Winning Space Race with Data Science

Eshan Wickramarachchi
08/06/2024

Outline

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

Executive Summary

Summary of methodologies

- Data Collection using SpaceX API
- Data Collection with Web Scraping
- Data Wrangling
- Exploratory Data Analysis using SQL
- EDA Data Visualization Using Python Pandas and Matplotlib
- Launch Sites Analysis with Folium-Interactive Visual Analytics and Ploty Dash
- Machine Learning Landing Prediction

- ## Summary of all results

- Exploratory Data Analysis initial impressions
- Interactive Visual Analytics and Dashboards
- Predictive Analysis(Classification)

Introduction

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.

I will predict if the Falcon 9 first stage will land successfully using data from previous Falcon 9 launches available from their website along with publicly available information

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data was collected using SpaceX REST API and web scrapping from Wikipedia
- Perform data wrangling
  - Unnecessary data was dropped and manipulated in way to make EDA with methods such as one-hot encoding for categorical features
- 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

How SpaceX data sets were collected.

- Data was first collected using SpaceX API (a RESTful API) by making a get request to the SpaceX API. This was done by first defining a series helper functions that would help in the use of the API with different end points 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 a Json result which was then converted into a Pandas data frame.
- Web scraping was used 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 a HTML. Using BeautifulSoup and request Libraries, I extracted the Falcon 9 launch HTML table records from the Wikipedia page, Parsed the table and converted it into a Pandas data frame

Data Collection – SpaceX API

Get request using the requests library to obtain rocket **launch data using API**

Use json_normalize function to convert json result to dataframe

Performed data cleaning and filling the missing value

```
spacex_url="https://api.spacexdata.com/v4/launches/past"
```

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

```
# Use json_normalize meethod to convert the json result into a dataframe
response.json()
data = pd.json_normalize(response.json())
```

```
# Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

# We will remove rows with multiple cores because those are falcon rockets with 2 extra rocket boosters and rows that have multiple payloads in a single rocket.
data = data[data['cores'].map(len)==1]
data = data[data['payloads'].map(len)==1]

# Since payloads and cores are lists of size 1 we will also extract the single value in the list and replace the feature.
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])

# We also want to convert the date_utc to a datetime datatype and then extracting the date leaving the time
data['date'] = pd.to_datetime(data['date_utc']).dt.date

# Using the date we will restrict the dates of the launches
data = data[data['date'] <= datetime.date(2020, 11, 13)]
```

The GitHub URL of the completed [SpaceX API calls notebook]

Data Collection - Scraping

Get request the Falcon9 Launch Wiki page from url

↓

Create a BeautifulSoup from the HTML response

↓

Extract all column/variable names from the HTML header

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

Create a `BeautifulSoup` object from the HTML `response`

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup=BeautifulSoup(response,'html.parser')
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
table = first_launch_table.find_all('th')
for header in table:
    name = extract_column_from_header(header)
    if name != None and len(name) > 0:
        column_names.append(name)
```

Completed web scraping notebook

Data Wrangling

Calculate the number of launches on each site

Calculate the number and occurrence of each orbit

Calculate the number and occurence of mission outcome per orbit type

Create a landing outcome label from Outcome column

```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

```
GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

```
landing_class = [0 if outcome in bad_outcomes else 1 for outcome in df['Outcome']]
landing_class
```

