IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

James Lynch
5 July 2024

Outline

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

Executive Summary

- **Summary of methodologies**
  - Data Collection through API
  - Data Wrangling
  - Exploratory Data Analysis
  - Interactive Visual Analytics with Folium
  - Predictive Analysis
- **Summary of all results**
  - Exploratory Data Analysis result
  - Interactive analytics with visuals
  - Predictive Analytics results

Introduction

- **Project background and context**

  SpaceX offers Falcon 9 launches for $62 million, significantly cheaper than other providers at $165 million, due to reusing the first stage. Predicting the first stage's landing success helps estimate launch costs, aiding competitors in bidding. This project aims to develop a machine learning pipeline to predict successful landings.

- **Problems you want to find answers**

  - Determine what factors influence the successful landing of a rocket?
  - The connection of features that determine a the success rate of a landing?
  - The necessary operating conditions to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**

  - Data from SpaceX was obtained using Web Scraping and Data Wrangling from the following sources:

    - SpaceXAPI(https://api.spacexdata.com/v4/rockets/)
    - WebScraping (https://en.wikipedia.org/wiki/List_of_Falcon/_9/_and_Falcon_Heavy_launches)

- **Perform data wrangling**
  - Data was collected by importing necessary libraries.
- **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 prepare the data, train models, evaluate the performance to make predictions.

Data Collection

- **The data was collected using various methods**
  - Data collection was done using get request to the SpaceX API.
  - Next, we decoded the response content as a Json using .json() function call and turn it into a pandas dataframe using .json_normalize().
  - We then cleaned the data, checked for missing values and fill in missing values where necessary.
  - In addition, we performed web scraping from Wikipedia for Falcon 9 launch records with BeautifulSoup.
  - The objective was to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for future analysis.

Data Collection – SpaceX API

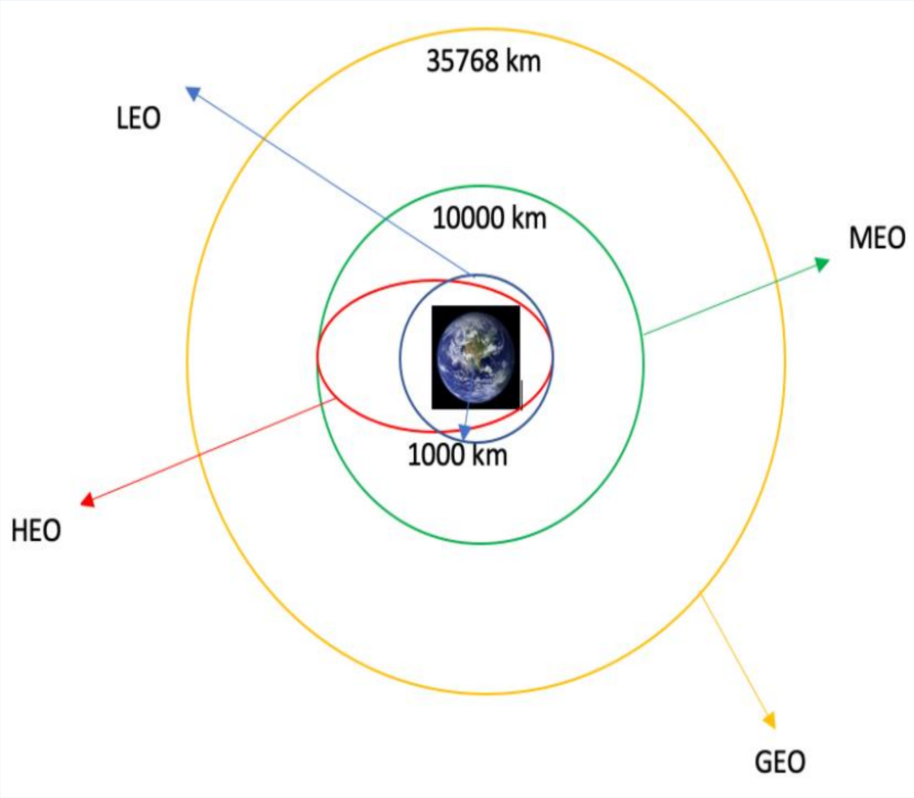
- We used the get request to the SpaceX API to collect data, clean the requested data and did some basic data wrangling and formatting.
- SpaceX API data collection is available on this Github : https://github.com/Jamsey911/spacex/blob/main/spacex_data_collection_api.ipynb

Data Collection - Scraping

- We applied web scrapping to webscrap Falcon 9 launch records with BeautifulSoup
- We parsed the table and converted it into a pandas dataframe.
- The link to the notebook is : https://github.com/Jamsey911/spacex/blob/main/webscraping.ipynb

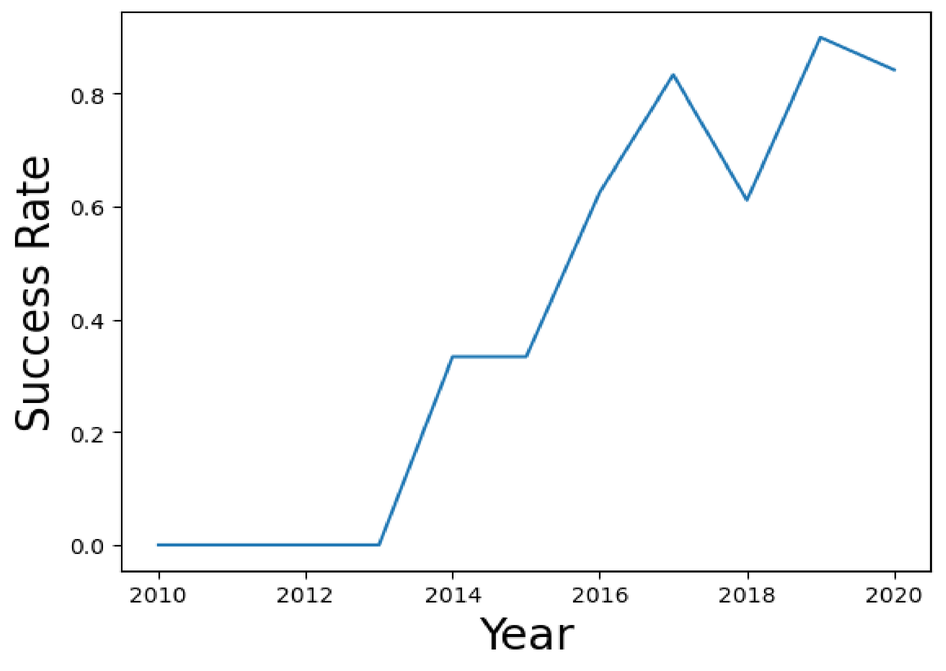
Data Wrangling

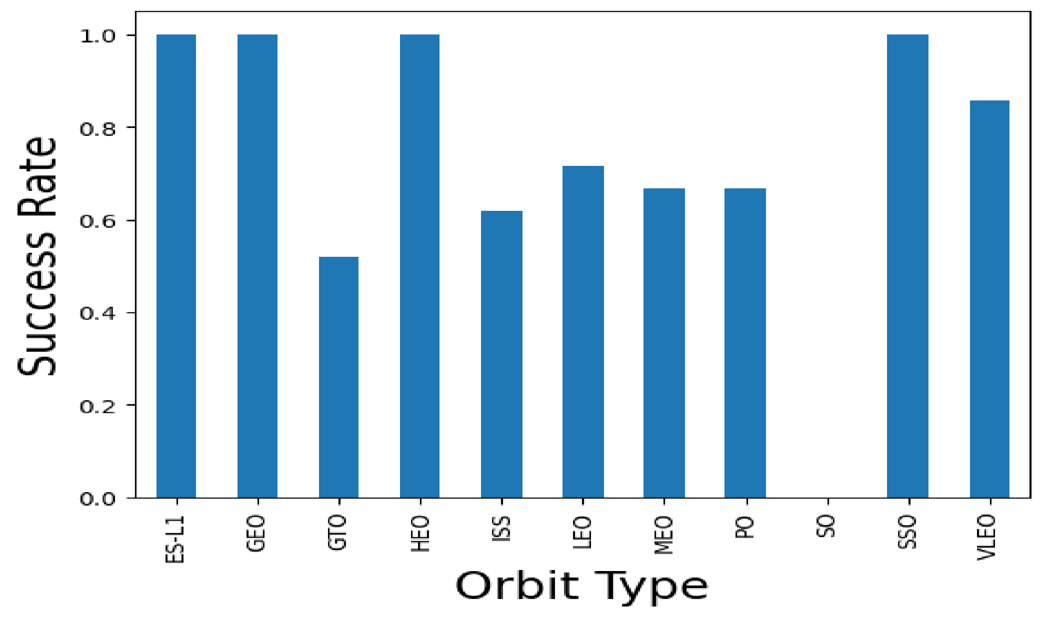
- We performed exploratory data analysis and determined the training labels.
- We calculated the number of launches at each site, and the number and occurrence of each orbits
- We created landing outcome label from outcome column and exported the results to csv.
- The link to the notebook is : https://github.com/Jamsey911/spacex/blob/main/Data_wrangling.ipynb



