Winning Space Race with Data Science

Asli Karamanlargil

24th April 2023

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
  - Interactive Analytics
  - Predictive Analytics

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.
- Problems you want to find answers
  - What are the successful landing indicators?
  - Are there any associations between multiple features that determine the success rate of a landing?
  - What are the feasibility requirements for successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX API and web scraping from Wikipedia.
- Perform data wrangling
  - Data was converted to pandas dataframe and then EDA was performed to find some patterns
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Finding the best hyperparameter for SVM, Classification Trees and Logistic Regression

Data Collection

- The data was collected from the SpaceX REST API and web scraping from Wikipedia

SpaceX REST API

1. SpaceX REST API endpoint
2. Get request using the requests library
3. Get past launch data as a JSON objects
4. Convert to JSON to a dataframe

Web scraping from Wikipedia

1. Web scraping Falcon 9 launches
2. Use BeautifulSoup to web scrape HTML tables
3. Parse data from tables
4. Convert tables into a dataframe

Data Collection – SpaceX API

- Imported libraries and defined auxiliary functions to extract information
- Requested and parsed the SpaceX launch data using the GET request data from SpaceX API
- Decoded the response content as a Json using .json() and turned it into a Pandas data frame using .json_normalize()
- Filtered the dataframe to only include Falcon 9 launches
- Dealt with the missing values

[Data Collection API - GitHub Link]

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

In [7]: response = requests.get(spacex_url)

In [11]: # Use json_normalize meethod to convert the json result into a dataframe

         data = pd.json_normalize(response.json())

In [24]: # Hint data['BoosterVersion']!='Falcon 1'
         data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

```
In [25]: data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
         data_falcon9

In [31]: # Calculate the mean value of PayloadMass column
         payloadmassavg = data_falcon9['PayloadMass'].mean(axis=0)


         # Replace the np.nan values with its mean value
         data_falcon9['PayloadMass'].replace(np.nan, payloadmassavg, inplace=True)

         data_falcon9.isnull().sum()
```

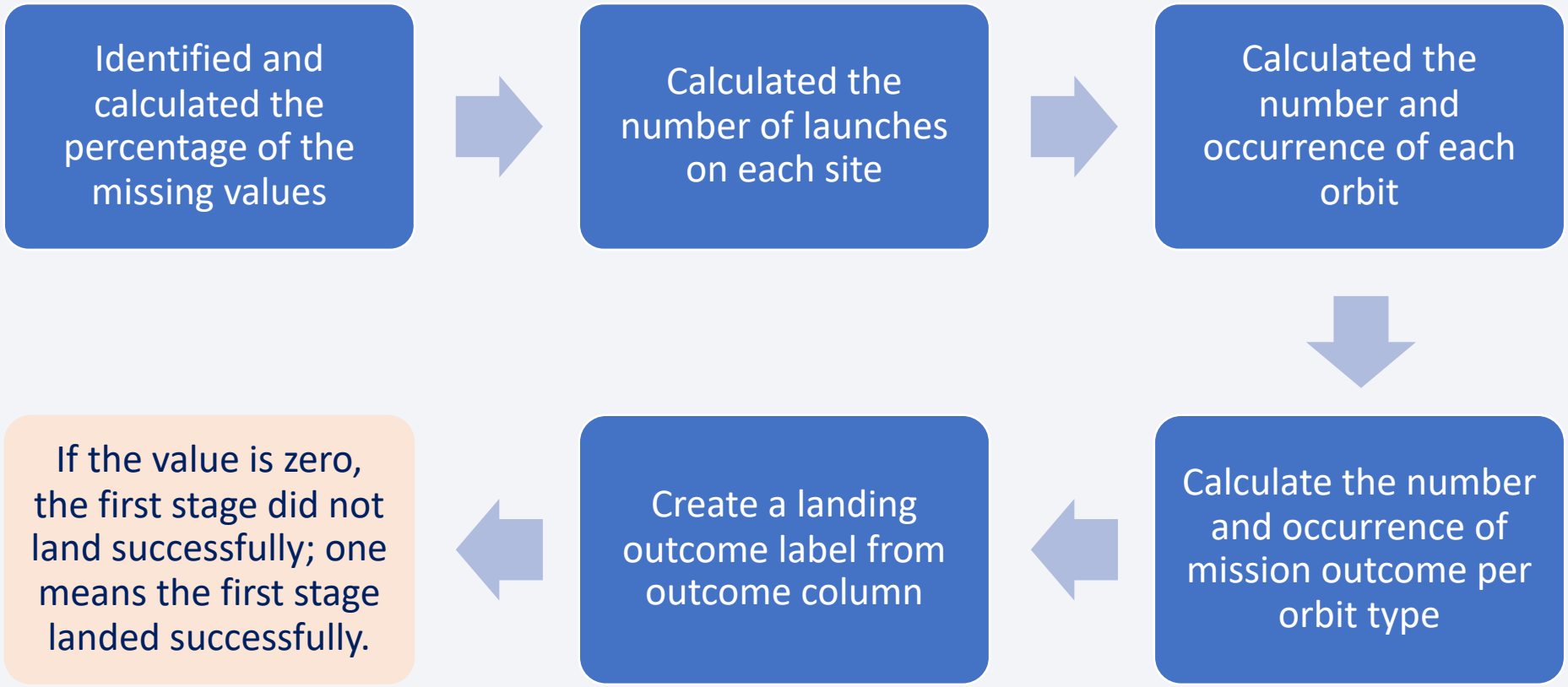
Data Collection - Scraping

1. Performed an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.
2. Created a BeautifulSoup object from the HTML response
3. Extracted all column/variable names from the HTML table header
4. Created an empty dictionary with keys from the extracted column names
5. Filled up the dictionary with launch records extracted from table rows
6. After filling the dictionary, convert this dictionary into a Pandas dataframe

Data Collection Scraping - GitHub Link

Data Wrangling

- Performed some Exploratory Data Analysis to find some patterns in the data and determine what would be the label for training supervised models.



Data Wrangling - GitHub Link

EDA with Data Visualization

- Summary of the charts were plotted:

1. Plotted the FlightNumber vs. PayloadMass by overlaying the outcome of the launch using catplot
2. Plotted the FlightNumber vs. LaunchSite by overlaying the outcome of the launch using catplot
3. Plotted the PayloadMass vs. LaunchSite by overlaying the outcome of the launch using catplot
4. Plotted the success rate of each orbit type using barchart
5. Plotted the FlightNumber vs. Orbit by overlaying the outcome of the launch using catplot
6. Plotted the PayloadMass vs. Orbit by overlaying the outcome of the launch using catplot
7. Plotted the average launch success trend in years using lineplot

EDA with Data Visualization – GitHub Link

EDA with SQL

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

  %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;

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

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

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

  %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer LIKE 'NASA (CRS)%';

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

  %sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass" FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%';

EDA with SQL – GitHub Link

EDA with SQL

- List the date when the first successful landing outcome in ground pad was achieved.

  %sql SELECT Date FROM SPACEXTBL WHERE "LANDING _OUTCOME" like 'Success (ground pad)%' limit 1;

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

  %sql SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ AS "Payload_Mass" FROM SPACEXTBL WHERE "Landing _Outcome" LIKE "Success (drone ship)%" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;

- List the total number of successful and failure mission outcomes

  %sql SELECT Mission_Outcome, COUNT(Mission_Outcome) as Total FROM SPACEXTBL GROUP BY Mission_Outcome;

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

  %sql SELECT Booster_Version, "PAYLOAD_MASS__KG_" AS "Payload_Mass" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);

EDA with SQL – GitHub Link

EDA with SQL

- 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.

  %sql SELECT substr(Date,7,4) as "Year", substr(Date, 4, 2) as "Month", "Booster_Version", "Launch_Site", "Landing _Outcome" FROM SPACEXTBL WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';

- Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

  %sql SELECT "Landing _Outcome" as "Landing Outcomes", COUNT("Landing _Outcome") as "Count" FROM SPACEXTBL WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' group by "Landing _Outcome" order by count("Landing _Outcome") desc;

EDA with SQL – GitHub Link

Build an Interactive Map with Folium

- Summary of the map objects (such as markers, circles, lines) that were created and added to the folium map

1. Marked all launch sites on a map using *folium.Circle* and *folium.map.Marker*
2. Marked the success/failed launches for each site on the map using *MarkerCluster* object
3. Calculate the distances between a launch site to its proximities using *PolyLine*

EDA with Folium – Github Link

Build a Dashboard with Plotly Dash

- Summary of the plots/graphs and interactions that were added to the dashboard:

1. **Pie chart:** We have four different launch sites, and we would like to first see which one has the largest success count. Then, we would like to select one specific site and check its detailed success rate.
    - We used a launch site drop-down input component and a callback function to render "Success-pie-chart" based on selected site dropdown.
2. **Scatter chart:** We plotted a scatter plot with the payload and the launch outcome. So that, we can visually observe how payload may be correlated with mission outcomes for selected sites. In addition, we want to color-label the Booster version on each scatter point so that we may observe mission outcomes with different boosters.
    - We used a range slider to select payload and a callback function to render the "Success-payload-scatter-chart" scatter plot.