```
df['Class']=landing_class
df[['Class']].head(8)
```

	Class
0	0
1	0
2	0
3	0
4	0
5	0
6	1
7	1

Completed data wrangling notebook

EDA with Data Visualization

Performed exploratory Data Analysis and Feature Engineering using **`Pandas`** and **`Matplotlib`**

- Scatter plots to Visualize the relationship between "Flight Number" and "Payload Mass (kg)", "Flight Number" and "Launch Site", "Payload" and "Launch Site", "Flight Number" and "Orbit type", "Payload" and "Orbit type".

- Bar chart to Visualize the relationship between success rate of each orbit type

- Line plot to Visualize the launch success yearly trend.

Completed EDA with data visualization notebook

EDA with SQL

SQL queries performed

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'KSC'
- 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 where the successful landing outcome in drone ship was achieved.
- List the names of the boosters which have success in ground pad 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. Use a subquery
- List the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- 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

Completed EDA with SQL notebook

Build an Interactive Map with Folium

- Launch sites were marked and circled on the interactive world map
- We then assigned the dataframe launch_outcomes(failure,success) to classes 0 and 1 with Red and Green markers on the map
- Launch site proximity to key variables such as coast line, railways, highways and Nearest city were calculated and marked on the map
- These were added as it can give us an insight to why a launch site would have a greater or lesser success rate with landings if any at all.

Completed interactive map with Folium notebook

Build a Dashboard with Plotly Dash

