IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kirstin Ferrie
5/12/2024

Outline

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

Executive Summary

The space age is upon us with SpaceX in the lead. As a competitor, it is vital to understand the success of SpaceX and improve on it.

By looking at SpaceX data through the SpaceX API and by web scraping, we were able to obtain launch site, booster version, success rate, payload masses, orbit types, and much more information.

Exploratory data analysis revealed KSC LC-39A was the most successful launch site where proximity to a coastline and railway is of importance.

Flight number and success rate is launch site dependent, where flight numbers greater than 25 are more successful for KSC LC-39A.

Introduction

The commercial space age is here as SpaceX gains worldwide attention for a series of historic milestones. SpaceX is the only private company ever to return a spacecraft from low-earth orbit, which it first accomplished in December 2010. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars, whereas other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.

In this project we investigated the cost of a launch of a rocket by determining if the first stage would land after initial take-off. Recovering the first stage (the most expensive stage of a rocket) can save millions of dollars per rocket launch, making space travel more affordable. We investigated the factors that influenced successful first stage recovery. By creating dashboards using data from similar scenarios, we can determine the cost of a launch and if the first stage can be re-used.

SpaceY is challenged with becoming a market competitor and providing affordable space travel and bidding against SpaceX for a rocket launch.

Section 1

Methodology

Methodology

Executive Summary

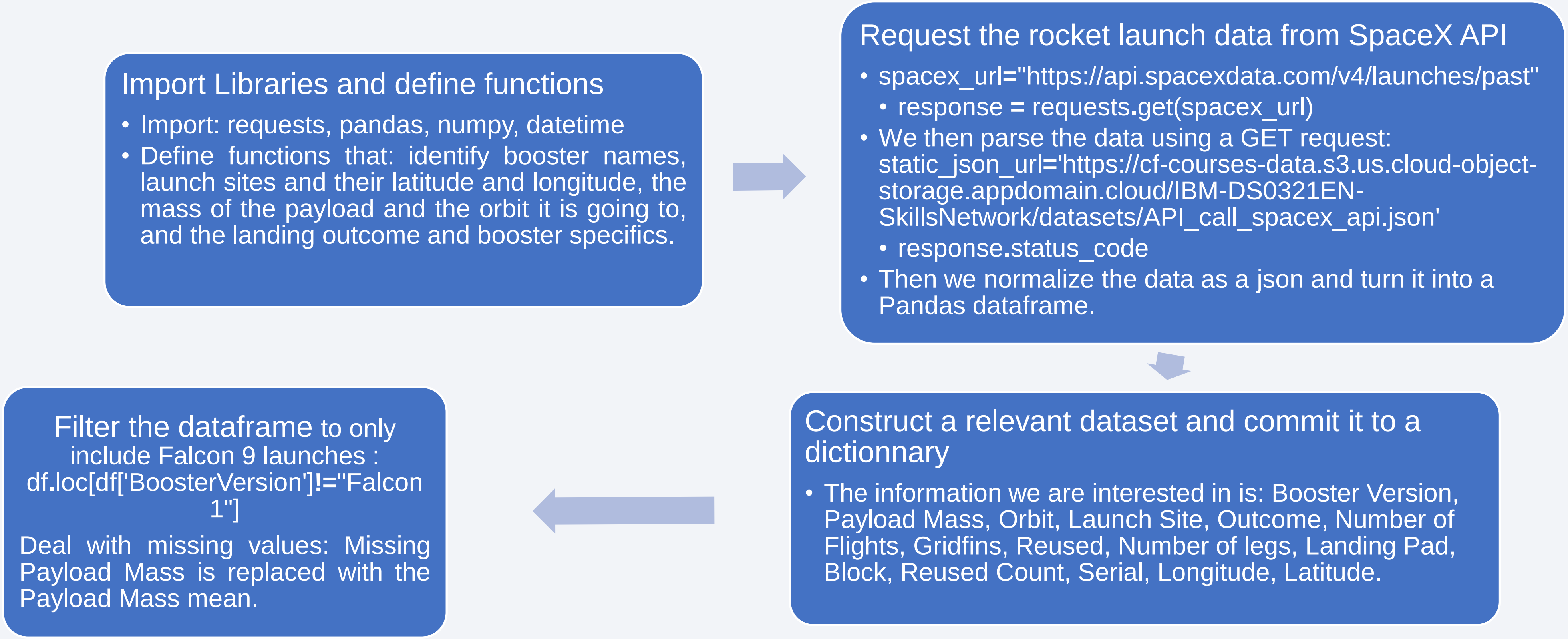
- Data collection methodology:
  - Data was obtained through the SpaceX API and by using BeautifulSoup to perform web scraping.
- Perform data wrangling
  - Using pandas and numpy, we were able to identify how many launches occurred at each site, how many launches occurred in each orbit and we were able to classify whether a launch was successful or not.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
- Using classification models we normalize the data, split the data and develop models from Logistic regression, Support vector machine, Decision trees and K nearest neighbor algorithms.

Data Collection

1. Perform a get request on the url https://api.spacexdata.com/v4/launches/past
2. We normalize the data into a data frame
3. We use the API and predetermined functions to make sense of the data and save this as a new data frame
4. We limit the data frame to information that is only applicable to us, Falcon 9 launches
5. We deal with missing values by using means

Data Collection – SpaceX API

https://github.com/kirsuf/Capstone/blob/6820f6cda9922969029e1963bddfb6e00838e4ed/jupyter-labs-spacex-data-collection-api%20(1).ipynb



Import Libraries and define functions

- Import: requests, pandas, numpy, datetime
- Define functions that: identify booster names, launch sites and their latitude and longitude, the mass of the payload and the orbit it is going to, and the landing outcome and booster specifics.

Request the rocket launch data from SpaceX API

- spacex_url="https://api.spacexdata.com/v4/launches/past"
  - response = requests.get(spacex_url)
- We then parse the data using a GET request: static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json'
  - response.status_code
- Then we normalize the data as a json and turn it into a Pandas dataframe.

Construct a relevant dataset and commit it to a dictionnary

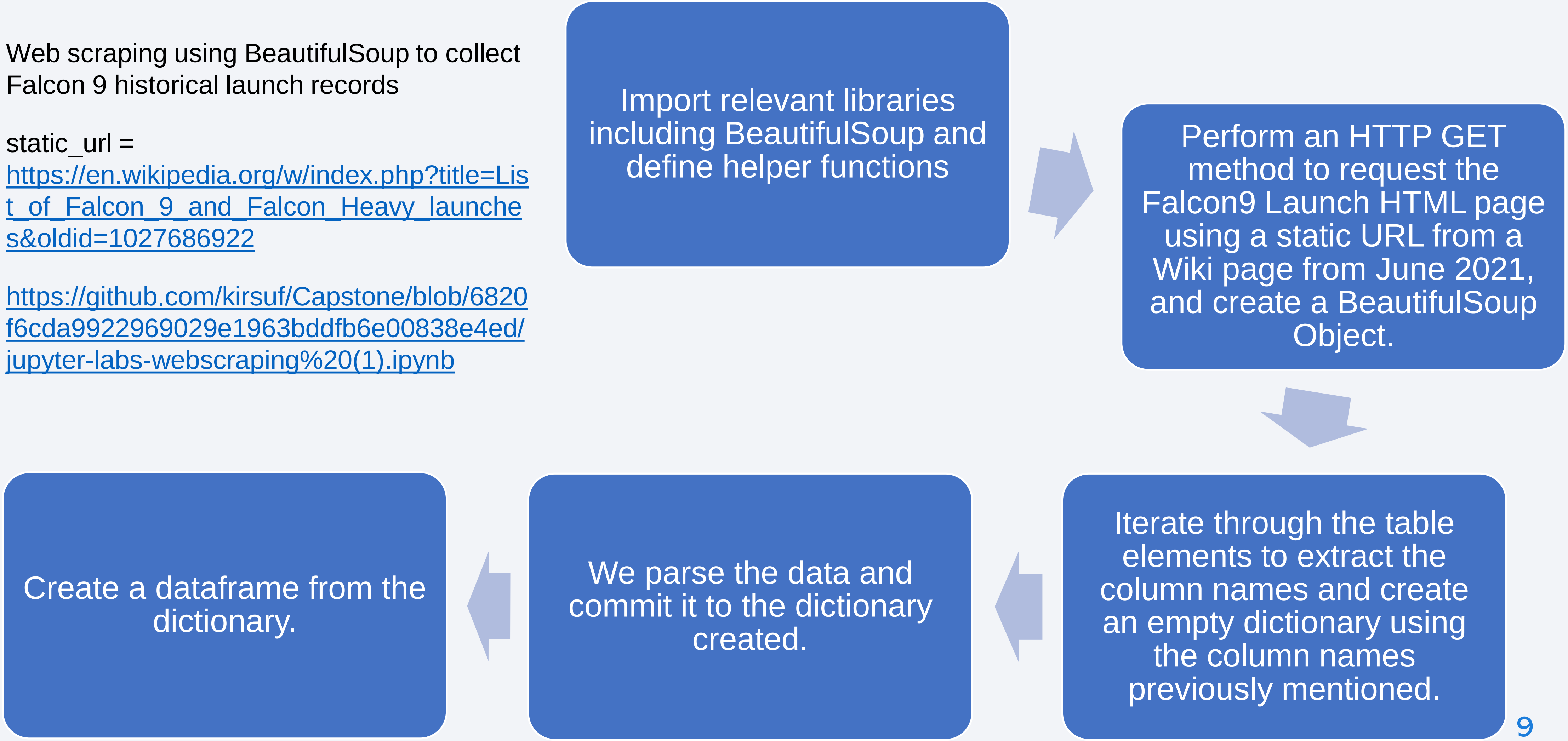
- The information we are interested in is: Booster Version, Payload Mass, Orbit, Launch Site, Outcome, Number of Flights, Gridfins, Reused, Number of legs, Landing Pad, Block, Reused Count, Serial, Longitude, Latitude.