EDA with Data Visualization

- We explored the data by visualizing the relationship between flight number and launch Site, payload and launch site, success rate of each orbit type, flight number and orbit type, the launch success yearly trend.





- The link to the notebook is https://github.com/Jamsey911/spacex/blob/main/eda_data_viz.ipynb

EDA with SQL

- We applied EDA with SQL to get insight from the data. We wrote queries to find out for instance:
  - The names of unique launch sites in the space mission.
  - The total payload mass carried by boosters launched by NASA (CRS)
  - The average payload mass carried by booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing outcomes in drone ship, their booster version and launch site names.
- The link to the notebook is : https://github.com/Jamsey911/spacex/blob/main/eda_sql.ipynb

Build an Interactive Map with Folium

- Folium was used to map and mark all launch sites using map objects such as markers, circles, lines to mark the success or failure of launches.
- A class was created to determine success (1) or failure (0).
- To identify launch sites with relatively high success rate, color-labeled marker clusters were used.
- Distances between a launch site to its proximities were then calculated and answered some question for instance:
  - Are launch sites near railways, highways and coastlines.
  - Do launch sites keep certain distance away from cities.

- The link to the notebook is : https://github.com/Jamsey911/spacex/blob/main/launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Plotly dash was used to build an interactive dashboard
- Pie charts were used to show the total launches for each site
- A Scatter graph is used to show how payload may be correlated with mission outcomes for selected site(s).
- The link to the notebook is https://github.com/Jamsey911/spacex/tree/main/plot_capstone.code-workspace

Predictive Analysis (Classification)

- Data was loaded using NumPy and pandas
- Data was then transformed to split our data into training and testing sets.
- Different machine learning models were created to tune different hyperparameters using GridSearchCV.
- Accuracy as the metric was then used utilized for our model to enhance its performance through feature engineering and algorithm tuning..
- The best performing classification model would then be determined.
- The link to the notebook is https://github.com/Jamsey911/spacex/blob/main/Machine_Learning_Prediction.ipynb

Results

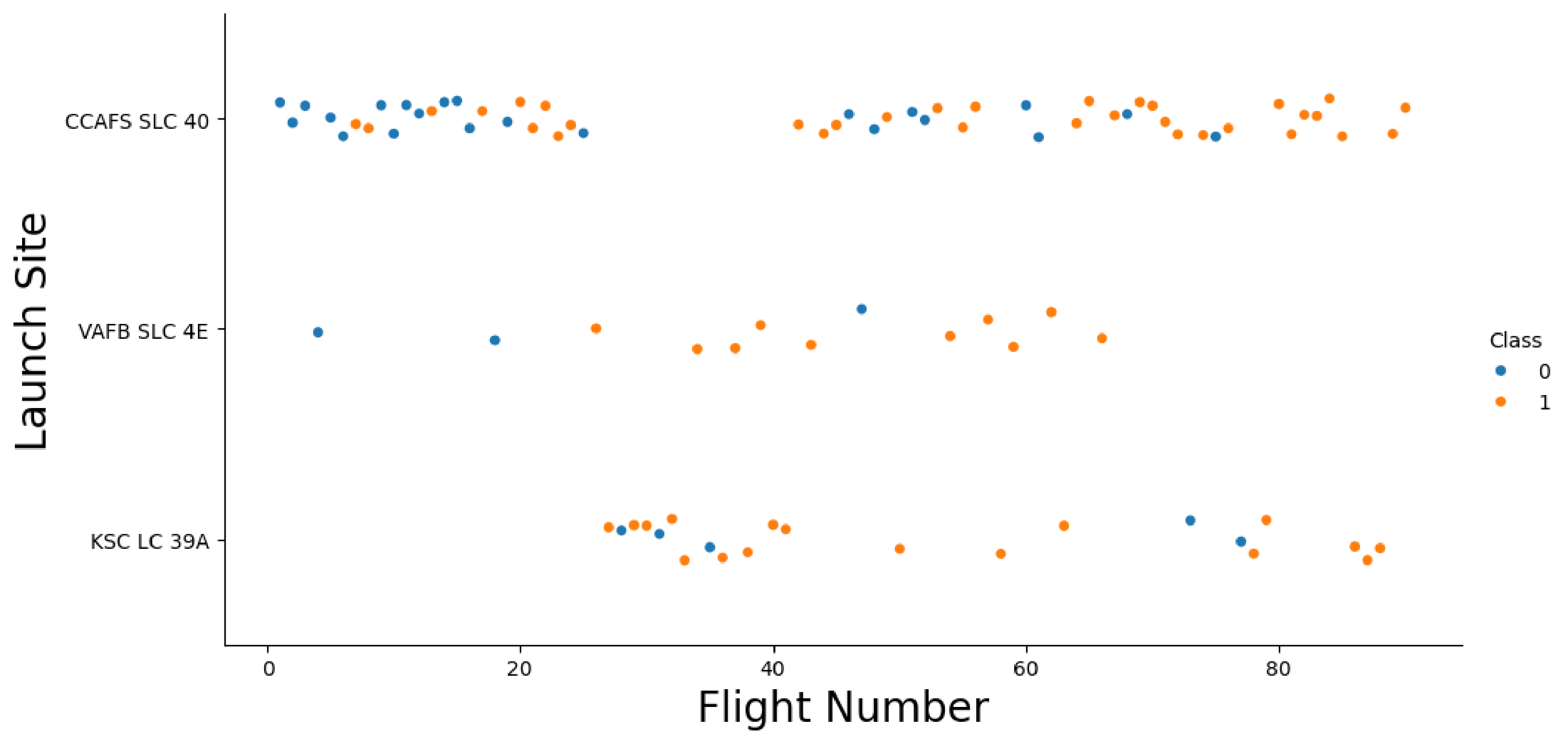
- Exploratory data analysis results:
  - Space X uses 4 different launch sites;
  - The average payload of F9 v1.1 booster is 2,928 kg;
  - The first successful landing outcome on a ground pad happened in 2015 five years after the first launch;
  - Many Falcon 9 booster versions were successful at landing in drone ships having payload above the average;
  - 66% of mission outcomes were successful;
  - Two booster versions failed at landing in drone ships in 2015: F9 v1.1 B1012 and F9 v1.1 B1015;
  - Successful outcomes increased as years passed.

