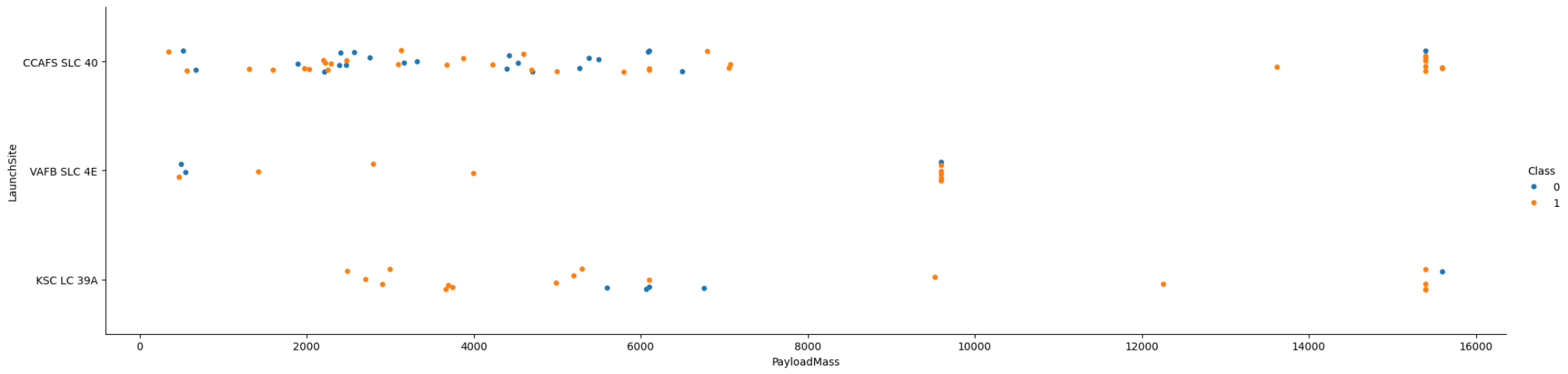

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



- We can observe that the success rate increases over time as the number of flights grows

