



IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Negar Farrokhian
06/13/2023



Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

➤ Summary of methodologies

- Data Collection API
- Data Collection with Web Scraping
- Data Wrangling
- EDA with SQL
- EDA with Data Visualization
- Interactive Visual Analytics with Folium
- Machine Learning Prediction

➤ Summary of all results

- EDA Result
- Interactive Analytics with Screenshots
- Predictive Analytics Result

Introduction

- **Project background and context**

Space X promotes the launch of its Falcon 9 rockets on its website, highlighting a price tag of \$62 million. In contrast, other providers charge over \$165 million for similar rocket launches. A significant factor contributing to Space X's cost savings is its ability to reuse the first stage of the rocket. Consequently, if we can accurately predict whether the first stage will land successfully, we can determine the cost of a launch. This information becomes valuable for another company that wishes to compete with Space X in bidding for rocket launches. The project's objective is to develop a machine learning pipeline that can predict the success of first stage landings.

- **Problems you want to find answers**

- ☐ What are the factors that influence the successful landing of a rocket?
- ☐ What are the necessary conditions that must be met for a landing program to be successful?
- ☐ How do various features interact with each other to determine the likelihood of a successful landing?
- ☐ What operating conditions should be in place to ensure a successful landing program?

Section 1

Methodology

Methodology

Executive Summary

- **Data collection methodology:**
 - Data collection is carried out using two main methods: accessing the SpaceX REST API and web scraping from Wiki pages. The SpaceX REST API, accessed through specific endpoints, provides data on past launches, including rocket details, payload information, launch and landing specifications, and outcomes. The API responses, received in JSON format, are converted into a structured table using the `json_normalize` function. Additionally, web scraping with BeautifulSoup is utilized to extract Falcon 9 launch records from HTML tables on Wiki pages, which are then parsed and transformed into a Pandas data frame. The collected data is further refined by filtering out Falcon 1 launches and addressing null values, such as replacing null values in PayloadMass with the mean value. The resulting clean dataset is prepared for subsequent analysis and visualization.
- **Perform data wrangling**
 - The data processing steps involved in this context include cleaning the dataset, encoding categorical variables (such as launch sites), classifying different types of orbits, and converting the outcome of the first stage landing into binary classes (0 for failure, 1 for success). These steps were taken to ensure the data was prepared and transformed in a way that facilitated analysis and modeling.

Methodology Continue

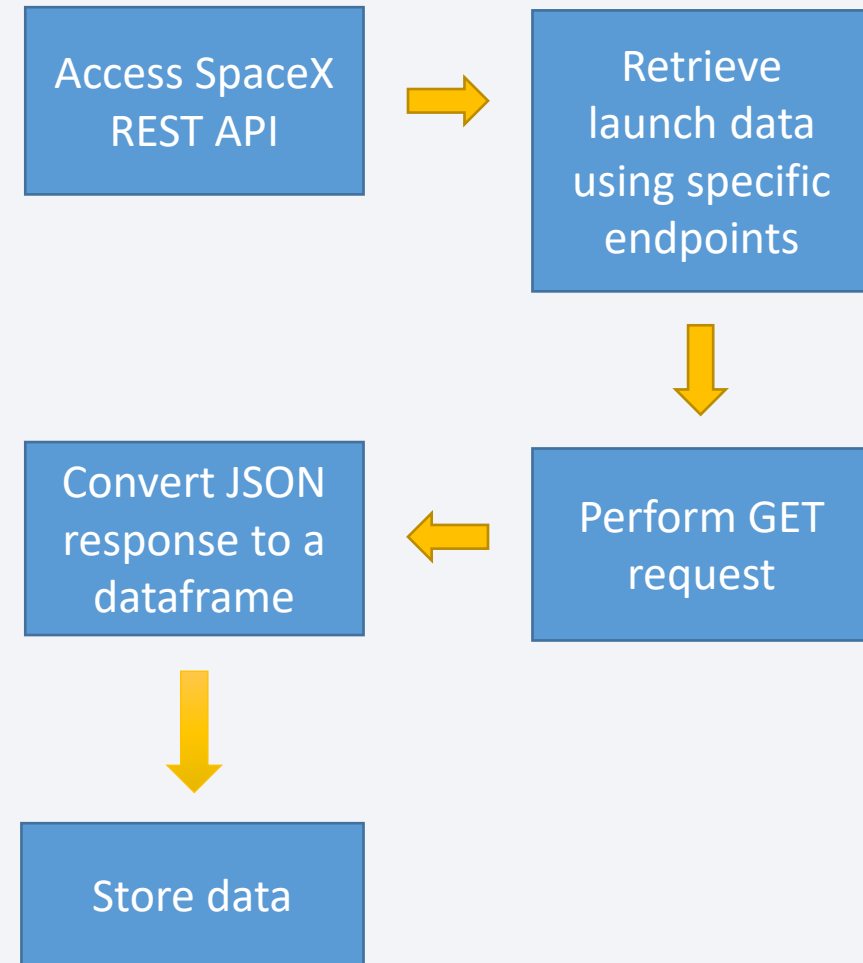
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
 - How to build, tune, evaluate classification models

Data Collection

- The data sets were collected using two methods: API data collection and web scraping. In the API data collection process, the SpaceX REST API was accessed to retrieve launch data by making GET requests to specific endpoints. The JSON responses obtained were then converted into structured tables using the `json_normalize` function. On the other hand, web scraping involved using the BeautifulSoup package to extract data from HTML tables on relevant Wiki pages. The extracted data was transformed into dataframes. These two approaches were used to gather the necessary data for further analysis.