Filter the dataframe to only include Falcon 9 launches : df.loc[df['BoosterVersion']!="Falcon 1"]

Deal with missing values: Missing Payload Mass is replaced with the Payload Mass mean.

Data Collection - Scraping

- Web scraping using BeautifulSoup to collect Falcon 9 historical launch records
- static_url = https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922
- https://github.com/kirsuf/Capstone/blob/6820f6cda9922969029e1963bddfb6e00838e4ed/jupyter-labs-webscraping%20(1).ipynb



Import relevant libraries including BeautifulSoup and define helper functions

Perform an HTTP GET method to request the Falcon9 Launch HTML page using a static URL from a Wiki page from June 2021, and create a BeautifulSoup Object.

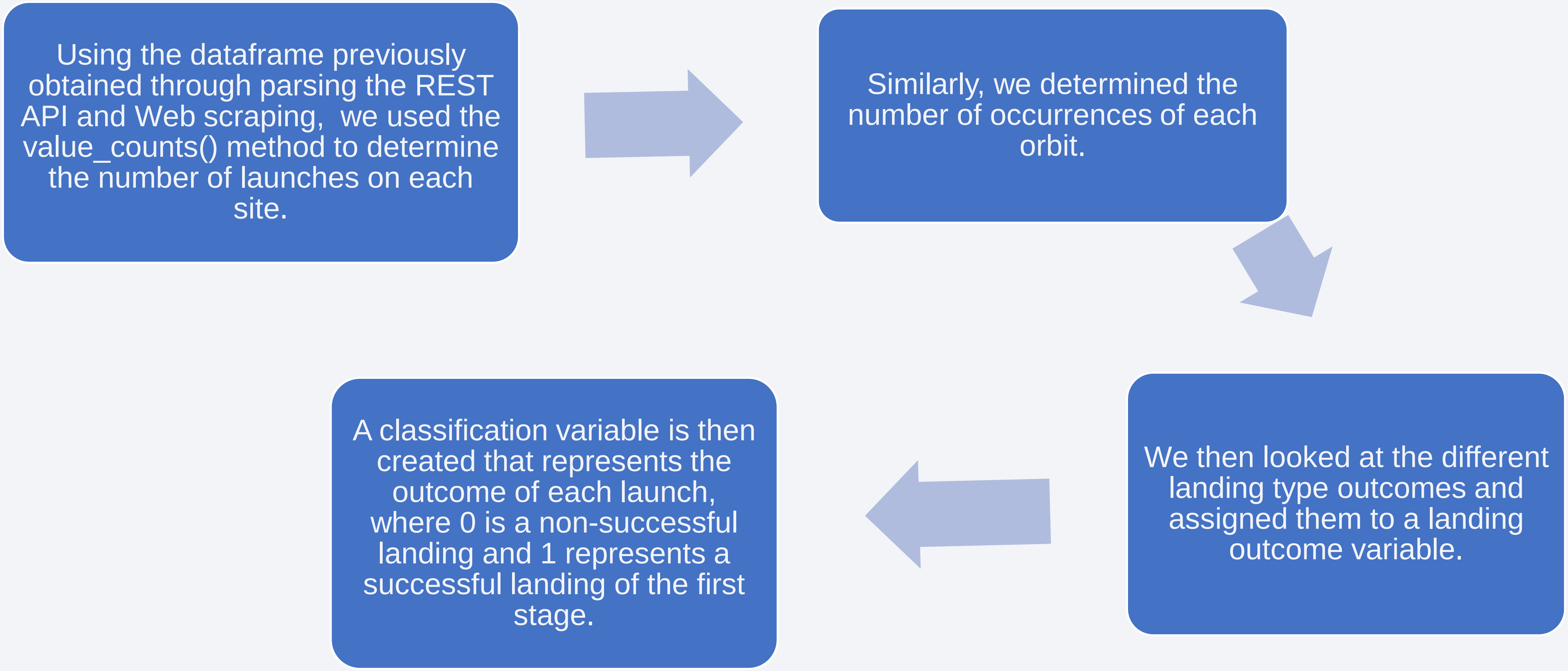
Iterate through the table elements to extract the column names and create an empty dictionary using the column names previously mentioned.

We parse the data and commit it to the dictionary created.

Create a dataframe from the dictionary.

Data Wrangling

https://github.com/kirsuf/Capstone/blob/6820f6cda9922969029e1963bddfb6e00838e4ed/labs-jupyter-spacex-Data%20wrangling%20(1).ipynb



Using the dataframe previously obtained through parsing the REST API and Web scraping, we used the value_counts() method to determine the number of launches on each site.

Similarly, we determined the number of occurrences of each orbit.

We then looked at the different landing type outcomes and assigned them to a landing outcome variable.

A classification variable is then created that represents the outcome of each launch, where 0 is a non-successful landing and 1 represents a successful landing of the first stage.

EDA with Data Visualization

Identifying the relationship between various variables and successful outcomes can help in the decision-making process when deciding on factors such as orbit types and the importance of payload mass based on hidden patterns and the insights gained from the data.

Scatter plots were drawn to analyze the relationships that existed between:

- Flight Number and Launch Site
- Payload Mass and Launch Site
- Flight Number and Orbit Type
- Payload Mass and Orbit Type

A bar chart was produced to represent the average success rate of each orbit type.

A line graph represents the yearly success trend.

https://github.com/kirsuf/Capstone/blob/6e38ead5d97a4f8dd240f73c3126078189fb81d6/EDAwithvisuals.ipynb

EDA with SQL

1. Determine the names of unique launch sites (DISTINCT).
2. Display 5 records where launch sites begin with the string 'CCA'.
3. Display the total payload mass carried by boosters launched by NASA (CRS).
4. Display average payload mass carried by booster version F9 v1.1.
5. List the date when the first successful landing outcome in ground pad was achieved.
6. List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
7. List the total number of successful and failure mission outcomes
8. List the names of the booster versions which have carried the maximum payload mass using a subquery.
9. List the records which will display the month names, failure landing outcomes in drone ship ,booster versions, launch site for the months in year 2015.
10. Rank the count of landing outcomes between the date 2010-06-04 and 2017-03-20, in descending order.

https://github.com/kirsuf/Capstone/blob/08c7e7bdb75c1cb4d9d34d5a1480573159954c18/jupyter-labs-eda-sql-coursera_sqllite%20(3).ipynb

Build an Interactive Map with Folium

The interactive Folium map contained markers for each site that indicated a successful or failed launch site. Launch sites were contained in marker clusters as there were many markers in a confined area with similar coordinates.

Distances were calculated from the launch site to its proximities such as railways, coastlines, cities and highways. Polylines were added to the map of the most successful launch site.

Knowing the maximum and minimum distances of launch sites to its proximities plays an important part in determining a location for a new launch site.

https://github.com/kirsuf/Capstone/blob/b3e629df0e0432dfc71c15b64340147b688b2552/Foliumlaunch-site-location-v2%20(1).ipynb

Build a Dashboard with Plotly Dash

A dashboard was created using the launch data from SpaceX.

A pie chart represents the success counts for launch sites, automatically defaulting to all sites. A dropdown allows for a specific launch site to be selected. This helps indicate which launch site is most successful.

A scatterplot indicating success count on payload mass with booster versions dictating the color of the markers on the scatterplot. A range slider can define payload range that is of interest and investigate patterns such as which boosters are more or less successful at higher or lower payloads.

https://github.com/kirsuf/Capstone/blob/c65c268058db2b6041db9d91dd29270ebd9705c6/spacex_dash_app.py

Predictive Analysis (Classification)

Now that we finished the exploratory analysis, the next step is to determine the training labels and build a predictor using machine learning algorithms. After using the ‘Class’ column as the label, first thing to do is normalizing the data. We split the normalized data into test/train sets, The training data is divided into validation data, a second set used for training data.

For the model development phase, we use the following algorithms:

- Logistic regression
- Support vector machine
- Decision trees
- K nearest neighbor

We build a grid search object for each of the algorithms and find the best parameters of the model (hyperparameters tuning), then we choose the most accurate model.

https://github.com/kirsuf/Capstone/blob/96179521211715c2b9b23ceda846fc06d6f67979/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- First stage landing success massively improved since 2013 with the first successful launch being 22 December 2015.
- Launch site proximity to a coastline and railway is of importance with the most successful launch site being KSC LC-39A.
- Payload masses that are towards the heavier end are more likely to be successful and orbits ES-L1, GEO, HEO and SSO are most successful.
- K Nearest Neighbour is the best predictor model, however, all models predicted false positives.
- Flight number and success rate is launch site dependent, where flight numbers greater than 25 are more successful for KSC LC-39A.

