IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Ryan Welsh
23.02.2025

Outline

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

Executive Summary

- Launch Data was collated from the SpaceX Api and Falcon 9 Wikipedia Page
- Data was then cleaned and filtered to only include Falcon 9 Launches and relevant data needed
- Four classification models were created and assessed on the data to find which model was the best for future predictions on if a launch will succeed or fail.
- The success rate of flights have increased overall over the 10 year period
- After analysis it was determined that the launch sites are located next to Coastlines primarily and are also near other proximities such as Highways, Railways and Cities
- Launch's that have a payload in the range of 2k-5.3k have the best success rate. anything over 5.3k seems to have an adverse effect which causes failures
- The Tree Decision model was the best classification model with an accuracy score of 89% on the Training data
- However, all models had an accuracy score of 83.3% when they were tasked with predicting the Test data.

Introduction

- After the successful rocket launches in the Space sector for Space X I have been hired as a Data Scientist for a competitor called Space Y
- 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, The reason SpaceX has such great savings over competitors is because SpaceX can reuse the first stage.
- If we able to determine if the first stage will land, we can determine the cost of a launch. This information can be used if we want to bid against SpaceX for a rocket launch.
- We want to use historical launch data and using values such as Payloads and Launch Sites build a predictive model that will help us determine if the first stage of the launch will return successfully.

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
    - Launch Data was collated from the SpaceX Api and Wikipedia Page
- Perform data wrangling
    - I then organized the launch results into 2 groups Successful and Failed
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
    - Creating Multiple Prediction Models which will be evaluated against each other with a prediction of the first stage.

Data Collection

- Data was collated through API Calls from the Space X API and also through the Wikipedia of Falcon 9 Launches.
- Get Requests was used to call the SpaceX API Data
- Beautiful Soup was used in order to scrap tables from the Falcon 9 Wikipedia Pages
- This Data was then Normalized, cleaned and formatted to only include Falcon 9 Launch's and make sure only data that was relevant for our analysis was included.

Data Collection – SpaceX API

- Using Get Requests and creating functions I was able to call the data into a Json String I then Noramized this data, created a list of headers to clean and filter the data it so that It would only include data of Falcon 9 Launch’s.

- GitHub URL: [IBM-Projects/jupyter-labs-spacex-data-collection-api.ipynb at main · Rywel94/IBM-Projects]()

Now let's start requesting rocket launch data from SpaceX API with the following URL:

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

Python

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

Python

Check the content of the response

```python
print(response.content)
```

Python

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/NN6Ph

Data Collection - Scraping

- Using Beautiful Soup to extract tabular data from the Wikipedia page displaying results of Falcon 9 Launch's. Parsing this data into a Panda's Dataframe with relevant clean header for future use.

- GitHub URL: [IBM-Projects/jupyter-labs-webscraping.ipynb at main · Rywel94/IBM-Projects]()

More specifically, the launch records are stored in a HTML table shown below:

2020 [edit]

In late 2019, Gwynne Shotwell stated that SpaceX hoped for as many as 24 launches for Starlink satellites in 2020,[490] in addition to 14 or 15 non-Starlink launches. At 26 launches, 13 of which for Starlink satellites, Falcon 9 had its most prolific year, and Falcon rockets were second most prolific rocket family of 2020, only behind China's Long March rocket family.[491]

[hide] Flight No.	Date and time (UTC)	Version, Booster[b]	Launch site	Payload[c]	Payload mass	Orbit	Customer	Launch outcome	Booster landing
78	7 January 2020, 02:19:21[492]	F9 B5 ♺ B1049.4	CCAFS, SLC-40	Starlink 2 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third large batch and second operational flight of Starlink constellation. One of the 60 satellites included a test coating to make the satellite less reflective, and thus less likely to interfere with ground-based astronomical observations.[493]								
79	19 January 2020, 15:30[494]	F9 B5 ♺ B1046.4	KSC, LC-39A	Crew Dragon in-flight abort test[495] (Dragon C205.1)	12,050 kg (26,570 lb)	Sub-orbital[496]	NASA (CTS)[497]	Success	No attempt
	An atmospheric test of the Dragon 2 abort system after Max Q. The capsule fired its SuperDraco engines, reached an apogee of 40 km (25 mi), deployed parachutes after reentry, and splashed down in the ocean 31 km (19 mi) downrange from the launch site. The test was previously slated to be accomplished with the Crew Dragon Demo-1 capsule;[498] but that test article exploded during a ground test of SuperDraco engines on 20 April 2019.[419] The abort test used the capsule originally intended for the first crewed flight.[499] As expected, the booster was destroyed by aerodynamic forces after the capsule aborted.[500] First flight of a Falcon 9 with only one functional stage — the second stage had a mass simulator in place of its engine.								
80	29 January 2020, 14:07[501]	F9 B5 ♺ B1051.3	CCAFS, SLC-40	Starlink 3 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)
	Third operational and fourth large batch of Starlink satellites, deployed in a circular 290 km (180 mi) orbit. One of the fairing halves was caught, while the other was fished out of the ocean.[502]								
81	17 February 2020, 15:05[503]	F9 B5 ♺ B1056.4	CCAFS, SLC-40	Starlink 4 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fourth operational and fifth large batch of Starlink satellites. Used a new flight profile which deployed into a 212 km × 386 km (132 mi × 240 mi) elliptical orbit instead of launching into a circular orbit and firing the second stage engine twice. The first stage booster failed to land on the drone ship[504] due to incorrect wind data.[505] This was the first time a flight proven booster failed to land.								
82	7 March 2020, 04:50[506]	F9 B5 ♺ B1059.2	CCAFS, SLC-40	SpaceX CRS-20 (Dragon C112.3 ♺)	1,977 kg (4,359 lb)[507]	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
	Last launch of phase 1 of the CRS contract. Carries *Bartolomeo*, an ESA platform for hosting external payloads onto ISS.[508] Originally scheduled to launch on 2 March 2020, the launch date was pushed back due to a second stage engine failure. SpaceX decided to swap out the second stage instead of replacing the faulty part.[509] It was SpaceX's 50th successful landing of a first stage booster, the third flight of the Dragon C112 and the last launch of the cargo Dragon spacecraft.								
83	18 March 2020, 12:16[510]	F9 B5 ♺ B1048.5	KSC, LC-39A	Starlink 5 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Failure (drone ship)
	Fifth operational launch of Starlink satellites. It was the first time a first stage booster flew for a fifth time and the second time the fairings were reused (Starlink flight in May 2019).[511] Towards the end of the first stage burn, the booster suffered premature shut down of an engine, the first of a Merlin 1D variant and first since the CRS-1 mission in October 2012. However, the payload still reached the targeted orbit.[512] This was the second Starlink launch booster landing failure in a row, later revealed to be caused by residual cleaning fluid trapped inside a sensor.[513]								
84	22 April 2020, 19:30[514]	F9 B5 ♺ B1051.4	KSC, LC-39A	Starlink 6 v1.0 (60 satellites)	15,600 kg (34,400 lb)[5]	LEO	SpaceX	Success	Success (drone ship)

Data Wrangling

- Performed EDA on the Dataset to view what columns had missing values (LandingPads 28)
- Inspected DTypes in the dataset to identify if any values needed amending
- Viewed how many launches were from each site and how many launches were to a specific orbit
- Created a training label Class to the dataset to standardize the Successful [1] and Failed [0] landings of the first class as these were previously outlined as:
  - True Ocean - The mission outcome was successfully landed to a specific region of the ocean
  - False Ocean - The mission outcome was unsuccessfully landed to a specific region of the ocean.
  - True RTLS - The mission outcome was successfully landed to a ground pad
  - False RTLS - The mission outcome was unsuccessfully landed to a ground pad
  - True ASDS - The mission outcome was successfully landed on a drone ship
  - False ASDS - The mission outcome was unsuccessfully landed on a drone ship.
- GitHub URL: IBM-Projects/labs-jupyter-spacex-Data wrangling.ipynb at main · Rywel94/IBM-Projects

EDA with Data Visualization

- Using the Seaborn and Matplotlib Libraries I produced the below charts:
  - Scatterplot to plot the relationship between FlightNumber vs PayloadMass,
  - Scatterplot to plot the relationship between FlightNumber vs LaunchSite,
  - Scatterplot to plot the relationship between LaunchSite vs PayloadMass,
  - Bar chart for the success rate of each orbit
  - Scatterplot to see if there is any relationship between FlightNumber and Orbit type.
  - Payload vs. Orbit Scatterplot chart to view a potential relationship between the two
  - A Line Chart to help visualize the launch success over a period of years to establish a yearly trend.
- GitHub URL: IBM-Projects/jupyter-labs-eda-dataviz-v2.ipynb at main · Rywel94/IBM-Projects

EDA with SQL

- After importing SQL Alchemy I ran the following queries using SQL to:
  - Create a new table with the Dataset
  - 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, failed landing outcomes when landing pad was drone ship , booster versions and launch site for the months in year 2015.
  - Ranked landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- GitHub URL: [IBM-Projects/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · Rywel94/IBM-Projects]

Build an Interactive Map with Folium

- Using Folium I was able to create markers for All Launch Sites
- These markers were then Colour coded to help visualize the successful and failed launches from the sites
- I then added a mouse cursor to my folium map therefore helping me create pinpoints to it's nearest proximities
  - Railway
  - City
  - Coastline
  - Highway
- I added these features to help me draw analysis and discussion points from the launch sites
  - Why do you think a launch site would want to be close to a coastline?
  - Do you expect a launch site to be close to a city?

