IBM Data Science Capstone Project

Adam Scott

06 September 2021

SPACEX

Space Exploration Technologies

Outline

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

Executive Summary

Summary of methodologies

- Data collection
- Data wrangling
- EDA with data visualisation
- EDA with SQL
- Building an interactive map with Folium
- Building a dashboard with Plotly Dash
- Predictive analysis (classification)

Summary of all results

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

Introduction

Project background and context

SpaceX advertises Falcon 9 rocket launches on its website, with a cost of 62 million dollars; other providers cost upward of 165 million dollars each. Much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch.

The aim of this project is to predict if the Falcon 9 first stage will land successfully. This will be done by closely analysing and comparing the different factors and variables which may have an impact on the rocket landing success rate, factors including payload mass, orbit type, launch site, and flight number (continuous launch attempts).

Instead of using rocket science to determine if the first stage will land successfully, I will train a number of different machine learning models and use public information to predict if SpaceX will reuse the first stage.

Methodology

Methodology

Data collection methodology:

- SpaceX Rest API
- Web scraping from ***Wikipedia***
- Parse the scraped HTML tables and convert into a Pandas DataFrame ready for analysis

Data wrangling (transforming data for machine learning)

- One Hot Encoding data fields for machine learning
- Drop irrelevant columns and replace null values with the mean value of their respective column

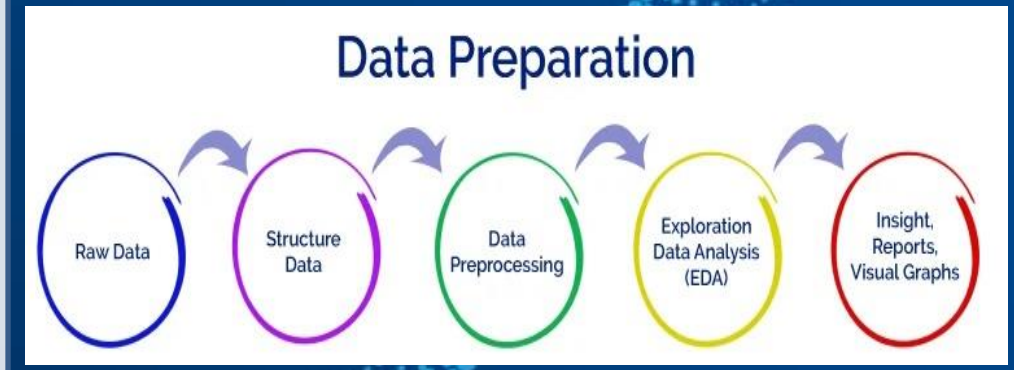
Exploratory Data Analysis (EDA) using visualisation and SQL

- Plotted scatter, bar, and line graphs to compare the relationships between independent variables and identify any patterns in the data

Performed interactive visual analytics using Folium and Plotly Dash

Performed predictive analysis using classification models

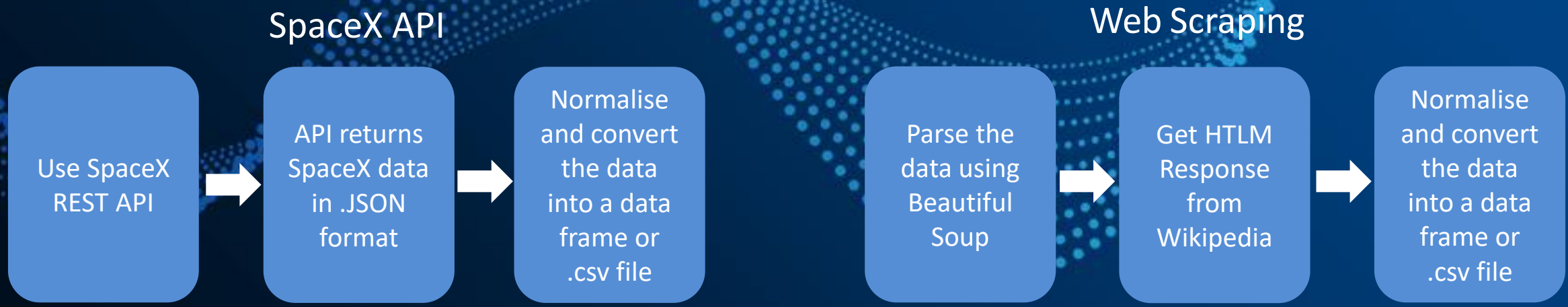
- How to build, tune, evaluate classification models



Methodology

How the datasets were collected:

- Using SpaceX REST API I was able to gather the SpaceX launch data, including information about the different rockets used (Falcon 1 and Falcon 9), the payloads carried, launch and landing specifications, and landing outcomes
- Analysing this data will help to achieve the goal of predicting whether SpaceX Falcon 9 Stage 1 launches will successfully land or not
- The SpaceX REST API endpoints, or URL, starts with api.spacexdata.com/v4/
- Another popular method for obtaining Falcon 9 Launch data is by web scraping Wikipedia using Beautiful Soup



Data collection – SpaceX API

1 . Get Response object from

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
response = requests.get(spacex_url)
```

2. Converting Response to a .json file

```
response_json = requests.get(static_json_url).json()
data = pd.json_normalize(response_json)
```

3. Apply custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
getBoosterVersion
```

4. Assign list to dictionary then data frame

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}
```

```
data = pd.DataFrame(launch_dict)
```

5. Filter the data frame and export to .csv file

```
data_falcon9 = data[data.BoosterVersion == 'Falcon 9']
```

```
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, mean)
```

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

GitHub URL to Notebook

Data collection – Web Scraping

1 .Get Response object from HTML

```
response = requests.get(static_url)
```

2. Creating BeautifulSoup Object

```
soup = BeautifulSoup(response.content, 'html.parser')
```

3. Find all the <table> elements

```
html_tables = soup.find_all('table')
```

4. Getting column names

```
column_names = []
table_header = first_launch_table.find_all('th')

for row in table_header:
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

5. Creation of dictionary

```
launch_dict= dict.fromkeys(column_names)
del launch_dict['Date and time ( )']

launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
launch_dict
```

6. Appending data to keys (refer) to notebook block

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table he
```

7. Converting dictionary to data frame