Section 2

Insights drawn from EDA

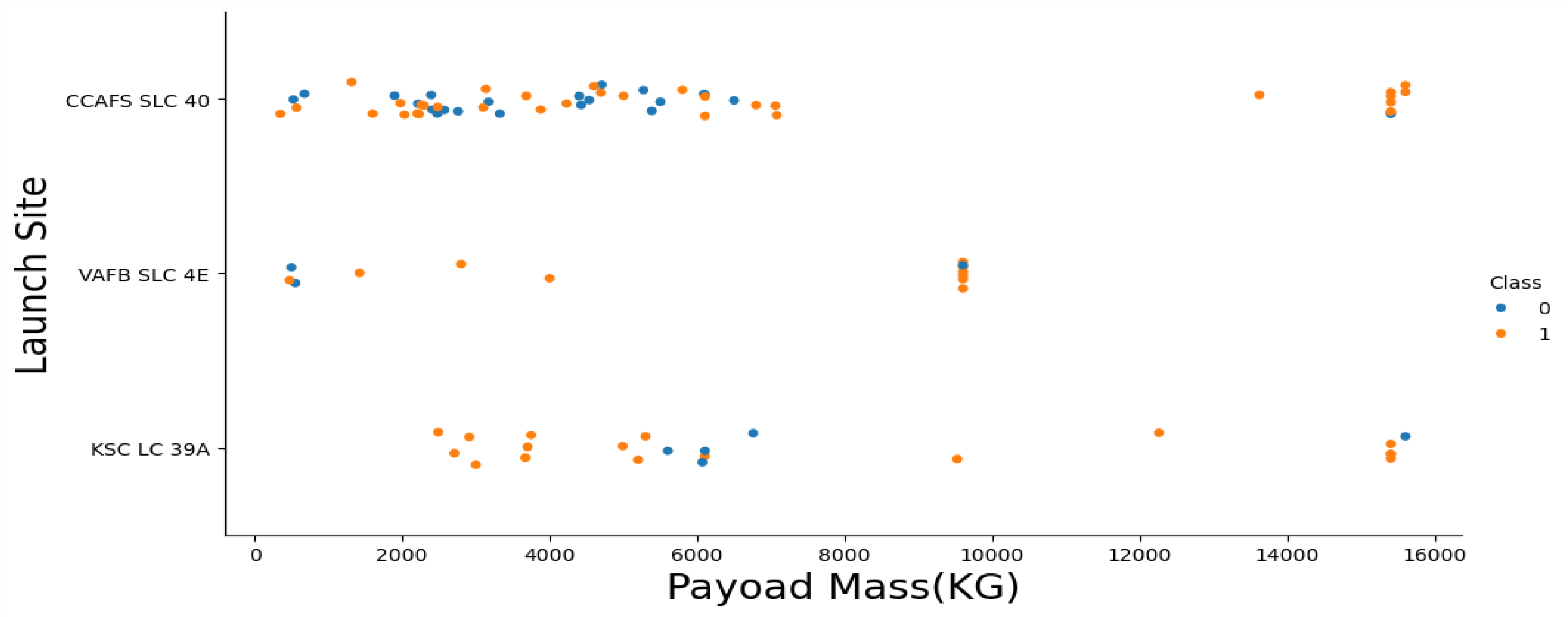
Flight Number vs. Launch Site

- From the plot, we see that the launch location CCAFS SLC 40 was used the most and the success rate of this site increased the more flights took place.



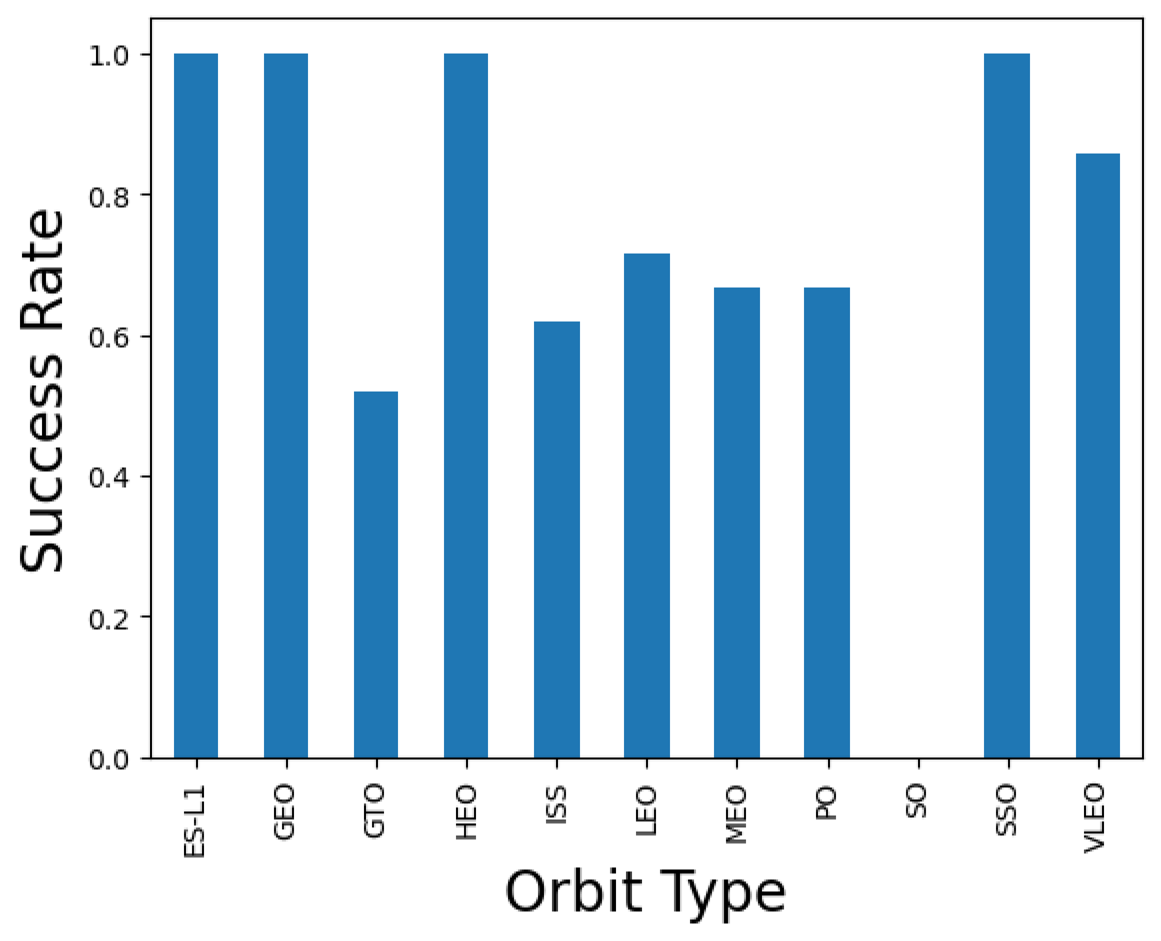
Payload vs. Launch Site

- From the plot, we see that the VAFB-SLC launch site launched no rockets for a heavy payload mass(greater than 10000).



