IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Jakaria
25 January 2024

Outline

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

Executive Summary

Summary of methodologies

Our approach encompassed a dual methodology involving API integration and web scraping techniques for data collection. Following the acquisition phase, we employed a suite of Python data manipulation methods to meticulously process and cleanse the dataset. Subsequently, SQL queries were employed to extract pertinent information from the refined dataset. Early insights were garnered through systematic data visualization and trend analysis. Concluding our analytical framework, we implemented supervised machine learning models to formulate predictions regarding the success of landing events. We applied supervised machine learning models to make predictions about the success of the landing event.

Summary of all results

Through meticulous data analysis, we identified discernible patterns and correlations among variables directly influencing the success of landing events. Leveraging these insights, we developed and trained a predictive model that demonstrated a notable capability to accurately forecast the probability of a successful landing event. Notably, the model achieved a commendable accuracy rate of 83%, underscoring its effectiveness in providing reliable prognostications within this domain.

Introduction

- SpaceX's commitment to reusable rockets has significantly mitigated space travel costs by strategically focusing on the retrieval of the first rocket phase. The recovery of this initial phase is paramount in preserving and reusing expensive components, contributing directly to cost reduction. An in-depth analysis of the success rate of these retrieval events serves as a valuable metric for evaluating efficiency and cost-effectiveness in SpaceX's pioneering approach. This particular project is geared towards predicting the success of the first phase retrieval event, thereby offering predictive insights aimed at enhancing decision-making within the space industry.
- Our objective is to forecast the success of first-phase rocket retrieval, with the overarching aim of optimizing resource allocation. By achieving this predictive capability, we seek to enhance mission success rates and contribute to substantial cost savings.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

Describe how data sets were collected:

Data was first collected using SpaceX API (a RESTful API) by making a get request to theSpaceX API. This was done by first defining a series helper functions that would help in the use of the API to extract information using identification numbers in the launch data and then requesting rocket launch data from the SpaceX API url.

Finally to make the requested JSON results more consistent, the SpaceX launch data was requested and parsed using the GET request and then decoded the response content as aJson result which was then converted into a Pandas data frame.

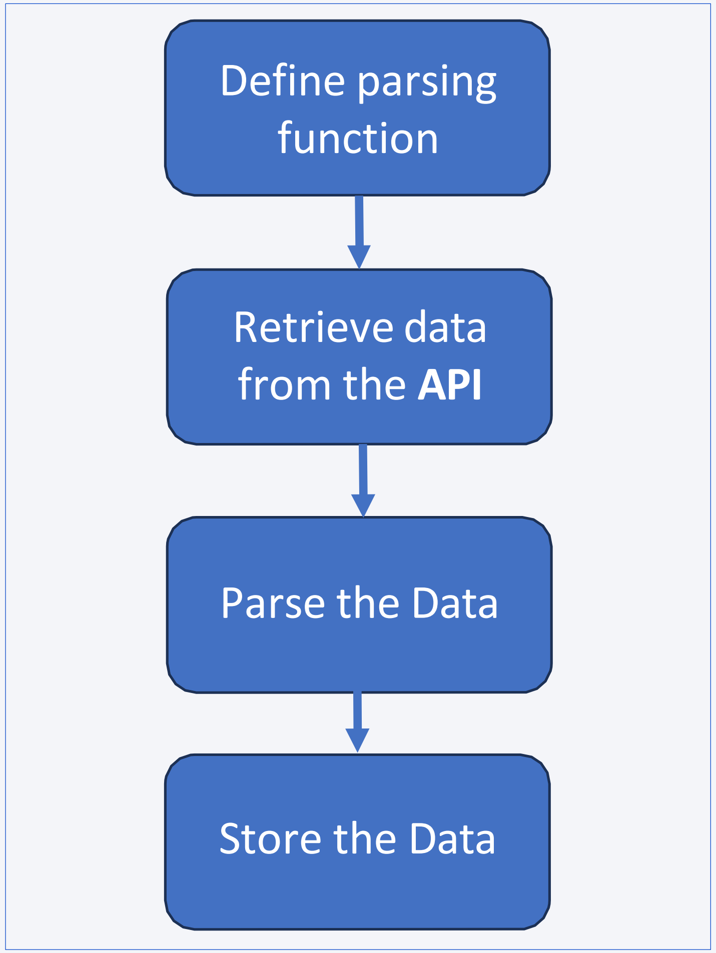
Also performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches of the launch records are stored in aHTML. Using BeautifulSoup and request Libraries, I extract the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas DataFrame.

Data Collection – SpaceX API

1) Define auxiliary function to parse the data.
2) Retrieve data from the **REST API** using the method **GET.**
3) Parse the data with the previously built auxiliary functions.
4) Store the data in **PANDAS DataFrame**.

GitHub URL of the completed SpaceX API calls notebook:

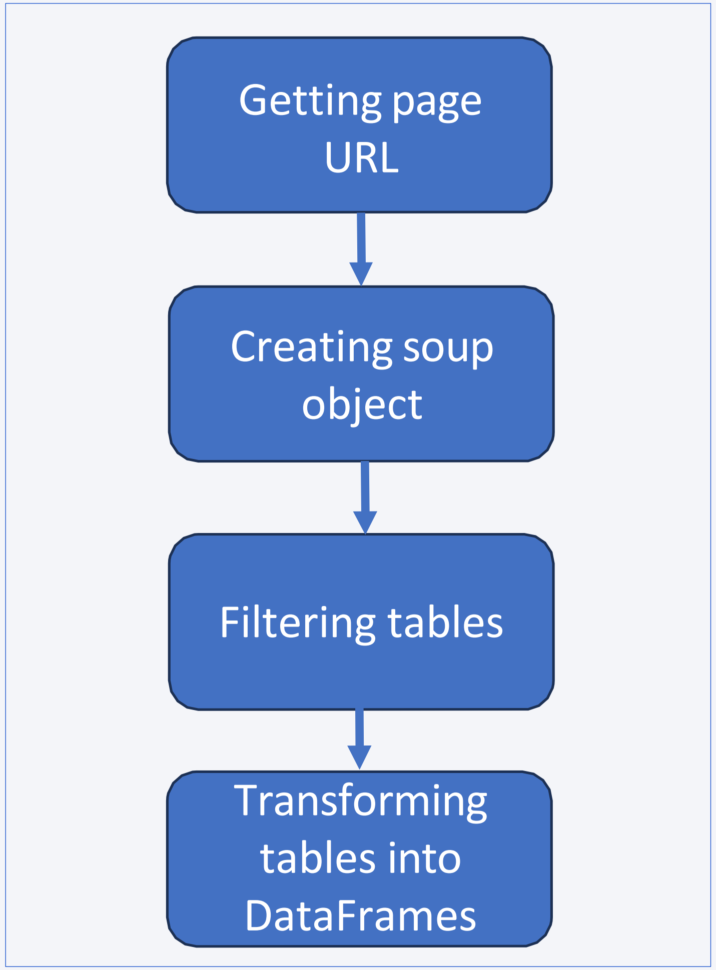
(Jupyter Labs SpaceX Data Collection API)