```
df = pd.DataFrame.from_dict(launch_dict)
```

8. Data frame to .csv file

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub URL to Notebook

Data Wrangling

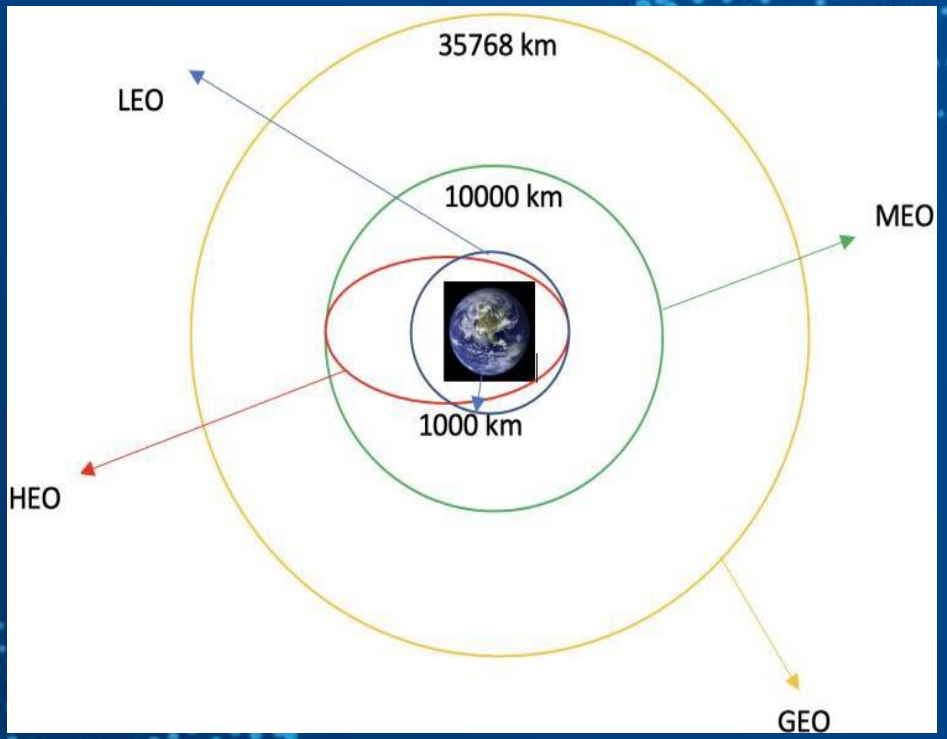
Introduction

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. There are 3 different landing locations for the boosters, each having either a True (successful) or false (failed) outcome:

1. Ocean
2. RTLS – ground pad
3. ASDS – drone ship

I have converted those landing outcome values into Training Labels: 1 means the booster successfully landed and 0 means it was unsuccessful.

Each launch aims to reach a certain orbit. Here are some common orbit types:



Perform Exploratory Data Analysis (EDA) on dataset

Calculate the number of launches at each site

Calculate the number and occurrence of each orbit

Calculate the number and occurrence of mission outcome per orbit type

Export dataset as .csv file

Create a landing outcome label from Outcome column

Work out success rate for every landing in dataset

GitHub URL to Notebook

EDA with Data Visualization

Scatter graphs being drawn:

Flight Number vs Payload Mass

Flight Number vs Launch Site

Payload Mass vs Launch Site

Flight Number vs Orbit Orbit VS. Payload Mass

Scatter plots show how much one variable is affected by another. The relationship between two variables is called their correlation. Scatter plots usually consist of a large body of data.

Bar graph being drawn:

Orbit vs Mean Success Rate (Class)

A bar diagram makes it easy to compare sets of data between different groups at a glance. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time.

Line graph being drawn:

Year vs Mean Success Rate (Class)

Line graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded.

GitHub URL to Notebook

EDA with SQL

Below are a list of questions I was able to answer by running different SQL queries to gather the required datasets:

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string ‘CCA’
- Display the total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass
- 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
- Rank the successful landing_outcomes between the date 2010-06-04 and 2017-03-20 in descending order

GitHub URL to Notebook

Build an interactive map with Folium

To visualise the Launch Data into an interactive map, I took the latitude and longitude coordinates of each launch site and added a circle marker around each site with a label of the site's name.

I assigned 2 different colors to the landing outcome of the launches:
- Green for launches which successfully landed (class 1)
- Red for launches which failed to land (class 0)

I was able to create a marker cluster object using the MarkerCluster plugin to show all the markers on the map in a simplified and clear display