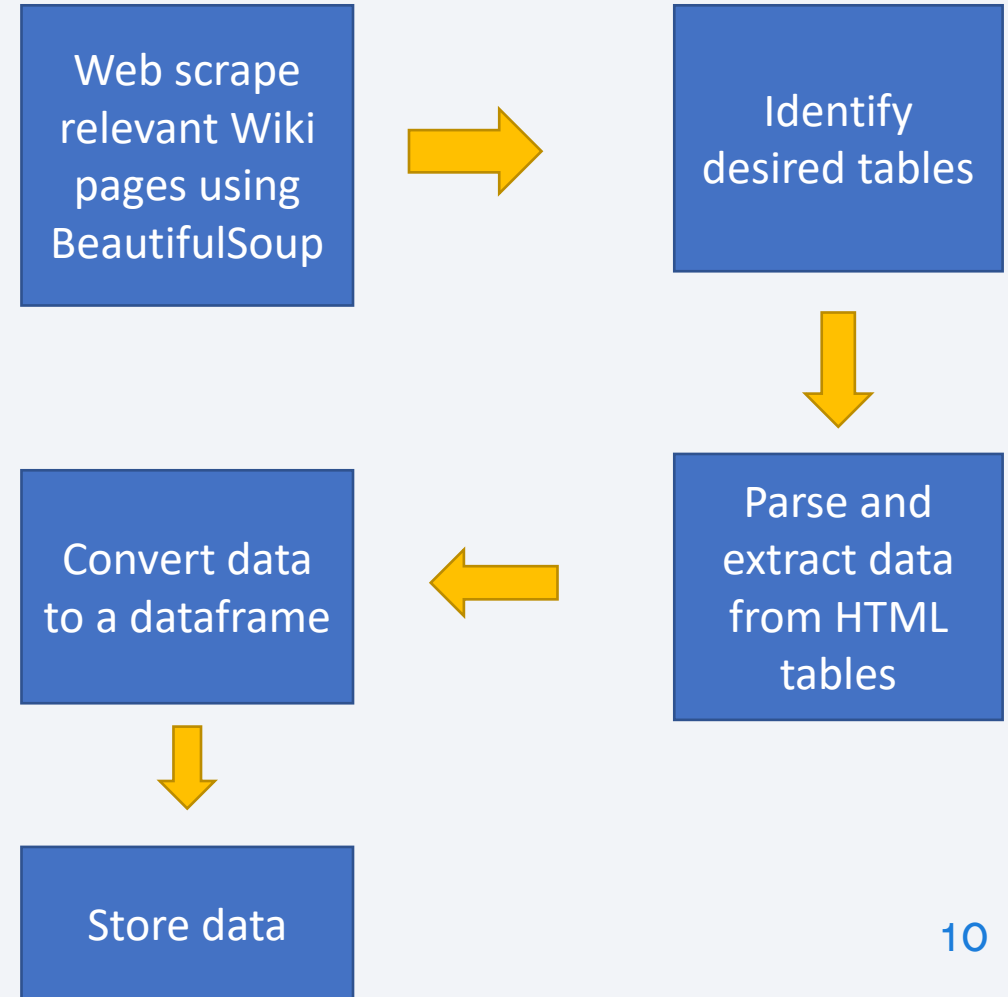
Data Collection – SpaceX API

- Access the SpaceX REST API to gather launch data.
- Utilize specific API endpoints, such as `"/launches/past"` to retrieve past launch data.
- Perform a GET request using the requests library to obtain the launch data from the API.
- Convert the API response, received in JSON format, into a structured table using the `json_normalize` function.
- Store the resulting data in a dataframe.
- <https://github.com/Negar1993/Applied-Data-Science-Capstone.git>



Data Collection - Scraping

- Use Python's BeautifulSoup package to scrape HTML tables from relevant Wiki pages.
- Identify the desired tables containing Falcon 9 launch records.
- Parse and extract the data from the HTML tables.
- Convert the scraped data into a Pandas dataframe.
- <https://github.com/Negar1993/Applied-Data-Science-Capstone.git>



Data Wrangling

- on data wrangling in the context of SpaceX launches. It highlights various attributes such as Flight Number, Date, Booster version, Payload mass, Orbit, Launch Site, Outcome (representing the first stage status), Grid Fins, Reused, Legs, Landing pad, Block, Reused count, Serial, and Longitude/Latitude of launch. The LaunchSite column contains different launch sites, and the Orbit column specifies the payload's orbit type. The Outcome column indicates the success or failure of the first stage landing, with True representing successful landings and False representing unsuccessful ones. The objective is to convert the landing outcomes into binary classes (0 or 1), where 0 signifies a failed landing and 1 signifies a successful landing. This classification variable, denoted as Y, will be used to represent the outcome of each launch for further analysis.

Data Wrangling Continue

- **Process**

Perform EDA on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Determine the count and frequency of mission outcomes for each type of orbit.

Export dataset as .csv

Generate a labeling system for landing outcomes based on the information in the outcome column.

Calculate the success rate for each landing recorded in the dataset

EDA with Data Visualization

- Scatter graphs being drawn:
 - Flight Number VS. Payload Mass
 - Flight Number VS. Launch Site
 - Payload VS. Launch Site
 - Orbit VS. Flight Number
 - Payload VS. Orbit Type
 - Orbit VS. Payload Mass
- Bar Graph being drawn:
 - Mean VS. Orbit
- Line graph being drawn:
 - Success Rate VS. Year

<https://github.com/Negar1993/Applied-Data-Science-Capstone.git>

EDA with SQL

- **The subsequent SQL queries were executed:**
 - Names of the unique launch sites in the space mission;
 - Top 5 launch sites whose name begins with the string 'CCA';
 - Total pay load mass carried by boosters launched by NASA (CRS);
 - Average payload mass carried by booster version F9 v1.1;
 - Date when the first successful landing outcome in ground pad was achieved;
 - Names of the boosters which have success in drone ship and have payload mass between 4000 and 6000 kg;
 - Total number of successful and failure mission outcomes;
 - Names of the booster versions which have carried the maximum payload mass;
 - Failed landing out comes in droneship, their booster versions, and launch site names for in year 2015; and
 - Rank of the count of landing outcomes (such as Failure (droneship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20.
- <https://github.com/Negar1993/Applied-Data-Science-Capstone.git>

Build an Interactive Map with Folium

- We created an interactive map to display Launch Data in a visually engaging way. By obtaining the Latitude and Longitude Coordinates for each launch site, we placed a Circle Marker around them and labeled each marker with the corresponding launch site name. The launch outcomes were categorized into classes 0 and 1, representing failures and successes, respectively. These classes were represented on the map by Green and Red markers, respectively, using the MarkerCluster() feature.
- To gather insights about the surroundings of the launch sites, we utilized Haversine's formula to calculate the distances between the launch site and various landmarks. This analysis helped identify certain trends and patterns. Additionally, we drew lines on the map to measure the distance from the launch site to these landmarks.
- Here are a few examples of the trends observed in the placement of launch sites:
 - Proximity to railways: Launch sites were not found to be in close proximity to railways.
 - Proximity to highways: Similarly, launch sites were not found to be in close proximity to highways.
 - Proximity to coastline: Launch sites were observed to be in close proximity to the coastline.
 - Distance from cities: Launch sites were observed to maintain a certain distance from cities.

Overall, these observations provide insights into the spatial characteristics and considerations when selecting launch sites.

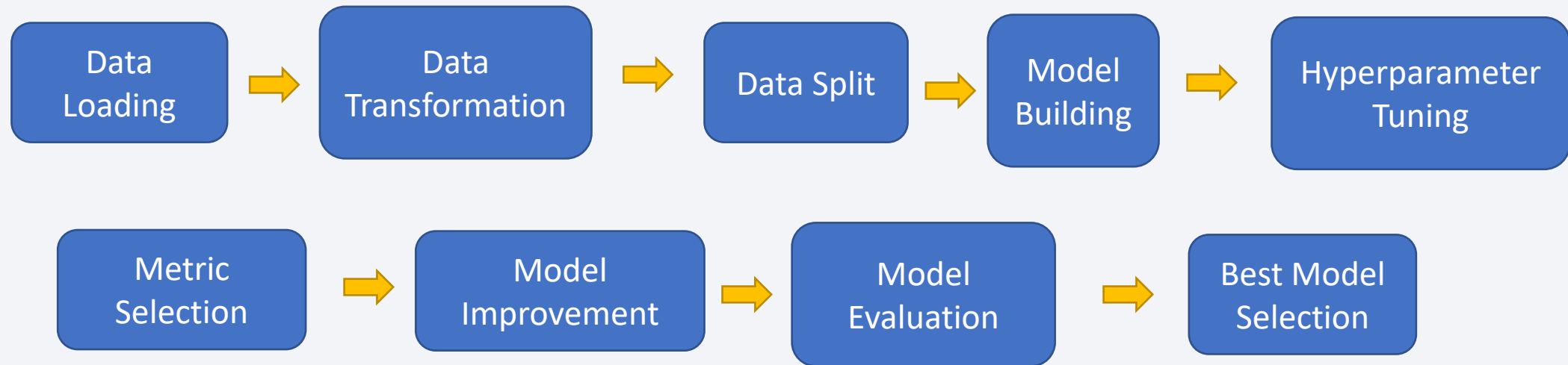
- <https://github.com/Negar1993/Applied-Data-Science-Capstone.git>