The dashboard first contains a pie chart showing the proportion of total successful landings for all the landing sites, this can be intereacted with through the dropdown where you are able to see the proportion of successful and unsuccessful landings for each specific site.

This gives us a good idea is the landing success rate can be due to choosing the correct landing site.

To further capitalize on this , below is a graph showing Payload mass vs landing success, this is partnered with a slider to choose and adjust payload mass to better understand if there is a correlation between payload mass and landing success

Completed Plotly Dash lab

Predictive Analysis (Classification)

Building Model

- Create a Numpy array for Y values
- Transform and standardise initial dataframe to produce an X array
- Use the function train_test_split to split the data X and Y into training and test data
- For each ML algorithm create a GridSearchCv object to go through the various parameters and fit the model to the dataset

Evaluating Model

- Check the accuracy for each model on the test data
- Get tuned hyperparameters for each type of algorithms.
- Plot the confusion matrix.

Improving Model

- Use Feature engineering and chosing the best hyperparameters for each model

Finding Best Model

- Make a table with all the model accuracies of the refined models and the one with the highest accuracy s the best model

Completed predictive analysis lab notebook

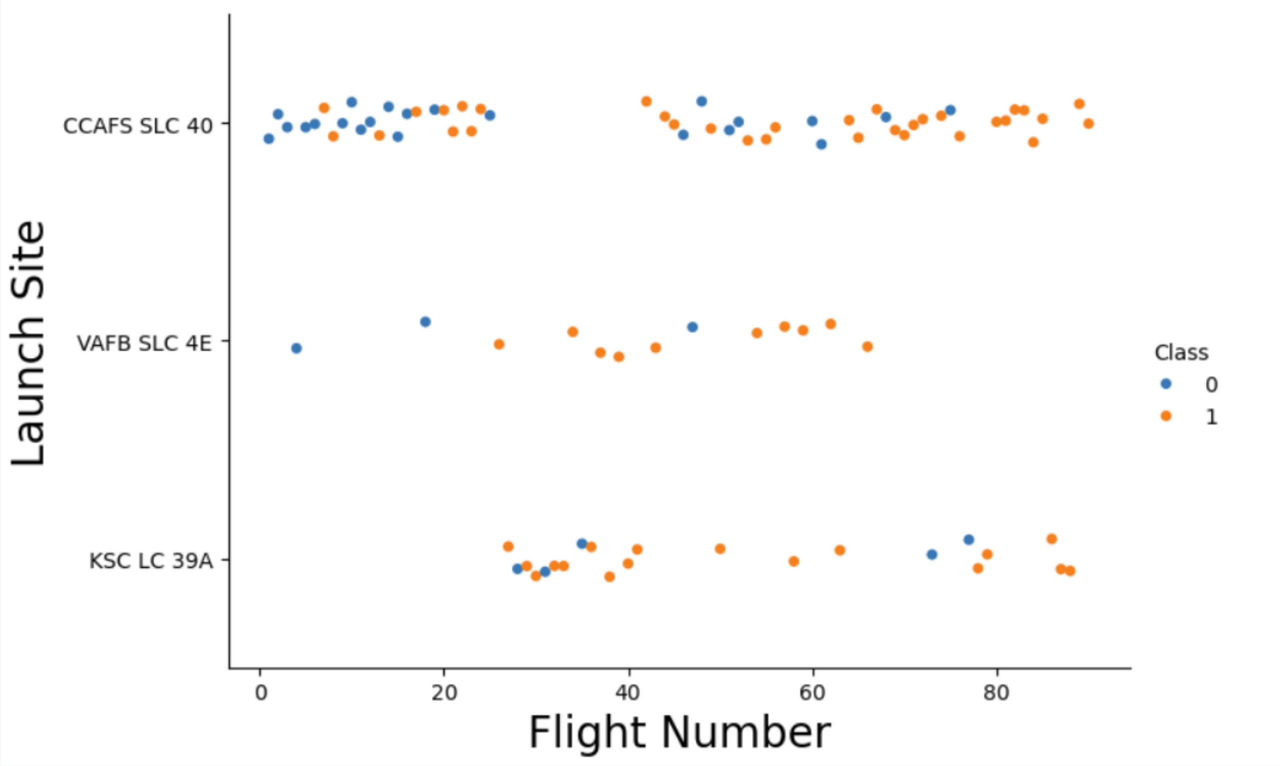
Results

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

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



This Plot shows the landing success' with respect to flight number and launch site.

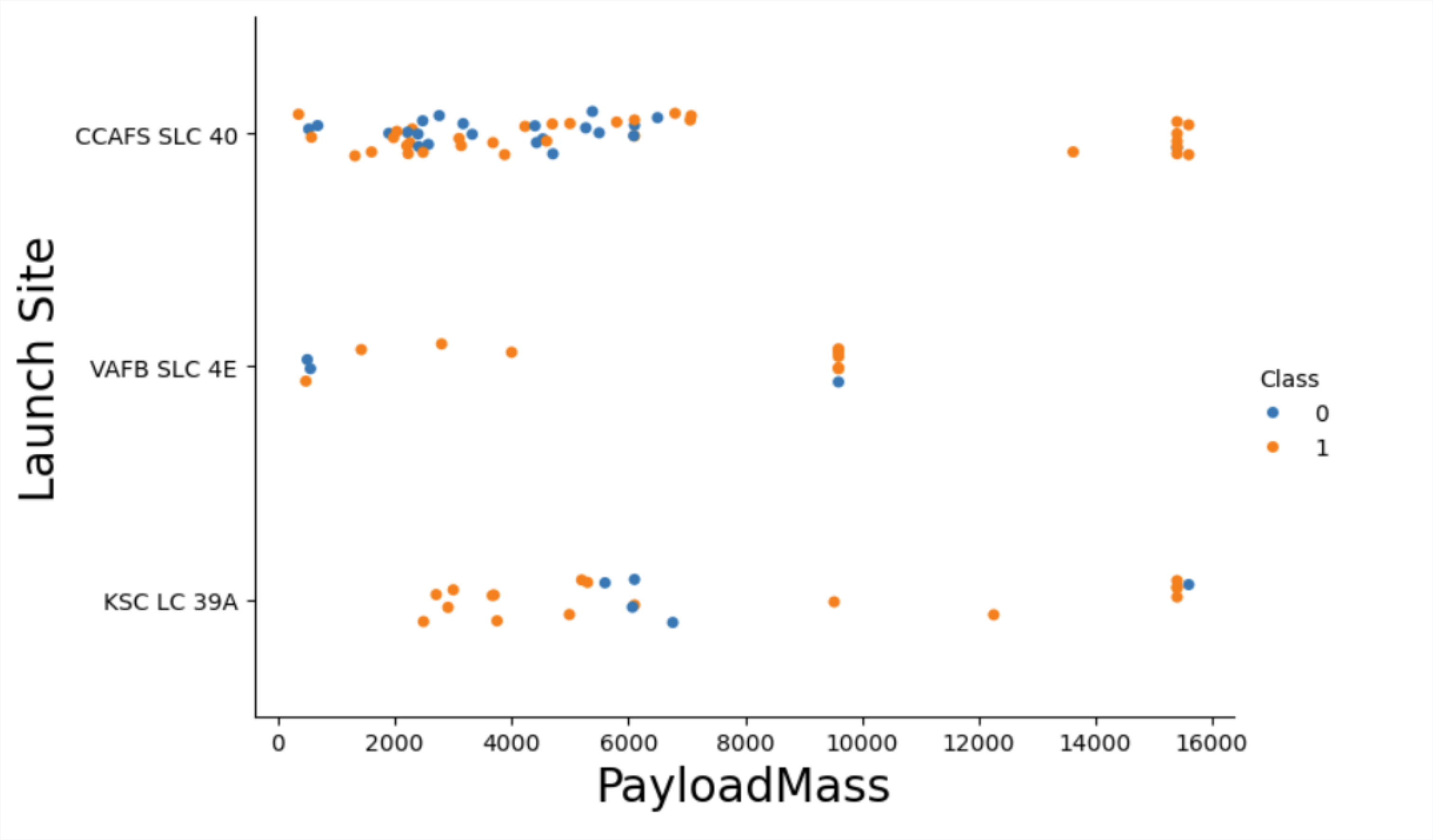
From looking at the graph it is possible to infer that the higher the flight number the higher the chances of a successful landing for all launch sites with a 100% success rate of all landings after the 80th flight