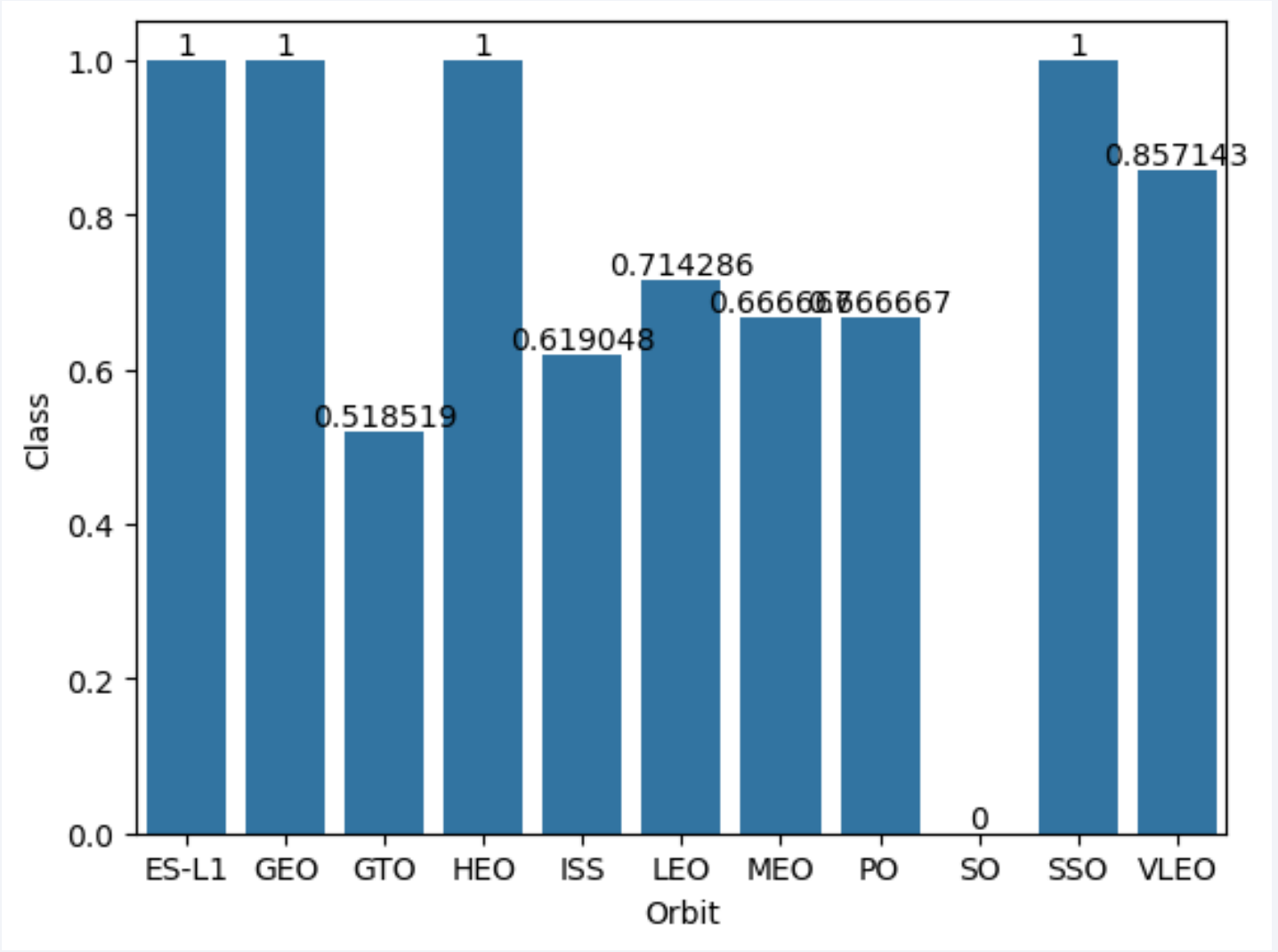
Payload vs. Launch Site



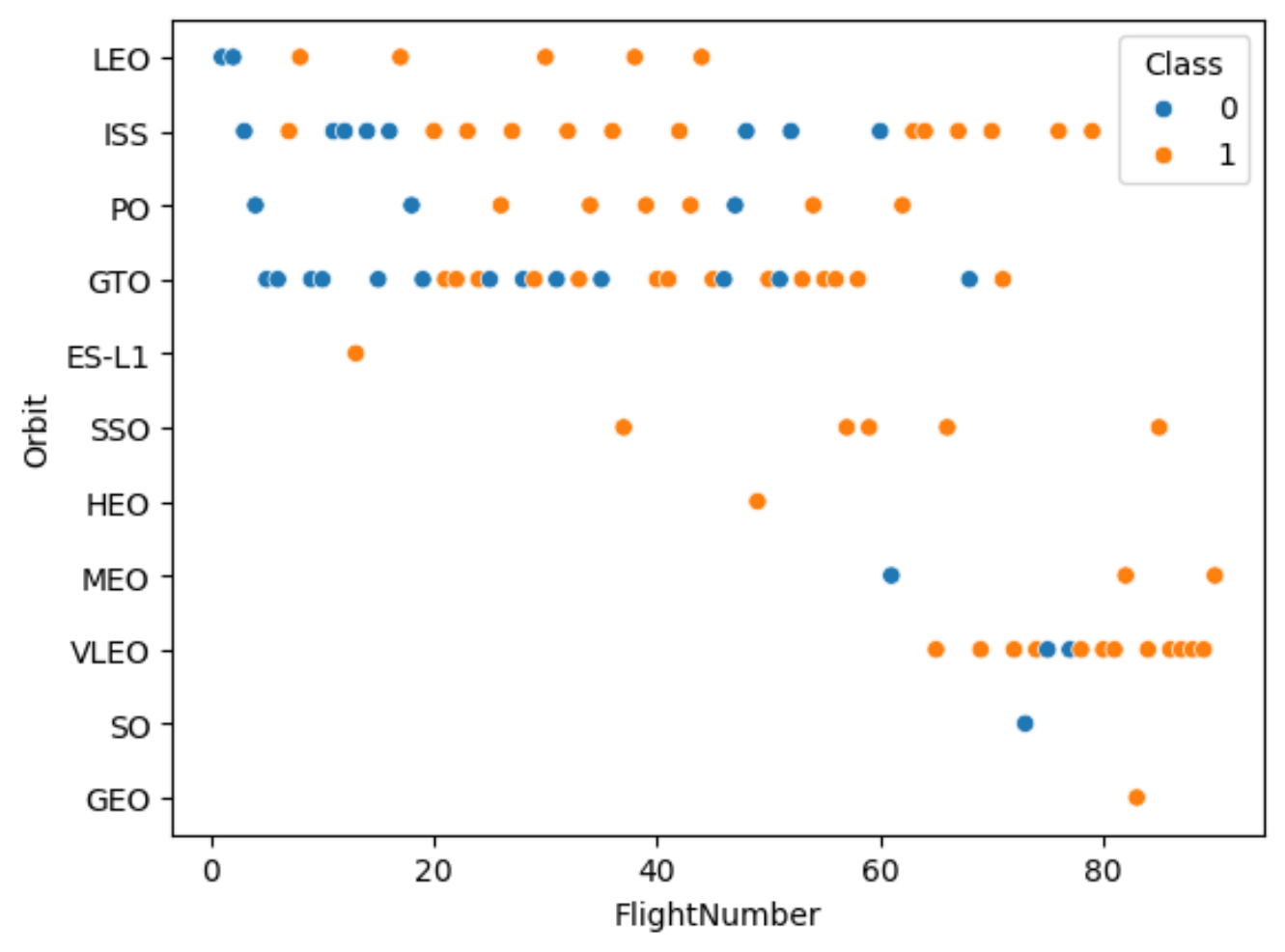
- The optimal Payload Mass for CCAFS SLC40 is between 15.000 and 16.000 Kg, while no rockets have been launched for Payload Mass between 8.000 and 13.000 kg
- At VAFB-SLC there have been no launches for heavy Payload Mass (greater than 10.000 kg) and also no launch with a Payload Mass between 5.000 and 9.000 kg
- The ideal Payload Mass range for KSC LC 39A is between 2.000 and 5.5000 kg

