IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Novahu Gondela
21 October 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API
  - Data Collection with Web Scraping
  - Data Wrangling - Exploratory Data Analysis with SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visual Analytics with Folium
  - Machine Learning Prediction
- Summary of all results
  - Exploratory Data Analysis result
  - Interactive analytics in screenshots
  - Predictive Analytics result from Machine Learning Lab

Introduction

SpaceX is a revolutionary company who has disrupt the space industry by offering a rocket launches specifically Falcon 9 as low as 62 million dollars; while other providers cost upward of 165 million dollar each. Most of this saving thanks to SpaceX astounding idea to reuse the first stage of the launch by re-land the rocket to be used on the next mission. Repeating this process will make the price down even further. As a data scientist of a startup rivaling SpaceX, the goal of this project is to create the machine learning pipeline to predict the landing outcome of the first stage in the future. This project is crucial in identifying the right price to bid against SpaceX for a rocket launch.

The problems included:

- Identifying all factors that influence the landing outcome.
- The relationship between each variables and how it is affecting the outcome.
- The best condition needed to increase the probability of successful landing.

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

- Data collection process involved a combination of API requests from SpaceX REST API and Web Scraping data from a table in SpaceX’s Wikipedia entry. We had to use both of these data collection methods in order to get complete information about the launches for a more detailed analysis.

- **Data Columns are obtained by using SpaceX REST API:**

- FlightNumber, Date, BoosterVersion, PayloadMass, Orbit, LaunchSite, Outcome, Flights, GridFins, Reused, Legs, LandingPad, Block, ReusedCount, Serial, Longitude, Latitude Data

- **Columns are obtained by using Wikipedia Web Scraping:**

- Flight No., Launch site, Payload, PayloadMass, Orbit, Customer, Launch outcome, Version Booster, Booster landing, Date, Time.

Data Collection – SpaceX API

1. Requesting rocket launch data from SpaceX API
2. Decoding the response content using .json() and turning it into a data frame using .json_normalize()
3. Requesting needed information about the launches from SpaceX API by applying custom functions
4. Constructing data we have obtained into a dictionary
5. Creating a dataframe from the dictionary
6. Filtering the data frame to only include Falcon 9 launches
7. Replacing missing values of Payload Mass column with calculated .mean() for this column
8. Exporting the data to CSV

[Github: Jupyter notebook for data collection-spacex api]

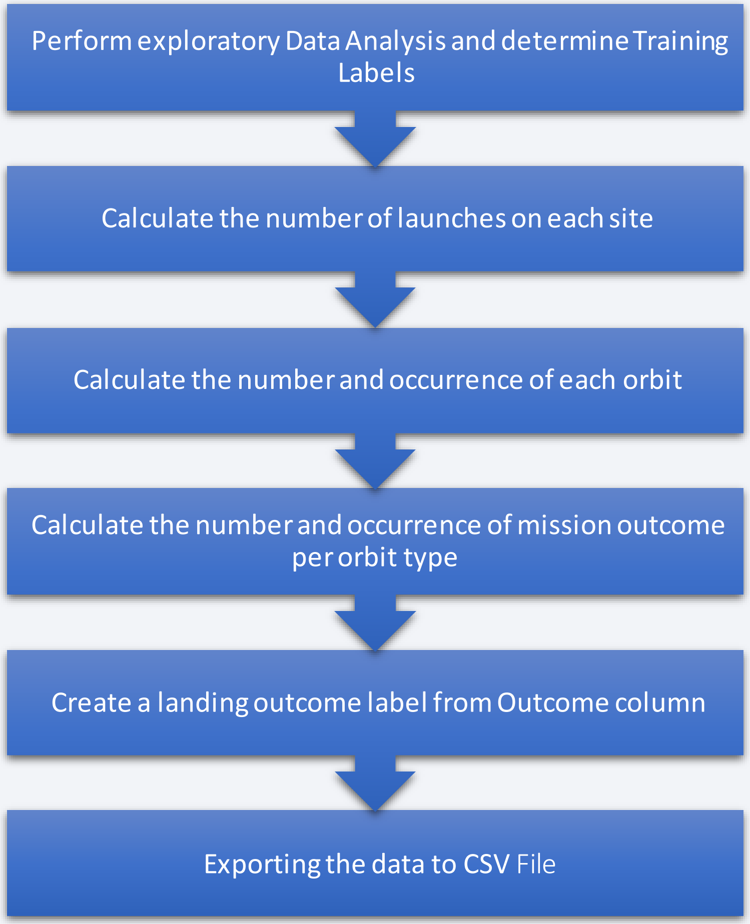
Data Collection - Scraping

1. Requesting Falcon 9 launch data from Wikipedia
2. Creating a Beautiful Soup object from the HTML response
3. Extracting all column names from the HTML table header
4. Collecting the data by parsing HTML tables
5. Constructing data we have obtained into a dictionary
6. Creating a dataframe from the dictionary
7. Exporting the data to CSV

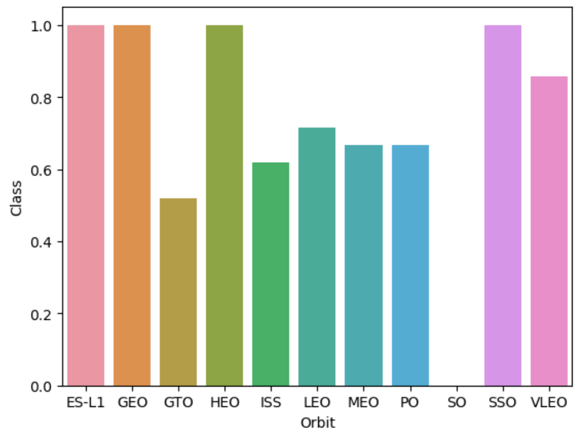
[Github: Jupyter notebook data collection-scraping]()

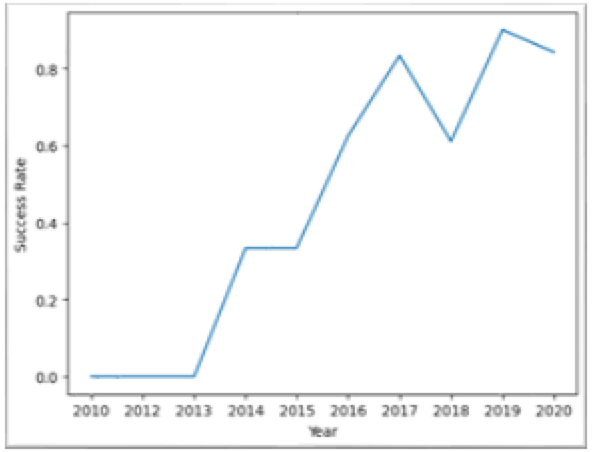
Data Wrangling

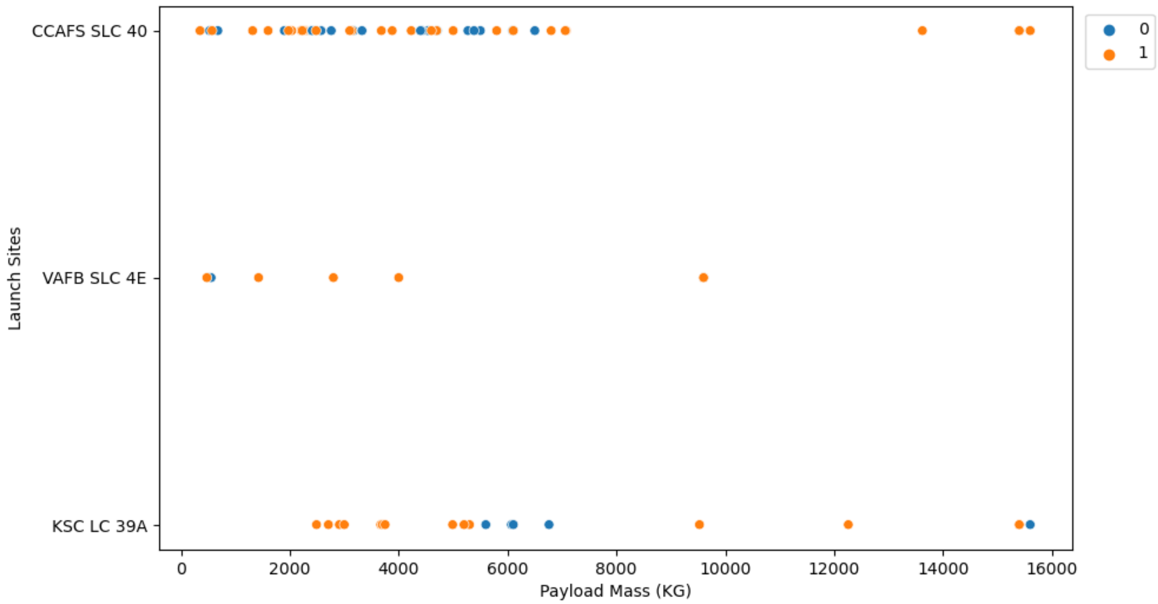
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; for example, True Ocean means the mission outcome was successfully landed to a specific region of the ocean while False Ocean means the mission outcome was unsuccessfully landed to a specific region of the ocean. True RTLS means the mission outcome was successfully landed to a ground pad False RTLS means the mission outcome was unsuccessfully landed to a ground pad. True ASDS means the mission outcome was successfully landed on a drone ship False ASDS means the mission outcome was unsuccessfully landed on a drone ship. We mainly convert those outcomes into Training Labels with “1” means the booster successfully landed, “0” means it was unsuccessful.