- GitHub URL: IBM-Projects/lab-jupyter-launch-site-location-v2.ipynb at main · Rywel94/IBM-Projects

Build a Dashboard with Plotly Dash

- I created a dashboard using the Plotly Dash library to present my visualization in an interactive manner.
- My Dashboard consisted of:
    - A dropdown list to drill down between each Launch Site
    - A Pie Chart showing Success Rates of launches
    - A range slider to adjust payload amount
    - A scatterplot that would display the relationship between Payload and Landing outcome.
        - This Plot was colour coded by booster version and each booster version was also able to be removed/filtered by deselecting it from the legend
- These features were added to the Dashboard as this was the detailed information required by stakeholders to determine what factors contributed to a successful launch.
- GitHub URL: [IBM-Projects/spacex_dash_app.py at main · Rywel94/IBM-Projects]()

Predictive Analysis (Classification)

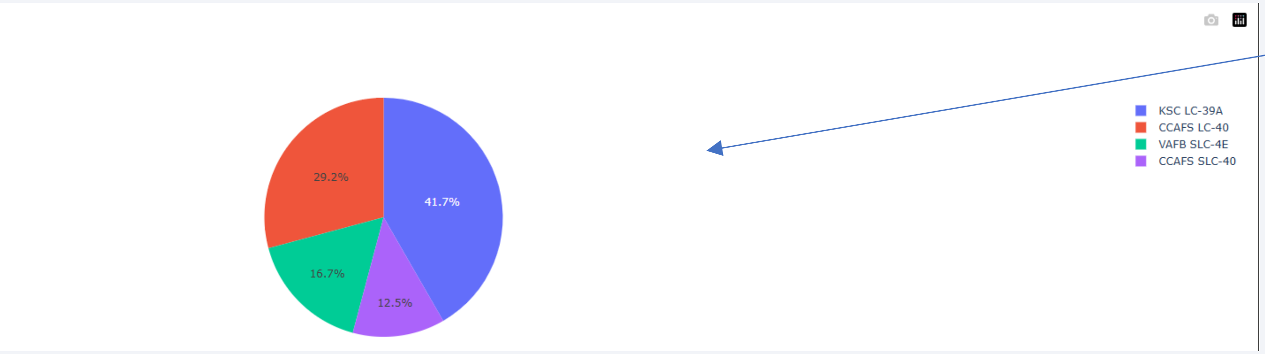
- I split the dataset into a train/test split then ran the training split through 4 models:
  - Logistic Regression
  - Support Vector Machine (SVM)
  - Decision Tree Classifier
  - K Nearest Neighbours (KNN)
- Each model was cross-validated and evaluated over a variety of hyperparameters using the GridSearchCV library to determine which model was the best for predicting the success of a flight.
- This would then enable us to assess the likelihood of the prediction value for the success of the first stage by using the best model to draw predictions using the test data.

- GitHub URL: [IBM-Projects/SpaceX-Machine-Learning-Prediction-Part-5-v1.ipynb at main · Rywel94/IBM-Projects]

Results

- Launch Flight to orbits ES-L1, GEO, HEO and SSO all have 100% success rate
- The success rate of flights from 2010-2020 was an overall increase.
- The 3 Launch Sites for SpaceX are all relatively close to the coastline, this is most likely due to having the landing pads located in the Ocean or on Drone ships

Success Rate from All Launch Sites – KSC LC-39A was the most successful with 41.7%



Predictive Analysis Results - Decision Tree was only slightly the best model however every model predicted a 83.33% Success Rate of the first launch from the test data

In [71]:

```
# Train Scores
print("accuracy :",logreg_cv.best_score_)
print("accuracy :",svm_cv.best_score_)
print("accuracy :",tree_cv.best_score_)
print("accuracy :",knn_cv.best_score_)
```

```
accuracy : 0.8464285714285713
accuracy : 0.8482142857142856
accuracy : 0.875
accuracy : 0.8482142857142858
```

In [72]:

```
# Test Score
print("test set accuracy :",logreg_cv.score(X_test, Y_test))
print("test set accuracy :",svm_cv.score(X_test, Y_test))
print("test set accuracy :",tree_cv.score(X_test, Y_test))
print("test set accuracy :",knn_cv.score(X_test, Y_test))
```

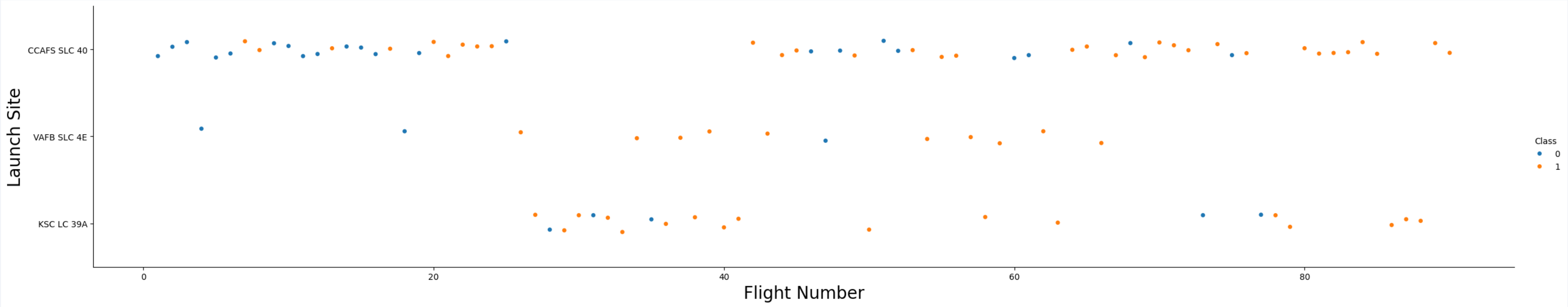
```
test set accuracy : 0.8333333333333334
test set accuracy : 0.8333333333333334
test set accuracy : 0.8333333333333334
test set accuracy : 0.8333333333333334
```

From comparing accuracy of the 4 models, they all perfomed to a similar standard the only significant difference was one example where the tree model training data fitted better than the other models.