Payload vs. Launch Site

This shows a scatter plot of Payload vs. Launch Site

From looking at the graph is is possible to say there might be a correlation between having a higher payload and landing success rate as all launches above 10000 have landed successfully with the exception of one which could be an anomaly.

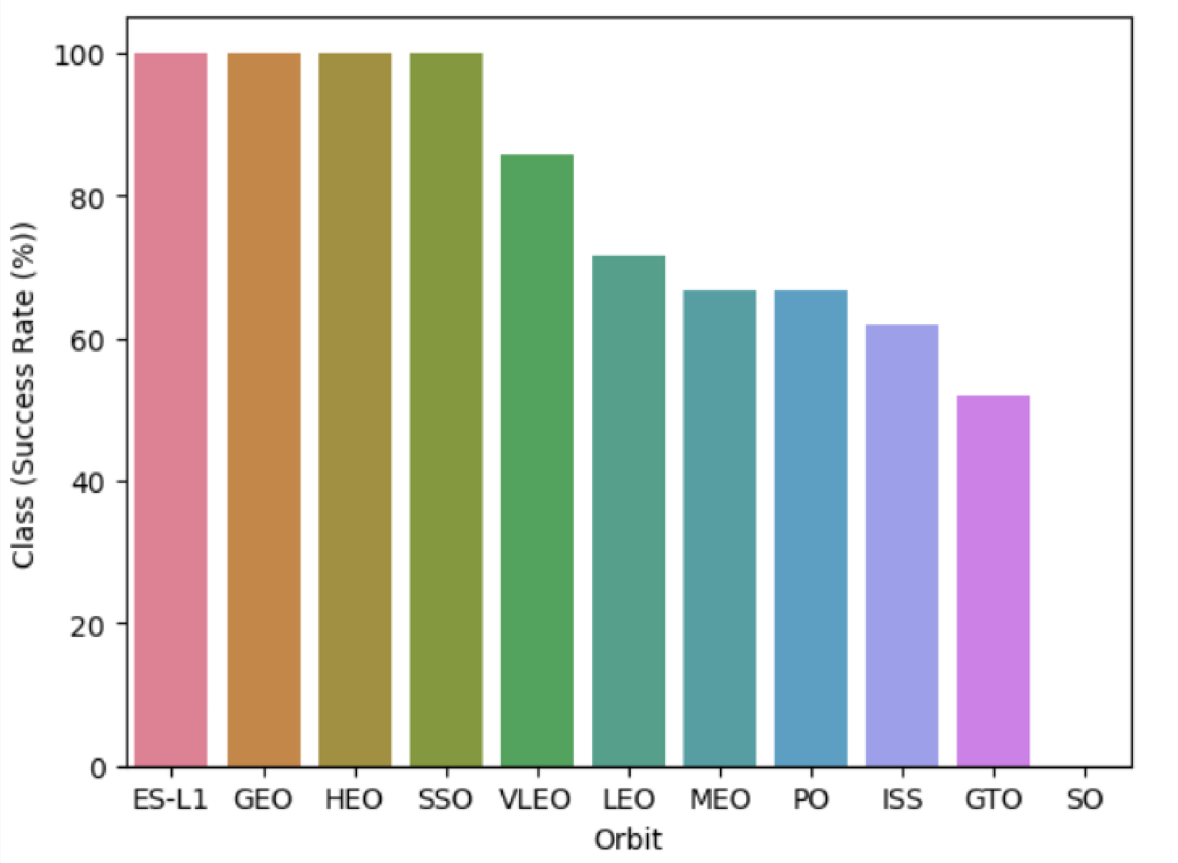
It is also seen that Launch site VAFB SLC 4E does not do any launches with a payload above 10000kgs



Success Rate vs. Orbit Type

- This shows a bar chart for the success rate of each orbit type

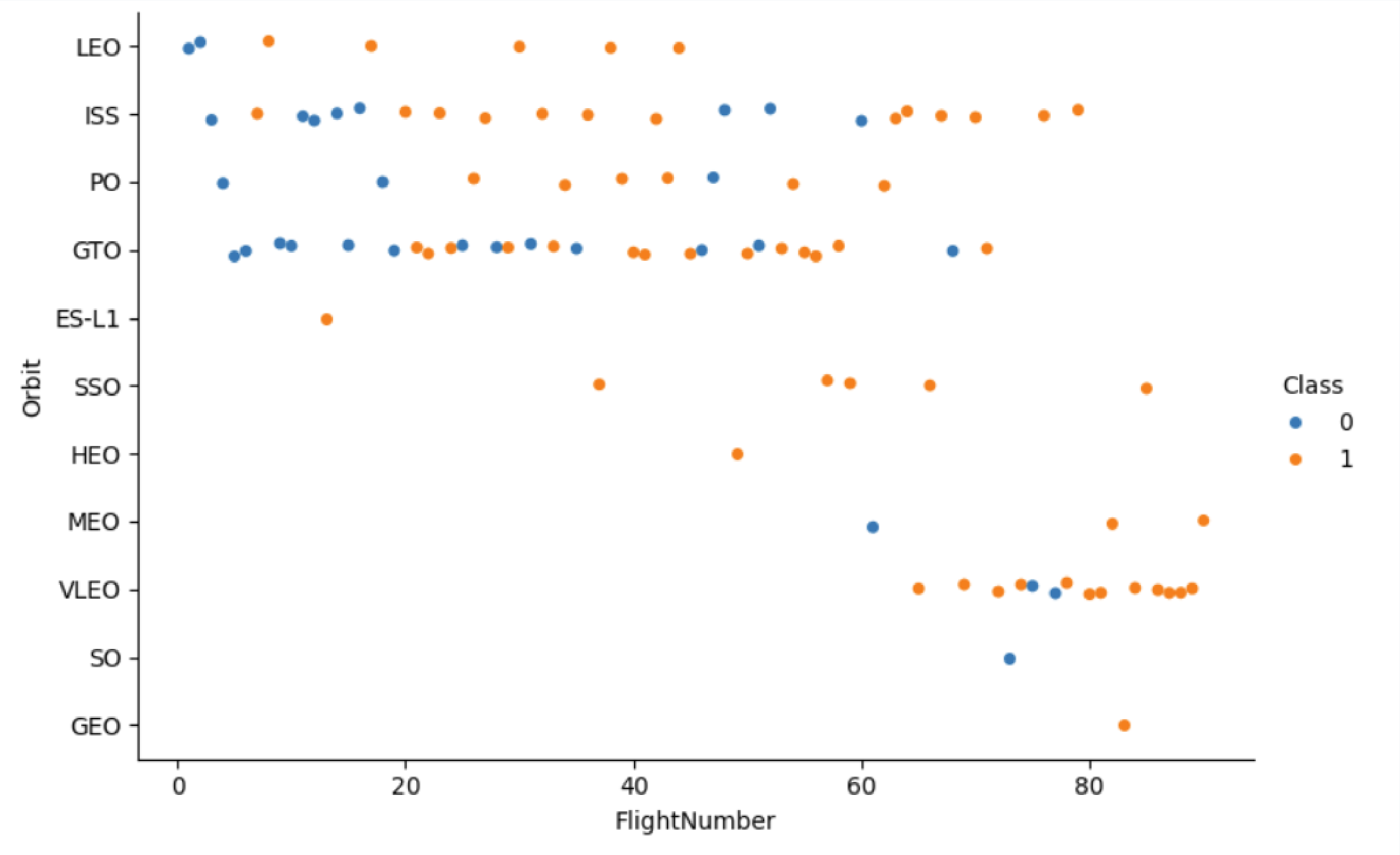
As show by the graph orbit ES-L1, GEO,HEO and SSO have a 100% success rate whilst SO having a 0% success rate so orbit type does have a strong effect on landing success for these .



Flight Number vs. Orbit Type

This shows a scatter point of Flight number vs. Orbit type

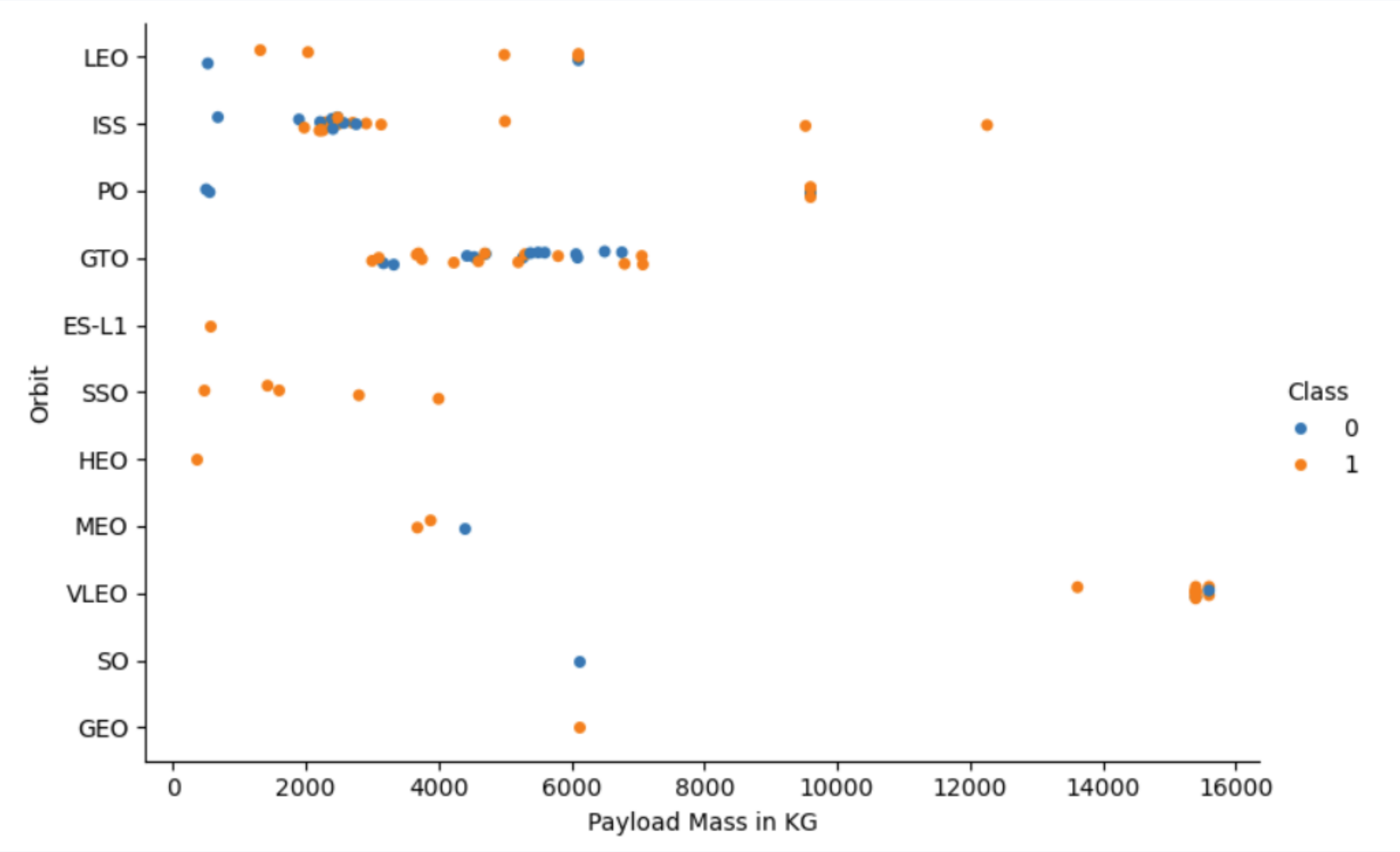
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

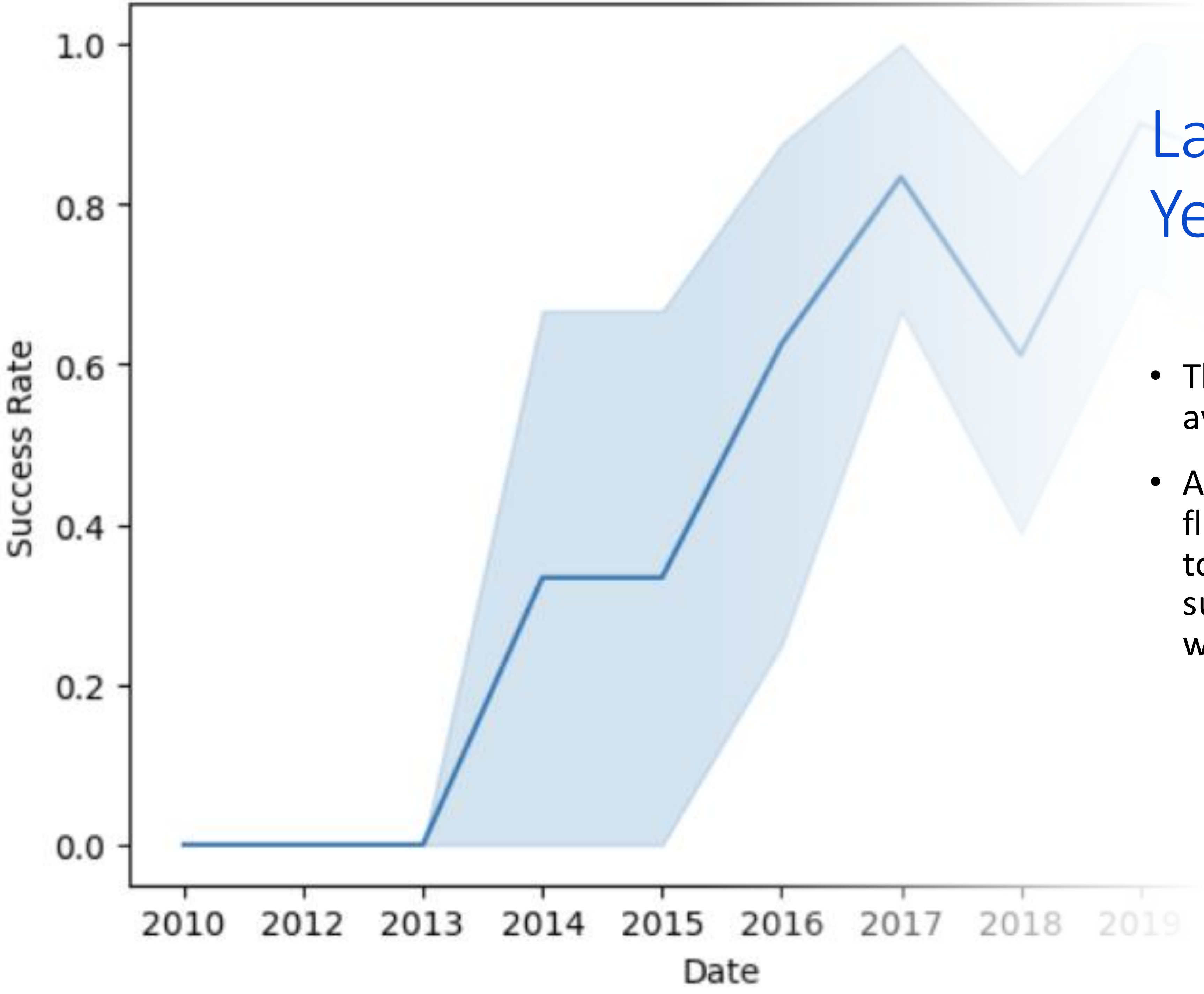


Payload vs. Orbit Type

This shows a scatter point of payload vs. orbit type

There is no new inference that can be made that hasn’t already previously been made but confirms previous ones such as higher payload higher success rate and specific orbit types wit 100% success rate but what's interesting is that all the 100% successful launches are oof a low payload mass which could infer that the orbit type has a greater influence on success than payload mass.





Launch Success Yearly Trend

- This shows a line chart of yearly average success rate
- As previously inferred with the flight number graphs it is easier to see and confirm that the success rate increases over time with more launches

All Launch Site Names

- Find the names of the unique launch sites
- The key word Distinct was used to get the unique values from the launch site column