Github: Jupyter Notebook Data Wrangling







EDA with Data Visualization

Charts plotted are:

- Scatter Plots of :
  - FlightNumber vs Launch Sites
  - Payload Mass (KG) vs Launch Sites
  - FlightNumber vs Orbit type
  - Payload Mass (KG) vs Orbit type
- Box plot for Success rate of each orbit type
- Line plot for trend of launch success yearly

Scatterplots show the relationship between variables. If a relationship exists, they could be used in machine learning model.

Bar charts show comparisons among discrete categories. The goal is to show the relationship between the specific categories being compared and a measured value.

Line charts show trends in data over time (time series).

[Github: Jupyter Notebook for EDA with Data Visualization]

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 and 2017-03-20, in descending order.

[Github: Jupyter Notebook for EDA with SQL]()

Build an Interactive Map with Folium

Markers of all Launch Sites:

- Added Marker with Circle, Popup Label and Text Label of NASA Johnson Space Center using its latitude and longitude coordinates as a start location.
- Added Markers with Circle, Popup Label and Text Label of all Launch Sites using their latitude and longitude coordinates to show their geographical locations and proximity to Equator and coasts.

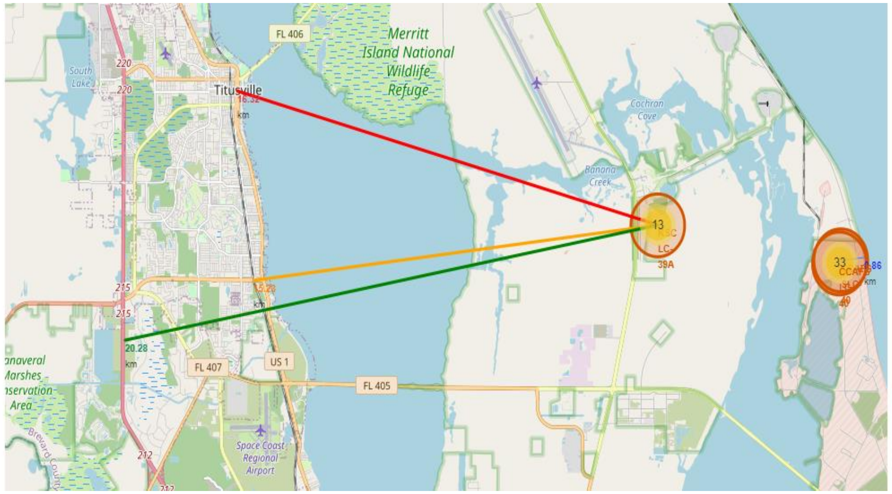
Colored Markers of the launch outcomes for each Launch Site:

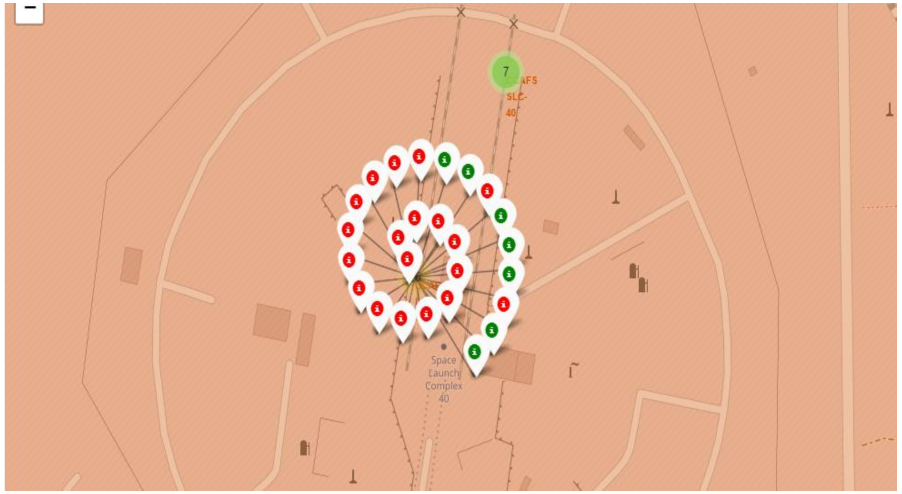
- Added colored Markers of success (Green) and failed (Red) launches using Marker Cluster to identify which launch sites have relatively high success rates.

Distances between a Launch Site to its proximities:

- Added coloured Lines to show distances between the Launch Site KSC LC-39A (as an example) and its proximities like Railway, Highway, Coastline and Closest City

[Github: Jupyter Notebook for Interactive map with folium]()





Build a Dashboard with Plotly Dash

Launch Sites Dropdown List:

- Added a dropdown list to enable Launch Site selection.

Pie Chart showing Success Launches (All Sites/Certain Site):

- Added a pie chart to show the total successful launches count for all sites and the Success vs. Failed counts for the site, if a specific Launch Site was selected.

- **Slider of Payload Mass Range:**

- Added a slider to select Payload range.

Scatter Chart of Payload Mass vs. Success Rate for the different Booster Versions:

- Added a scatter chart to show the correlation between Payload and Launch Success.

Github: Dashboard app with plotly dash

Predictive Analysis (Classification)

1. Creating a NumPy array from the column “Class” in data
2. Standardizing the data with StandardScaler, then fitting and transforming it
3. Splitting the data into training and testing sets with train_test_split function
4. Creating a GridSearchCV object with cv = 10 to find the best parameters
5. Applying GridSearchCV on LogReg, SVM, Decision Tree, and KNN models
6. Calculating the accuracy on the test data using the method .score() for all models
7. Examining the confusion matrix for all models
8. Finding the method performs best by examining the best accuracy score of all algorthms