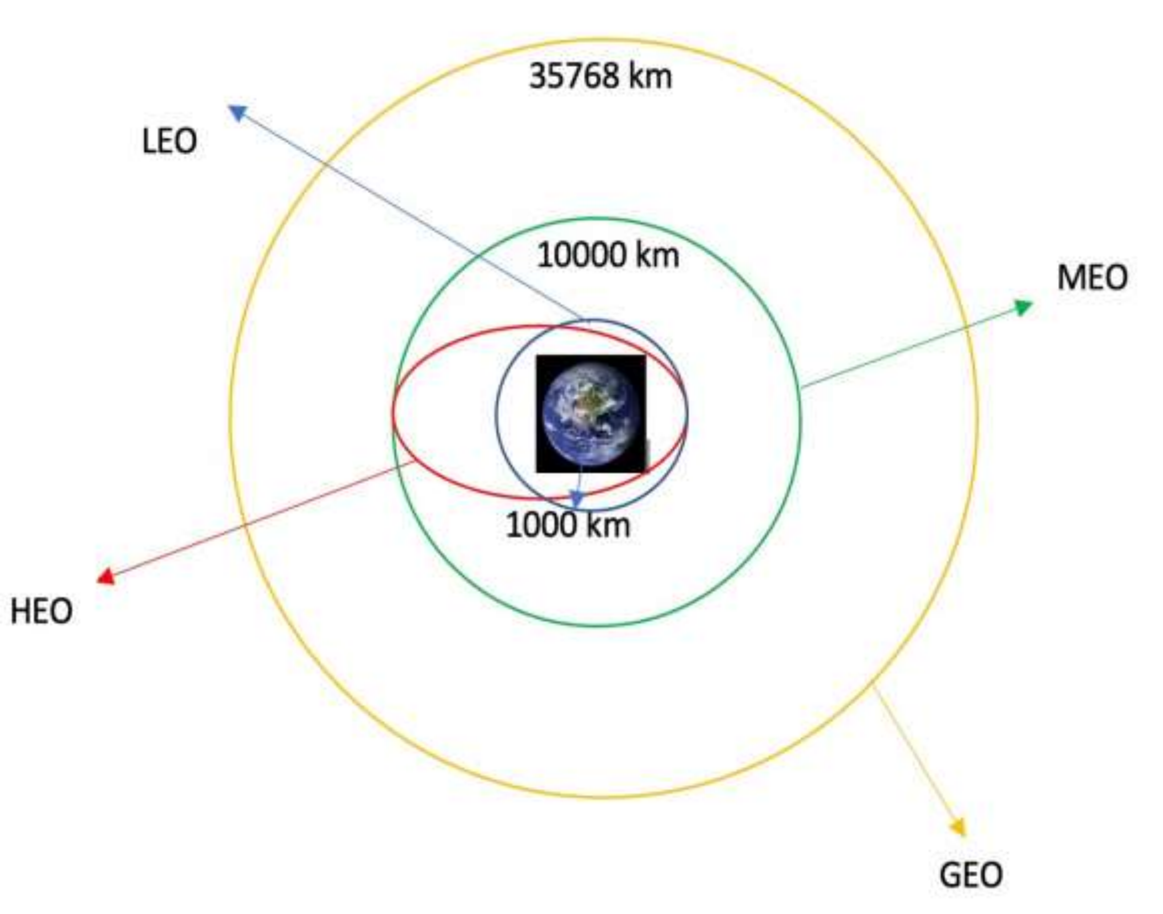
Data Collection - Scraping

1) Using the get request.get method to download page code.
2) Created a BeautifulSoup object to manipulate the html text.
3) Filtered the desired tables using soup manipulation methods.
4) Converted the data from the HTML to pandas **DataFrame** format

GitHub URL of the completed web scraping notebook:
[Jupyter Labs Webscraping]



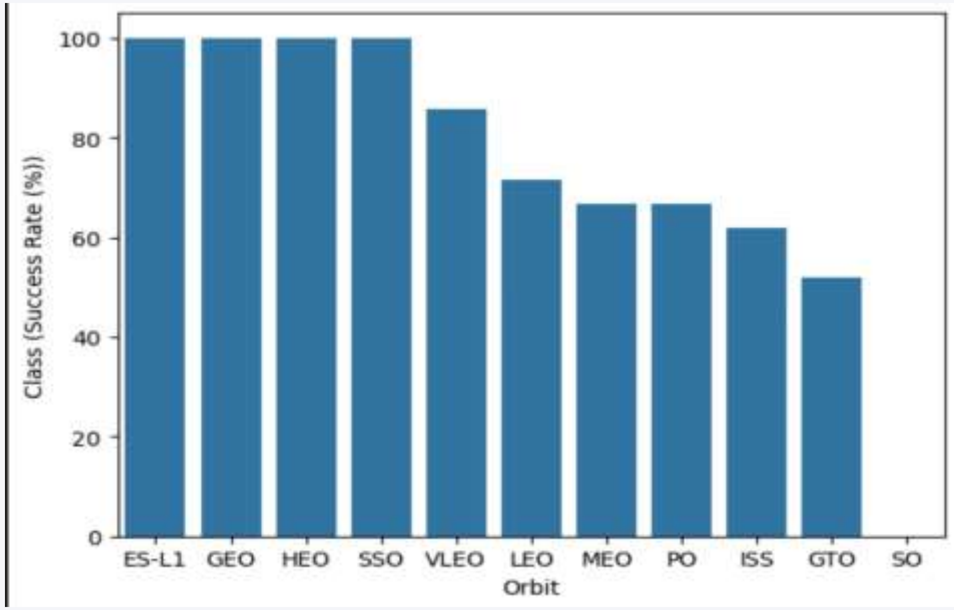
Data Wrangling



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**.

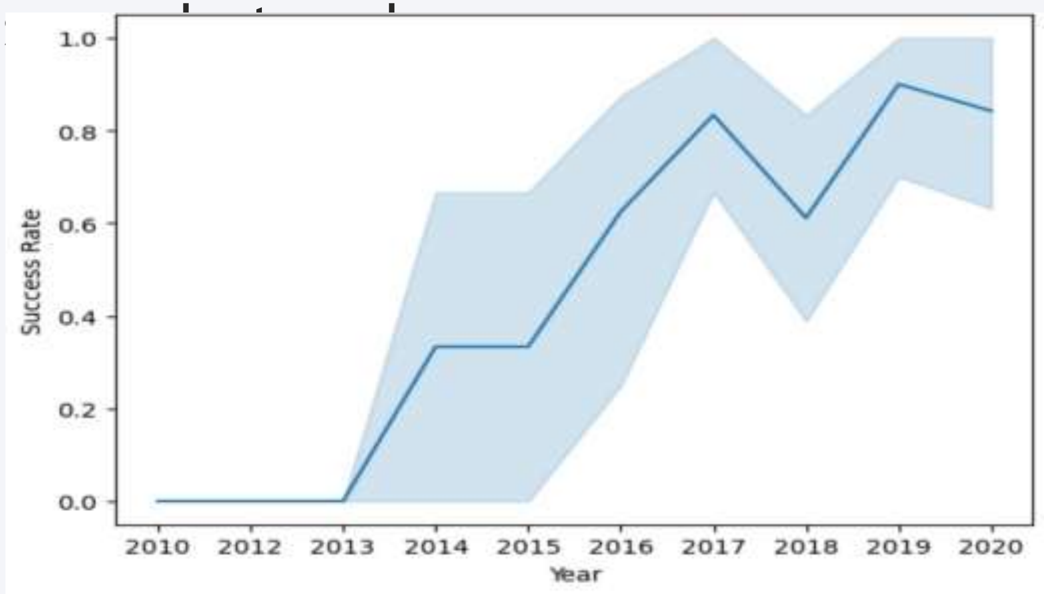
- For Girhub Notebook File [Click Here]()

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 succes

- For Github Notebook File [Click Here]



EDA with SQL

Using SQL, we had performed many queries to get better understanding of the dataset

- Displaying the names of the launch sites.
- Displaying 5 records where launch sites begin with the string 'CCA'.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.- Rank the count of landing outcomes or success between the date 2010-06-04 and2017-03-20, in descending order.

For Github Notebook File [Click Here]

Build an Interactive Map with Folium

We marked all launch sites, and added map objects such as markers, circles, lines to mark the success or failure of launches for each site on the folium map.

We assigned the feature launch outcomes (failure or success) to class 0 and 1.i.e., 0for failure, and 1 for success.

Using the color-labeled marker clusters, we identified which launch sites have relatively high success rate.

We calculated the distances between a launch site to its proximities. We answered some question for instance:

- Are launch sites near railways, highways and coastlines.
- Do launch sites keep certain distance away from cities.

For Github Notebook File Click Here

Build a Dashboard with Plotly Dash

We built an interactive dashboard with Plotly dash.

We plotted pie charts showing the total launches by a certain sites.