Section 2

Insights drawn from EDA

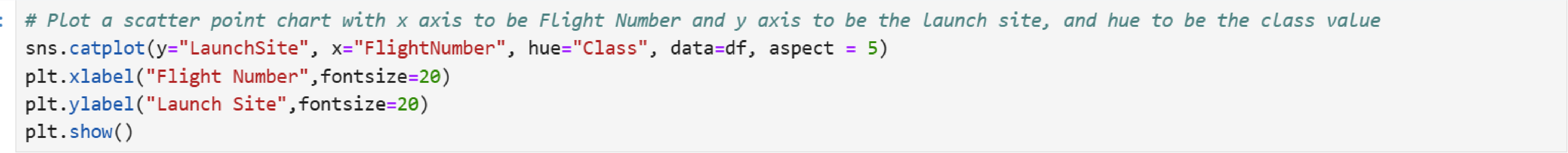
Flight Number vs. Launch Site

Flight Number between 0 and 30 were mostly unsuccessful at Launch Site CCAFS SLC 40.

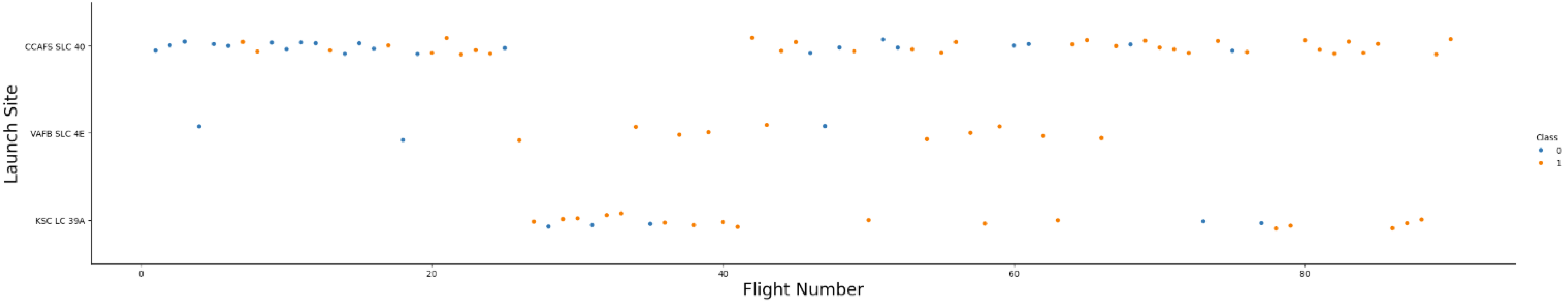
Flight Number more than 40 were mostly successful at Launch Site CCAFS SLC 40.

Flight Number range 25 to 65 were mostly successful at Launch site VAFB SLC 4E.

Flight Number more than 25 was mostly successful at location KSC LC 39A.



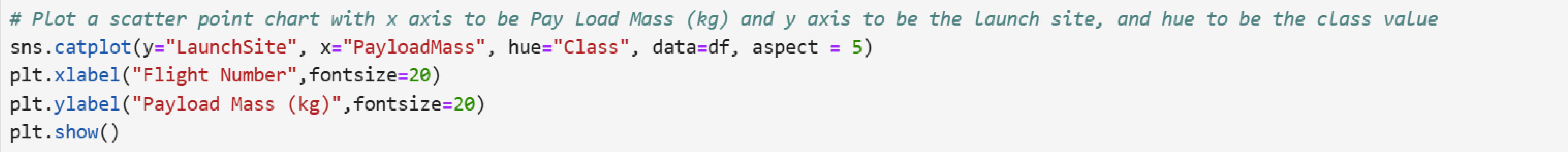
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



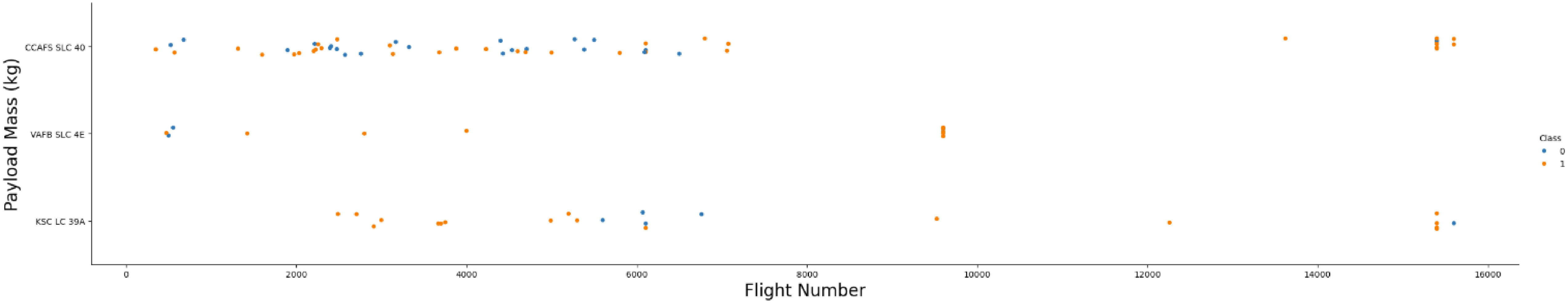
Payload vs. Launch Site

No rockets were launched at VAFB-SLC where the payload mass was greater than 10000kg.

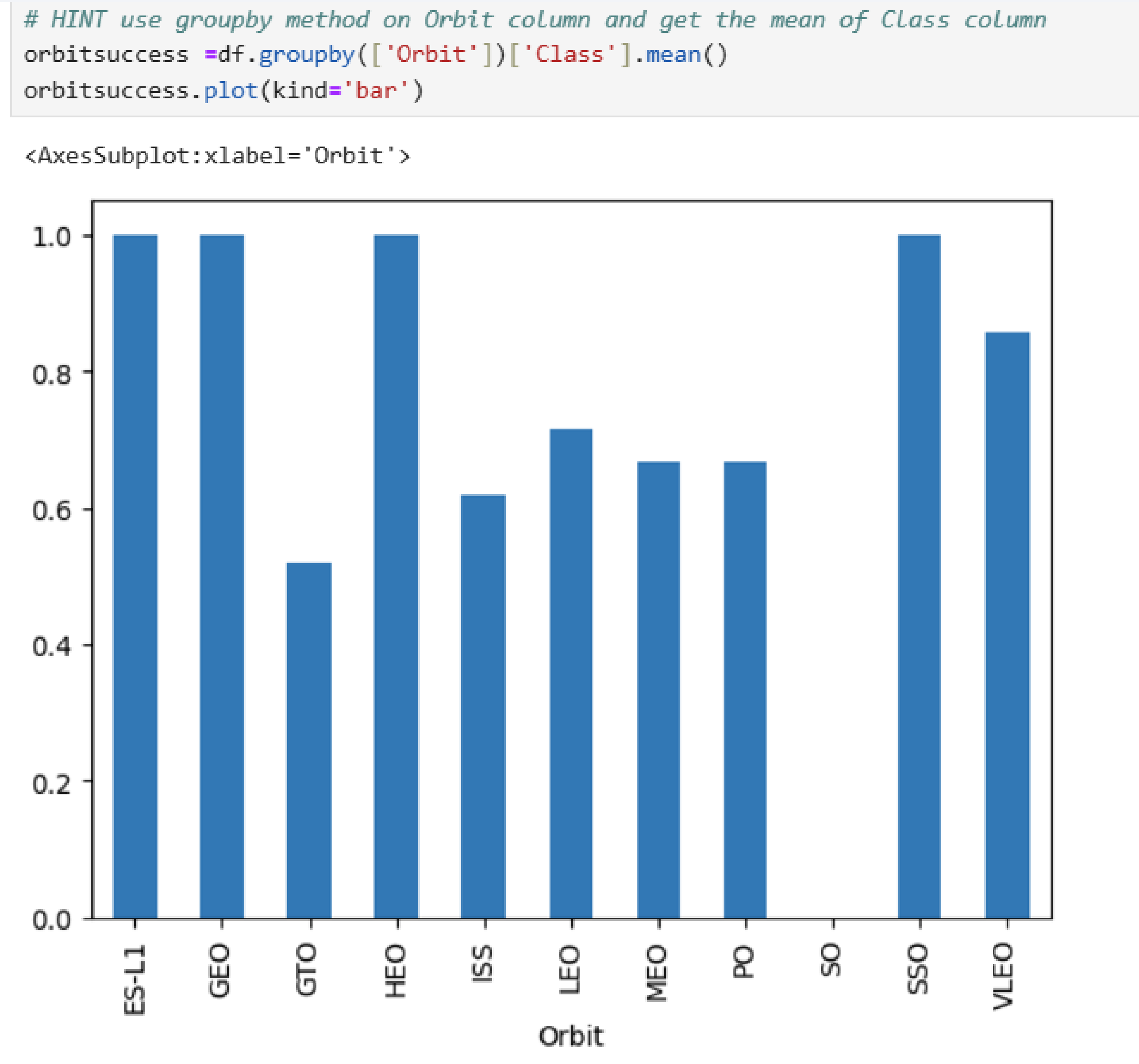
When the payload mass is greater than10000, particularly closer to 16000, it is more likely to be successful.



```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Payload Mass (kg)",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type



```
# HINT use groupby method on Orbit column and get the mean of Class column
orbitsuccess =df.groupby(['Orbit'])['Class'].mean()
orbitsuccess.plot(kind='bar')
```