Using Haversine's formula I calculated the distance from the Launch Site to various landmarks to find various trends about what is around the Launch Site to measure patterns. Lines are drawn on the map to measure distance to landmarks.

Examples of some trends in which the Launch Site locations are situated:
• Close proximity to railways - No
• Close proximity to highways - No
• Close proximity to coastline? - Yes
• Certain distance away from cities? - Yes

GitHub URL to Notebook

Build an interactive dashboard with Plotly Dash

Using **Plotly Dash** I was able to create an interactive dashboard with pie charts and scatter graphs to better analyse and explain the data. Below is a summary of the graphs created and how they can be used interactively to provide a closer look at the data we have been working with.

Pie chart showing the comparison between the total successful launches for all sites as well as the success vs failure of launches for each individual site.

From these charts I can determine the following:

- Site KSC LC-39A has the highest number of successful launches
- Site KSC LC-39A has the highest launch success rate at 76.9%, whereas site CCAFS LC-40 has the lowest launch success rate at 26.9%

Scatter graph showing the relationship between the Launch Outcome (class 1 or 0) and Payload Mass (Kg) for the different Booster Versions.

From this graph I can determine the following:

- The majority of the launches were in the payload range between 2000kg and 4000kg
- Booster Version F9 B5 has the highest launch success rate of 100%, however, only 1 launch for this version has been registered
- Booster Version F9 FT has the most launches with a total of 21 launches at a success rate of 67%

GitHub Link to Notebook

Predictive analysis (Classification)

BUILDING MODEL

- Load our dataset into NumPy and Pandas
- Data wrangling – transform and standardise the data
- Split our data into training and test data sets
- Check how many test data samples there are
- Use 4 different machine learning algorithms: Logistic Regression, Decision Trees, k-nearest neighbors (KNN), and Support Vector Machines (SVM)
- Use GridSearchCV to test the parameters of each classification algorithm to find the best parameters
- Fit the datasets into the GridSearchCV objects and train the dataset

EVALUATING MODEL

- Check accuracy for each model
- Get tuned hyperparameters for each type of algorithm
- Plot Confusion Matrix

IMPROVING MODEL

- Feature engineering
- Algorithm tuning

FINDING THE BEST PERFORMING CLASSIFICATION MODEL

- The model with the highest accuracy score wins the best performing model
- The accuracy (score) of the best algorithm is shared at the end of the notebook along with a dictionary of the best parameters for that algorithm

