Unlocking the Future of Space Travel:

Predicting Rocket Reusability

Arunraj Kondetharayil Soman
27-09-2023

Outline

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

Executive Summary

The project employs a range of methodologies, encompassing data collection, data wrangling, exploratory data analysis (EDA) with data visualization and SQL, as well as creating interactive maps and dashboards. These approaches enable a comprehensive assessment of launch success factors and predictive model development.

The project primarily focuses on evaluating launch costs and forecasting first-stage reuse success. Notably, success rates increase with higher flight numbers across all launch sites. Heavier payloads at CCAFS SLC 40 correlate with higher success rates. Some orbits, such as ES-LI, GEO, HEO, and SSO, maintain a perfect 100% success rate, while others exhibit varying patterns. Lighter payloads in LEO, ISS, and PO orbits show higher success rates. Over time, annual launch success rates have grown significantly. KSC LC-39A leads in successful launches. The decision tree model achieves the highest accuracy, with a training accuracy of 0.875 and testing accuracy of 0.944. In essence, the project underscores the influence of past missions on current success rates and successfully addresses its core objectives.

Introduction

- In the booming commercial space industry, emerging companies like Space Y aim to compete with industry giants like SpaceX. This project employs data science and machine learning to empower Space Y with competitive pricing strategies based on data from SpaceX's launches.

- The project tackles two primary challenges:

  - Determining Launch Cost
  - Predicting First Stage Reuse

Section 1

Methodology

Methodology

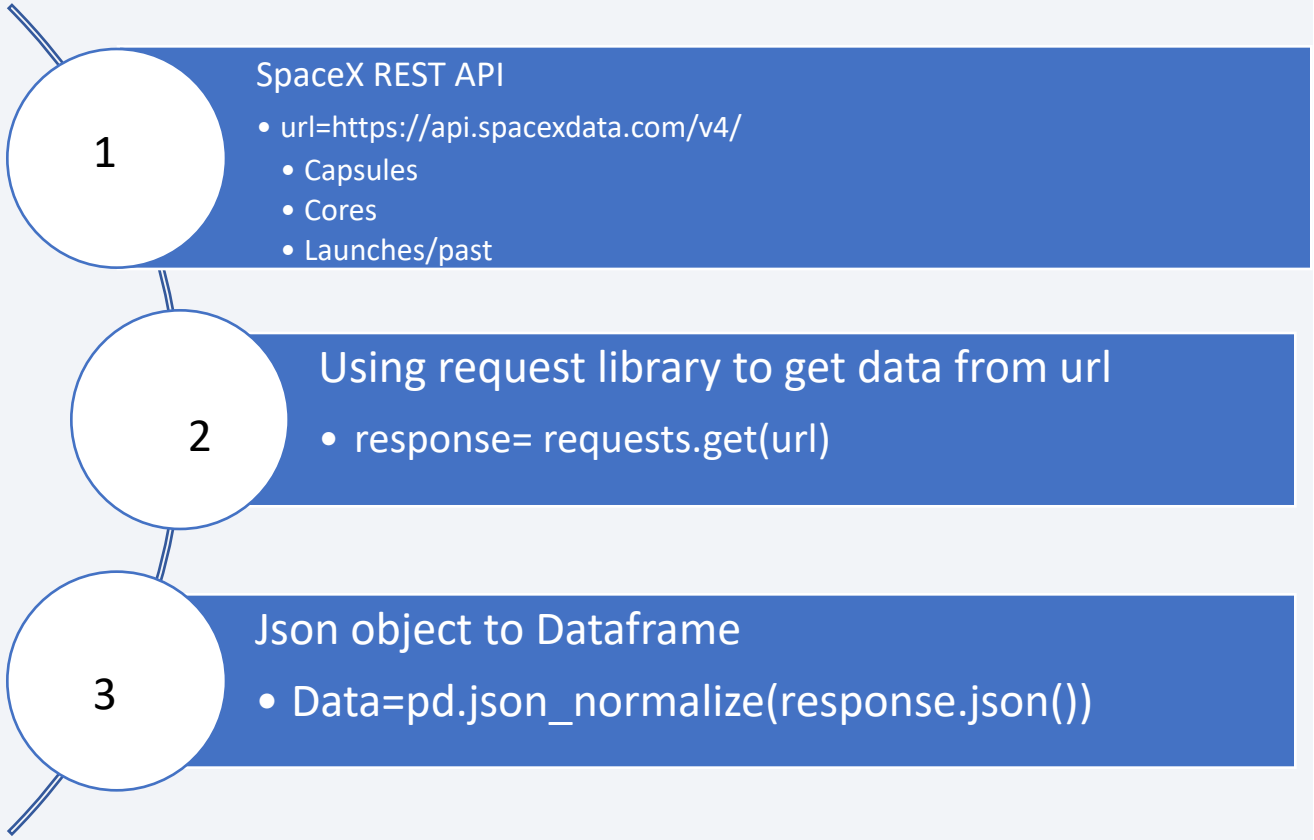
Executive Summary

- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Identify the Features
  - One-hot encoding to categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Build and tune model Using GridSearchCV
  - Evaluate classification models using accuracy score and confusion matrix

Data Collection – SpaceX API

- The data is downloaded using SpaceX REST API using following python packages
  - Requests
  - Pandas

- The GitHub notebook containing the completed SpaceX API calls can be accessed at Data_collection_notebook



Data Collection - Web Scraping

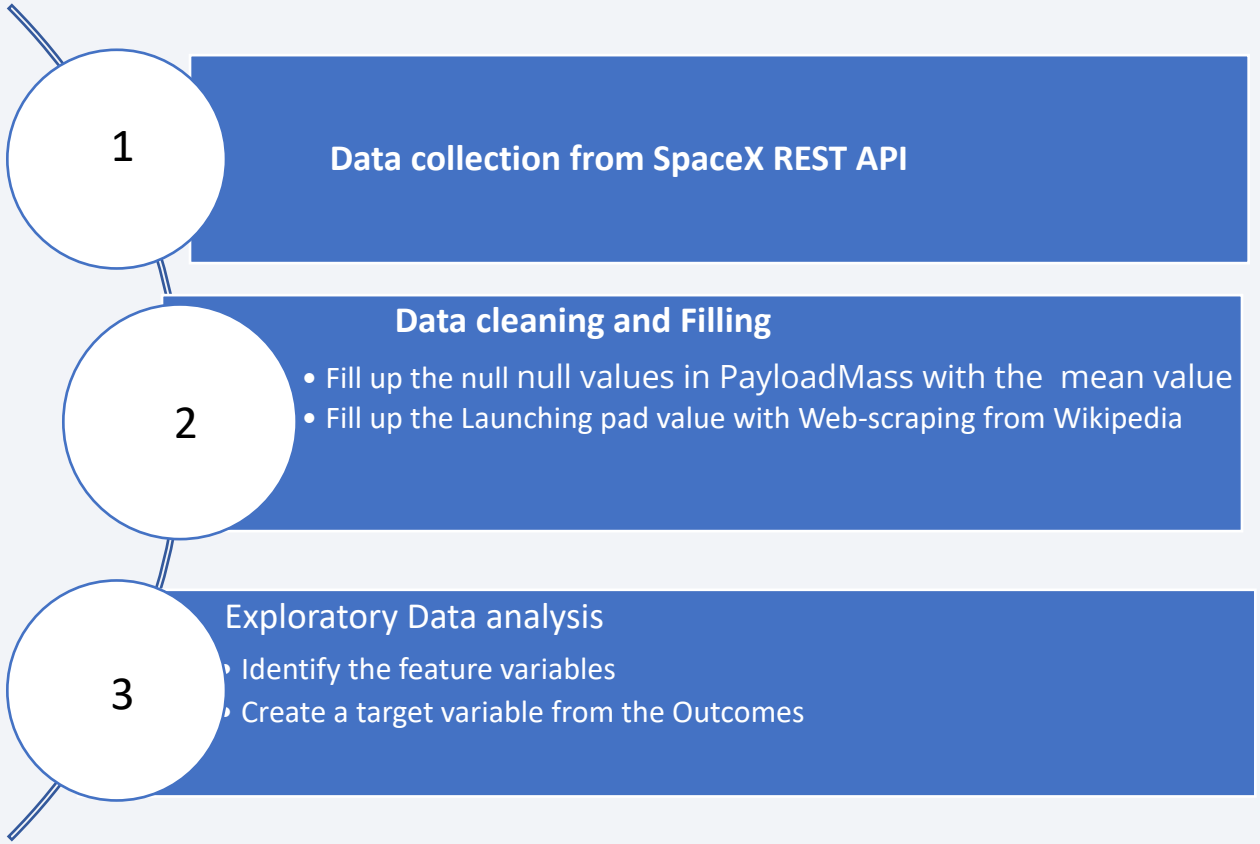
Web scrap Falcon 9 launch records with BeautifulSoup python package

- Extract a Falcon 9 launch records HTML table from Wikipedia
- Parse the table and convert it into a Pandas data frame