Dashboard with Plotly Dash – GitHub Link

Predictive Analysis (Classification)

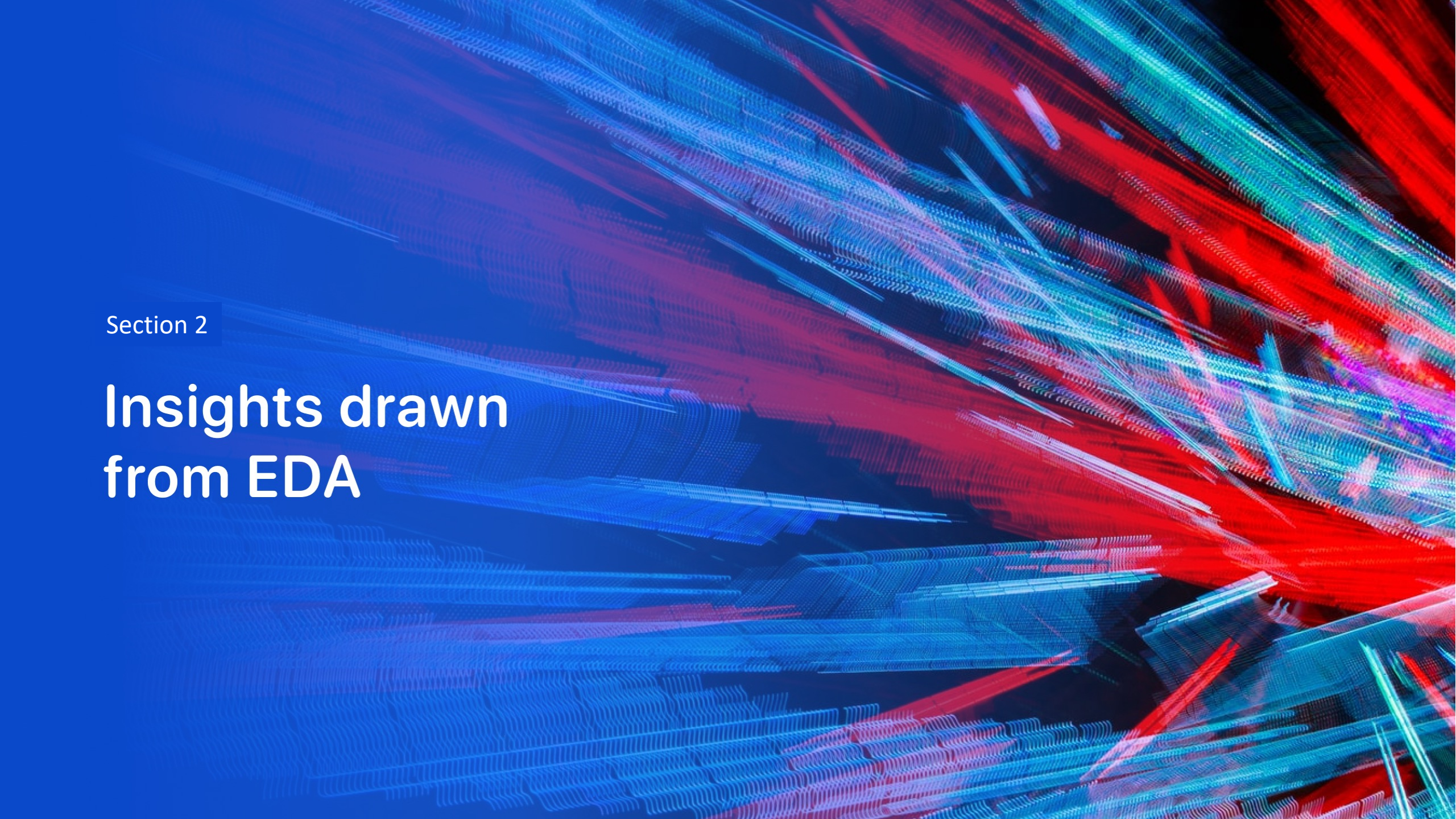
- Summary of building, evaluating and improving a classification model and finding the best performing one

1. Created a column for the class
2. Standardized the data
3. Split the data into training data and test data
4. Searched the best hyperparameter for Logistic Regression, SVM, Decision Trees and KNN
5. Searched the method performs best using test data

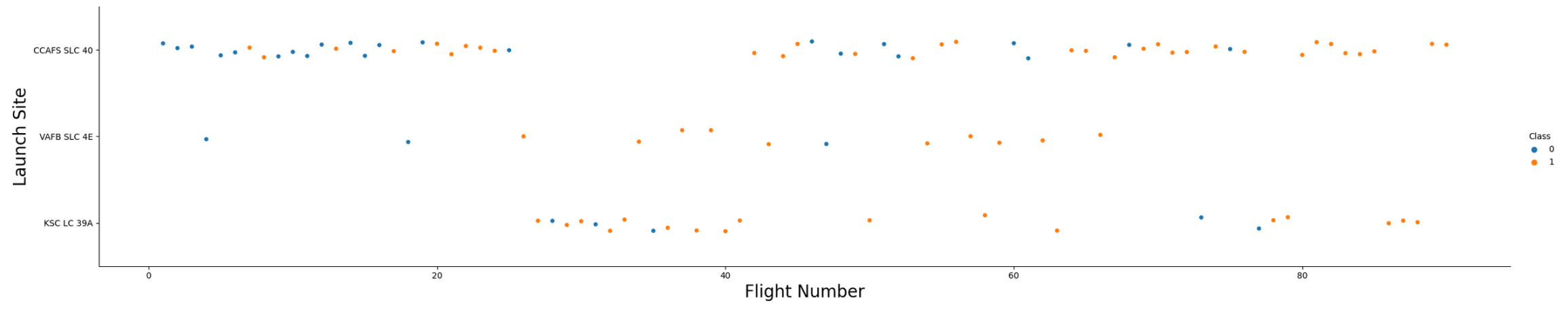
Machine Learning Prediction – Github Link

Results

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

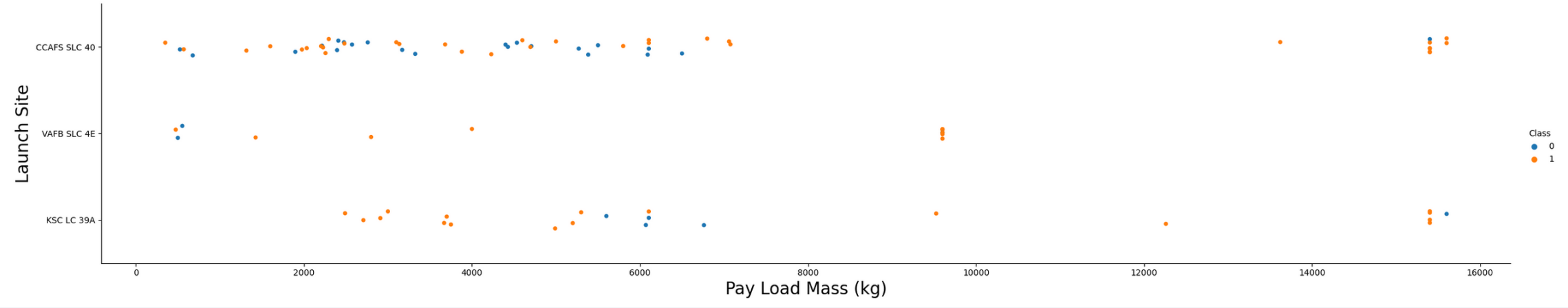

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



We can interpret that as the flight number increases in each of the 3 launch sites, so does the success rate. For instance, the success rate for the VAFB SLC 4E launch site is 100% after the Flight number 50. Both KSC LC 39A and CCAFS SLC 40 have a 100% success rates after 80th flight.