GitHub Link to Notebook

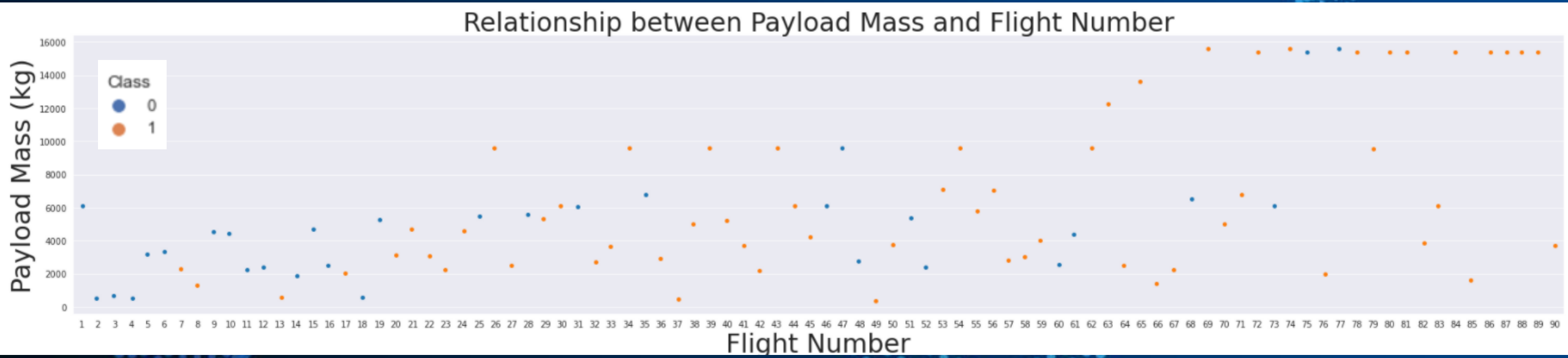
Results

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

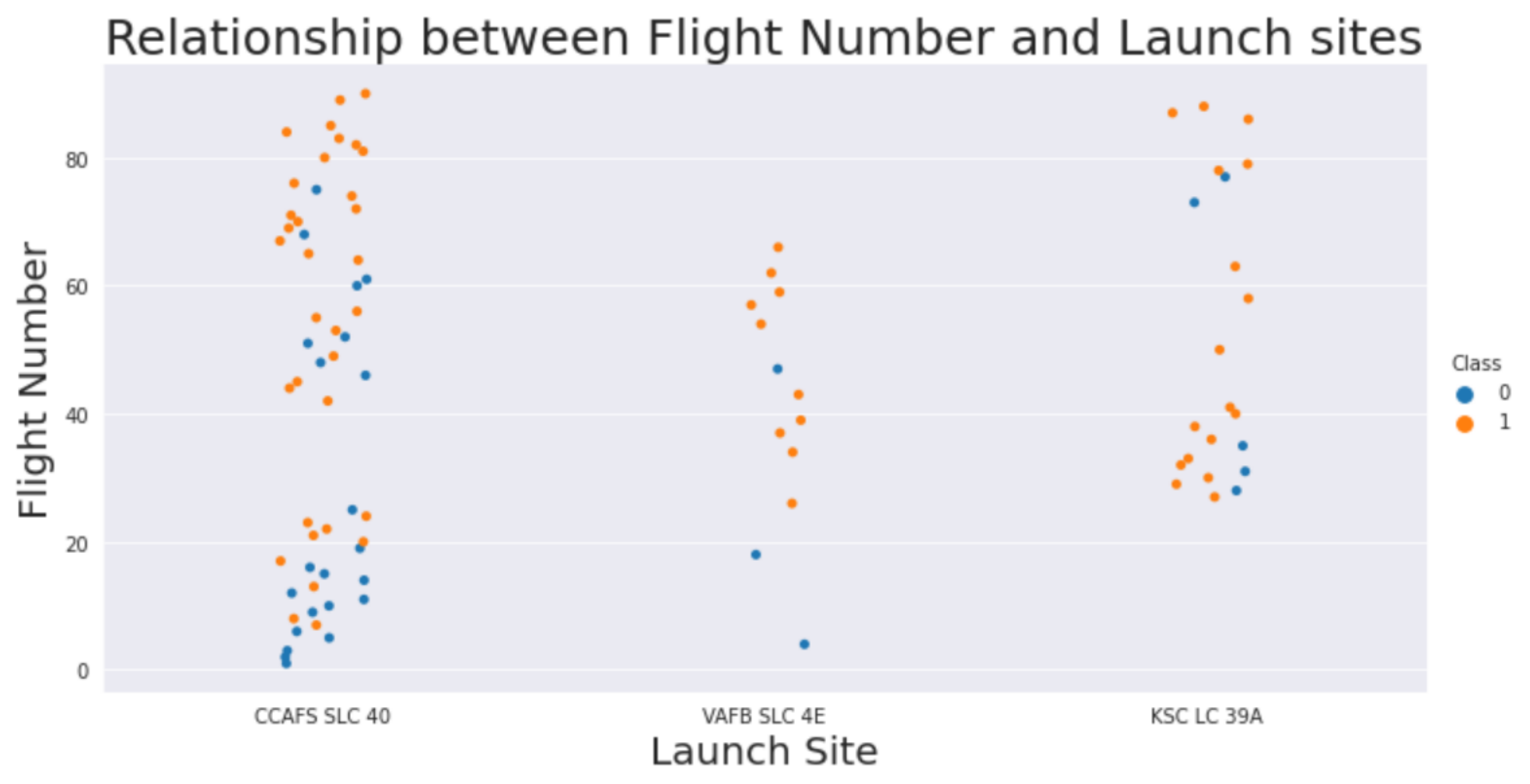
EDA with Visualisation

Flight Number vs. Payload Mass

This graph shows that as the number of flights increase, so does the success rate for the landing of the first stage. It’s not clear to say the same for payload mass with this graph.



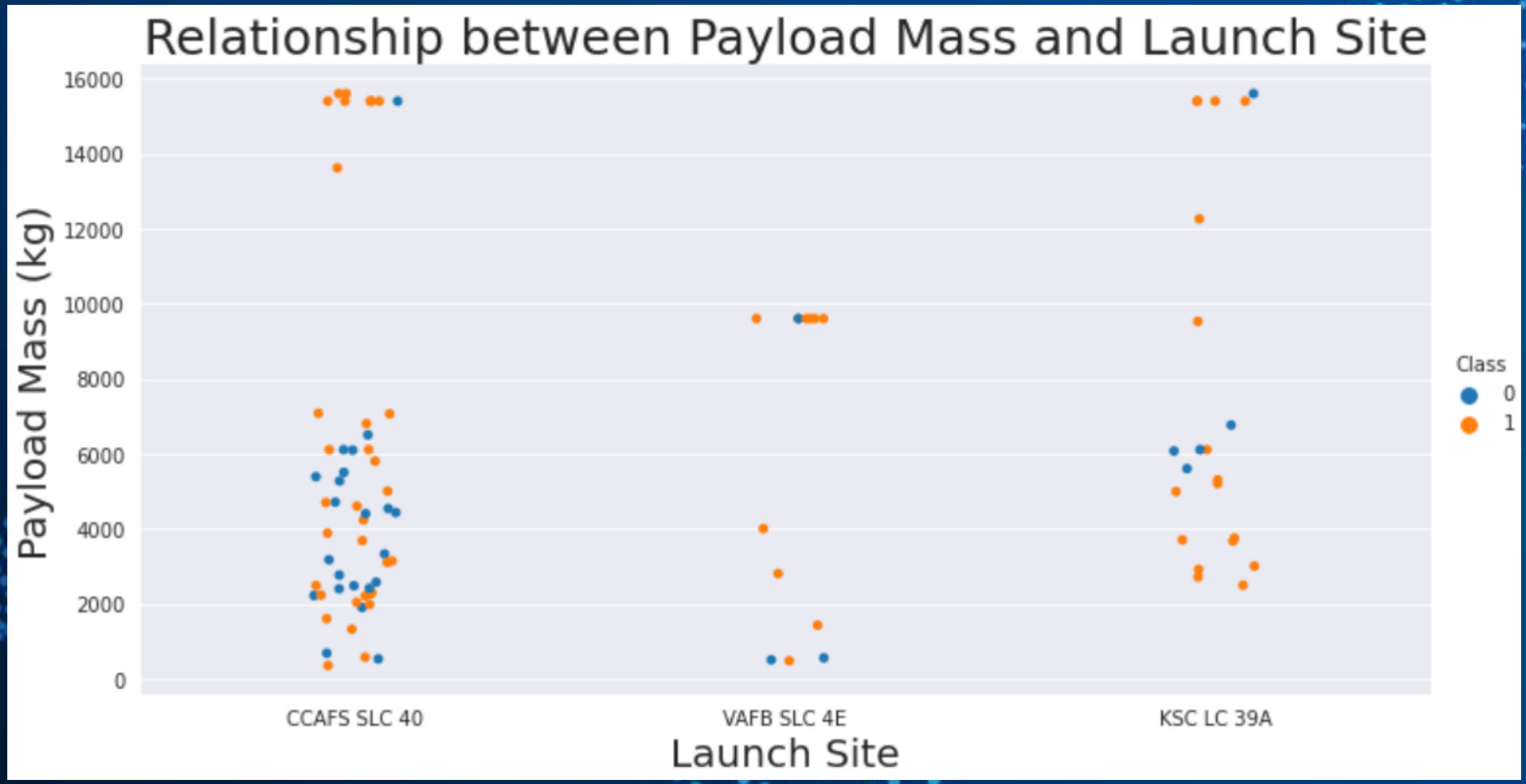
Flight Number vs. Launch Site



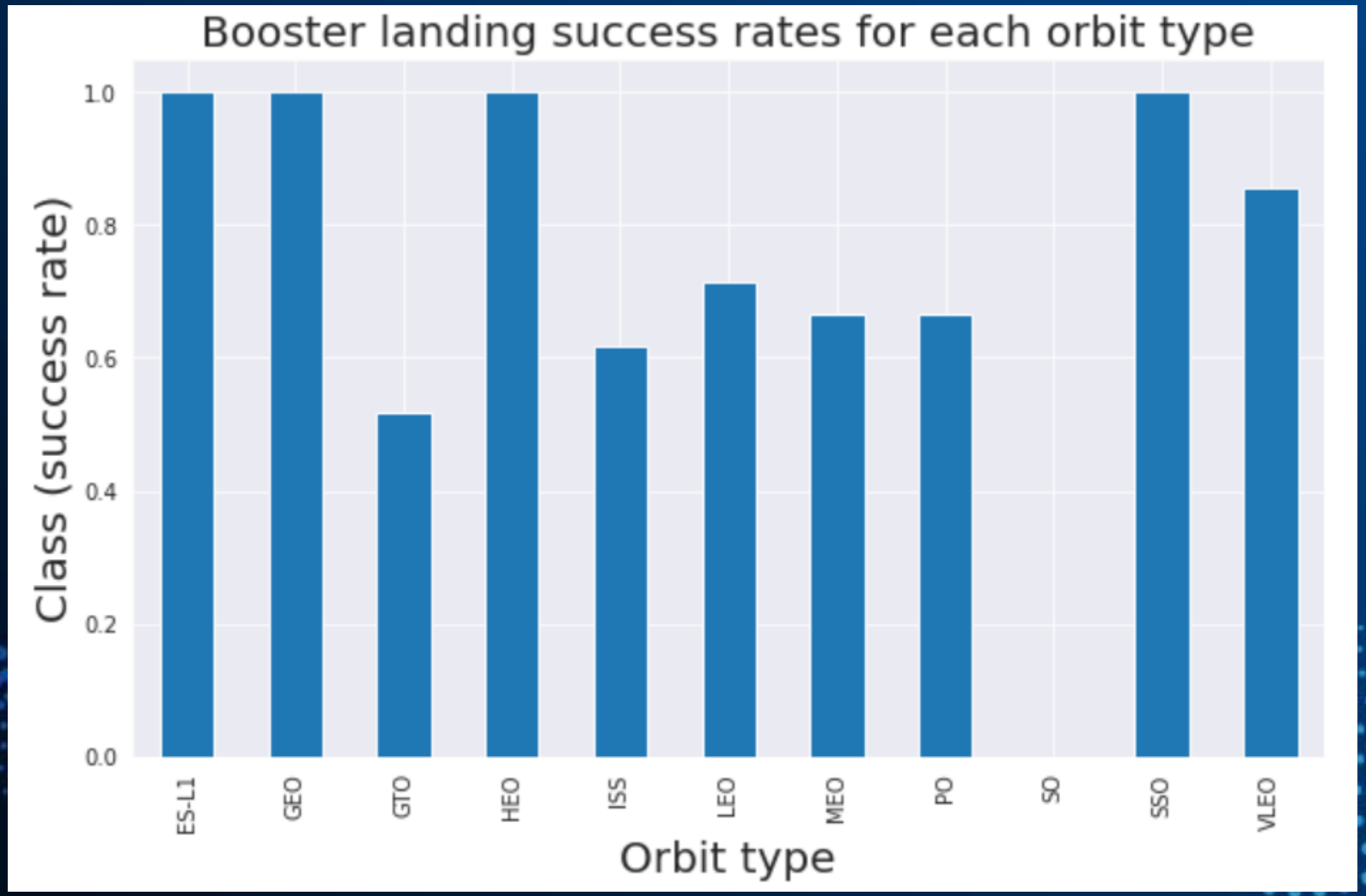
This graph shows the landing success rate improves at each launch site as the number of flights are increased

Payload Mass vs. Launch Site

This graph shows launch site CCAFS SLC 40 has a high launch success rate when the payload is very high, around 15000kg. There isn't enough data samples for each of the sites across a broad range of payload mass values to provide a definitive answer as to whether either launch sites are dependent on payload mass for a successful launch.

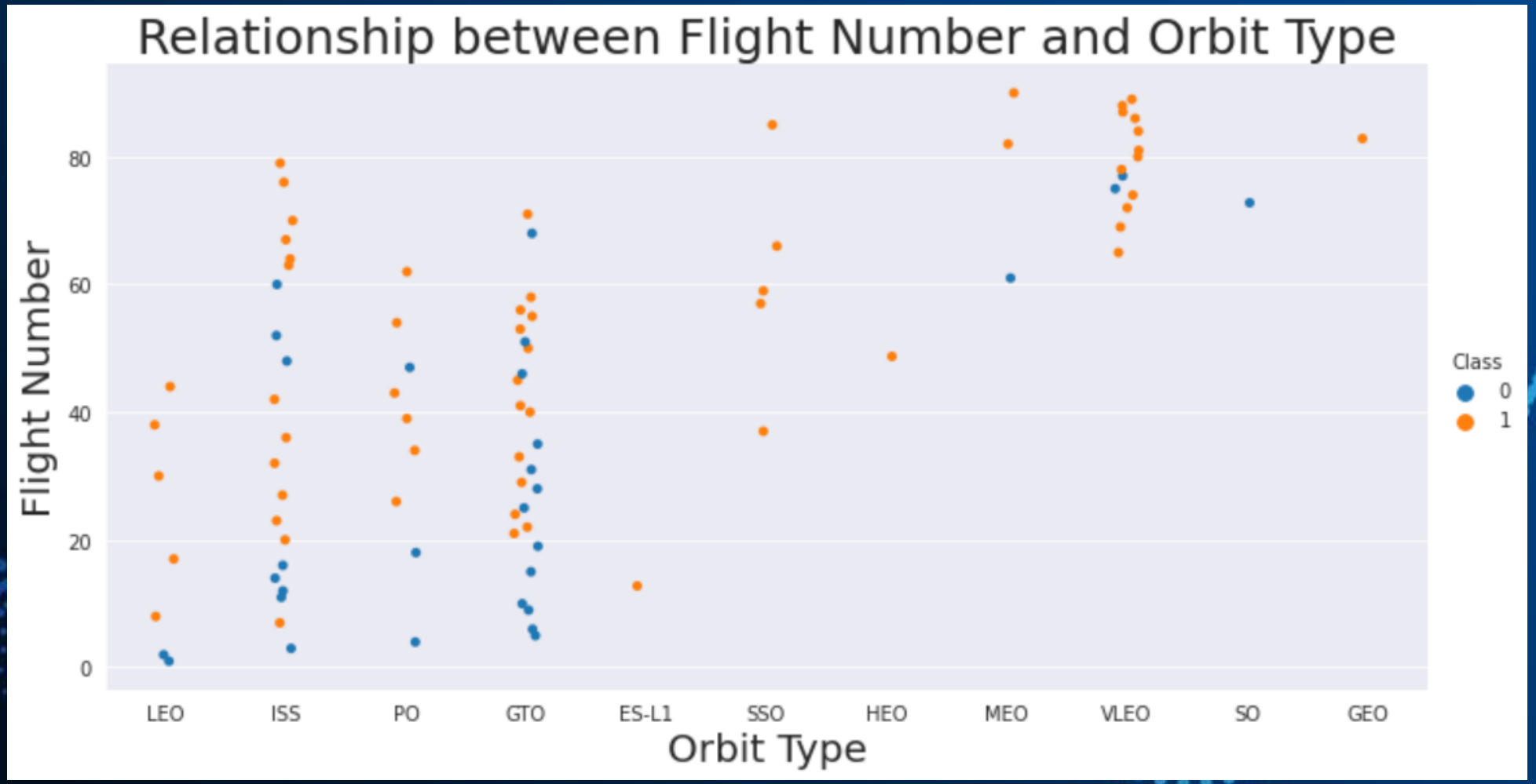


Success rate vs. Orbit type



This graph shows that orbit types ES-L1, GEO, HEO, and SSO all have the best success rate of 100%

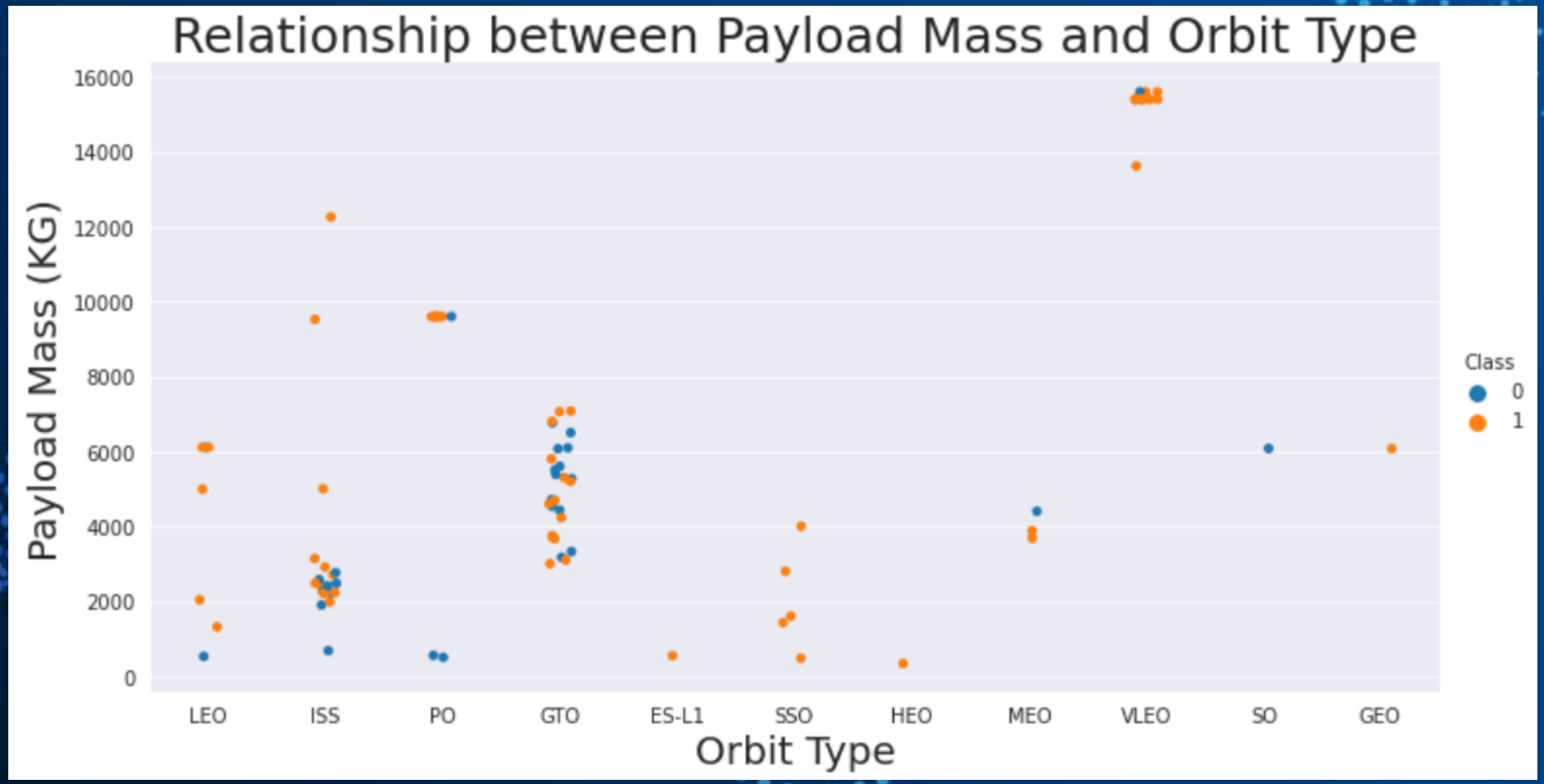
Flight Number vs. Orbit type



This graph suggests that as flight numbers increase for most of the launch sites, the launch success rate increases. This isn't clear for all sites since some sites don't have many data samples.

Payload vs. Orbit type

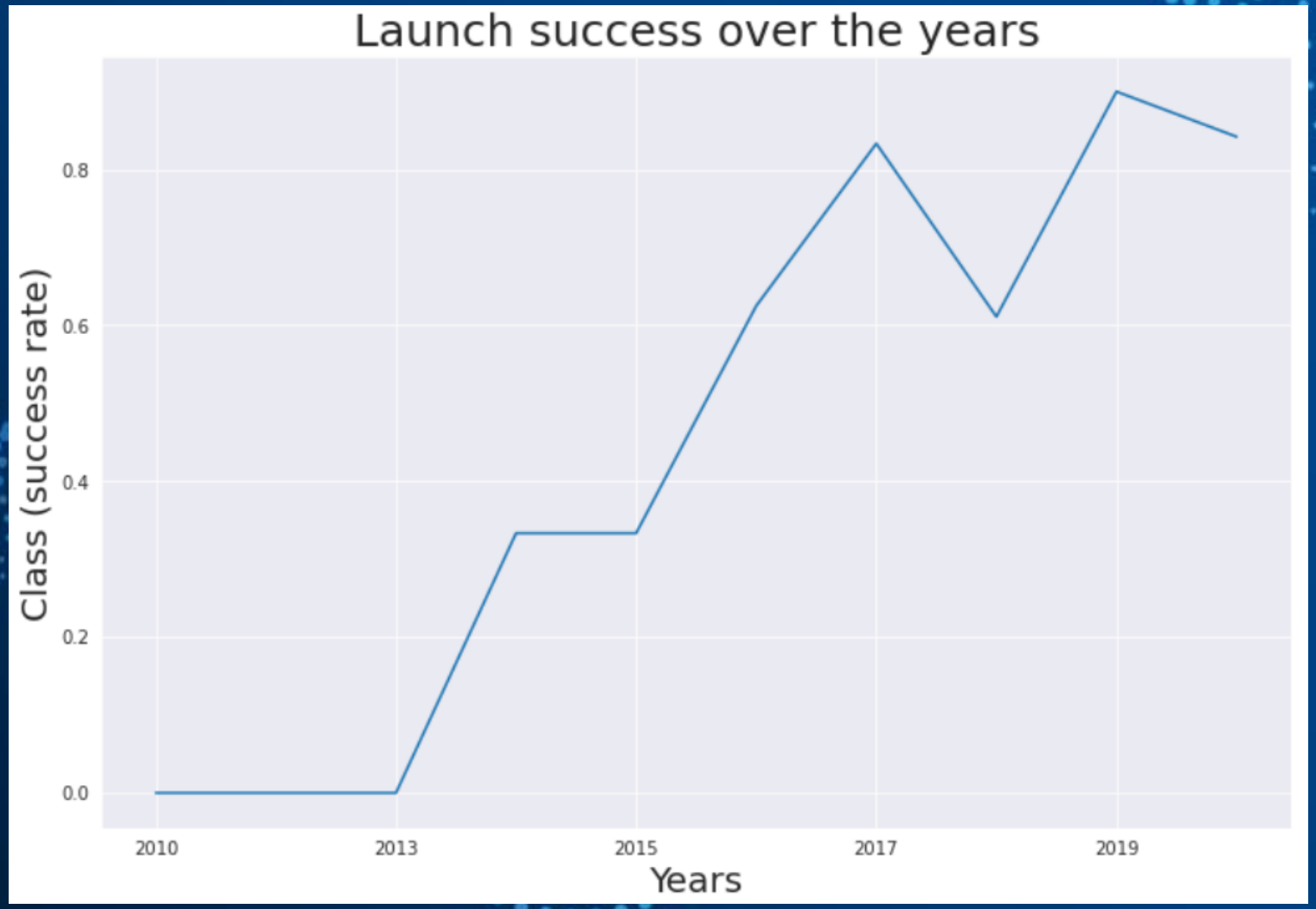
It’s not clear to show the relationship between payload mass and launch success rate for the different orbit types since there isn’t a broad range of payload mass values used for each orbit.

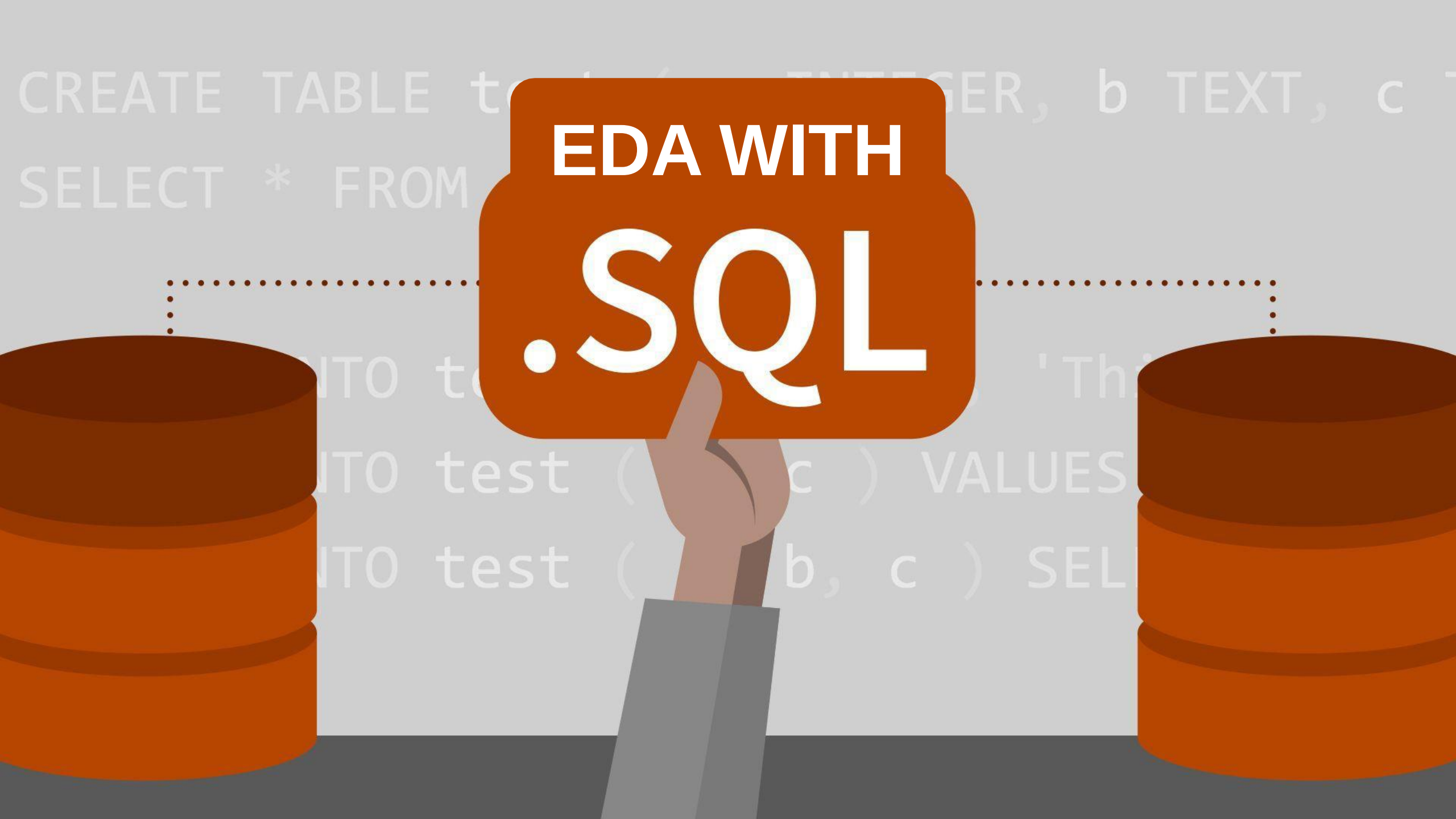


Launch success yearly trend

This graph clearly shows that the success rates over the years have been increasing, with slight drops in 2018 and 2020.

There were no successful launches until 2014.




EDA WITH
.SQL

Unique Launch Sites

SQL QUERY

SELECT DISTINCT Launch_Site as "Unique_Launch_Sites" from SPACEXTBL

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

QUERY EXPLAINATION
Using **DISTINCT** in the query means it will only show the Unique values in the Launch_Site column from spacextbl.

5 Launch site names beginning with 'CCA'

SQL QUERY

select * from spacextbl where launch_site like 'CCA%' limit 5

QUERY EXPLAINATION

Limit 5 in the query means it will only show 5 records from *spacextbl*. ***LIKE*** keyword with *'CCA'* followed by '%' indicates that the *Launch_Site* name must start with *CCA*.

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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 by customer NASA (CRS)

SQL QUERY