Success Rate vs. Orbit Type

- The Orbit with the highest success rates are: ES-L1, GEO, HEO and SSO

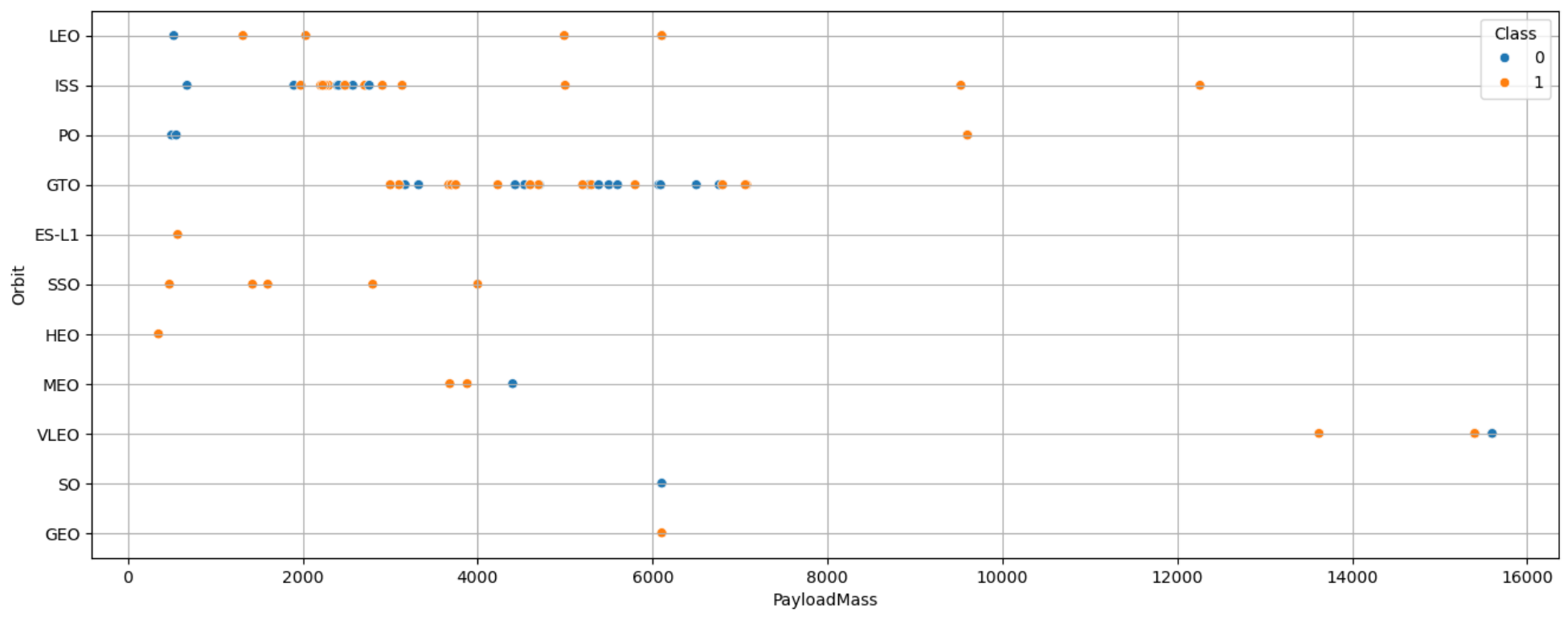


Flight Number vs. Orbit Type



- From the plot we can observe that in the LEO orbit, the success rate appears to be correlated with the number of flights. In contrast, for the GTO orbit, there seems to be no noticeable relationship between the two

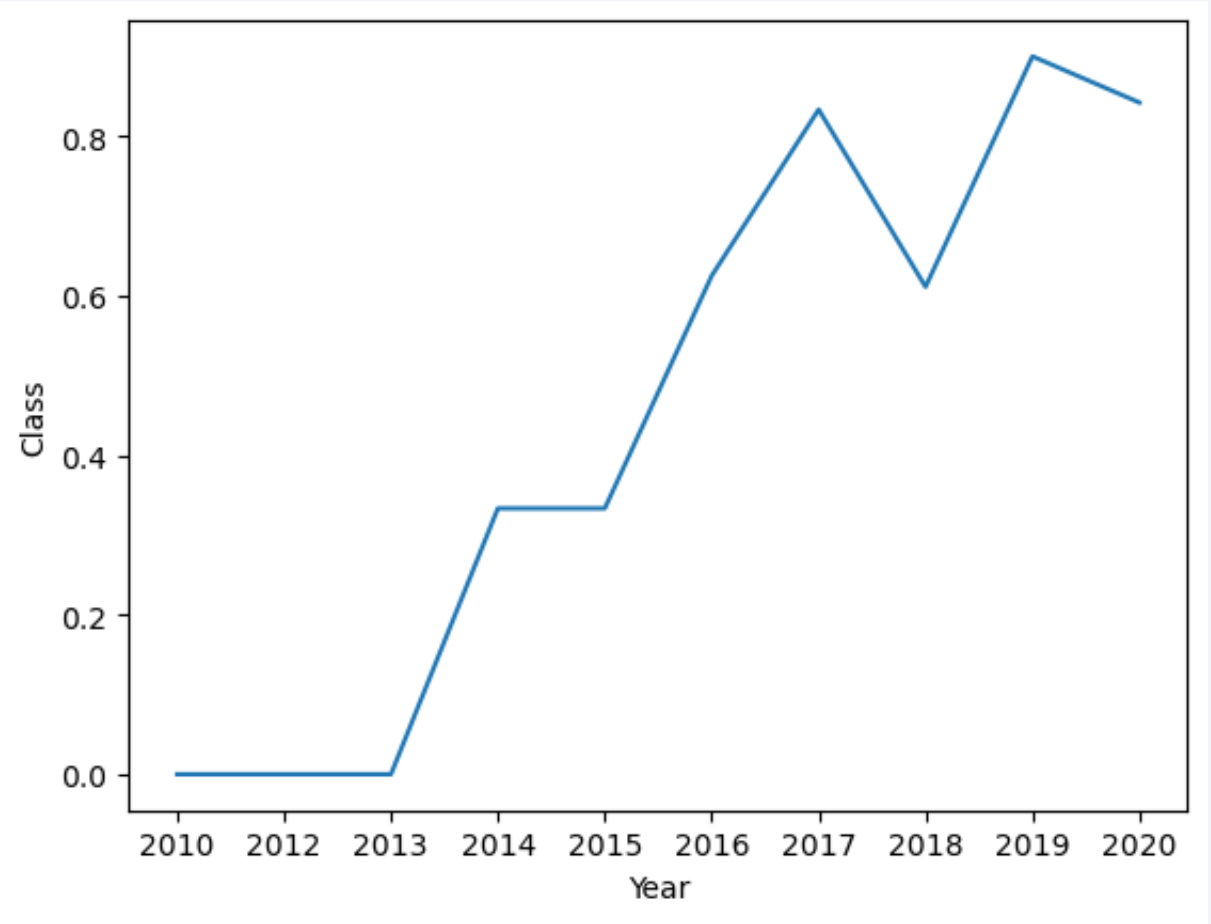
Payload vs. Orbit Type



- With heavy payload mass, the successful landing rates are higher for PO, ISS and LEO orbits.
- In contrast, the ES-L1, SSO, HEO and MEO orbits exhibit higher success rate for low payload masses.

Launch Success Yearly Trend

- We can observer that the success rate since 2013 kept increasing till 2020



All Launch Site Names

Task 1

Display the names of the unique launch sites in the space mission

```
[11]: %%sql
SELECT DISTINCT Launch_Site FROM SPACETABLE
```

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

[11]:

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

- We used DISTINCT keyword to display the names of the unique launch sites in the space mission dataset

Launch Site Names Begin with 'CCA'

Task 2

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

[12]:
```
%%sql
SELECT * FROM SPACETABLE
WHERE Launch_Site LIKE 'CCA%'
LIMIT 5
```

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

[12]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

- We used **LIKE 'CCA'** clause along with **LIMIT 5** to retrieve 5 records where the launch site names begin with `CCA`

Total Payload Mass

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

       * sqlite:///my_data1.db
      Done.