Build a Dashboard with Plotly Dash

- Graphs can be utilized to visually represent data. One type of graph that serves this purpose is a Pie Chart, which showcases the total number of launches either by a specific site or collectively by all sites. It effectively illustrates the relative proportions of different data classes. Furthermore, the size of each segment in the Pie Chart can be adjusted to reflect the corresponding total quantity it represents.
- A Scatter Graph is used to visually depict the relationship between the Outcome and Payload Mass (measured in Kg) for different Booster Versions. It effectively presents the correlation between these two variables. This graph is particularly useful for displaying non-linear patterns in the data. By observing the graph, it is easy to determine the range of data flow, including the maximum and minimum values. The graph allows for straightforward observation and interpretation of the data.
- <https://github.com/Negar1993/Applied-Data-Science-Capstone.git>

Predictive Analysis (Classification)

- We imported and processed the data using libraries such as numpy and pandas. The data was transformed and divided into separate training and testing sets. Various machine learning models were constructed, and different hyperparameters were tuned using GridSearchCV. The accuracy metric was used to evaluate the performance of the models. To enhance the model's effectiveness, we employed feature engineering techniques and fine-tuned the algorithms. Ultimately, we identified the classification model that yielded the best performance.



Results

results of the exploratory data analysis:

- Space X operates from four different launch sites.
- Initial launches were conducted by Space X itself and NASA.
- On average, the payload of the F9 v1.1 booster is 2,928 kg.
- The first successful landing occurred in 2015, five years after the initial launch.
- Several versions of the Falcon 9 booster successfully landed on drone ships, particularly when the payload exceeded the average.
- Nearly 100% of the missions had successful outcomes.
- Two booster versions, namely F9 v1.1 B1012 and F9 v1.1 B1015, failed to land on drone ships in 2015.
- The success rate of landing outcomes improved over the years.

Results

- By employing interactive analytics, it became evident that launch sites are strategically located in safe areas, often near the sea, and possess a robust logistic infrastructure.
- Additionally, a majority of launches occur at launch sites situated on the east coast.



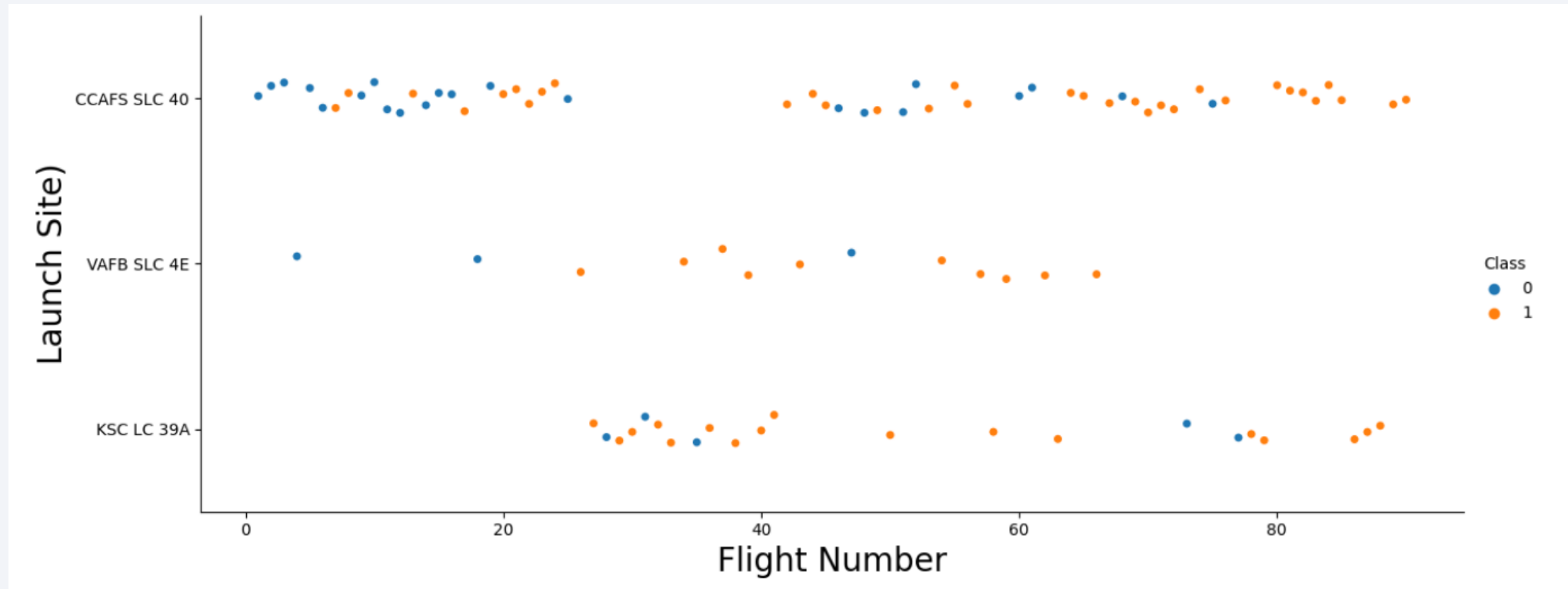
<https://github.com/Negar1993/Applied-Data-Science-Capstone.git>