The GitHub notebook containing the completed Web Scraping Notebook can be accessed at [Web_Scraping_notebook]

Data Wrangling

- The main objectives include
  - Collect the data from SpaceX REST API
  - Process the data collected
  - Transform to presentable format
  - Exploratory Data Analysis
  - Determine Training Labels
  - Use one-hot encoder to transform categorical variables
- Add The GitHub notebook containing the completed Data Wrangling notebook can be accessed at Data_Wrangling_notebook



EDA with Data Visualization

Scatter plot

1. Visualize relationship between flight number and launch site
2. Visualize relationship between Payload and launch site
3. Visualize relationship between flight number and Orbit type
4. Visualize relationship between Payload and Orbit type

Bar chart

1. Visualize relationship between flight number and launch site

Line chart

1. Visualize the Launch Success yearly trend

- The GitHub link to the EDA with data visualization notebook

EDA with SQL

1. Using DISTINCT command to display unique launch sites
2. Using LIKE command to find the string 'CCA' and limit command to restrict the number of display records
3. Using Where command to select a customer and aggregate function to sum the payload for the customer
4. Using Where command to select a Booster version and aggregate function to average the payload mass
5. Use Like command to get the success from landing and order by date to get the first success
6. Use of and command in where clause to satisfy two conditions. Use of between commands to select the payload range
7. Use of Group By command to get the mission outcomes
8. Use of subquery in the where clause to get the max payload mass and corresponding booster
9. Use of Substr command to get the month and year and between command to filter the date and like command to identify the failure
10. Use of Group by and Order by command to display the landing outcome for a certain period

The GitHub notebook containing the completed EDA with SQL Notebook can be accessed at SQL_notebook

Build an Interactive Map with Folium

- Added added following marker objects to folium map
  - Markers
  - Circles
  - Lines
  - Mouse pointer
  - Added color labels markers to identify success or failure at a launch site
- Calculate the distances between a launch site to its proximities.
  - nearby railways, highways, coastlines and cities
  - Railway, highway and coastlines are nearby but the launch sites are usually away from cities
- The interactive map can be found at the GitHub URL Folium_notebook

Build a Dashboard with Plotly Dash

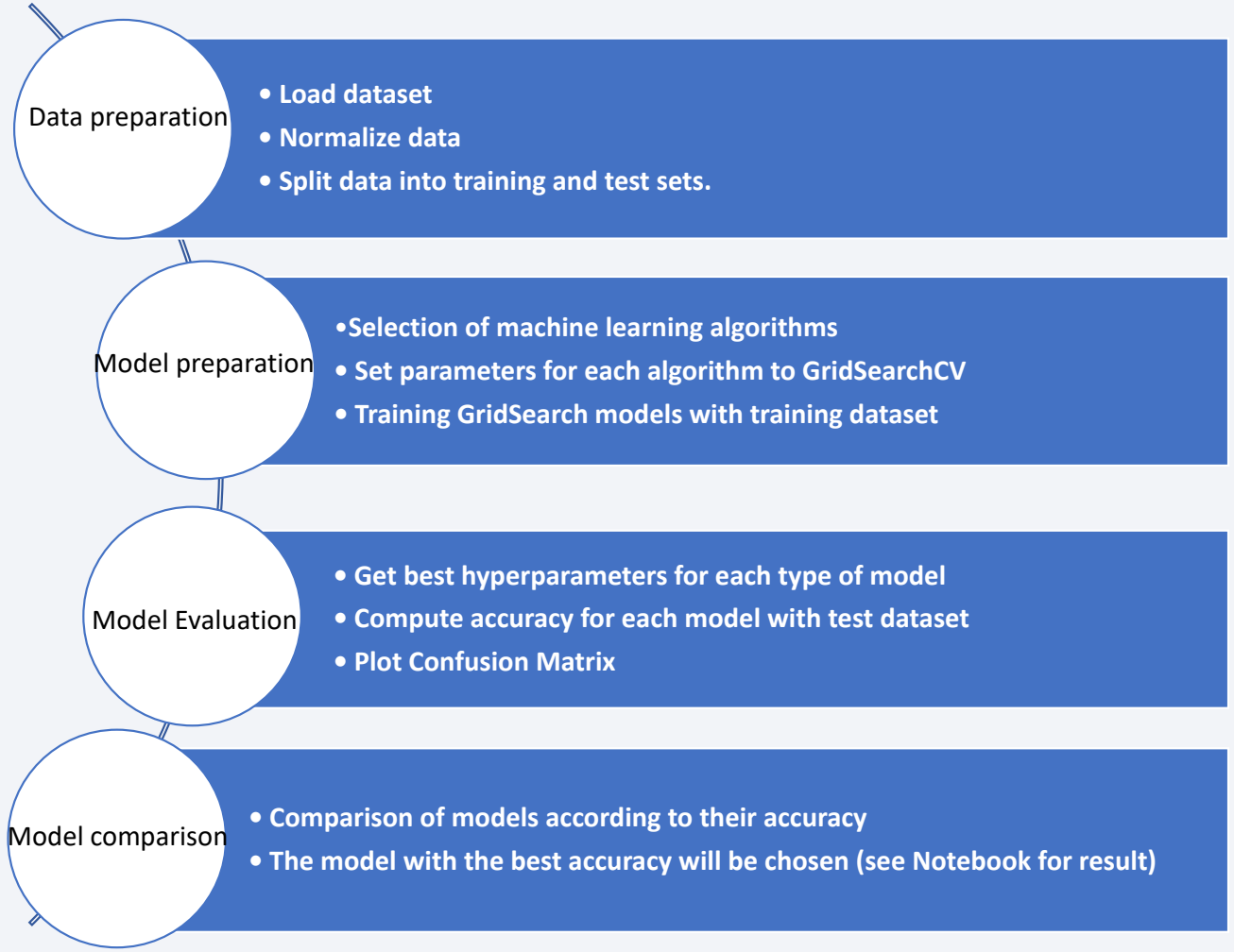
Built an interactive dashboard with Plotly dash

- Added a dropdown menu to select a particular launch site
- A Pie chart was added to identify success rate at each site
- Scatter plots were included to identify correlation between payload vs success rate. And howe does it vary for different version of boosters
- Add a slider to select payload range

The GitHub link to the dashboard is Plotly Dash lab

Predictive Analysis (Classification)

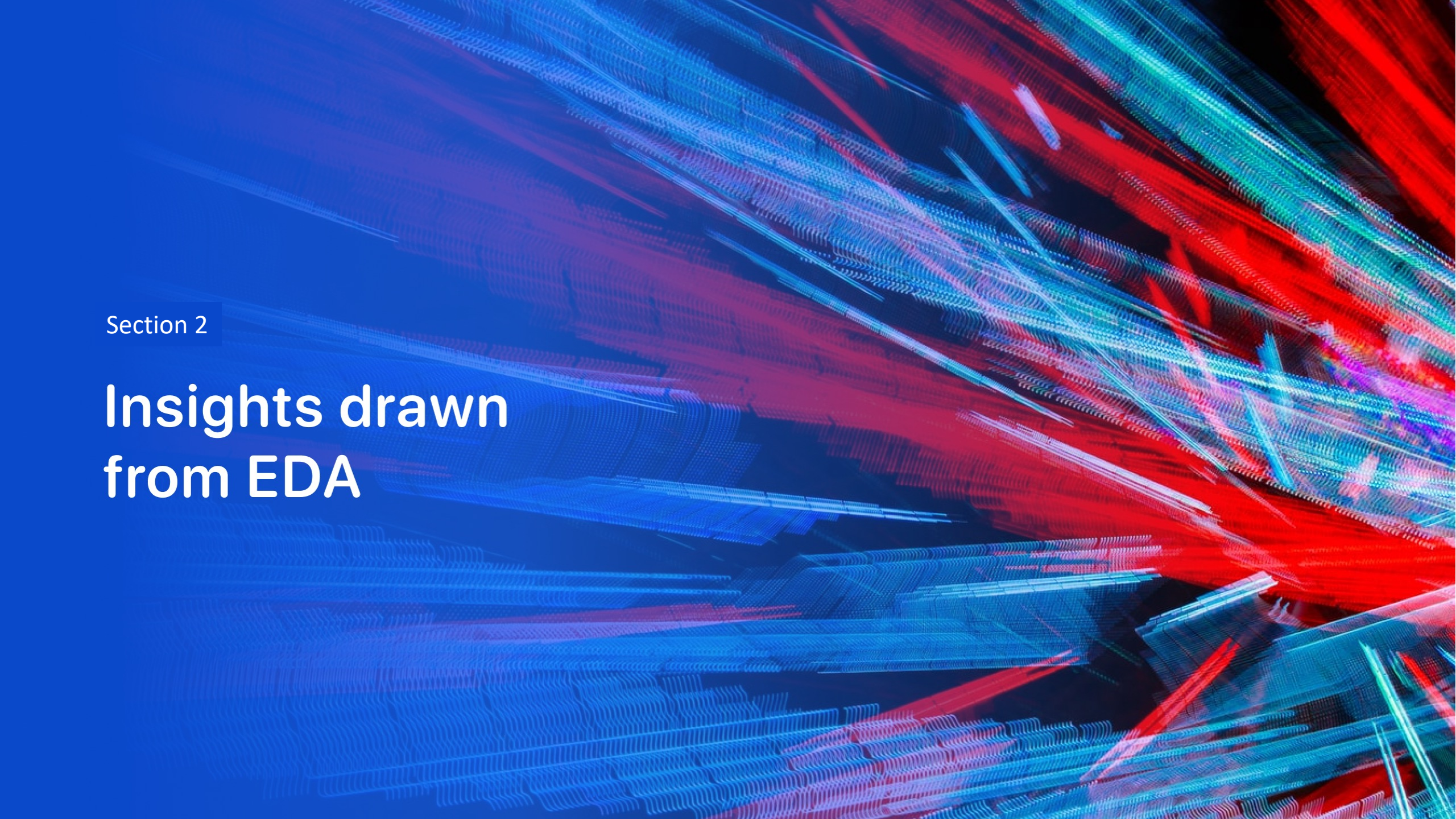
- Flow chart showing the development of a machine learning model