```
<AxesSubplot:xlabel='Orbit'>
```

The most successful orbit types are ES-L1, GEO, HEO and SSO, with a success rate of 1, meaning they are always successful.

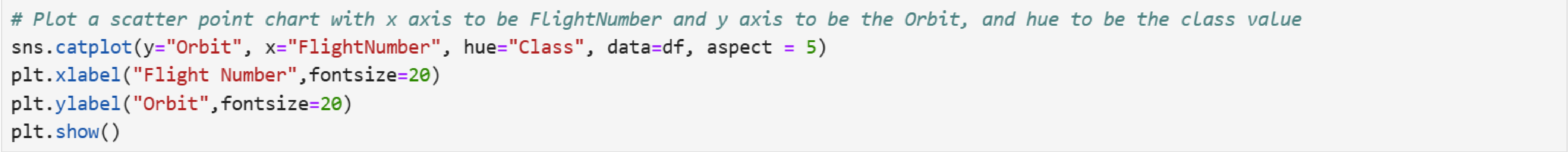
The least successful orbit is GTO with 0.5, meaning it is only successful half the time.

No launches were performed in orbit type SO.

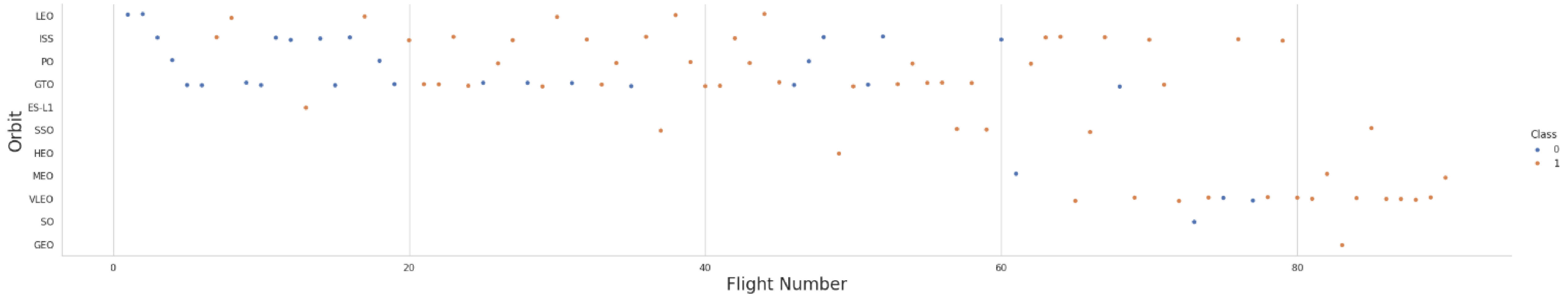
Flight Number vs. Orbit Type

A correlation exists between flight number and orbit type when looking at LEO, however, no other orbit types exhibit similar findings.

LEO, ISS, PO and GTO have majority of their data spread between flight number 0 and 60.



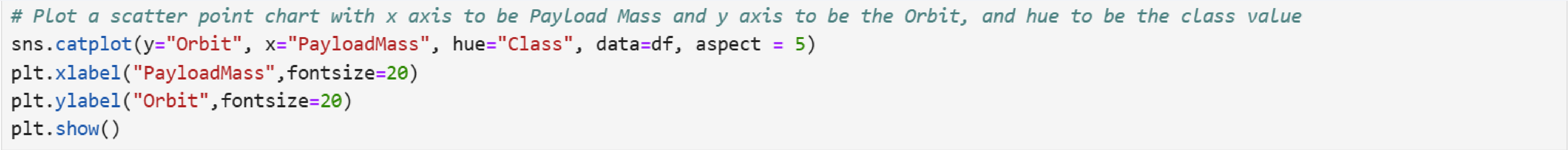
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



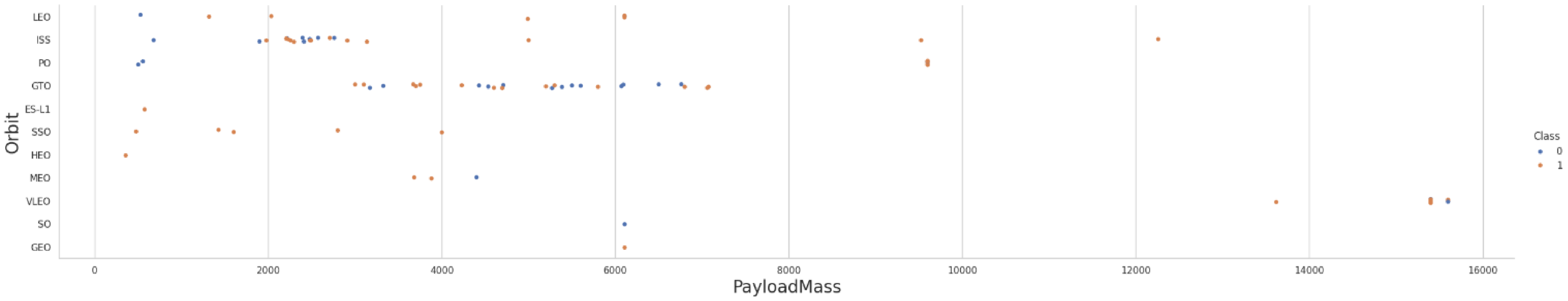
Payload vs. Orbit Type

Orbit type VLEO that has high success rate also has heavy payload.

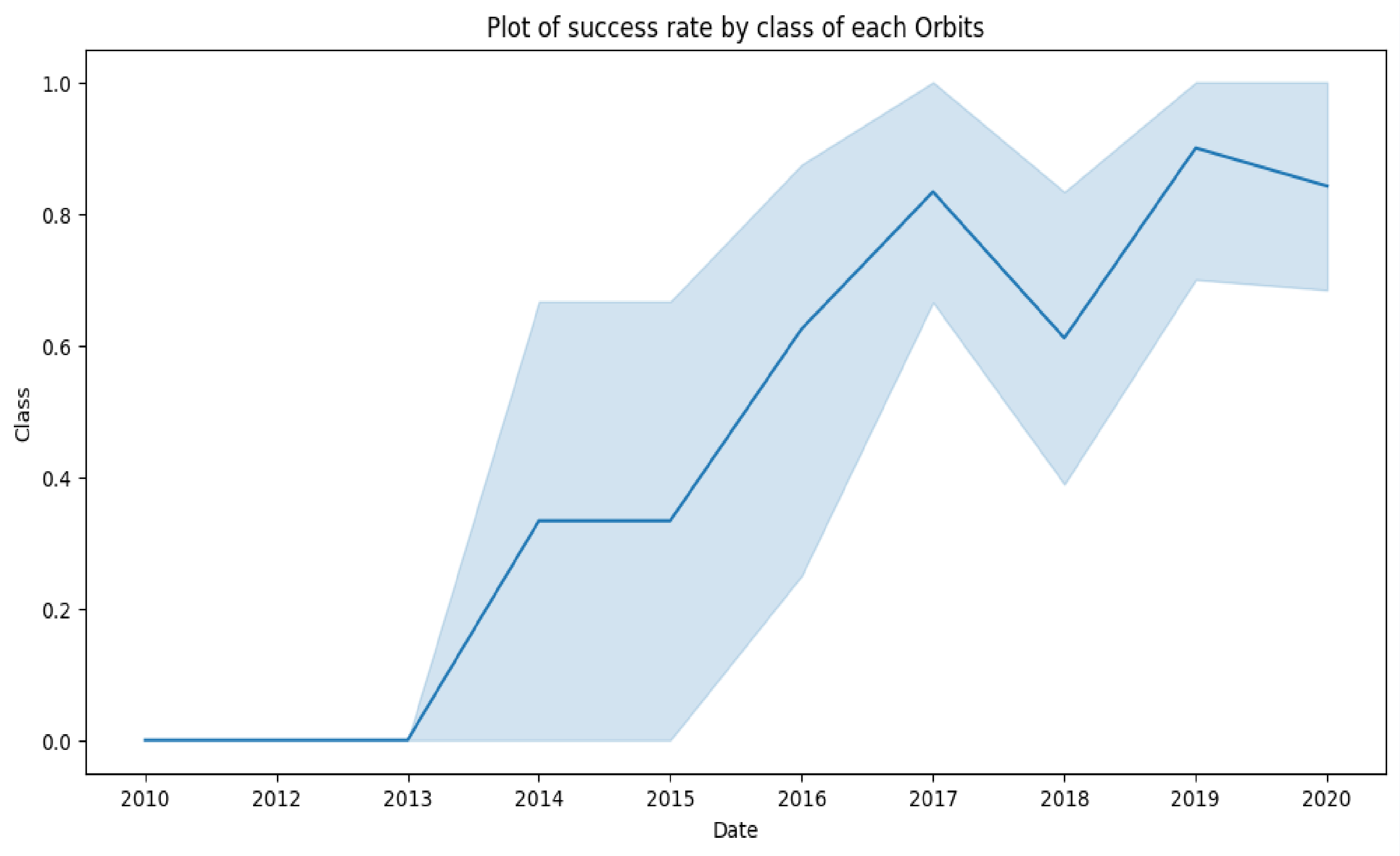
There is a possibility that the heavier the payload, the higher the probability of success.



```
# Plot a scatter point chart with x axis to be Payload Mass and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Launch Success Yearly Trend



2019 was the most successful year.