Section 2

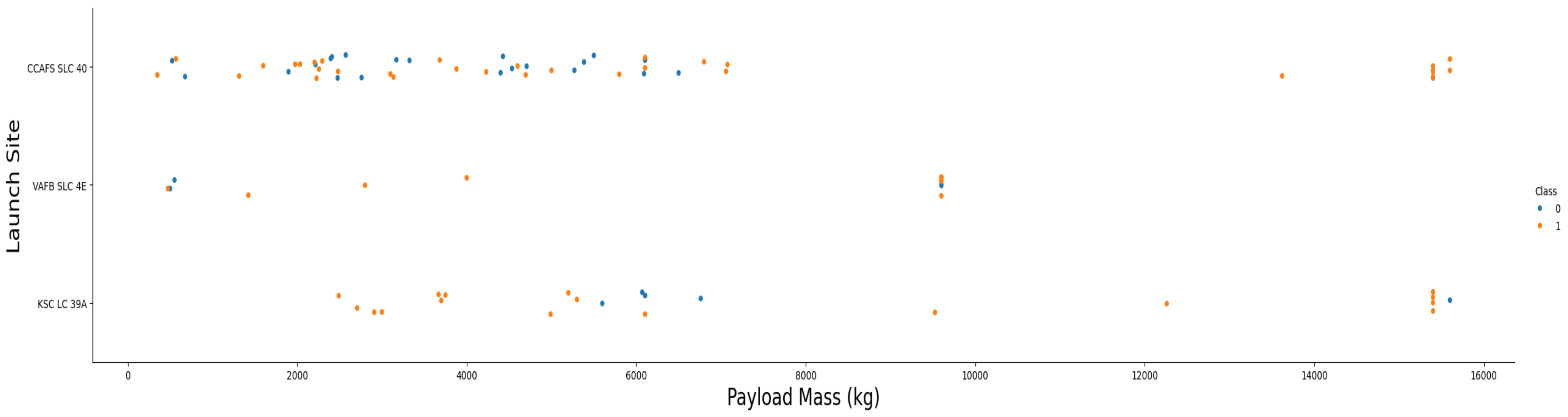
Insights drawn from EDA

Flight Number vs. Launch Site



- CCFCS SLC 40 Appears to be where the most of the early flight launches took place
- As the flight number increases the success rate of the flight also increased
- The flight numbers seem to be in sequence from two specific launch sites, CCAFS SLC 40 and KSC LC 93a, a reason for this may be that weather conditions greatly differentiate between the two sites depending on the time of the year or in a specific season than the other.

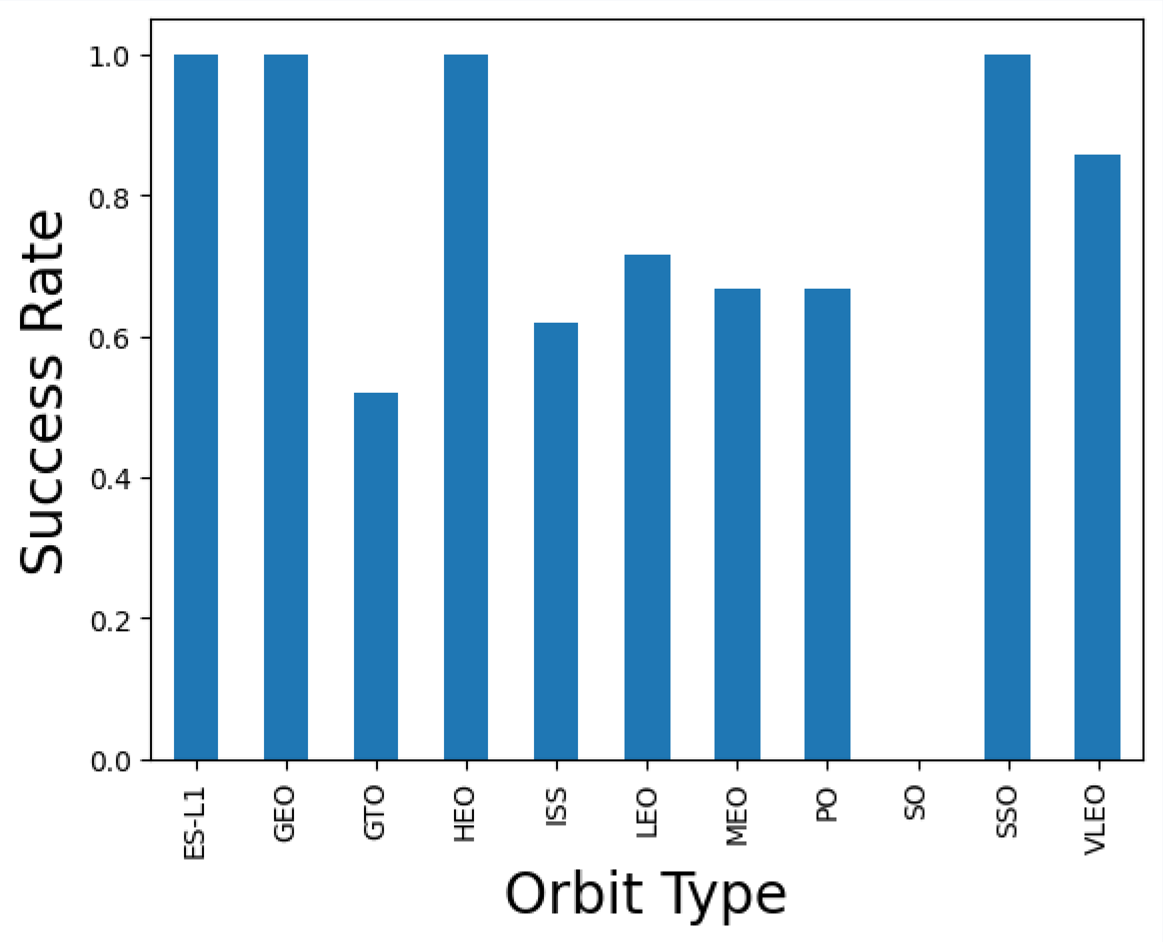
Payload vs. Launch Site



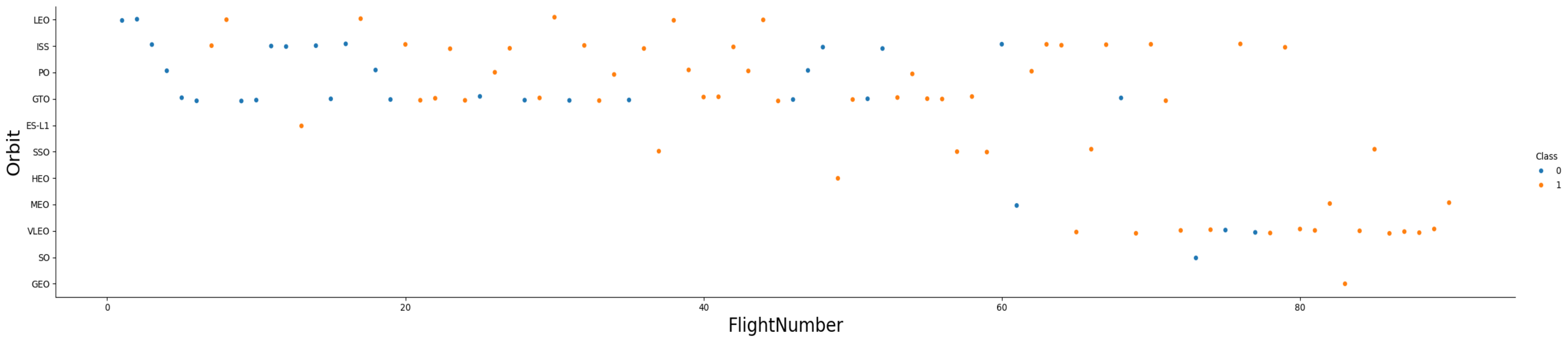
- CCAFS SLC 40 and KSC LC 93a appear to be in favor of heavier payloads
- The success rate seems to be indifferent at lower payloads under the threshold of 8000, however it does seem to be more successful at higher payloads

Success Rate vs. Orbit Type

- As Previously stated Launch Flight's to orbits ES-L1, GEO, HEO and SSO all have 100% success rate.
- VLEO is another orbit with a high success rate
- GTO, ISS, LEO, MEO, PO all seem to have around average success rate
- SO has never had a successful launch flight