We plotted scatter graph showing the relationship with Outcome and PayloadMass (Kg) for the different booster version.

For Github Notebook File Click Here

Predictive Analysis (Classification)

We loaded the data using numpy and pandas, transformed the data, split our data into training and testing.

We built different machine learning models and tune different hyperparameters using GridSearchCV.

We used accuracy as the metric for our model, improved the model using feature engineering and algorithm tuning.

We found the best performing classification model.

For Github Notebook File Click Here

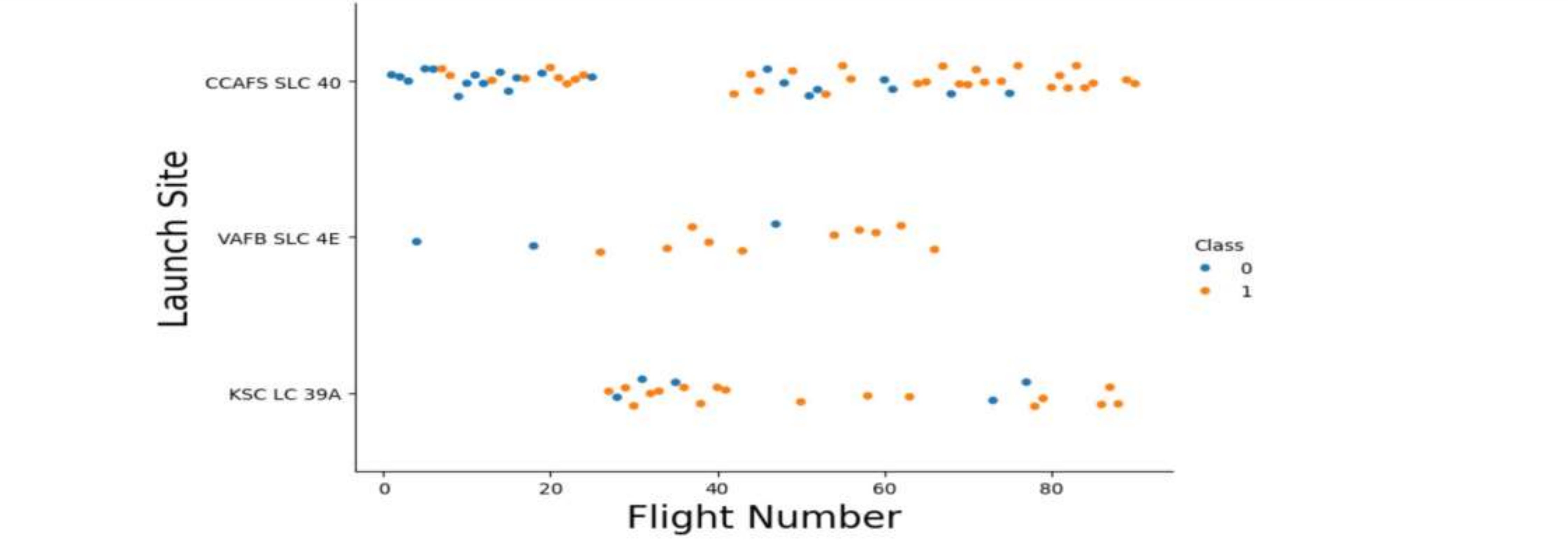
Results

