IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Pranav Joshi

28th October 2024

Outline

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

Executive Summary

- Summary of methodologies:
    1. Data collection using an API and Web Scraping
    2. Data wrangling
    3. Exploratory Data Analysis with SQL
    4. Exploratory Data Analysis with Visualizations
    5. Interactive Visual Analytics and Dashboards using Folium and Plotly Dash
    6. Machine Learning Predictive Analysis (Classification)
- Summary of all results:
    1. Interactive Dashboards for the available data
    2. Best hyperparameters for the various classification algorithms
    3. The best performing method amongst the algorithms on test data

Introduction

The current leading organization in the space industry, SpaceX, offers rocket launches at relatively much lower costs as compared to other organizations. (Falcon 9 for 62 million dollars as compared to around 165 million dollars by other organizations). This is largely attributed to their reuse of the first stage for further launches, thus saving on additional costs. As part of a rival organization, SpaceY, the main goal of this project is to create a Machine Learning Pipeline to predict if the first stage will land successfully based on available data.

Problems that required answers:

1. Identification of factors affecting landing outcomes.
2. Interpretation and analysis of the variables in the available data.
3. The exact conditions required to get the best possible outcome.

Section 1

Methodology

Methodology

Executive Summary

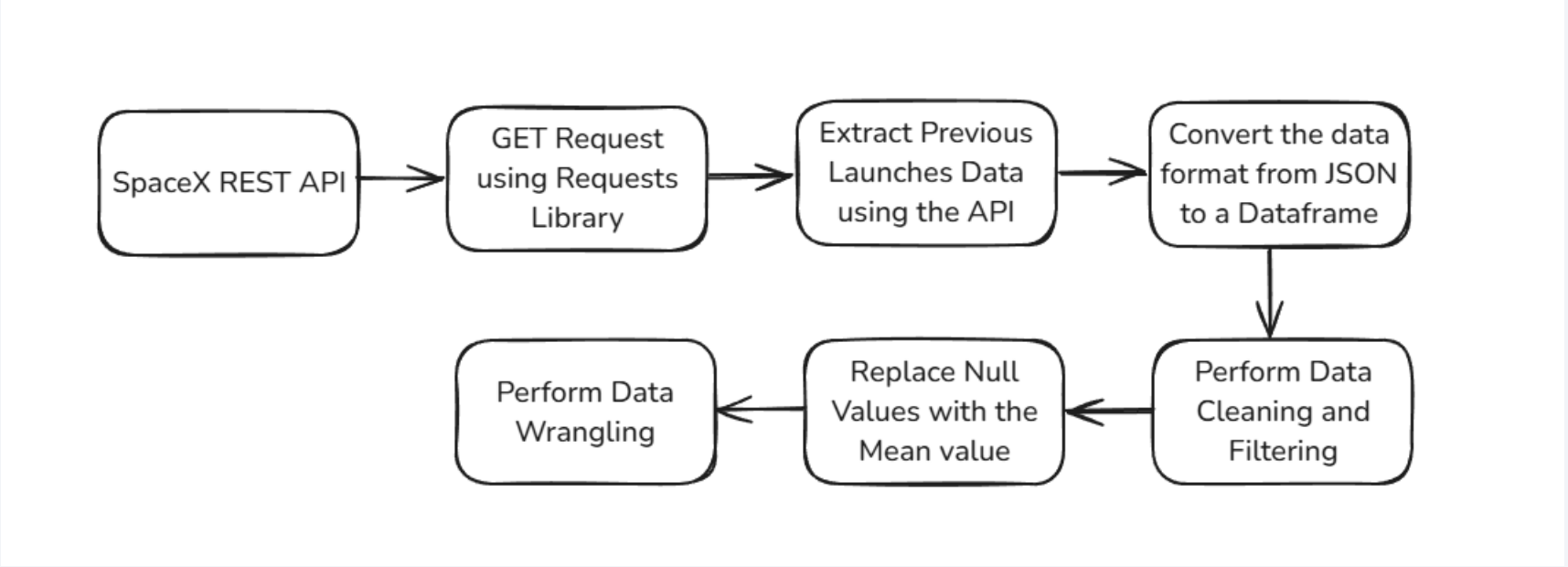
- Data collection methodology:
  - Data was collected using the SpaceX REST API and Web Scraping from Wikipedia using BeautifulSoup.
- Perform data wrangling
  - The acquired data was assigned labels depending on the outcome and exact number of cases per category were identified.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Different Machine Learning models were trained to predict the landing outcomes and compared to identify the best model for the task.

Data Collection

- The data was collected using SpaceX REST API and Web Scraping from Wikipedia using BeautifulSoup.
- The process of data collection using the REST API involved the following steps:
  - Use the GET Request to receive data
  - Extract required information as JSON Object, then convert to Pandas Dataframe
  - Clean, filter data and check for null values. Then proceed with Data Wrangling.
- The process of data collection using the Web Scraping involved the following steps:
  - Use BeautifulSoup to extract required data as HTML table
  - Parse the table, convert to Pandas Dataframe then proceed with Data Wrangling.

Data Collection – SpaceX API

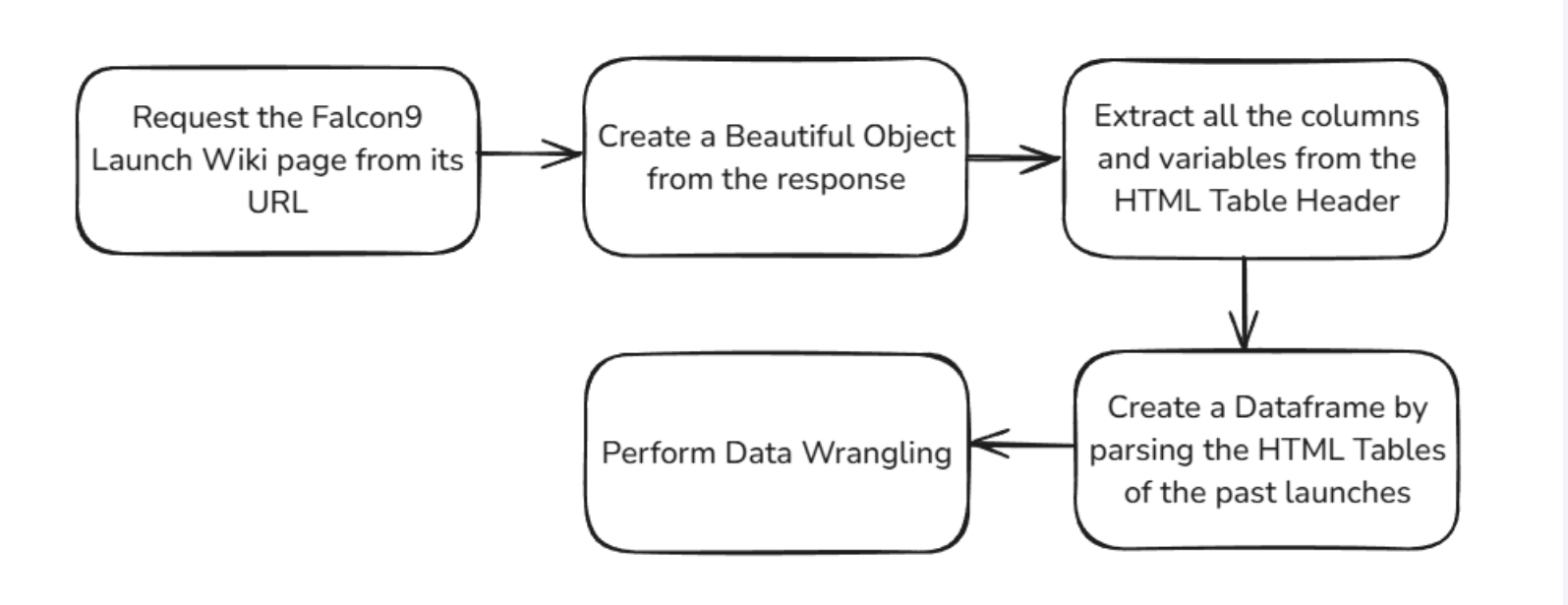
- This process involved collecting data in the correct format by using the SpaceX REST API.



- Data collection API - GitHub

Data Collection - Scraping

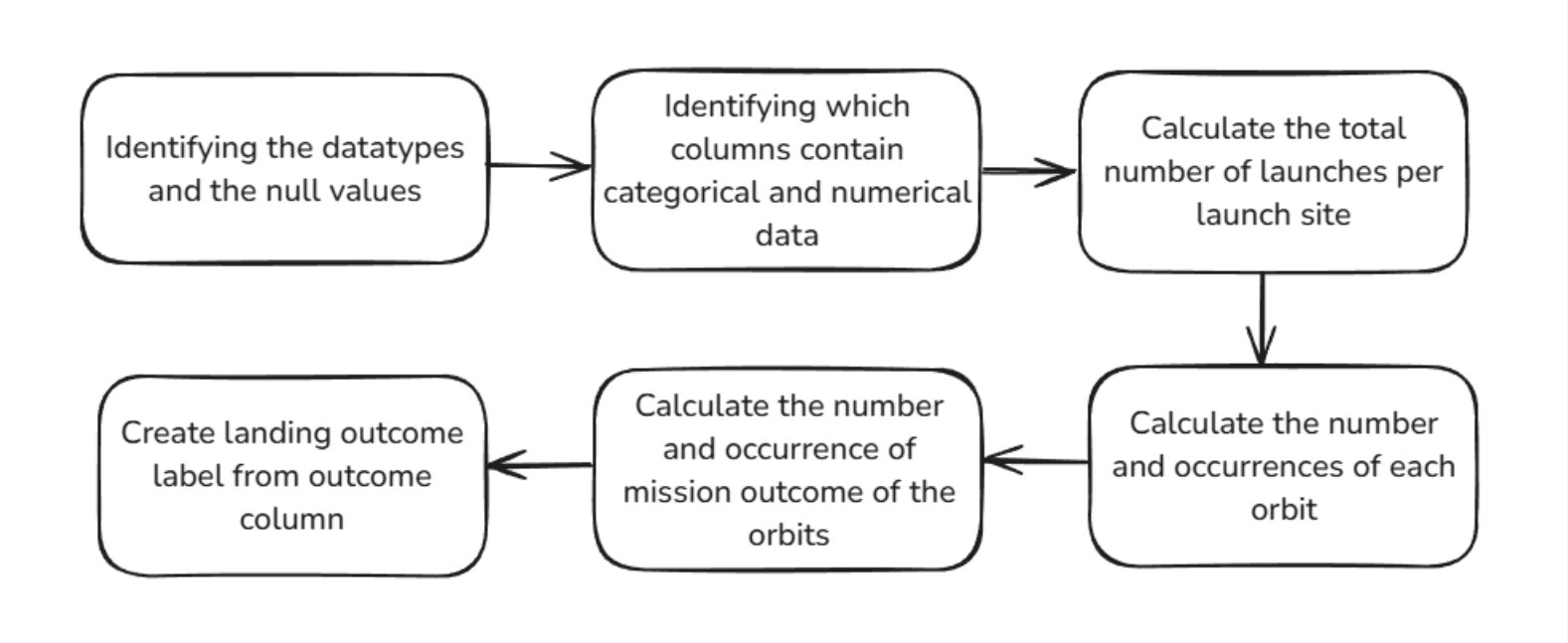
- This process involved collecting data by performing Webscraping using BeautifulSoup to collect past launches of Falcon 9 from Wikipedia pages.