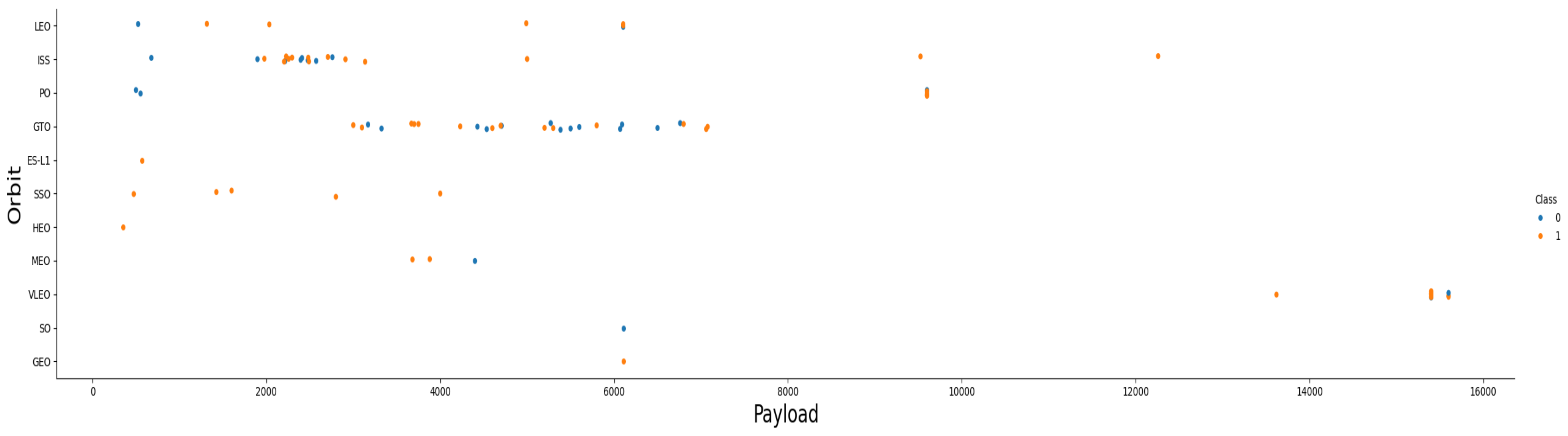


Flight Number vs. Orbit Type



- Overall the Flight Number positively correlates with the first stage recovery of launch flights
- 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.
- The SO and GEO orbits have only had one flight, this shows a potential skewing of the previous data in terms of how choice of orbit is in terms of successful flight.

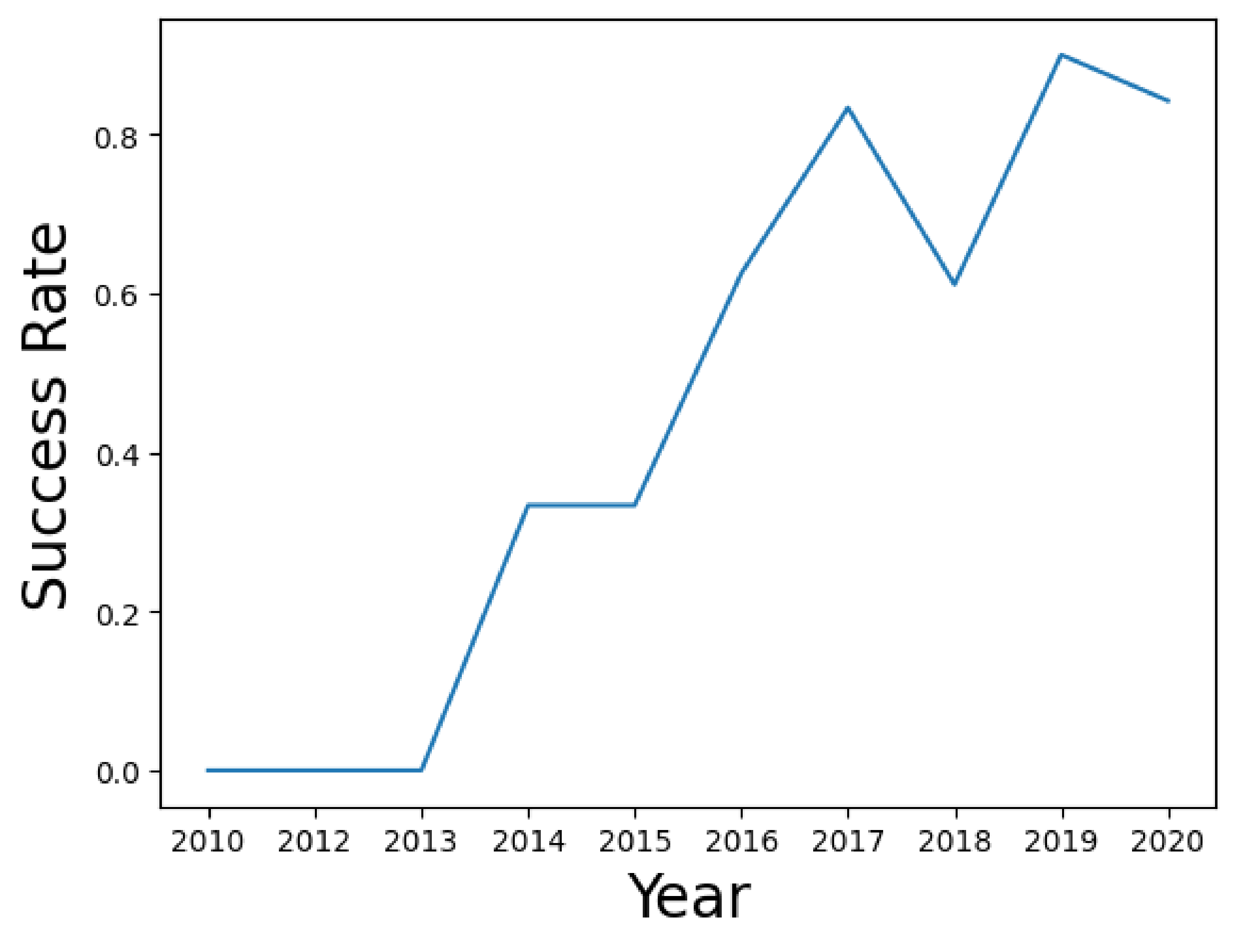
Payload vs. Orbit Type



- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- The success rate of flights from 2010-2020 was an overall increase.
- The start of the increasing trend started in 2013
- 2018 saw a decrease In the success rate in comparison to 2017 taking the success rate back to 2016 levels
- The success rate hit its peak in the year 2019 with it hitting around 90%



All Launch Site Names

- To find the names of the unique launch sites I ran an SQL query
- The results from the query were presented here.
- This gave me the 4 individual launch sites where flights were launched from.

```
%%sql
SELECT DISTINCT LAUNCH_SITE
FROM SPACEXTBL;
```

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

Launch Site Names Begin with 'CCA'

- I wished to drill down further into the launch sites and Find 5 records where launch sites begin with `CCA`
- To do this I ran another SQL query
- The results only displayed one of the launch sites which began with the characters ‘CCA’
- From this I can deduce that the records most be in alphabetical order currently as no CCAFS SLC-39 records appear

```
%%sql
SELECT LAUNCH_SITE
FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE 'CCA%'
LIMIT 5;
```

Launch_Site
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

Total Payload Mass

- To calculate the total payload carried by boosters from NASA I ran the following query.
- From the result of the query I established that the total amount of payload NASA carried on launch sites from SpaceX was 45596kg.

```
%%sql
SELECT SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Customer = 'NASA (CRS)';
```

SUM(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

- To find average payload mass carried by booster version F9 v1.1, I ran the following query.

```
%%sql
SELECT AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL
WHERE Booster_Version LIKE 'F9 v1.0%';
```

- The result of the displays that the Average amount of payload booster version F9 v1.1 carries is 340.4kg.

AVG(PAYLOAD_MASS__KG_)
340.4

First Successful Ground Landing Date

- A significant date for SpaceX would have been the first successful landing outcome on ground pad
- In order to find this date, I ran the following query.
- The 22nd of December 2015 was the date of the first successful landing.
- It took more than 5 years to achieve a successful landing. (The first ever Falcon 9 launch was on the 4th of June 2010)

```
%%sql
SELECT MIN(Date)
FROM SPACEXTBL
WHERE Landing_Outcome = 'Success (ground pad)';
```

MIN(Date)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- I wanted to filter down my data of the names of boosters which have successfully landed on drone ship and were in a payload mass range greater than 4000 but less than 6000
- I ran the following query to achieve this result
- The list of Booster that fit the criteria are listed in the graphic.

```
%%sql
SELECT BOOSTER_VERSION
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Success (drone ship)'
    AND 4000 < PAYLOAD_MASS__KG_ < 6000;
```

Booster_Version
F9 FT B1021.1
F9 FT B1022
F9 FT B1023.1
F9 FT B1026
F9 FT B1029.1
F9 FT B1021.2
F9 FT B1029.2
F9 FT B1036.1
F9 FT B1038.1
F9 B4 B1041.1
F9 FT B1031.2
F9 B4 B1042.1
F9 B4 B1045.1
F9 B5 B1046.1