The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

Section 2

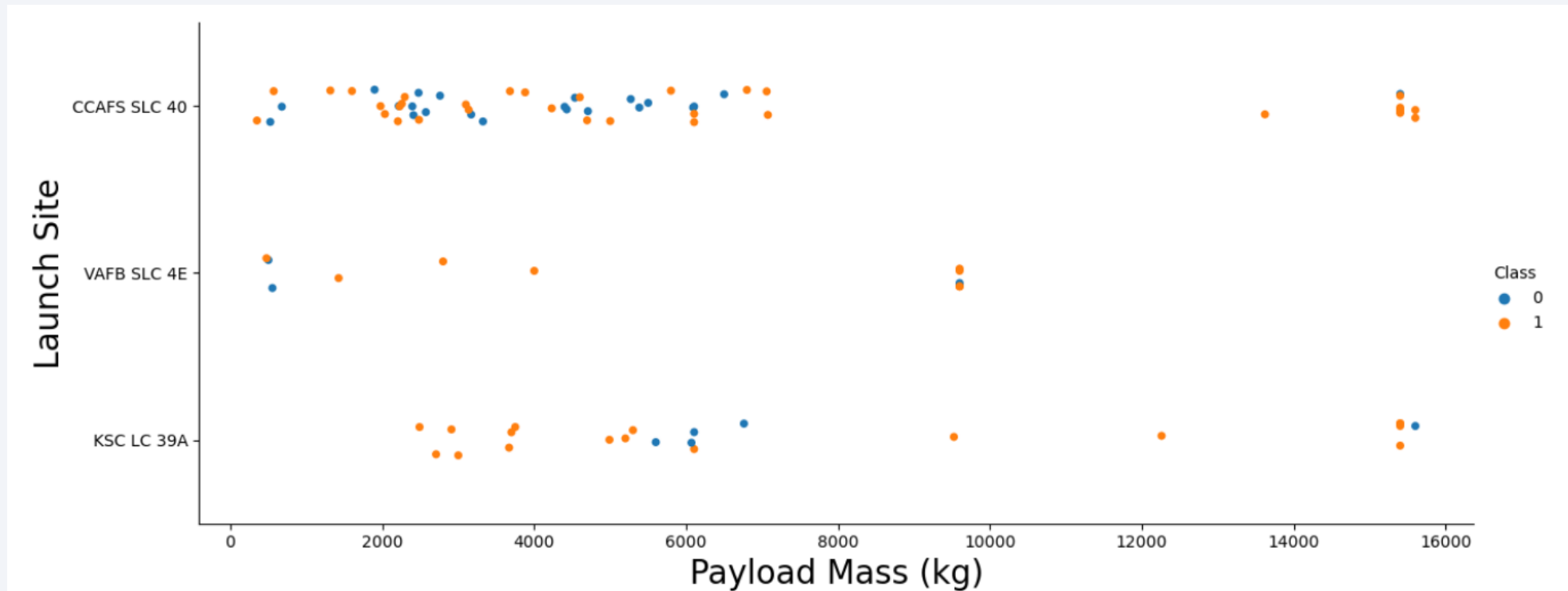
Insights drawn from EDA

Flight Number vs. Launch Site



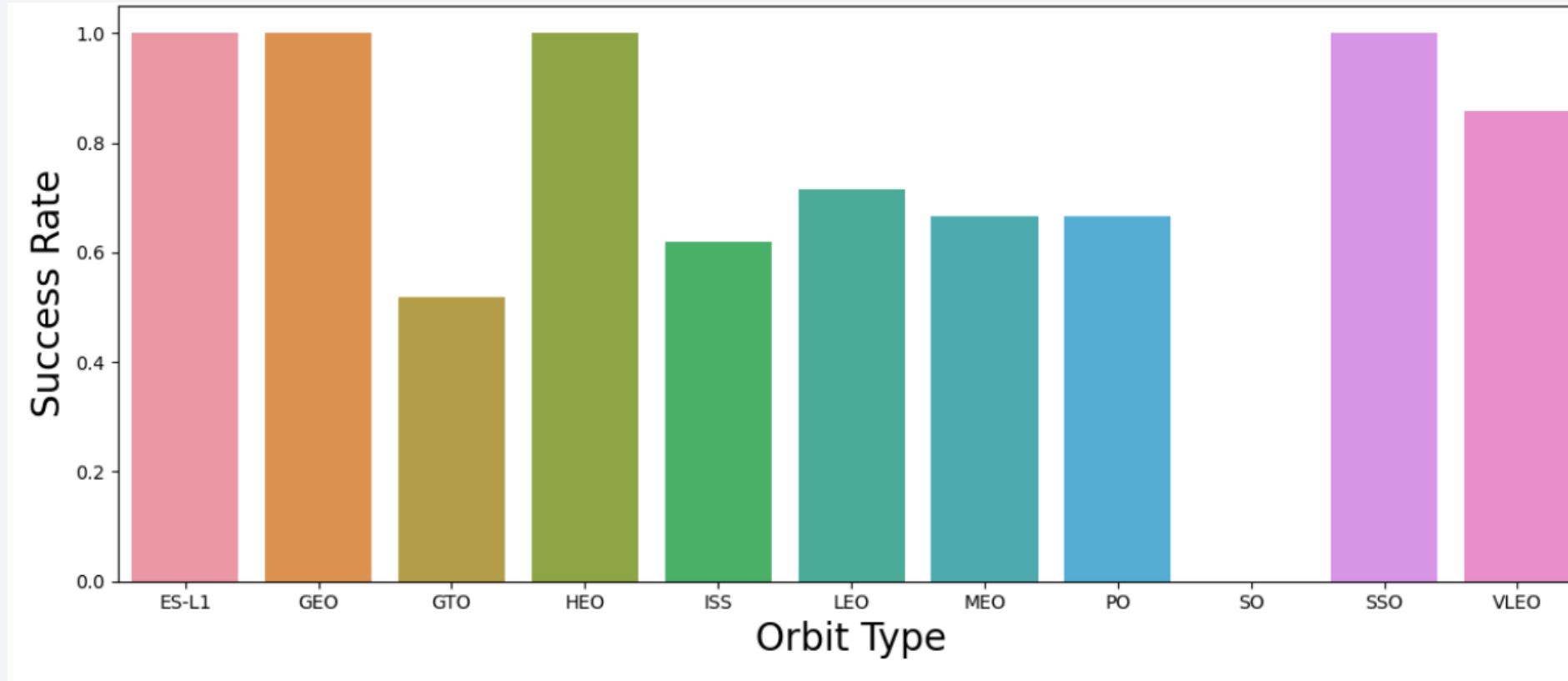
Based on the plot, it can be inferred that there is a positive correlation between the number of flights at a launch site and the success rate at that site. In other words, a higher number of flights tends to result in a greater likelihood of success.

Payload vs. Launch Site



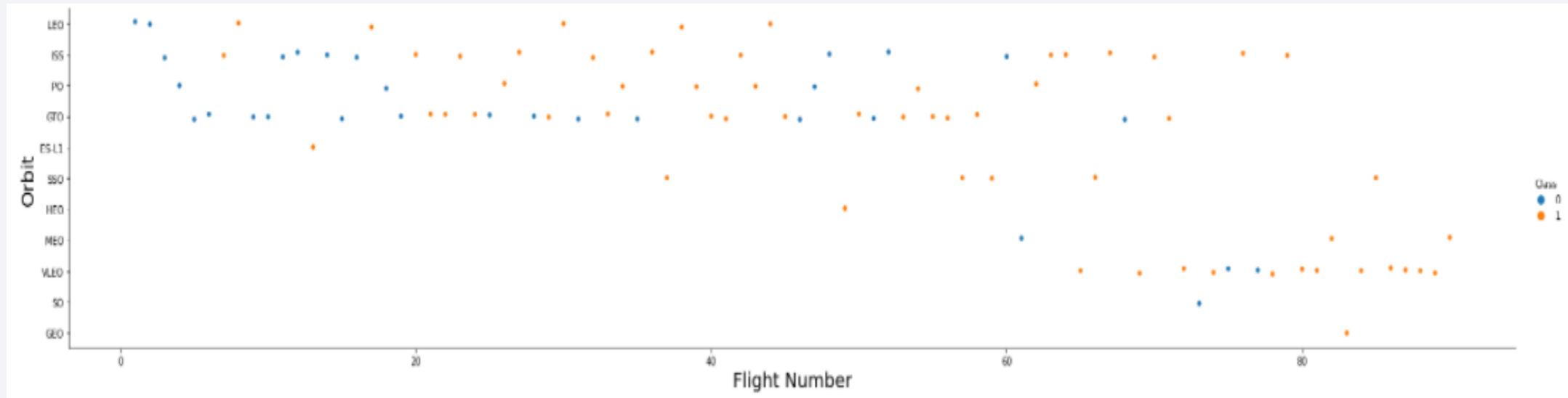
The success rate for the rocket at Launch Site CCAFS SLC 40 increases as the payload mass becomes greater.