The GitHub link for the predictive analysis lab

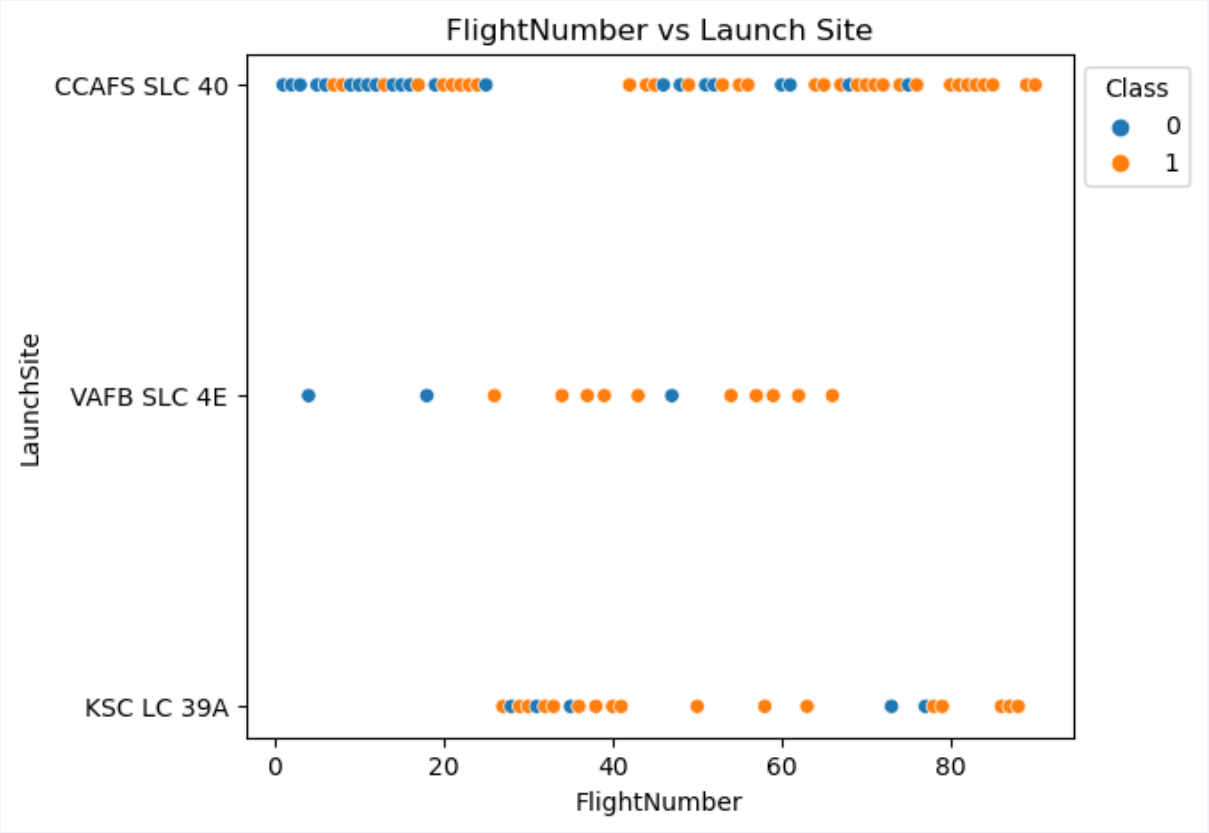
Results

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


Section 2
Insights drawn from EDA

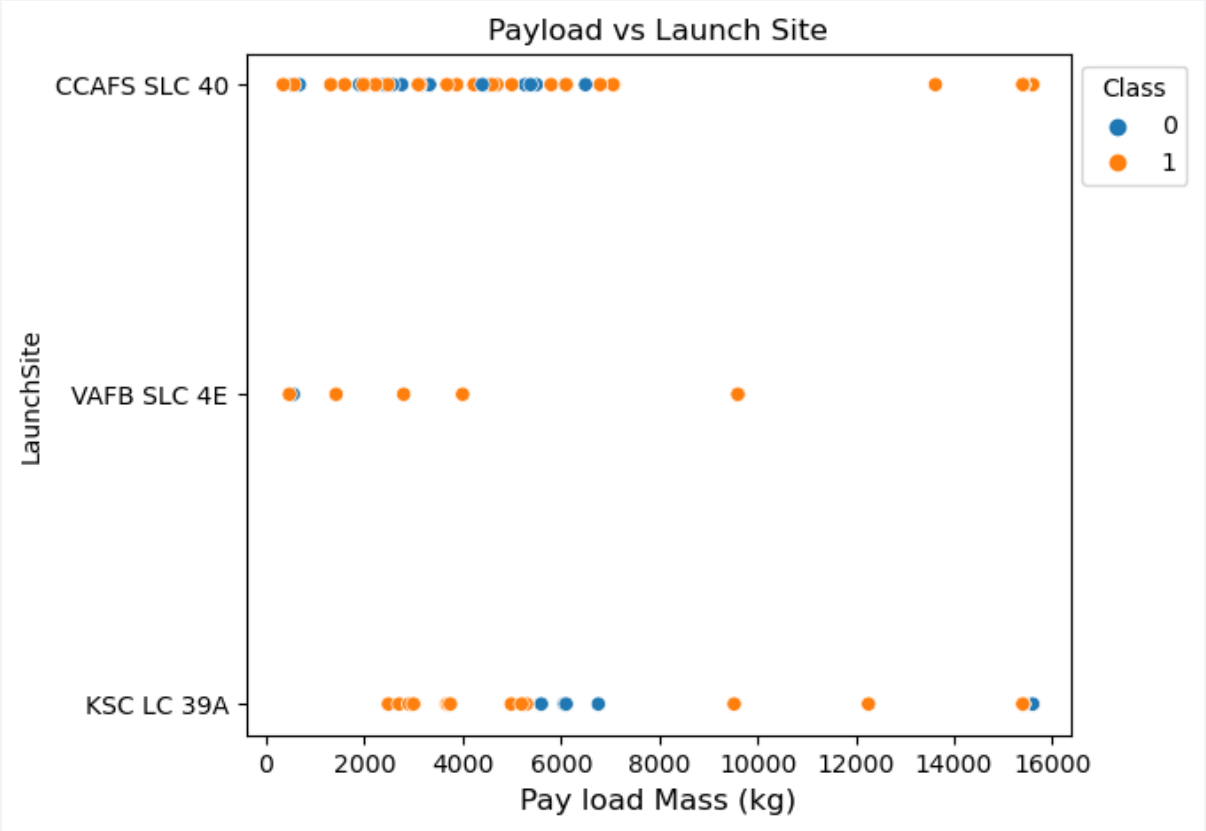
Flight Number vs. Launch Site

- The success rate increases as the Fight number increases at all the sites



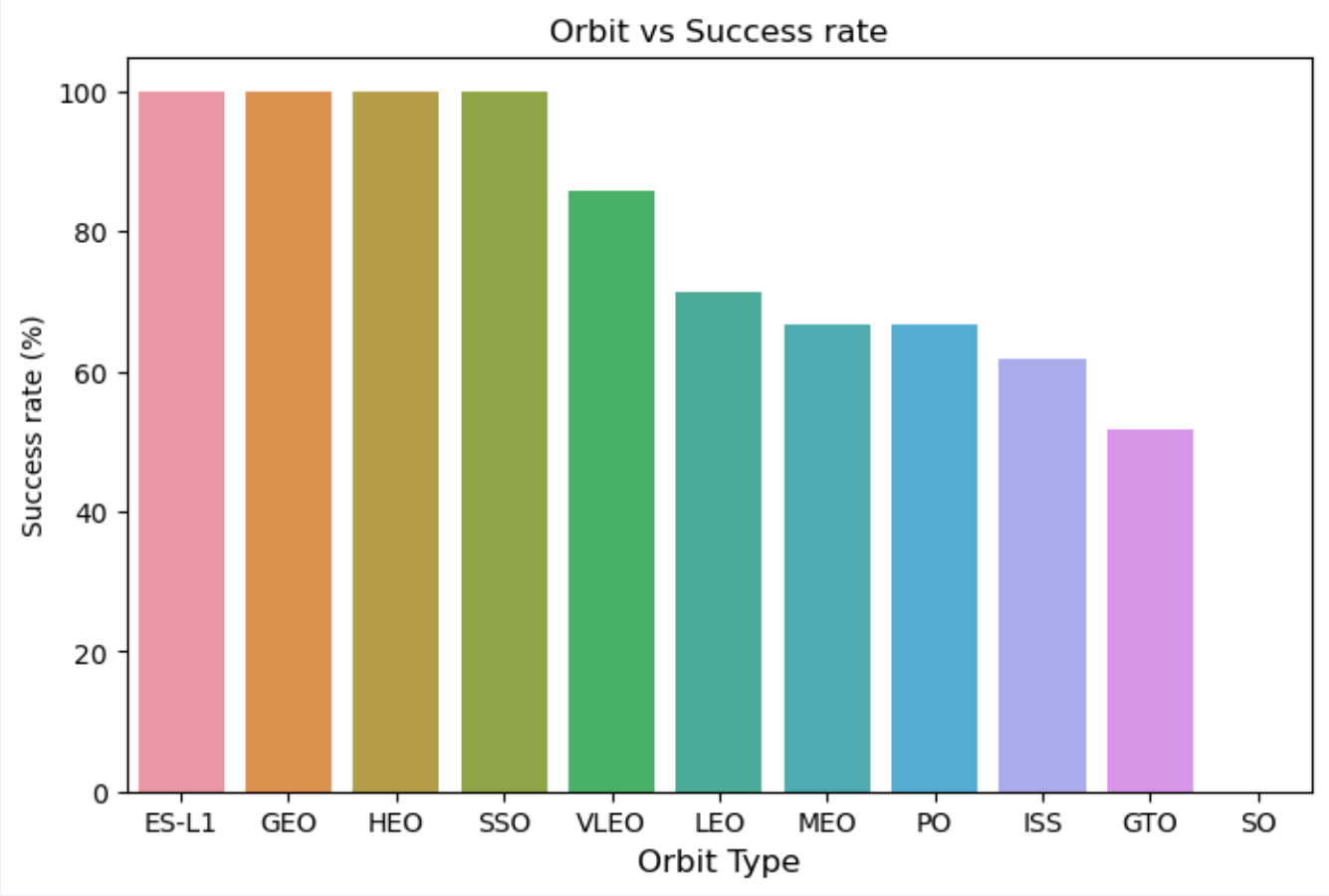
Payload vs. Launch Site

- Success rates are higher for heavier payloads at launch site CCAFS SLC 40.