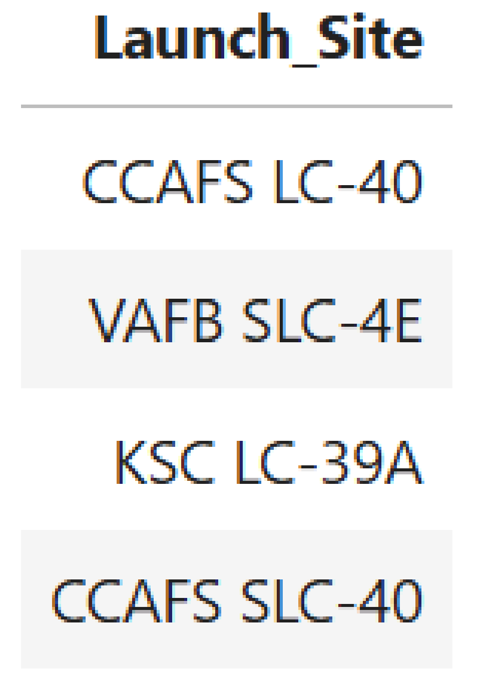
There has been a steady increase in success rate from 2013 to 2020.

2010 through to 2013 were the least successful years.

All Launch Site Names

Using the DISTINCT query, we identified the unique launch sites from the data within the table.

```
%sql select DISTINCT ("LAUNCH_SITE") from SPACEXTBL;
```

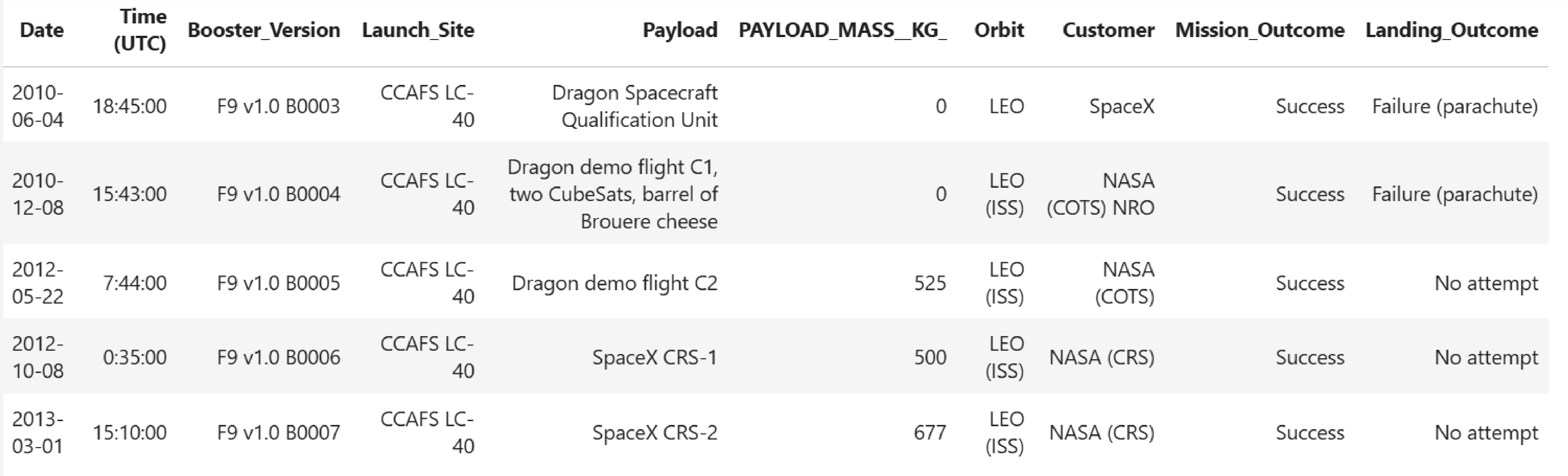


Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40

Launch Site Names Begin with 'CCA'

Displaying 5 records where the launch sites begins with `CCA

%sql SELECT * from SPACEXTBL where (LAUNCH_SITE) LIKE 'CCA%' LIMIT 5;

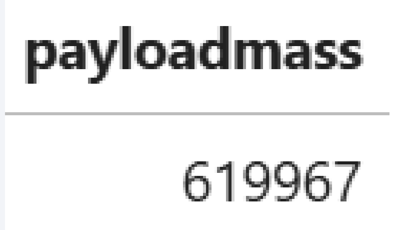


Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight 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

The sum of the total payload masses carried by all the boosters across all launches by NASA, in kilograms.

```
%sql select sum(PAYLOAD_MASS__KG_) as payloadmass from SPACEXTBL;
```

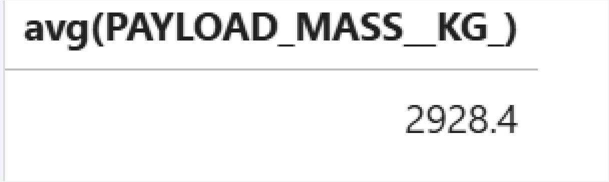


payloadmass
619967

Average Payload Mass by F9 v1.1

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

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where
Booster_Version = ‘F9 v1.1’
```

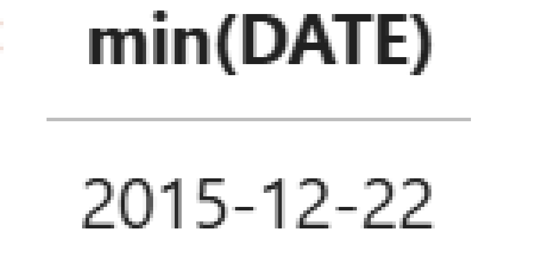


avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

Using the minimum function, we discovered the first successful landing outcome on a groundpad Find the dates of the first successful landing outcome on ground pad.

```
%sql select min(DATE) from SPACEXTBL where (Landing_Outcome) like 'Success (ground pad)’;
```

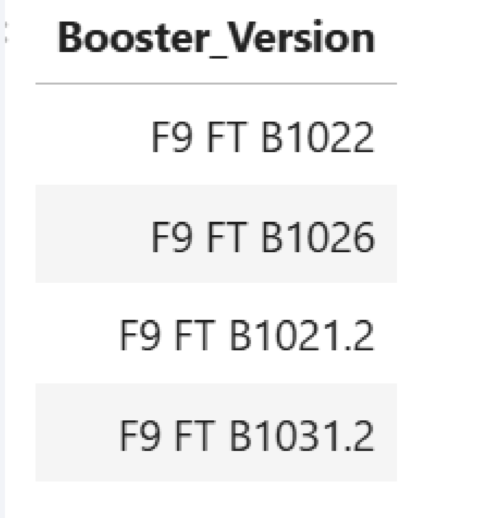


min(DATE)
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

The booster version names that successfully landed on drone ships and had payload mass greater than 4000 but less than 6000kg

```
%sql select BOOSTER_VERSION from SPACEXTBL where
Landing_Outcome='Success (drone ship)' and PAYLOAD_MASS__KG_ BETWEEN
4000 and 6000;
```

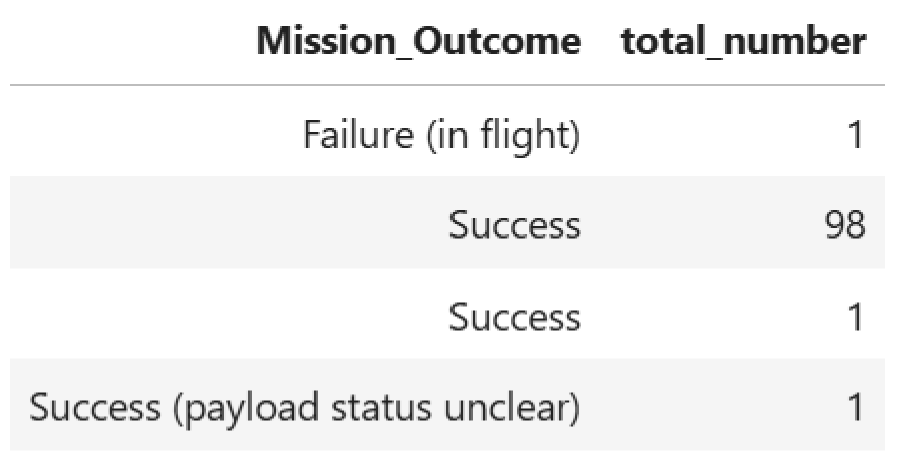


Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Total Number of Successful and Failure Mission Outcomes

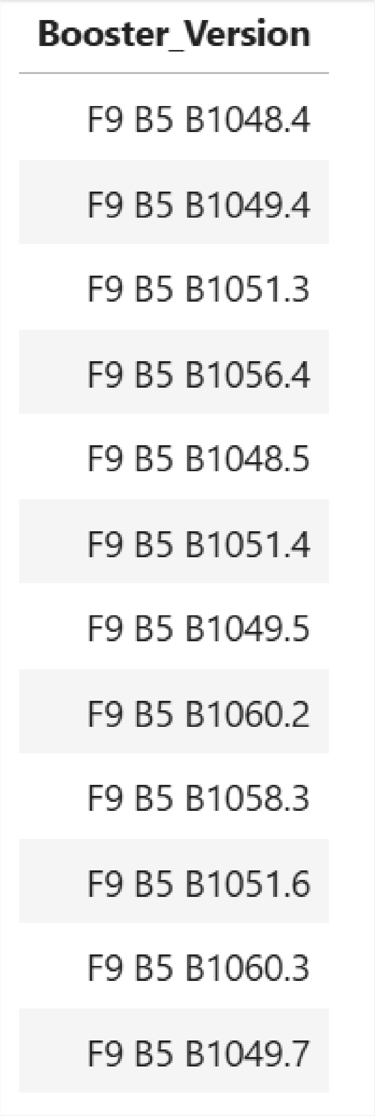
The total number of successful and failure mission outcomes, grouped by mission outcome.