Success Rate vs. Orbit Type

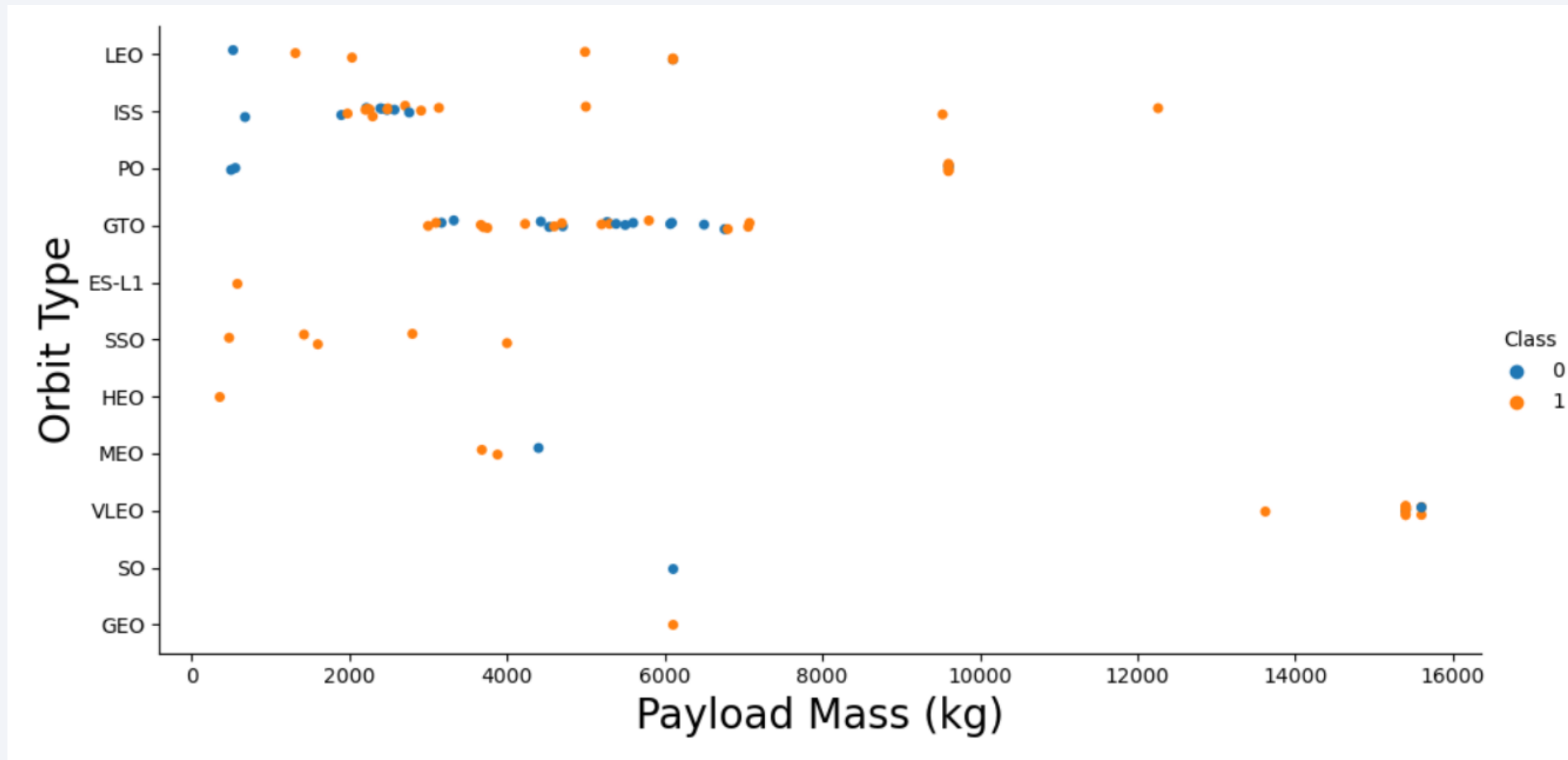


Based on the plot, it is evident that ES-L1, GEO, HEO, SSO exhibited the highest success rates.

Flight Number vs. Orbit Type

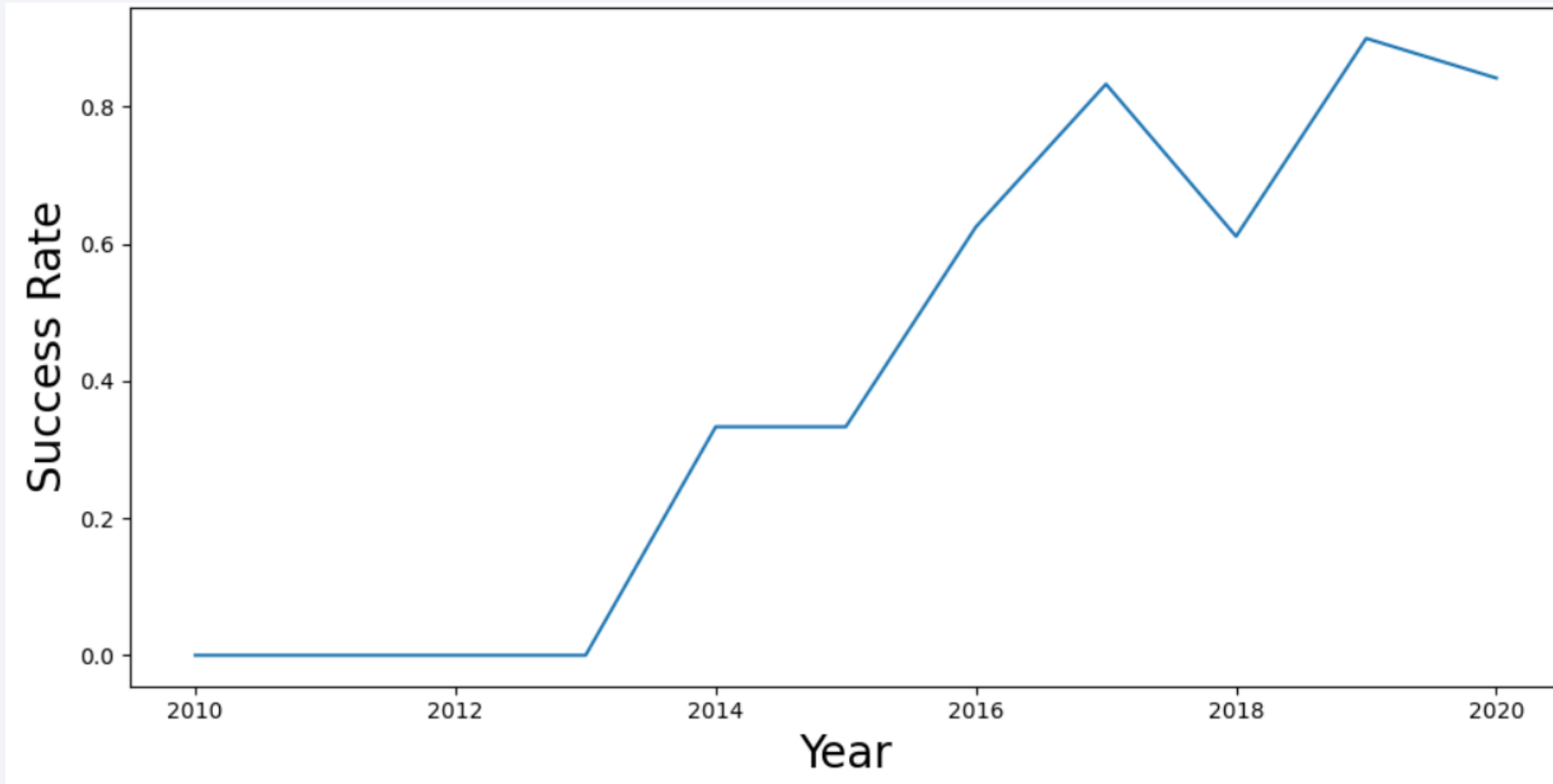


Payload vs. Orbit Type



It can be observed that heavy payloads are associated with a higher rate of successful landings in PO, LEO, and ISS orbits.

Launch Success Yearly Trend



It is evident that the success rate has been consistently rising from 2013 to 2020.

All Launch Site Names

```
In [28]: %sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL ORDER BY 1
```

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

```
Out[28]: Launch_Site
```

```
None
```

```
CCAFS LC-40
```

```
CCAFS SLC-40
```

```
KSC LC-39A
```

```
VAFB SLC-4E
```

To display only unique launch sites from the SpaceX data, we utilized the keyword "DISTINCT."