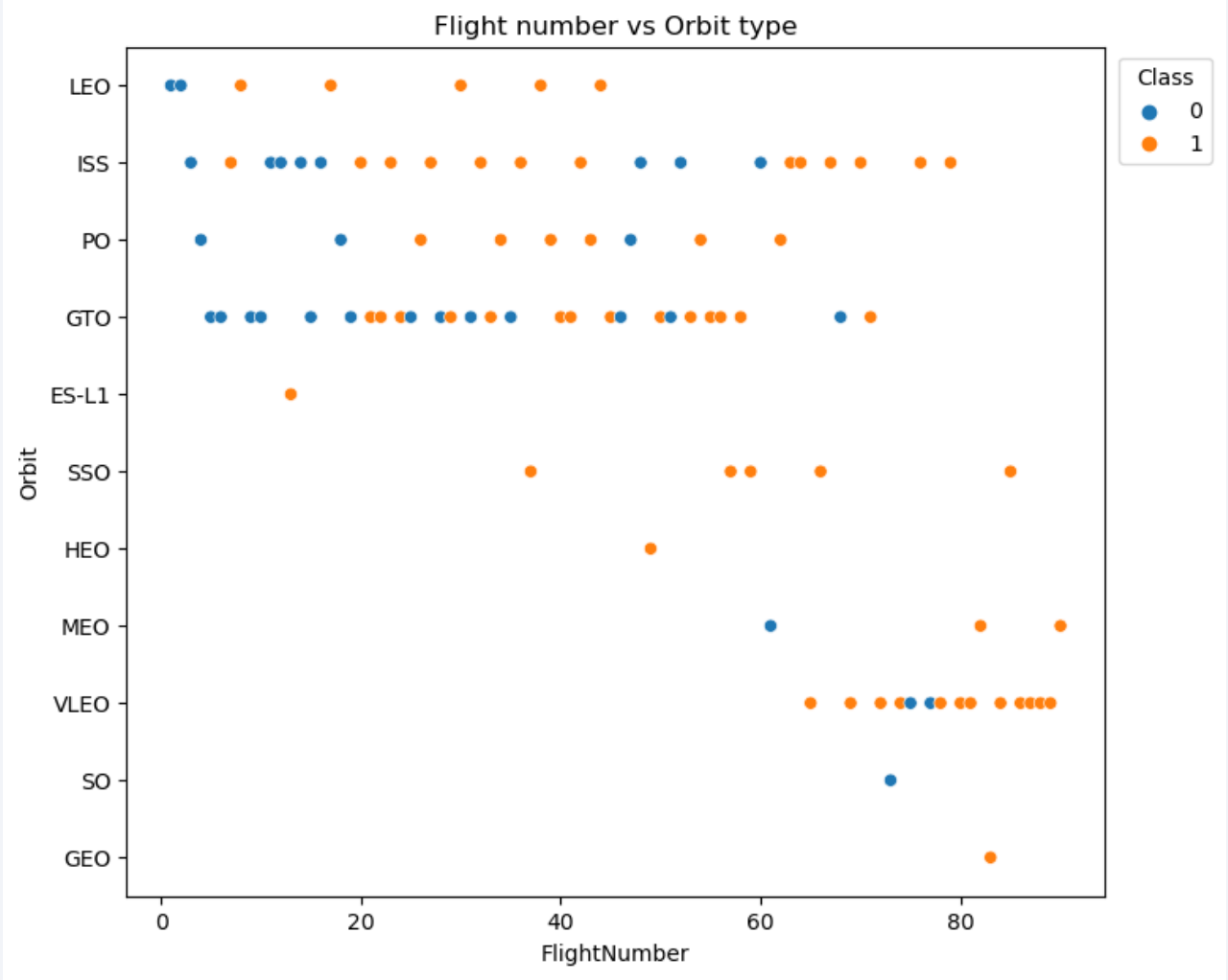
Success Rate vs. Orbit Type

- The launch success rate depend on the orbit of the satellite
- ES-LI,GEO,HEO,SSO orbits have 100% success rate



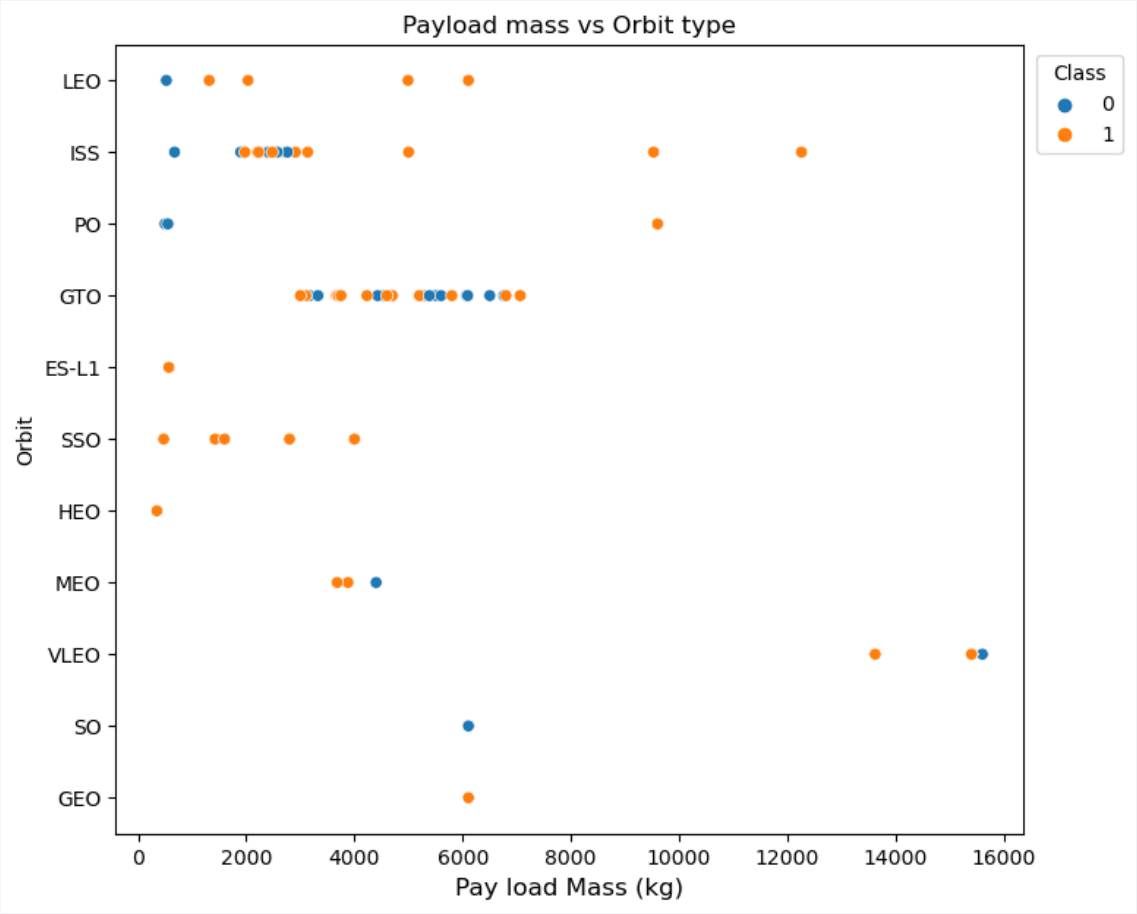
Flight Number vs. Orbit Type

- Success rate is independent of Flight number for ES-LI, GEO, HEO, and SSO orbits.
- Though the success rate increases with the flight number at certain orbits, it lacks a pattern.