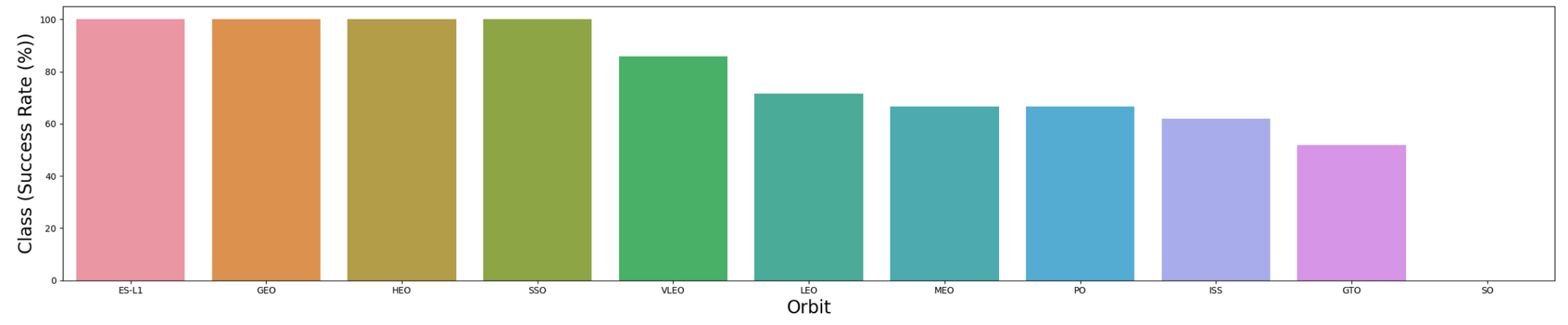
Payload vs. Launch Site



We can observe that;

- There are no rockets launched for heavy payload mass (greater than 10000) at the VAFB-SLC 4E.
- There are no rockets launched for lower payload mas (lower than 2500) at the KSC LC 39A.
- Between 7500 and 13000 payload mass there is no rockets launched at CCAFS SLC 40.

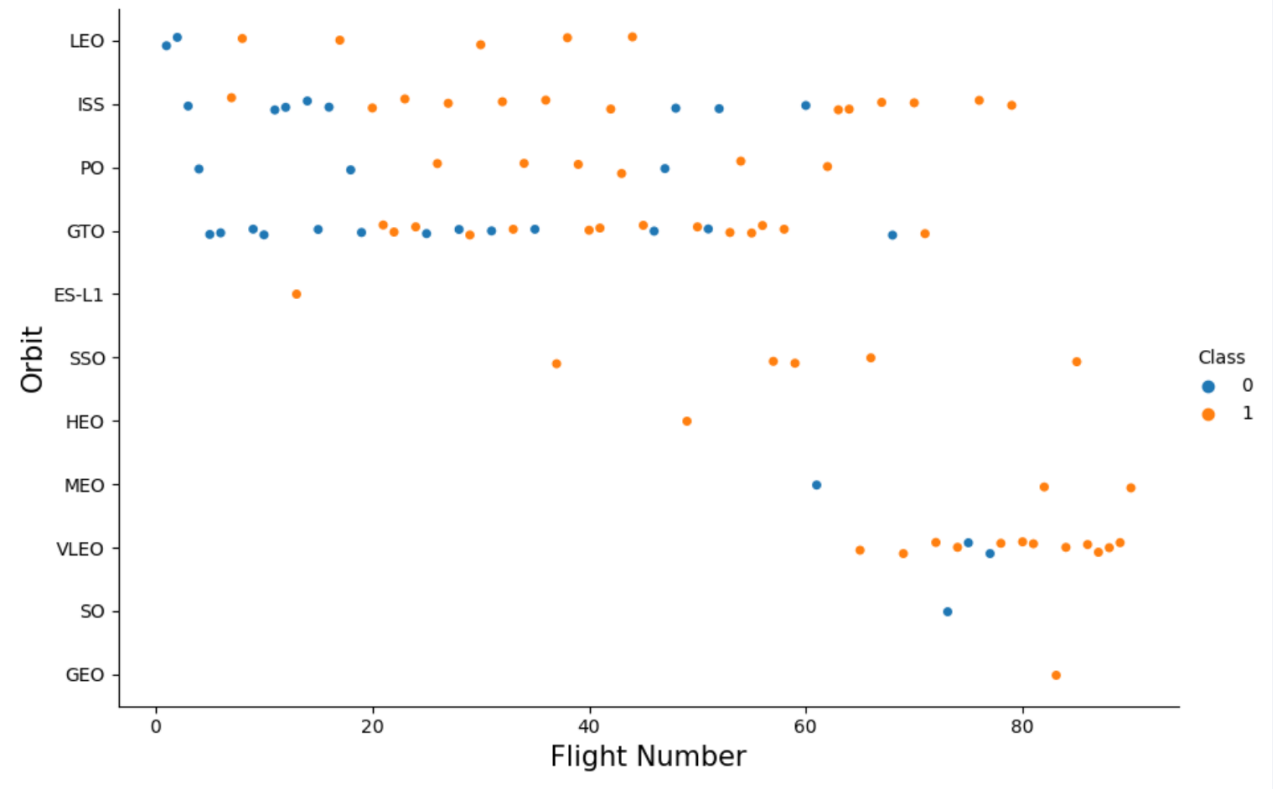
Success Rate vs. Orbit Type



We can observe that;

- ES-L1, GEO, HEO, SSO orbit type have the highest success rate.
- GTO orbit has the success rate at ~50%. Orbit SO has 0% success rate.
- But we need to investigate the flight attempt to interpret better.

Flight Number vs. Orbit Type



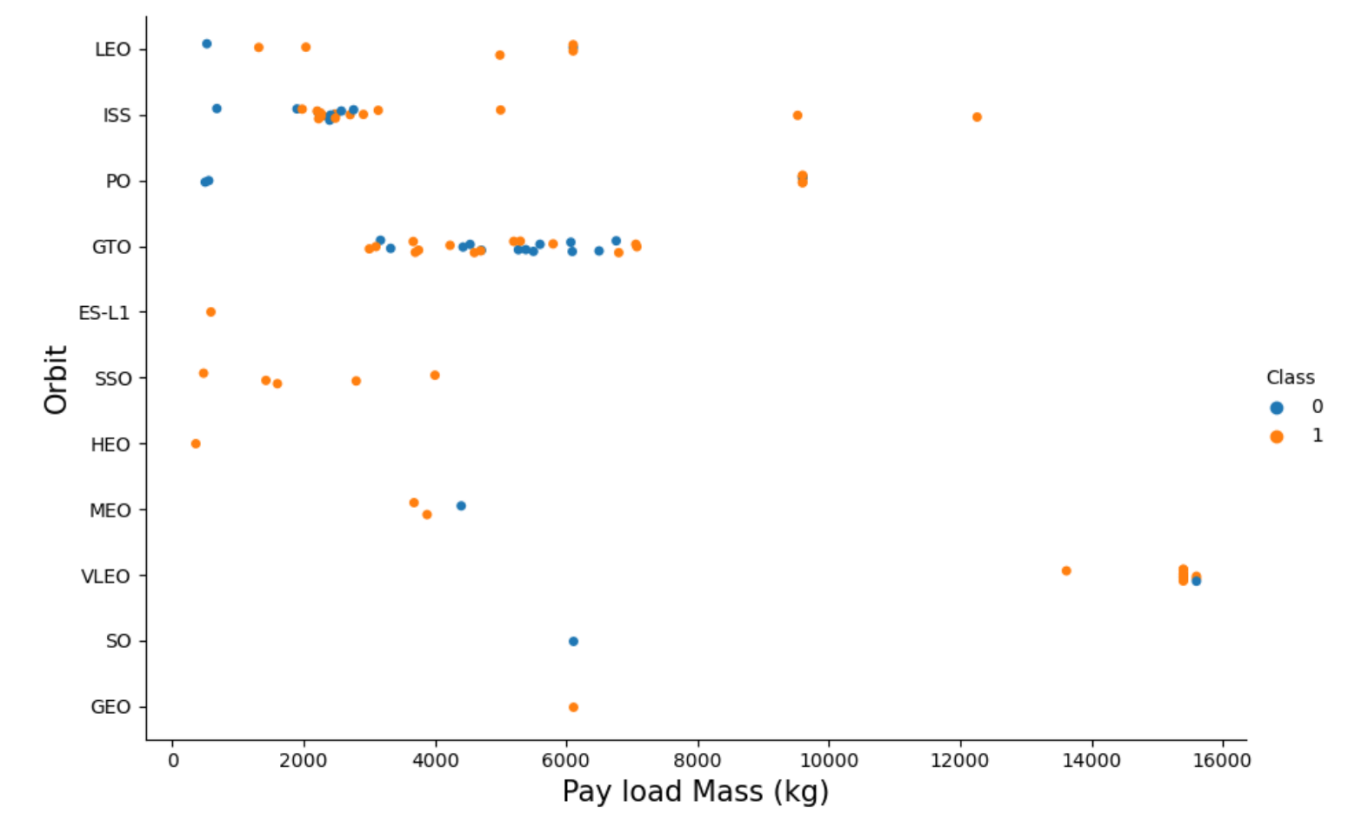
We can observe that;

- ES-L1, GEO, HEO have only one successful launch attempt with 100% success rate, while SO has one unsuccessful launch attempt with 0% success rate.