Launch Site Names Begin with 'CCA'

```
In [29]: %sql SELECT * FROM SPACEXTBL WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

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

Out[29]:		Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS_KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
		06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachute)
		12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
		22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attempt
		10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attempt
		03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attempt

To retrieve five records from the SpaceX data where the launch sites start with "CCA," we employed the aforementioned query.

Total Payload Mass

```
In [31]: %sql SELECT SUM (PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE CUSTOMER='NASA (CRS)'
```

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

```
Out[31]: SUM (PAYLOAD_MASS__KG_)  
         45596.0
```

By executing the query provided above, we computed that the total payload transported by NASA boosters amounts to 45,596.

Average Payload Mass by F9 v1.1

```
In [32]: %sql SELECT AVG (PAYLOAD_MASS_KG_) FROM SPACEXTBL WHERE Booster_Version like 'F9 v1.1%'
* sqlite:///my_data1.db
Done.
Out[32]: AVG (PAYLOAD_MASS_KG_)
          2534.6666666666665
```

Through our analysis, we determined that the average payload mass carried by the booster version F9 v1.1 is 2534.65

First Successful Ground Landing Date

```
In [33]: %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Mission_outcome LIKE 'Success%'
* sqlite:///my_data1.db
Done.
Out[33]: MIN(Date)
         01/06/2014
```

Our observation reveals that the first successful landing on a ground pad took place on 01/06/2014

Successful Drone Ship Landing with Payload between 4000 and 6000

```
In [45]: sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS_KG_ BETWEEN 4000 AND 6000 AND "Landing_Outcome" = 'Success (drone ship)'

* sqlite:///my_data1.db
(sqlite3.OperationalError) no such column: Landing_Outcome
[SQL: SELECT Booster_Version FROM SPACEXTBL WHERE [Landing_Outcome] = 'Success (drone ship)' AND PAYLOAD_MASS_KG_ > 4000 AND PAYLOAD_MASS_KG_ < 6000;]
(Background on this error at: http://sqlalche.me/e/e3q8)
```

By utilizing the WHERE clause, we filtered for boosters that achieved successful landings on a drone ship. Additionally, we applied the AND condition to identify successful landings with a payload mass greater than 4000 but less than 6000.

Total Number of Successful and Failure Mission Outcomes

```
In [36]: %sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME like 'Success%'
```

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

```
Done.
```

```
Out[36]:
```

Mission_Outcome	COUNT(MISSION_OUTCOME)
-----------------	------------------------

Success	100
---------	-----

```
In [37]: %sql SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) FROM SPACEXTBL WHERE MISSION_OUTCOME = 'Failure (in flight)'
```

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

```
Done.
```

```
Out[37]:
```

Mission_Outcome	COUNT(MISSION_OUTCOME)
-----------------	------------------------

Failure (in flight)	1
---------------------	---

To filter for records where the MissionOutcome was either a success or a failure, we utilized the wildcard '%' as part of the filtering criteria.

Boosters Carried Maximum Payload

```
In [38]: %sql SELECT BOOSTER_VERSION FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
* sqlite:///my_data1.db
Done.
```

```
Out[38]: 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

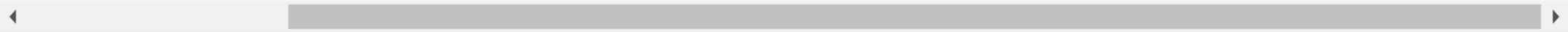
F9 B5 B1060.3

F9 B5 B1049.7

By incorporating a subquery within the WHERE clause and utilizing the MAX() function, we were able to identify the booster that has carried the highest payload.

2015 Launch Records

```
[27]: LaunchSite, LandingOutcome from SpaceX where LandingOutcome Like 'Failure (drone ship)' AND Date Between '2015-01-01' And '2015-12-31'
```



The screenshot shows a Jupyter Notebook cell with a SQL query. Below the query, there is a scrollable area containing the error message. The error message indicates that the table 'SpaceX' does not exist in the database 'sqlite:///my_data1.db'. The SQL query is: `SELECT BoosterVersion, LaunchSite, LandingOutcome from SpaceX where LandingOutcome Like 'Failure (drone ship)' AND Date Between '2015-01-01' And '2015-12-31'`. A link is provided for more information about the error: <http://sqlalche.me/e/e3q8>.

```
* sqlite:///my_data1.db
(sqlite3.OperationalError) no such table: SpaceX
[SQL: SELECT BoosterVersion, LaunchSite, LandingOutcome from SpaceX where LandingOutcome Like 'Failure (drone ship)' AND Date Between '2015-01-01' And '2015-12-31']
(Background on this error at: http://sqlalche.me/e/e3q8)
```

To filter for failed landing outcomes on a drone ship, their corresponding booster versions, and launch site names specifically for the year 2015, we employed a combination of the WHERE clause, LIKE, AND, and BETWEEN conditions.

A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

Launch Sites Proximities Analysis

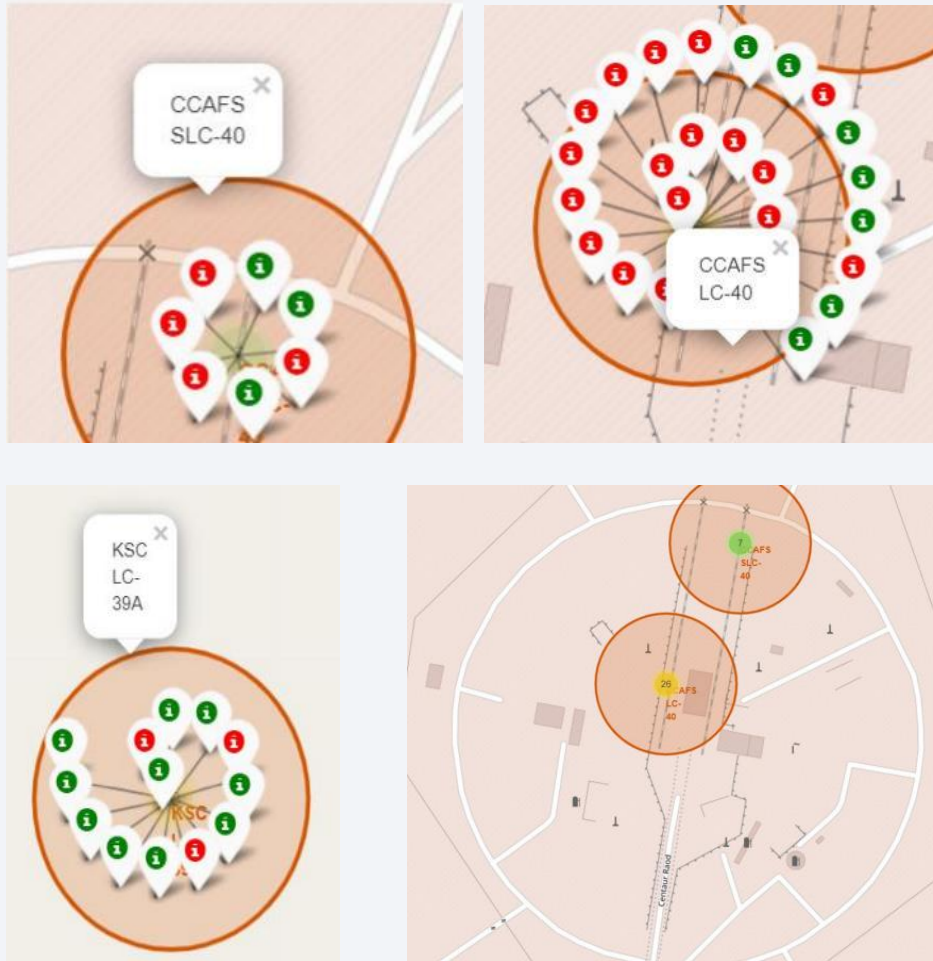
All launch sites global map markers



It is evident from the data that SpaceX launch sites are situated along the coastlines of the United States of America, specifically in Florida and California.