[17]: SUM(PAYLOAD_MASS__KG_)
      ----------------------
                       45596
```

- We filtered the customer names by ‘NASA (CRS)’ and then calculated the total payload mass for this customer

Average Payload Mass by F9 v1.1

Task 4 ¶

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

[19]:
```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACETABLE
WHERE Booster_Version = 'F9 v1.1'
```

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

[19]:

AVG(PAYLOAD_MASS__KG_)
2928.4

- The average payload mass for F9 v1.1 was calculated by filtering the Booster version with ‘F9 v1.1’ keyword and then averaging the value using AVG function

First Successful Ground Landing Date

- To calculate the first successful ground landing date, we find the minimum date value after filtering the Landing_Outcome field with 'Success (ground pad)' keyword

```
[23]: %%sql
      SELECT MIN(Date) FROM SPACETABLE
      WHERE Landing_Outcome = 'Success (ground pad)'

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

[23]:

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

```
[25]: %%sql
      SELECT DISTINCT(Booster_Version) FROM SPACETABLE
      WHERE Landing_Outcome = 'Success (drone ship)'
          AND PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000;
```

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

[25]:

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

- We used WHERE and AND clauses to filter the dataset on multiple conditions.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

[15]:
```
%%sql
SELECT COUNT(Mission_Outcome) AS successful FROM SPACETABLE
WHERE Mission_Outcome LIKE '%Success%';
```

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

[15]:

successful
100

[17]:
```
%%sql
SELECT COUNT(Mission_Outcome) AS failed FROM SPACETABLE
WHERE Mission_Outcome LIKE '%Failure%'
```

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

[17]:

failed
1

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here

Boosters Carried Maximum Payload

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

```
[14]: %%sql SELECT DISTINCT(Booster_Version) FROM SPACETABLE
      WHERE PAYLOAD_MASS__KG_ =  (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACETABLE)
```

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

[14]:

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 list the names of the Booster versions that have carried the maximum payload mass, we used a Subquery to find the maximum payload mass from the dataset then filtered the Booster version by their payload mass

2015 Launch Records

```
[16]: %%sql
SELECT SUBSTR(Date, 6, 2) AS Month, Landing_Outcome, Booster_Version, Launch_Site
FROM SPACETABLE
WHERE (SUBSTR(Date, 0, 5) = '2015') AND (Landing_Outcome LIKE '%Failure (drone ship)')
```

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

[16]:

Month	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

- We used the **SUBSTR** function to extract the Month and Year values from Date column. Then we filtered the dataset by **Year = 2015** and **Landing outcome = Failure (drone ship)** to list the records. These records display the month names, failure landing_outcomes in drone ship, booster version, Launch site for the months in year 2015

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

```
[44]: %%sql
SELECT Landing_Outcome, count(*) AS COUNTS FROM SPACETABLE
WHERE Date BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY Landing_Outcome
ORDER BY COUNTS DESC
```

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

[44]:

Landing_Outcome	COUNTS
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

- To rank the count of landing outcomes within a given period of time and list them in a descending order, we first need to filter the dataset using the WHERE clause with the Date range between 2010-06-04 and 2017-30-20. Filtered data then will be grouped by Landing_outcomes and the results will be ordered in a descending order base in the count.

Section 3

Launch Sites Proximities Analysis

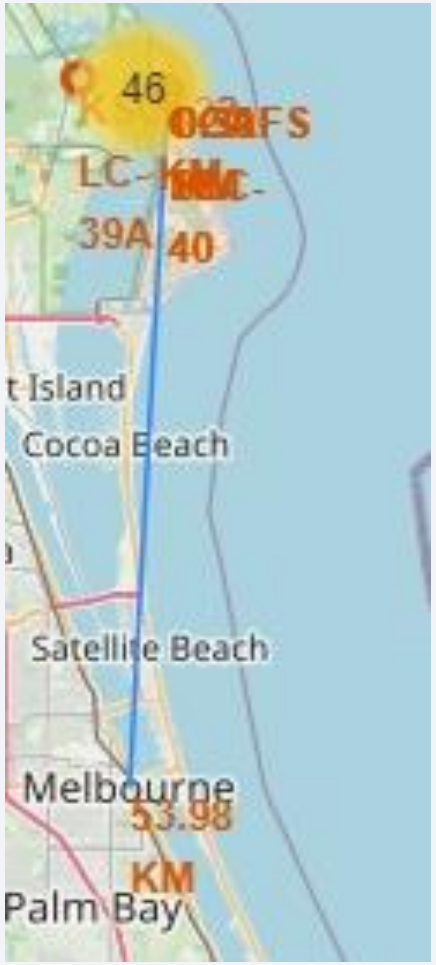
Generated Map with marked launch sites

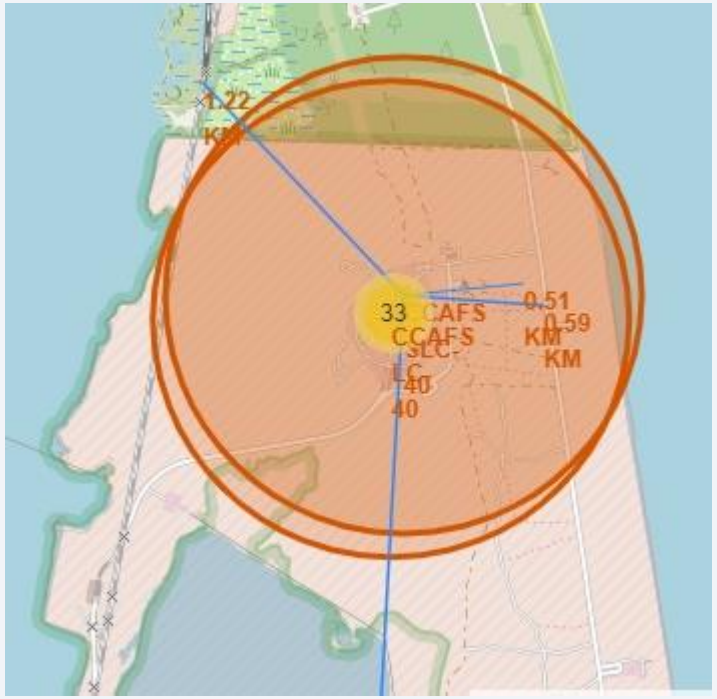
- All the Launch sites are located above the Equator line
- All the Launch sites are in very close proximity to the coast