```
select sum(payload_mass__kg_) as "Total_Payload_Mass_NASA_CRS" from spacextbl
where customer='NASA (CRS)'
```

total_payload_mass_nasa_crs
45596

QUERY EXPLAINATION

Using the function ***SUM*** calculates the total in the column *PAYLOAD_MASS__KG_*

The ***WHERE*** clause filters the dataset to only collect results where customer = *NASA (CRS)*

Average Payload Mass carried by booster version F9 v1.1

SQL QUERY

```
select avg(payload_mass__kg_) as "Avg_Payload_Mass_F9_v1_1" from spacextbl
where booster_version='F9 v1.1'
```

avg_payload_mass_f9_v1_1
2928

QUERY EXPLAINATION

Using the function ***AVG*** calculates the average in the column *PAYLOAD_MASS__KG_*

The ***WHERE*** clause filters the dataset to only perform calculations for *Booster_version F9 v1.1*

The date of the first successful landing outcome in ground pad

SQL QUERY

```
select min(date) as "First_Successful_Ground_Pad_Landing" from spacextbl where Landing__Outcome='Success (ground pad)'
```

first_successful_ground_pad_landing
2015-12-22

QUERY EXPLAINATION

Using the function ***MIN*** works out the earliest/minimum date in the column *Date.*

The ***WHERE*** clause filters the dataset to only perform calculations on *Landing_Outcome Success (ground pad).*

Successful drone ship landing with payload between 4000 and 6000kg

SQL QUERY

```
select distinct booster_version from spacextbl where landing__outcome='Success (drone ship)'
and payload_mass__kg_ between 4000 and 6000
```

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

QUERY EXPLAINATION

The ***WHERE*** clause filters the dataset to show only Landing_Outcome = Success (drone ship).

The ***AND*** clause specifies additional filter conditions Payload_MASS_KG_ > 4000 AND Payload_MASS_KG_ < 6000.

Total Number of Successful and Failure Mission Outcomes

SQL QUERY