We should also see that;

- In the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

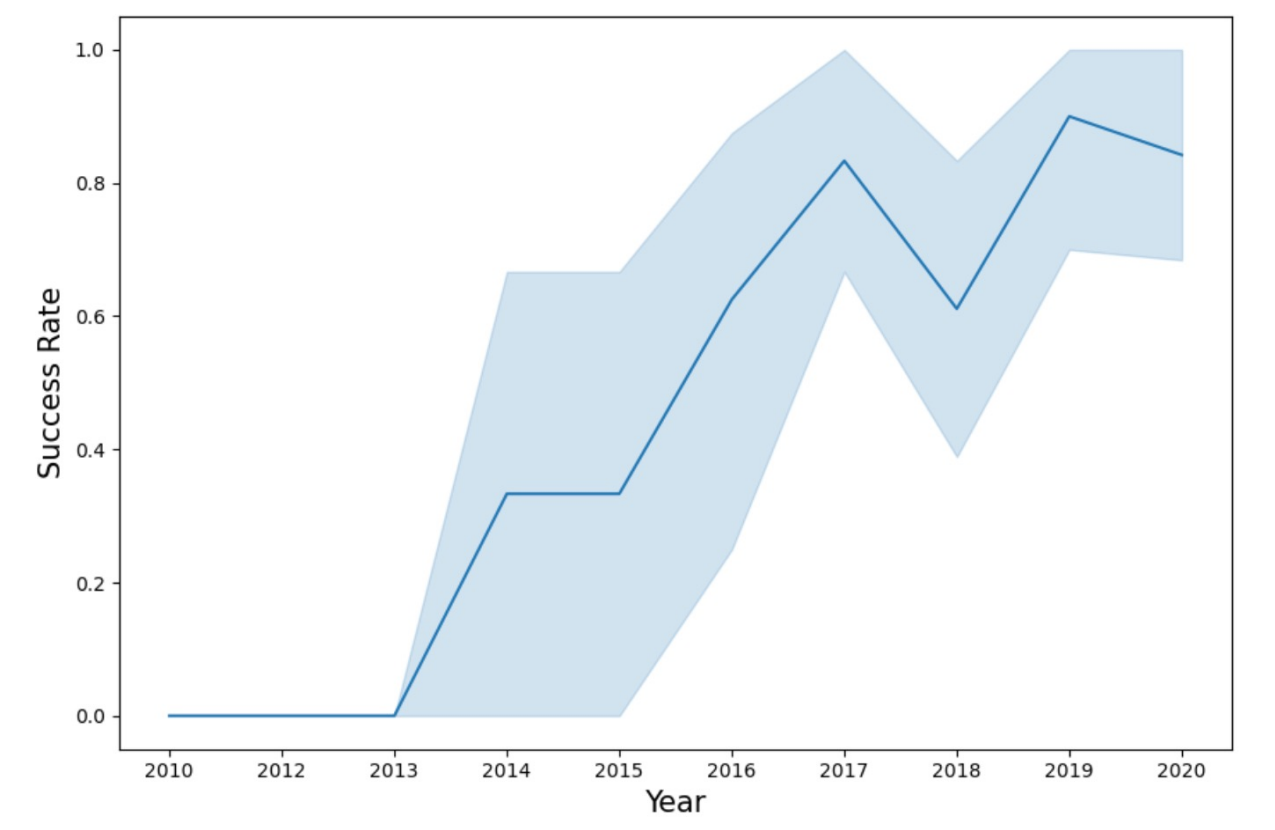
Payload vs. Orbit Type



With heavy payloads the successful landing or positive landing rate increases for Polar, LEO and ISS.

However, for GTO we cannot distinguish this well as both positive negative landings are mixed for all pay loads.

Launch Success Yearly Trend



We can observe that; as the success rate since 2013 kept increasing till 2020.

All Launch Site Names

- The names of the four unique launch sites
- DISTINCT statement was used to extract unique launch site names.

```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

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

Launch Site Names Begin with 'CCA'

- First 5 records where launch sites begin with `CCA`.
- LIMIT clause was used to list the first 5 records.

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

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

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

Total Payload Mass

- The total payload carried by boosters from “NASA (CRS)”
- SUM function and WHERE clause with LIMIT operator were used to extract total payload carried by boosters from “NASA (CRS)”

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", \
Customer FROM 'SPACEXTBL' WHERE Customer LIKE 'NASA (CRS)%';
```

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

Total Payload Mass(Kgs)	Customer
48213	NASA (CRS)

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1
- AVG function and WHERE clause with LIKE operator were used to calculate.

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass" FROM SPACEXTBL WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Payload Mass
2534.6666666666665

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad was found by using WHERE clause with LIKE operator by limiting 1.

```
%sql SELECT Date FROM SPACEXTBL WHERE "LANDING _OUTCOME" like 'Success (ground pad)%' limit 1;
```

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

Date
22-12-2015

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 was calculated by using DISTINCT statement and WHERE clause with LIKE and AND operator.

```
%sql SELECT DISTINCT Booster_Version, PAYLOAD_MASS__KG_ AS "Payload_Mass" FROM SPACEXTBL \
WHERE "Landing _Outcome" LIKE "Success (drone ship)%" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000;
```

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

Booster_Version	Payload_Mass
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes was listed by using GROUP BY clause.

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

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

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

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass was listed by using WHERE clause.

```
%sql SELECT Booster_Version, "PAYLOAD_MASS__KG_" AS "Payload_Mass" FROM SPACEXTBL \
WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Booster_Version	Payload_Mass
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

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015 were listed using SUBSTR function to extract year and month data and WHERE clause with AND operator.