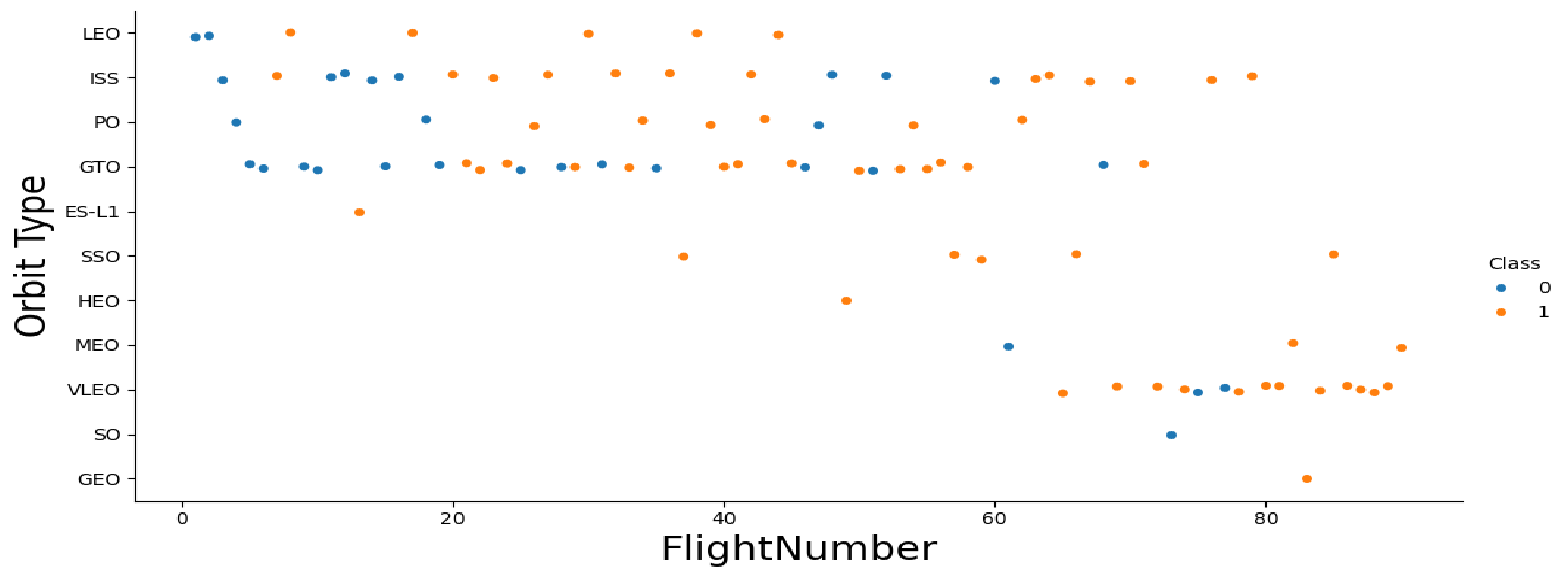
Success Rate vs. Orbit Type

- From the plot, we can see that ES-L1, GEO, HEO and SSO orbit types had the most success rate.
- We see that the SO orbit type has a zero 0% success rate



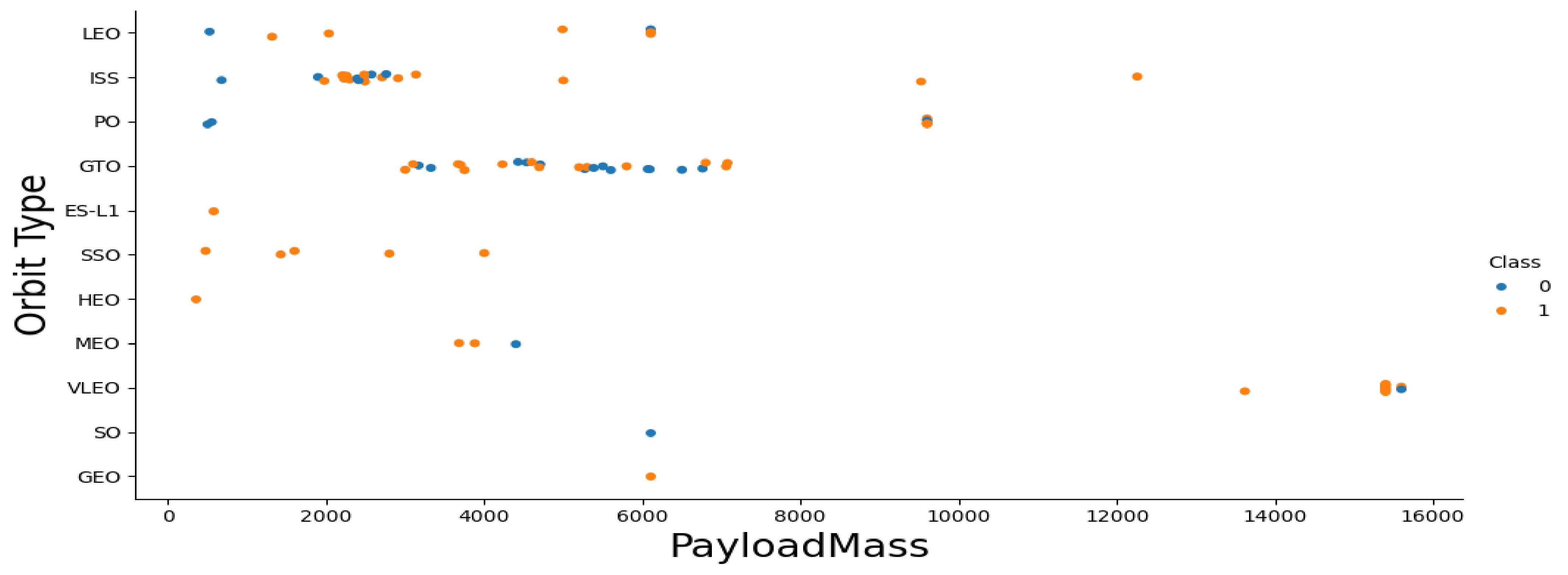
Flight Number vs. Orbit Type

- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.



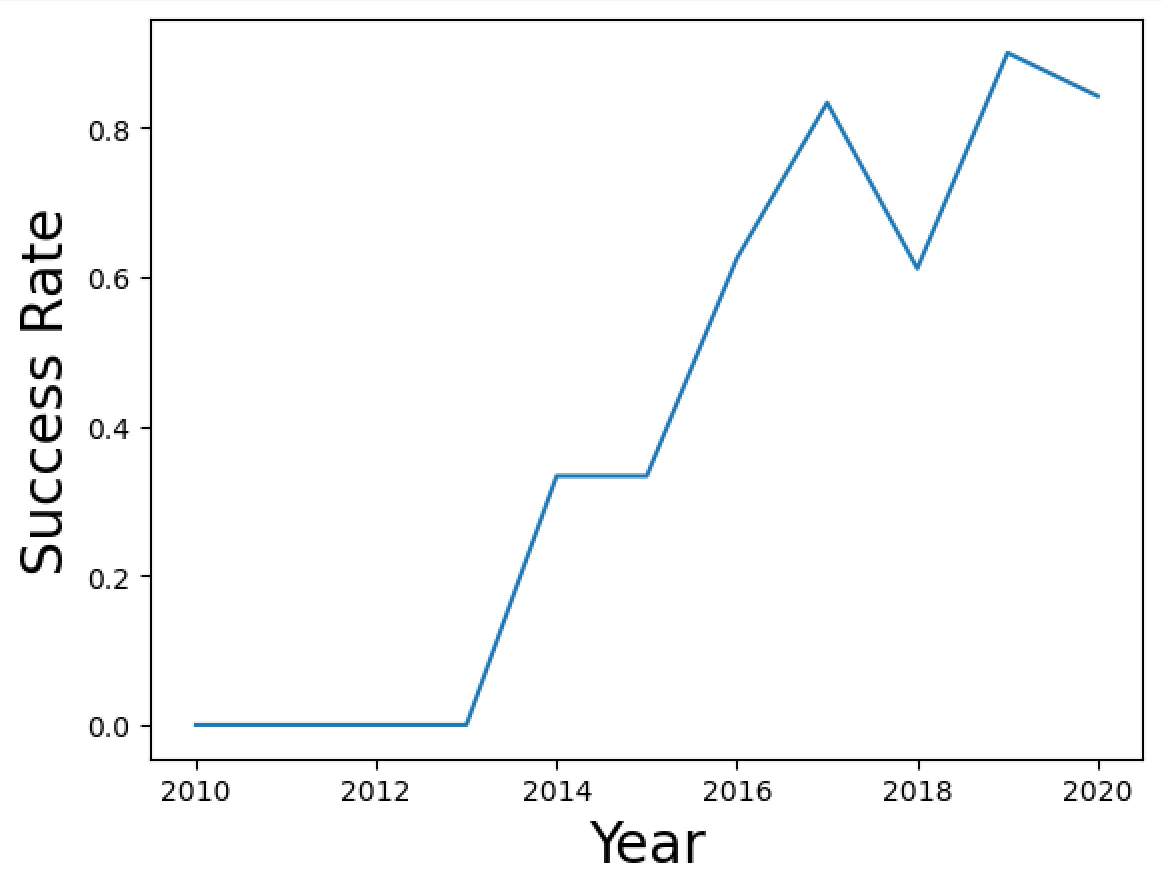
Payload vs. Orbit Type

- We can observe that with heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.



Launch Success Yearly Trend

- From the plot, we can observe that success rate since 2013 kept on increasing till 2020



All Launch Site Names

- We filtered the data received from our SpaceX data frame using the key word **DISTINCT** to show only unique launch sites.