```
Select
count(CASE when Mission_Outcome LIKE '%Success%' then 1 end) as Successful_Mission_Outcomes,
count(case when Mission_Outcome LIKE '%Failure%' then 1 end) as Failed_Mission_Outcomes
from spacextbl
```

successful_mission_outcomes	failed_mission_outcomes
100	1

Total Number of Successful and Failure Mission Outcomes (simplified)

SQL QUERY

```
select Mission_Outcome, count(*) from spacextbl group by Mission_Outcome
```

mission_outcome	2
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

QUERY EXPLAINATION
Count(*) to show the total number of successful and failed mission outcomes.

Boosters which carried maximum payload

SQL QUERY

```
select booster_version, PAYLOAD_MASS__KG_ as Payload_Mass from
spacextbl where payload_mass__kg_ = (
        select max(payload_mass__kg_) from spacextbl)
```

QUERY EXPLAINATION
Using a **nested** query to extract the **MAX** payload mass value to show which booster versions carried that max value.

booster_version	payload_mass
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

SQL QUERY

```
select monthname(date) as Month, landing__outcome, booster_version, launch_site from spacextbl
where landing__outcome='Failure (drone ship)' and year(date)=2015
```

MONTH	landing__outcome	booster_version	launch_site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

QUERY EXPLAINATION

select ***monthname(date)*** extracts the name of the month from the values in the date column.

Year(date)=2015 extracts only those launches which occurred in the year 2015.

Rank success count between 2010-06-04 and 2017-03-20 in descending order

SQL QUERY

select * from spacextbl where landing__outcome like 'Success%' and date between '2010-06-04' and '2017-03-20' ORDER BY date DESC

QUERY EXPLAINATION
Order by date desc lists the values in descending order by date

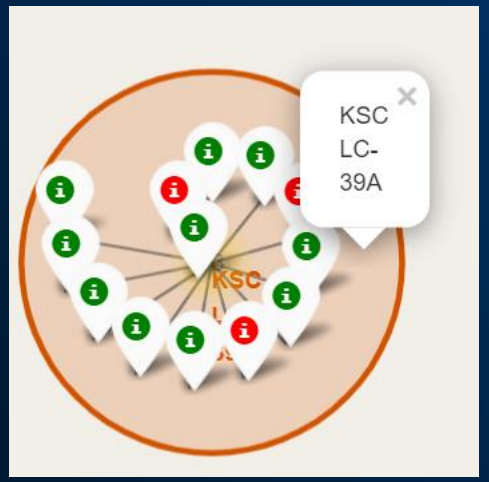
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

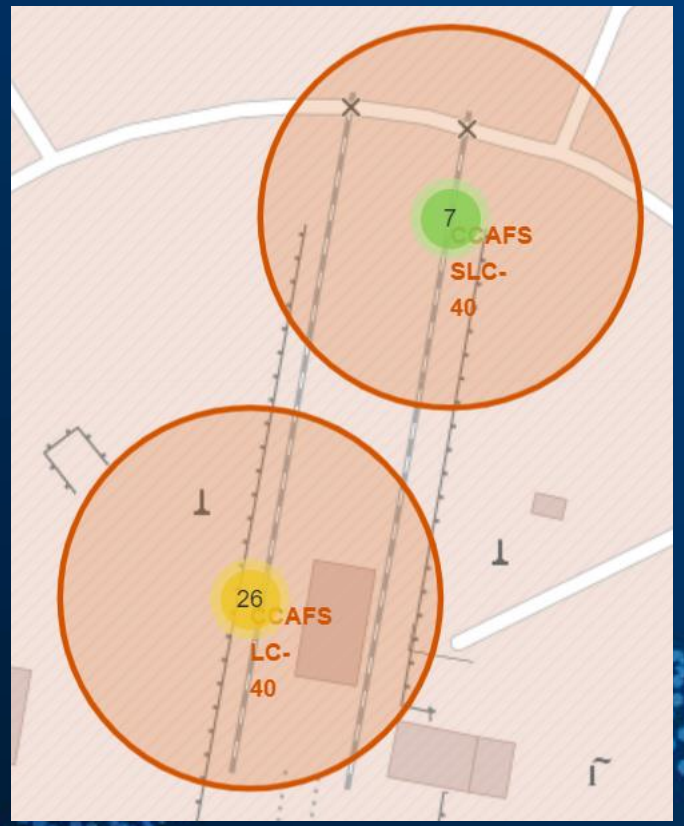
Interactive map with Folium

All launch sites global map markers

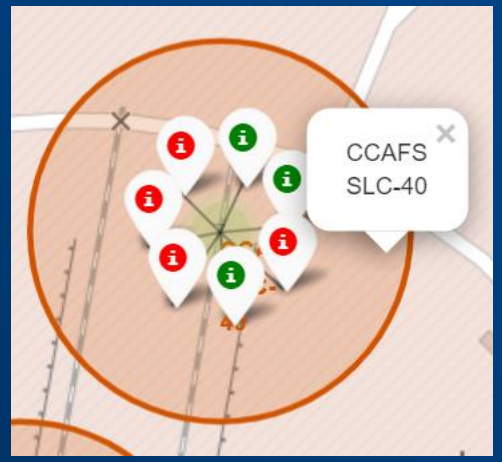
We can see that the SpaceX launch sites are all on the coasts of the United States of America, in Florida and California

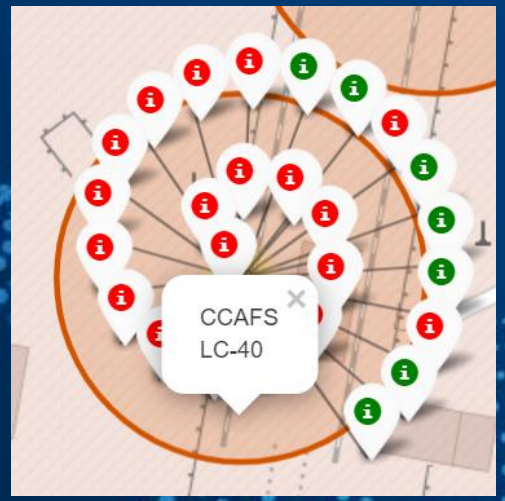
Colour Labelled Markers