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

Section 2

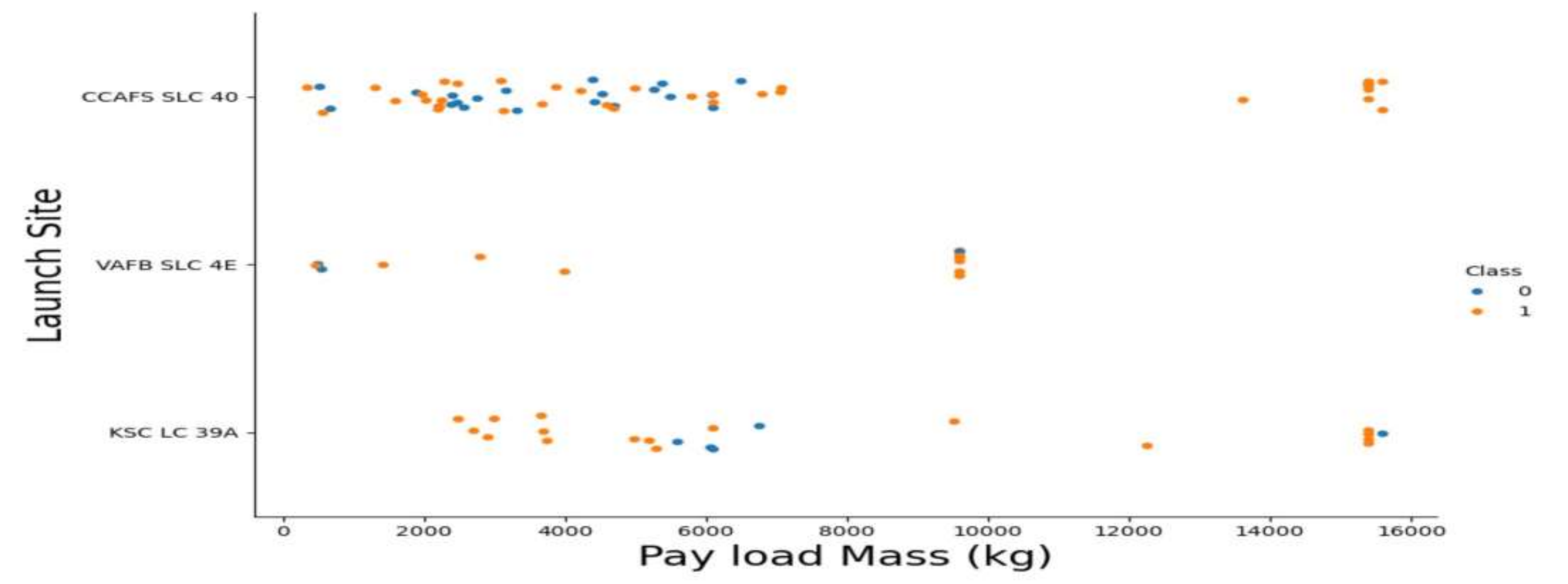
Insights drawn from EDA

Flight Number vs. Launch Site



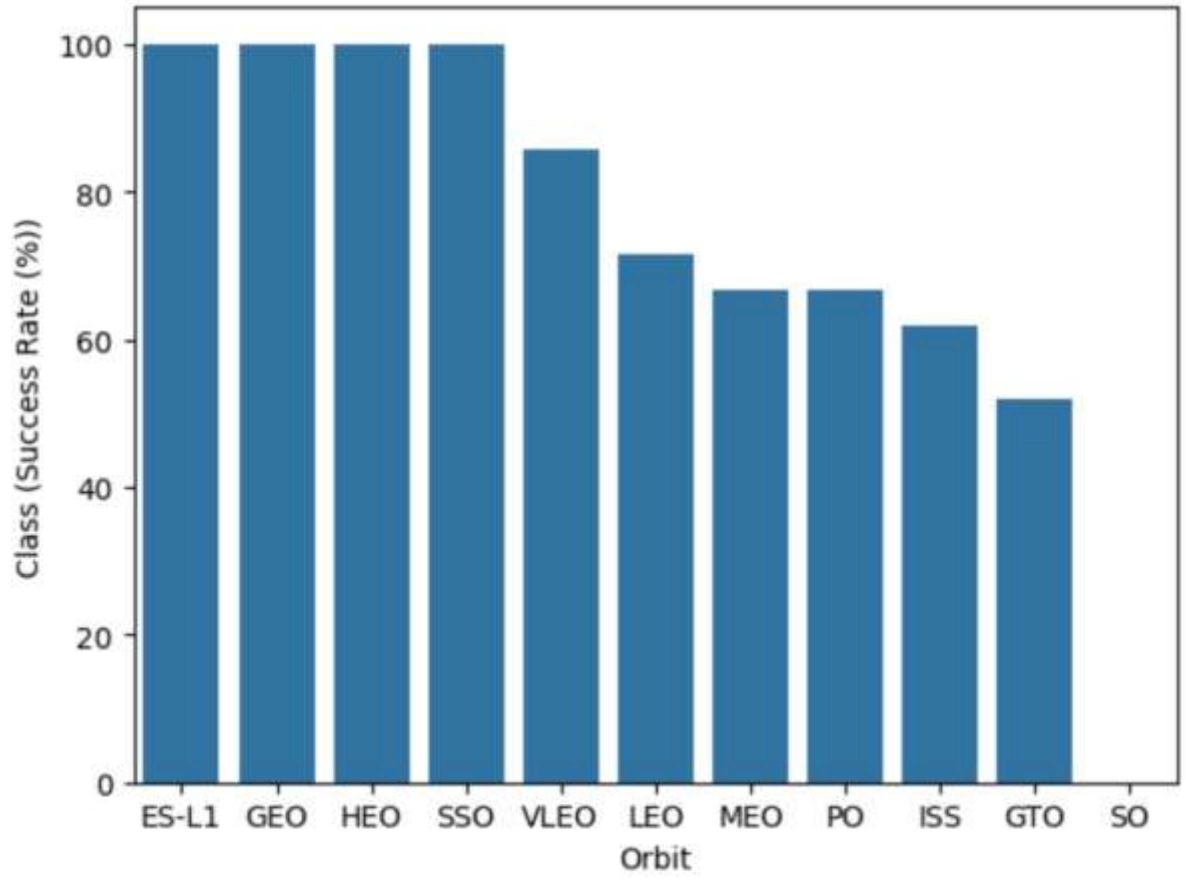
From the plot, we found that the larger the flight amount at a launch site, the greater the success rate at a launch site.

Payload vs. Launch Site



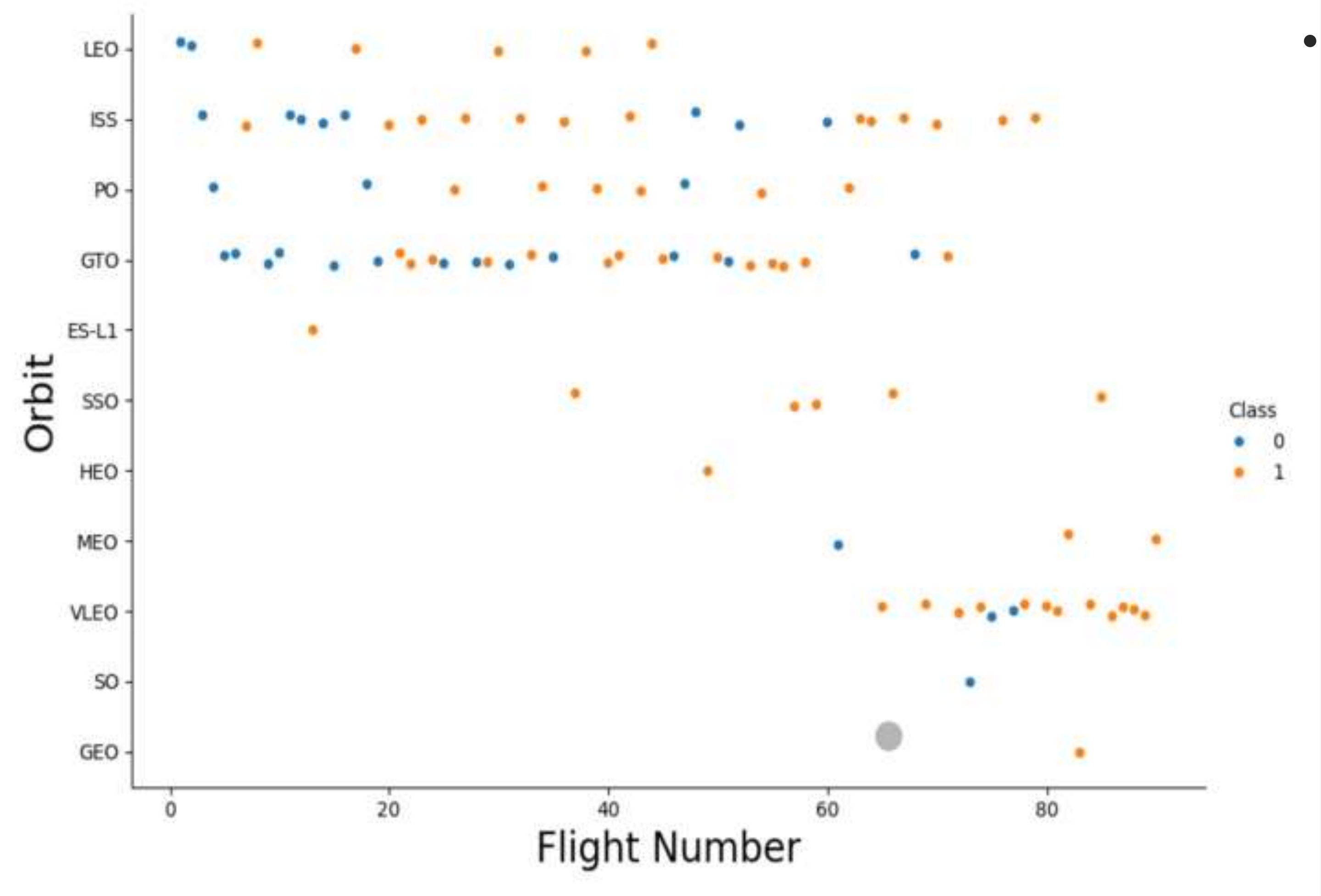
The greater the payload mass for launch site CCAFS SLC 40 the higher the success rate for the rocket.

Success Rate vs. Orbit Type



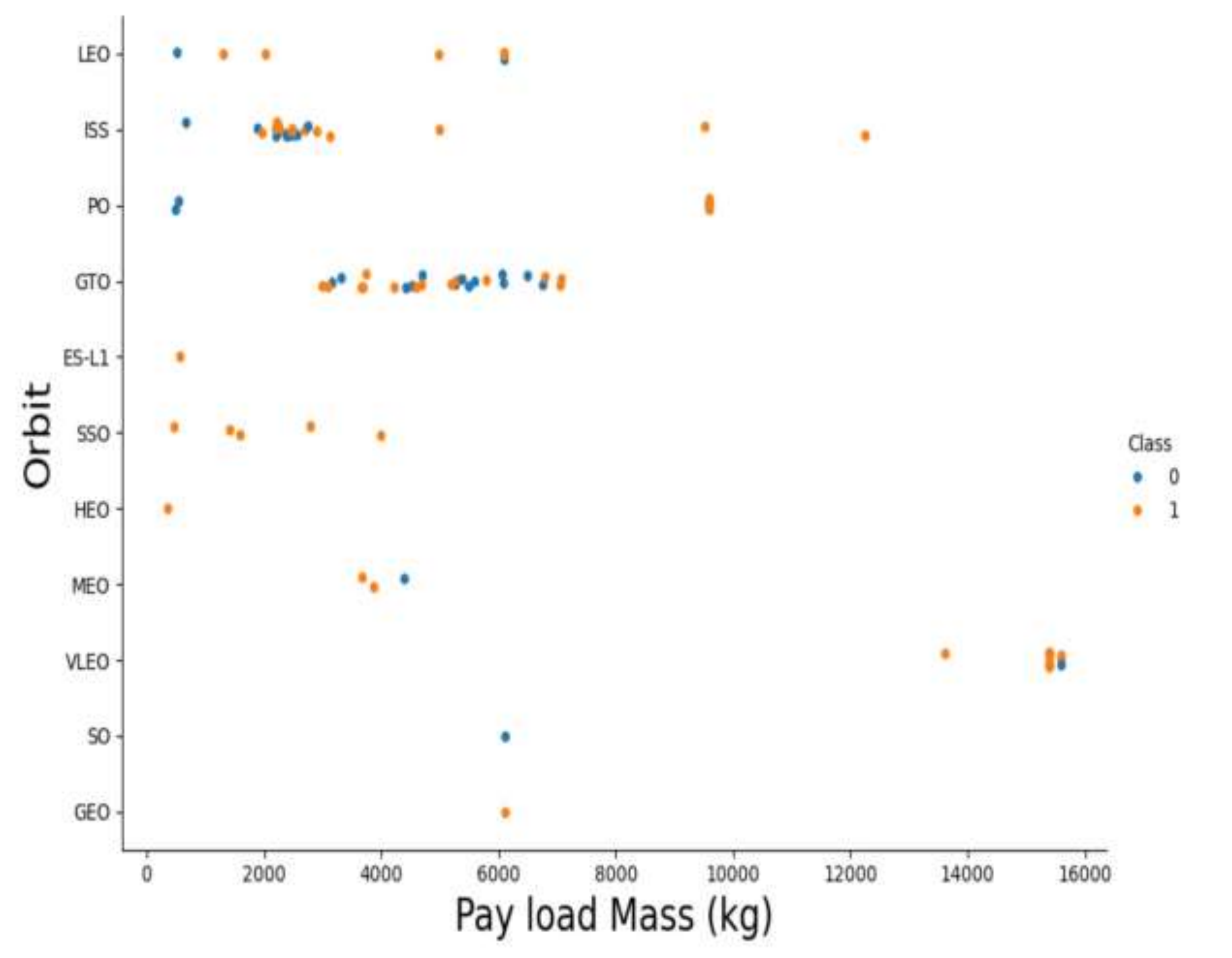
- From the plot, we can see that ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