Github: Jupyter notebook for ML classification

Results

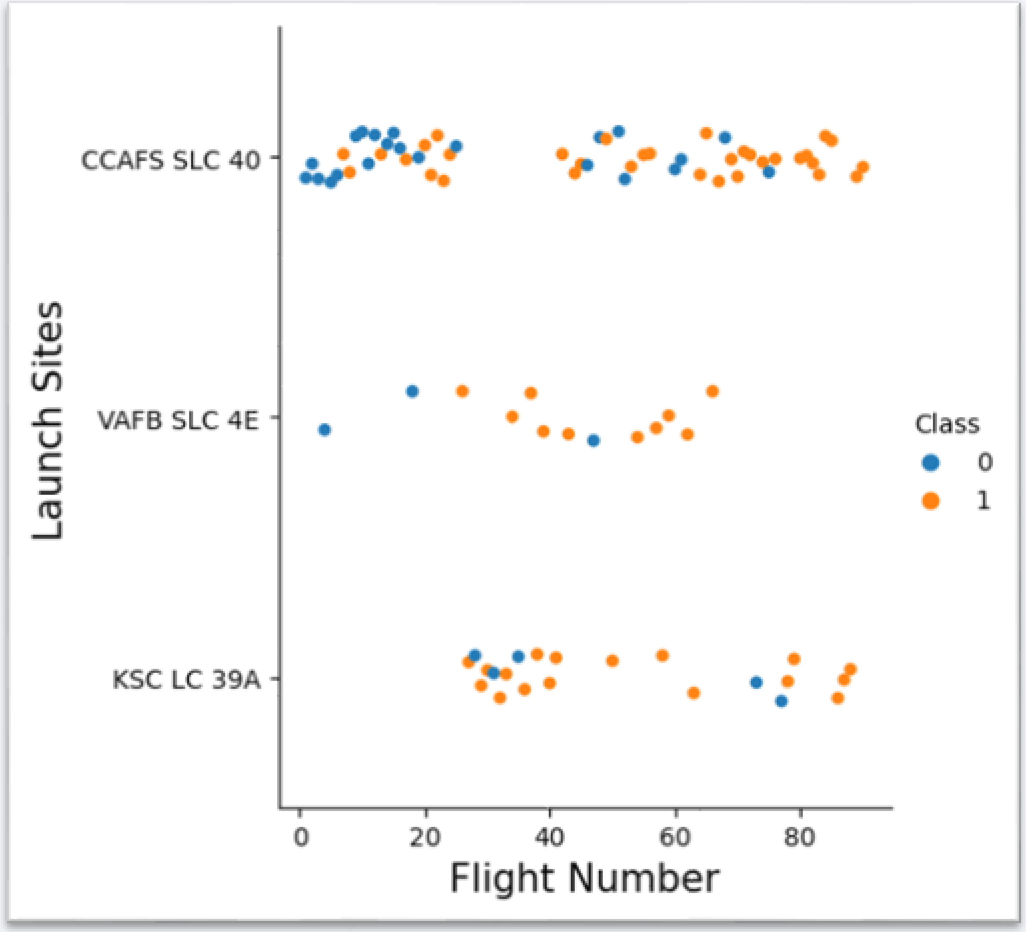
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

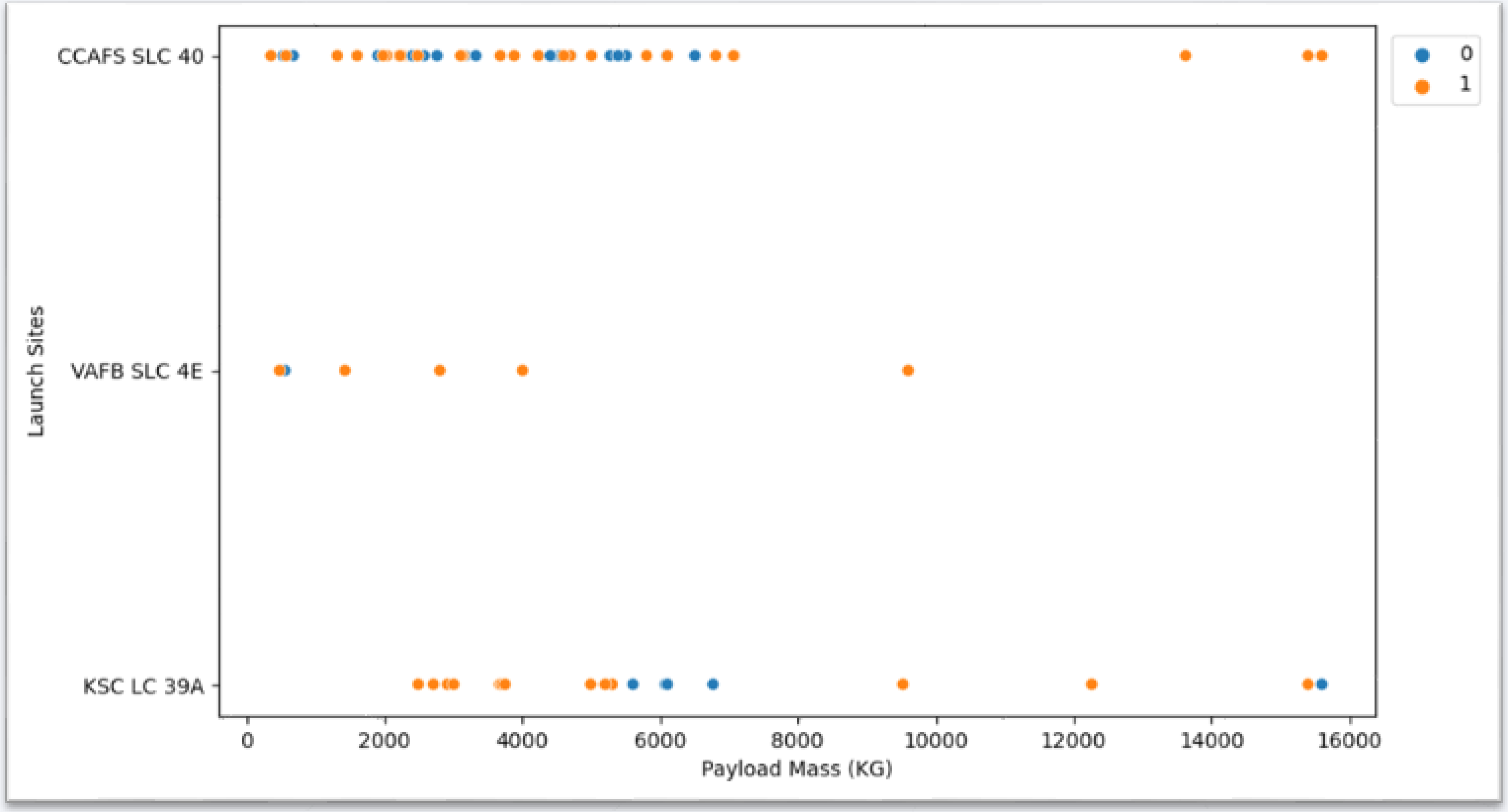
Flight Number vs. Launch Site

- The earliest flights all failed while the latest flights all succeeded.
- The CCAFS SLC 40 launch site has about a half of all launches.
- VAFB SLC 4E and KSC LC 39A have higher success rates.
- It can be assumed that each new launch has a higher rate of success.