```
%sql SELECT substr(Date,7,4) as "Year", substr(Date, 4, 2) as "Month", \
"Booster_Version", "Launch_Site", "Landing _Outcome" FROM SPACEXTBL \
WHERE substr(Date,7,4)='2015' AND "Landing _Outcome" = 'Failure (drone ship)';
```

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

Year	Month	Booster_Version	Launch_Site	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- 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 was ranked.
- COUNT function, WHERE clause with BETWEEN operator and GROUP BY clause with DESC operator were used.

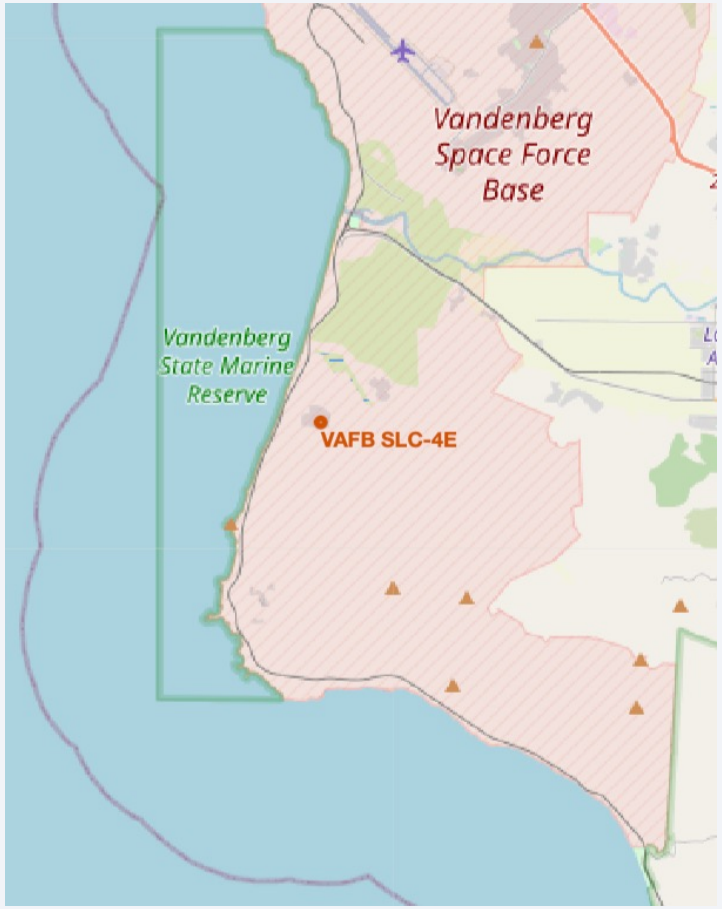
```
%sql SELECT "Landing _Outcome" as "Landing Outcomes", \
COUNT("Landing _Outcome") as "Count" FROM SPACEXTBL \
WHERE Date BETWEEN '04-06-2010' AND '20-03-2017' \
group by "Landing _Outcome" order by count("Landing _Outcome") desc;
```

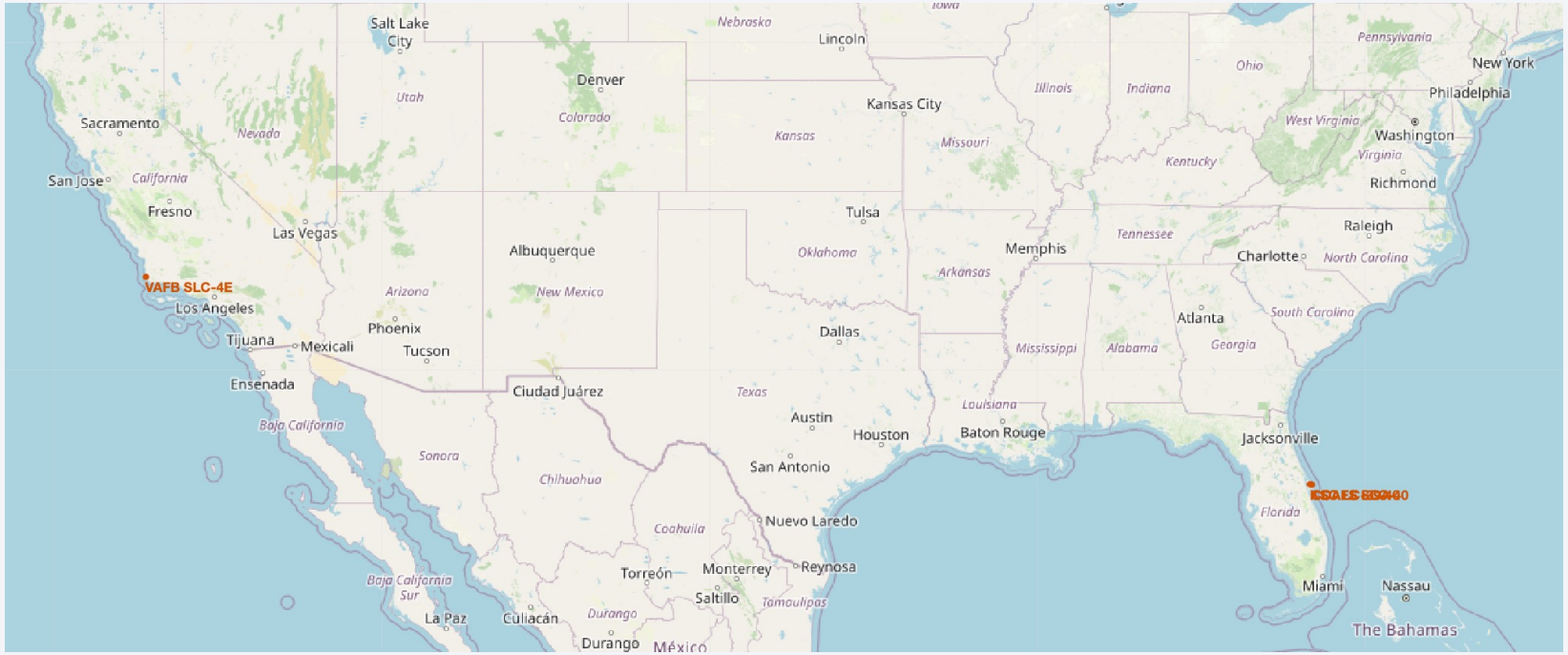
```
 * sqlite:///my_data1.db
Done.
```

Landing Outcomes	Count
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	6
Failure (drone ship)	4
Failure	3
Controlled (ocean)	3
Failure (parachute)	2
No attempt	1


Section 3
Launch Sites
Proximities Analysis

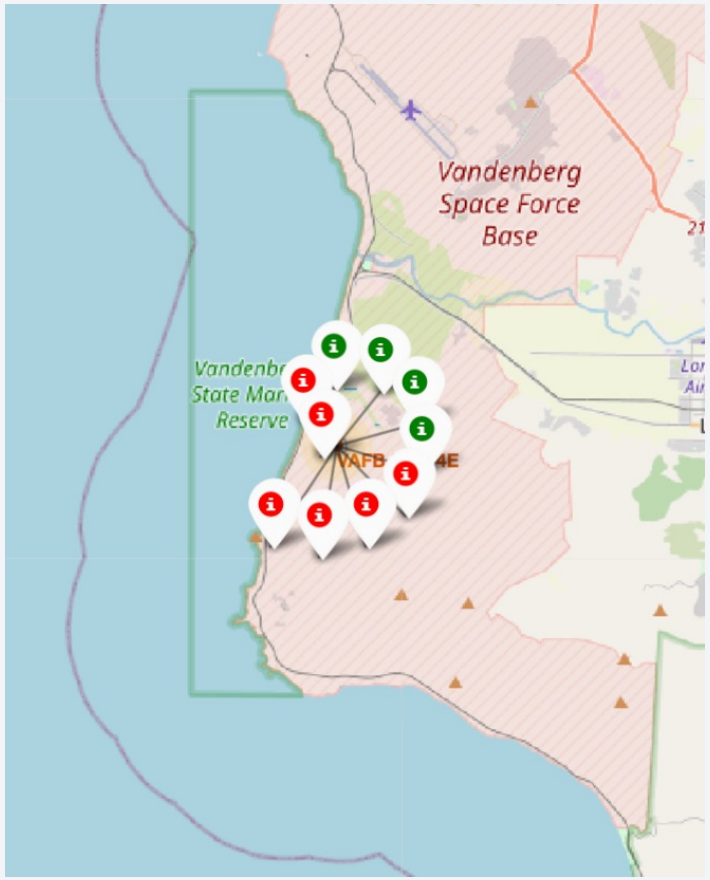
All launch sites