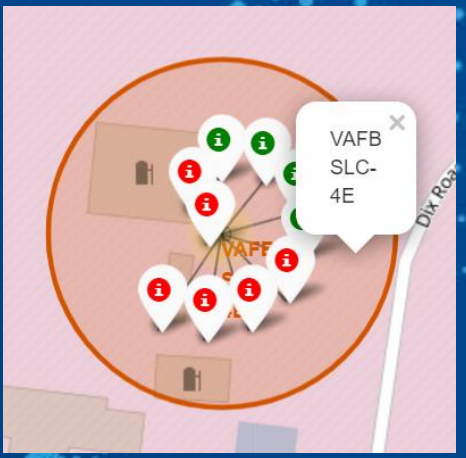




Florida Launch Sites







California Launch Site

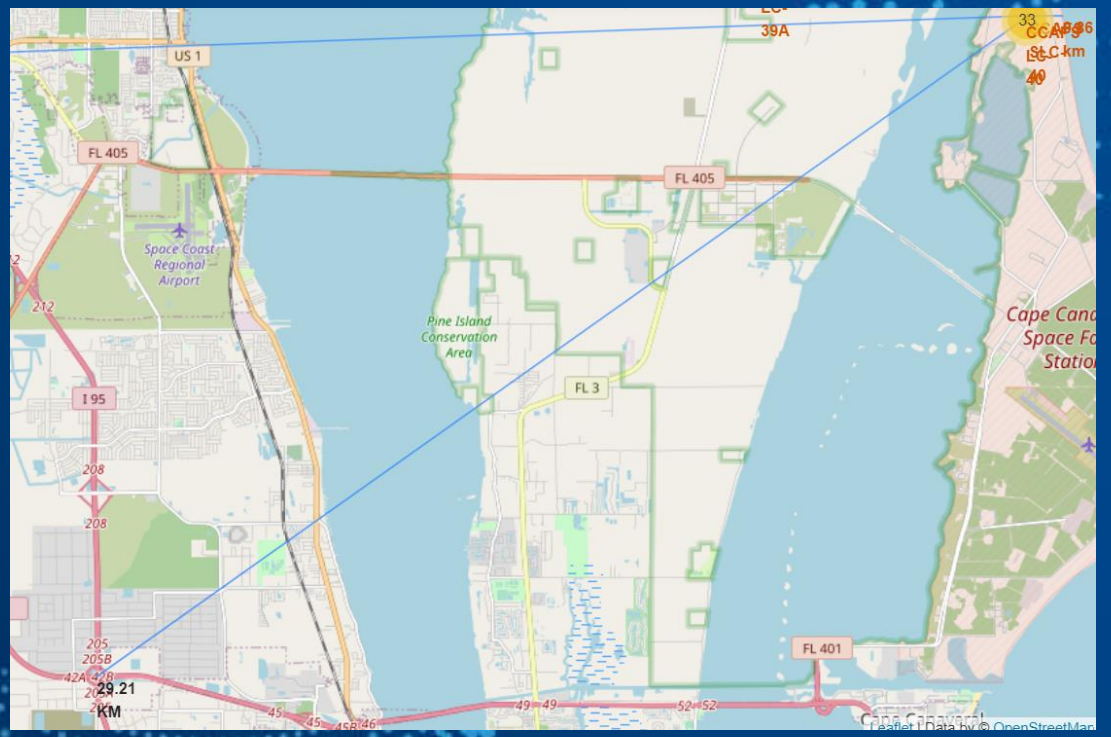
Green Markers show launches with successful landing outcomes.
Red Markers show launches with failed landing outcomes.

Based on these screenshots of clustered markers, we can clearly see that KSC LC-39A has the highest landing outcome success rate.

Working out Launch Sites distance to landmarks to find trends with Haversine formula using CCAFS-SLC-40 as a reference

Distance to coastline

Distance to city

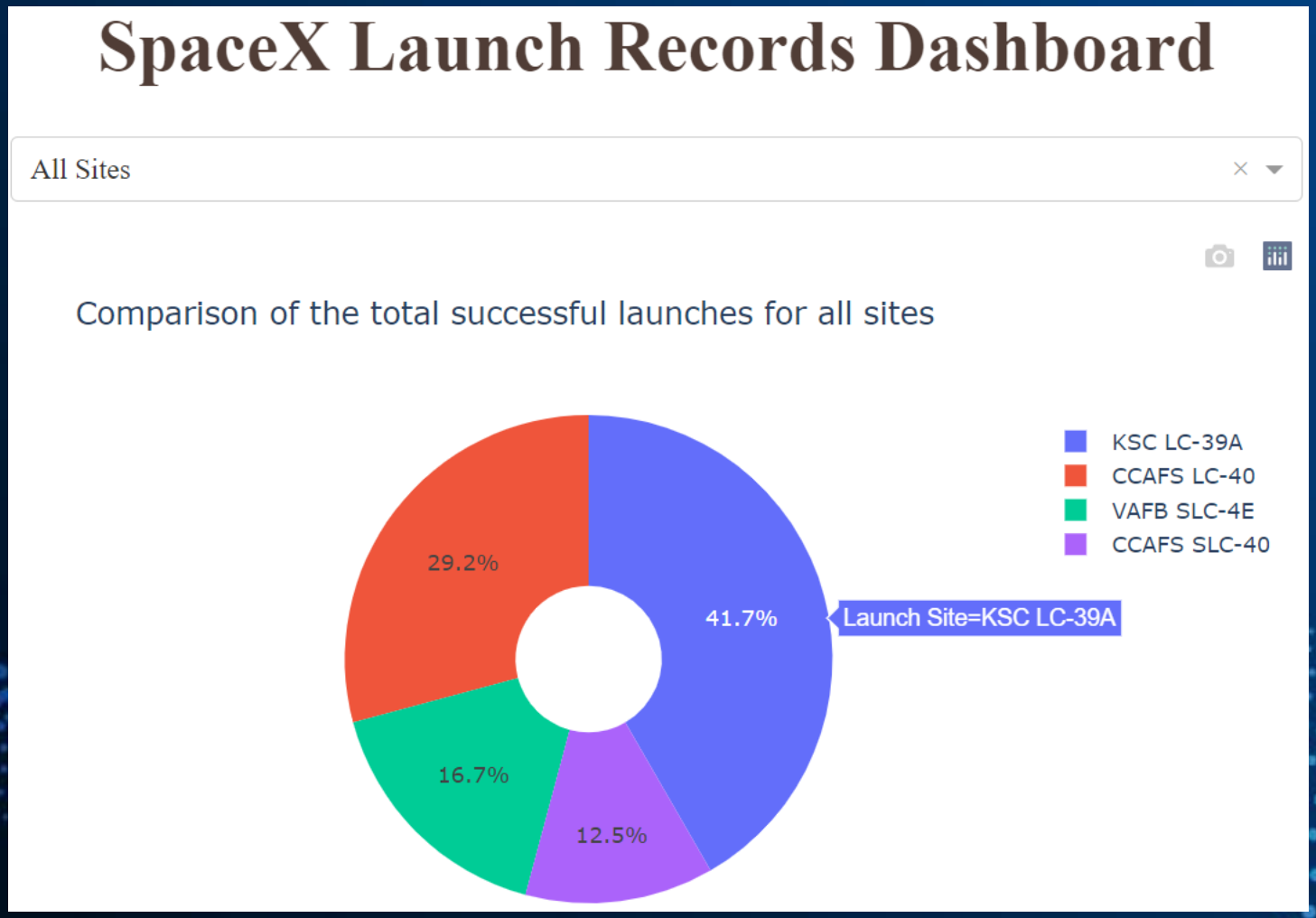


Distance to highway

- 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

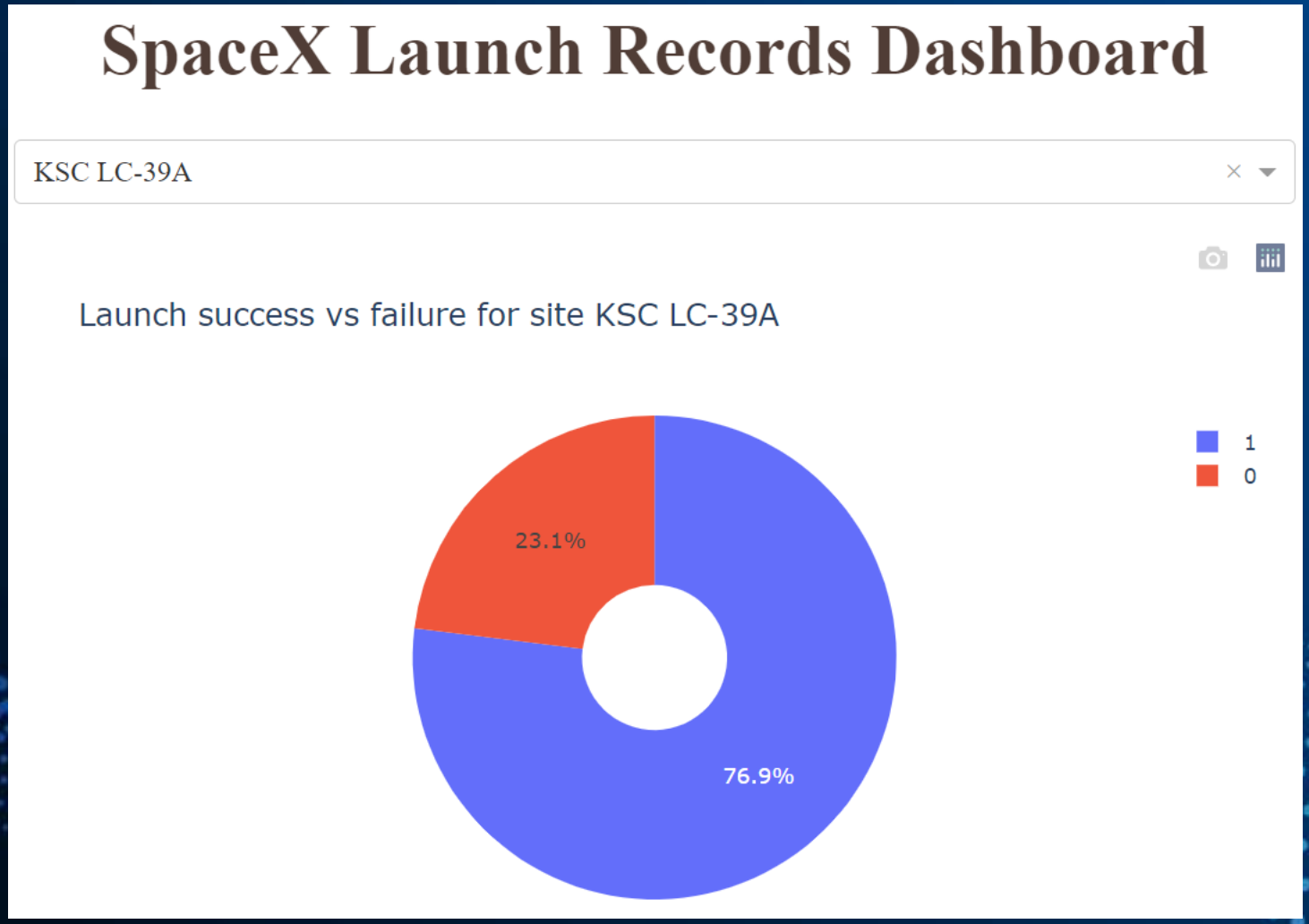
Dashboard with plotly | Dash

DASHBOARD – Pie chart showing the success percentage achieved by each launch site



It's clear to see that KSC LC-39A had the most successful launches from all the sites

DASHBOARD – Pie chart for the launch site with highest launch success ratio



KSC LC-39A achieved a 76.9% launch success rate and had a 23.1% failure rate

Slice labels follow the values in the Class column. 1 = success, 0 = failure

DASHBOARD – Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Payload range 3000kg – 4000kg

This range shows the highest success rate of any of the 1000kg ranges, with a success rate of 70% (7 successful, 3 failed)

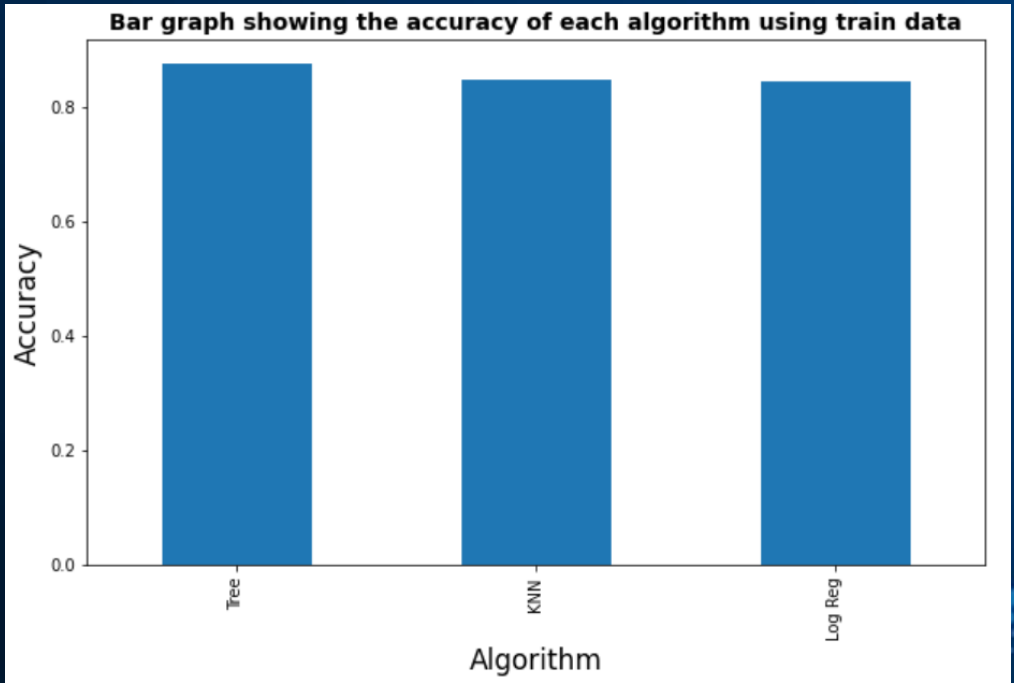
DASHBOARD – Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider

Payload range 9000kg – 10000kg

This plot shows only 2 different results, successful and failed launches at 9600kg. However, the legend shows there were 5 different booster versions used for launches within this payload range. This is because all 5 launches were carried out with a payload mass of 9600kg and this scatter plot doesn't show multiple results carried out with the same payload mass and the same landing outcome (success/failed). An improvement would be to use clustering when there are identical results for different booster versions.

Predictive analysis (Classification)

Classification Accuracy - using the test and train dataset



I was then able to select the best hyperparameters for the ***Decision Tree Classifier***, *and* using the validation/test data, achieved 83.33% accuracy.