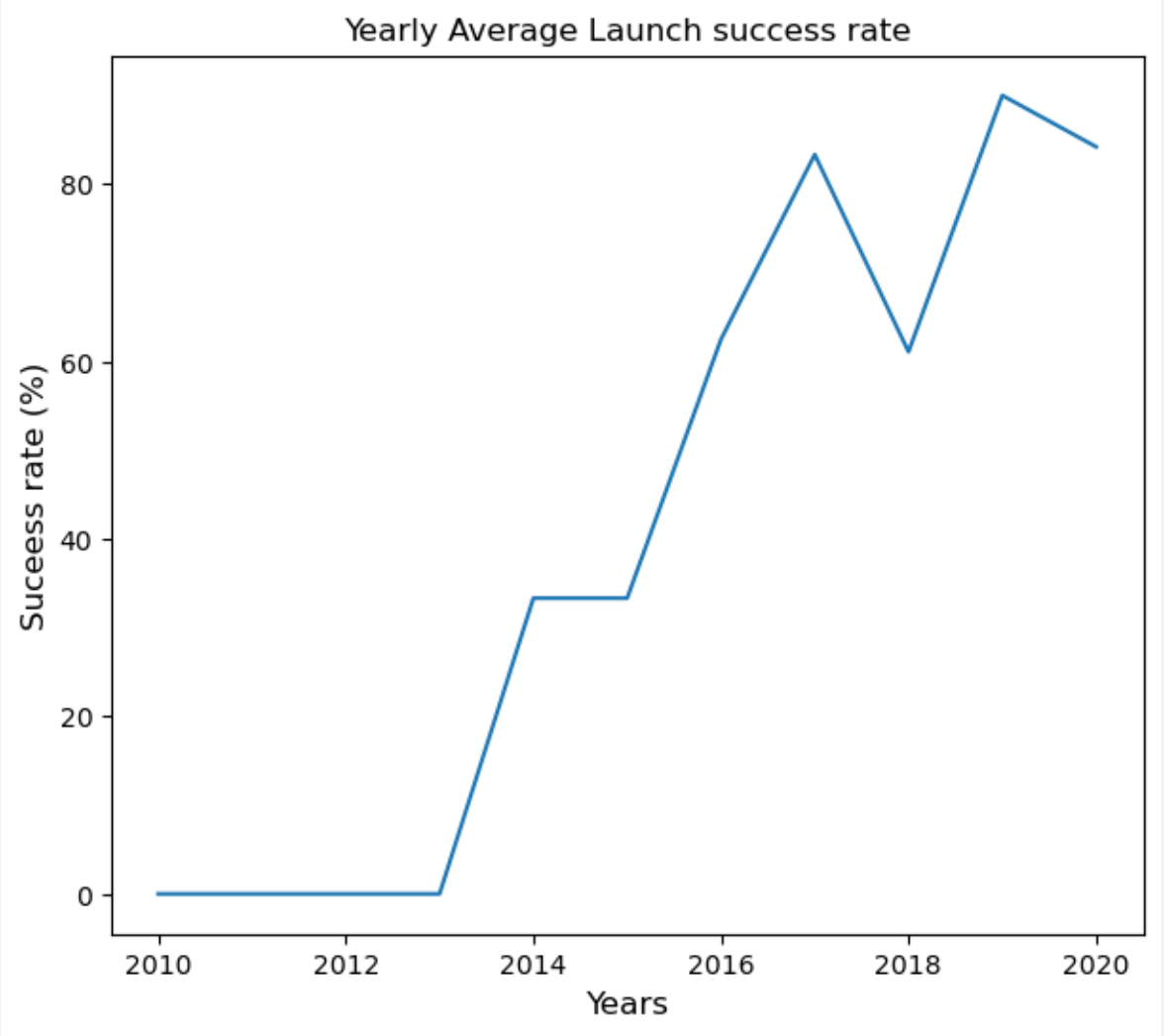
Payload vs. Orbit Type

- The success rate is higher at heavier payloads for LEO,ISS and PO orbits



Launch Success Yearly Trend

- The annual average launch success rate has seen significant growth, rising from zero percent in 2000 to over 80% by 2020.



All Launch Site Names

- In the space mission, there are four unique launch sites, and the DISTINCT command in SQL is employed to obtain these results

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

```
%sql select distinct(Launch_Site) from SPACEXTABLE
```

✓ 0.0s

```
 * 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 WHERE clause, followed by the LIKE clause, filters launch sites containing the substring 'CCA.' Using LIMIT 5 displays the first 5 records from this filtered result

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

```
%sql Select Launch_Site from SPACEXTABLE Where Launch_Site like 'CCA%' Limit 5
```

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

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

Total Payload Mass

- The SUM aggregate function is employed to calculate the total payload carried by boosters after filtering using the WHERE clause

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

```
%sql select Sum(PAYLOAD_MASS__KG_) as total_payload_mass_in_kg from SPACEXTABLE Where Customer =='NASA (CRS)'
```

✓ 0.0s

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

total_payload_mass_in_kg
45596

Average Payload Mass by F9 v1.1

- The AVG aggregate function is employed to calculate the average payload mass carried by booster version F9 v1.1

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

```
%sql select AVG(PAYLOAD_MASS__KG_) as avg_payload_in_kg_F9_v1_1  from SPACEXTABLE where Booster_version='F9 v1.1'
```

✓ 0.0s

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

avg_payload_in_kg_F9_v1_1
2928.4

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad is ‘2015-12-22’
- Have used ORDER BY combined with the LIMIT command to obtain the result

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

Hint:Use min function

```
%sql select Date,Landing_Outcome from SPACEXTABLE where Landing_Outcome like '%success%' order by Date limit 1
```

✓ 0.0s

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

Date	Landing_Outcome
2015-12-22	Success (ground pad)

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 between 4000 and 6000 kg is shown in the figure.
- The AND command is used in the query to intersect the two conditions

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
        Booster_Version,Landing_Outcome,PAYLOAD_MASS__KG_
        from SPACEXTABLE
        where (Landing_Outcome like '%success (dron%') and (PAYLOAD_MASS__KG_ between 4000 and 6000)
```

✓ 0.0s

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

Booster_Version	Landing_Outcome	PAYLOAD_MASS__KG_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

Total Number of Successful and Failure Mission Outcomes

- The total number of successful outcome is 100 and failure mission outcome is 1.
- Aggregate function COUNT is applied to the GROUP BY clause on the mission outcome

List the total number of successful and failure mission outcomes

```
%sql select Mission_Outcome,count(Mission_Outcome) from SPACEXTABLE group by Mission_Outcome
# df
```

✓ 0.0s

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

Mission_Outcome	count(Mission_Outcome)
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 are given in figure
- Here a subquery is used inside the WHERE clause to obtain the maximum payload

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

```
%%sql select
    distinct(Booster_Version),
    PAYLOAD_MASS__KG_
    from SPACEXTABLE
    where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

✓ 0.0s

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

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

2015 Launch Records

- The failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015 is displayed in the figure
- Used substr to extract month and year from dates, AND command to satisfy the conditions in the WHERE clause.

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, 4, 2) as month to get the months and substr(Date,7,4)='2015' for year.

