IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Nicholas Bates
08/11/2025

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion

Executive Summary

The primary goal of this project is to predict whether Space Y can compete with Allon Mask's Space X.

This is achieved through use of machine learning modelling to predict if the first stage of Space X will be successful and able to be reused. Also, through calculating the cost of each Space X launch.

Methodologies

- Data collection: Request API & web scrapping
- Exploratory data analysis: Python
- Visualizations & Dashboards : Folium and Seaborn

Results

- Through EDA & geographical visualisation the factors of payload mass and launch site seem to affect success rate
- The most appropriate model to predict success of the first stage is a Decision tree model with hyper tuned parameters.

Introduction

There is a current boom in the commercial space age. Numerous companies are offering affordable space travel. Here at Space Y, the goal is to see if it is feasible to compete in this market.

The current market leader is Allon Mask's Space X. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars. These savings stem mainly from reuse of the first stage.

By determining if the first phase lands, the cost of a launch by Space X can be determined. Therefore, this information can be used to see if Space Y is able to compete with Space X.

To compete with Space X the following need to be determined:

- the cost of each phase of Space X's Falcon 9 launch,
- whether the first phase will be reused,
- compare with what Space Y can offer.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Through use of the Space X API – details of each Falcon 9 launch
  - Web Scraping – Wikipedia table of Falcon 9 data
- Perform data wrangling
  - Python was used to calculate value counts for various attributes & to create an outcome attribute
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- SpaceX launch data was gathered from the SpaceX REST API and python Request library. This API provides data about launches such as rocket used, payload delivered, landing outcome, etc. The response is in JSON, which was converted to a Pandas dataframe.
- Data is also found through web scraping related Wiki pages. The Python Beautiful Soup package was used to scrape these Wiki HTML tables containing Falcon 9 launch records. These were parsed and converted into a Pandas dataframe.
- Data was then cleaned, specifically filtering Falcon 9 data from Falcon 1 booster data.
- NULL data in Payload Mass was then replaced with mean values for this attribute.

Data Collection – SpaceX API

API request

- Use python requests library
- website: https://api.spacexdata.com/v4/launches/past
- response = requests.get(spacex_url)

Parse Data

- Create dataframe from JSON data
- Use IDs from rocket, payloads, launchpad, and cores attributes & helper functions to make requests for data about launches
- Combine data into a dataframe

Data Cleaning

- Filter data to only include Falcon 9 data
- Replace NULL values in Payload Mass with mean values

https://github.com/nicbates/github-final-project/blob/main/1%20jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

Request

- Use python requests library
- Make get request to Falcon 9 Wiki
- Create a Beautiful Soup object from response

Extraction

- Extract all column/variable names from the HTML table header
- Parse the launch HTML tables

Dataframe

- Create Dataframe from extracted data
- Export to CSV

https://github.com/nicbates/github-final-project/blob/main/2%20jupyter-labs-webscraping.ipynb

Data Wrangling

Missing Values
- Calculate missing values

Launches
- Calculate the number of launches on each site

Orbit
- Calculate the number and occurrence of each orbit

Outcome
- Calculate the number and occurence of mission outcome of the orbits

Outcome Class

Create a landing outcome label from Outcome column

https://github.com/nicbates/github-final-project/blob/main/3%20labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

- For each outcome class a scatter graph was plotted for:
  - Flight Outcome vs Payload Mass
  - Light Number vs Launch Site
  - Payload Mass vs Launch Site
  - Flight Number & Orbit Type
  - Payload Mass vs Orbit Type
- Line graph for yearly launch success
- A bar chart for success rate of each orbit type

https://github.com/nicbates/github-final-project/blob/main/5%20eda-data-viz.ipynb

EDA with SQL

- 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 all the booster_versions that have carried the maximum payload mass, using a subquery with a suitable aggregate function.
- 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 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.

https://github.com/nicbates/github-final-project/blob/main/4%20jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Folium was used to create an interactive map
- All launch sites were plotted on the interactive map
- Each launch was tagged with a success/failed launch
- The distance between a specific launch site to the following proximities was calculated and a polyline displayed:
  - coastline
  - railway
  - city

https://github.com/nicbates/github-final-project/blob/main/6%20lab_jupyter_launch_site_location.ipynb

Predictive Analysis (Classification)

Data Preparation
- Standardize data using Standard Scaler
- Create a 80/20 train test split

Choose Models
- 4 models were chosen: Logistic Regression, K- Nearest Neighbours, Decision Tree Classifier and Support Vector Machine

Pipeline
- Create dictionary of parameters for GridSearchCV object
- Get best parameters using training data and GridSearchCV object

Compare Models
- Use test data and best parameters to calculate accuracy
- Create a confusion matrix for each model

Best Model
- Choose model with best test results

https://github.com/nicbates/github-final-project/blob/main/7%20SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