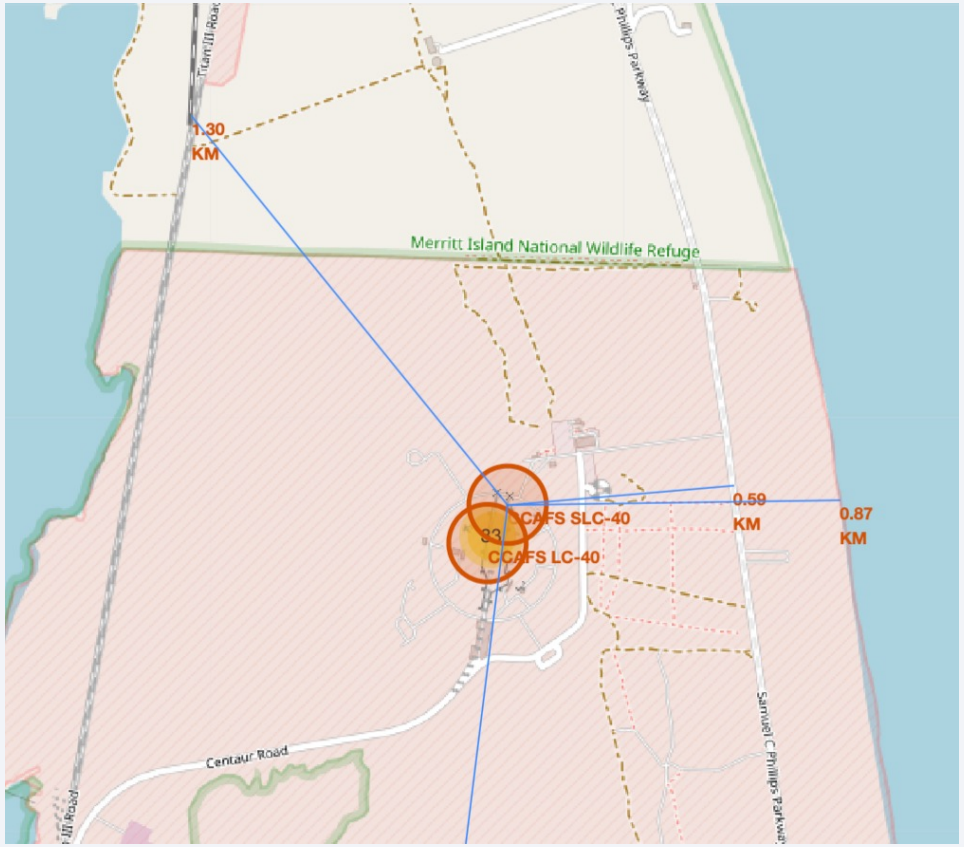
- All launch sites are in proximity to the Equator line.
- All launch sites are in very close proximity to the coast.

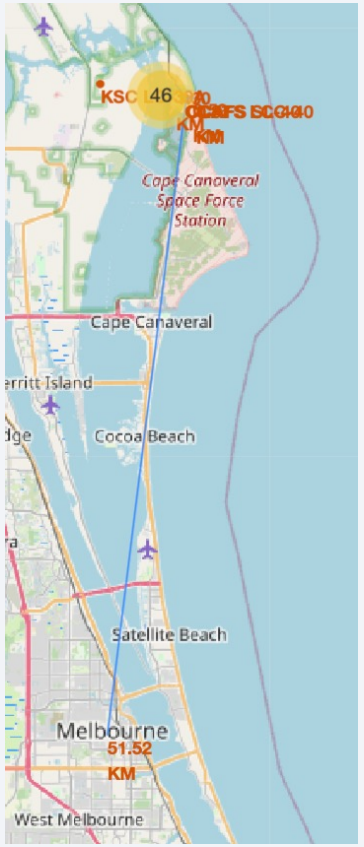
The successful/failed launches for each site on the map



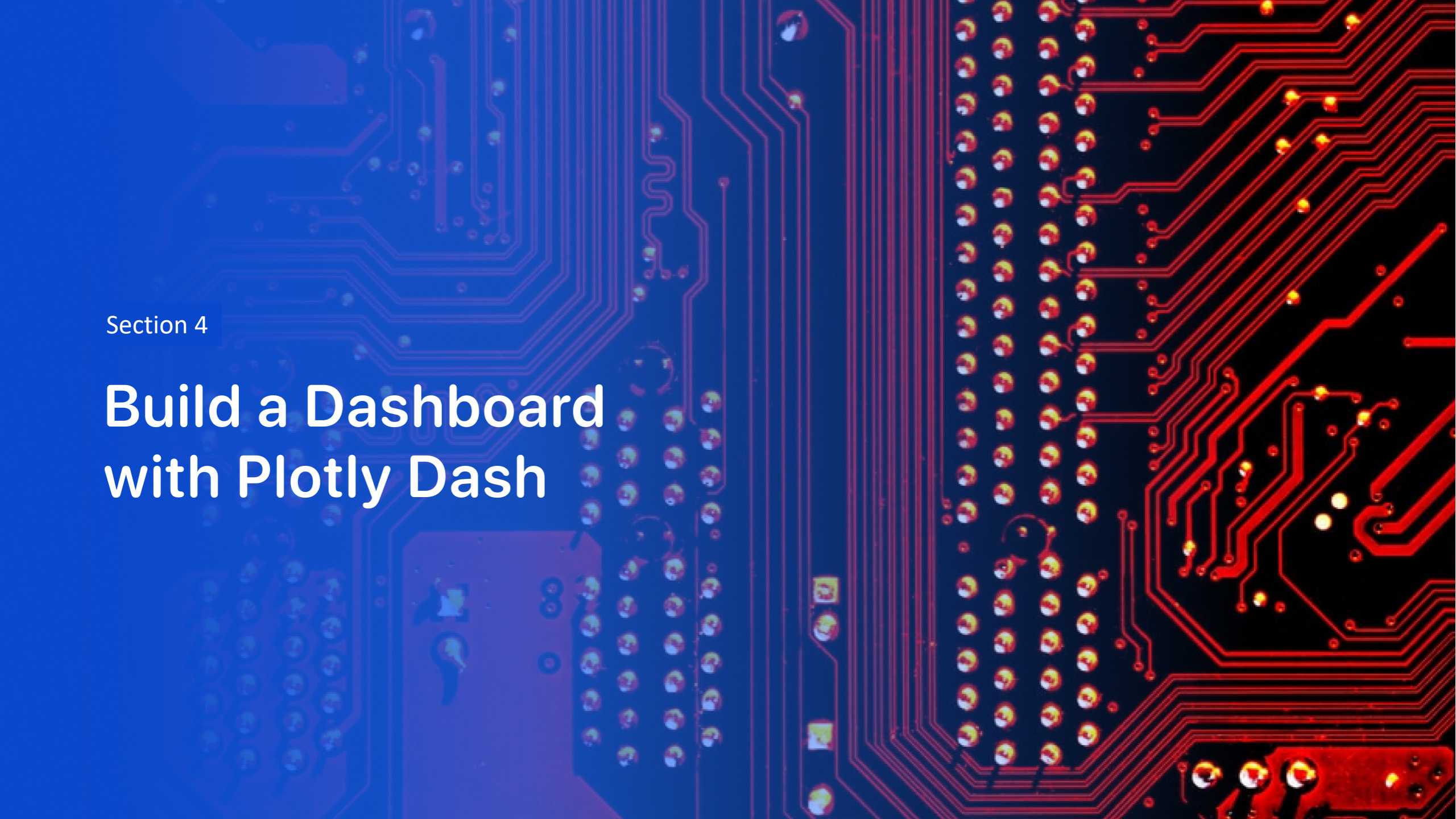
- We created markers for all launch records. If a launch was successful, then we used a green marker and if a launch was failed, we used a red marker.
- From the color-labeled markers in marker clusters, we can identify which launch sites have relatively high success rates.

The distances between a launch site to its proximities



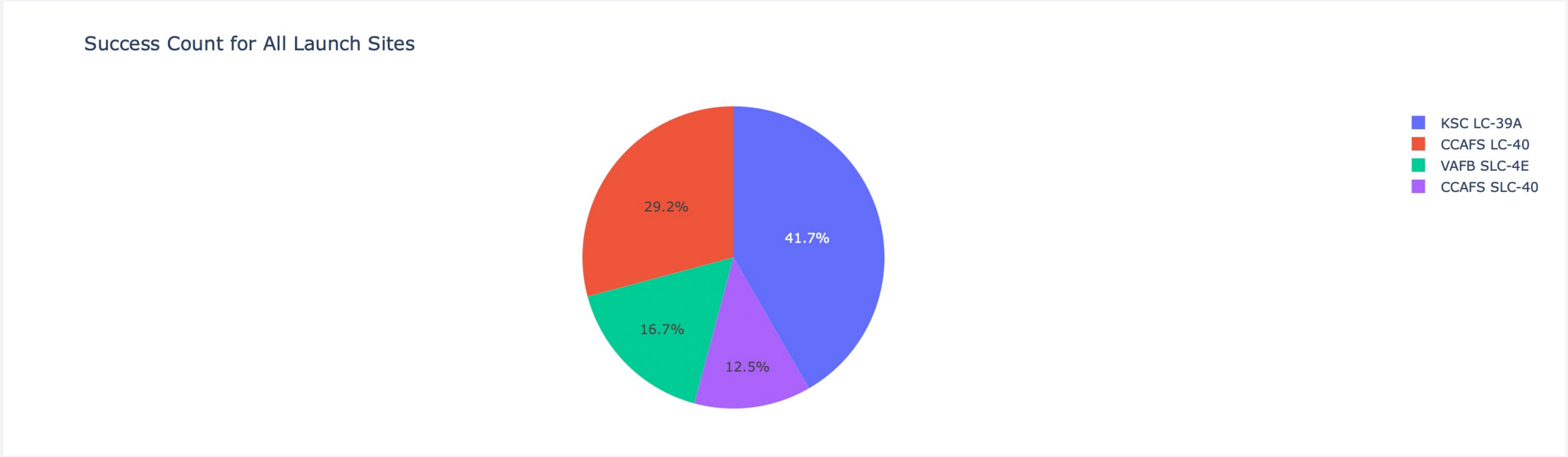


- Launch sites are in close proximity to railways and highways in order to organize and transport heavy logistic operations.
- They are in close proximity to coastline. There could be two reasons for this. First one is to emergency landing on the sea surface, second one is to left the debris to the ocean in case of any accident.
- They keep certain distance away to cities so that populated areas are not affected in case of any accident.


Section 4
Build a Dashboard with Plotly Dash

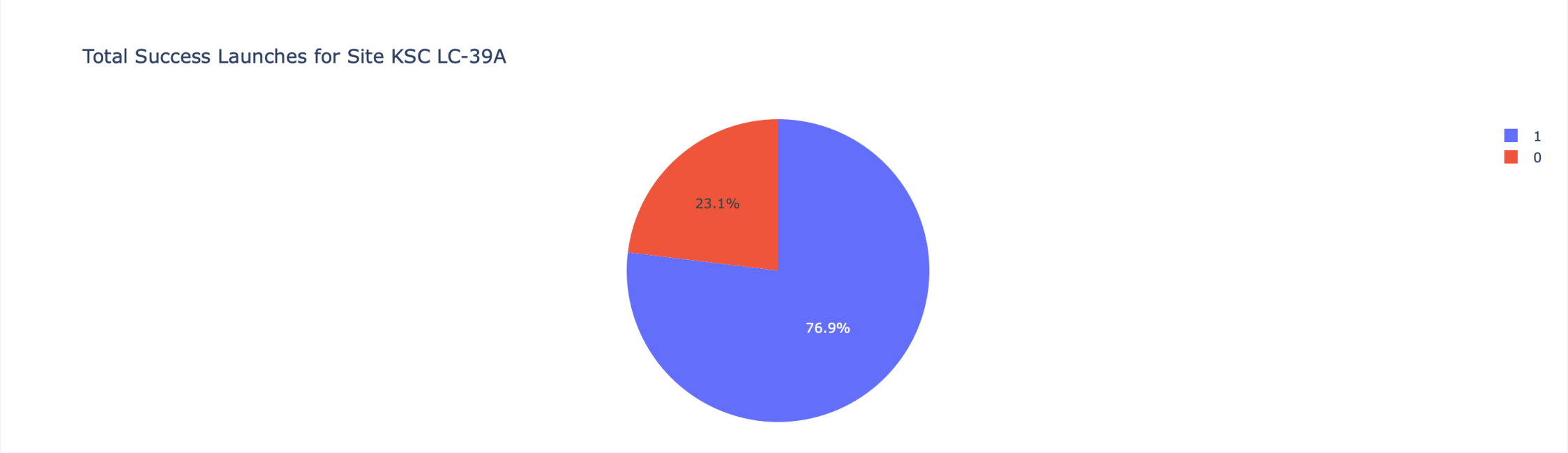