```
%sql SELECT MISSION_OUTCOME, COUNT(*) 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



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

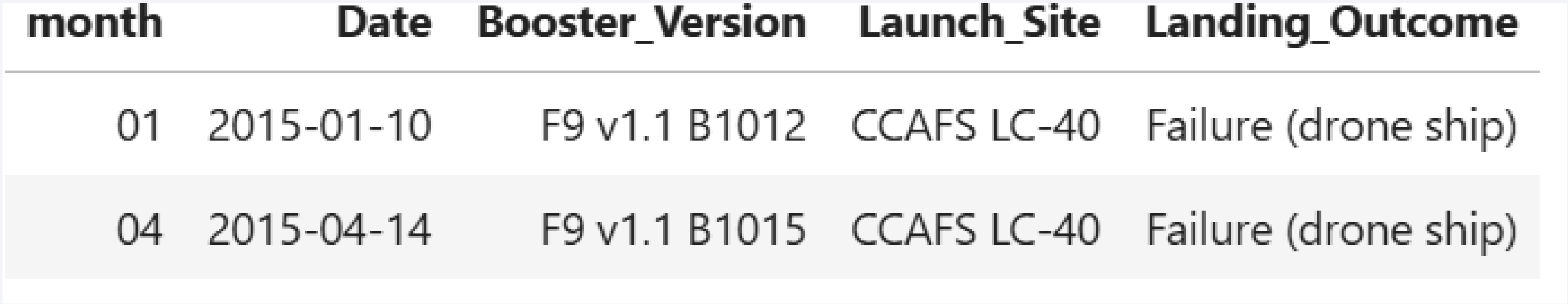
The booster names which have carried the maximum payload mass

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

2015 Launch Records

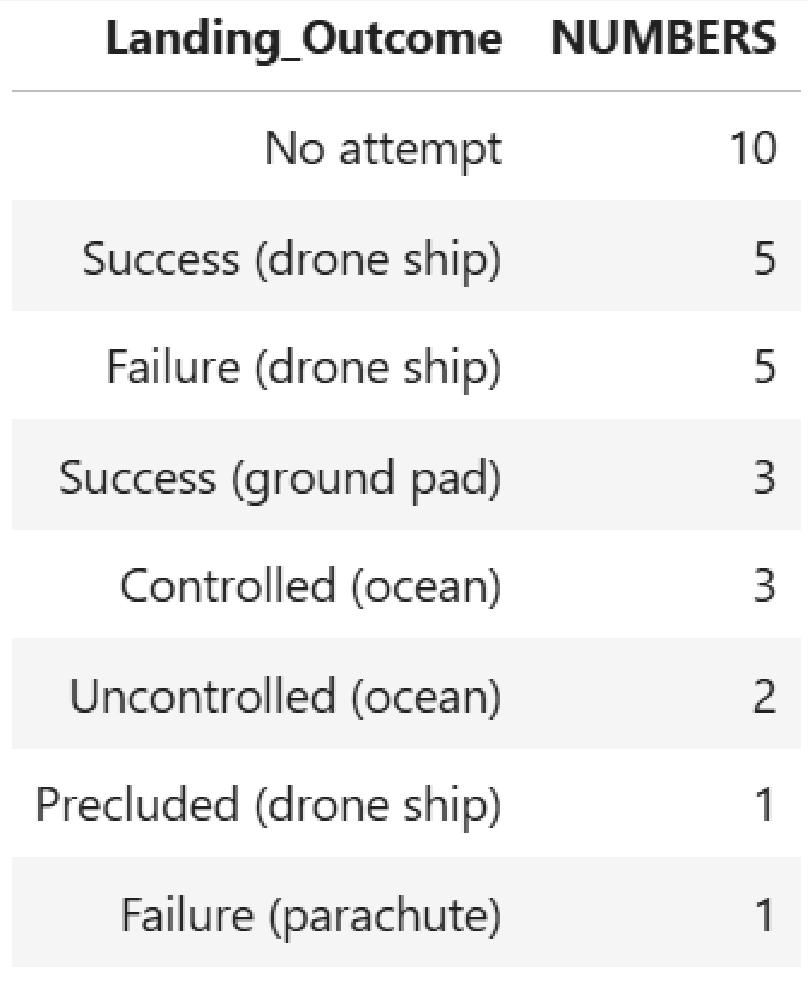
The failed landing outcomes in drone ship, their booster versions, month, and launch site names for in year 2015.

```
%sql SELECT substr(Date, 6,2) as month, DATE,BOOSTER_VERSION,
LAUNCH_SITE, Landing_Outcome \FROM SPACEXTBL \where Landing_Outcome
= 'Failure (drone ship)' and substr(Date, 0,5)='2015';
```



month	Date	Booster_Version	Launch_Site	Landing_Outcome
01	2015-01-10	F9 v1.1 B1012	CCAFS LC-40	Failure (drone ship)
04	2015-04-14	F9 v1.1 B1015	CCAFS LC-40	Failure (drone ship)

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



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

The count of landing outcomes between the date 2010-06-04 and 2017-03-20, ranked in descending order

```
%sql SELECT landing outcome, COUNT(*) AS NUMBERS FROM SPACEXTBL WHERE DATE>'2010-06-04' AND  DATE < '2017-03-20' GROUP BY landing outcome ORDER BY NUMBERS DESC;
```

Section 3

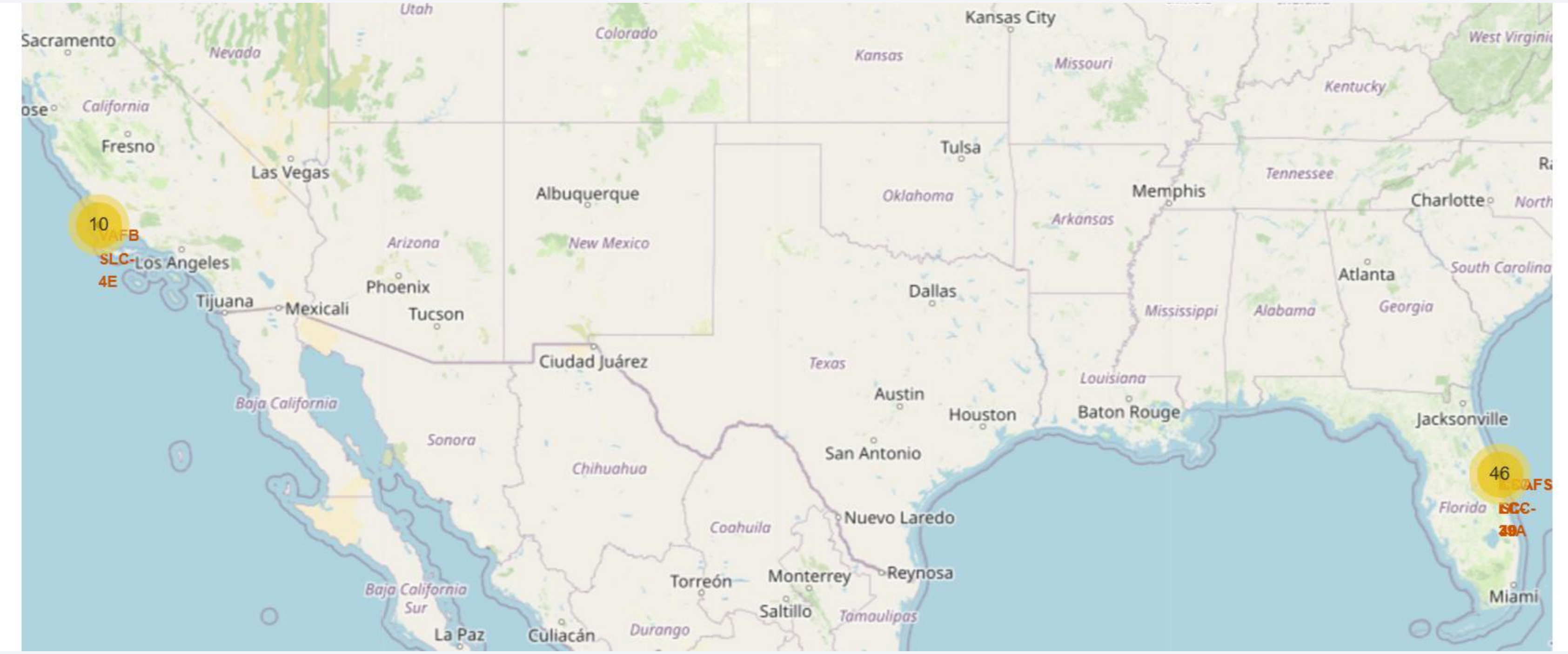
Launch Sites Proximities Analysis

A map showing all launch sites

Launch sites are spread across 3 larger areas.

We can see most of the launch sites are on the east coast.