```
%%sql select
        substr(Date,6,2) as month,
        substr(Date,1,4) as year,
        Landing_Outcome,
        Booster_Version,
        Launch_Site
        from SPACEXTABLE
        where (Landing_Outcome like '%failure (d%') and (Date between '2015-01-01' and '2015-12-31')
```

✓ 0.0s Pytho

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

month	year	Landing_Outcome	Booster_Version	Launch_Site
10	2015	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	2015	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- The Rank of the landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order is shown here
- Used GROUP BY and ORDER By command in the query

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(Landing_Outcome) as outcomes
        from SPACEXTABLE
        where Date between '2010-06-04' and '2017-03-20'
        group by Landing_Outcome
        order by outcomes desc
```

✓ 0.0s

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

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

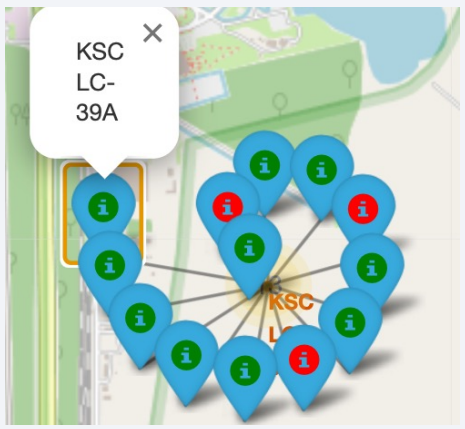

Section 3
Launch Sites
Proximities Analysis

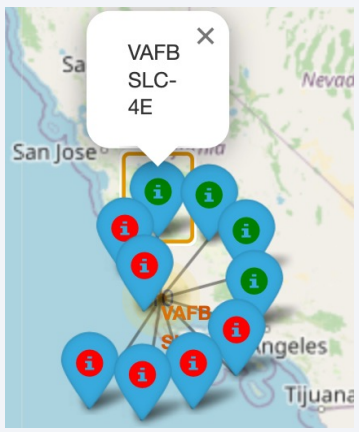
Launch site on a Global Map

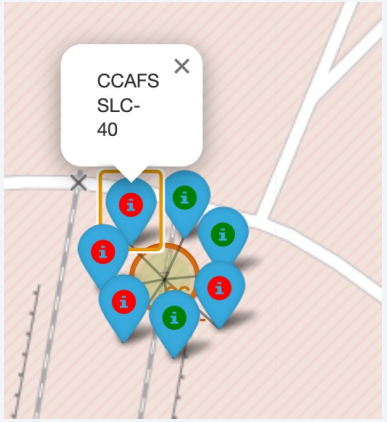
- All the launch sites are along the coast
- East coast of US has more number of launch sites

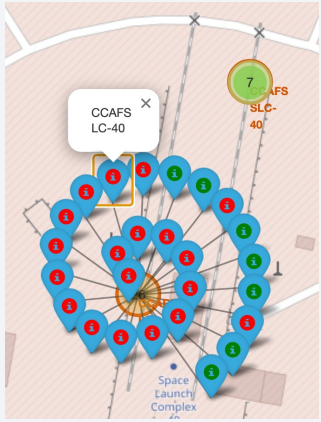
Success and Failure at each site

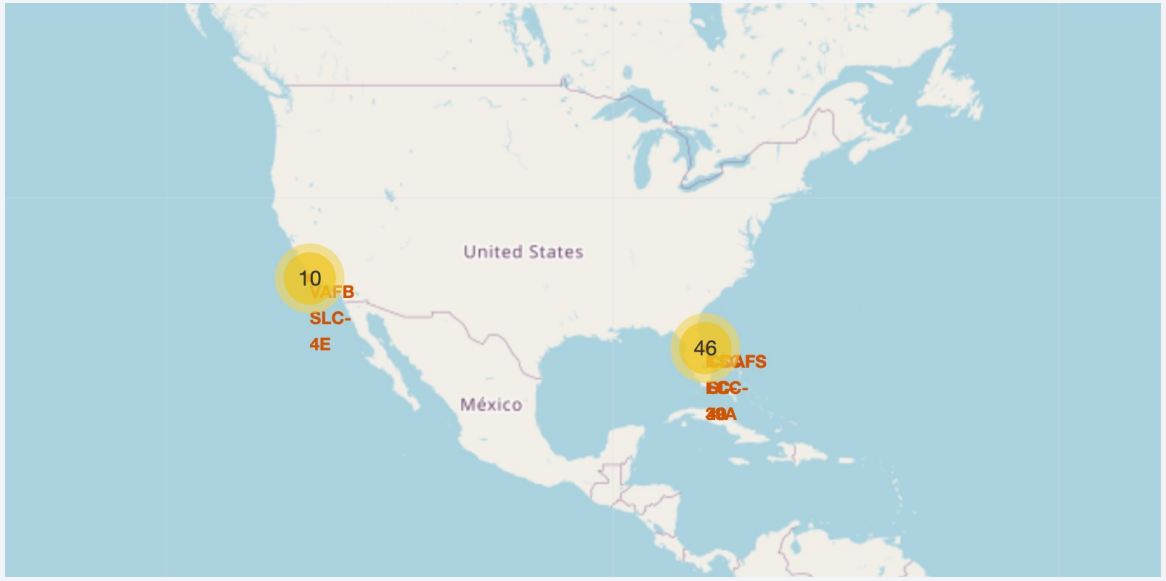
- The red markers at each location shows the failed launches
- The green markers at each location shows the successful launches