- Data Collection Webscraping - GitHub

Data Wrangling

- Data Wrangling involved cleaning data and transforming the raw data for EDA (Exploratory Data Analysis).



- Data Wrangling - GitHub

EDA with Data Visualization

- While performing EDA, the following graphs were introduced – Scatterplots, Bar graph and Line graph – each having their own use case.
- Following is a brief summary of the graphs introduced –
  - Scatter Plots to find relationship between the following:
    1. Flight Number and Payload
    2. Flight Number and Launch Site
    3. Payload and Launch Site
    4. Flight Number and Orbit Type
    5. Payload and Orbit Type

  Scatterplots help to find dependency between attributes and thus look for factors affecting an outcome.

EDA with Data Visualization

- Bar graph to visualize the success rate of each orbit type. Bar Graphs are the easiest way to compare results, in this case allowing us to find out the probability of success for each orbit type.
- Line graph to visualize the success rate over the years for the launches. Line graphs are used to show the pattern followed by an attribute over time. Here, it helped to visualize how the success rates increased tremendously over the years 2015-2017 and reached the peak in 2019.

- After performing data visualization, feature engineering was performed to predict success rates in the future. In feature engineering the features that would prove useful for the prediction analysis were selected. Dummy variables were created to categorical columns.

EDA with SQL

- SQL Queries were performed to analyze and acquire valuable insights from the data. The queries performed include:
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - 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

EDA with SQL

- List the names of the booster versions which have carried the maximum payload mass using a subquery
- 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
- 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

- The above SQL queries helped in identifying numerous important facts within the data such as the launch sites, success rate of the previous launches and the landing outcomes for each launch.
- The above SQL queries can be found in the following notebook:

EDA with SQL - GitHub

Build an Interactive Map with Folium

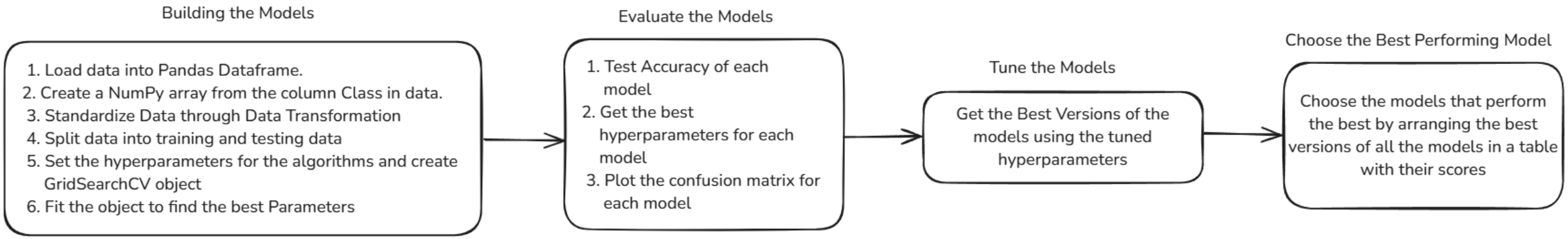
- An interactive map was created using Folium to visualize the launch site locations. The latitude and longitude was taken from the data to mark the locations on the map. The objects used include:
  - Circle markers were added to highlight the areas with a circle to identify the sites easily with their corresponding labels. (folium.Circle and folium.Marker)
  - Launch Outcomes were marked with classes 0/1 for Failure/Success with Red and Green Markers. (MarkerCluster)
  - Coordinate of the Mouse over a point on the Map to easily identify specific locations. (MousePosition)
  - A line was drawn between the launch site and the closest city, railway, highway and coastline. (folium.PolyLine)
- These objects allowed to identify the exact locations of the sites as well as the outcome of the launches from the sites.
- Interactive Map with Folium - GitHub

Build a Dashboard with Plotly Dash

- A Dashboard containing a Pie Chart and a Scatterplot was created using Plotly Dash to provide insights into the relationship between the Successful launches and the Payload Mass used for particular launch sites.
- The dashboard featured a drop down option to select the specific launch site. The plots were then rendered with the help of a callback function, allowing us to see the successful launches in the pie chart and the scatterplot showcasing the success counts with respect to Payload Mass.
- There is also a slider allowing us to select a particular range of the Payload Mass to identify the success counts accordingly for different booster versions.
- Dashboard with Plotly Dash - GitHub

Predictive Analysis (Classification)

- Finally, a Machine Learning (ML) pipeline was built to predict if the first stage will land given the data. This involved creating several ML models, tuning their hyperparameters and finally comparing the results.

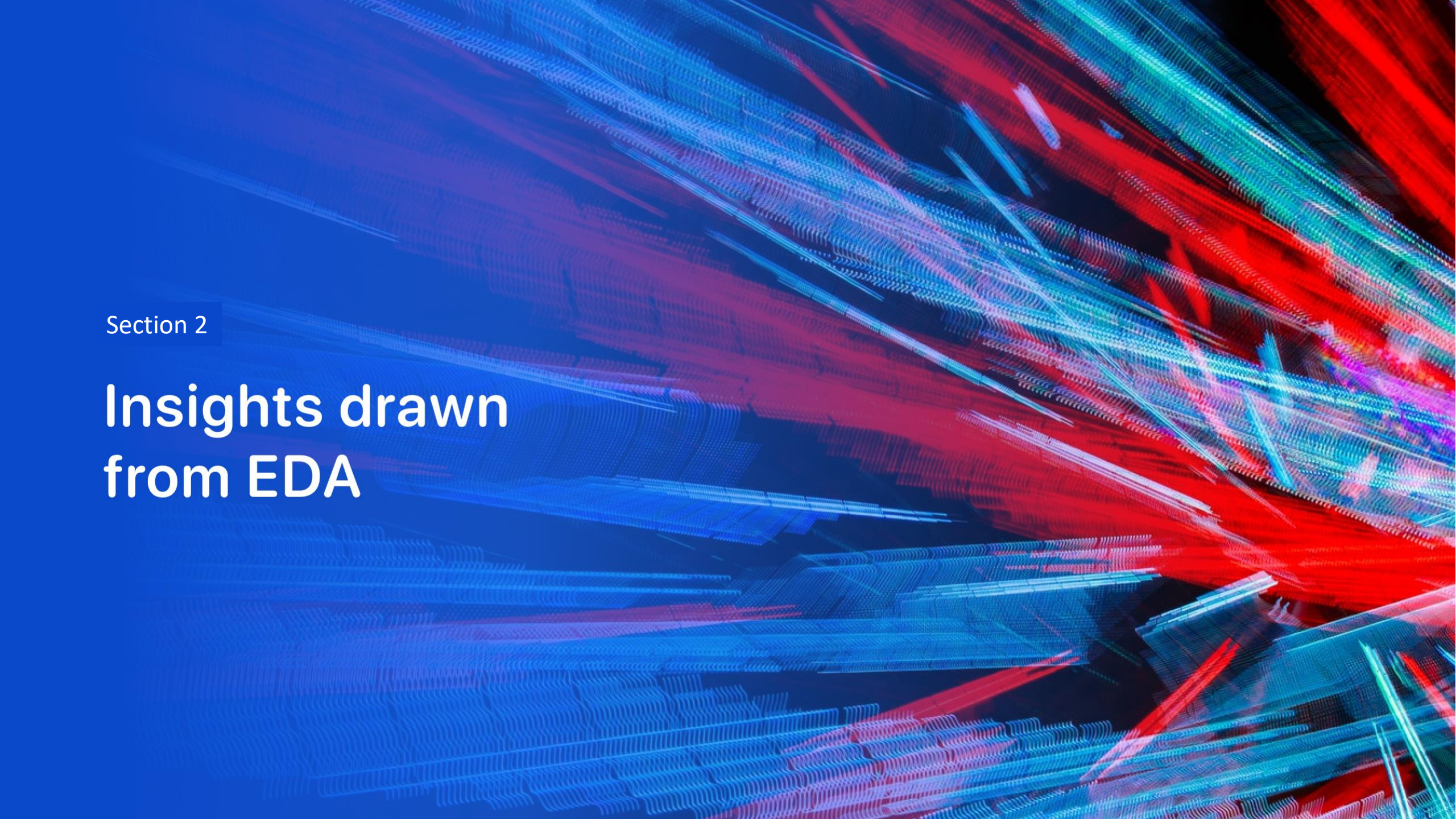


- Predictive Analysis - GitHub

Results

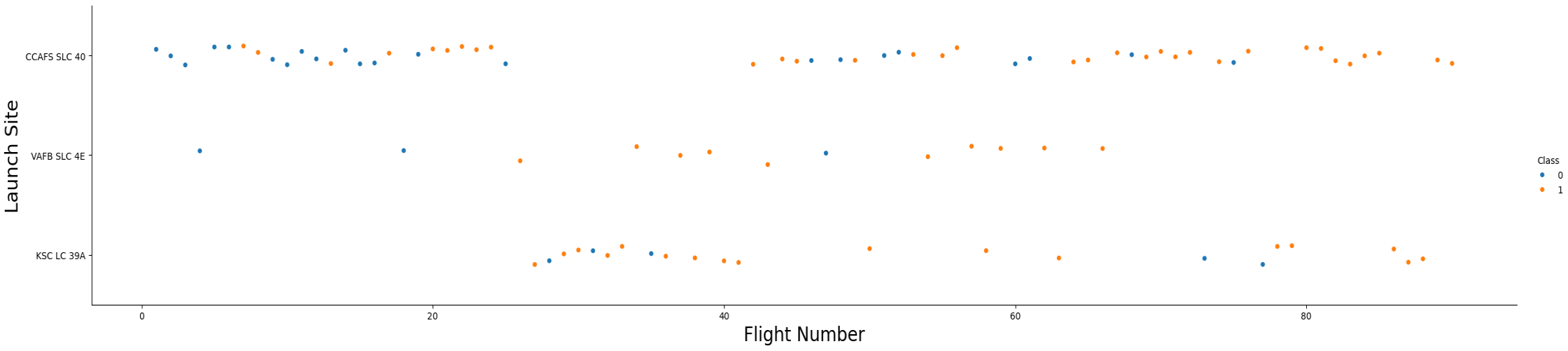
The following were the key results obtained:

- Exploratory data analysis (EDA) results: Important information required for comparing the results of the launches and the relationship between the factors involved.
- Interactive analytics demo in screenshots: Insights regarding the launch site locations and the success rates of the launches.
- Predictive analysis results: The best performing model to predict future launches based on the involved factors.


Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

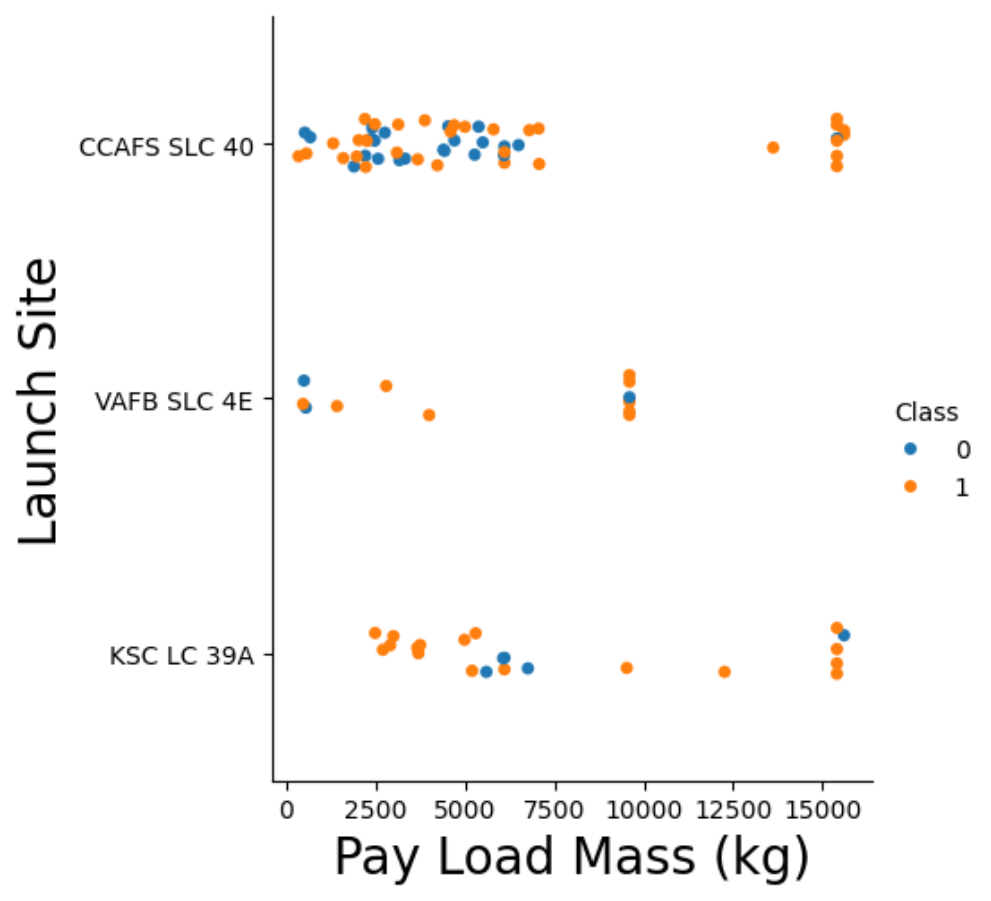
- Scatter plot of Flight Number vs. Launch Site



- It can be seen that increasing Flight Number resulted in increasing successful launches. The majority of the launches are from launch site CCAFS SLC 40 where ratio successful attempts post 60 flights became the highest.
- VAFB SLC 4E and KSC LC 39A have comparatively lesser number of flights but have higher success probability than the former.

Payload vs. Launch Site

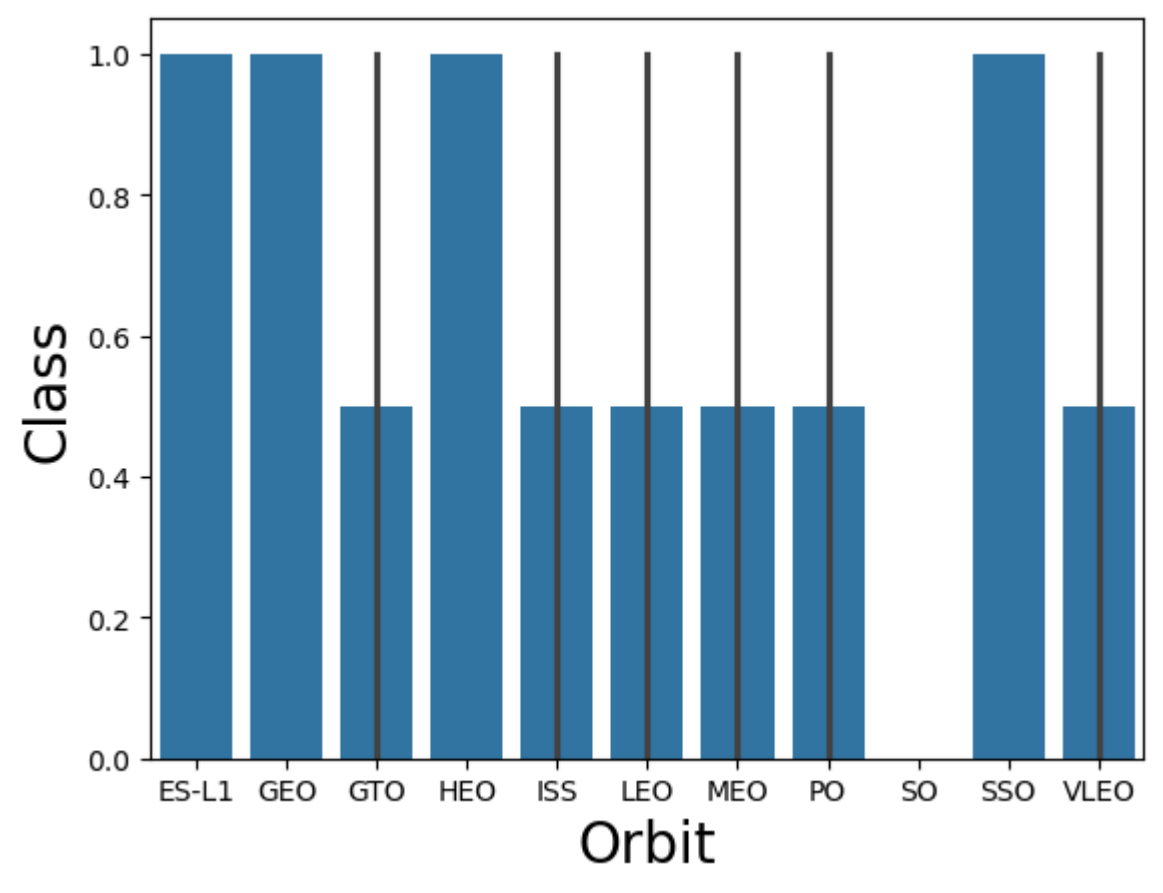
- Show a scatter plot of Payload vs. Launch Site



- For the VAFB-SLC launch site, there are no rockets launched for heavy payload mass (greater than 10000).
- CCAFS SLC 40 does not have launches for payload mass between 7500 and 12500 kg.
- KSC LC 39A does not have any rocket launches for lower payload mass (less than 2500 kg)
- The highest ratio of success rates was seen when the payload mass was around 15000 Kg for CCAFS SLC 40 launch site.
- However, there is no clear pattern to suggest that payload mass singularly affects success rate.

Success Rate vs. Orbit Type

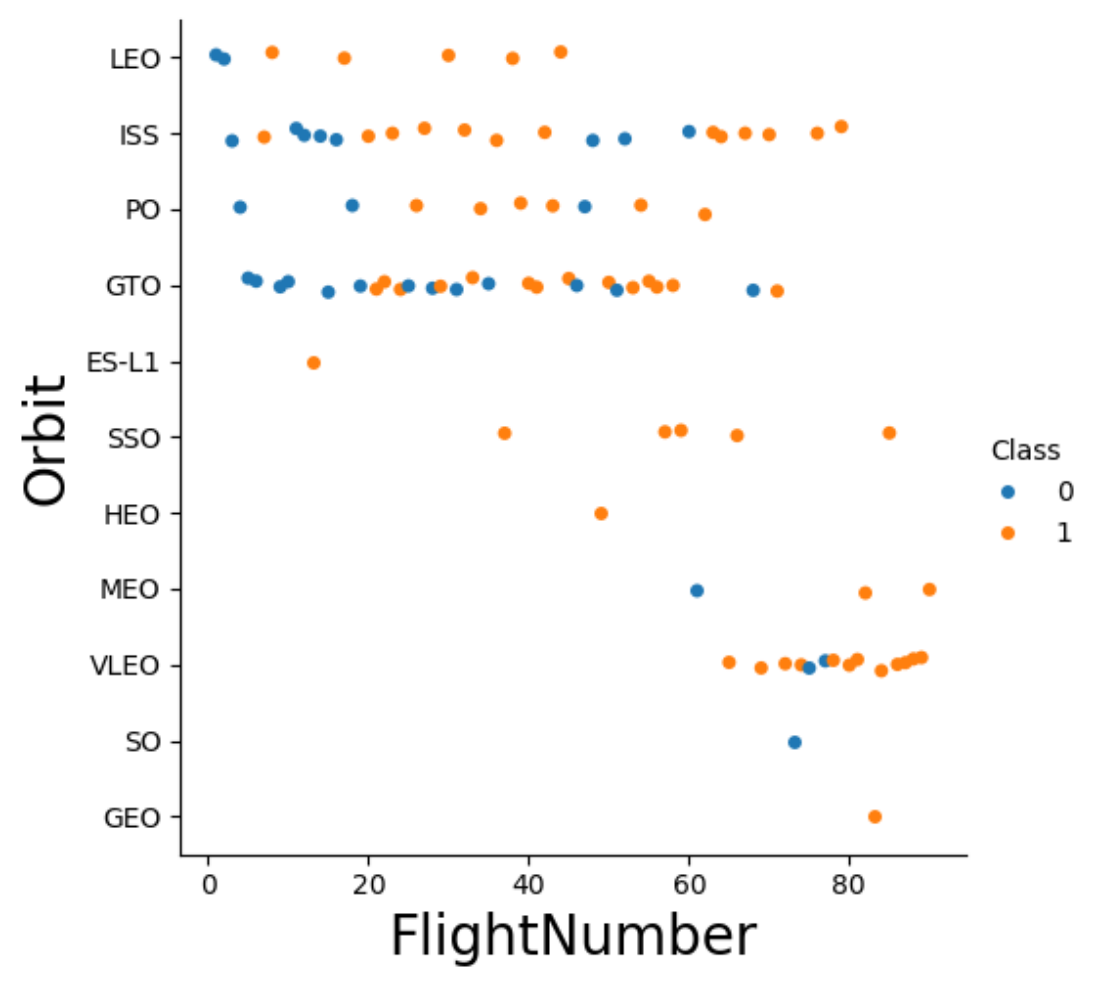
- Show a bar chart for the success rate of each orbit type



- The orbits having the highest success rate are ES-L1, GEO, HEO and SSO with a success rate of 100%, while the lowest success rate is from the SO orbit with a 0% success rate.
- GTO, ISS, LEO, MEO, PO and VLEO all have around 50% success rate.
- However, the data is skewed and cannot be used to acquire concrete information regarding the orbit with the best chances of success as all the orbits with 100% success rates have very low number of flights.