Total Number of Successful and Failure Mission Outcomes

- Finding the total number of successful and failure mission outcomes is an important part of our business question
- To find this I had to write a query that would display the mission outcome of every launch flight and group them.
- The results shown show 1 failed in flight, 99 were successful but may be represented differently in the records and one was Successful but has been grouped as it's own individual record as the Payload Status is unclear

```
%%sql
SELECT MISSION_OUTCOME, COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
GROUP BY MISSION_OUTCOME;
```

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

Boosters Carried Maximum Payload

- Like my previous query I wanted to List the names of the booster which have carried the maximum payload mass
- To do this I wrote this query
- The list that fit the criteria consists of 12 Boosters which are listed in the graphic.

```
%%sql
SELECT DISTINCT BOOSTER_VERSION
FROM SPACEXTBL
WHERE PAYLOAD_MASS__KG_ = (
    SELECT MAX(PAYLOAD_MASS__KG_)
    FROM SPACEXTBL);
```

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

2015 Launch Records

- To try a specify a potential reason why a failure would occur I wanted to find the failed landing outcomes when the landing pad was a drone ship, the booster version used, and launch site for failed launches in the first quarter of 2015
- There was only two failures, the booster versions were different however the launch site was the same.
- I am unable to determine if this was a factor for the failure however as it is such a small result.

```
%%sql
SELECT LANDING_OUTCOME, BOOSTER_VERSION, LAUNCH_SITE, SUBSTR(Date, 6, 2) AS Month
FROM SPACEXTBL
WHERE LANDING_OUTCOME = 'Failure (drone ship)'
    AND SUBSTR(Date, 1, 4) = '2015';
```

Landing_Outcome	Booster_Version	Launch_Site	Month
Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	01
Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	04

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

- I then ranked 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.
- As shown in the image the most prevalent outcome was that of no attempt of landing the 1st phase
- Drone ship landings occurred at a similar amount however the success rate was 50%
- Controlled ship landing were successful 60% of the time but this results for this are drawn from a sample size that is 50% of no attempt and Drone Ship

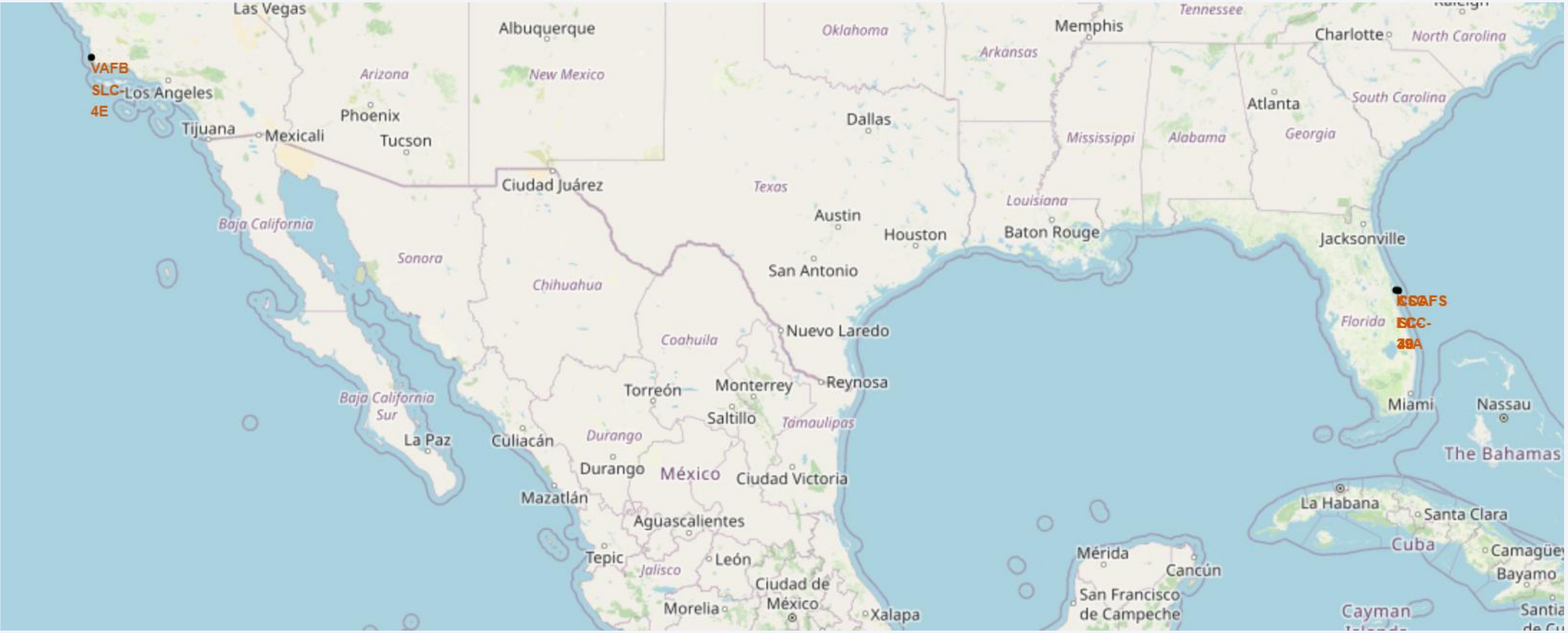
```
%%sql
SELECT LANDING_OUTCOME, COUNT(LANDING_OUTCOME) AS TOTAL_NUMBER
FROM SPACEXTBL
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
GROUP BY LANDING_OUTCOME
ORDER BY TOTAL_NUMBER DESC
```

Landing_Outcome	TOTAL_NUMBER
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

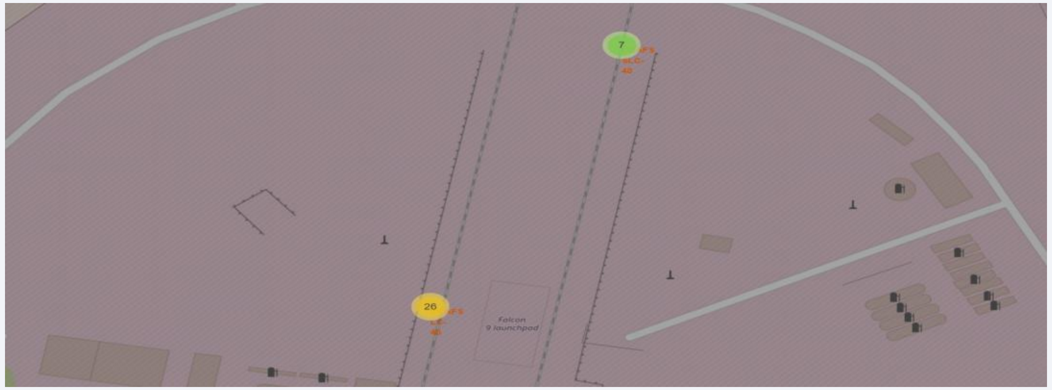
SpaceX Launch Sites

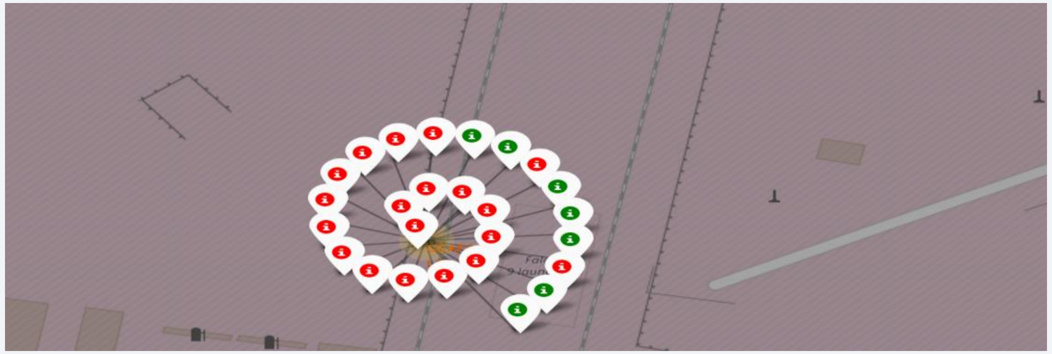


- The Above map shows the locations of the SpaceX Launch sites, from the locations marked we can determine that these locations are:
  - Quite close to the Equator Line,
  - Close to the coast, This will be due to having landing pads, drone ships located in the oceans either side therefore less distance to recuperate the parts.