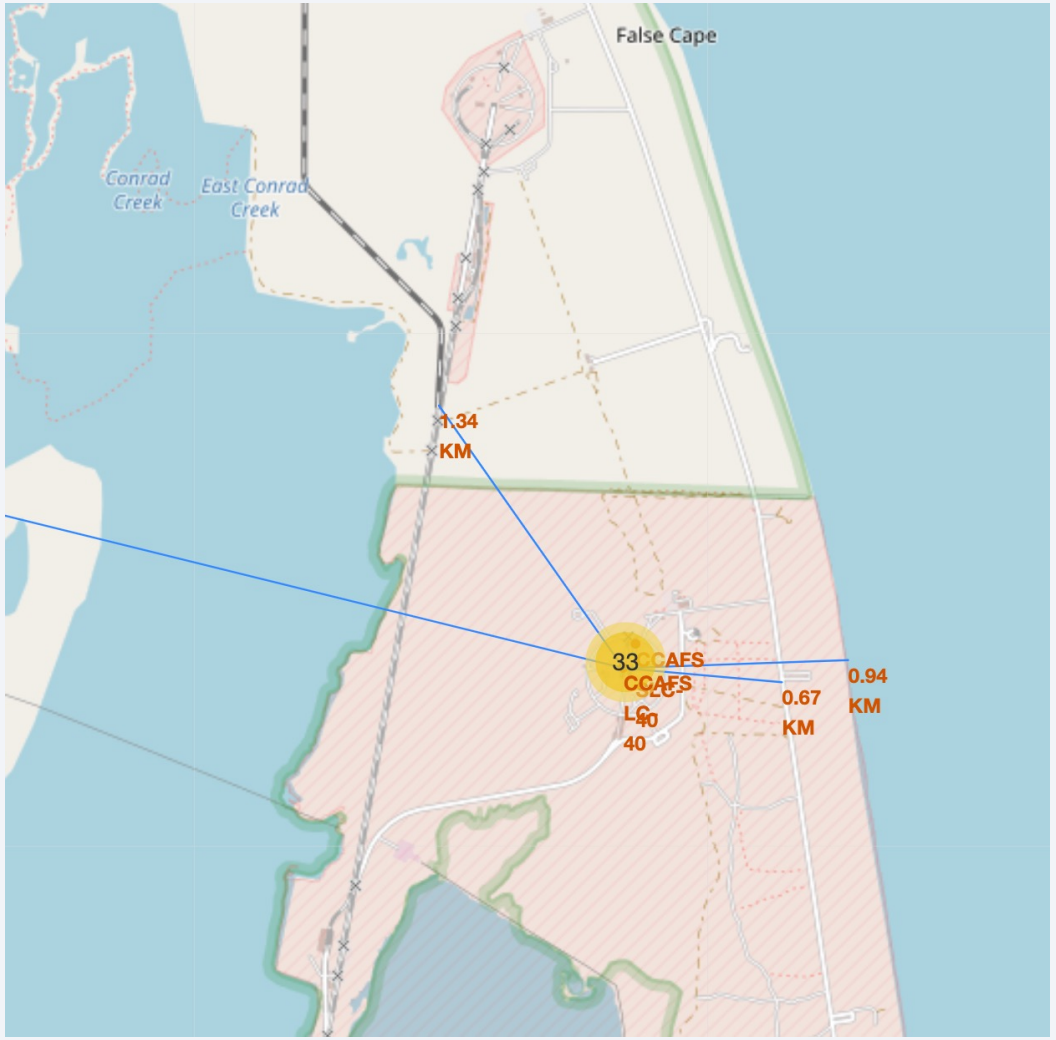


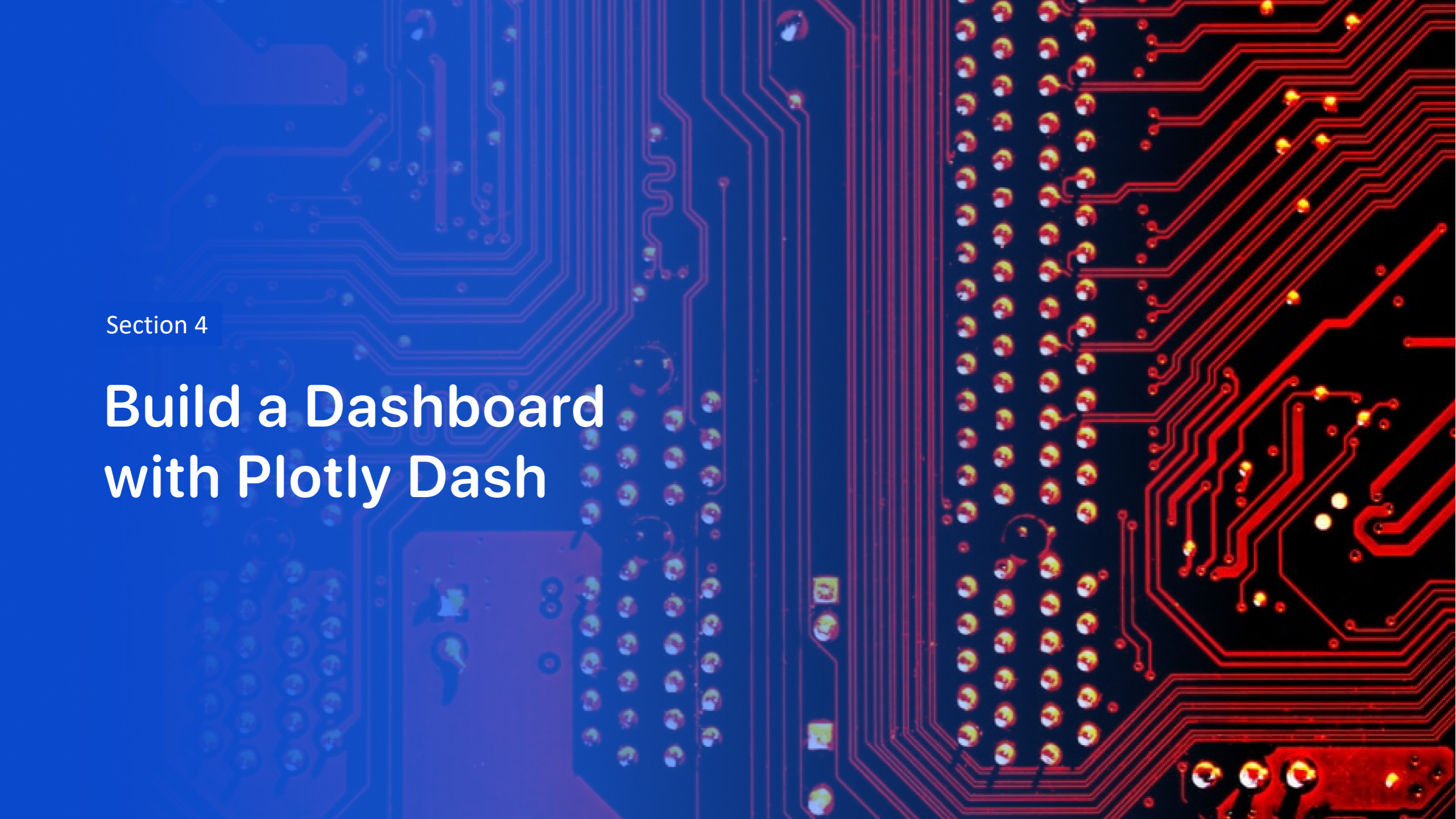




Launch sites proximity to landmarks

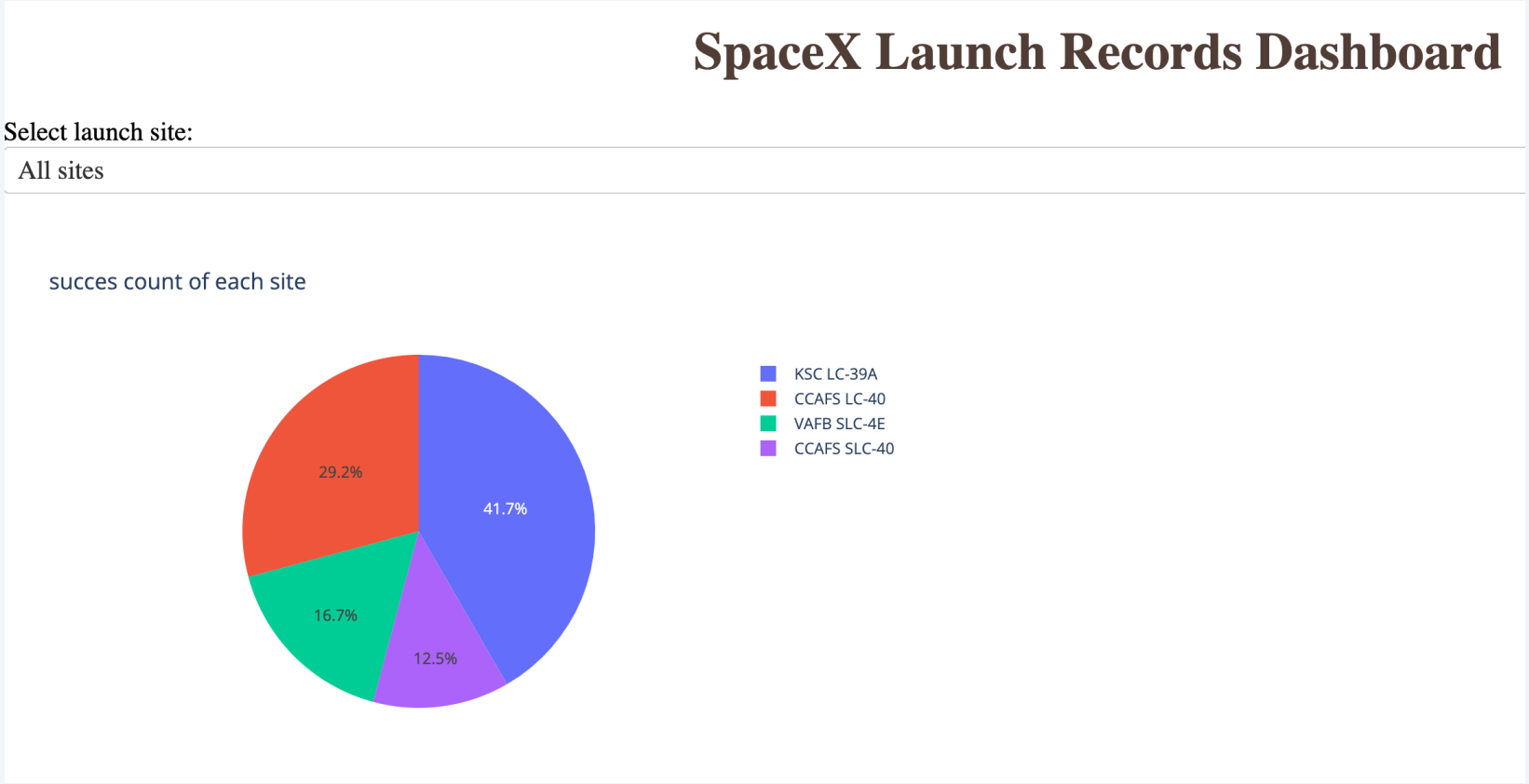
- The Launch sites are with 2km distance from railway, coast and a highway. But far from the cities




Section 4
Build a Dashboard with Plotly Dash

Launch success at each site

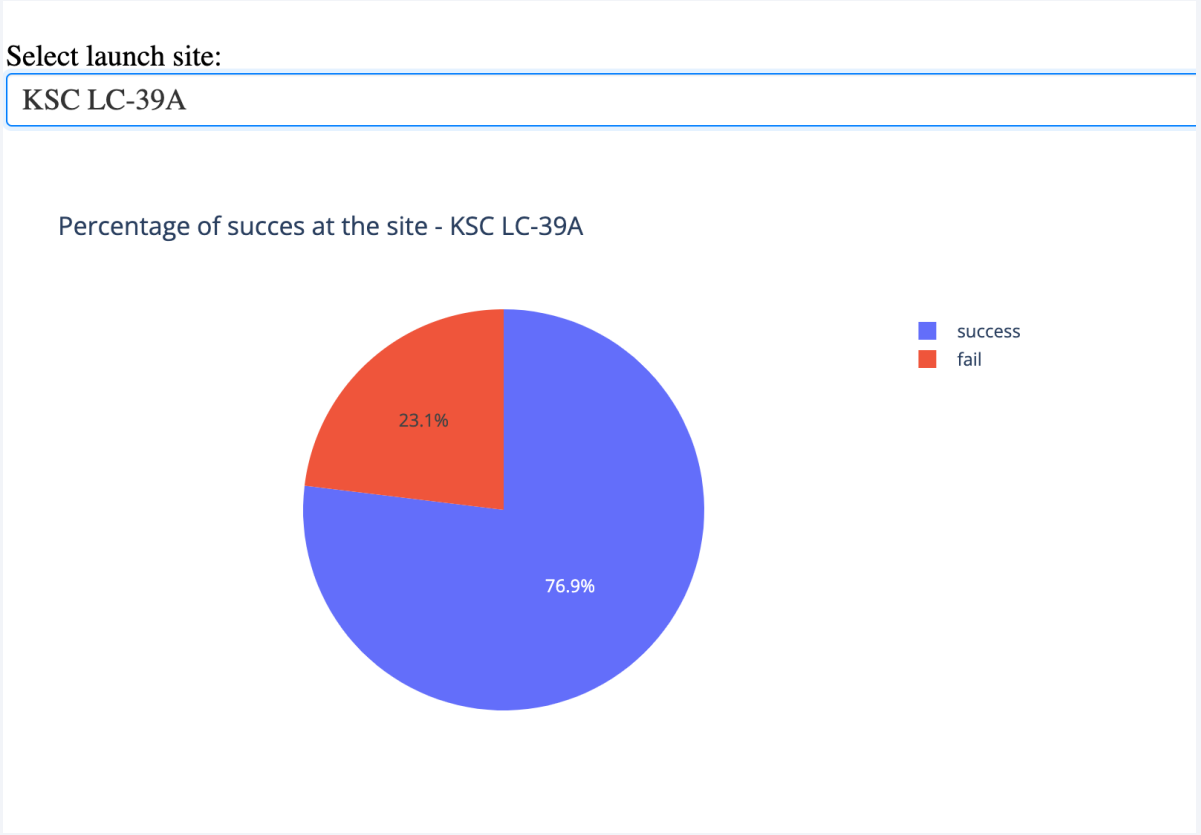
- KSC LC-39A has the highest number of successful launches