Flight Number vs. Orbit Type

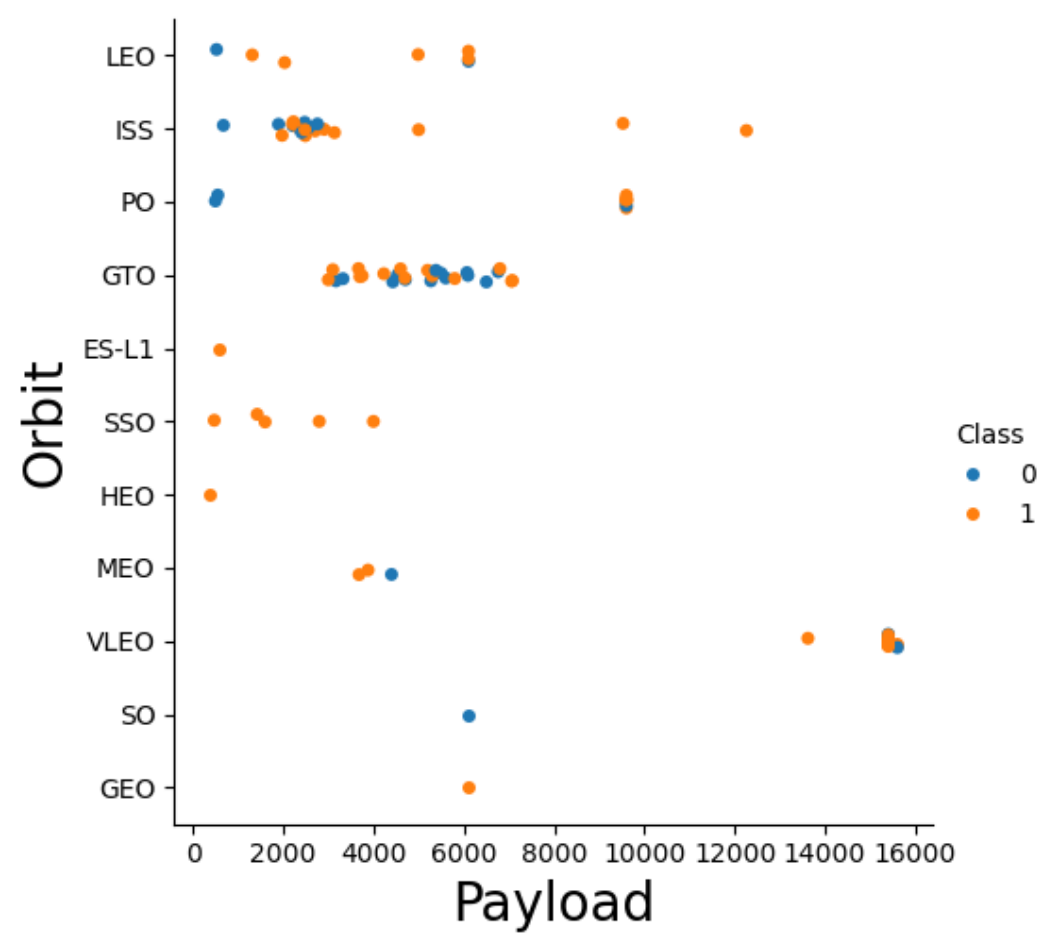
- Show a scatter point of Flight number vs. Orbit type



- In the LEO orbit, success seems to be related to the number of flights as the success rates are much higher after initial failures.
- Conversely, in the GTO orbit, there appears to be no relationship between flight number and success as the failures can also be seen after increasing flight numbers.
- Multiple orbits have very low flight numbers making the relationship between flight numbers wrt orbit in terms of success rates inconclusive for the orbits.

Payload vs. Orbit Type

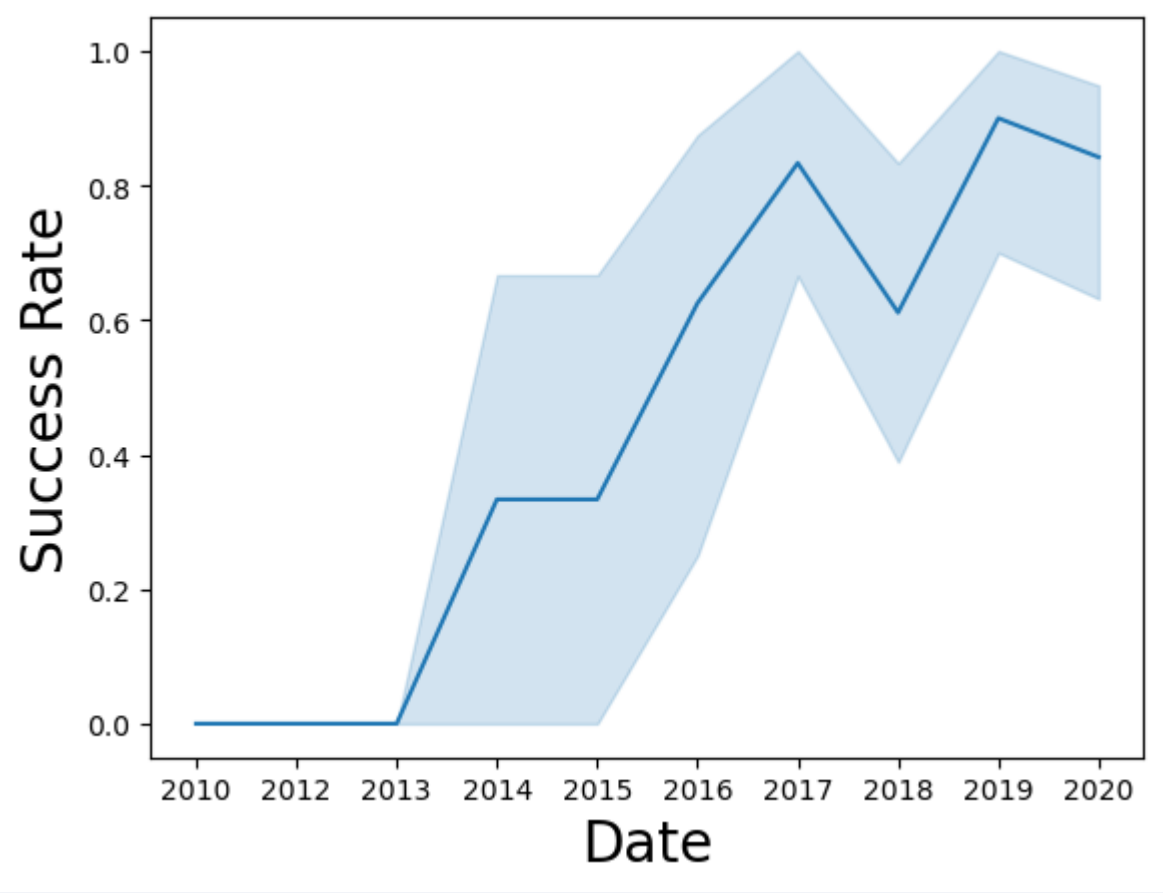
- Show a scatter point of payload vs. orbit type



- With heavy payloads the rate of successful landings are higher for Polar, LEO and ISS.
- However, for GTO, it's difficult to distinguish between successful and unsuccessful landings as both outcomes are present. This case can also be seen for ISS with lower payloads.
- ES-L1, HEO, GEO and SO contain very few launches to make out any concrete pattern.
- SSO has successful landings for lower payloads, while VLEO has a combination of successful and failed landings for higher payloads. Thus, its not possible to see a common trend amongst all the sites.

Launch Success Yearly Trend

- Show a line chart of yearly average success rate



- The success rate since 2013 kept increasing till 2020.
- There was a slight dip in the success rate observed during 2017-2018.
- The success rate was the highest in the year 2019.
- The biggest increase in success rate was observed during the range of years 2015-2017.

All Launch Site Names

- To find the names of the unique launch sites from the SpaceX data:

```
In [13]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
Out[13]:
```

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

- The *DISTINCT* statement allows us to search for the unique values present within the launch site column.

Launch Site Names Begin with 'CCA'

- To find 5 records where launch sites begin with `CCA`:

```
In [16]:
%%sql
SELECT LAUNCH_SITE
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;

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

Out[16]:

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

- By using LIMIT 5, the first 5 records with launch sites having ‘CCA’ were found.

Total Payload Mass

- To calculate the total payload carried by boosters from NASA

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

 * sqlite:///my_data1.db
Done.

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

- By using the *SUM* function and the *WHERE* clause the sum of the payload mass of boosters from NASA can be calculated.

Average Payload Mass by F9 v1.1

- To calculate the average payload mass carried by booster version F9 v1.1:

```
In [18]: %%sql
         SELECT AVG(PAYLOAD_MASS__KG_)
         FROM SPACEXTBL
         WHERE Booster_Version LIKE 'F9 v1.0%';

          * sqlite:///my_data1.db
         Done.
Out[18]: AVG(PAYLOAD_MASS__KG_)
         ------------------------
                           340.4
```

- By using the *AVG* function and *WHERE-LIKE* clauses the average payload mass of a specific version of a booster (F9 v1.1) can be calculated.

First Successful Ground Landing Date

- To find the dates of the first successful landing outcome on ground pad:

```
%sql SELECT min(DATE) FROM SPACEXTBL WHERE LANDING__OUTCOME='Success (ground pad)'
 * sqlite:///my_data1.db
Done.
        1
2015-12-22
```

- By using the *MIN* function and the *WHERE* clause the first successful landing for ground pad can be found.

Successful Drone Ship Landing with Payload between 4000 and 6000

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

In [21]:
```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

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

Out[21]:

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

- By using the *WHERE* clause and defining range of the payload mass, the names of boosters with successful landings having the defined range of payload mass were found.