Success/Failed launches for each site on the map with color labeled

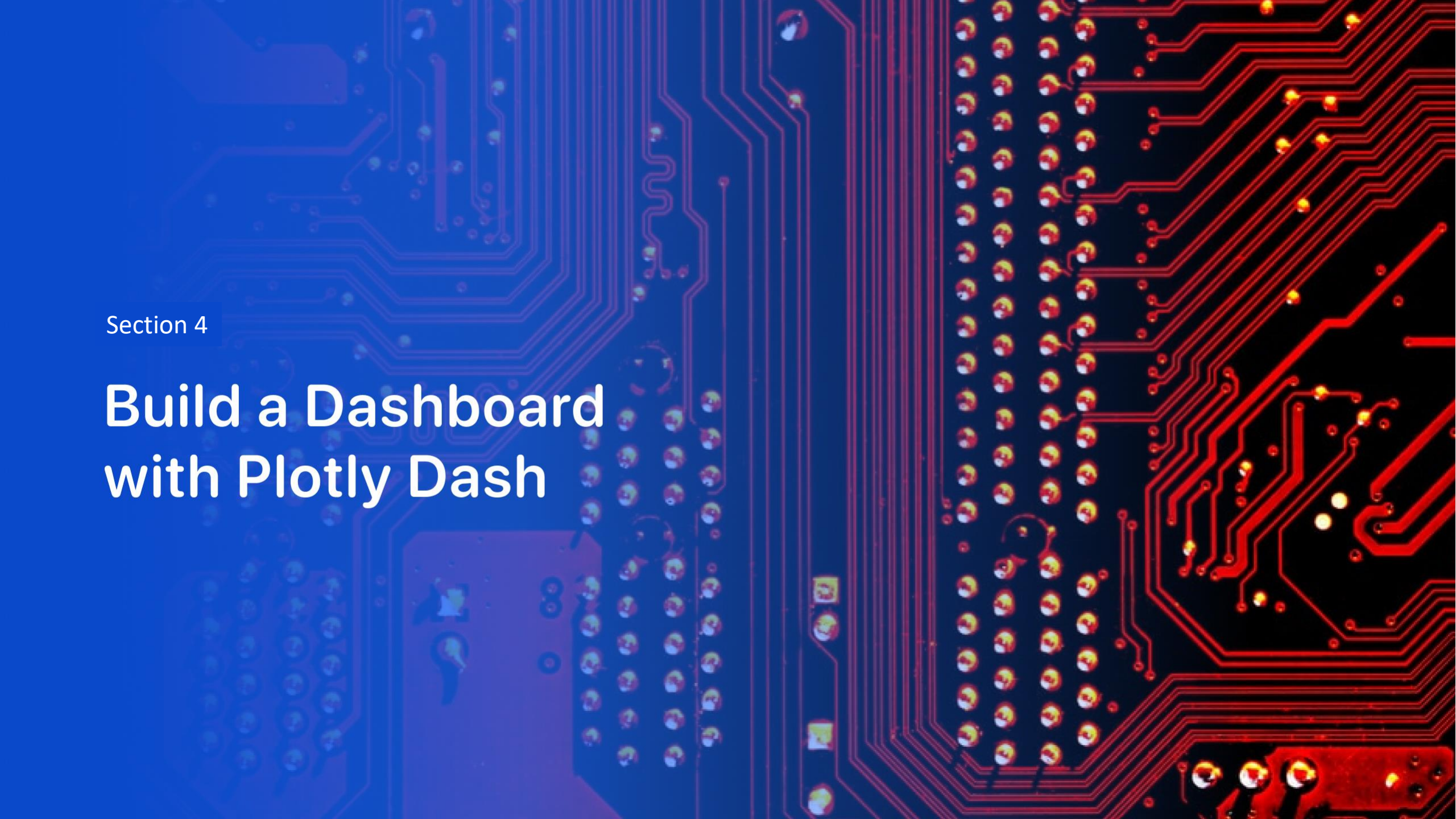
- Green show successful launches and Red show Failed launches
- From the color-labeled markers we can easily identify which sites have relatively high success rates.

Distance between a launch site to its proximities

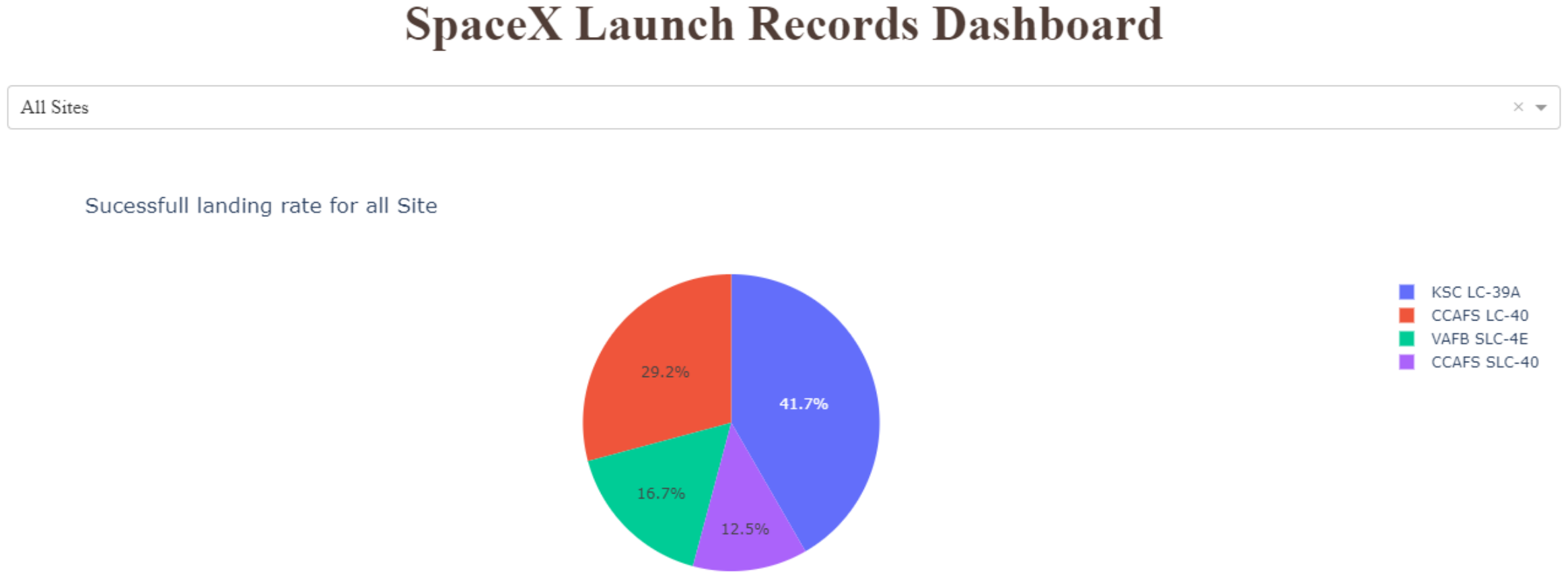




- We choose the SLC40 site to calculate the distances
- From the resulted line distances on the map, we could observe that:
  - Launch sites are in close proximity to railways, highways and coastline to facilitate in transportation and logistics
  - Launch sites keep a certain distance away from cities to ensure public safety.