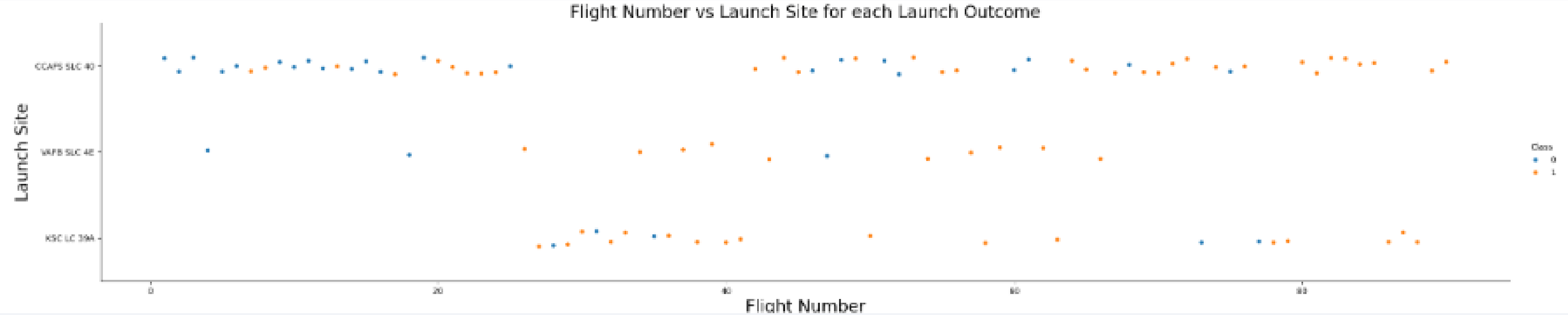
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

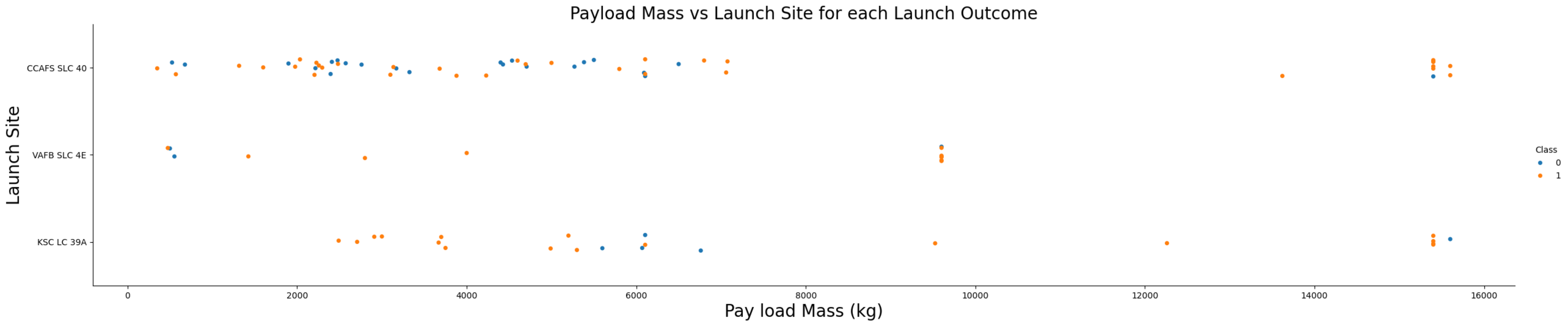
Flight Number vs. Launch Site

As the number of flights increases from a site the chances of success increases. It is evident that there are a lot more launches from CCAFS SLC 40 and so they have more experience. From the graph it is hard to see if there is a larger proportion of successes from either of the sites