Markers showing launch sites with color labels

Florida Launch Sites

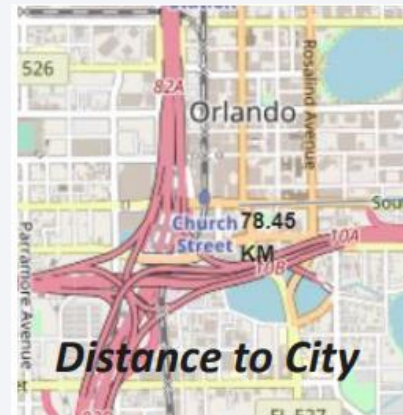
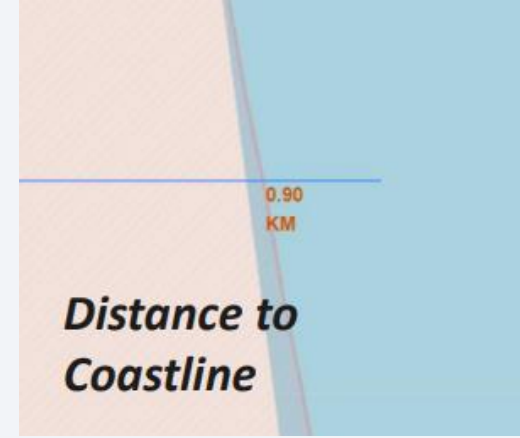
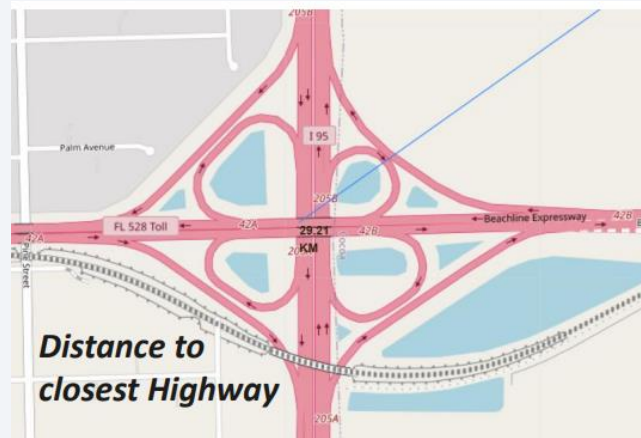


California Launch Site



Successful launches are indicated by green markers, while failures are represented by red markers.

Launch Site distance to landmarks



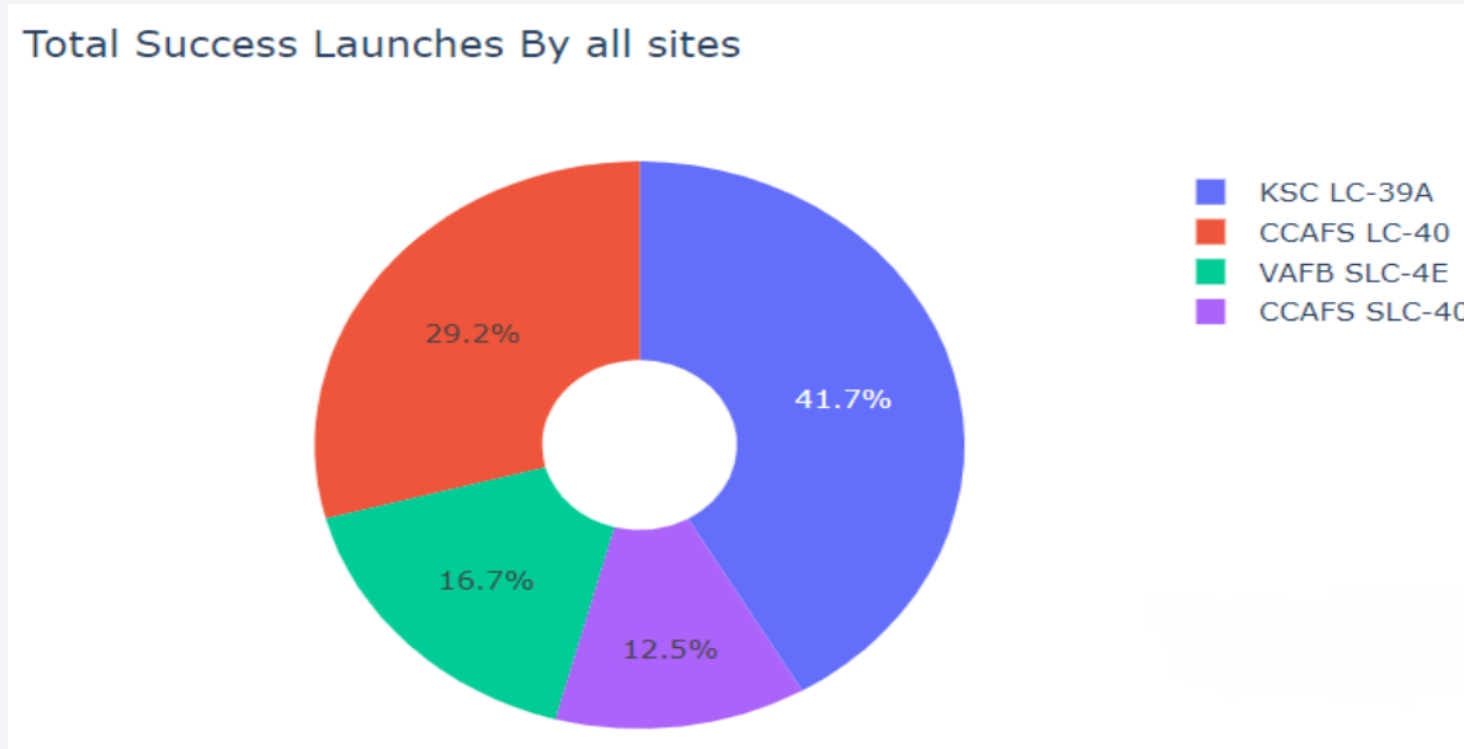
Launch sites are not located near railways or highways. However, they are typically situated in close proximity to coastlines. Additionally, launch sites tend to maintain a certain distance from cities.