```
%sql select Distinct Launch_Site from(SPACEXTABLE)
```

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

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

Launch Site Names Begin with 'KSC'

- Find 5 records where launch sites' names start with `KSC`
- 'like' was used along side with a '%' after KSC to show that we are looking for sites that start with the name KSC

Task 2

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

```
9]: %sql select * from 'SPACEXTABLE' where Launch_Site like "KSC%" limit 5
```

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

9]:

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)
2017-03-16	6:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- The function sum() is used to add all values in the column specified which is Payload mass and the where clause is used to make sure its just the sum of Nasa boosters

Task 3

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

```
%sql select sum(PAYLOAD_MASS__KG_) from 'SPACEXTABLE' where Customer = 'NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Used the avg() function to calculate the average of the payload mass and the 'where' function to specific the booster version

Task 4

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

```
%sql select avg(PAYLOAD_MASS__KG_) from 'SPACEXTABLE' where Booster_version like 'F9 v1.1%'
```

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

avg(PAYLOAD_MASS__KG_)
2534.6666666666665

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on drone ship.
- The min() function was used for the date column to get the first successful ground landing date which was specified by the where clause

Task 5

List the date where the succesful landing outcome in drone ship was acheived.

Hint:Use min function

```
%sql select min(Date) from(SPACEXTABLE) where Landing_Outcome = 'Success (drone ship)'
```

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

min(Date)
2016-04-08

Successful Drone Ship Landing with Payload between 4000 and 6000

- List the names of boosters which have successfully landed on ground pad and had payload mass greater than 4000 but less than 6000
- Once again 'distinct' was used to get unique names and a where clause was used to specify the range of the payload mass required along with and 'and' for specifying the landing site of a ground pad

Task 6

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

```
%sql select Distinct Booster_Version, Payload from(SPACEXTABLE) where PAYLOAD_MASS__KG_ >4000 and PAYLOAD_MASS__KG_ <6000 ar
```

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

Booster_Version	Payload
F9 FT B1032.1	NROL-76
F9 B4 B1040.1	Boeing X-37B OTV-5
F9 B4 B1043.1	Zuma

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- The count() function was used to get the total value of successful and failing missions alongside with a group by clause to group the mission outcomes

Task 7

List the total number of successful and failure mission outcomes

```
%sql select Mission_outcome, count(Mission_outcome) from(SPACEXTABLE) group by Mission_Outcome
```

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

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

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Distinct was used again to get unique booster names for the maximum payload which was found using a sub query with the max() function to select the maximum payload

Task 8

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

```
select Distinct Booster_Version, PAYLOAD_MASS__KG_ from(SPACEXTABLE) where PAYLOAD_MASS__KG_      = (select max(PAYLOAD_MAS
```

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

Booster_Version	PAYLOAD_MASS__KG_
F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

2017 Launch Records

- List the records which will display the month names, successful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017
- A where clause was used to specify the date of 2017 and the landing site of ground pads to get the specific data required

Task 9

List the records which will display the month names, succesful landing_outcomes in ground pad ,booster versions, launch_site for the months in year 2017

Note: SQLLite does not support monthnames. So you need to use substr(Date,6,2) for month, substr(Date,9,2) for date, substr(Date,0,5),='2017' for year.

```
%sql select substr(Date,6,2),substr(Date,0,5),Booster_Version,Launch_Site Landing_Outcome from(SPACEXTABLE)where substr(Date
```

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

substr(Date,6,2)	substr(Date,0,5)	Booster_Version	Landing_Outcome
02	2017	F9 FT B1031.1	KSC LC-39A
05	2017	F9 FT B1032.1	KSC LC-39A
06	2017	F9 FT B1035.1	KSC LC-39A
08	2017	F9 B4 B1039.1	KSC LC-39A
09	2017	F9 B4 B1040.1	KSC LC-39A
12	2017	F9 FT B1035.2	CCAFS SLC-40

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
- Count was used to count the values of the landing outcomes that where specified by the where clause by date and grouped by the group by clause

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

```
%sql select Date,Landing_Outcome, count(Landing_Outcome) as count from(SPACEXTABLE) where Date between '2010-06-04' and  '2(
```

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

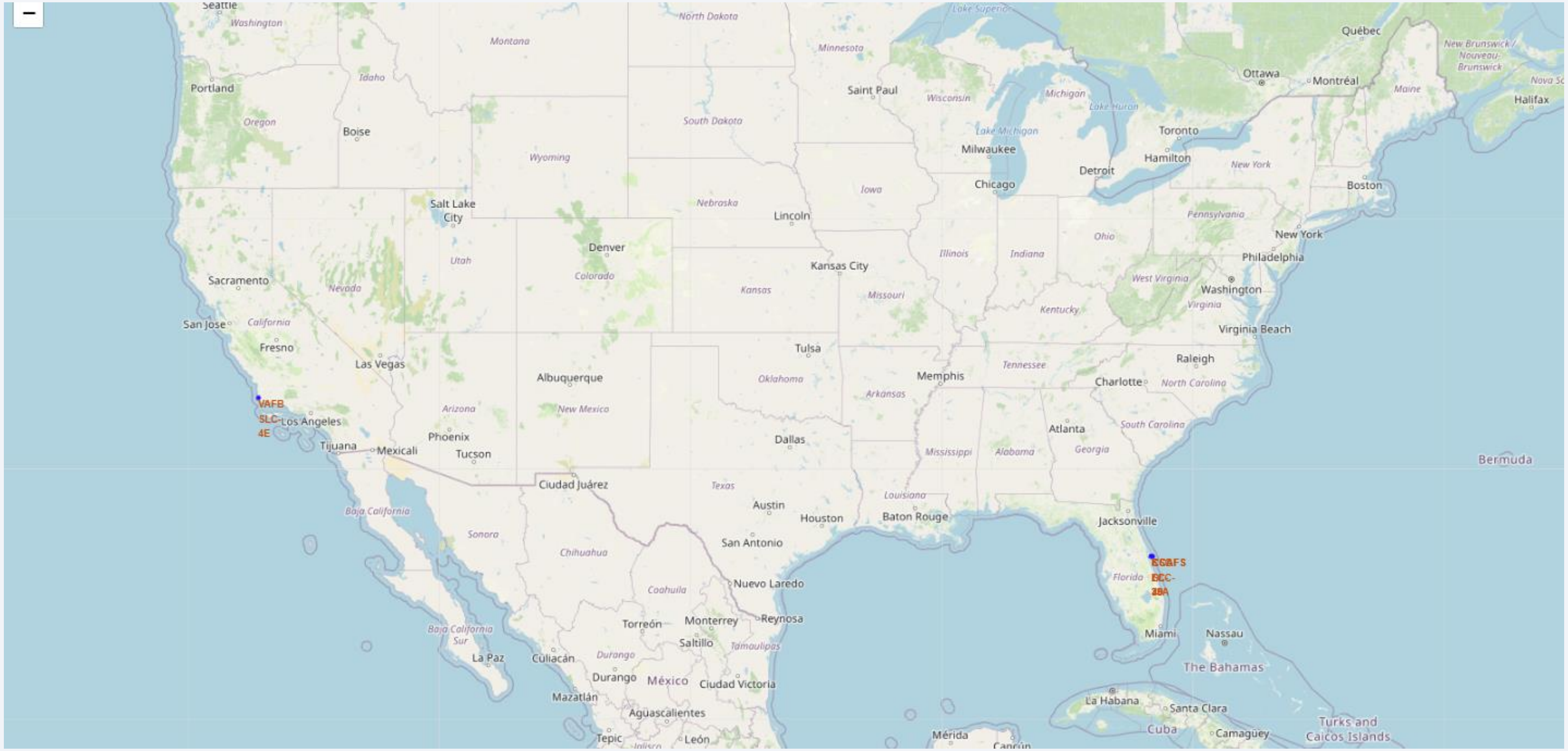
Date	Landing_Outcome	count
2012-05-22	No attempt	10
2016-04-08	Success (drone ship)	5
2015-01-10	Failure (drone ship)	5
2015-12-22	Success (ground pad)	3
2014-04-18	Controlled (ocean)	3
2013-09-29	Uncontrolled (ocean)	2
2010-06-04	Failure (parachute)	2
2015-06-28	Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