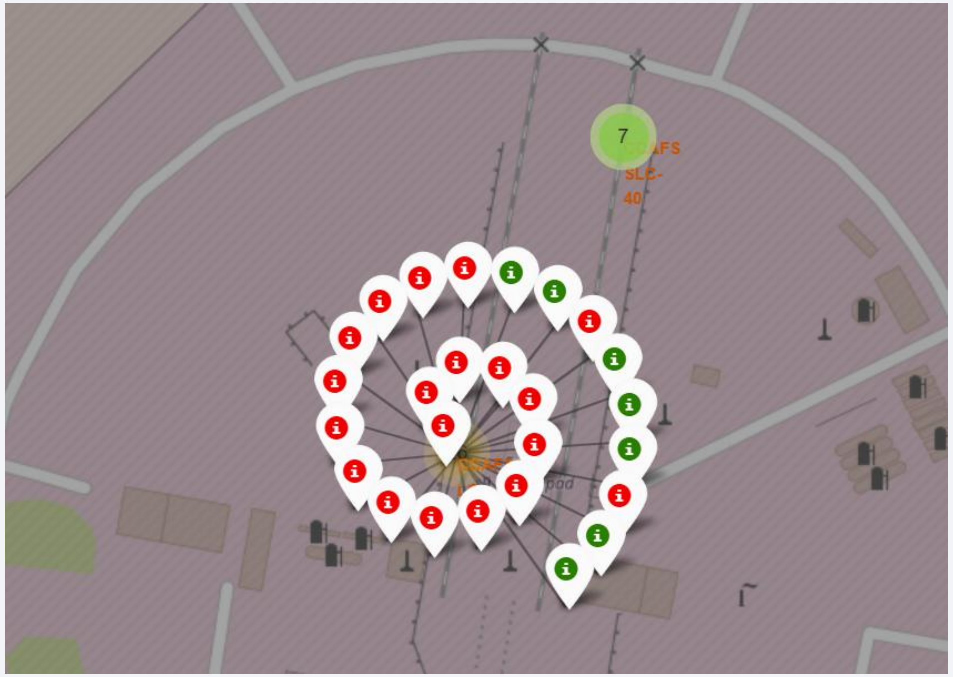
Launch Sites are always located close to a coastline.

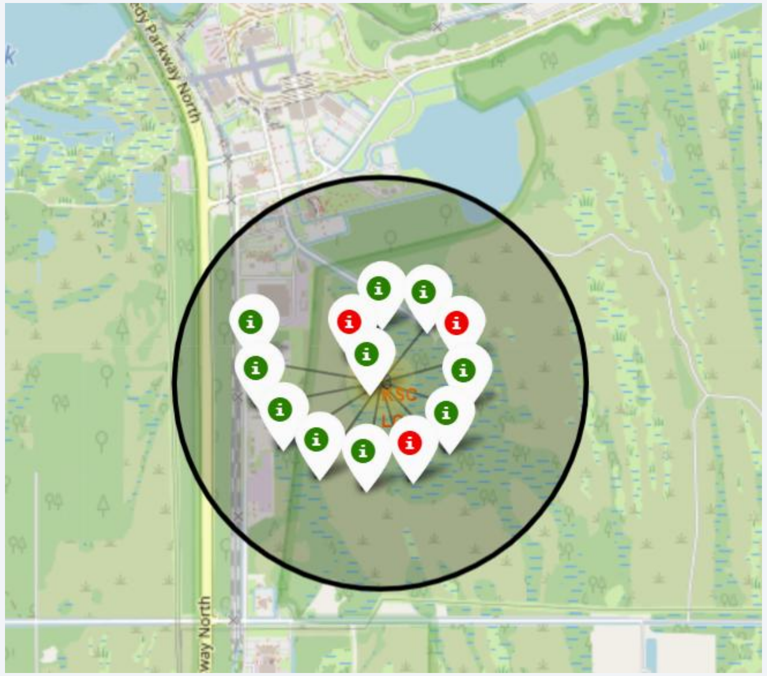


Launch Site success cluster

A cluster can help us identify the success of a launch site. As we can see from these launch sites, the first has a relatively low success rate, as red markers represent a failure and green markers represent successes.

The second site is the most successful site (KSC LC-39A), represented by more green markers.





Folium maps: KSC LC-39A to its proximities

We can see that KSC LC-39A is 7.73km away from a coastline, 16.43km away from a city, 0.84km away from a highway and 0.69km away from a railway.

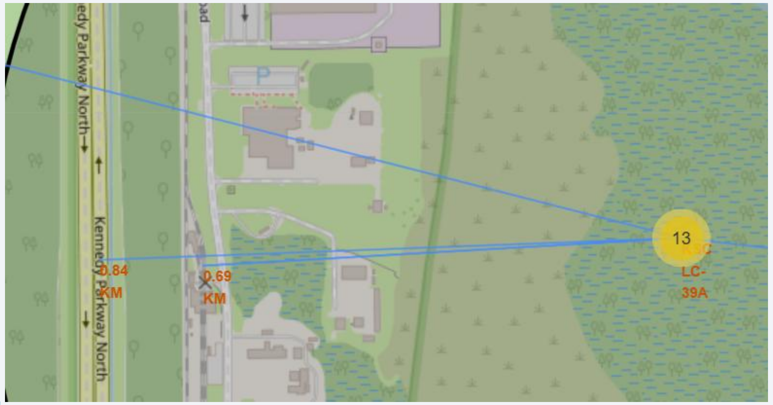
This is the most successful launch site and is furthest way from a coastline at 7.73km.

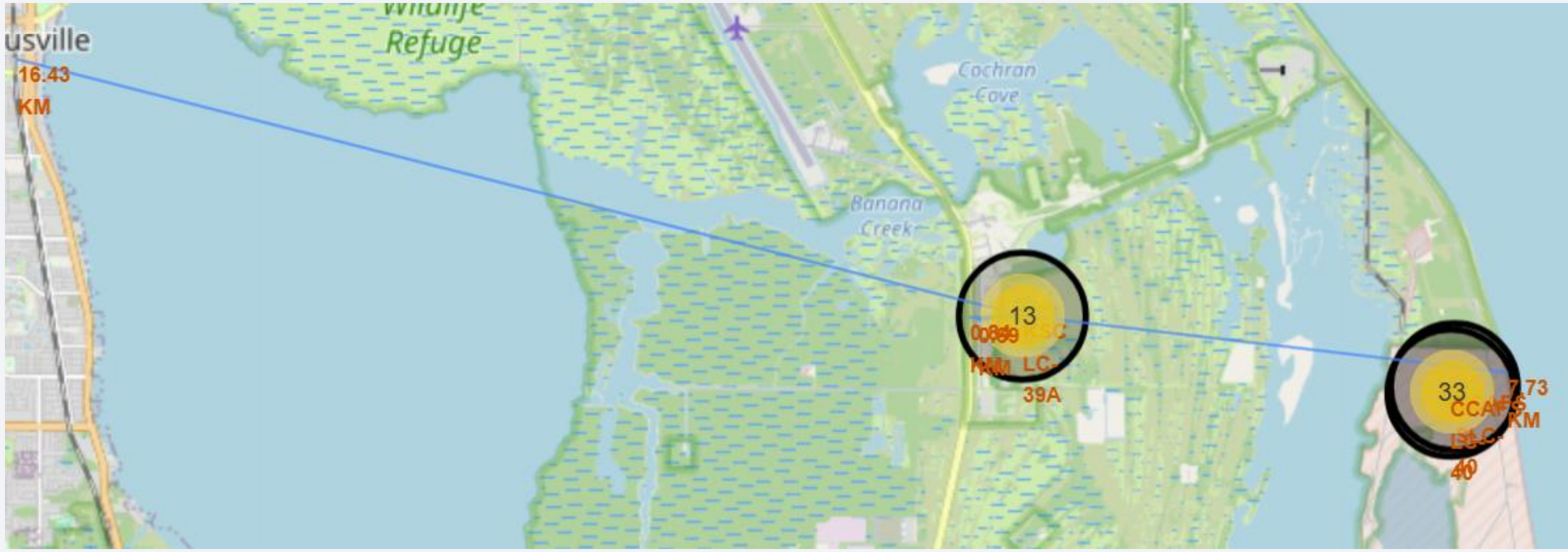
A relationship between highway and sites does not seem to exist.