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

```
In [9]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTABLE ;

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

Out[9]:

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

Launch Site Names Begin with 'CCA'

- We used the **LIKE** query to display 5 records where launch sites begin with `CCA`

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

In [15]:
```
#%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%'  LIMIT 5;
%sql SELECT * FROM SPACEXTABLE WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Out[15]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- We calculated the total payload carried by boosters from NASA as 48213(KG) using the query below

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

In [11]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS_TOTAL FROM SPACEXTABLE WHERE Customer LIKE '%NASA (CRS)%';
```

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

Out[11]:

PAYLOAD_MASS_TOTAL
48213

Average Payload Mass by F9 v1.1

- We calculated the average payload mass carried by booster version F9 v1.1 as 2534.7

Task 4

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

In [12]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS PAYLOAD_MASS__AVG FROM SPACEXTABLE WHERE Booster_Version LIKE '%F9 v1.1%';
```

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

Out[12]:

PAYLOAD_MASS__AVG
2534.6666666666665

First Successful Ground Landing Date

- We observed that the date of the first successful landing outcome on ground pad was 22nd December 2015

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

Hint:Use min function

In [13]:

```
%sql SELECT MIN(Date) AS First_Successful_GP FROM SPACEXTABLE WHERE Landing_Outcome LIKE '%Success (ground pad)%';
```

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

Out[13]:

First_Successful_GP
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The **WHERE** clause was used to filter for boosters which have successfully landed on drone ships and applied the **AND** condition to determine successful landing with a payload mass greater than 4000 but less than 6000

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

In [14]:
```
%sql SELECT Booster_Version FROM SPACEXTABLE WHERE Landing_Outcome = 'Success (drone ship)' AND PAYLOAD_MASS__KG_ > 4000 AND
```

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

Out[14]:

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

Successful Drone Ship Landing with Payload between 4000 and 6000

- 2 **CASE's** were used to create the success and failed outcomes along with the wildcard **LIKE** with '%' to filter the Mission_Outcome.

List the total number of successful and failure mission outcomes

[44]:
```
%%sql SELECT SUM(CASE WHEN Mission_Outcome LIKE 'Success%' THEN 1 ELSE 0 END) AS successful_count,
    SUM(CASE WHEN Mission_Outcome LIKE 'Failure%' THEN 1 ELSE 0 END) AS failed_count
FROM SPACEXTABLE
WHERE Mission_Outcome LIKE 'Success%' OR Mission_Outcome LIKE 'Failure%';
```

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

[44]:

successful_count	failed_count
100	1

Successful Drone Ship Landing with Payload between 4000 and 6000

- The booster that carried the maximum payload was determined using a subquery in the **WHERE** clause and the **MAX()** function was used to find highest amount of weight.

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

[45]:
```
%%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ FROM SPACEXTABLE
    WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTABLE)
```

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

[45]:

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

- A combination of **WHERE** , **LIKE**, **AND**, and **BETWEEN** clauses was used to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015
- A **CASE** clause was created to display the month's name when the data is displayed.

List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

Note: SQLLite does not support monthnames. So you need to use substr(Date, 6,2) as month to get the months and substr(Date,0,5)='2015' for year.

In [17]:
```
%%sql
SELECT
    (CASE substr(Date, 6, 2)
        WHEN '01' THEN 'January'
        WHEN '02' THEN 'February'
        WHEN '03' THEN 'March'
        WHEN '04' THEN 'April'
        WHEN '05' THEN 'May'
        WHEN '06' THEN 'June'
        WHEN '07' THEN 'July'
        WHEN '08' THEN 'August'
        WHEN '09' THEN 'September'
        WHEN '10' THEN 'October'
        WHEN '11' THEN 'November'
        WHEN '12' THEN 'December'
        ELSE 'Unknown'
    END) AS Month,
    substr(Date, 0, 5) AS Year,
    Booster_Version,
    Launch_Site,
    Landing_Outcome
FROM SPACEXTBL
WHERE Landing_Outcome = 'Failure (drone ship)' AND substr(Date, 0, 5) = '2015';
```

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

Out[17]:

Month	Year	Booster_Version	Launch_Site	Landing_Outcome
January	2015	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
April	2015	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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

- We selected Landing outcomes and the **COUNT** of landing outcomes from the data and used the **WHERE** clause to filter for landing outcomes **BETWEEN** 2010-06-04 to 2010-03-20.
- We applied the **GROUP BY** clause to group the landing outcomes and the **ORDER BY** clause to order the grouped landing outcome in descending order.

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 [18]:

```
%%sql
SELECT Landing_Outcome, count(*) as count_outcomes FROM SPACEXTBL
WHERE DATE between '2010-06-04' and '2017-03-20'
group by [Landing_Outcome] order by count_outcomes DESC;
```

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

Out[18]:

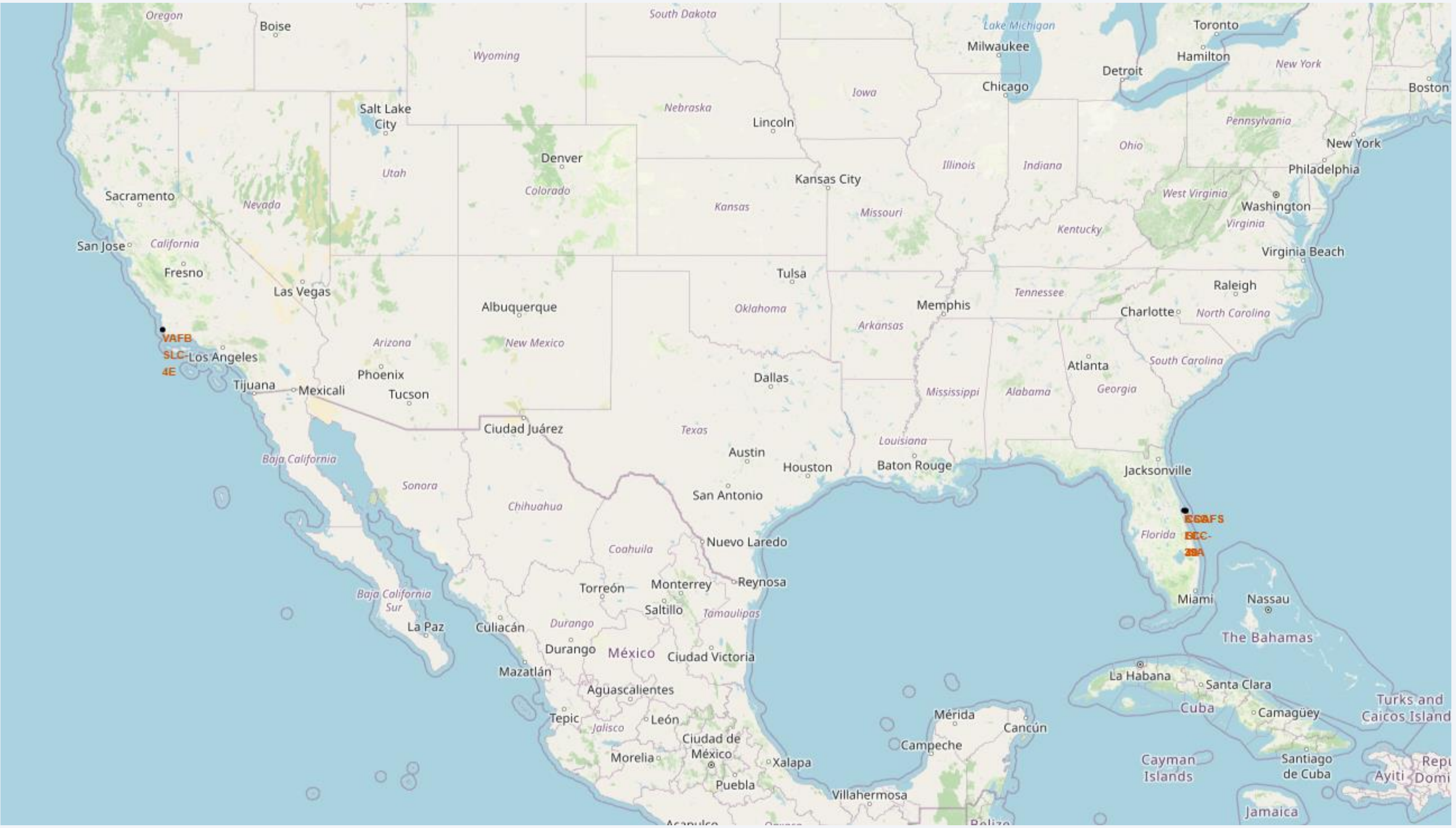
Landing_Outcome	count_outcomes
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 4