Total Number of Successful and Failure Mission Outcomes

- To calculate the total number of successful and failure mission outcomes:

In [22]:
```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

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

Out[22]:

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

- By using *COUNT* function and *GROUP BY* statement the total number of mission outcomes can be found for the particular group (successful/failure).

Boosters Carried Maximum Payload

- To list the names of the boosters which have carried the maximum payload mass:

In [23]:
```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

Out[23]:

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

- By using *DISTINCT* statement and a subquery using *MAX* function, the names of boosters carrying the max payload mass from the data can be found.

2015 Launch Records

- To list the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015:

```
In [26]:
%%sql
SELECT LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)'
    AND substr(Date,0,5)='2015';

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

Out[26]:

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

- By using *WHERE* statement and a *substring* to mark the year, the list of failed landing outcomes along with their booster versions and launch sites can be found.

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

- To 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:

In [27]:
```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

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

Out[27]:

Landing_Outcome	TOTAL_NUMBER
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

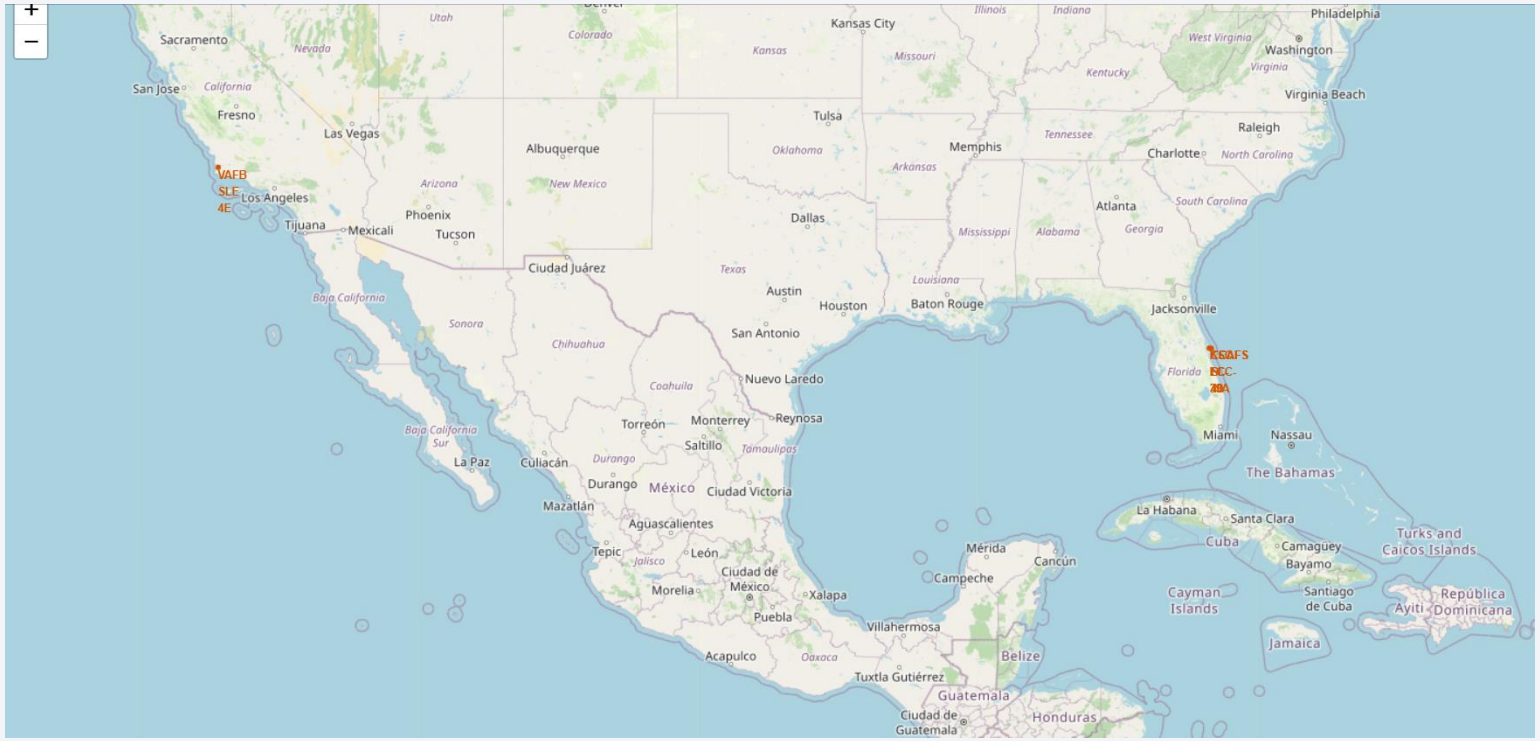
- By using the *COUNT* function and *WHERE* and *GROUP BY* clauses with *DESC* statement, the total number of landing outcomes per type can be found out in the decreasing order of numbers.

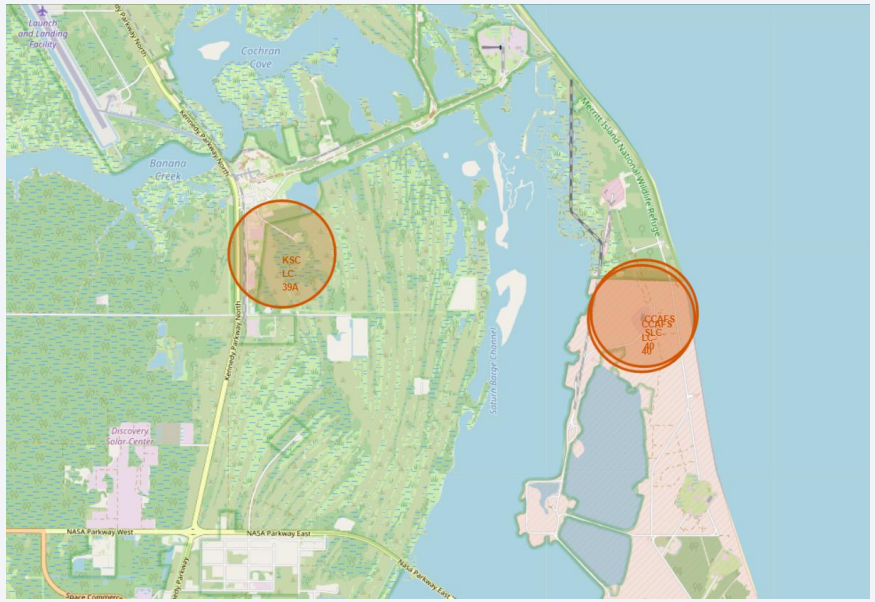
Section 3

Launch Sites Proximities Analysis

Launch Site Locations

- The following map depicts the locations of all the featured launch sites within the data:

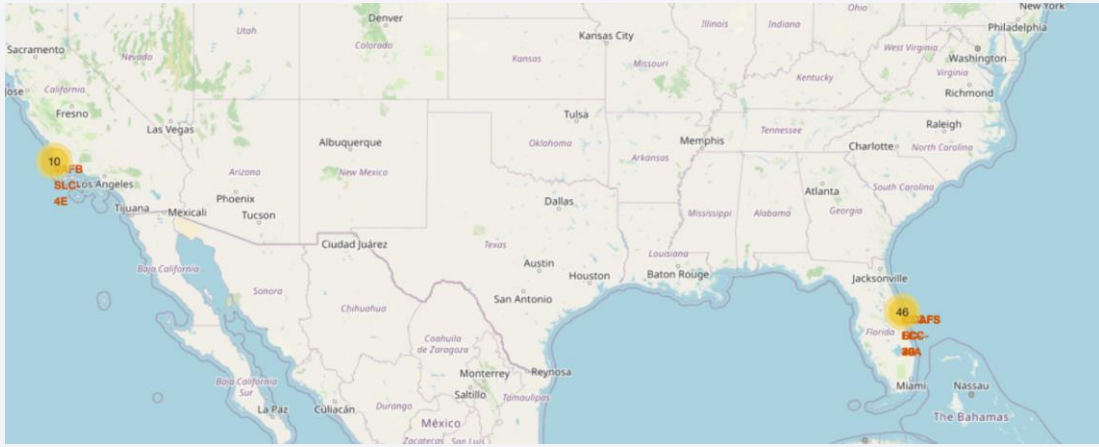




- All the launch sites are located in restricted spaces and are in close proximity to coastal areas.