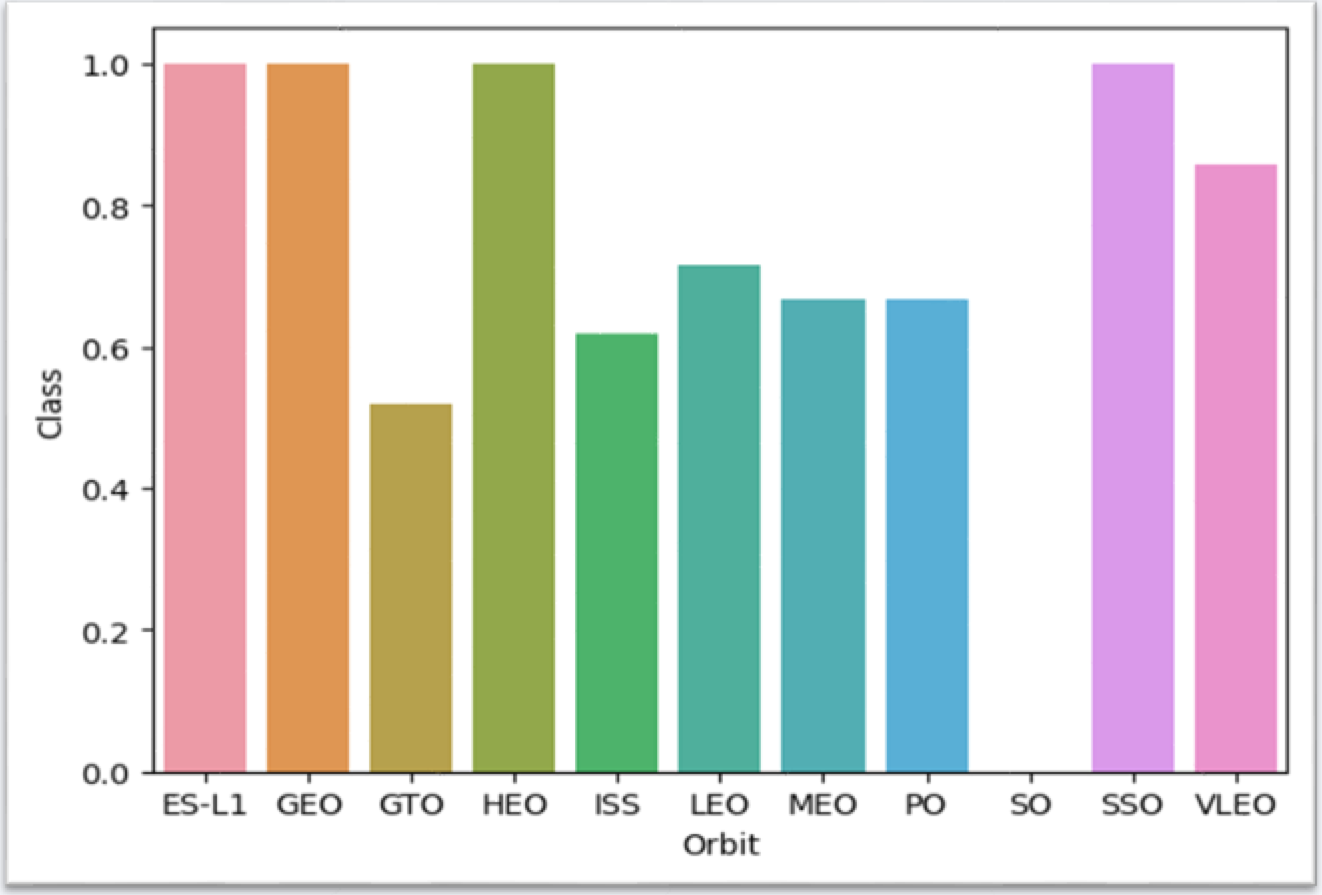
Payload vs. Launch Site

- For every launch site the higher the payload mass, the higher the success rate.
- Most of the launches with payload mass over 7000 kg were successful.
- KSC LC 39A has a 100% success rate for payload mass under 5500 kg too.



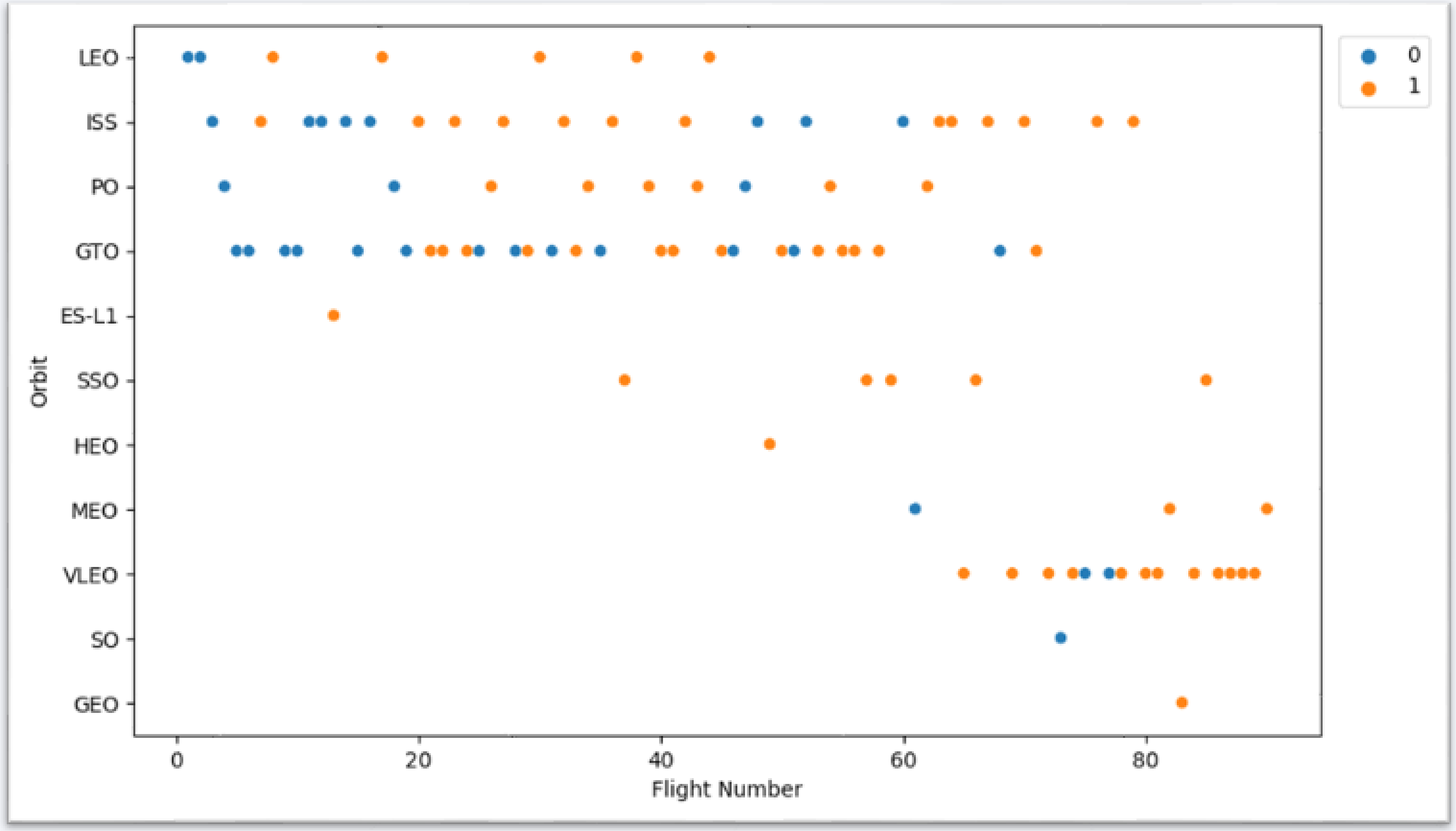
Success Rate vs. Orbit Type

- Orbits with 100% success rate: - ES-L1, GEO, HEO, SSO
- Orbits with 0% success rate: - SO
- Orbits with success rate between 50% and 85%: - GTO, ISS, LEO, MEO, PO