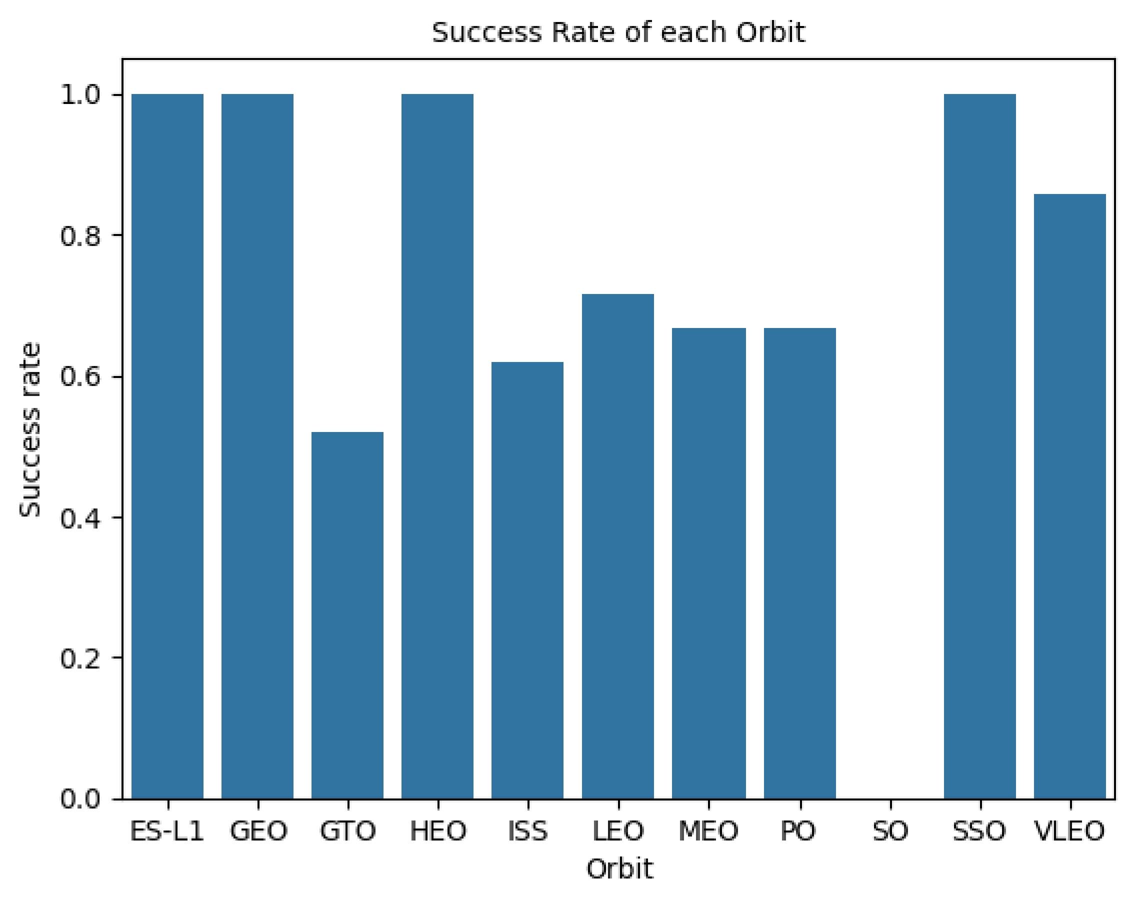
Payload vs. Launch Site

KSC LC 39A & CCAFS SLC 40 both have more experience with heavier payloads than VAFB SLC 4E. It seems that for the former sites as payload increases above 8000 Kg chances of success increase. However, for KSC LC 39A payloads between 2000 – 5000 Kg also increase success rate.



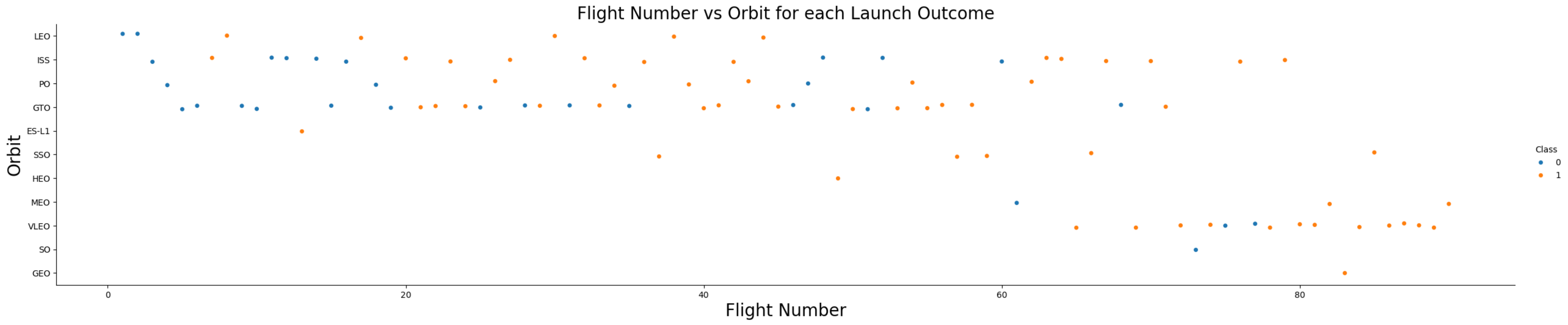
Success Rate vs. Orbit Type

- The highest success rates approaching 100% are with orbits ES-L1, GEO, HEO & SSO
- However, a GTO orbit has almost a 50-50 chance of success.



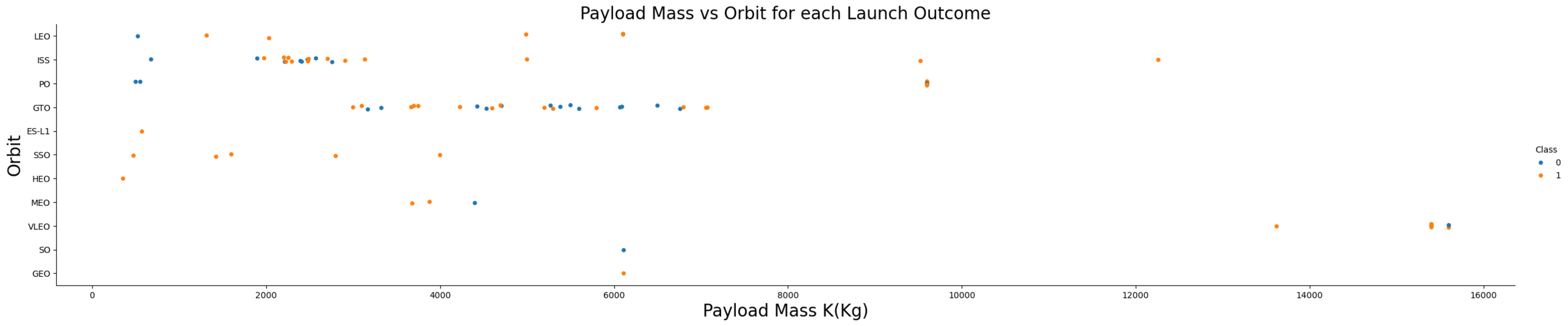
Flight Number vs. Orbit Type

It appears that as flight number increases the success of a launch in any orbit also increases possibly due to an increase in experience of launches from each site. However, it also shows that orbits like ES-L1where there were 100% success were based on 1 launch where as VLEO, which had a ~90% success rate, had 14 launches. Therefore, orbits based on success rate only is possibly not a good metric to use.