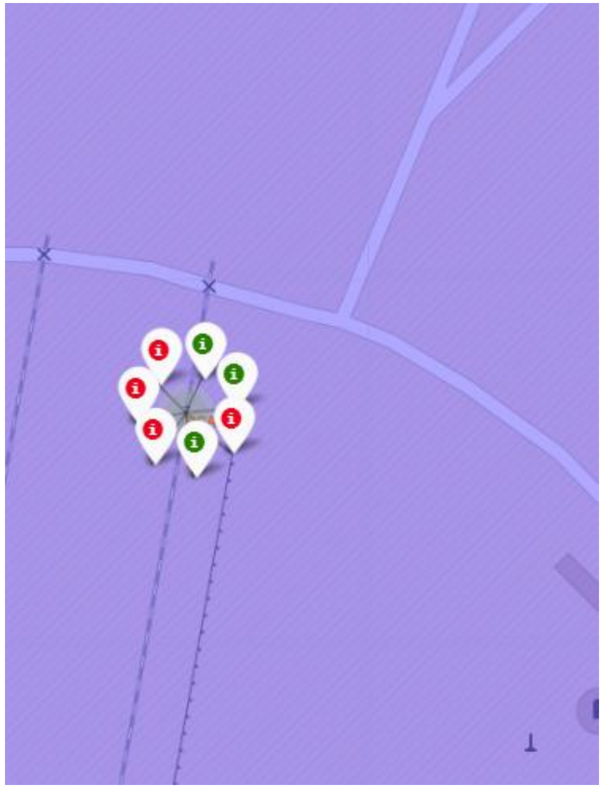
Launch Site Locations

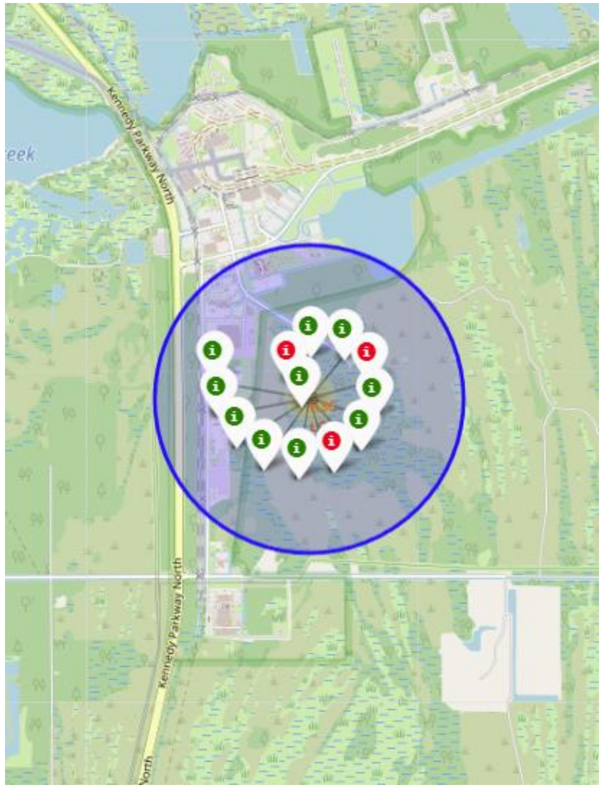
This shows all the SpaceX launch sites and as shown you can clearly tell that all the launch sites are on the coastline

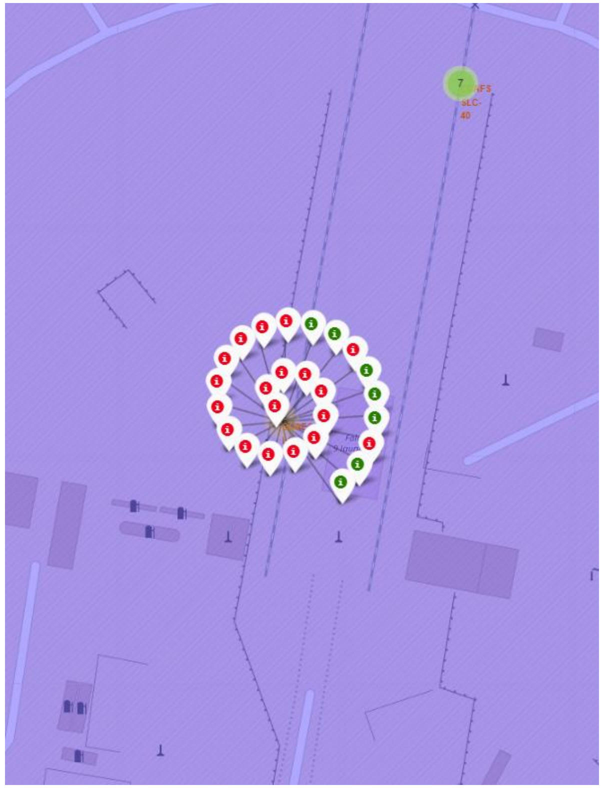


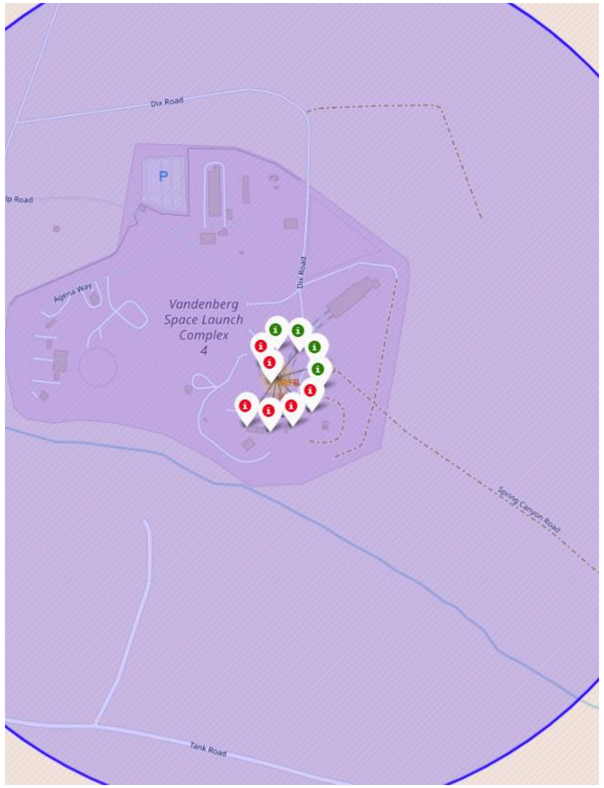
Launch site clusters with mission outcomes