```
print("accuracy :",tree_cv.score(X_test, Y_test))
```

```
accuracy : 0.8333333333333334
```

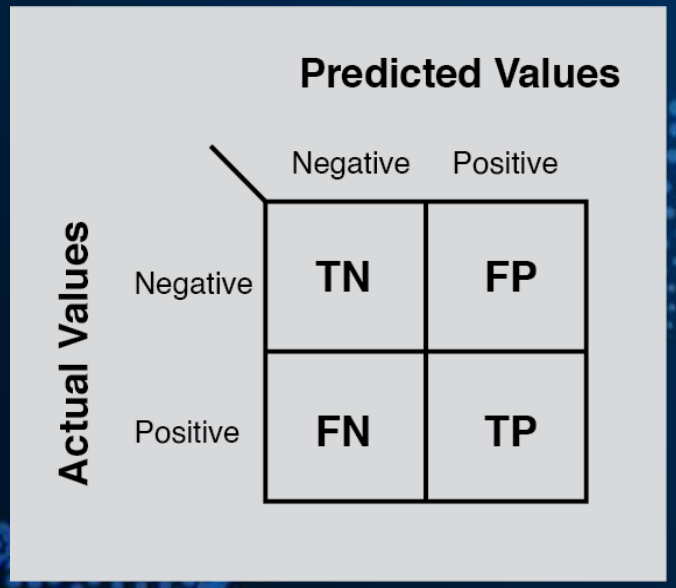
After splitting the data into train and test datasets, I was able to check the accuracy of each of the models and found them to be very similar. The winning algorithm is the ***Decision Tree Classifier (Tree)*** with an accuracy of 0.875 on the training dataset.

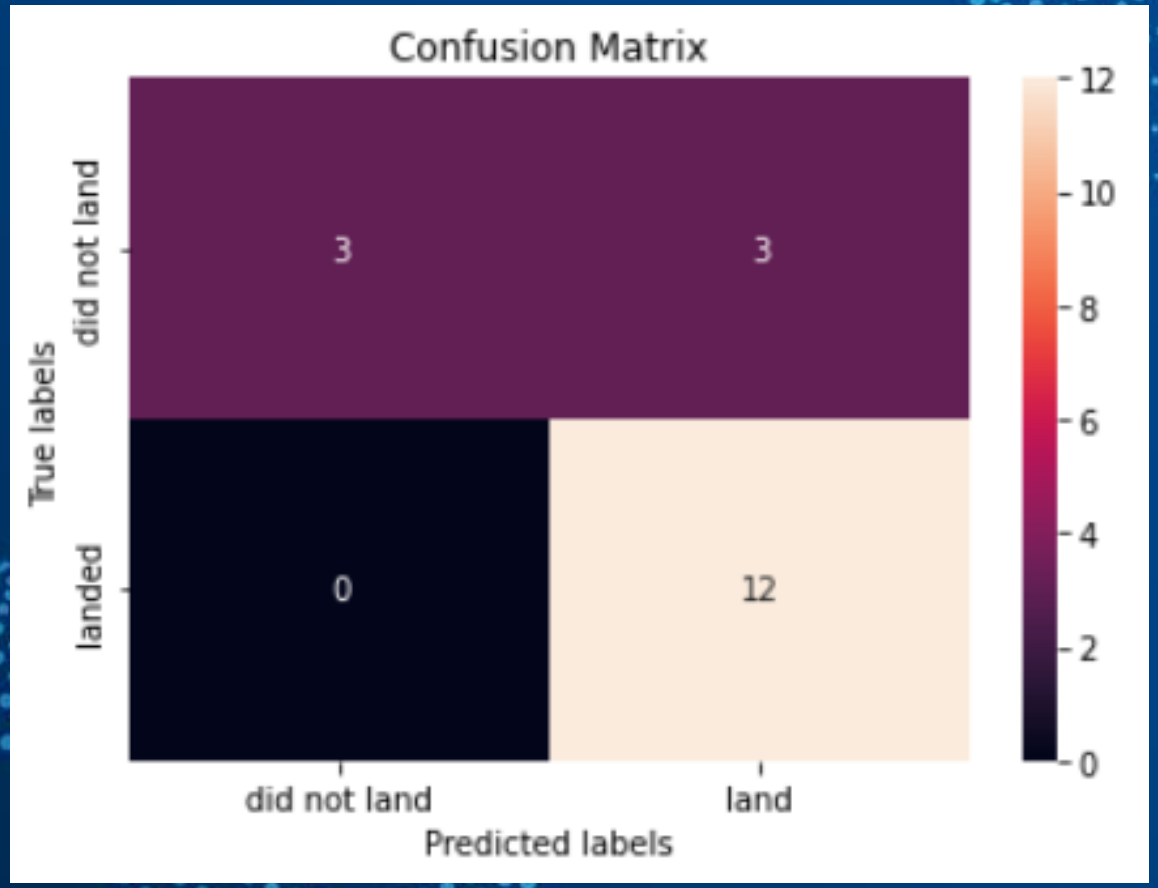
	Accuracy	Algorithm
0	0.875000	Tree
1	0.848214	KNN
2	0.846429	Log Reg

```
Best Algorithm is Tree with a score of 0.875
Best Params are: {'criterion': 'gini', 'max_depth': 6, 'max_features': 'auto', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random'}
```

Confusion Matrix for the Decision Tree Classifier

Examining the confusion matrix, we see that ***Decision Tree Classifier*** can distinguish between the different classes. We see that the major problem is false positives.





Conclusion

- The Decision Tree Classifier Algorithm is the best for Machine Learning for this dataset
- The success rate for SpaceX launches is proportional to the flight number (continuous launch attempts) and based on this we can predict that in the following years to come, they will eventually perfect the launches and almost guarantee the first stage can be reused
- There’s no clear indication whether Payload Mass affects the success rate of launches
- Launch site KSC LC-39A had the most successful launches as well as the highest launch success rate
- Orbit types ES-L1, GEO, HEO, and SSO all have the best success rate of 100%