Flight Number vs. Orbit Type



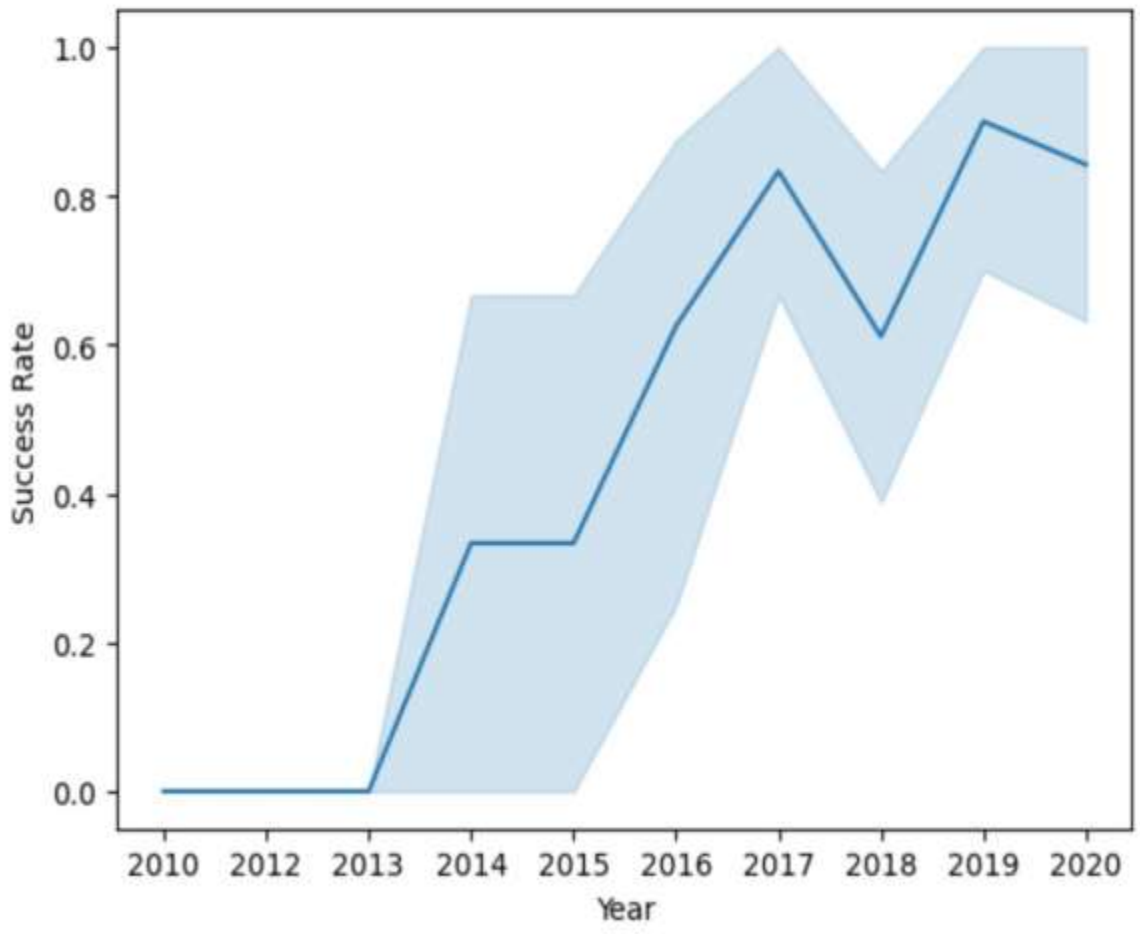
- The plot below shows the Flight Number vs. Orbit type. We observe that in the LEO orbit, success is related to the number of flights whereas in the GTO orbit, there is no relationship between flight number and the orbit.

Payload vs. Orbit Type



- We can observe that with heavy payloads, the successful landing are more for PO,LEO and ISS orbits.

Launch Success Yearly Trend



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

All Launch Site Names

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

In [31]:
```
%sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
```

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

Out[31]:

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

We used the key word DISTINCT to show only unique launch sites from the SpaceX data.

Launch Site Names Begin with 'CCA'

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

```
%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	Landin _Outcom
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failu (parachut
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	Failu (parachut
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attem
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attem
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attem

We used the query above to display 5 records where launch sites begin with CCA.

Total Payload Mass

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

In [17]:
```
%sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA
```

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

Out[17]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

We calculated the total payload carried by boosters from NASA as 45596 using the query below

Average Payload Mass by F9 v1.1

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

In [19]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booste
```

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

Out[19]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

We calculated the average payload mass carried by booster version F9 v1.1 B1003 as 2534.66666666666.

First Successful Ground Landing Date

Task 5

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

Hint:Use min function

In [21]:
```
%sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
```

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

Out[21]:

MIN(DATE)
01-05-2017

We observed that the dates of the first successful landing outcome on ground pad was **1st May 2017.**

Successful Drone Ship Landing with Payload between 4000 and 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 [26]:
```
# %sql SELECT * FROM 'SPACEXTBL'
```

In [27]:
```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AN
```

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

Out[27]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

We used the WHERE clause to filter for boosters which have success fully landed on drone-ship and applied the and condition to determine successful landing with payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

List the total number of successful and failure mission outcomes

In [28]:
```
%sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
```

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

Out[28]:

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

Total number of Mission outcome was a success or a failure.

Boosters Carried Maximum Payload

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

In [30]:
```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX
```

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

Out[30]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

- We determined the booster that have carried the maximum payload using a subquery in the WHERE clause and the MAX() function.

2015 Launch Records

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.

In [68]:
```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_",
```

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

Out[68]:

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)

We used a combinations of the WHERE clause, LIKE, AND, and Between conditions to filter for failed landing outcomes in drone ship, their booster versions, and launch site names for year 2015.

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

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 [45]:

```
%sql SELECT LANDING_OUTCOME, COUNT(*) AS COUNT_LAUNCHES FROM SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-
```

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

Out[45]:

Landing_Outcome	COUNT_LAUNCHES
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

- We selected Landing outcome sand 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 outcomes in descending order.