Green = Successful
Red = Unsuccessful

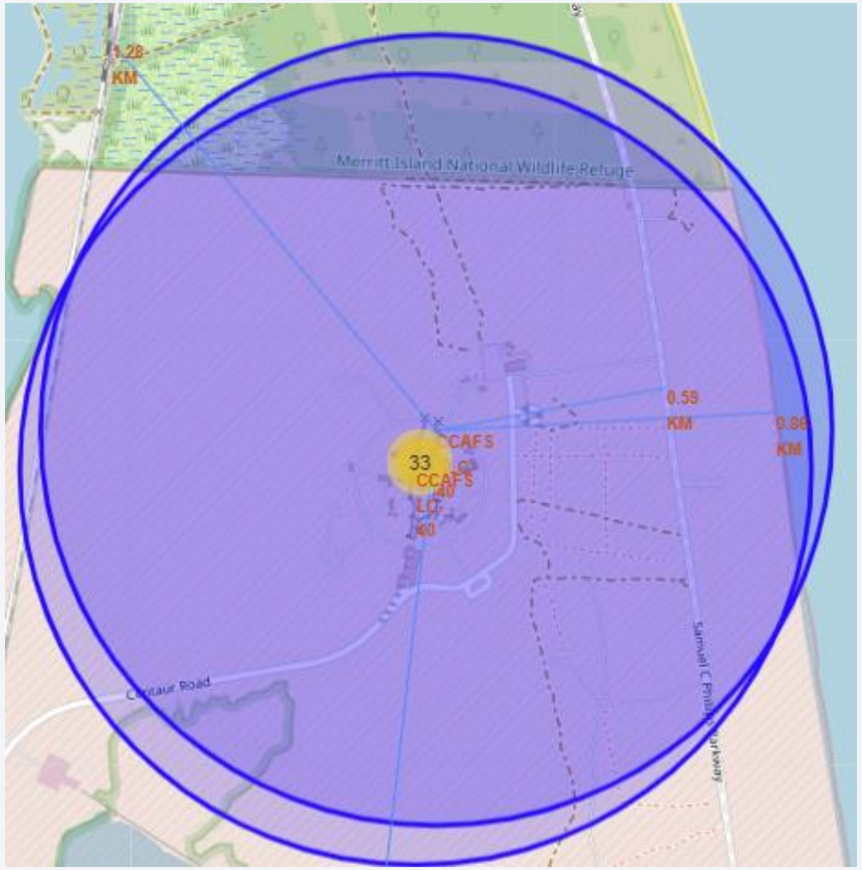






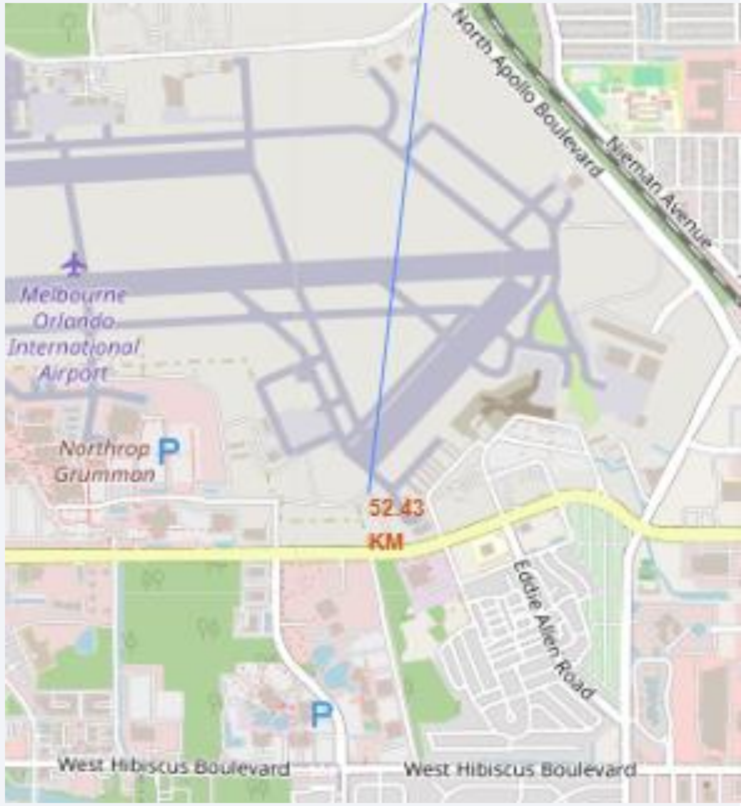


Launch Site Proximities



As you can see from the figure on the right, the nearest city is Melbourne with it being 52.43kms away.

The figure on the left shows the nearest railway is 1.28kms away, the nearest highway is 0.59kms and the coastline is 0.86kms away



Section 4

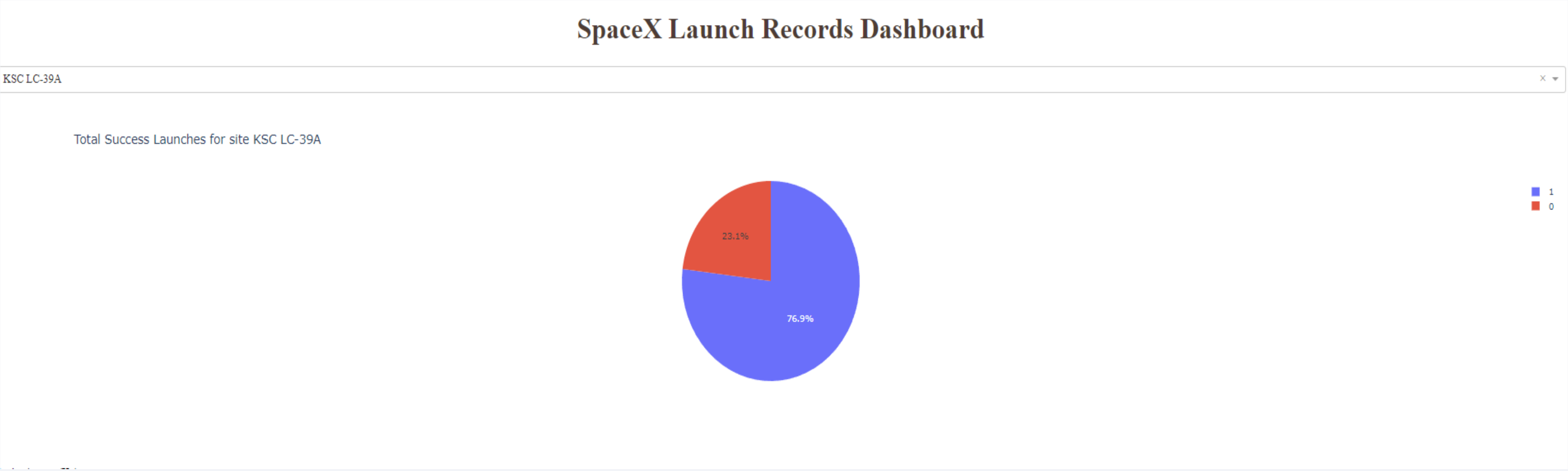
Build a Dashboard with Plotly Dash

Success percentages for each Launch site

From the pie chart we can see that the launch site with the most successful launches was KSC LC-39A