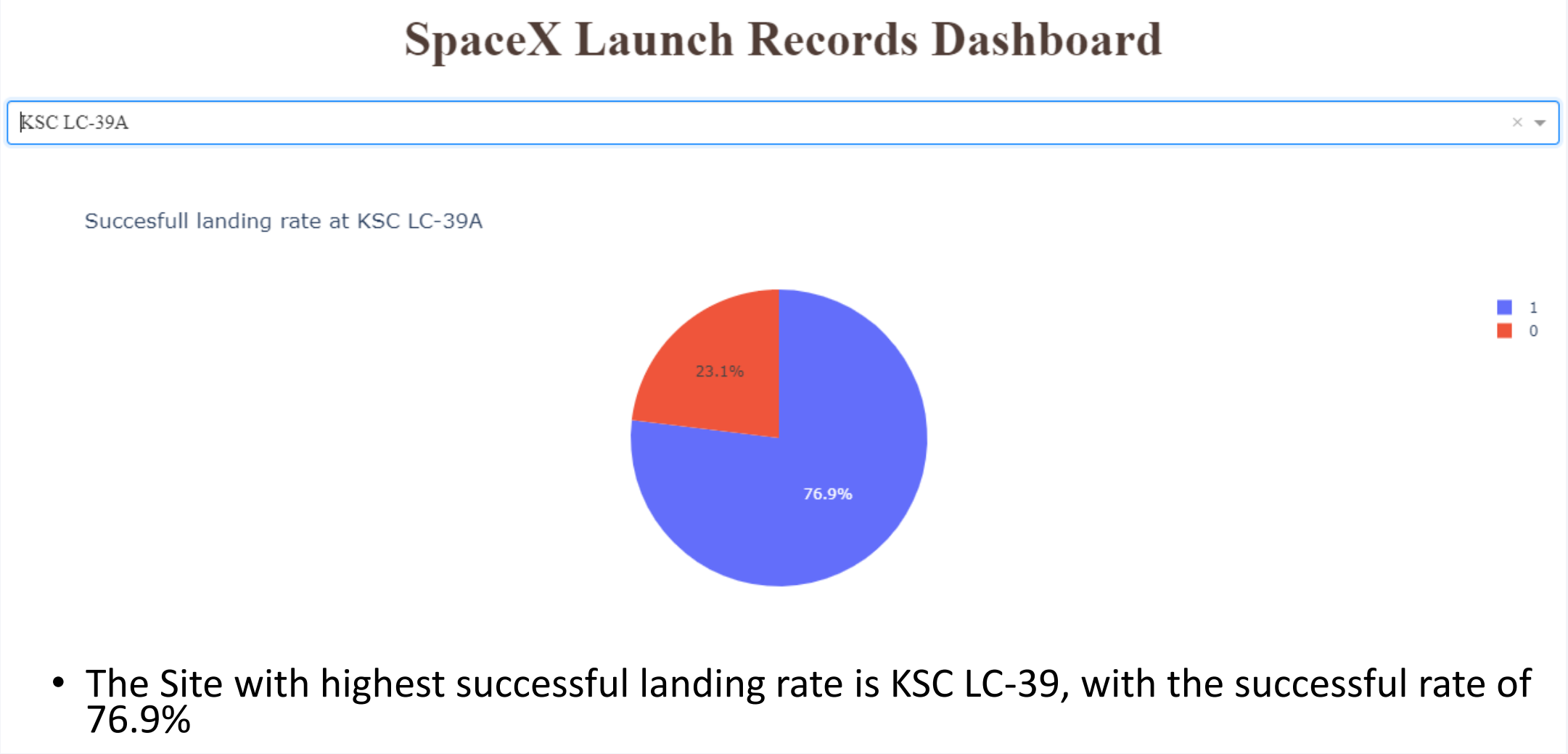

Section 4
Build a Dashboard with Plotly Dash

Success launches for all site



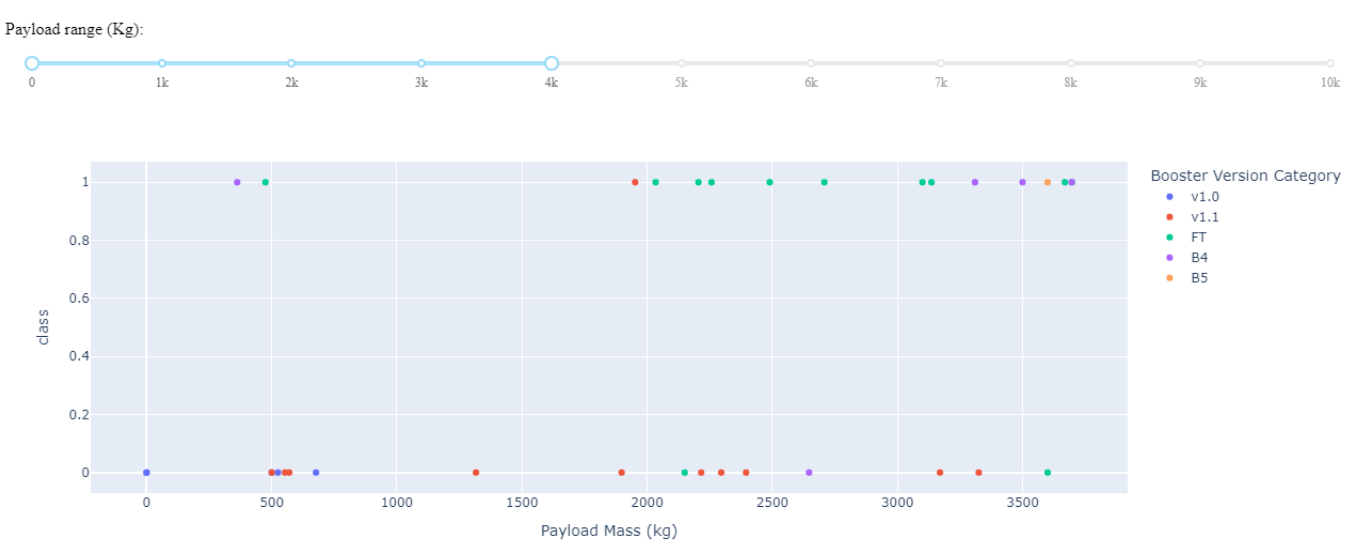
- From the Pie chart, we observed that KSC LC-39 has the highest number of successful launches