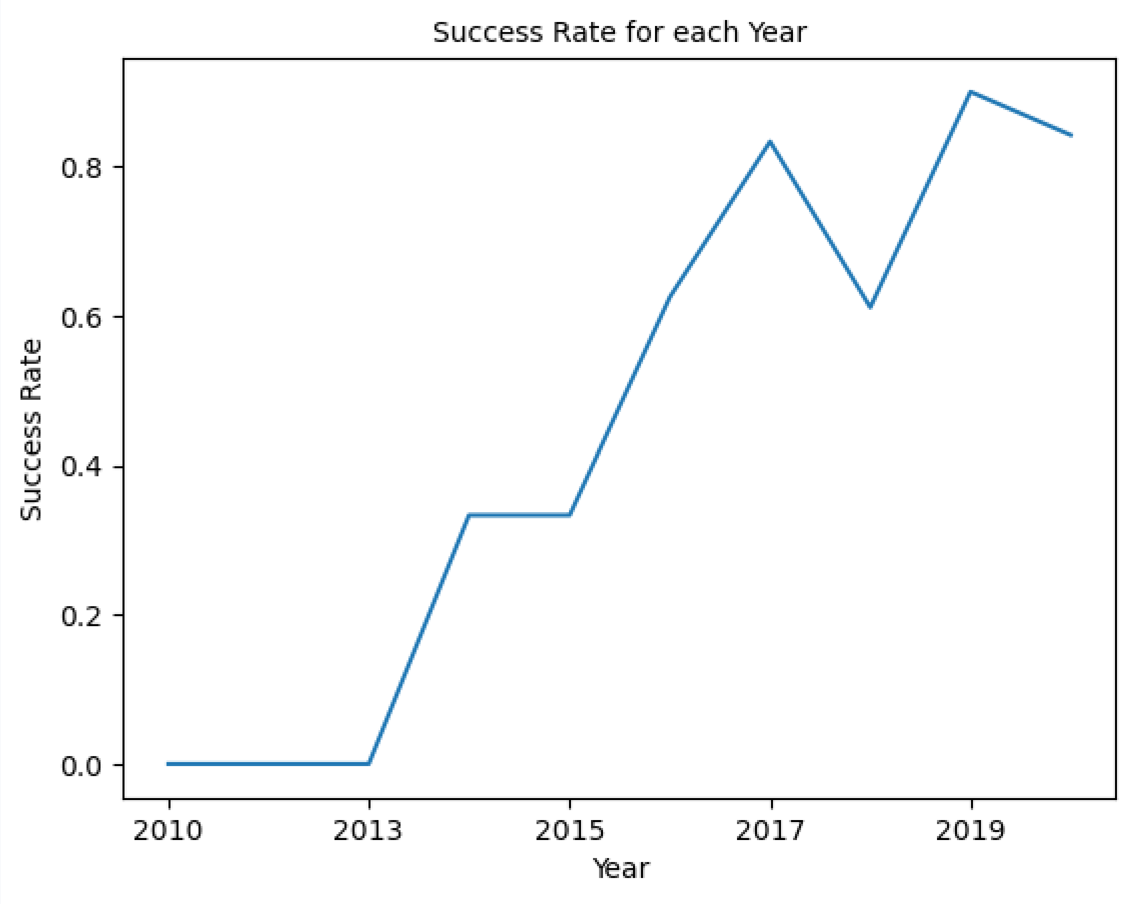
Payload vs. Orbit Type

It is evident that there are only three orbits that have launches with over 8000 kg. Also, most launches have a payload between 2000 – 6000 kg, with the orbits LEO, SEO & ISS showing complete success in this range.



Launch Success Yearly Trend

- The initial 3 years of launches all proved to be failures until 2013
- Between 2013- 2017 there is generally a linear trend in increasing success rate.
- After this there is slight fluctuation with peaks and troughs evident – perhaps indicating experimentation and optimization of launch parameters



All Launch Site Names

```
In [60]: %%sql
         SELECT DISTINCT "Launch_Site"
         FROM SPACEXTABLE;

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

Out[60]:

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

Launch Site Names Begin with 'CCA'

In [61]:

```
%%sql
SELECT *
FROM SPACEXTABLE
WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5;
```

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

Out[61]:

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	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	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

In [62]:
```
%%sql
SELECT
    Customer,
    SUM("PAYLOAD_MASS__KG_" ) as "Total Payload Mass"
FROM SPACEXTABLE
WHERE Customer == "NASA (CRS)";
```

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

Out[62]:

Customer	Total Payload Mass
NASA (CRS)	45596

Average Payload Mass by F9 v1.1

In [63]:
```
%%sql
SELECT
    "Booster_Version" ,
    AVG("PAYLOAD_MASS__KG_" ) as "Average Payload Mass"
FROM SPACEXTABLE
WHERE "Booster_Version" == "F9 v1.1";
```

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

Out[63]:

Booster_Version	Average Payload Mass
F9 v1.1	2928.4

First Successful Ground Landing Date

In [64]:
```
%%sql
SELECT
    Customer,
    "Landing_Outcome",
    MIN("Date") as "First Landing Success"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (ground pad)';
```

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

Out[64]:

Customer	Landing_Outcome	First Landing Success
Orbcomm	Success (ground pad)	2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

In [65]:

```
%%sql
SELECT DISTINCT
    "Booster_Version",
    "PAYLOAD_MASS__KG_" as "Payload Mass"
FROM SPACEXTABLE
WHERE "Landing_Outcome" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000;
```

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

Out[65]:

Booster_Version	Payload Mass
F9 FT B1022	4696
F9 FT B1026	4600
F9 FT B1021.2	5300
F9 FT B1031.2	5200

Total Number of Successful and Failure Mission Outcomes

In [66]:

```
%%sql
SELECT
    "Mission_Outcome",
    COUNT(*) AS "Mission Outcome Totals"
FROM SPACEXTABLE
GROUP BY "Mission_Outcome";
```

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

Out[66]:

Mission_Outcome	Mission Outcome Totals
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

In [67]:

```
%%sql
SELECT
    "Booster_Version",
    "PAYLOAD_MASS__KG_" as "Max Payload"
FROM SPACEXTABLE
WHERE "PAYLOAD_MASS__KG_" =
    (SELECT MAX("PAYLOAD_MASS__KG_")
     FROM SPACEXTABLE);
```

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

Out[67]:

Booster_Version	Max Payload
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

In [68]:

```
%%sql
SELECT
    CASE
        WHEN substr("Date", 6, 2) = '01' THEN 'January'
        WHEN substr("Date", 6, 2) = '02' THEN 'February'
        WHEN substr("Date", 6, 2) = '03' THEN 'March'
        WHEN substr("Date", 6, 2) = '04' THEN 'April'
        WHEN substr("Date", 6, 2) = '05' THEN 'May'
        WHEN substr("Date", 6, 2) = '06' THEN 'June'
        WHEN substr("Date", 6, 2) = '07' THEN 'July'
        WHEN substr("Date", 6, 2) = '08' THEN 'August'
        WHEN substr("Date", 6, 2) = '09' THEN 'September'
        WHEN substr("Date", 6, 2) = '10' THEN 'October'
        WHEN substr("Date", 6, 2) = '11' THEN 'November'
        WHEN substr("Date", 6, 2) = '12' THEN 'December'
        ELSE 'Unknown'
    END AS "Month",
    "Date"
    "Landing_Outcome",
    "Booster_Version",
    "Launch_Site"
FROM SPACEXTABLE
WHERE substr("Date", 0, 5) = '2015' AND Landing_Outcome = 'Failure (drone ship)';
```

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

Out[68]:

Month	Landing_Outcome	Booster_Version	Launch_Site
January	2015-01-10	F9 v1.1 B1012	CCAFS LC-40
April	2015-04-14	F9 v1.1 B1015	CCAFS LC-40

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

In [69]:

```
%%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.
```

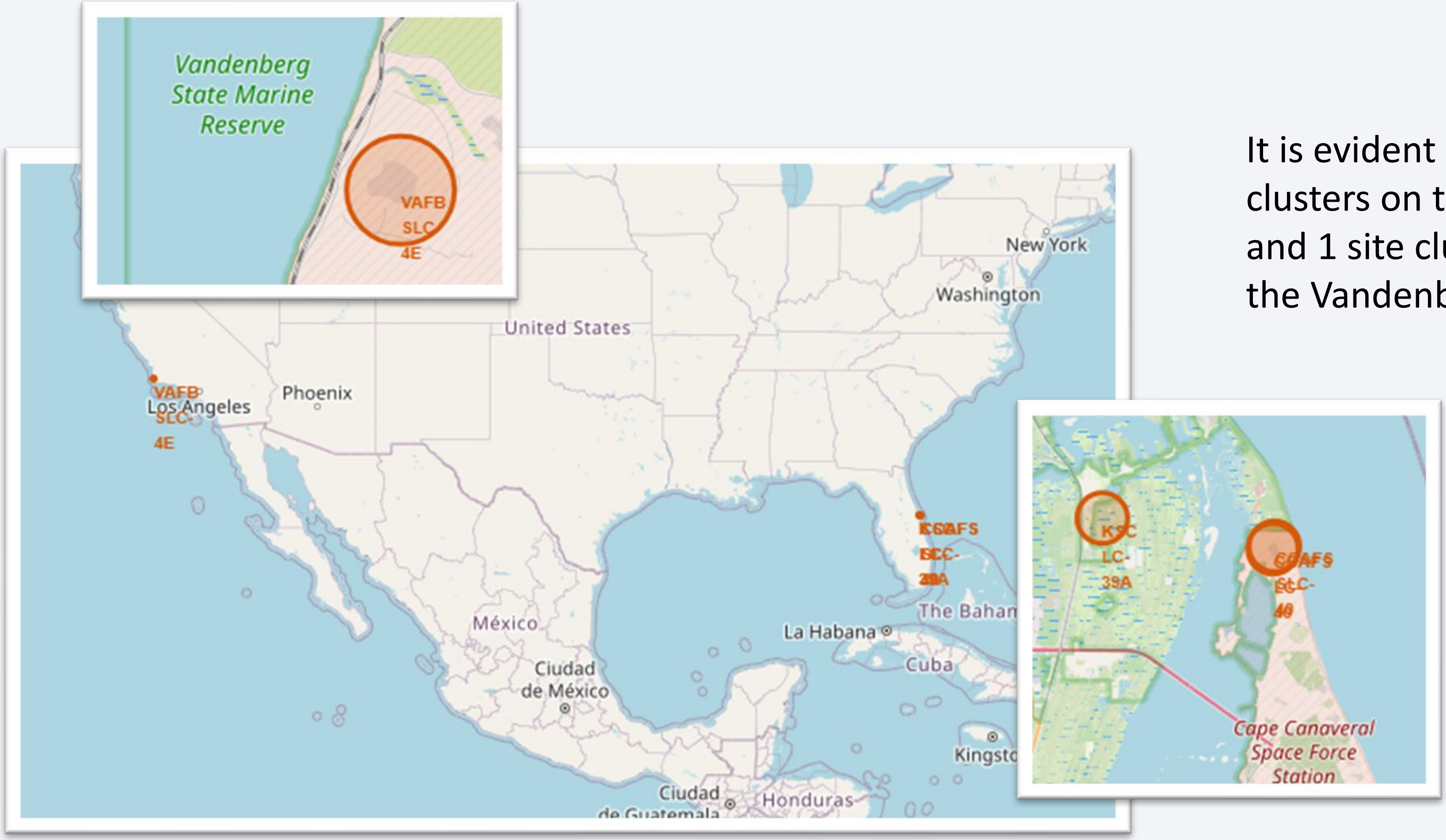
Out[69]:

Landing_Outcome	Count
No attempt	10
Success (drone ship)	5
Failure (drone ship)	5
Success (ground pad)	3
Controlled (ocean)	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1

Section 3

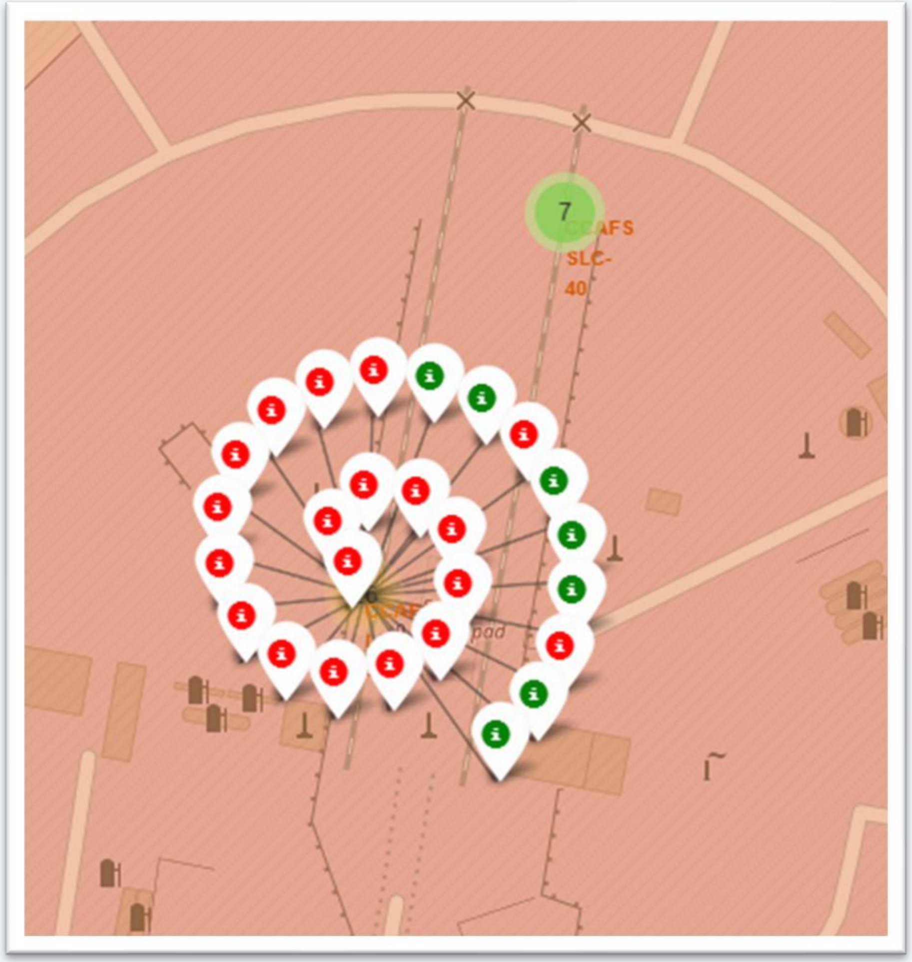
Launch Sites Proximities Analysis

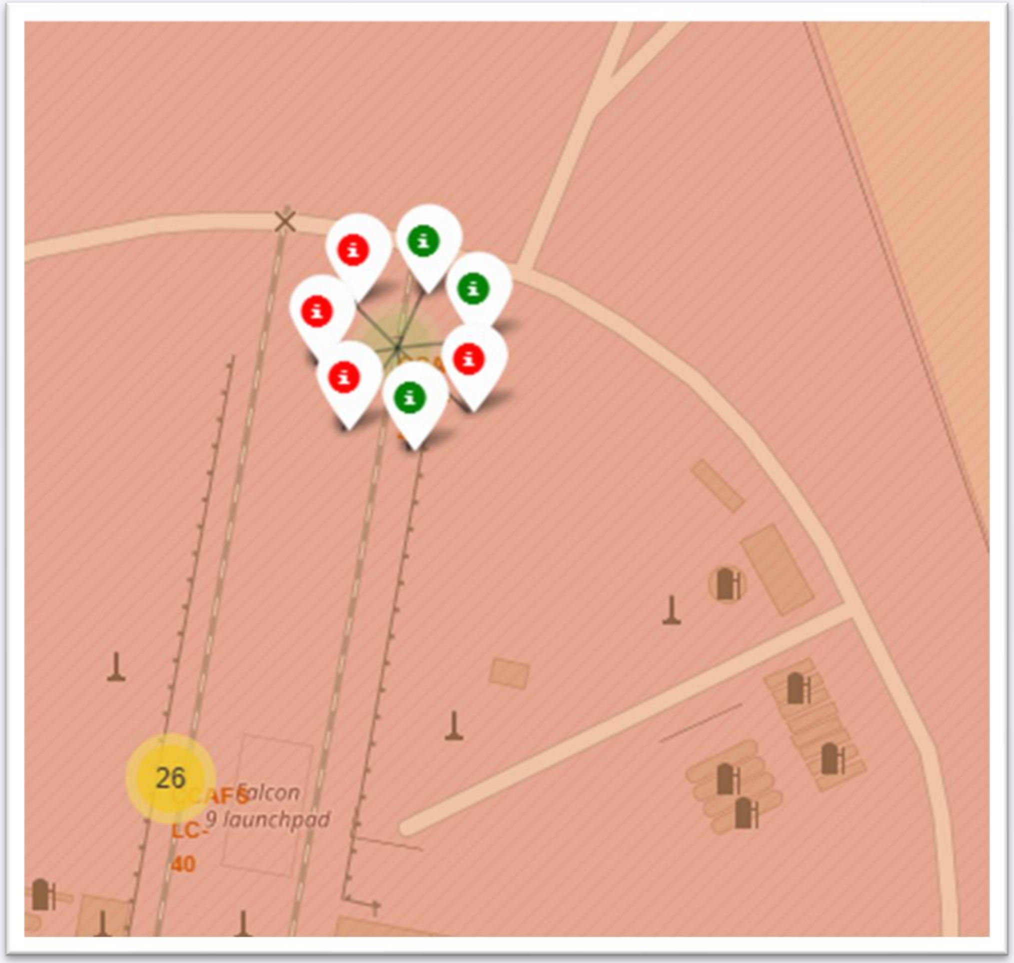
Launch Site Cluster Locations



It is evident there are 3 launch site clusters. clusters on the East coast near Cape Canaveral and 1 site cluster on the West coast near the Vandenberg State Marine Reserve.

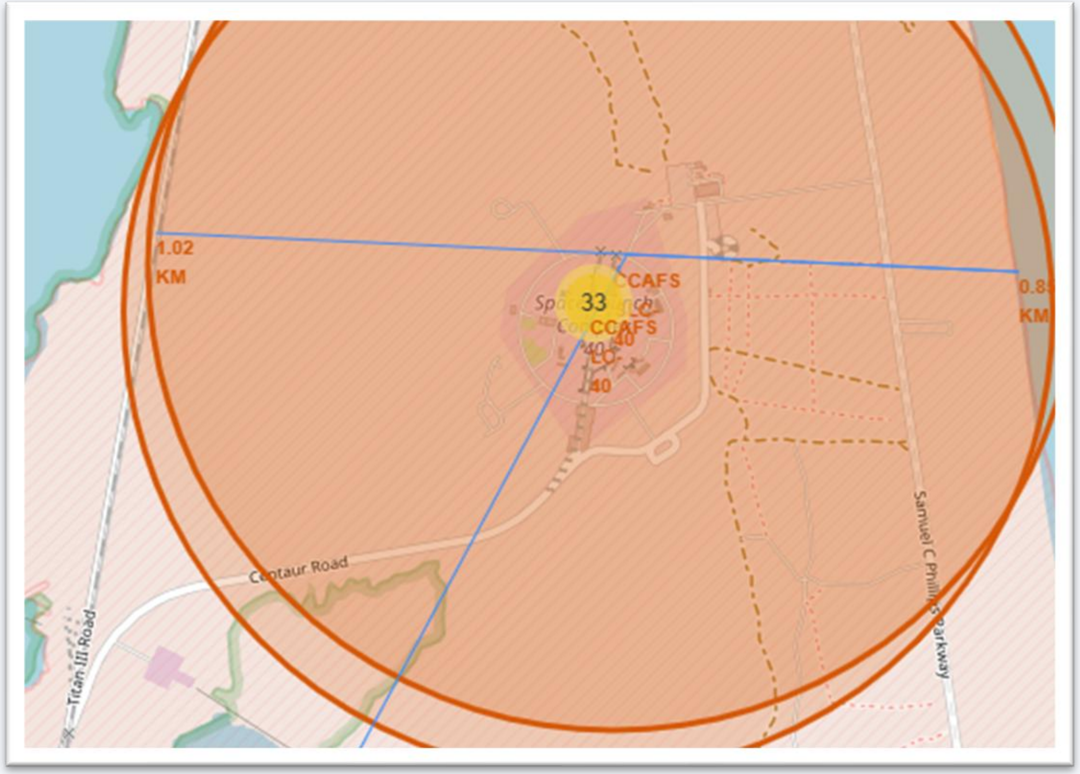
Launch Site Outcomes





Launch Site Proximity

Proximity to nearest city is 19.77 km