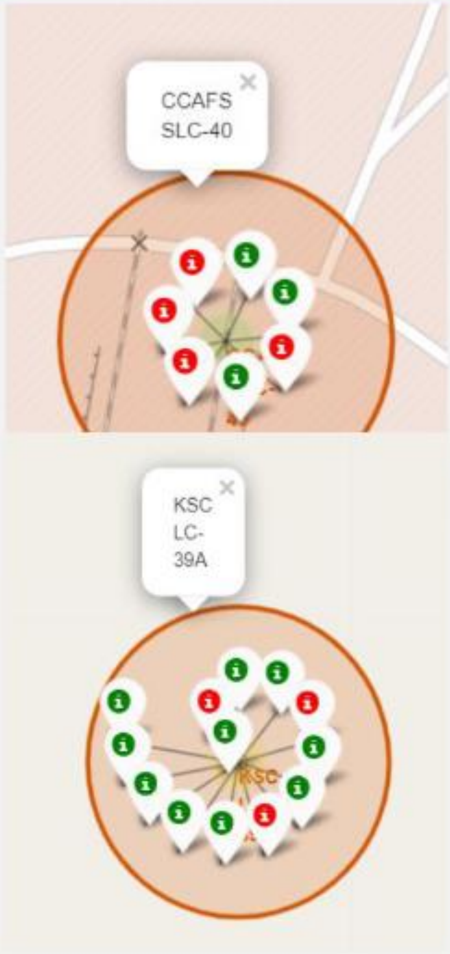
Launch Sites Proximities Analysis

All launch sites global map markers

- **Map of all sites**
  - We can see from the map that all sites are located on the east and west coasts of America



Markers showing launch sites with color labels



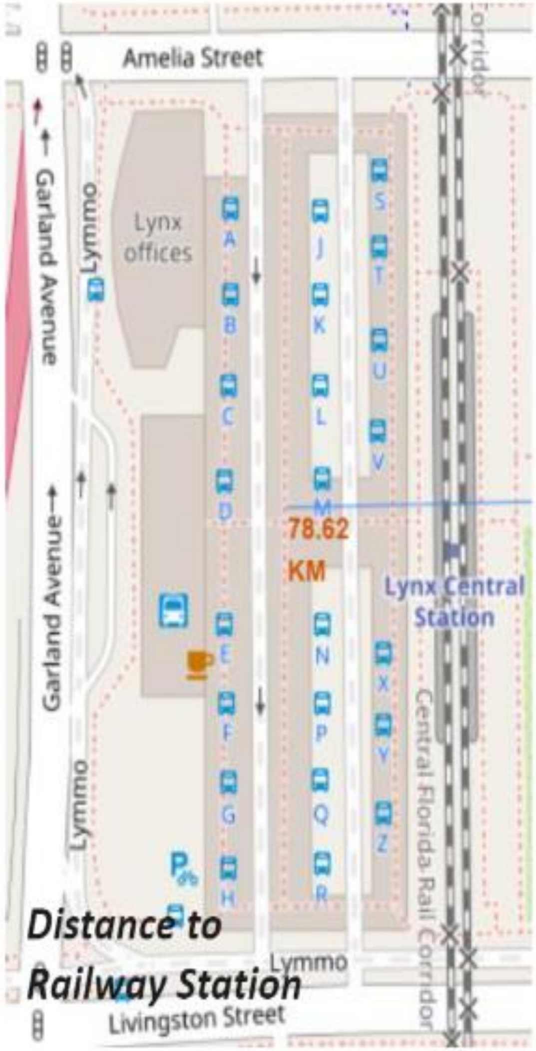
California Launch Site

37

Florida Launch Sites

Green Marker shows successful Launches and Red Marker shows Failures

Launch Site distance to landmarks



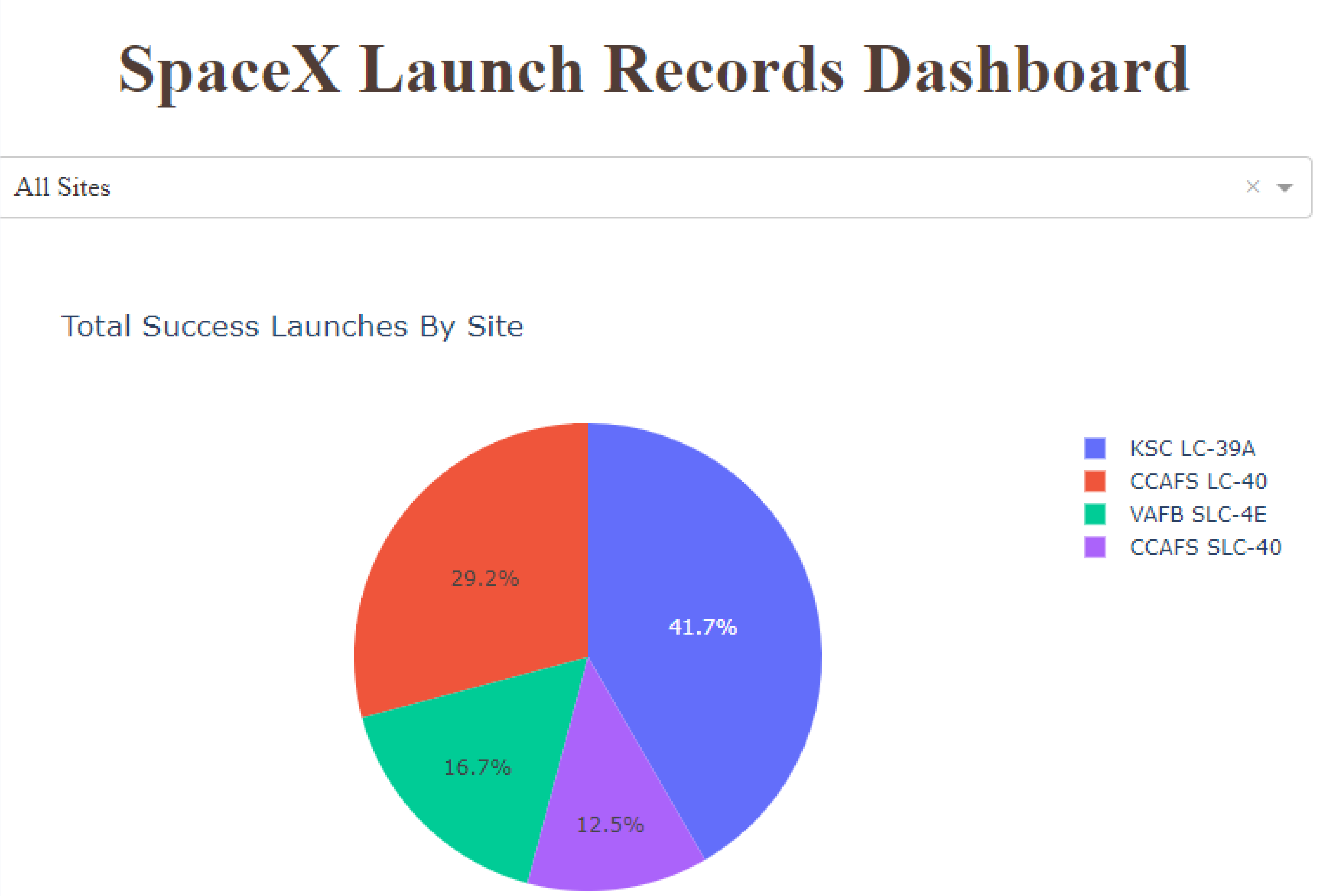
- Are launch sites in close proximity to railways? No
- Are launch sites in close proximity to highways? No
- Are launch sites in close proximity to coastline? Yes
- Do launch sites keep certain distance away from cities? Yes