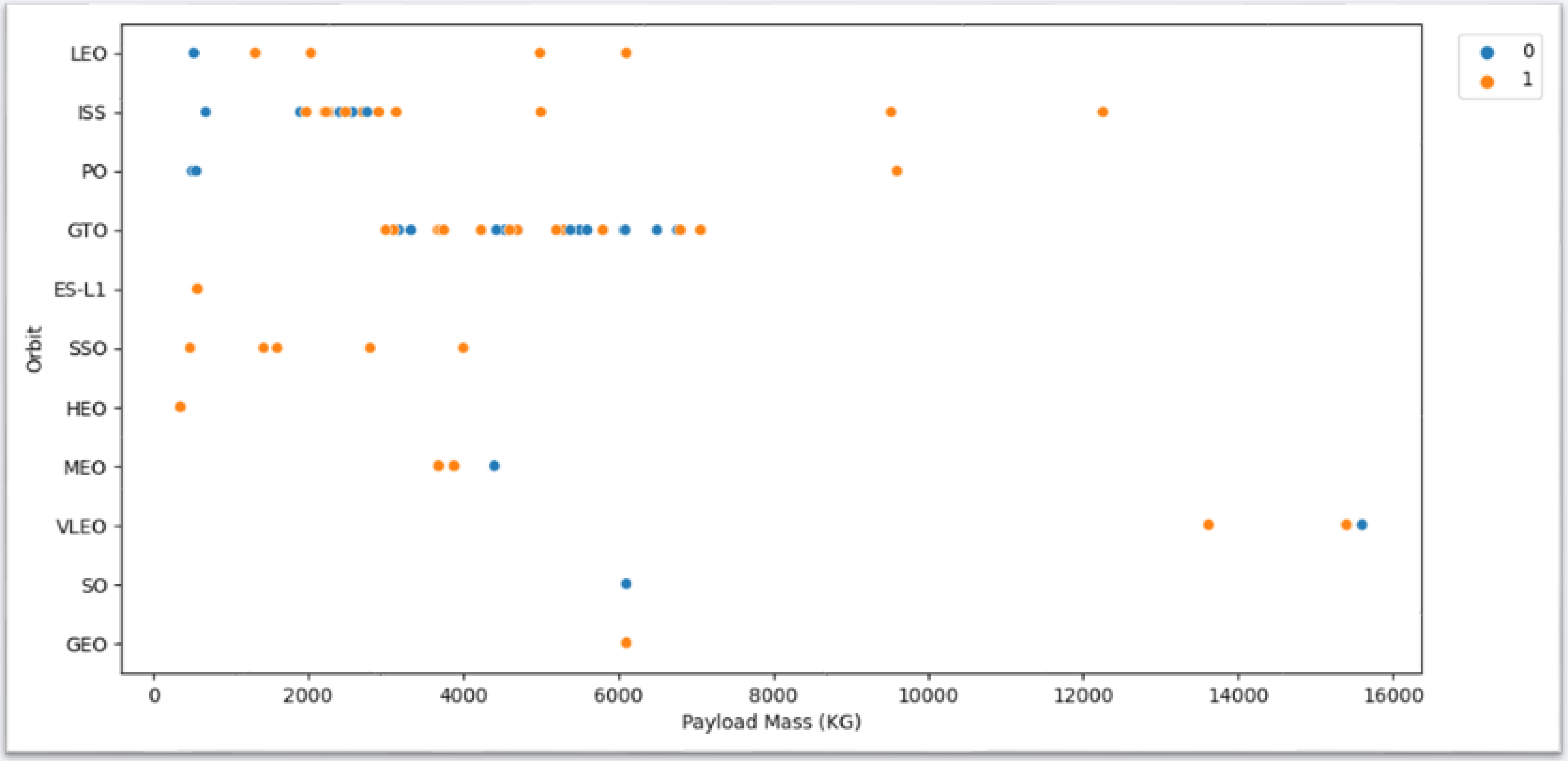
Flight Number vs. Orbit Type

- In the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



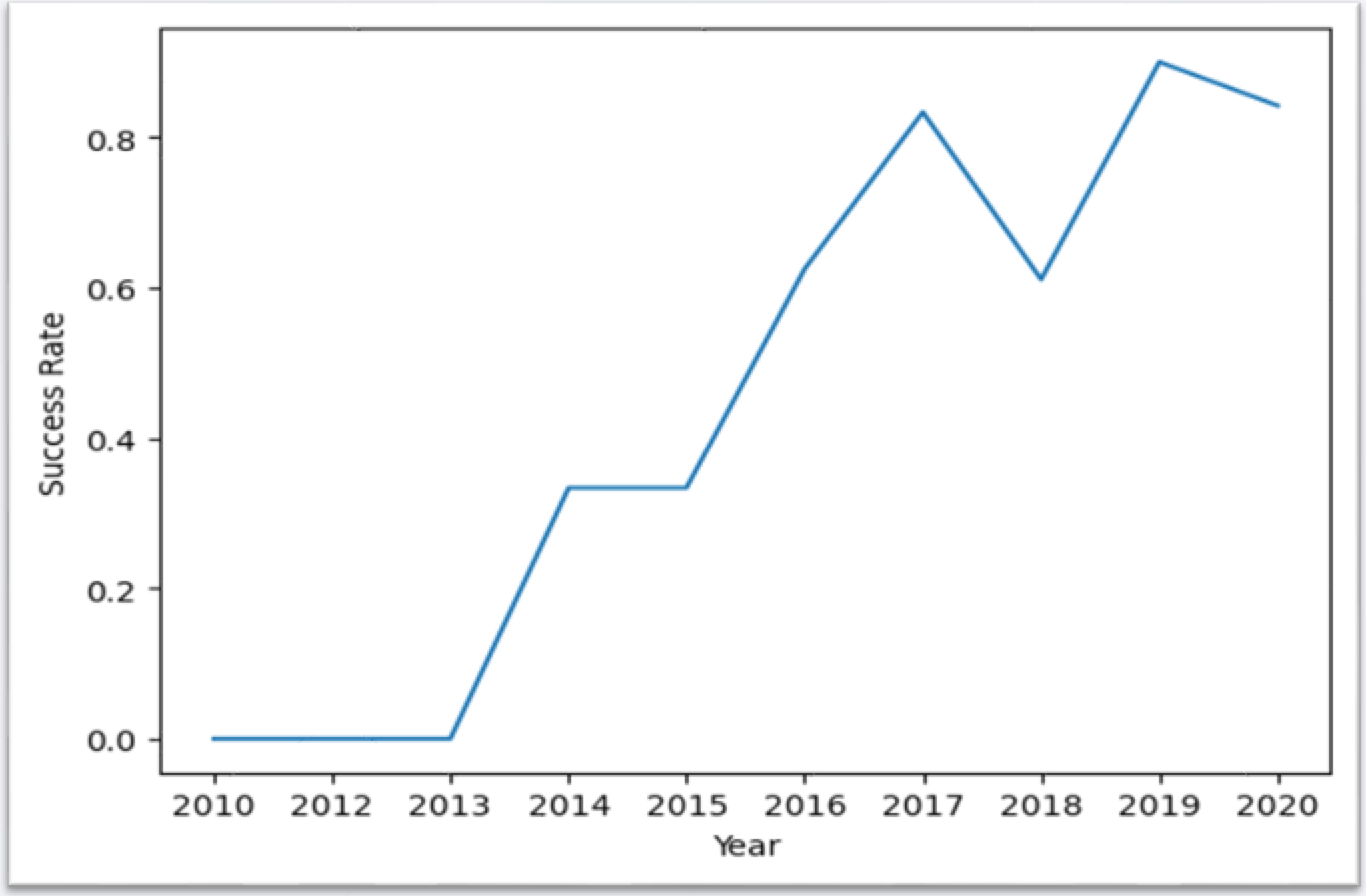
Payload vs. Orbit Type

- Heavy payloads have a negative influence on GTO orbits and positive on GTO and Polar LEO (ISS) orbits.



Launch Success Yearly Trend

- The success rate since 2013 kept increasing till 2020.



All Launch Site Names

Task 1

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

In [8]:
```
%sql select distinct launch_site from spacextable
```

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

Out[8]:

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

- Displaying the names of the unique launch sites in the space mission.

Launch Site Names Begin with 'CCA'

Task 2

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

In [9]:
```
%sql select * from spacextable where launch_site like 'CCA%' limit 5
```

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

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Displaying 5 records where launch sites begin with the string 'CCA'.

Total Payload Mass

Task 3

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

```
In [10]: %sql select sum(payload_mass__kg_) from spacextable where customer like 'NASA (CRS)'

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

Out[10]:

sum(payload_mass__kg_)
45596

- Displaying the total payload mass carried by boosters launched by NASA (CRS).

Average Payload Mass by F9 v1.1

Task 4

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

In [11]:
```
%sql select avg(payload_mass__kg_) from spacextable where booster_version like 'F9 v1.1%'
```

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

Out[11]:

avg(payload_mass__kg_)
2534.6666666666665

- Displaying average payload mass carried by booster version F9 v1.1.

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 [12]:
```
%%sql
select min(date) as first_successful_landing_date from spacextable
where landing_outcome == 'Success (ground pad)'
```

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

Out[12]:

first_successful_landing_date
2015-12-22

- Listing the date when the first successful landing outcome in ground pad was achieved.

Successful Drone Ship Landing with Payload between 4000 and 6000

Task 6

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

In [13]:
```
%%sql
select booster_version from spacextable
where landing_outcome == 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000
```

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

Out[13]:

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

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

Total Number of Successful and Failure Mission Outcomes

Task 7

List the total number of successful and failure mission outcomes

In [14]:

```
%%sql
select mission_outcome,count(*) from spacextable
group by mission_outcome
```

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

Out[14]:

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

- Listing the total number of successful and failure mission outcomes.

Boosters Carried Maximum Payload

Task 8

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

In [15]:
```
%%sql
select booster_version from spacextable
where payload_mass__kg_ == (select max(payload_mass__kg_) from spacextable)
```

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

Out[15]:

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

- Listing the names of the booster versions which have carried the maximum payload mass.

2015 Launch Records

Task 9

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 [16]:

```
%%sql
select substr(Date, 6,2) as month, booster_version, launch_site, landing_outcome from spacextable
where substr(date,0,5)='2015' and landing_outcome like 'Failure%'
```

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

Out[16]:

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

- Listing the failed landing outcomes in drone ship, their booster versions and launch site names for the months in year 2015.

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

Task 10

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 [17]:

```
%%sql
select landing_outcome,count(*) as count from spacextable
where date between '2010-06-04' and '2017-03-20'
group by landing_outcome
order by count desc
```

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

Out[17]:

Landing_Outcome	count
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

- Ranking 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.

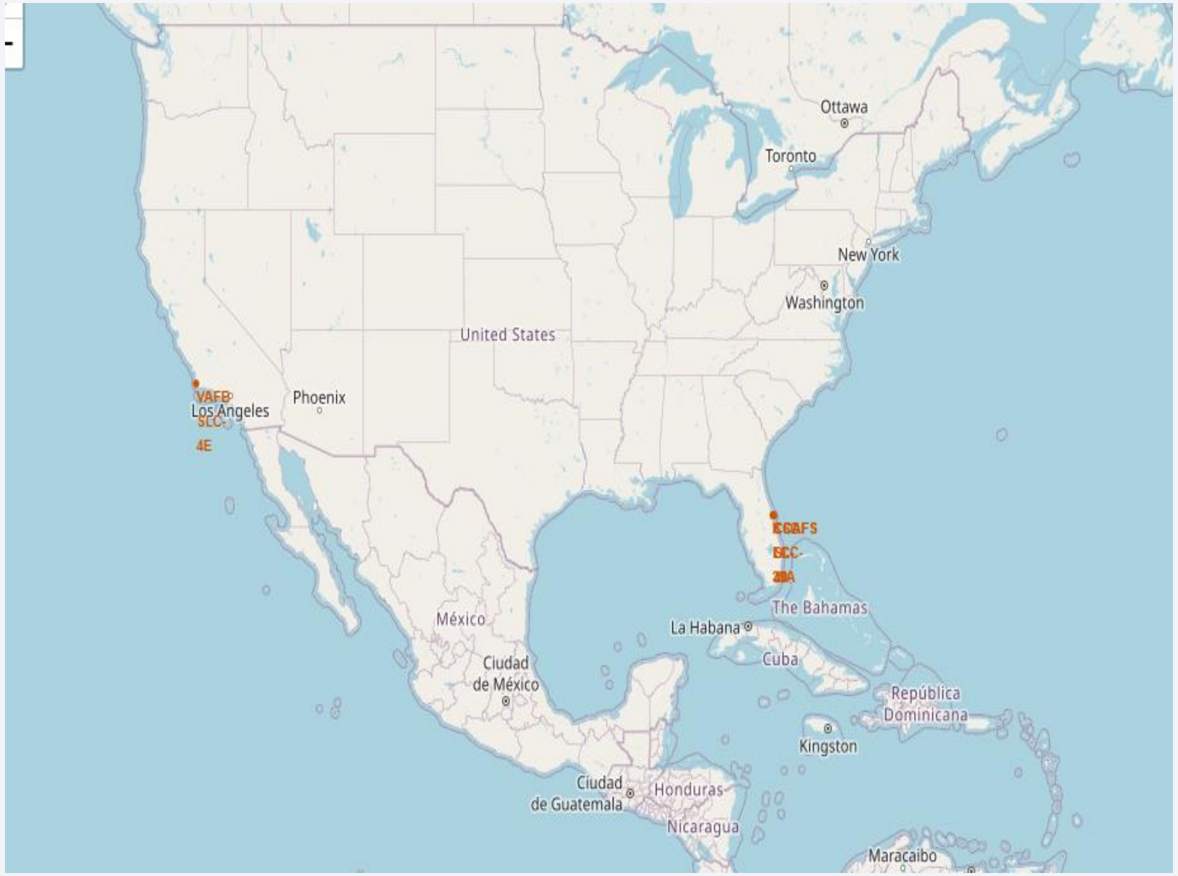
Section 3

Launch Sites Proximities Analysis

All launch sites' location markers on a global map

Explanation:

Most of Launch sites are in proximity to the Equator line. The land is moving faster at the equator than any other place on the surface of the Earth. Anything on the surface of the Earth at the equator is already moving at 1670 km/hour. If a ship is launched from the equator it goes up into space, and it is also moving around the Earth at the same speed it was moving before launching. This is because of inertia. This speed will help the spacecraft keep up a good enough speed to stay in orbit. All launch sites are in very close proximity to the coast, while launching rockets towards the ocean it minimises the risk of having any debris dropping or exploding near people.



Color-labeled launch records on the map

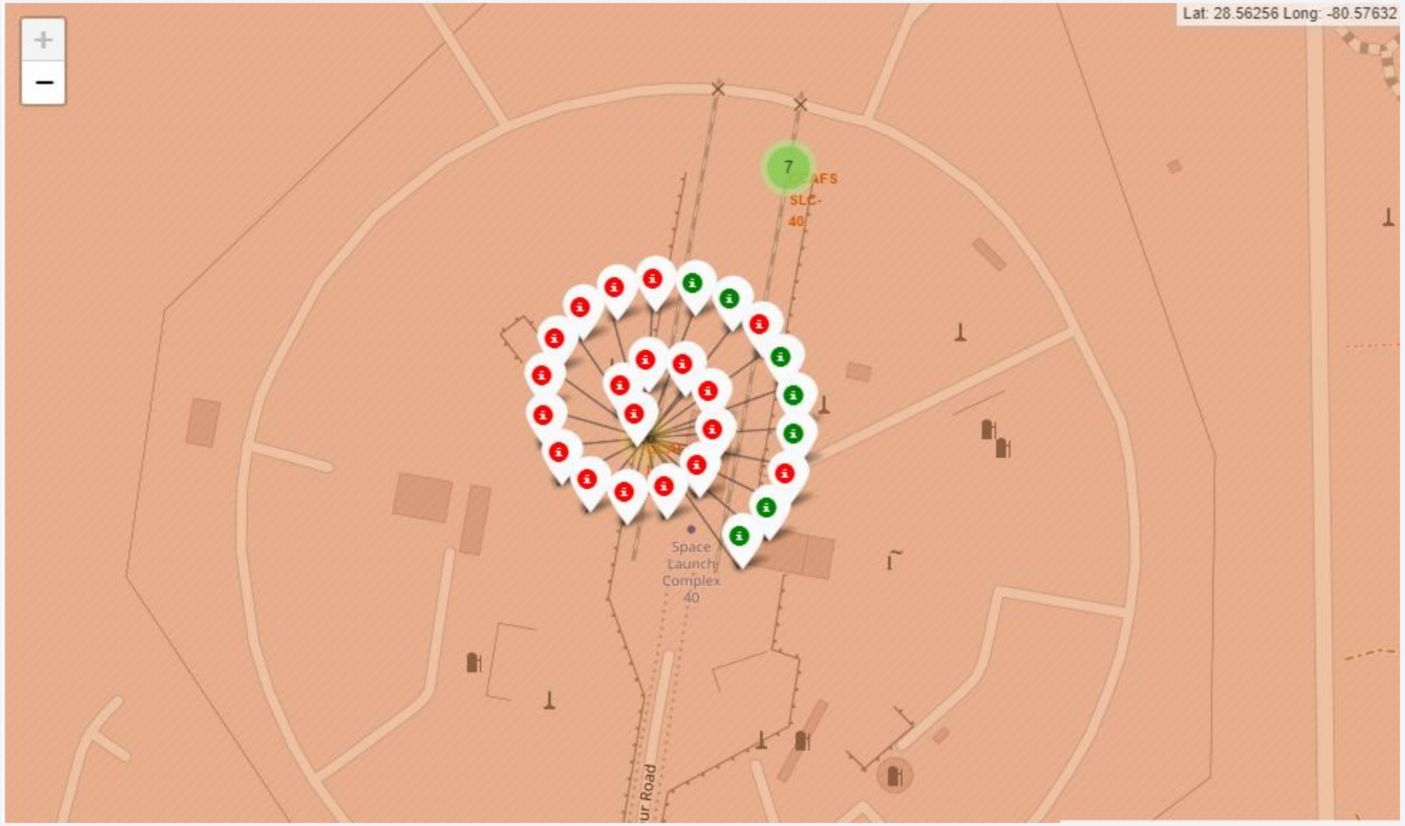
Explanation:

From the color-labeled markers we should be able to easily identify which launch sites have relatively high success rates.