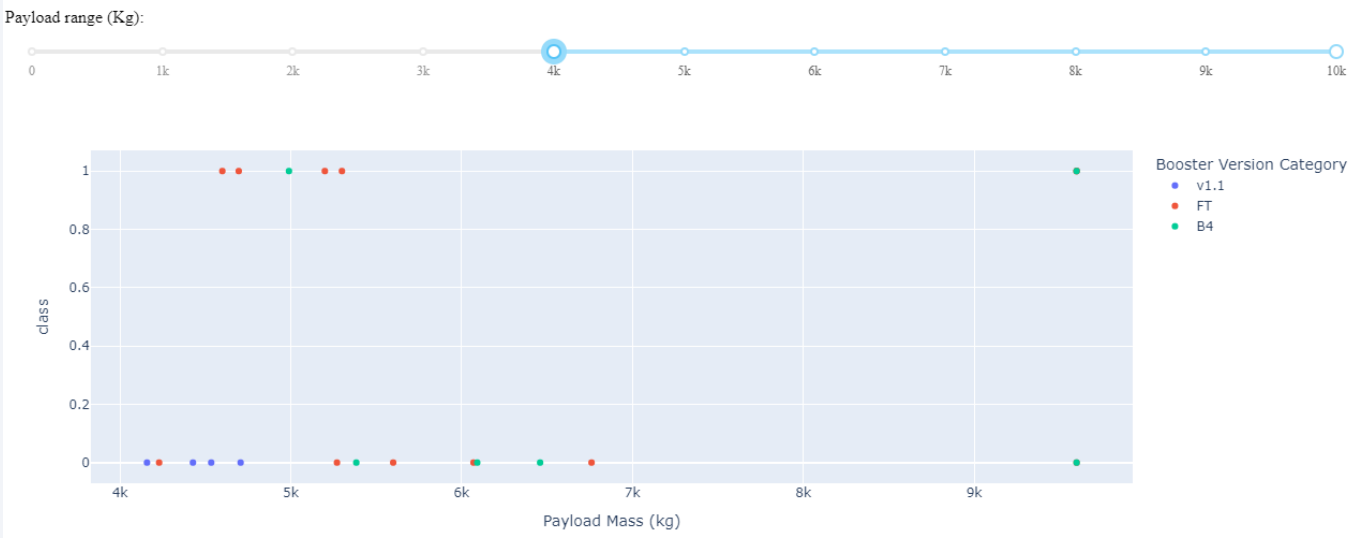
Site with the highest success rate



- The Site with highest successful landing rate is KSC LC-39, with the successful rate of 76.9%