Section 3

Launch Sites Proximities Analysis

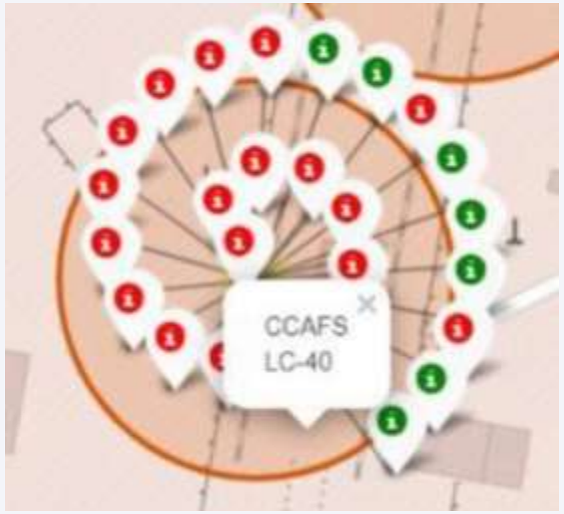
All launch sites global map markers

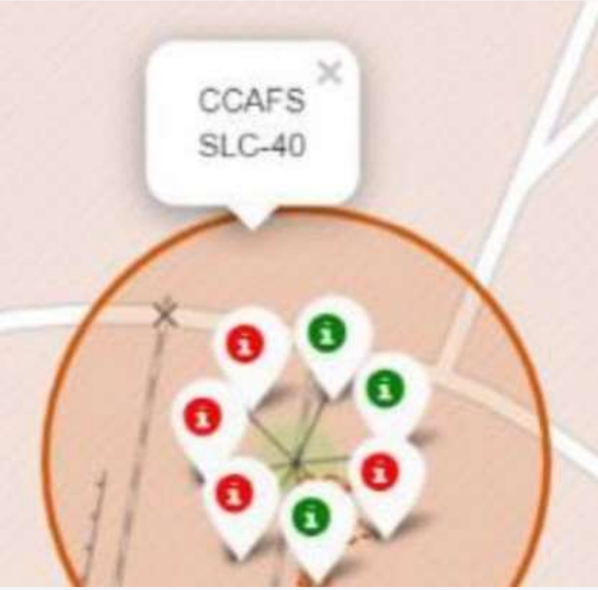
We can see that the Space launch sites are in the United States of America coasts. Florida and California

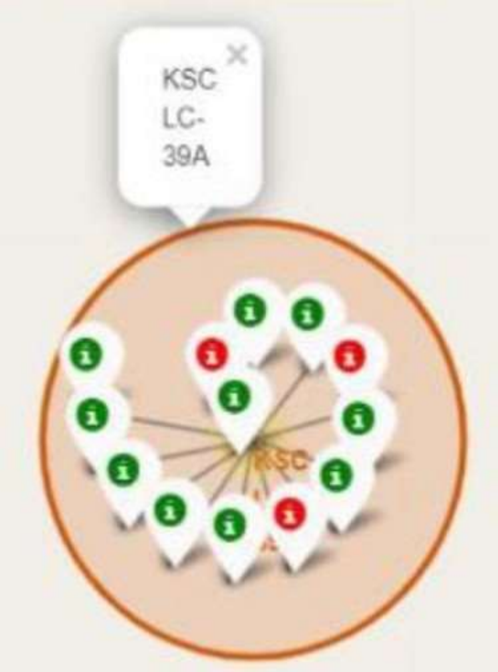
Markers showing launch sites with color labels

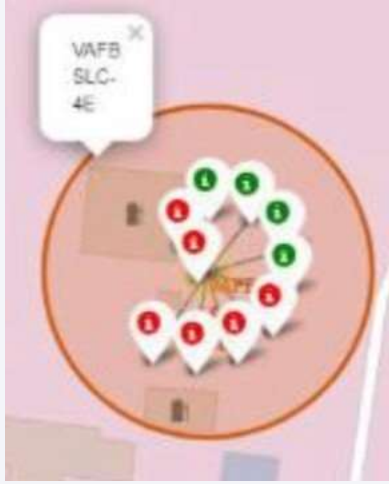
Green marker showing **Successful** Launches.

Red marker showing **Unsuccessful** Launches.









Launch Site distance to landmarks

Are lunch sites is close proximity to railways?

No

Are lunch sites is close proximity to highway?

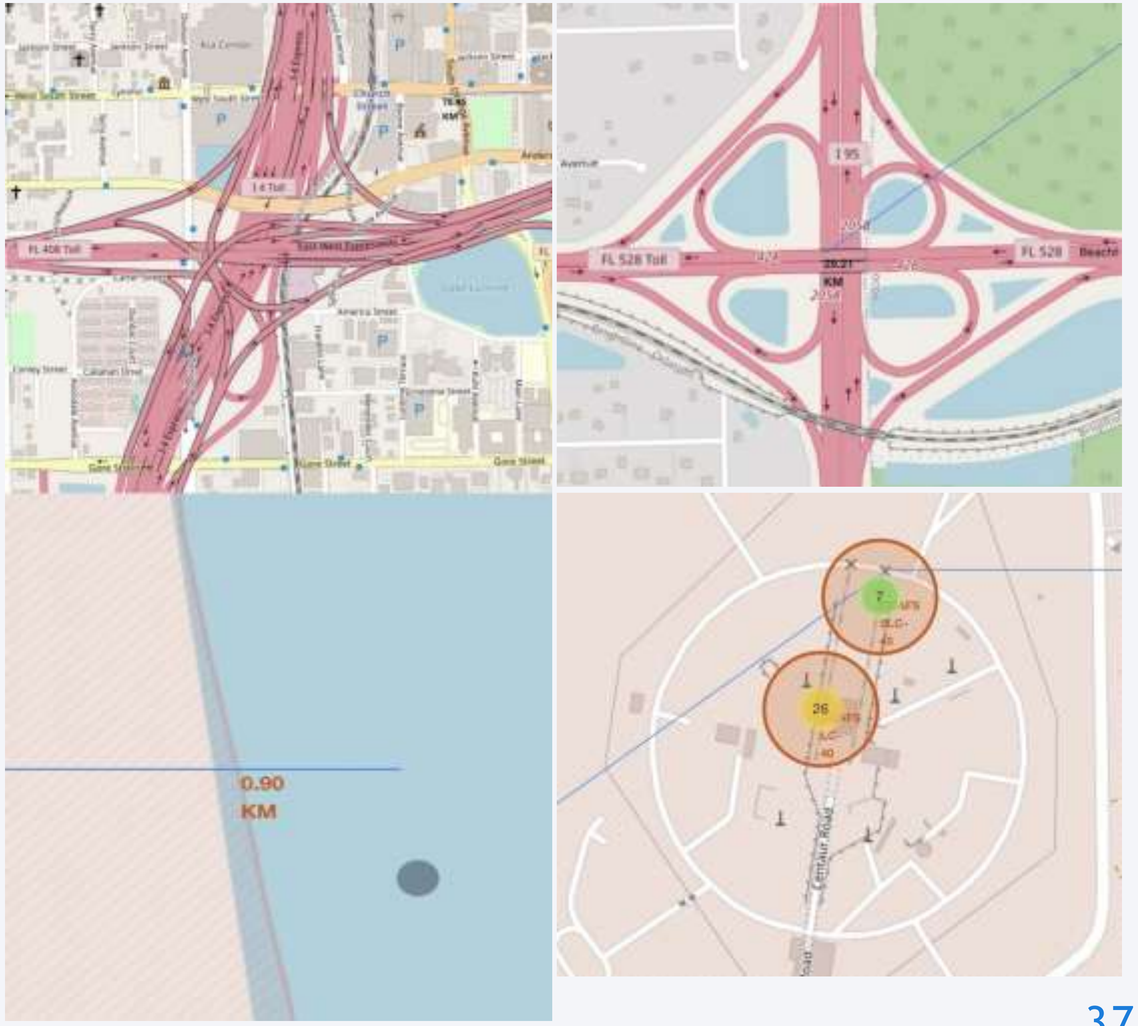
No

Are lunch sites is close proximity to coastline?

Yes

Do launch sites keep certain distance away from cities?