Initial Launch Site Analysis

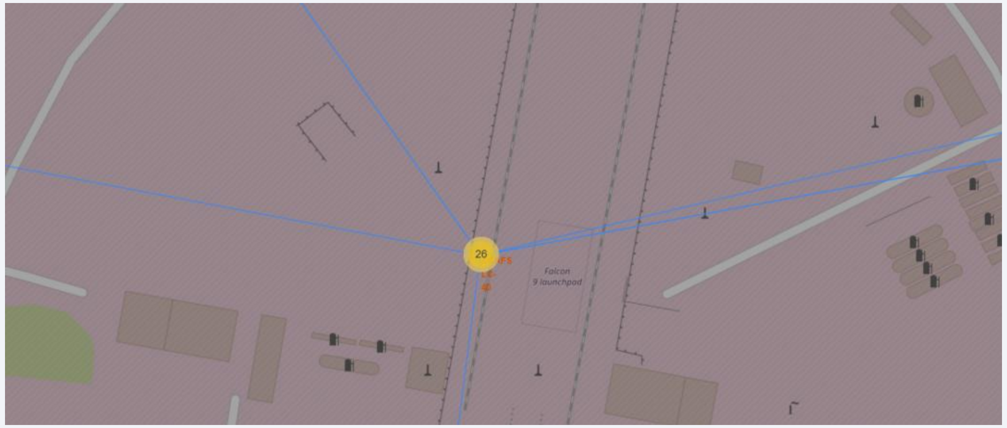
- I then added markers to each launch site to display the number of launches from each specific area and site
- We could then drill down into each site to see via a colour coded marker the flights that were successful or failures
- As you can see for this site CCAFS-LC-40 there was 26 Launches in total with only 7 of these resulting in a successful landing of the 1st phase





What other factors influence a Launch Site location?

- Next I wanted to further explore the site I had initially looked at and see how far important proximities are to the launch site to see if this influences where a launch site is located.
- The 4 proximities I chose were;
    - Coastline
    - Railway
    - City
    - Highway
- To do this I added a map cursor to my map so I was able to grab the latitude and longitude these proximities.
- The results are displayed on the side
- From these results we can draw reasons why a launch site would be close to these proximities
    - Launch Sites are close to railways, this could be due to the need of transportation of parts
    - Launch Sites relationship to highways is similar of that to railways, this could be due to same reason as outlined above
    - Launch Sites appear to be next to the coastline This will be due to having landing pads, drone ships located in the oceans therefore less distance to recuperate the parts.
    - Launch Sites seem to be a fair distance away from cities but not as far as first anticipated, seems their is a balance between making sure citizens will be safe from launches but at the same time you will need to be close to a city if you wish to have employees as they will need somewhere to live



```
# find coordinate of the closet coastline
# e.g.,: Lat: 28.56367  Lon: -80.57163
launch_site_lat = 28.5623
launch_site_lon = -80.57736
coastline_lat = 28.56462
coastline_lon = -80.56613
distance_coastline = calculate_distance(launch_site_lat, launch_site_lon, coastline_lat, coastline_lon)
print(distance_coastline,' km')

1.127020188343149  km
```

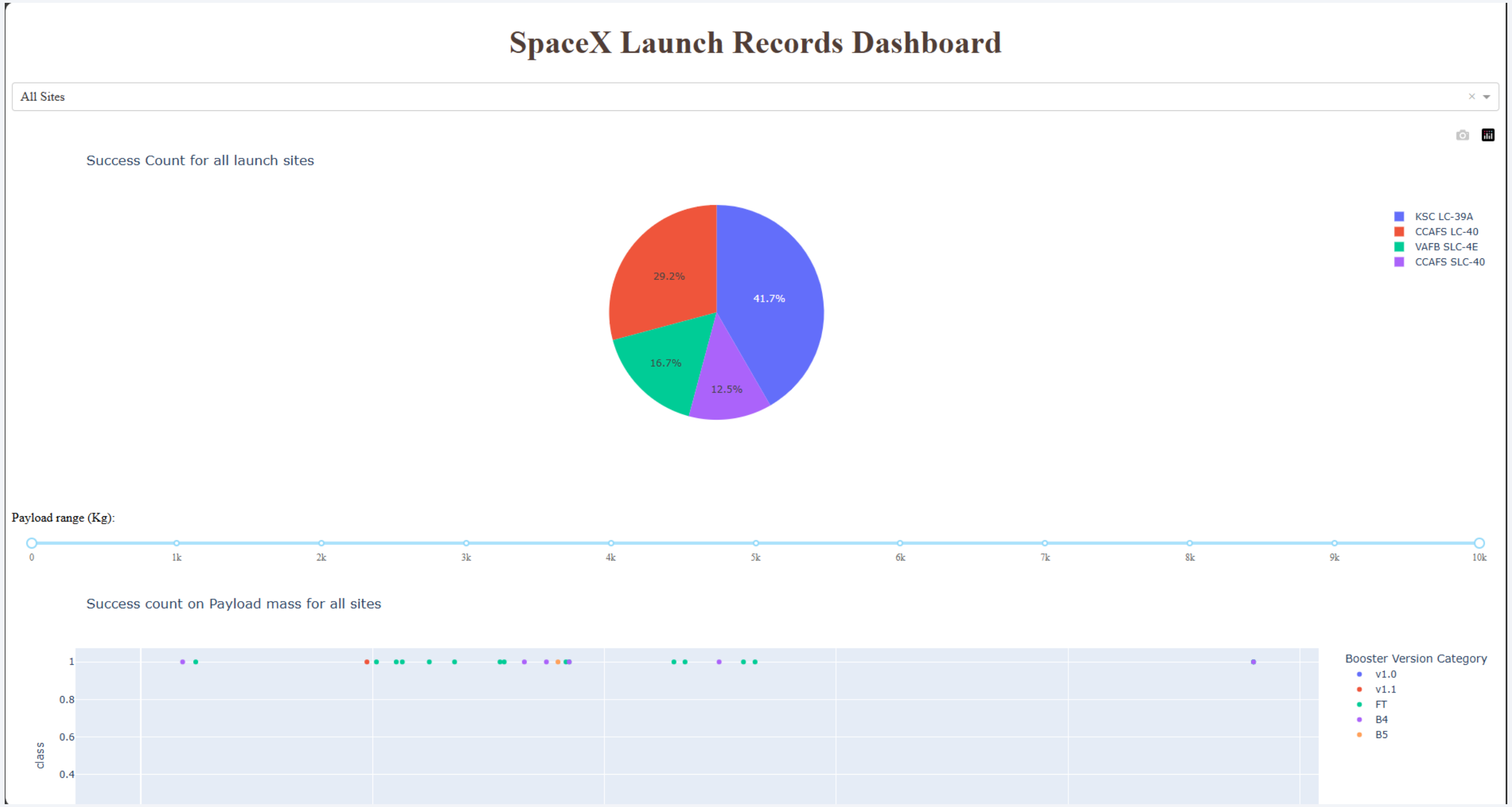
```
distance_highway = calculate_distance(launch_site_lat, launch_site_lon, closest_highway[0], closest_highway[1])
print('distance_highway =',distance_highway, ' km')
distance_railroad = calculate_distance(launch_site_lat, launch_site_lon, closest_railroad[0], closest_railroad[1])
print('distance_railroad =',distance_railroad, ' km')
distance_city = calculate_distance(launch_site_lat, launch_site_lon, closest_city[0], closest_city[1])
print('distance_city =',distance_city, ' km')

distance_highway = 0.6466125886656682  km
distance_railroad = 1.331392990978995  km
distance_city = 23.148682326538864  km
```