Launch site with highest success ratio

- KSC LC-39A has the highest launch success ratio

- KSC LC-39A has nearly 77% success rate in launching



Payload VS launch Outcomes for all sites

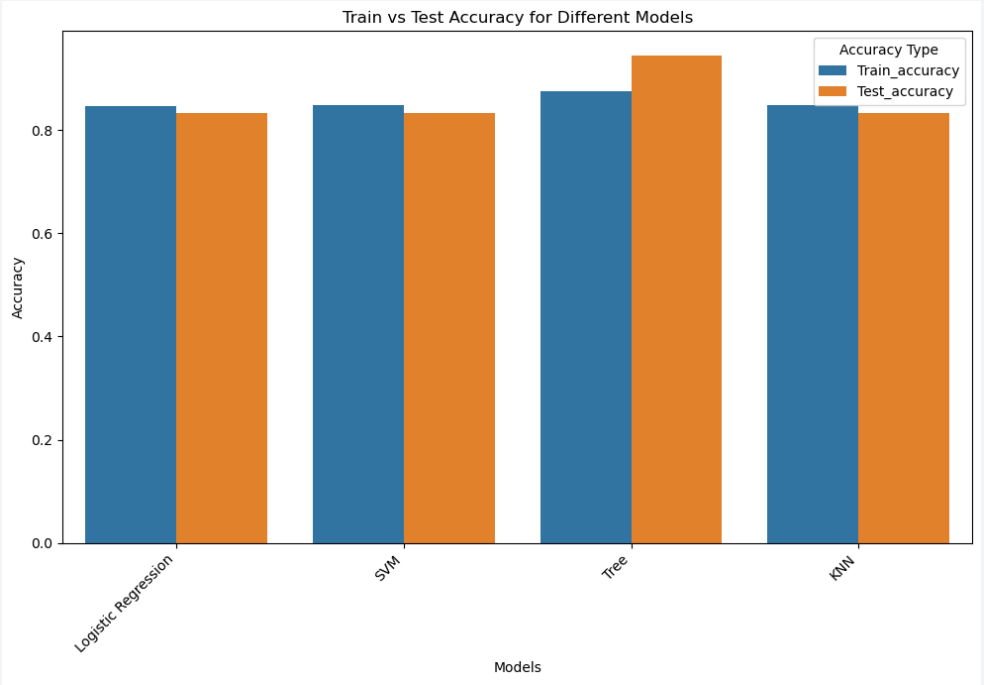
- The success rate is higher for payloads in the lower range (0-5500kg) compared to those above 5500kg

Section 5

Predictive Analysis (Classification)

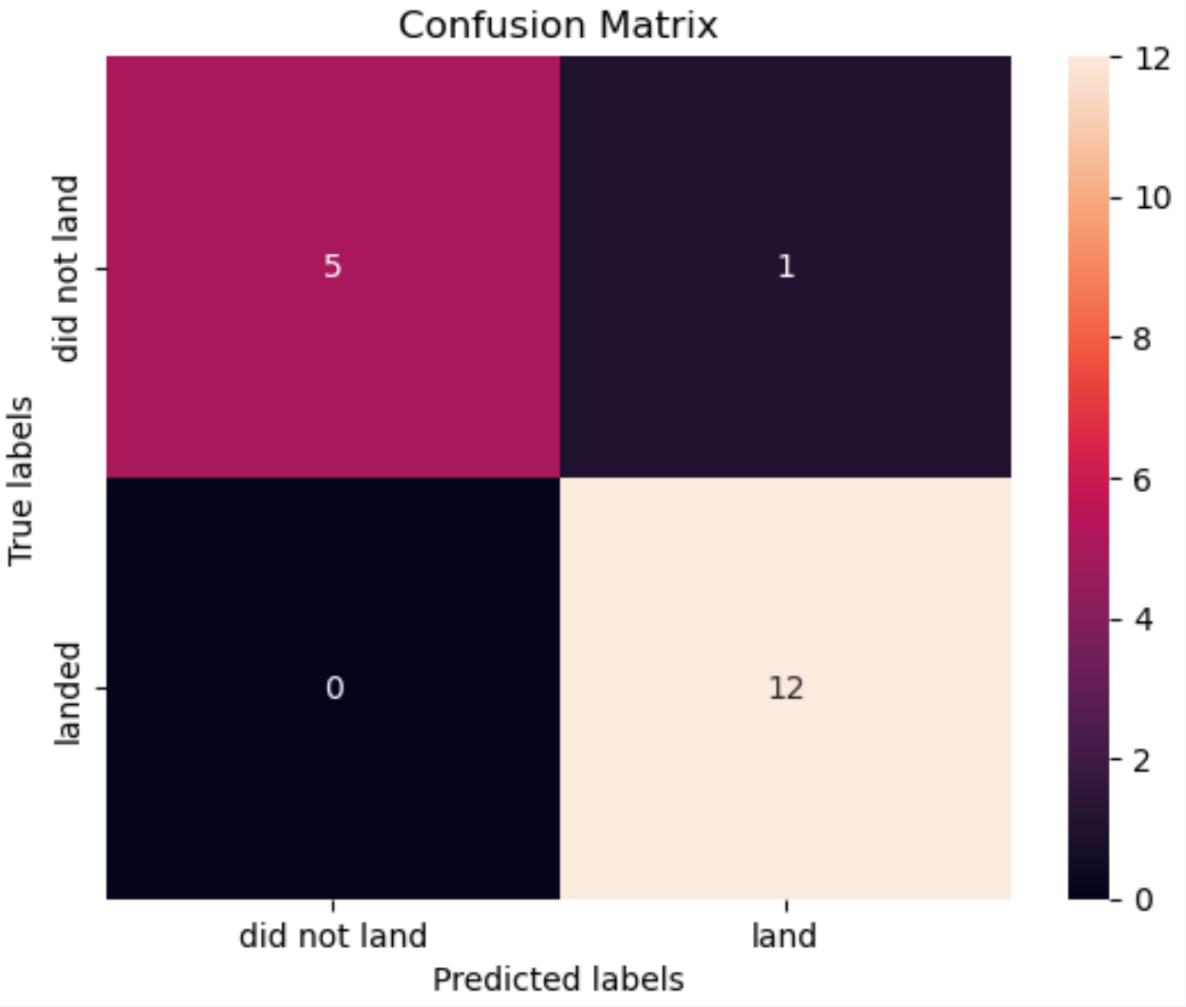
Classification Accuracy

- The dataset includes both training and testing subsets.
- The models undergo training using the training data.
- Accuracy is assessed on both the training and testing datasets, as illustrated in the figure.
- The decision tree model achieves the highest classification accuracy, with a training accuracy of 0.875 and a testing accuracy of 0.944.



	model	Train_accuracy	Test_accuracy
0	Logistic Regression	0.846429	0.833333
1	SVM	0.848214	0.833333
2	Tree	0.875000	0.944444
3	KNN	0.848214	0.833333

Confusion Matrix



- The confusion matrix from the decision tree model
  - It correctly predicted 12 positive cases
  - It made no false negative predictions.
  - It incorrectly predicted 1 negative case as positive.
  - It correctly predicted 5 negative cases

Conclusions

- The success of a mission can be attributed to several factors, including the launch site, the orbit, payload, and, notably, the number of previous launches.
- The success rate of launches increased from 2010 to 2020, with the launch site KSC LC-39A having the highest success rate. Additionally, missions with lower payload masses showed higher success rates.
- The orbits with the best success rates are GEO, HEO, SSO, and ES-L1.
- We can indeed assume that knowledge gained from previous launches has contributed to the transition from launch failures to successes, as evident from the increase in success rates with higher flight numbers

Thank you!