Proximity to railway 1.02 km
Proximity to coastline 0.84 km



Section 5

Predictive Analysis (Classification)

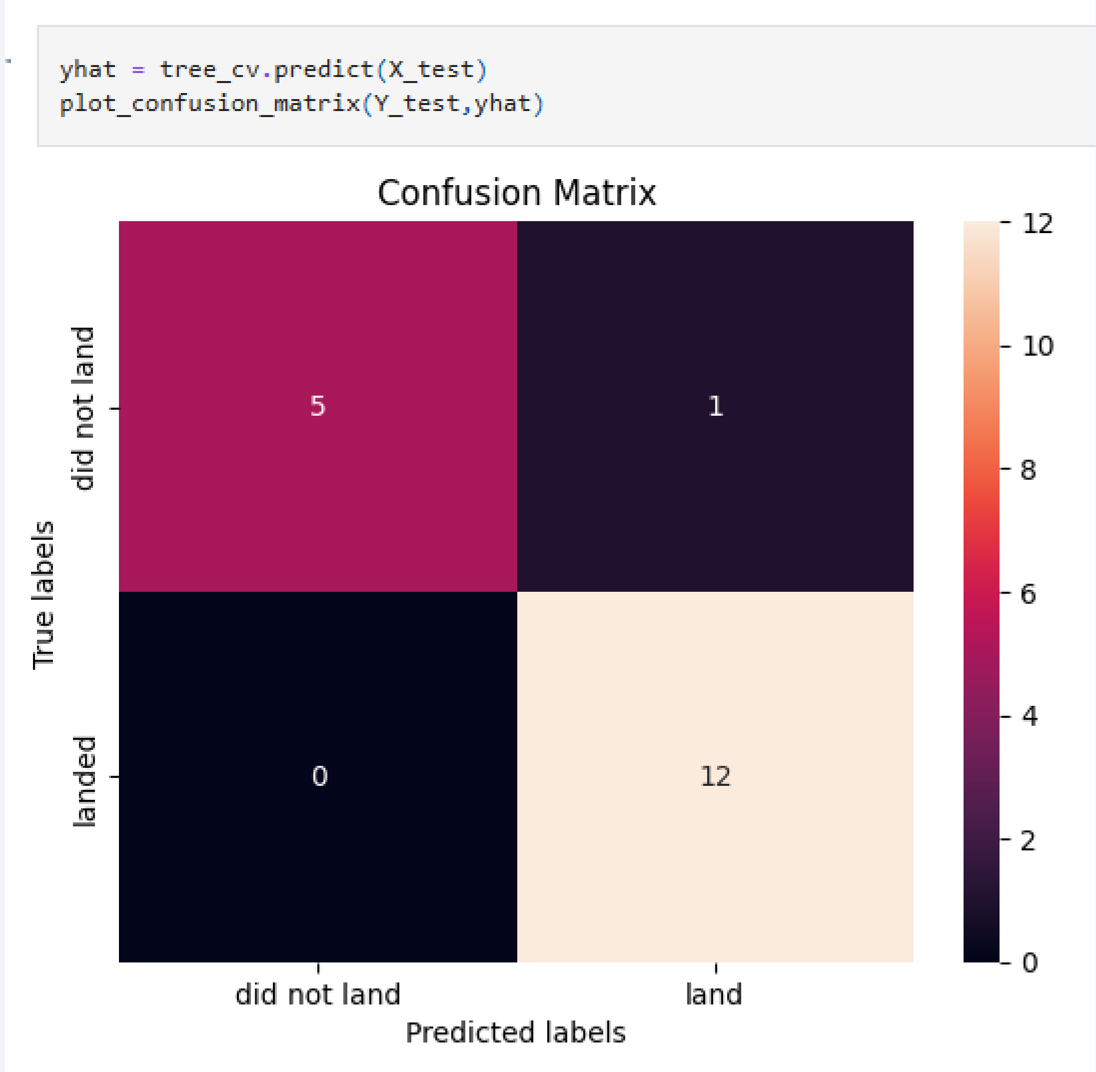
Classification Accuracy

- The best model was the Decision Tree Classifier with tuned parameters of:
- 'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random’
- It achieved train accuracy of 88% and test accuracy of 94%

```python
import pandas as pd, matplotlib.pyplot as plt

df.plot(y=["Best Scores Training", "Test Accuracy"], kind="bar", rot=0)
plt.ylabel("Scores")
plt.legend(loc='lower left');
```

Confusion Matrix



12 True Positives: True label is landed and predicted label is landed

1 False Positives: True label is not landed and predicted label is landed

Conclusions

- It is important to match orbit type and payload mass to launch site.
- Orbit ES-L1, GEO, HEO & SSO have 100% success rate, but ES-L1 is only based on one launch
- VLEO has a good success rate over 90% with many launches from there. However, it tends to have payload masses over 6000 kg
- The site KSC LC-39A has the most successful launches although there is a larger number of successes with payload masses between 2000 – 6000 kg
- Generally, it seems the best match is payload mass over 6000 kg, VLEO orbit and KSC LC-39A launch site
- The best machine learning model to identify if a launch is successful is a Decision Tree Classifier with parameters of 'criterion': 'entropy', 'max_depth': 6, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'splitter': 'random’

Thank you!