Section 4

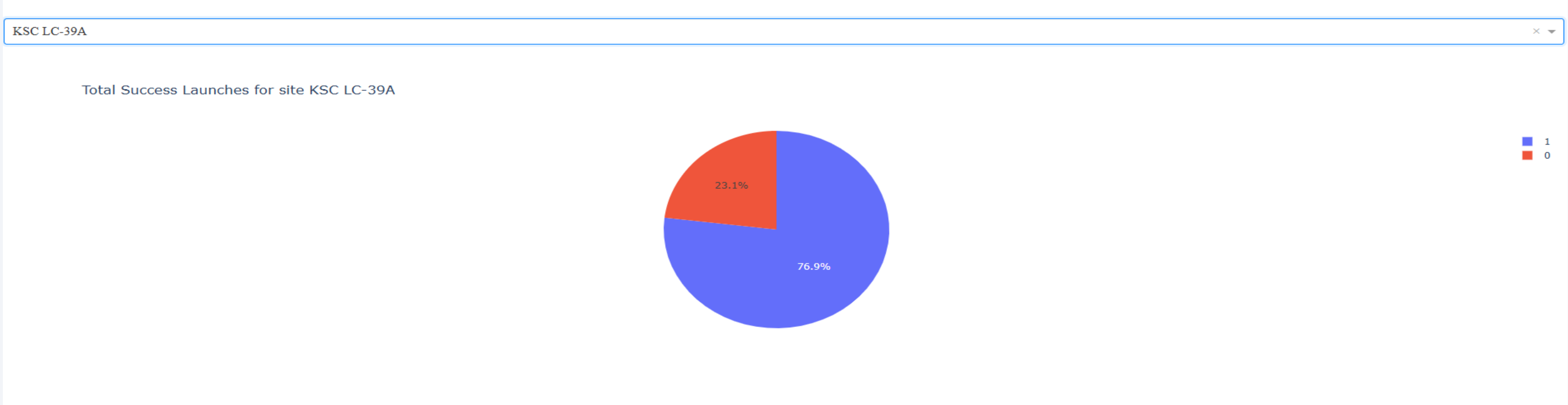
Build a Dashboard with Plotly Dash

Launch Records Dashboard

- To present my findings I decided to create an interactive dashboard
- This includes dropdown options of the different launch sites, a payload range slider which can be used to adjust this measure and a checkbox function on the booster versions which enables you to filter based on that parameter
- The initial pie chart when first loading the dashboard is a pie chart displaying the successful launches from all launch sites, as you can see from the image KSC-LC-39A was the best site with 41.7% of successful launches coming from that site



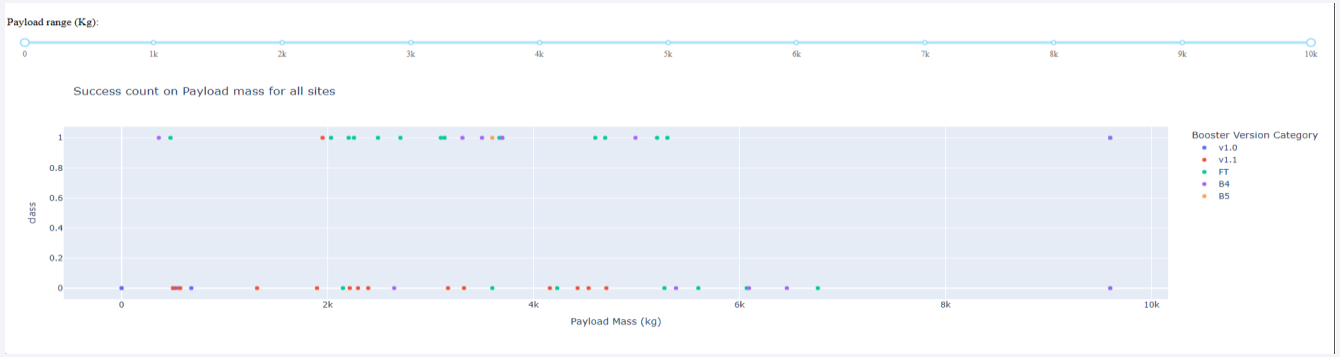
How successful was the most successful?

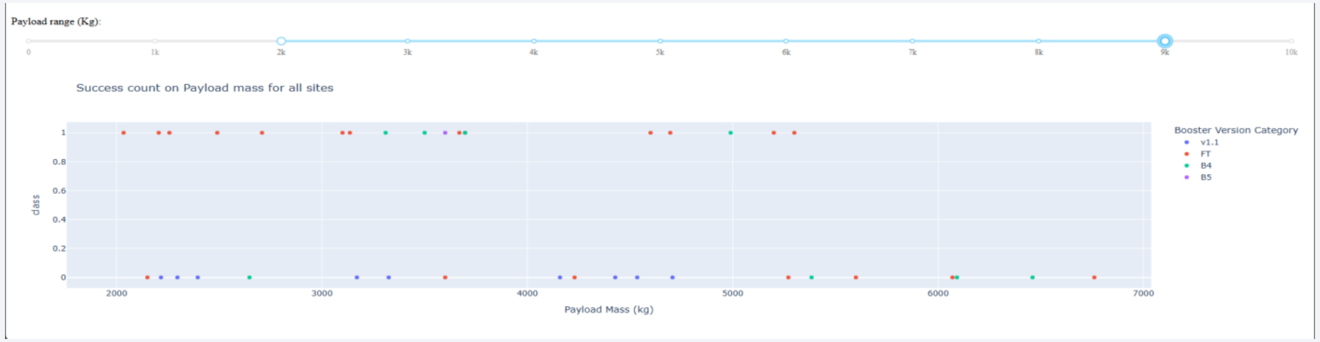


- As shown above KSC-LC-39A had a success rate of 76.9%
- This means 3 out of 4 launches are likely to have the phase 1 return if launched from this site.

Payload Vs Launch Outcome

- The first image of this is the default scatterplot of the Payload vs the Launch outcome from viewing this graph I seen their was two ranges I wanted to further analyse, the lower payload level under 2k and the higher range over 5k.
- The second graph has the ranges adjusted from 0 – 2k. From this graph we can see that at lower levels of payload it Is more likely to affect the success rate
- However when we adjust the range to over 2k as shown in the third graph we see that when the payload hits a certain threshold (5.3k) all the payload that are above this have a 0% success rate. There is 2 additional examples of when the payload is at 10k but this has a 50% rate.

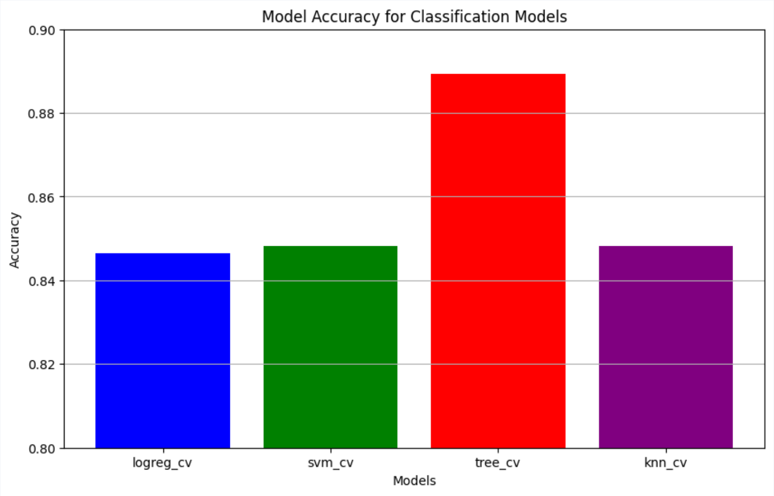


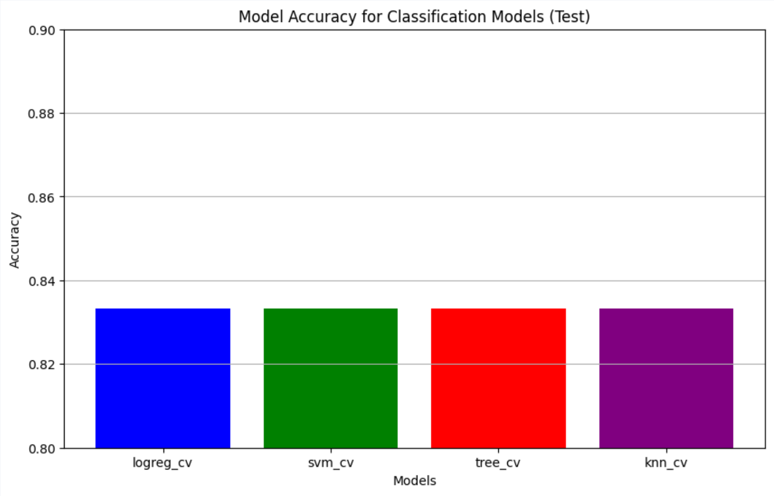


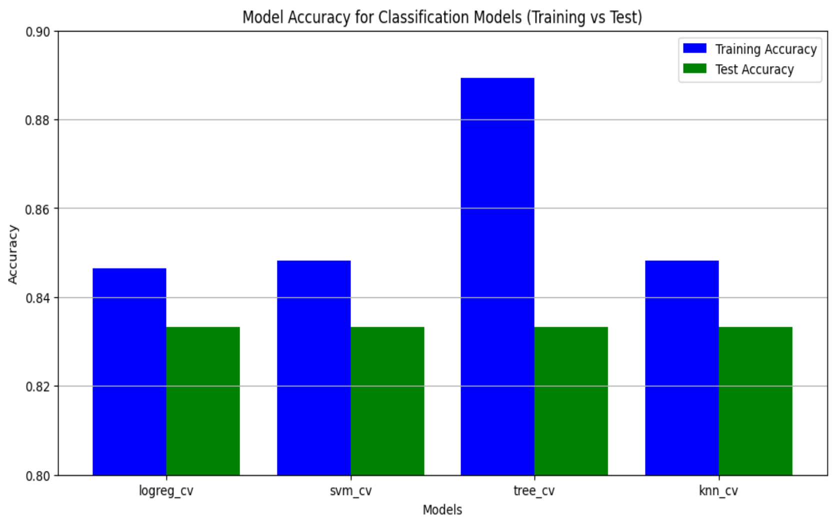
Section 5

Predictive Analysis (Classification)