Launch Outcomes for the Sites

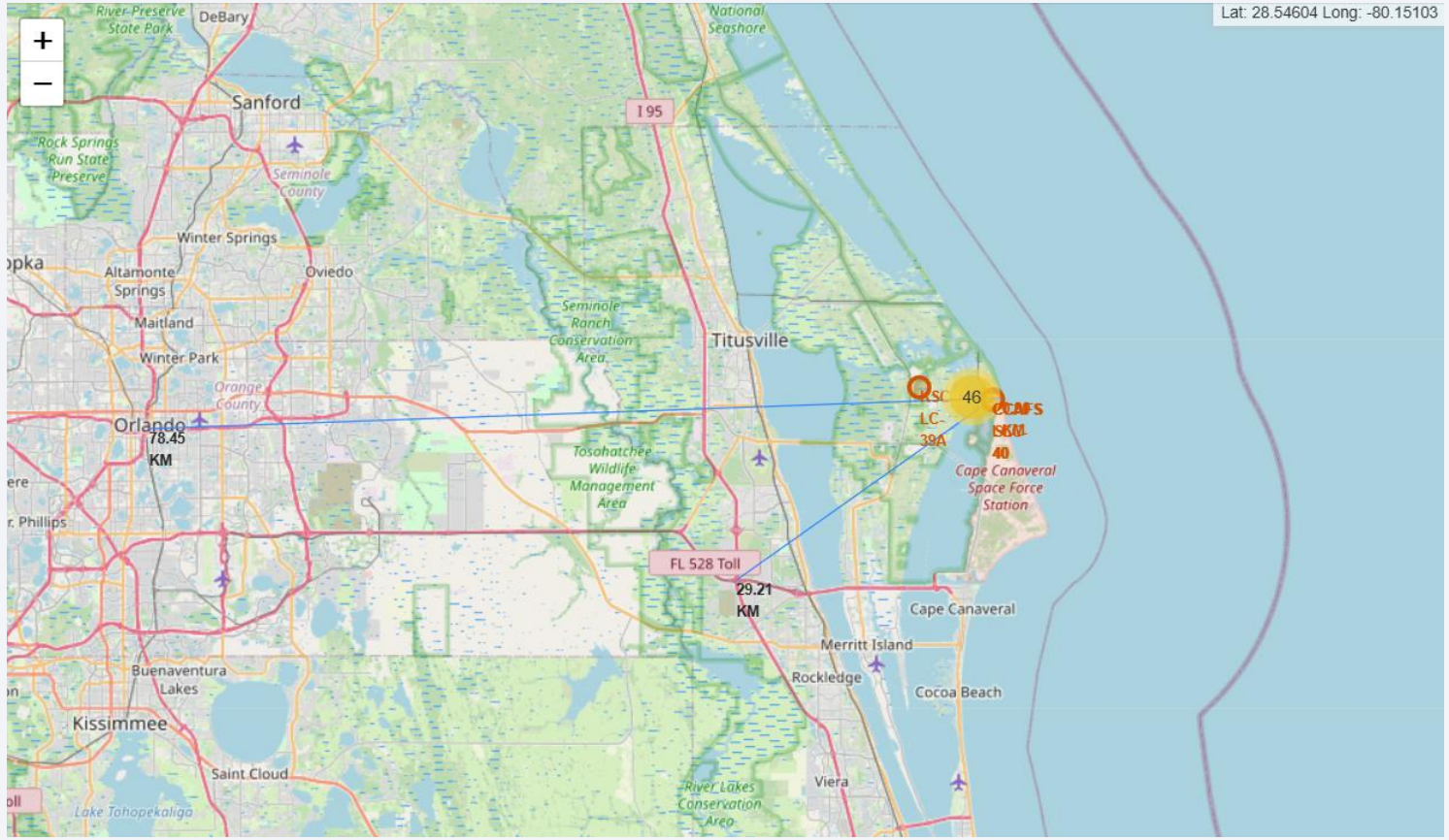
- The following map shows the landing outcomes of the launch sites:



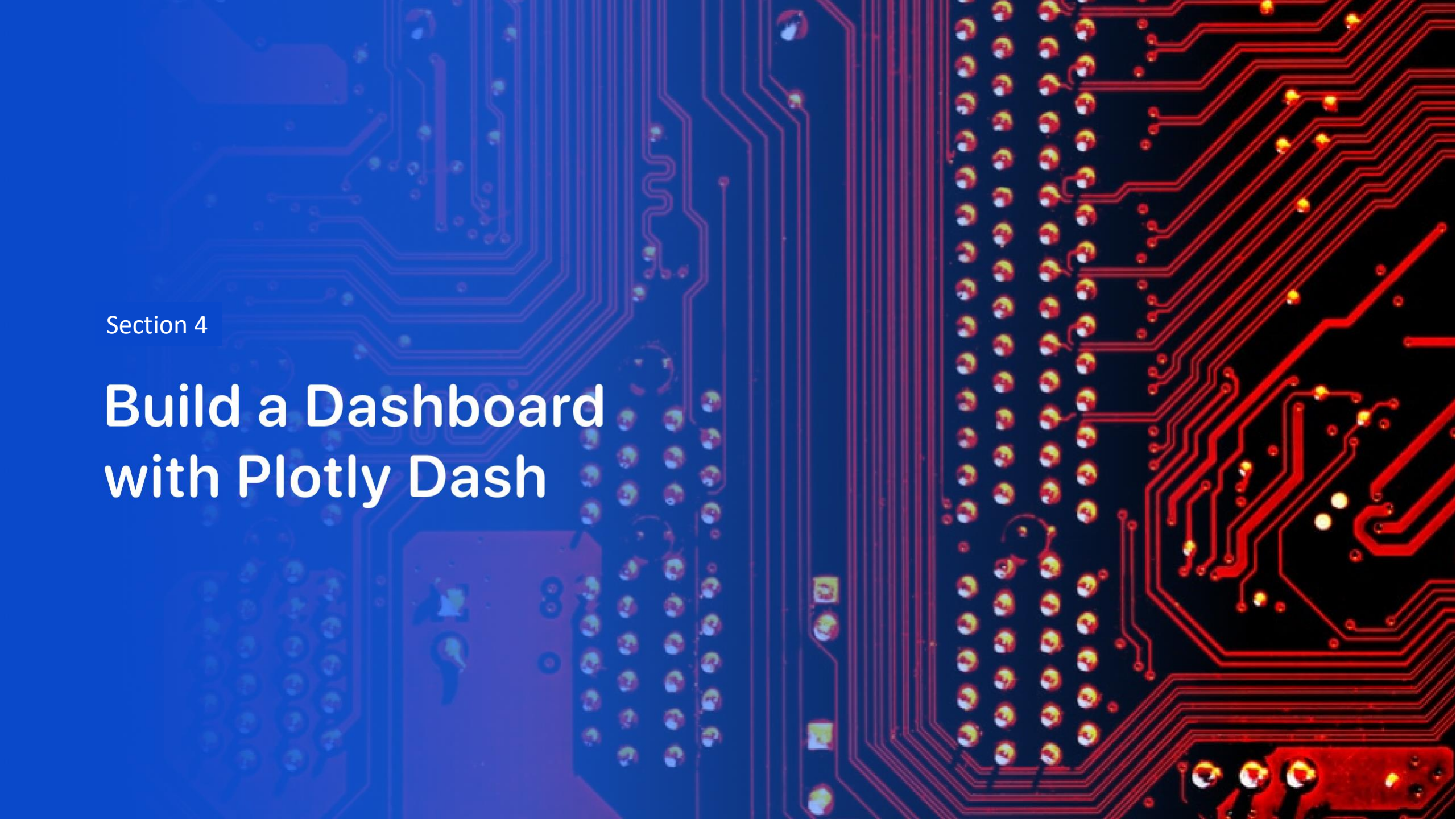
- The Red Markers denote failed outcomes whereas the Green Markers are for successful ones.

Launch Site and its Proximities

- The following map shows launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed:

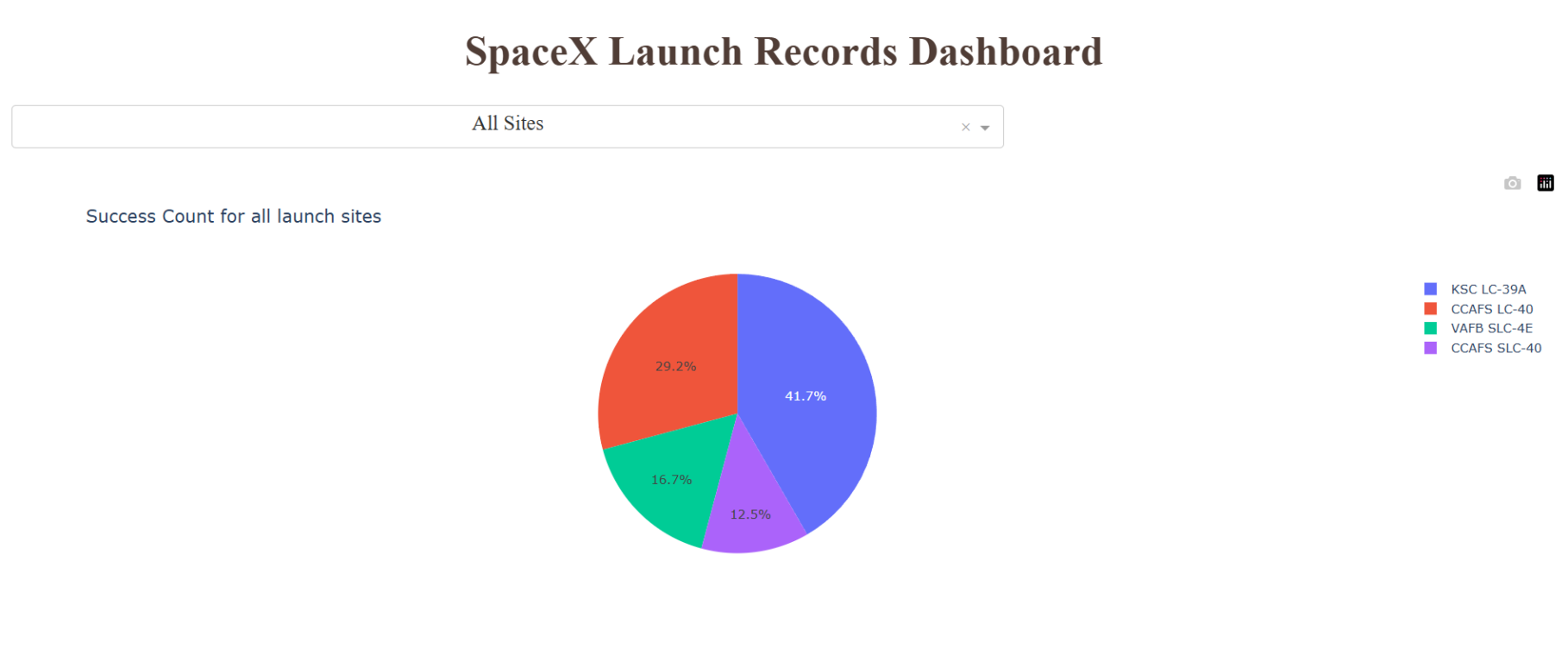


- The launch sites are located in close proximity to the coastline and are away from inhabited regions for safety purposes.
- Although away from the main city these launch sites have decent connections with respect to railway lines and roads.