Section 5

Build a Dashboard with Plotly Dash

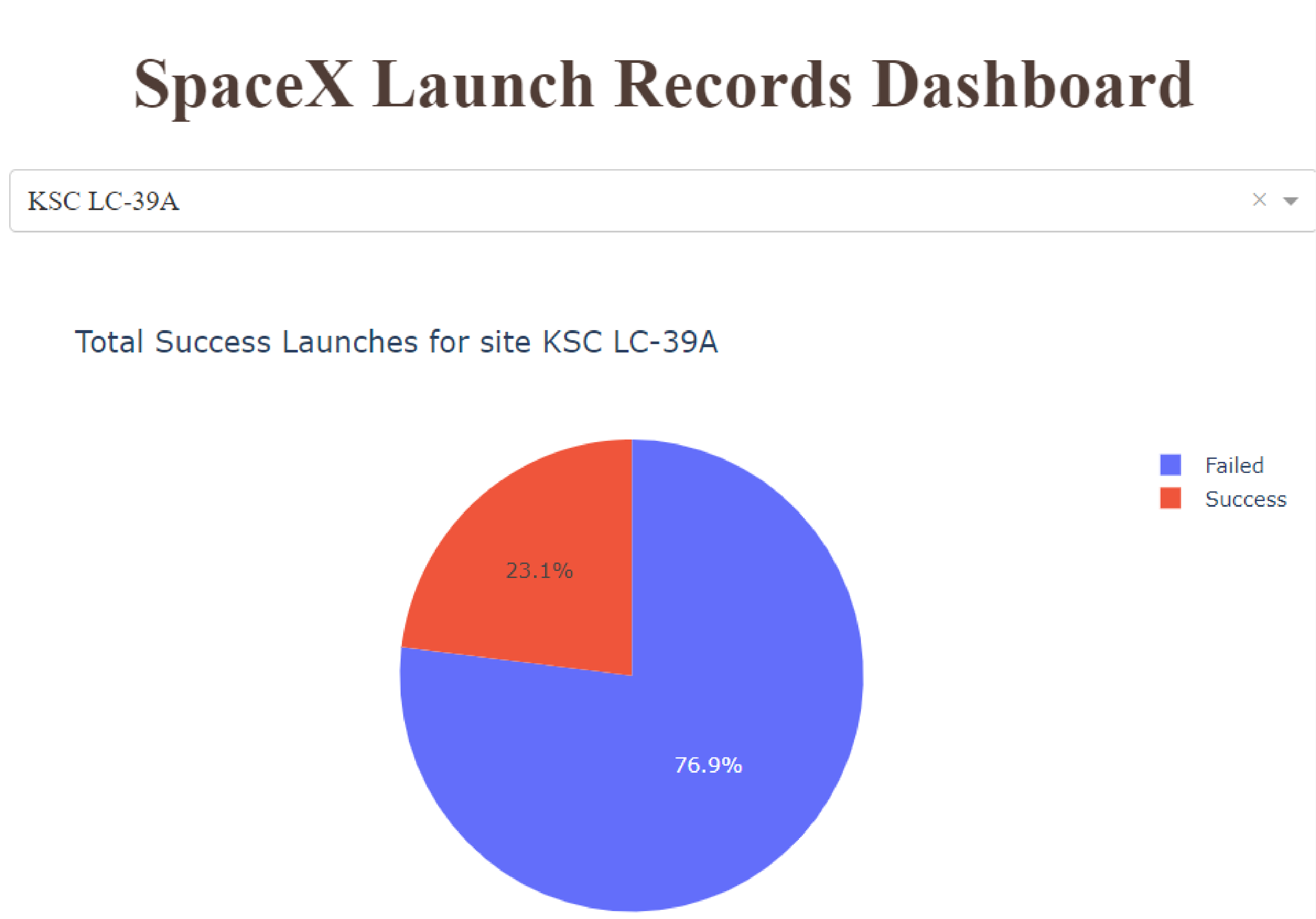
Launch site success rate

- Pie chart showing the success rate by percentage for each launch site.
- Data shows that KSC LC-39A has the highest success rate from the 4 sites and CCAFS SLC-40 has the lowest



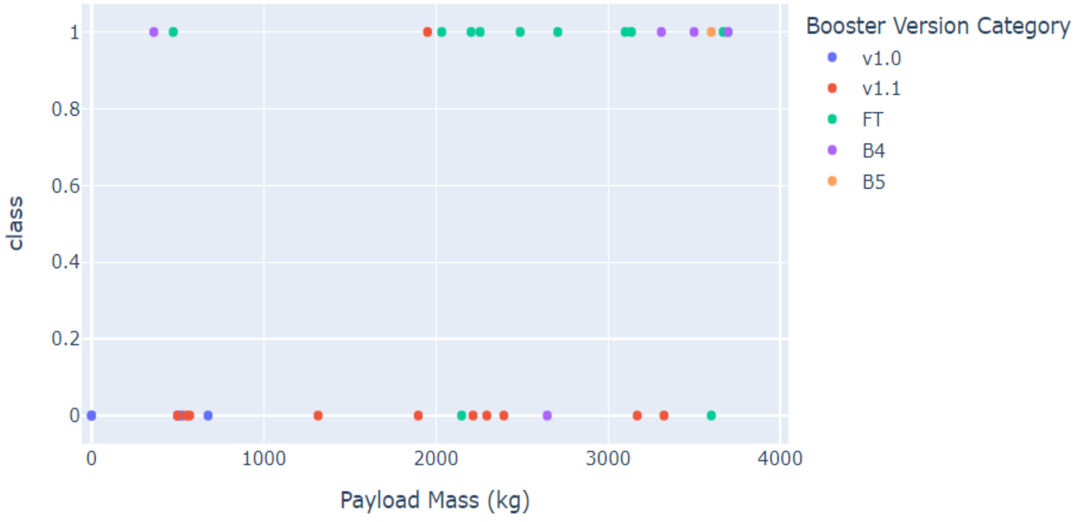
Launch site with highest success rate

- Pie chart showing the success rate for the most successful launch site.
- Data shows that KSC LC-39A has the success rate of 76.9% and a failure rate of 23.1%

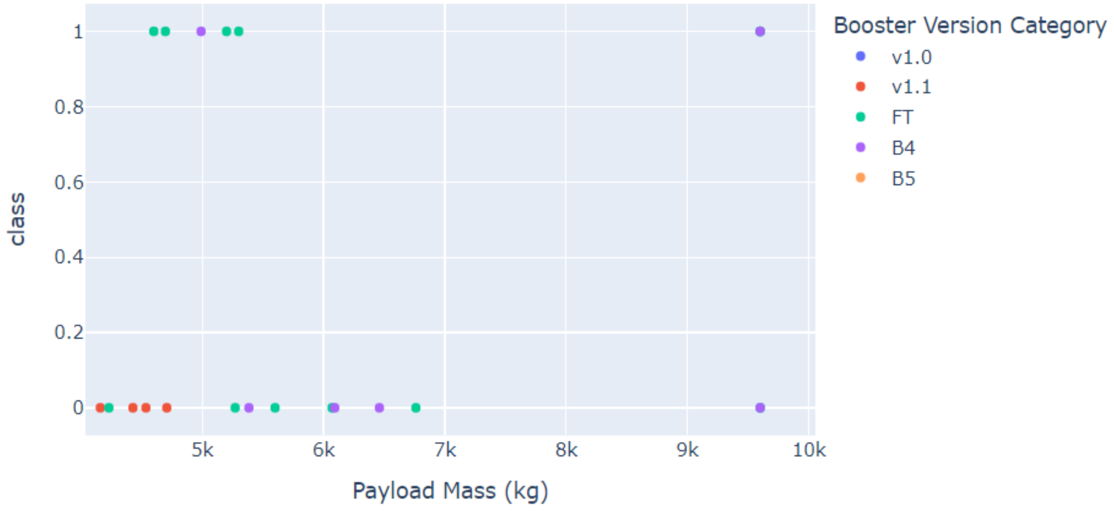


Scatter plot of Payload vs Launch Outcome for all sites, with different payload selected in the range slider

0-4,000(KG)



4000-10,000(KG)



- Here we have 2 scatter plots representing the Payload mass and success rate for all sites with different ranges using a range slider.