Green Marker = Successful Launch

Red Marker = Failed Launch

Launch Site KSC LC-39A has a very high Success Rate.



Distance from the launch site KSC LC-39A to its proximities

Explanation:

From the visual analysis of the launch site KSC LC-39A we can clearly see that it is:

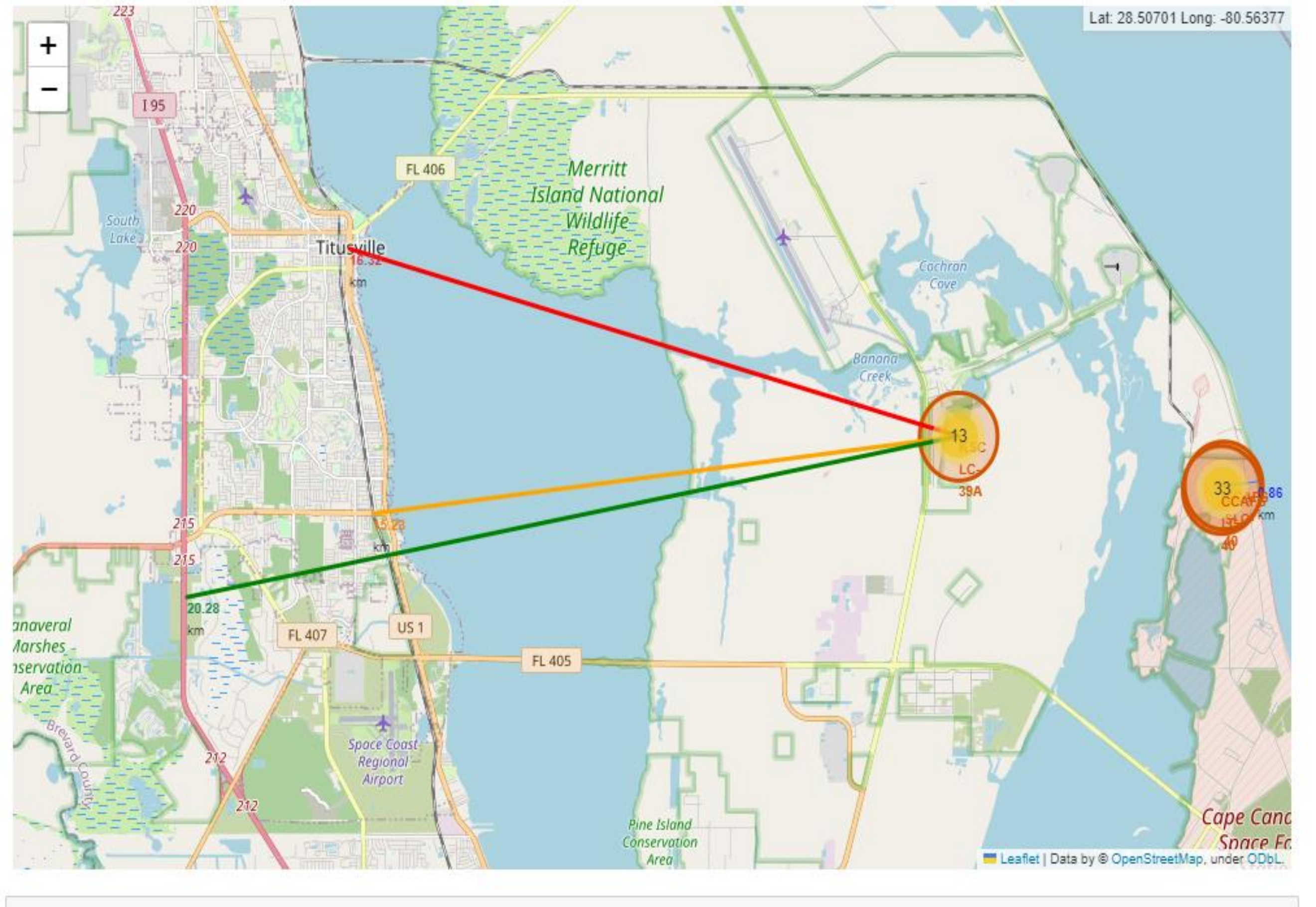
relative close to railway (15.23 km)

relative close to highway (20.28 km)

relative close to coastline (14.99 km)

Also the launch site KSC LC-39A is relative close to its closest city Titusville (16.32 km).

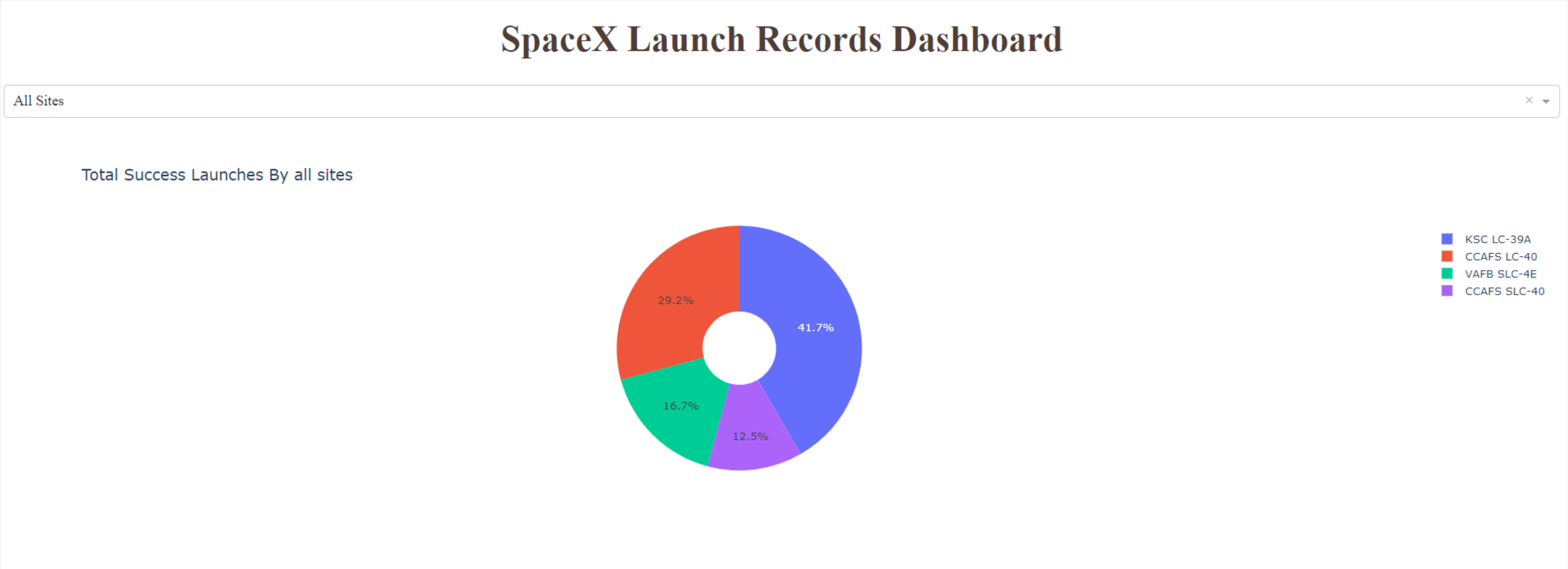
Failed rocket with its high speed can cover distances like 15-20 km in few seconds. It could be potentially dangerous to populated areas.



Section 4

Build a Dashboard with Plotly Dash

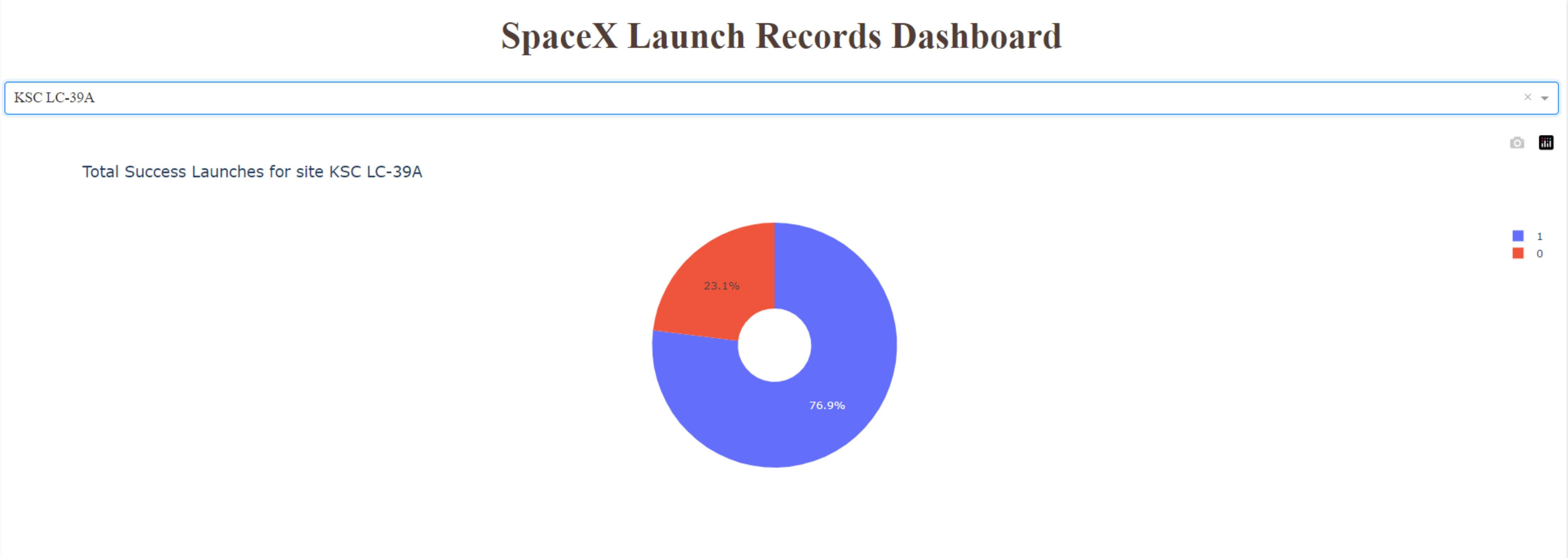
Launch success count for all sites



Explanation:

The chart clearly shows that from all the sites, KSC LC-39A has the most successful launches.

Launch site with highest launch success ratio



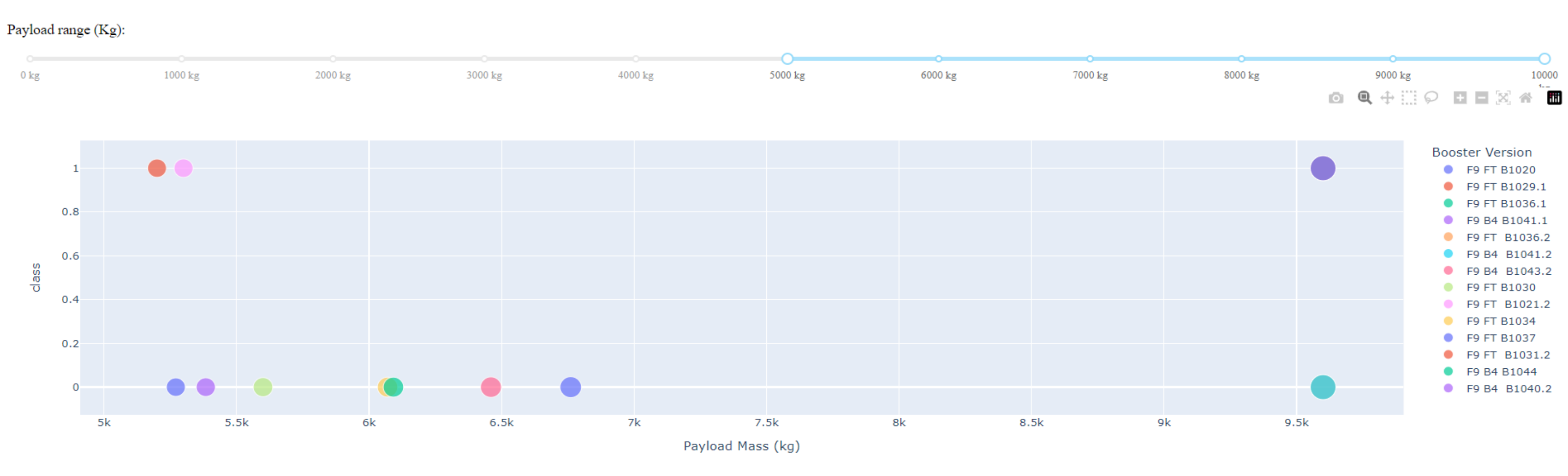
Explanation:

KSC LC-39A has the highest launch success rate (76.9%) with 10 successful and only 3 failed landings.

Payload Mass vs. Launch Outcome for all sites

Explanation:

The charts show that payloads between 2000 and 5500 kg have the highest success rate.



Section 5

Predictive Analysis (Classification)

Classification Accuracy