Section 4
Build a Dashboard with Plotly Dash

Successful Launches per Launch Site

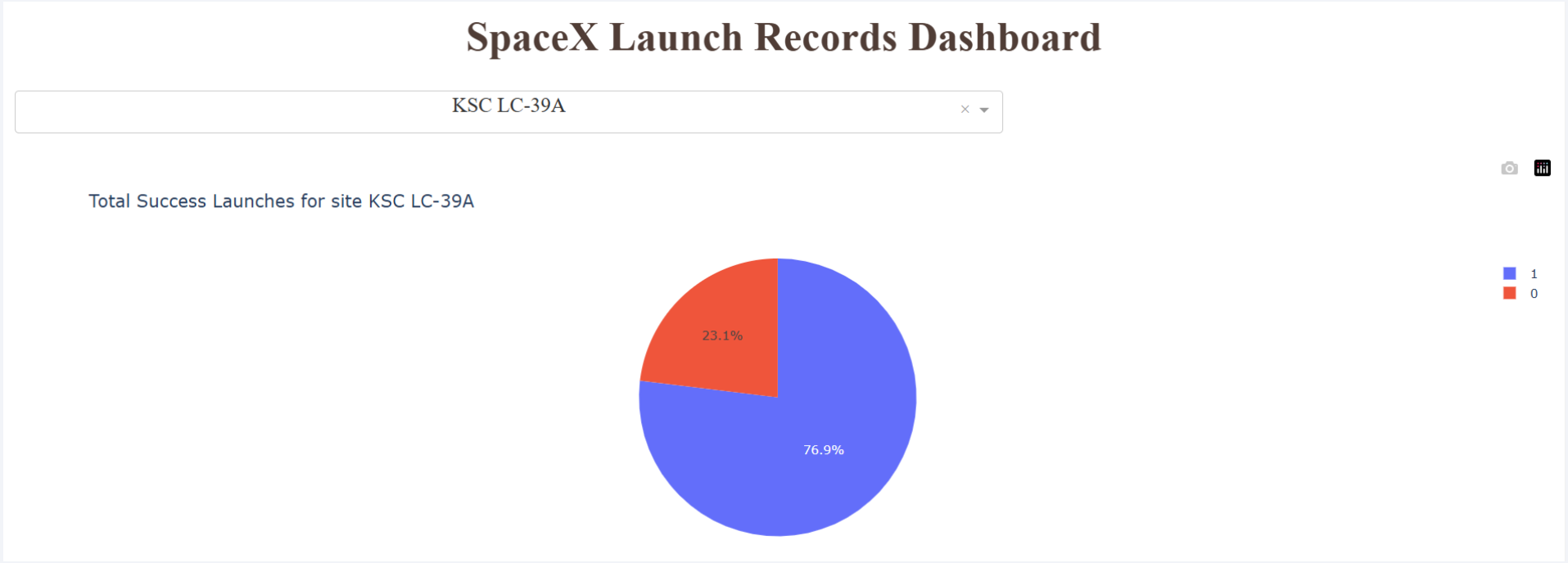
- The following pie chart shows the percentage of successful launches per launch site:



- KSC LC-39A has the highest number of successful launches whereas CCAFS SLC-40 has the least number of successful launches.

KSC LC 39-A

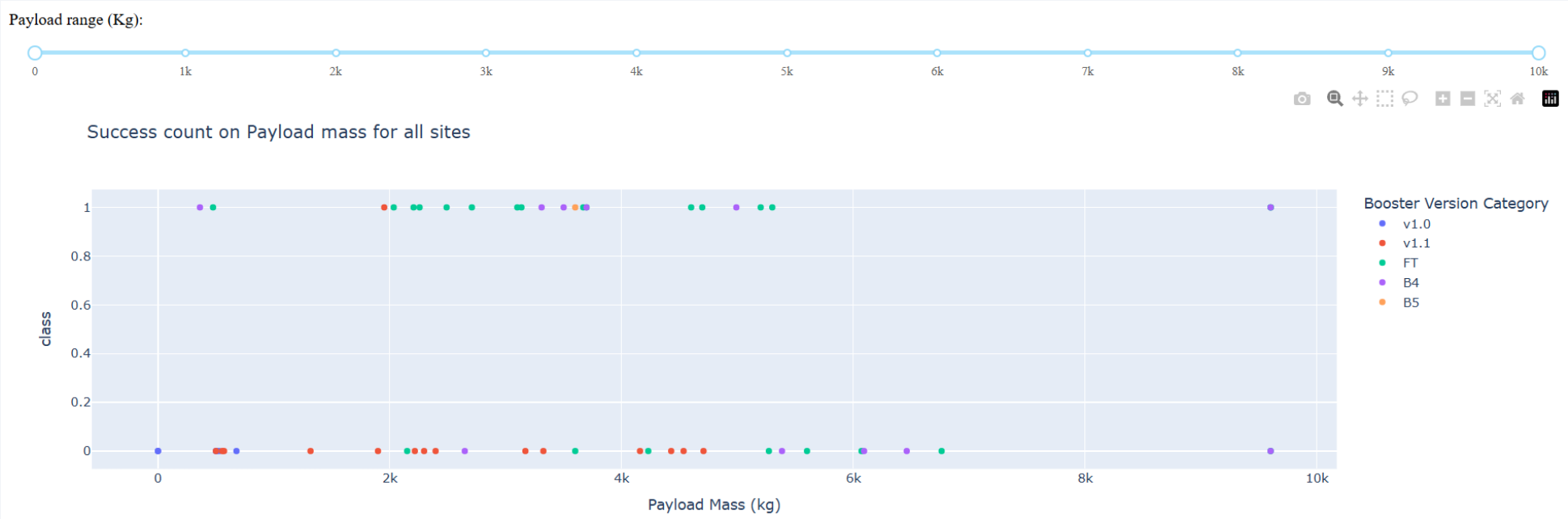
- KSC LC 39-A has the highest launch success rates amongst all the launch sites. The following pie chart shows the ratio of the launch site's success rates:



- KSC LC 39-A has a success ratio of 76.9% successful launches.

Payload vs. Launch Outcomes

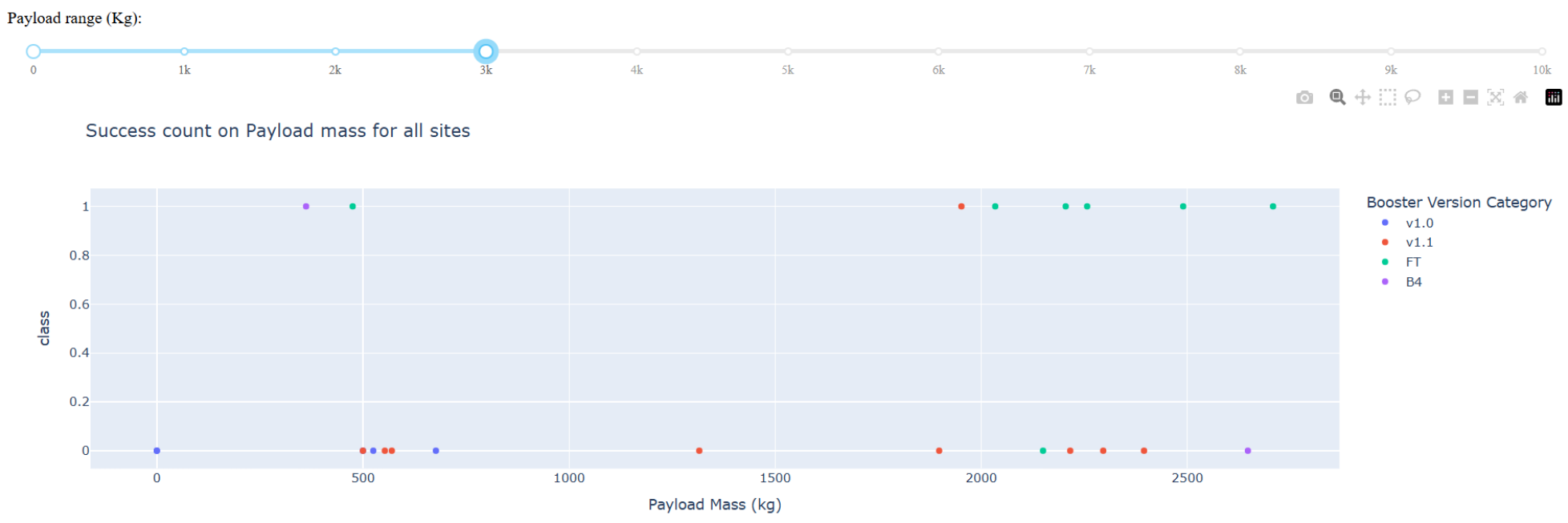
- The following scatterplots show the launch outcomes seen on the launch sites with particular range of payload mass:



- On the entire scale (0-10k), the Booster Version FT has the best success ratio as compared to v1.1, which has the lowest success ratio.