Classification Accuracy



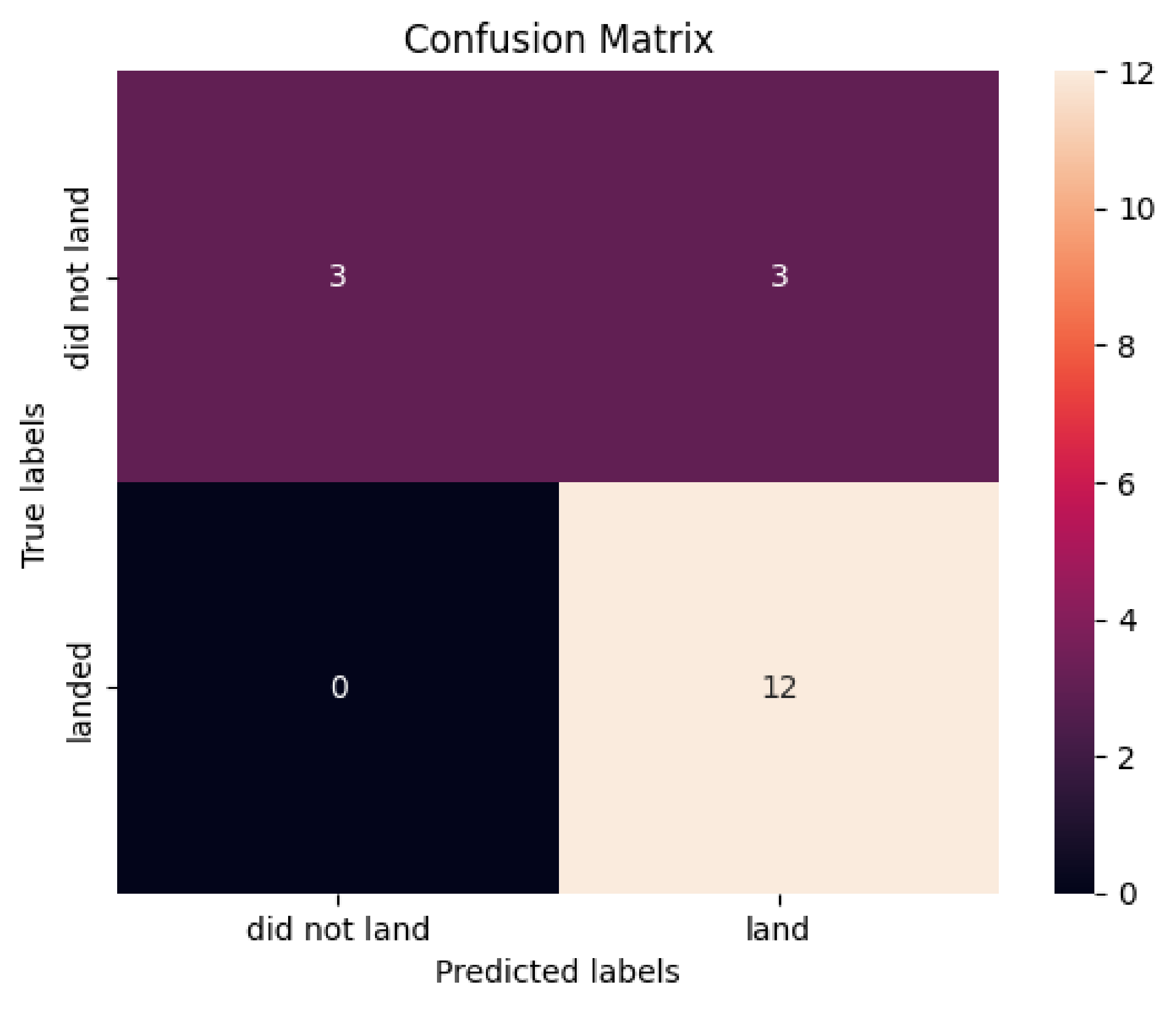




- The results for the classification models shows the following
  - Tree Decision making was the best model with the training data with a result of 89% Accuracy
  - However, when it came to the testing data it performed to same level as the other models

Confusion Matrix

- The confusion matrix for the tree decision model shows that the model accurately picked the 12 that landed without any false negatives
- The model also was accurate in predicting 3 failed landings however it did unfortunately have some false positives predicting a successful launch when the actual result was a failure.



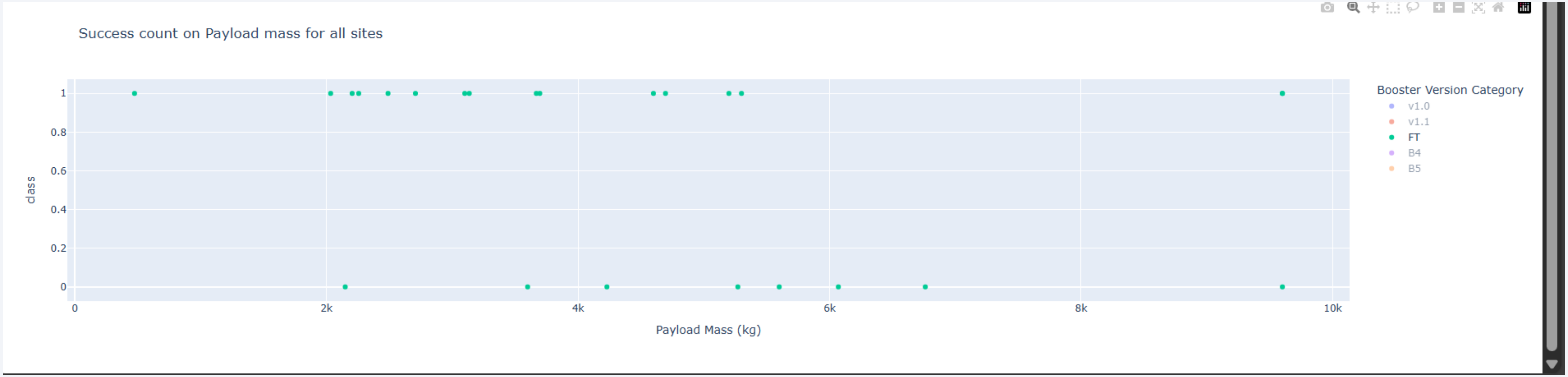
Conclusions

- Launch sites are located next to Coastlines the reason for this is as the landing pads are mainly ocean based such as drone ships and landing pads. This also gives room for error on failed launches or launches that are deemed no attempt.
- Launch sites were also located near Railways, Highways and Cities this will be due to the need to get resources delivered to sites Aswell as recruitment for staff
- Launch's that have a payload in the range of 2k-5.3k have the best success rate.
- Payloads over 5.3k seem to have a high fail rate. This perhaps is the law of diminishing returns.
- The Tree Decision model was the best classification model with an accuracy score of 89% on the Training data
- Despite the above models had an accuracy score of 83.3% when they were tasked with predicting the Test data.
- I feel the next steps would be to run the Tree decision model with another subset of launch data perhaps the Falcon 1 to see If it is able to produce similar results of that of the Falcon 9 data

Appendix

- All notebooks for this project can be found on my GitHub

- The below image is a snippet from my dashboard where I have used the legend to only show the results from version booster FT



Thank you!