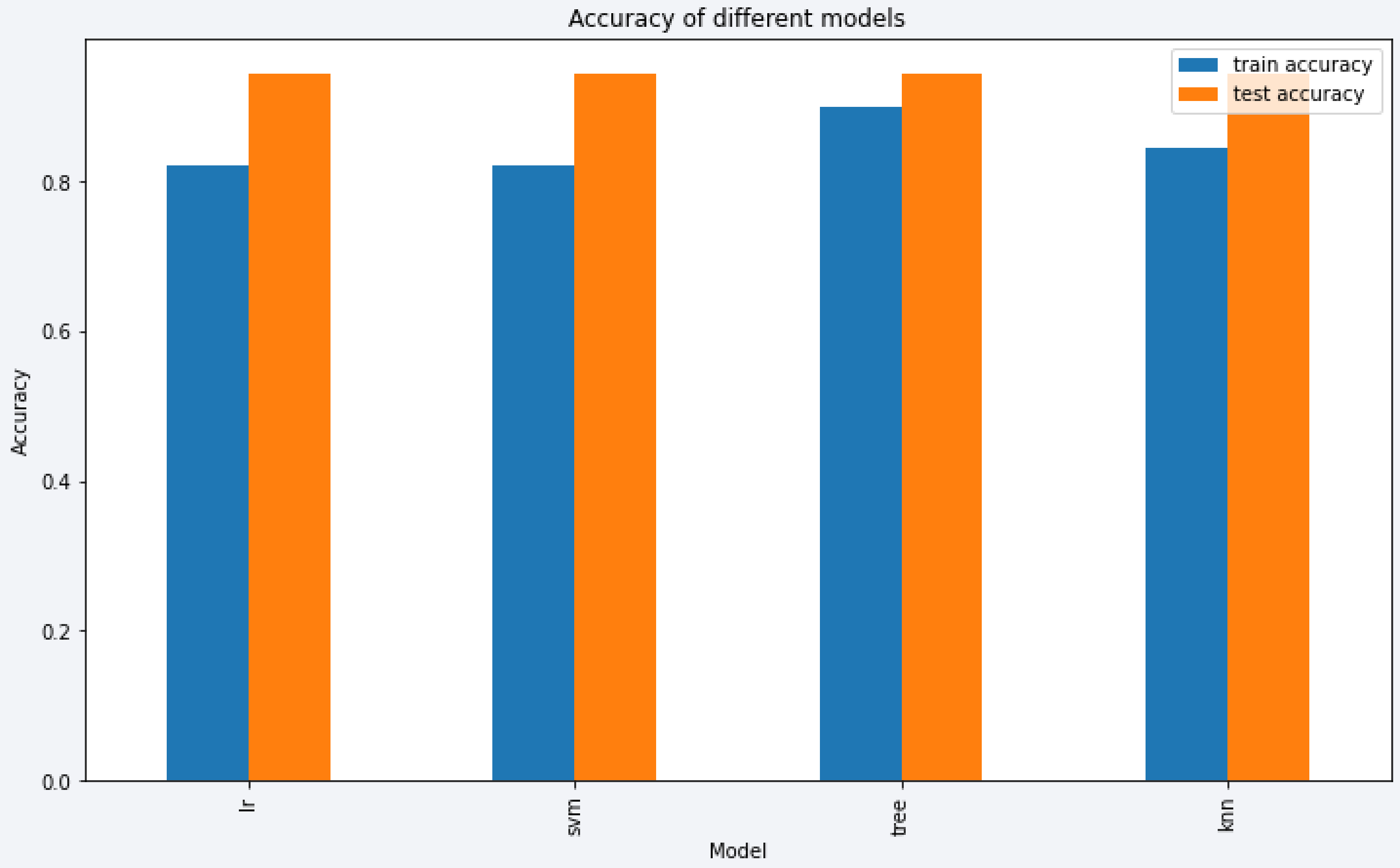
Section 6

Predictive Analysis (Classification)

Classification Accuracy

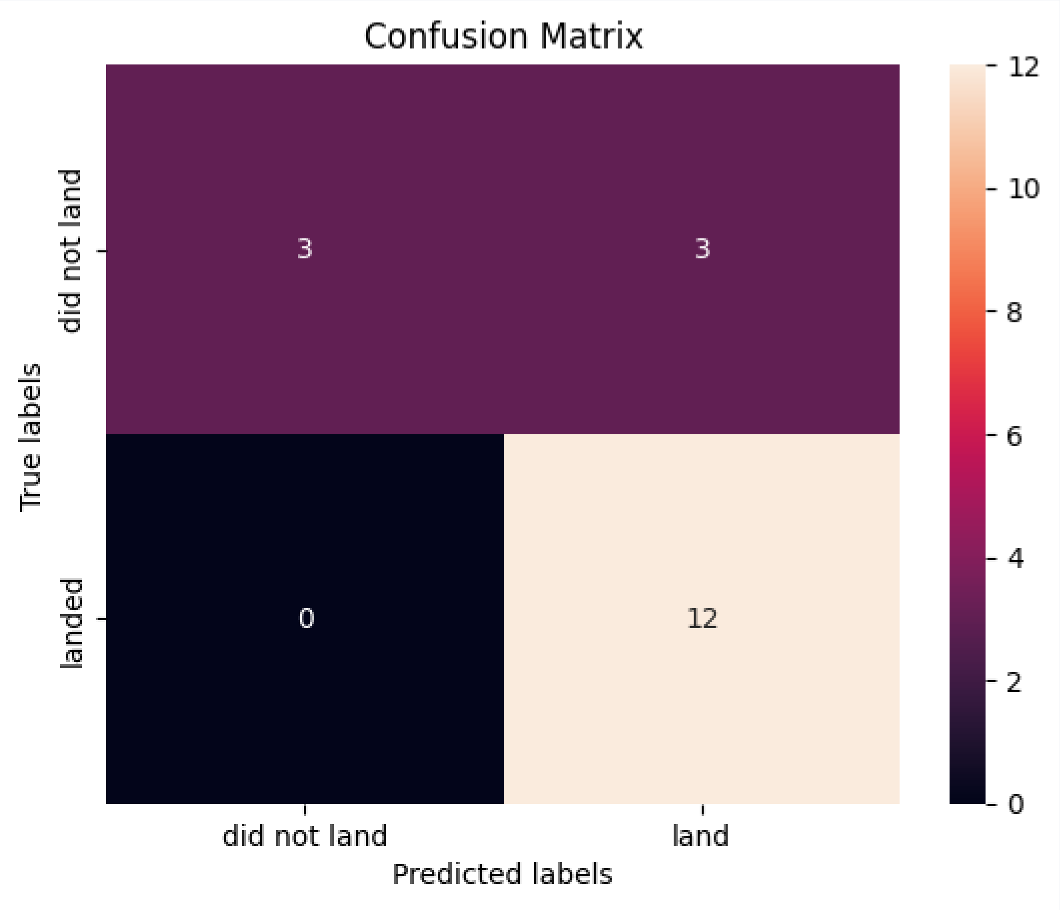
Explanation:

- Four classification models were tested, and their accuracies are plotted beside
- The model with the highest classification accuracy is Decision Tree Classifier, which has accuracies over than 87%



Confusion Matrix

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



Conclusions

We can conclude that:

- The larger the flight amount at a launch site, the greater the success rate at a launch site.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate.
- KSC LC-39A had the most successful launches of any sites.
- The Decision tree classifier is the best machine learning algorithm for this task.

Thank you!