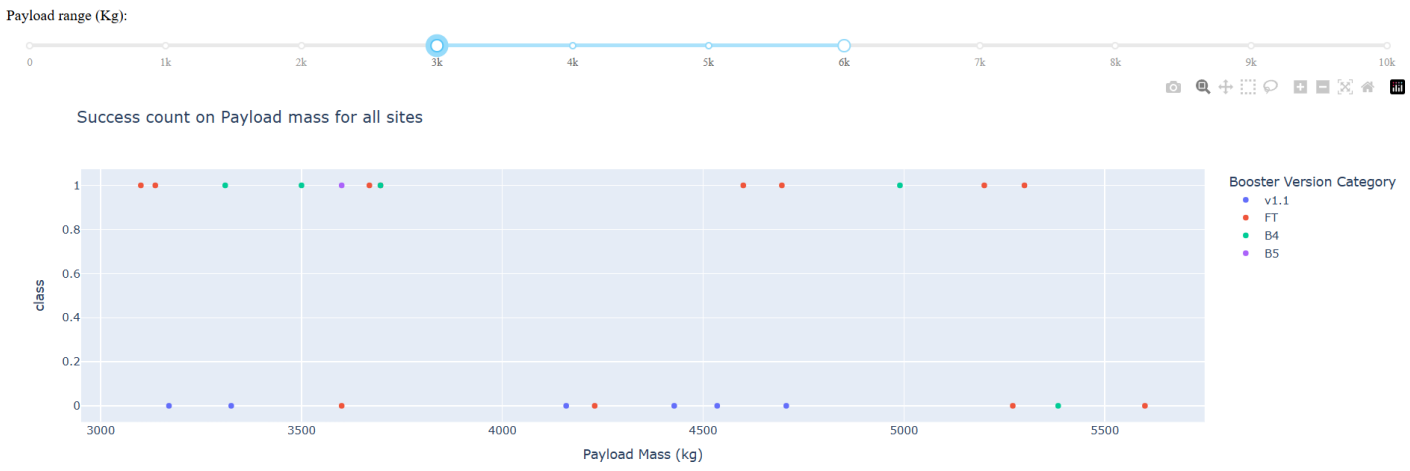
Payload vs. Launch Outcomes

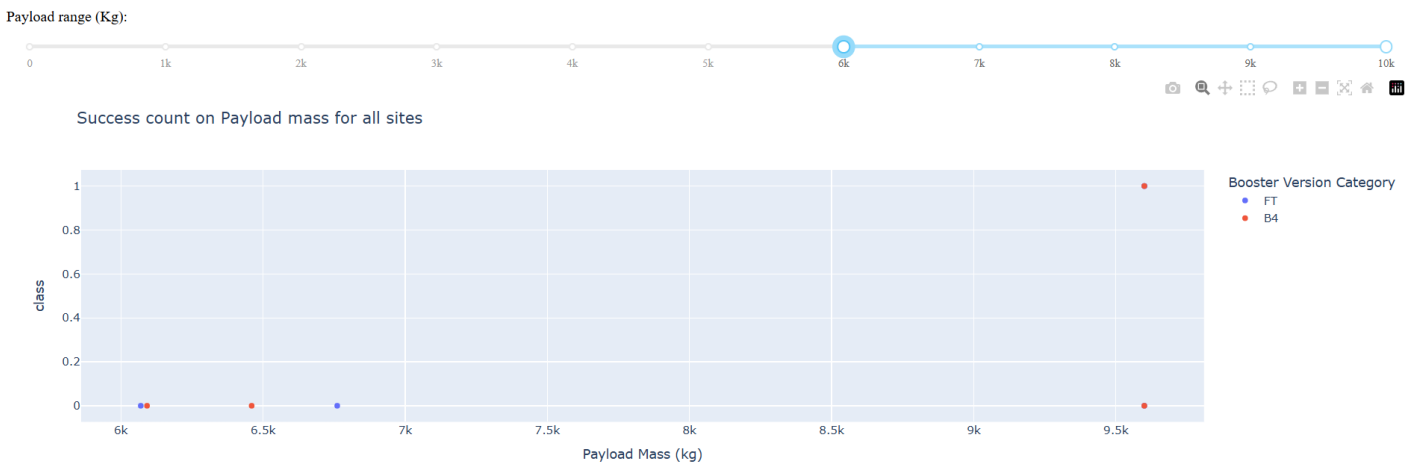
- After dividing the scale of payload mass into 3 ranges (0-3k, 3k-6k and 6k-10k), the following plots were provided:



- Most of the failures were concentrated in this range (0-3k) of payload mass, especially for booster version v1.1. However, in contrast, booster version FT faired comparatively well.

Payload vs. Launch Outcomes





- In the range (3k-6k) of payload mass, the success ratio is observed to be around 50%. Here booster version FT has comparatively higher failures as compared to the lower payload range. Meanwhile v1.1, has a 100% failure rate in this range.

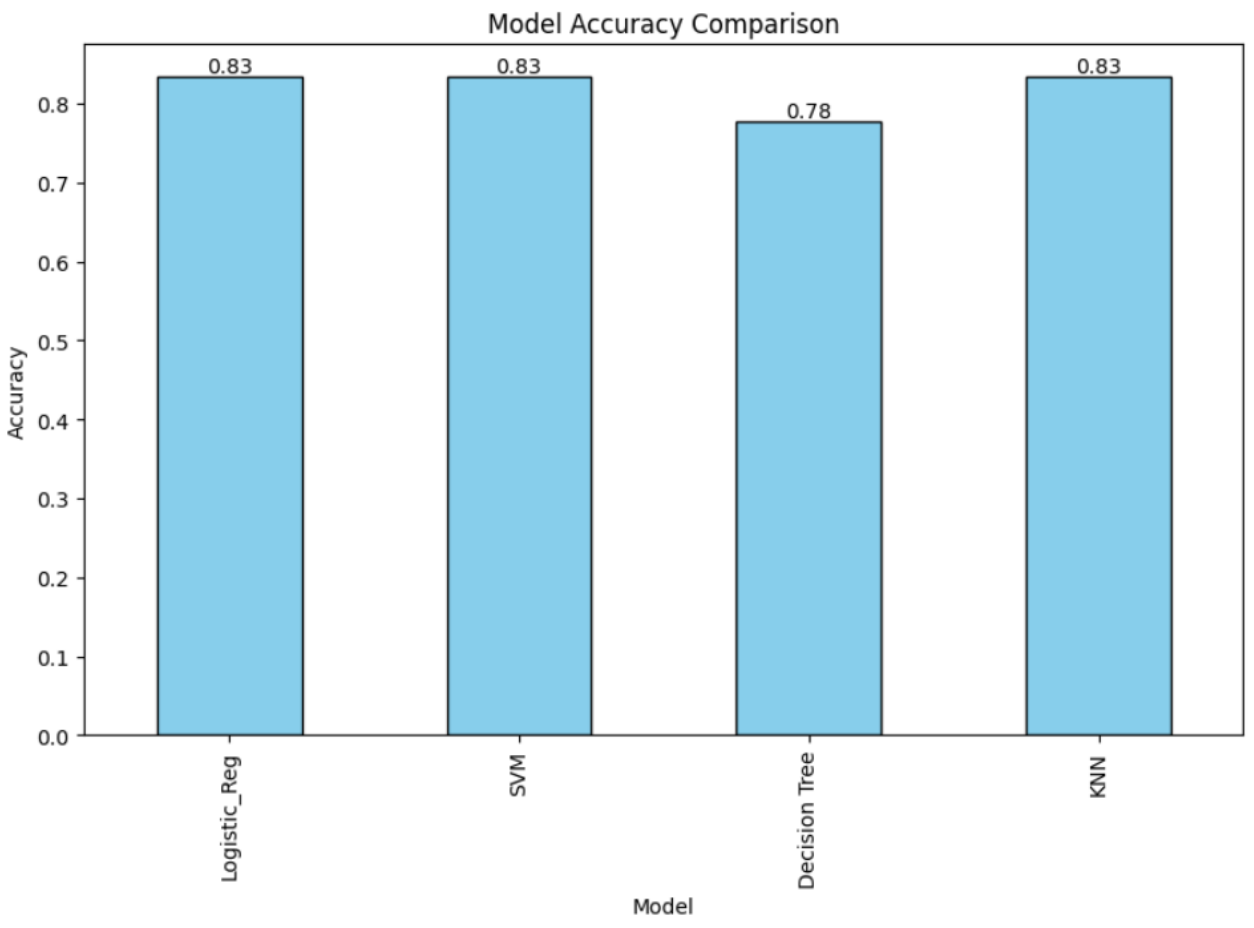
- In the range (6k-10k) of payload mass, the success ratio is extremely low, but it also has the least amount of launches. Only two booster versions are featured with only one of them having a successful launch.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The bar chart visualizes the built model accuracy for all the classification models:

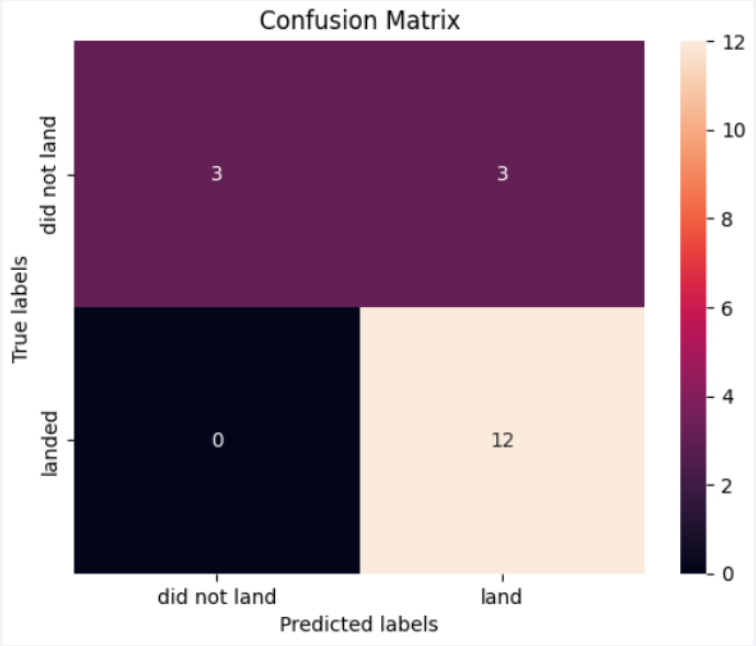


- The accuracy for Logistic Regression, SVM and KNN models was equally highest at ~83% while for Decision Trees it was ~78%.
- This shows that three of the models had consistent performance in predicting the outcomes.
- Table of the accuracies is as follows:

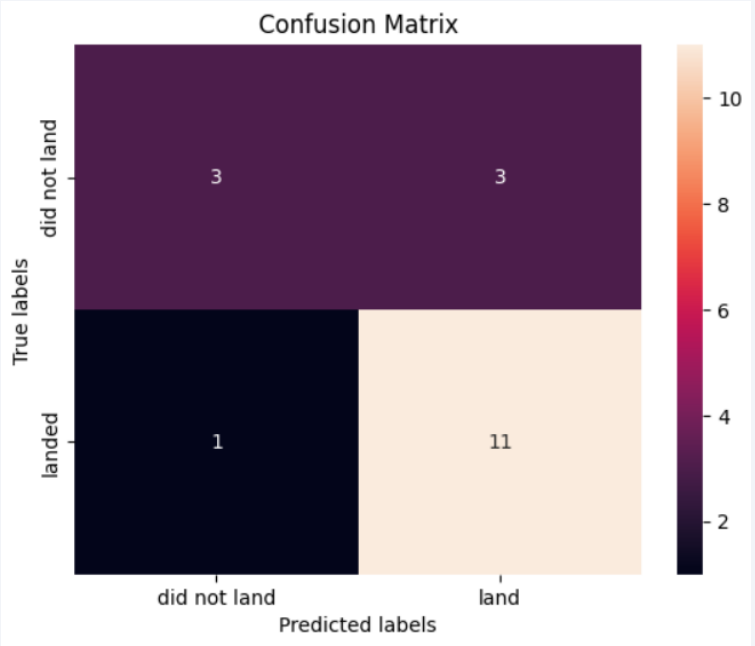
	0
Method	Test Data Accuracy
Logistic_Reg	0.833333
SVM	0.833333
Decision Tree	0.777778
KNN	0.833333

Confusion Matrix

- The Confusion Matrix is the same for Logistic Regression, SVM and KNN models.



LR, SVM, KNN Confusion Matrix



Decision Tree Confusion Matrix

- The Confusion Matrix for Decision Tree shows that it only predicted one outcome wrong by assigning it as ‘did not land’ label in place of ‘land’ label, resulting in its lower accuracy.

Conclusions

- The EDA followed by visualizations and dashboards provided detailed insights into the data helped to understand that the data is skewed and requires more data points for better model accuracies.
- The model accuracies were found to be exactly the same for Logistic Regression, SVM and KNN at ~83%. Higher accuracies can be achieved as part of a future scope.
- The goal of creating a machine learning pipeline to predict launch outcomes was achieved and can be used the first stage launches.

Appendix

- All the images for reference as well the code snippets can be found on the GitHub repository:

Coursera Applied Data Science Capstone

Thank you!