Payload vs Launch Outcome



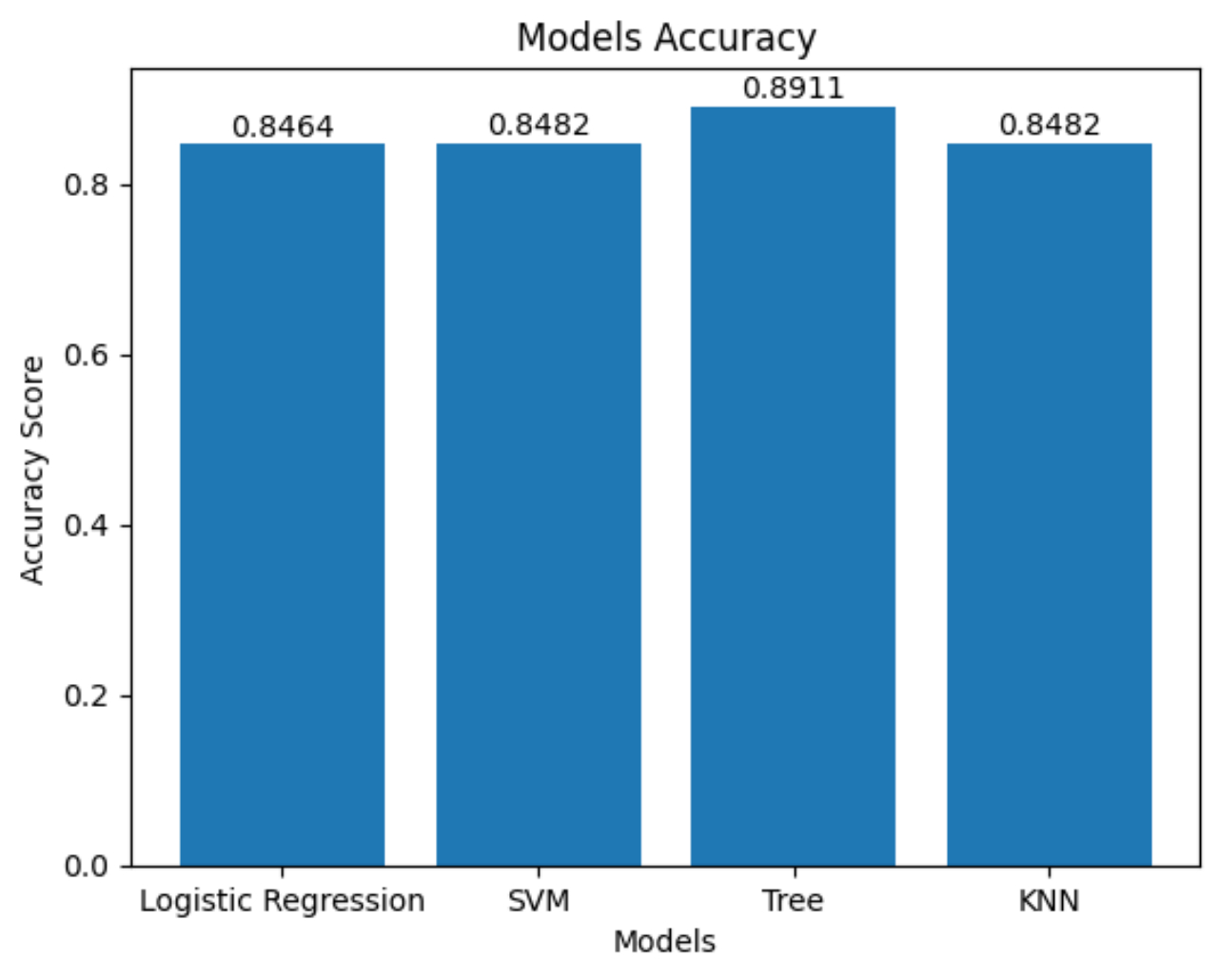


- Low payload mass (between 0 and 4,000 kg) has a higher success rate compared to high payload mass (between 4,000 and 10,000 kg)

Section 5

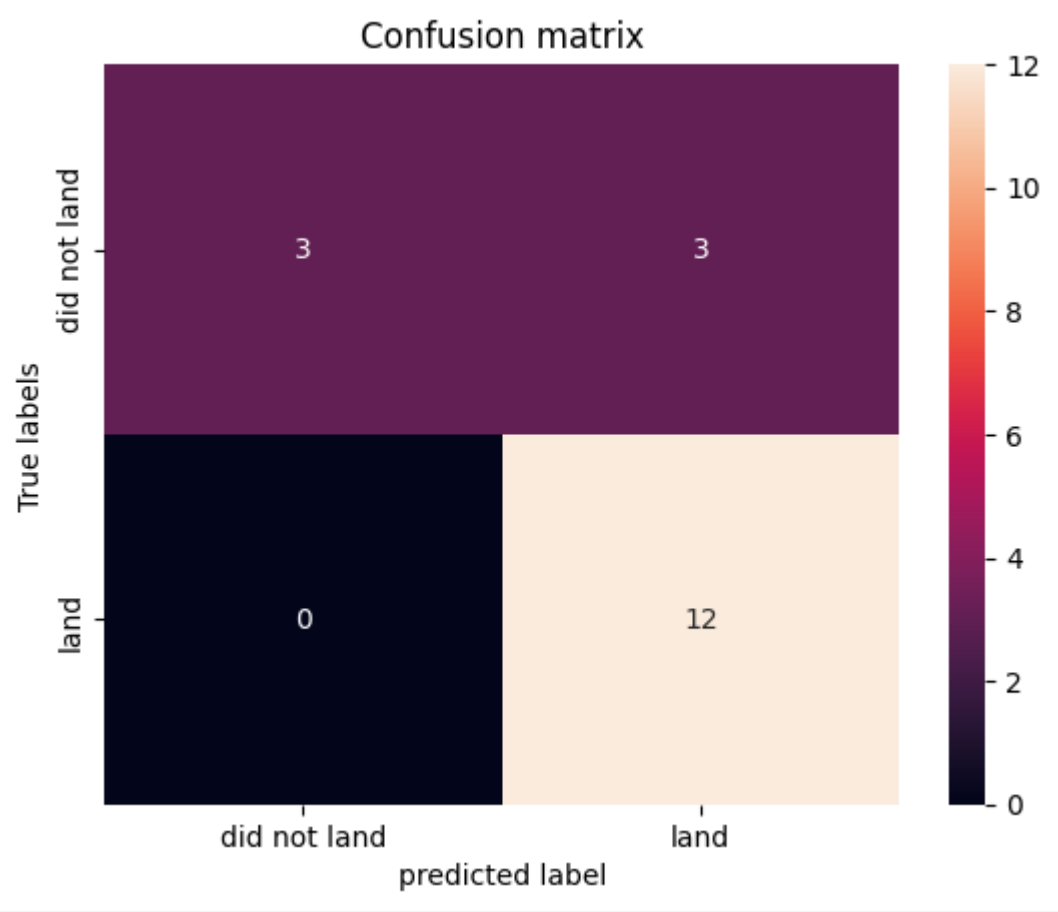
Predictive Analysis (Classification)

Classification Accuracy



- As observed from the bar chart, the highest accuracy score is belongs to the Tree classifier model.

Confusion Matrix



- By examining the confusion matrix, we observed that Tree classifier model can effectively distinguish between the different classes. However, we can notice an issue with false positive, where the True label indicates 'not landing', but Predicted label indicates 'landed')

Conclusions

From the above analyses, we can conclude that:

- The success rate of launches has shown a steady increase over time as the number of flight grows. Notably, it has consistently risen from 2013 to 2020.
- The orbit that exhibit the highest success rates are ES-L1, GWO, HEO and SSO
- For heavy payload masses, the success rates are higher in PO, ISS and LEO orbits. In contrast, ES-L1, SSO, HEO and MEO orbits demonstrate higher success rates for low payload masses.
- Launches with low payload masses (between 0 and 4,000 kg) have a significantly higher success rate compared to those with high payload masses (between 4,000 and 10,000 kg)
- All launch sites are located above the equator line and in close proximity to coastlines. This position facilitates transportation and logistics through near by railways, highways, and coastal access. Additionally, these sites are strategically distanced from urban areas to ensure public safety.
- KSC LC-39 stands out as the site with the highest number of successful launches.
- The Tree classifier model achieves the highest accuracy score among the model evaluated.

Appendix

No.	Task	Jupyter Notebook	Dataset
1	Data Collection through API	jupyter-labs-spacex-data-collection-api	dataset_part1
2	Data Collection with Web Scraping	jupyter-labs-webscraping	spacex_web_scaped
3	Data Wrangling	labs-jupyter-spacex-Data wrangling	dataset_part2
4	Exploratory Data Analysis with Data Visualization	edadataviz	dataset_part3
5	Exploratory Data Analysis with SQL	jupyter-labs-eda-sql-coursera_sqllite	Spacex
6	Interactive Visual Analytics with Foilum	lab_jupyter_launch_site_location	
7	nteractive Visual Analytics with Plotly Dash	spacex_dash_app_1	spacex_launch_dash
8	Machine Leaning prediction (classification)	SpaceX_Machine Learning Prediction_Part_5 (1)	

Thank you!