Yes

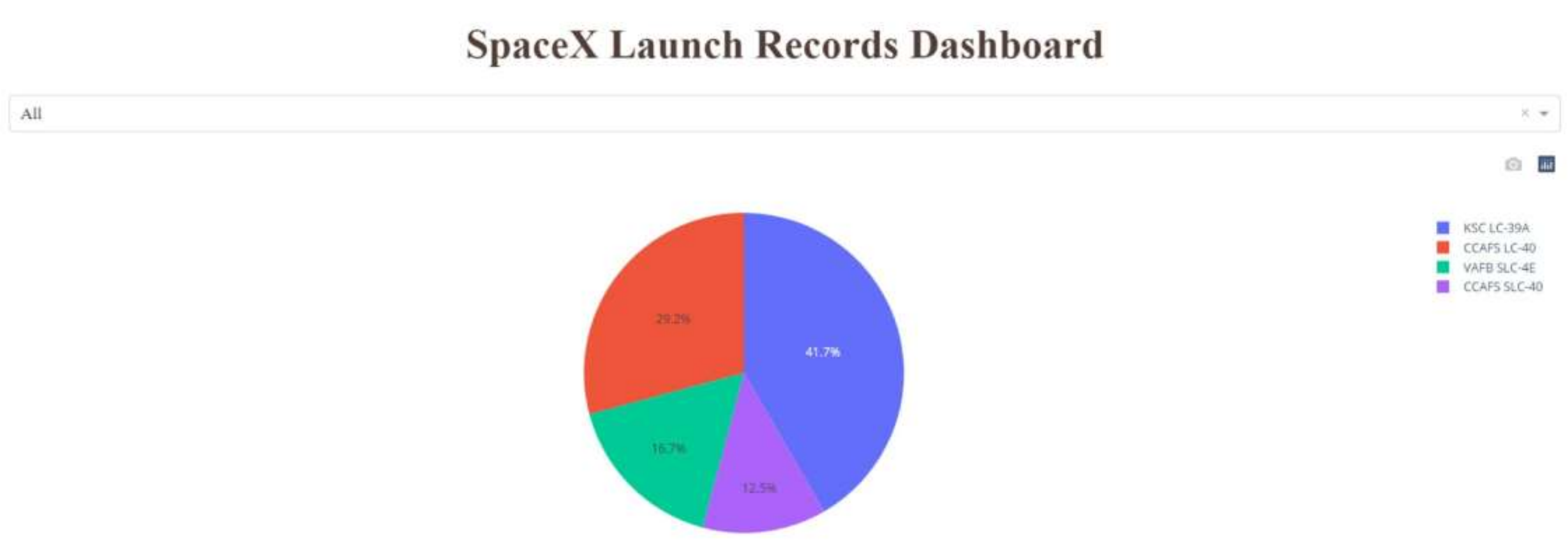


Section 4

Build a Dashboard with Plotly Dash

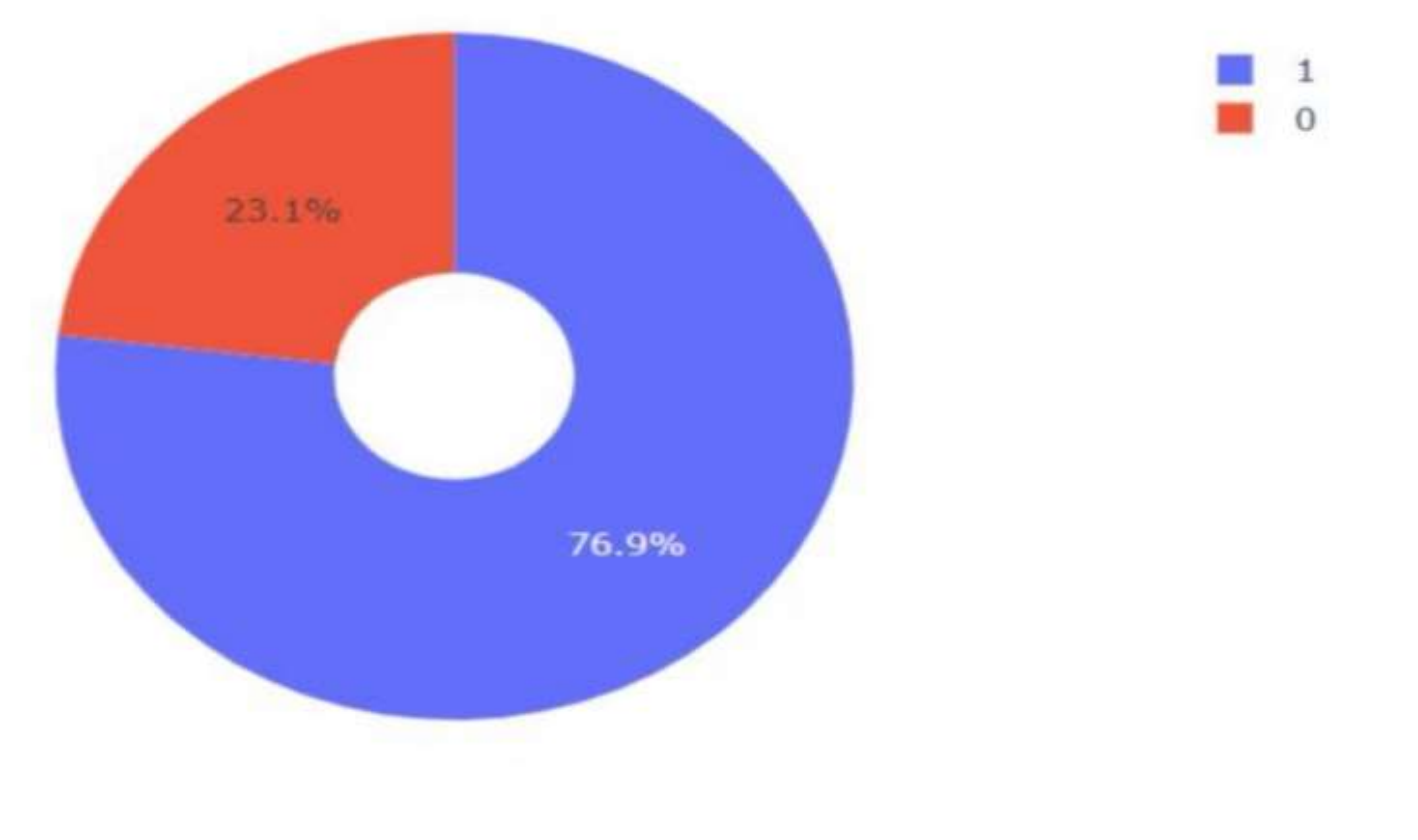
Pie chart showing the success percentage achieved by each launch site

Total success launches by all sites



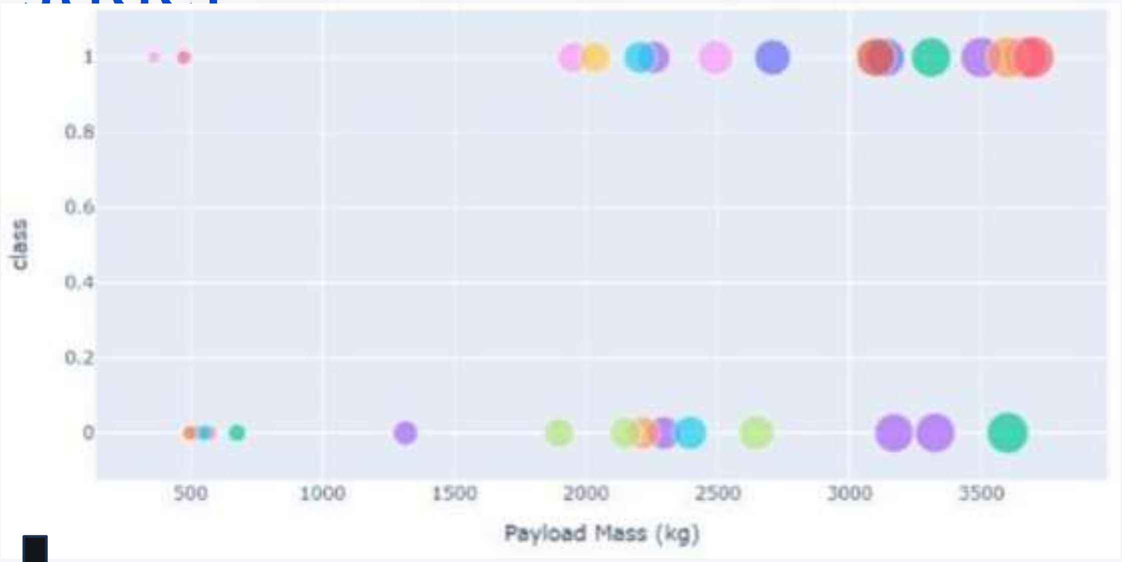
We can see that KSC LC-39A had the most successful launches from all the sities.