Section 4

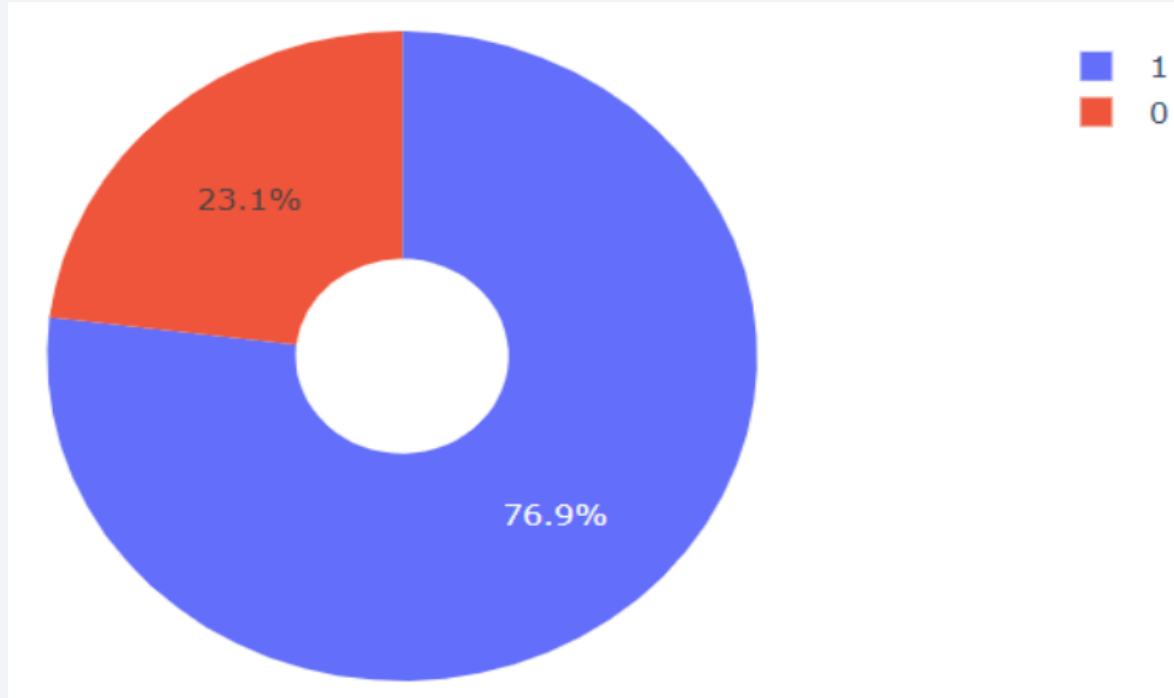
Build a Dashboard with Plotly Dash

Pie chart showing the success percentage achieved by each launch site



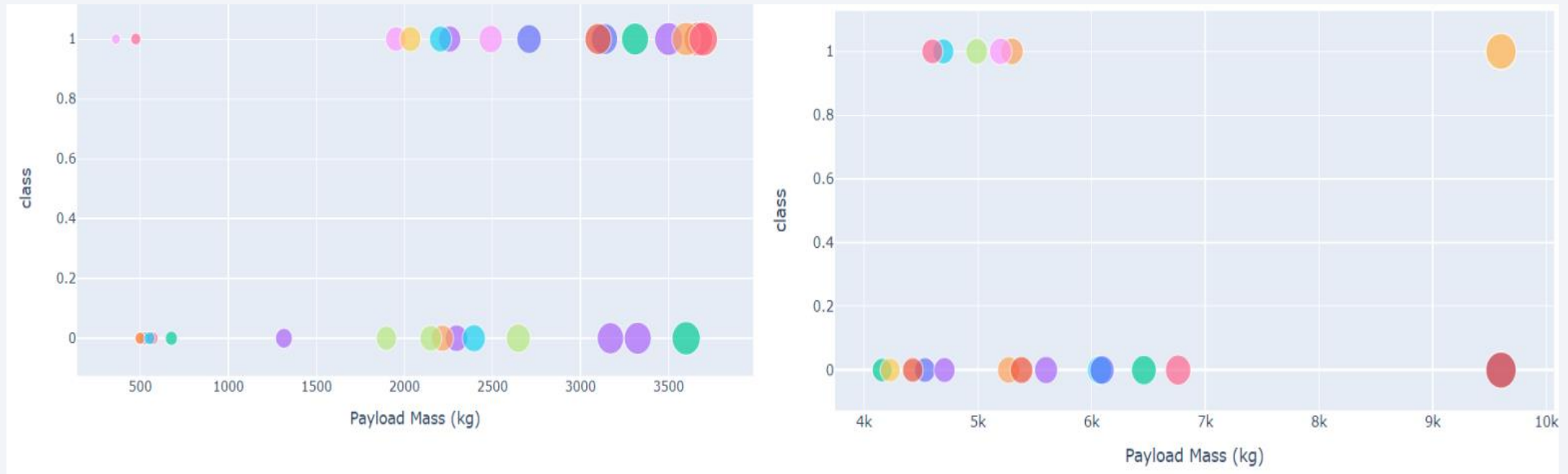
From the data, it is evident that KSC LC-39A had the highest number of successful launches among all the launch sites.

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



KSC LC-39A attained a success rate of 76.9% and experienced a failure rate of 23.1%.

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



The data reveals that the success rates are higher for payloads with lower weights compared to those with heavier weights.

Section 5

Predictive Analysis (Classification)

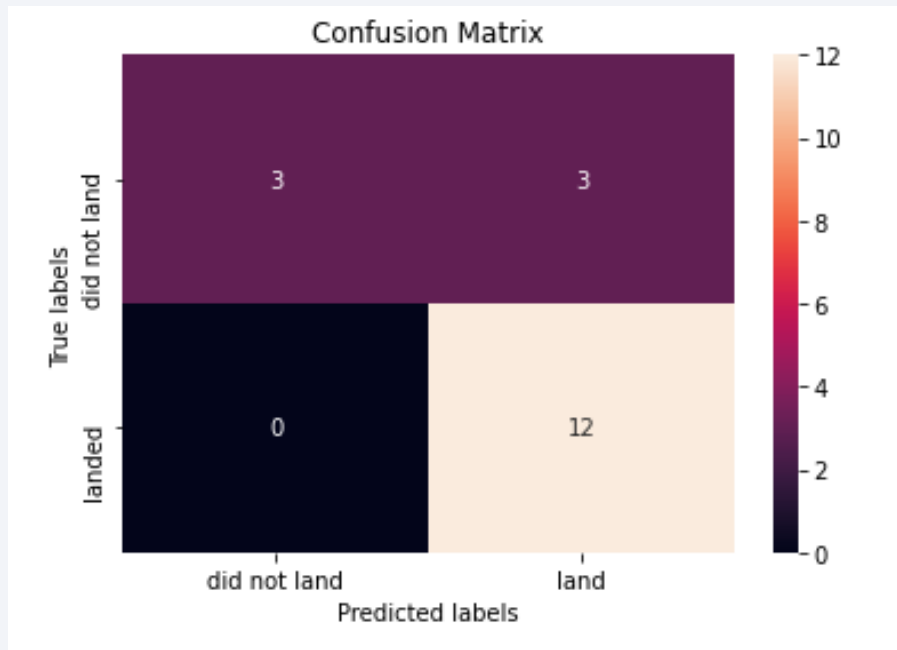
Classification Accuracy

```
In [40]: predictors = [knn_cv, svm_cv, logreg_cv, tree_cv]
best_predictor = ""
best_result = 0
for predictor in predictors:

    predictor.score(X_test, Y_test)
```

Confusion Matrix

The confusion matrix of the decision tree classifier reveals its ability to differentiate between the various classes. However, a significant issue arises in the form of false positives, where the classifier incorrectly identifies unsuccessful landings as successful landings.



Conclusions

- Based on the dataset, the Tree Classifier Algorithm proves to be the most suitable for machine learning.
- It is observed that low weighted payloads exhibit better performance compared to heavier payloads.
- The success rates of SpaceX launches show a direct correlation with the passage of time, indicating an improvement over the years.
- Among all the launch sites, KSC LC-39A stands out with the highest number of successful launches.
- Furthermore, orbits such as GEO, HEO, SSO, and ES-L1 demonstrate the best success rates.

Thank you!