As we can see, by using the code as below: we could identify that the best algorithm to be the Tree Algorithm which have the highest classification accuracy.

TASK 12

Find the method performs best:

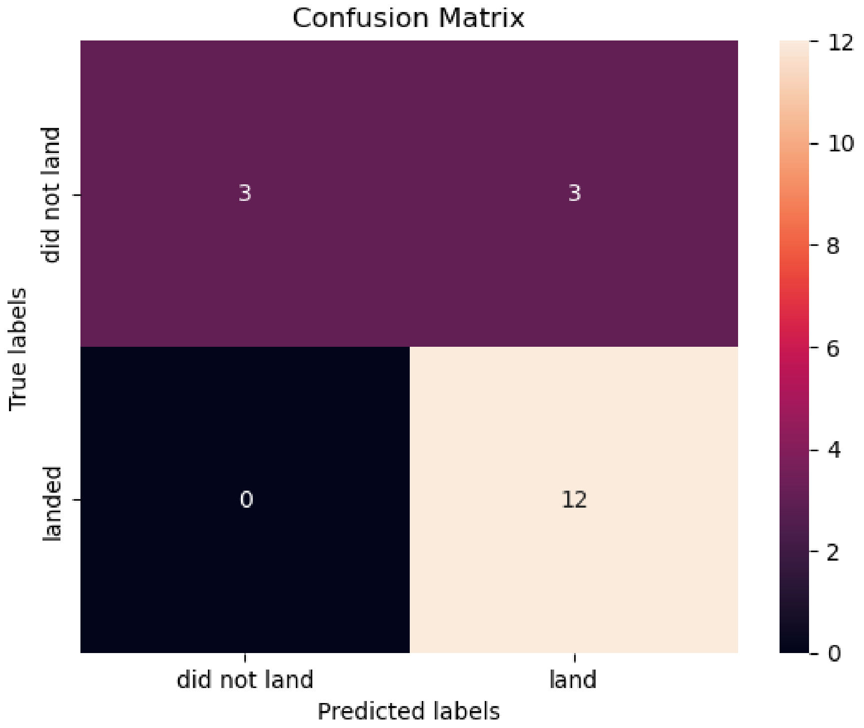
```
In [32]: algorithms = {"Logistic Regression":logreg_cv.best_score_,
                       "Support Vector Machine":svm_cv.best_score_,
                       "Decision Trees":tree_cv.best_score_,
                       "K-Nearest Neighbours":knn_cv.best_score_}

         best_algorithm = max(algorithms,key=algorithms.get)

         print(f'Best Algorithm is {best_algorithm} with a accuracy score of {algorithms[best_algorithm]}')

         Best Algorithm is Decision Trees with a accuracy score of 0.8767857142857143
```

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.

We can plot the confusion matrix

```
[26]: tree_yhat = tree_cv.predict(X_test)
plot_confusion_matrix(Y_test,tree_yhat)
```



Conclusions

- Decision Tree Model is the best algorithm for this dataset.
- Launches with a low payload mass show better results than launches with a larger payload mass.
- Most of launch sites are in proximity to the Equator line and all the sites are in very close proximity to the coast.
- The success rate of launches increases over the years.
- KSC LC-39A has the highest success rate of the launches from all the sites.
- Orbits ES-L1, GEO, HEO and SSO have 100% success rate.

Thank you!