Launch success count for all sites

- KSC LC-39A has the highest launch success rate.



KSC LC-39A launch site

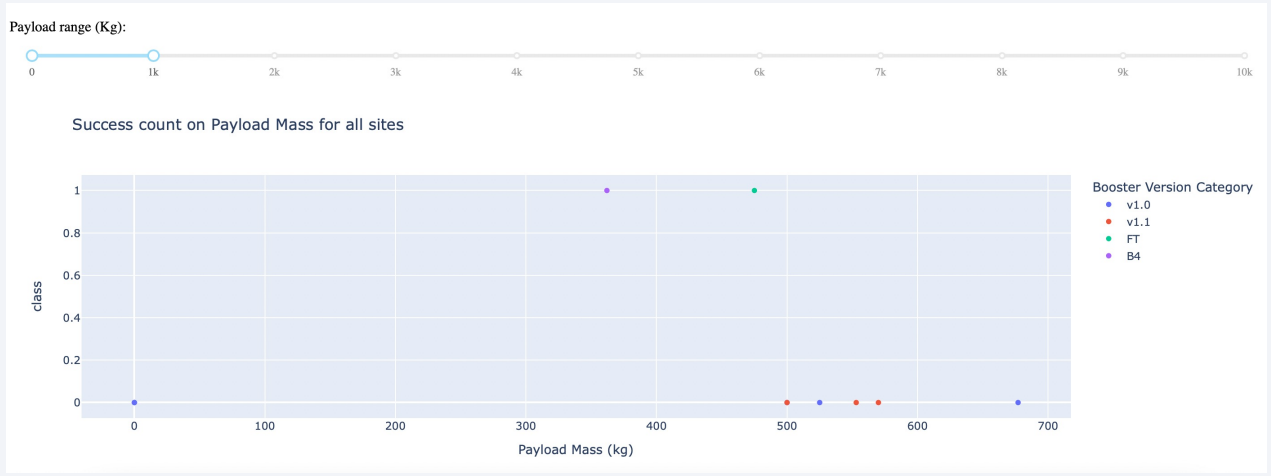
- The pie chart shows the launch site (KSC LC-39A) which has the highest launch success rate.



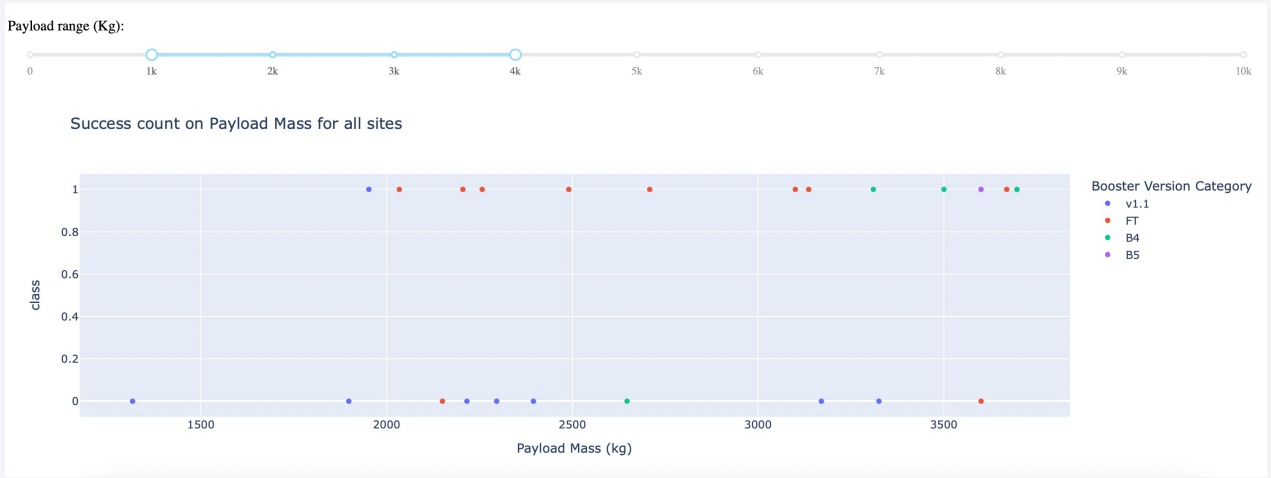
Payload vs. Launch Outcome scatter plot for all sites

- Screenshots show the Payload vs. Launch Outcome scatter plot for all sites with different payload selected in the range slider.

- 0 – 1000 kg payload range
  - Only B4 and FT booster versions are successful in this range



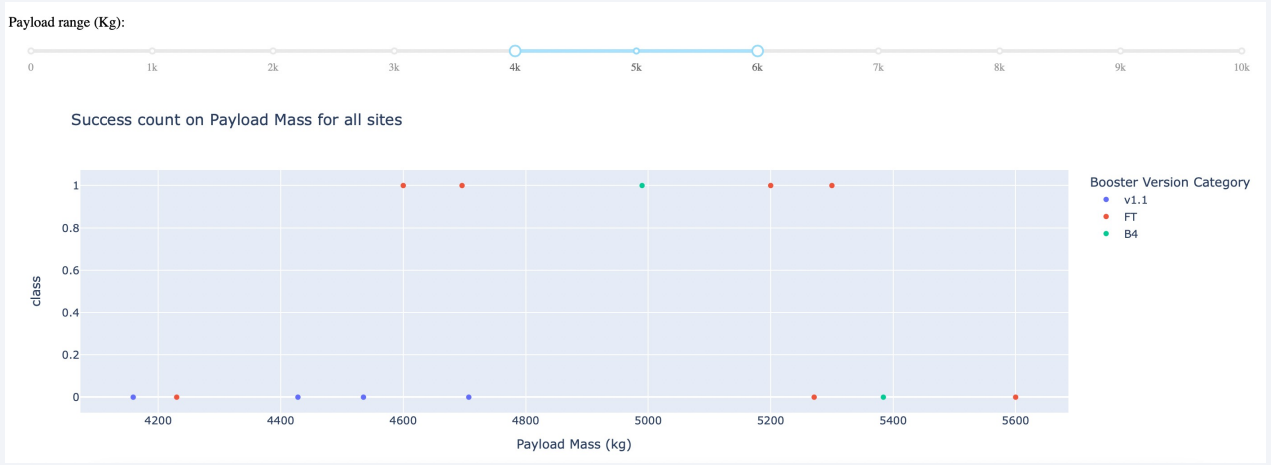
- 1000 – 4000 kg payload range
  - FT booster version has the highest success count



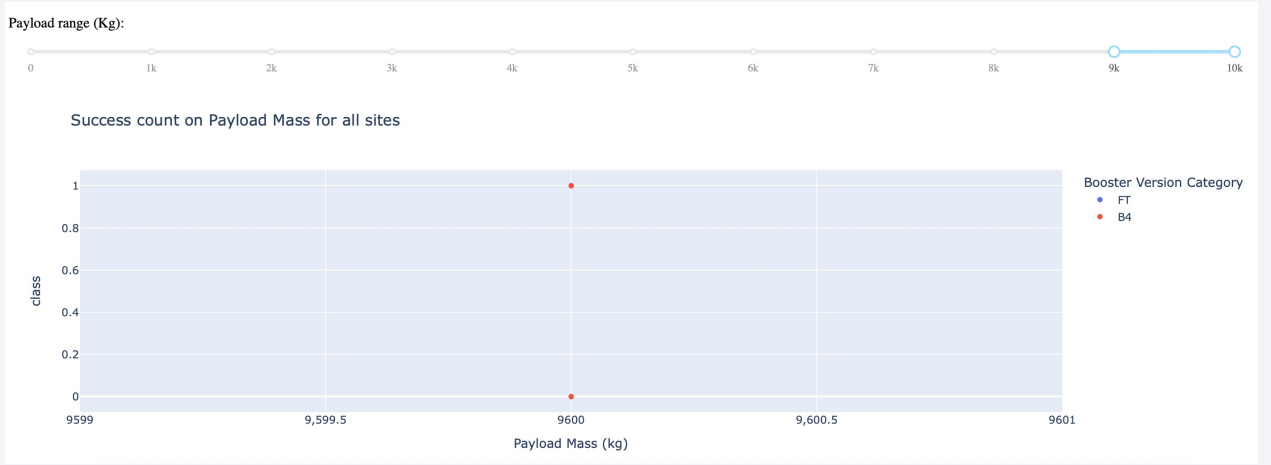
Payload vs. Launch Outcome scatter plot for all sites

- Screenshots show the Payload vs. Launch Outcome scatter plot for all sites with different payload selected in the range slider.

- 4000 – 6000 kg payload range
  - FT booster version has the highest success count



- 9000 – 10000 kg payload range
  - FT and B4 booster versions have the same success and fail count

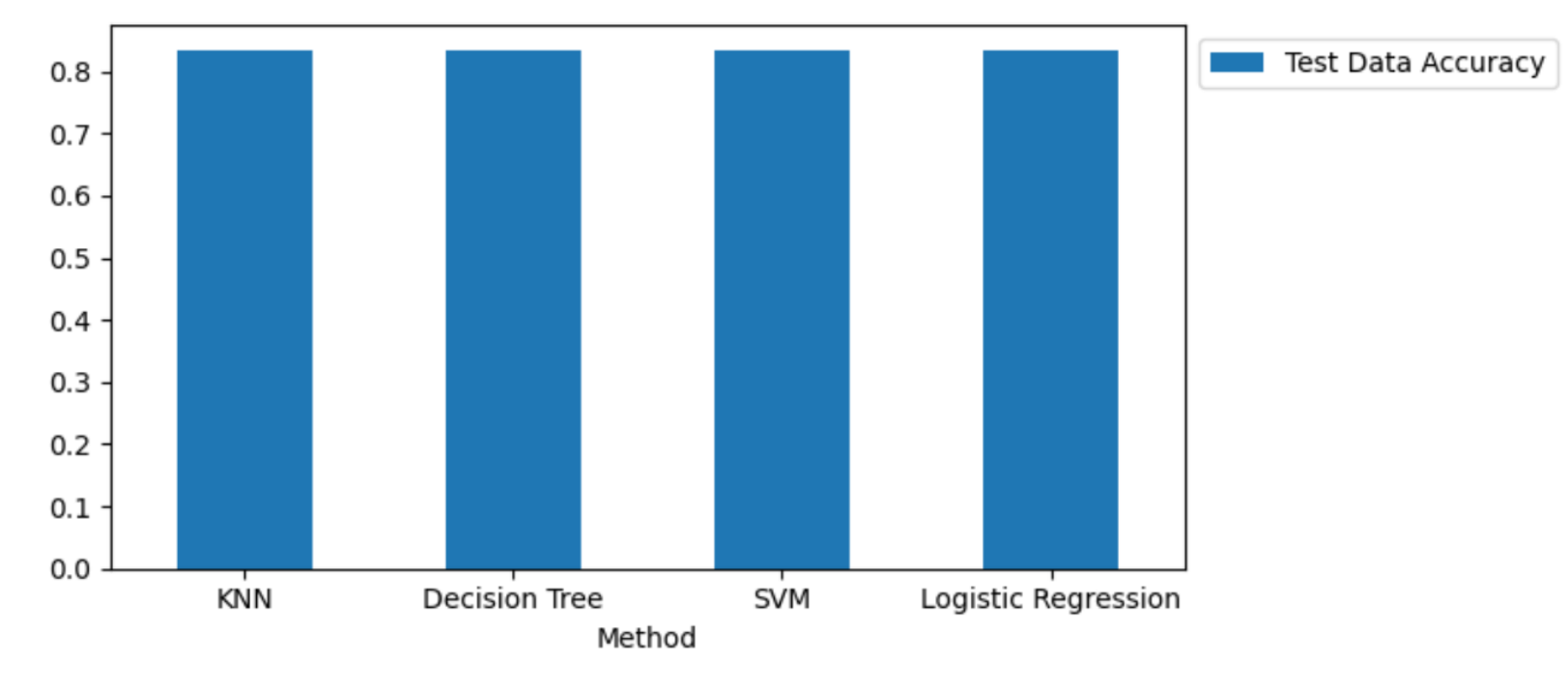


Section 5