Pie chart showing the Launch site with the highest launch success ratio

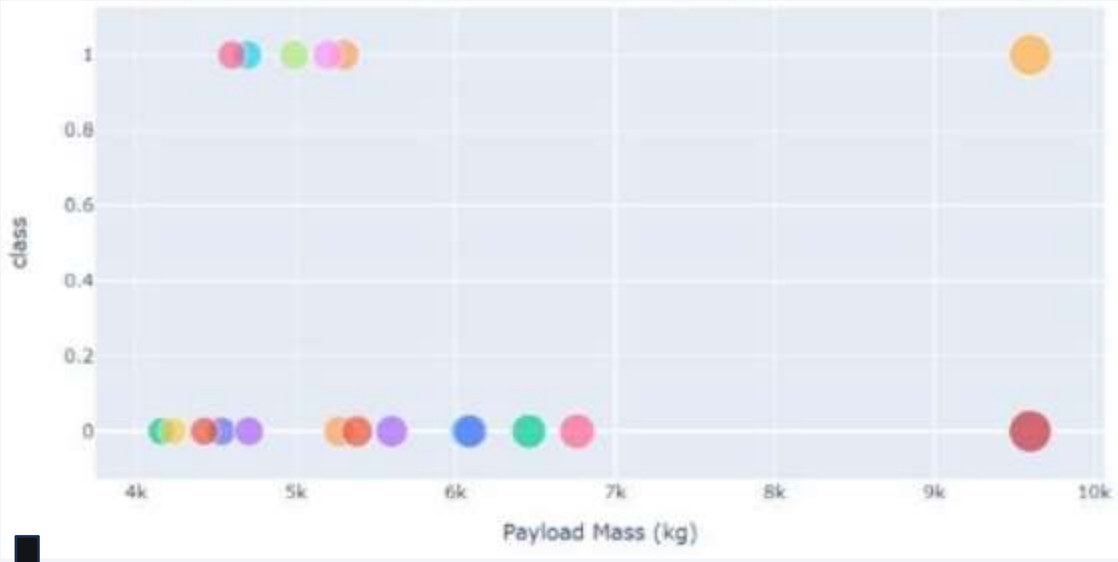


KSC LC-39A achieved a 76.9% success rate while getting a 23.1% failure rate

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



Low Weighted Payload 0kg - 4000kg



Heavy Weighted Payload 4000kg - 10000kg

We can see the success rates for low weighted payloads is higher than the heavy weighted payloads.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

- The decision tree classifier is the model with the highest classification accuracy.

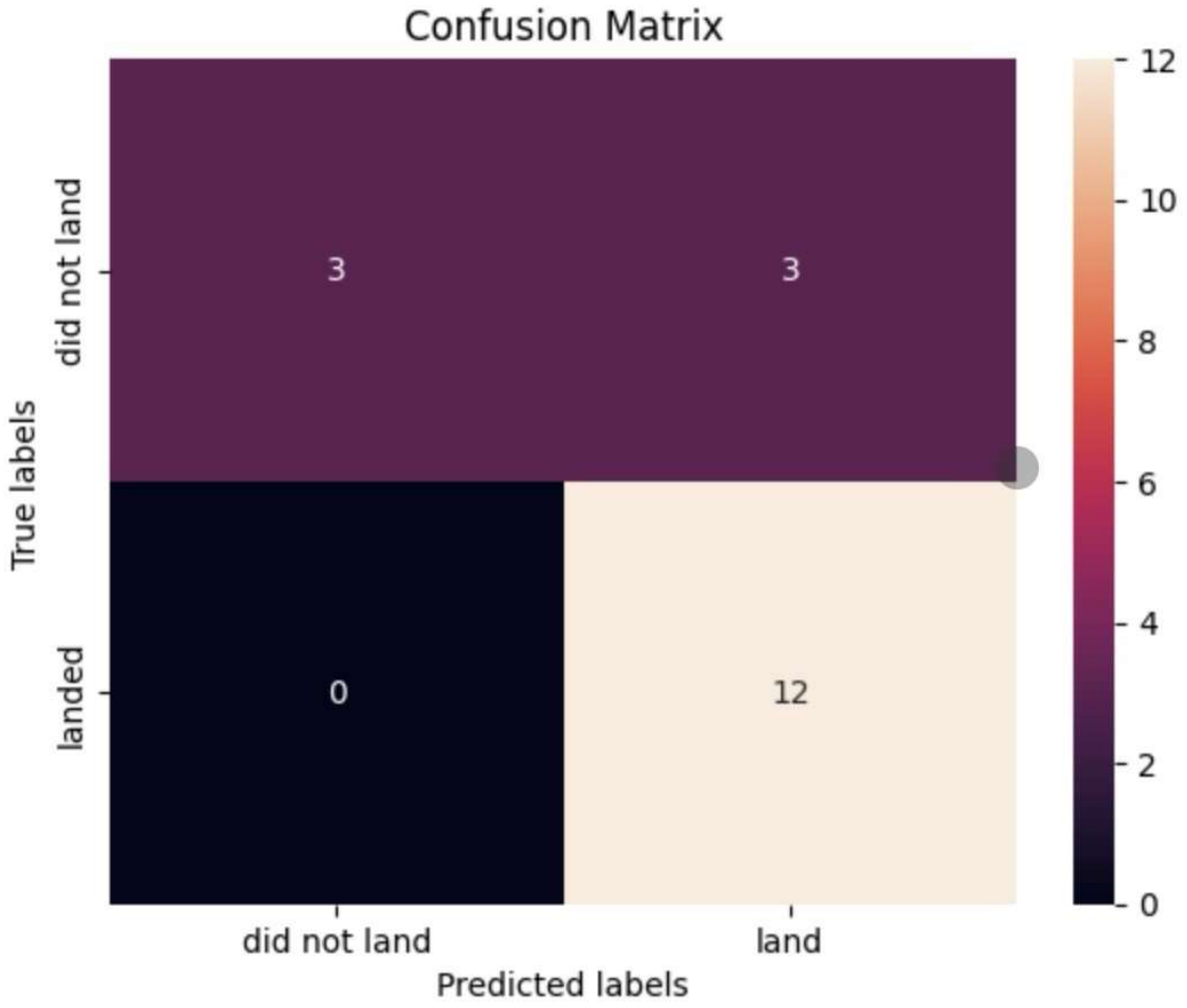
In [46]:

```
algorithms = {'KNN':knn_cv.best_score_,'Tree':tree_cv.best_score_,'LogisticRegression':logreg_cv.best_score_}
bestalgorithm = max(algorithms, key=algorithms.get)
print('Best Algorithm is',bestalgorithm,'with a score of',algorithms[bestalgorithm])
if bestalgorithm == 'Tree':
    print('Best Params is :',tree_cv.best_params_)
if bestalgorithm == 'KNN':
    print('Best Params is :',knn_cv.best_params_)
if bestalgorithm == 'LogisticRegression':
    print('Best Params is :',logreg_cv.best_params_)
```

```
Best Algorithm is Tree with a score of 0.8767857142857143
Best Params is : {'criterion': 'entropy', 'max_depth': 2, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_sampl
es_split': 2, 'splitter': 'random'}
```

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.

Appendix

Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!