KSC LC-39A success ratio

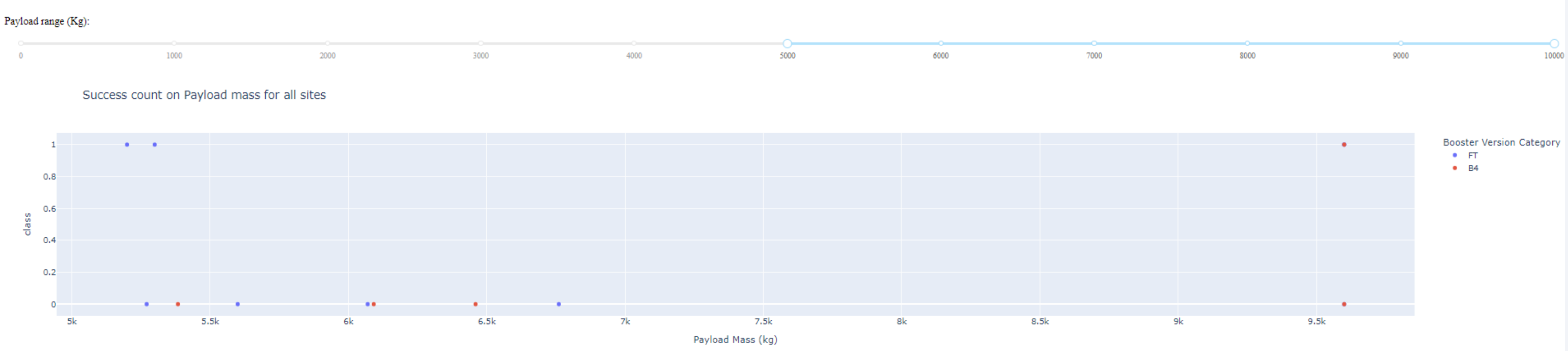
Here we explore the success rate of the most successful launch site and we can see that there is a 76.9% success rate for the site



Payload Mass and Booster Version vs Landing Outcome

Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc. As seen by the graphs we can conclude that a payload of less than 5000kg has no influence on the landing success byt a payload between 5000 – 10000kg has a much lower success rate.

As for boosters its seen that FT has the highest success rate which is especially seen below 5000kg whilst v1.1 has by far the worst success rates for any payload value



Section 5

Predictive Analysis (Classification)

Classification Accuracy

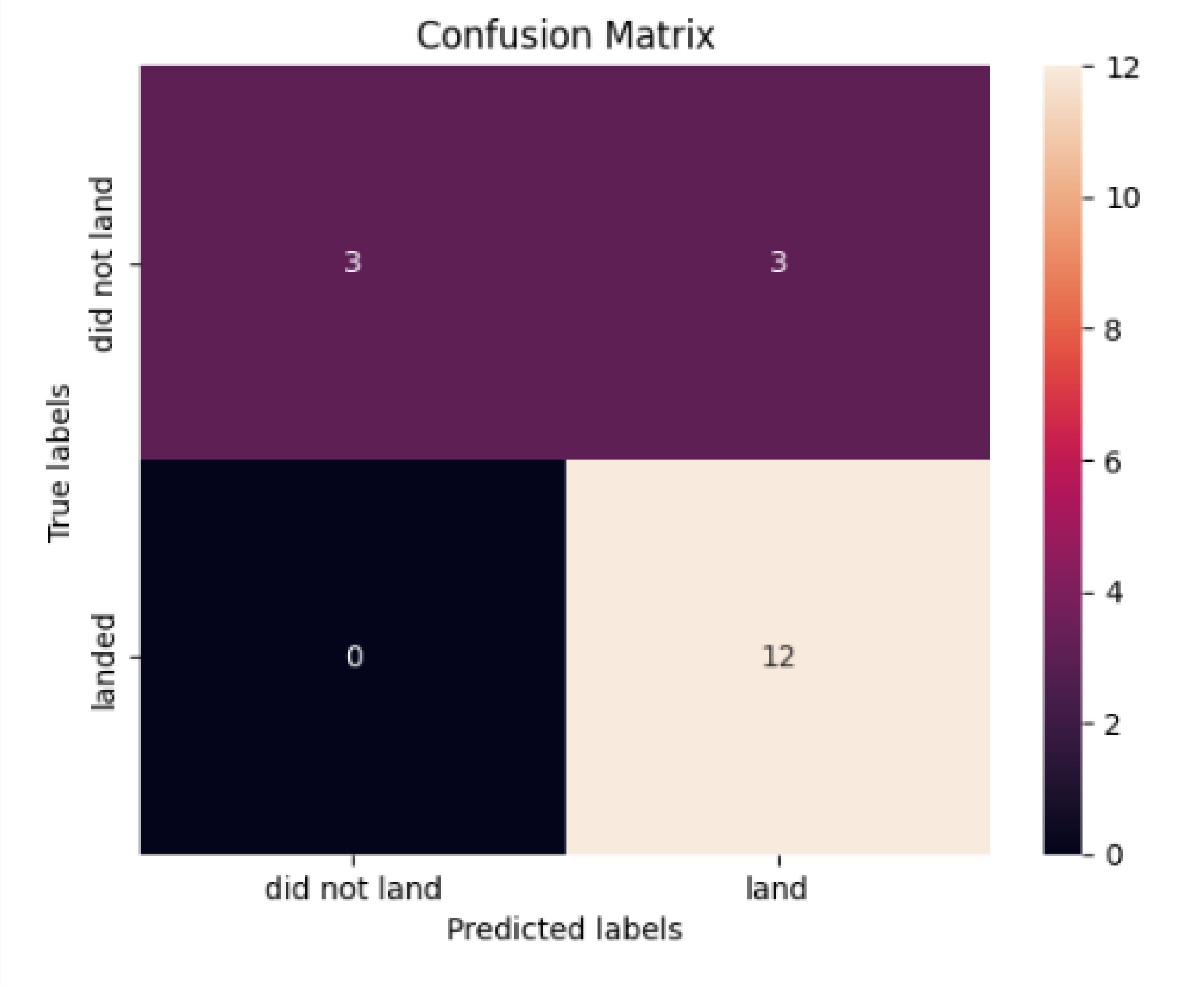
Machine Learning Method	Test Data Accuracy
Logistic Regression	0.833333
SVM	0.833333
Decision Tree	0.833333
KNN	0.833333

- As you can see from the table, all models had the same accuracy on the test data so model choice is up to the user.

Confusion Matrix

As all models had the same accuracy their confusion matrices are all also the same as showed in the figure on the right.

The main area for concern would be the 3 false positives



Conclusions

- Different Launch sites have different chances of success but all launch sites are more liekly to succeed than fail with the lowest launch site success rate being above 50%
- When looking overall at the data , one of the strongest trend is the increase in succeess rate with flight number meaning the success rate has increased with time with each launch as show by the Yearly Launch Success graph
- Aside from that to get an even more accurate prediciton we can see that Orbits ES-L1, GEO, HEO & SSO have a 100% success rate and SO orbit having a 0% success rate so orbits are a very neccecery variable in predicitng the landing success.
- It is also important to consider payload alongside orbit types as some orbits seem directly affected by is such ass ISS orbit seems to have a 100% success rate with payload over 3500KG
- It is not only payload that can affect the orbit types though for example its seeen that LEO has a higher success rate with more flight numbers as the last 5launches have landed successfully but this effect of flight number and time does not effect alll orbits such as gto which has seen no correlation between the two
- All these point mean that to predict the landing outcome successfully many factors such as the missions payload, orbit type and landing site will have to be taken under consideration to make an accurate predicition but if all these details are given I think we can predict the mission out come to a fair degree of certainty.

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!