Predictive Analysis (Classification)

Classification Accuracy

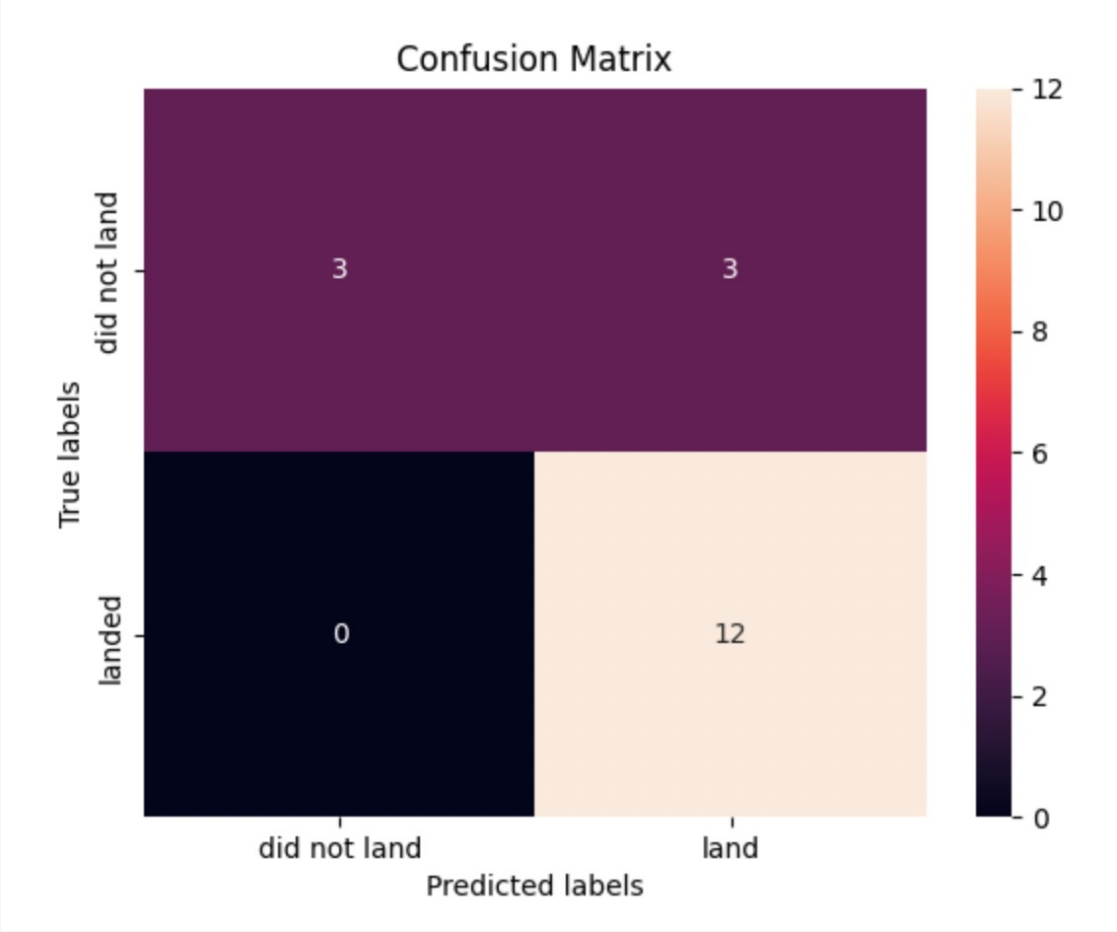
- The accuracy is the same for all models for the test data.



Method	Test Data Accuracy
KNN	0.833333
Decision Tree	0.833333
SVM	0.833333
Logistic Regression	0.833333

Confusion Matrix

- The confusion matrix is the same for all models.



Conclusions

- We can conclude that:
  - We can interpret that as the flight number increases in each of the 3 launch sites, so does the success rate.
  - ES-L1, GEO, HEO, SSO orbit type have the highest success rate.
  - Launch success rate since 2013 kept increasing till 2020.
  - Launch sites are in close to railways and highways in order to organize and transport heavy logistic operations. All launch sites are in close to coastline. They keep certain distance away to cities so that populated areas are not affected in case of any accident.
  - All four machine learning algorithms gave the same test data accuracy. We can use any of them to predict if the first stage will land and calculate the cost of a launch.

Appendix

- All other relevant notebooks, scripts and datasets are provided in this GitHub link.

Data Science Capstone Project

Thank you!