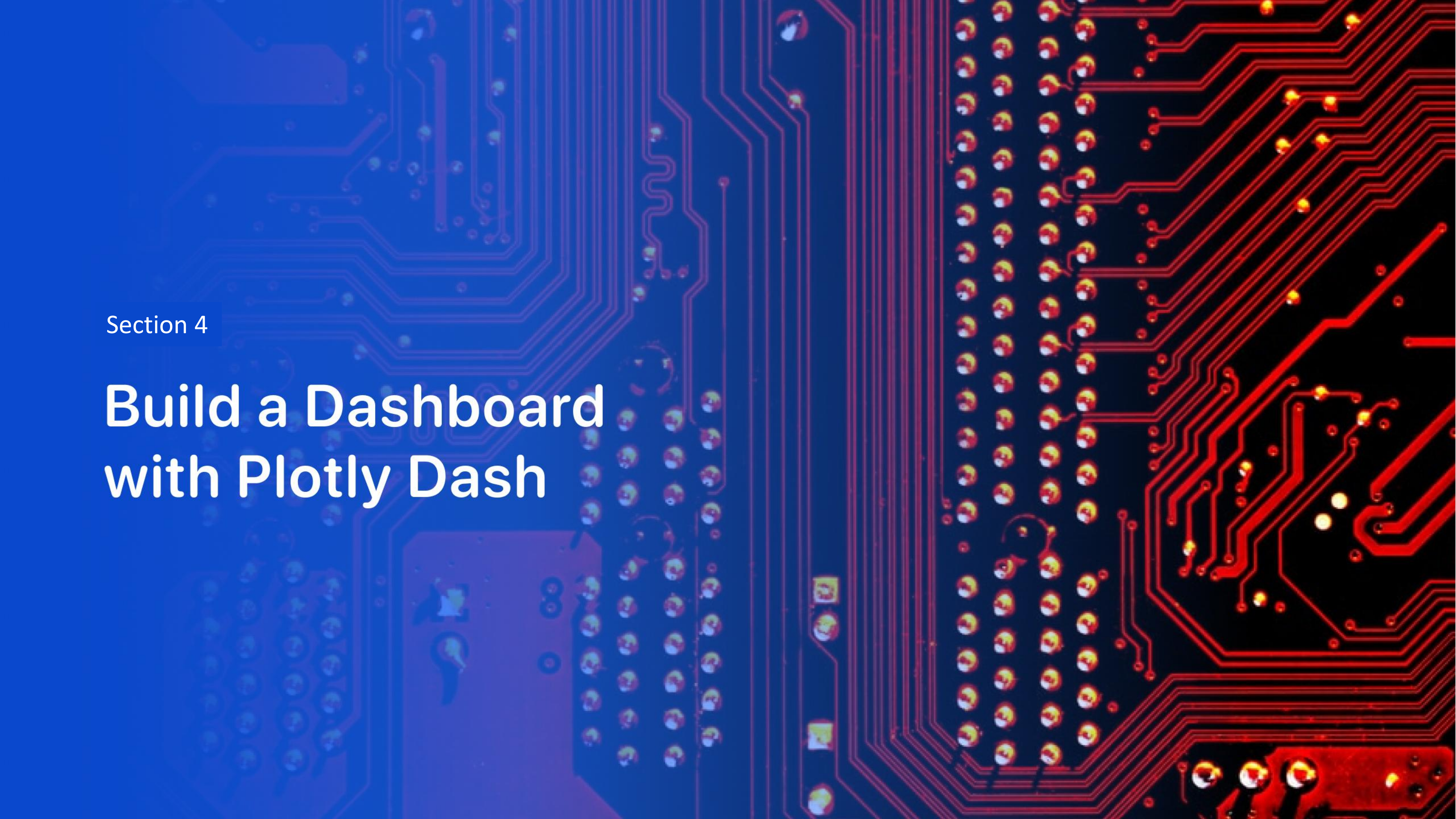
The closest launch site to a city is VAFB SLC-4E to Lompoc at 14km.

All launch sites are within a kilometer of a railway line.

It would be safe to assume that a railway line and coastline are extremely important in locations for launch sites.





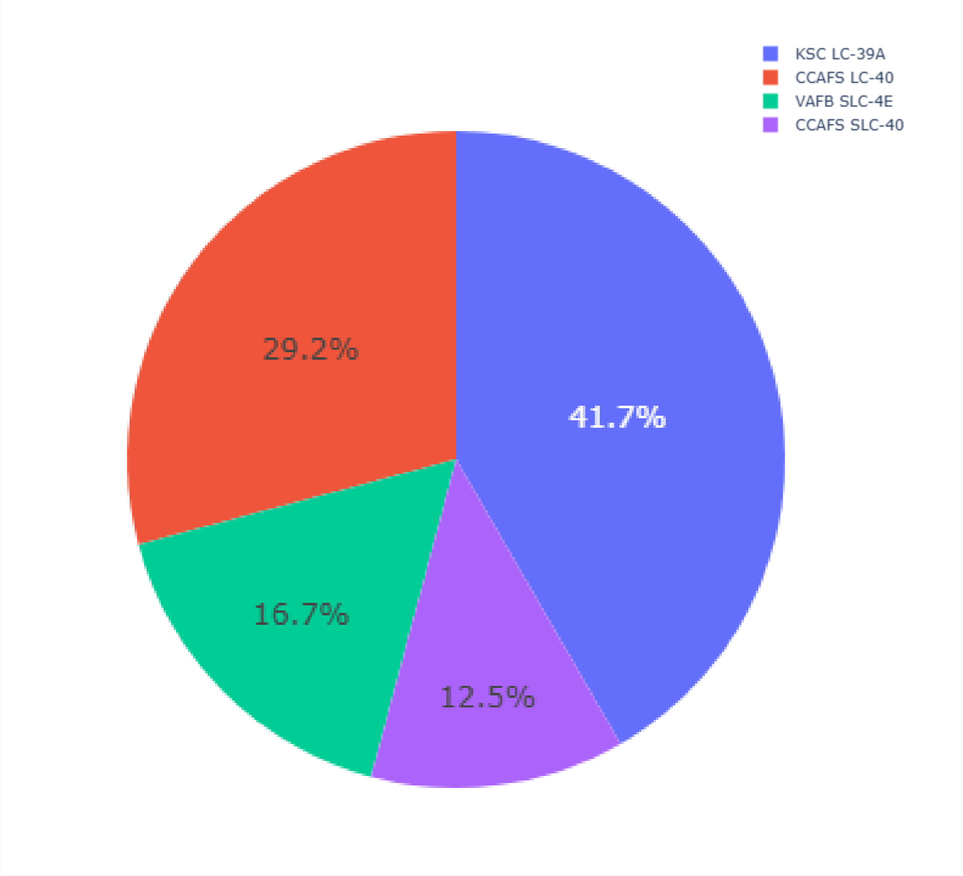


Section 4

Build a Dashboard with Plotly Dash

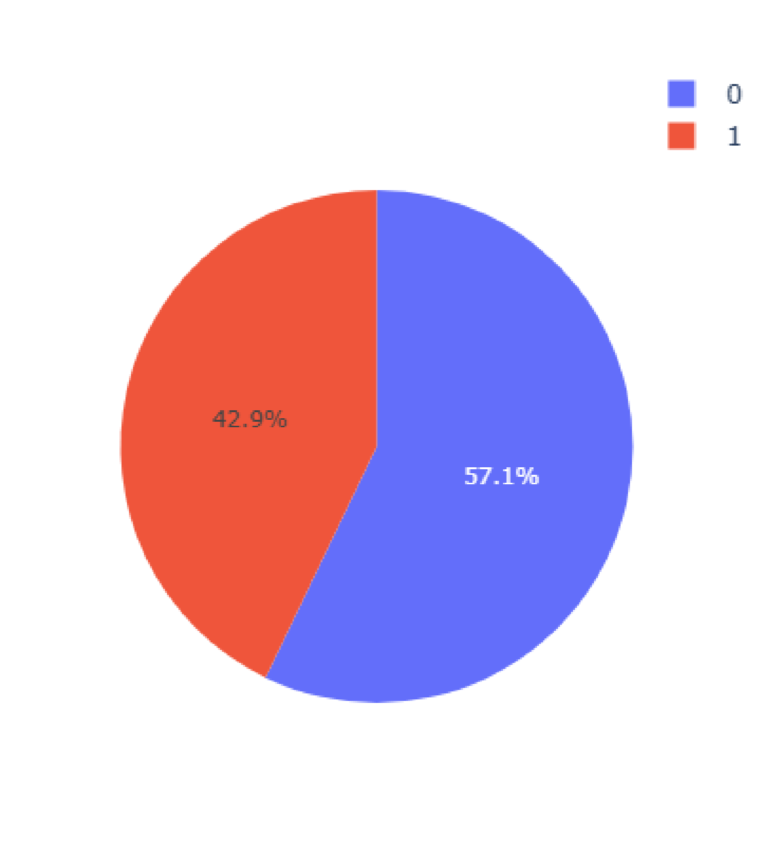
The Success Count for all the launch sites

KSC LC-39A is the most successful launch site, whereas CCAFS SLC-40 is the least successful.



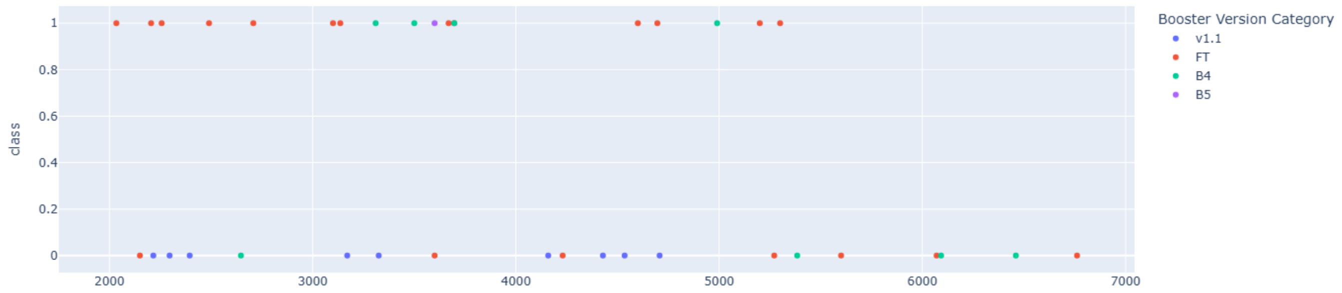
Success ratio for site CCAFS SLC-40

Although CCFS SLC-40 is the least successful launch site overall, it has the greatest success ratio for attempted launches.

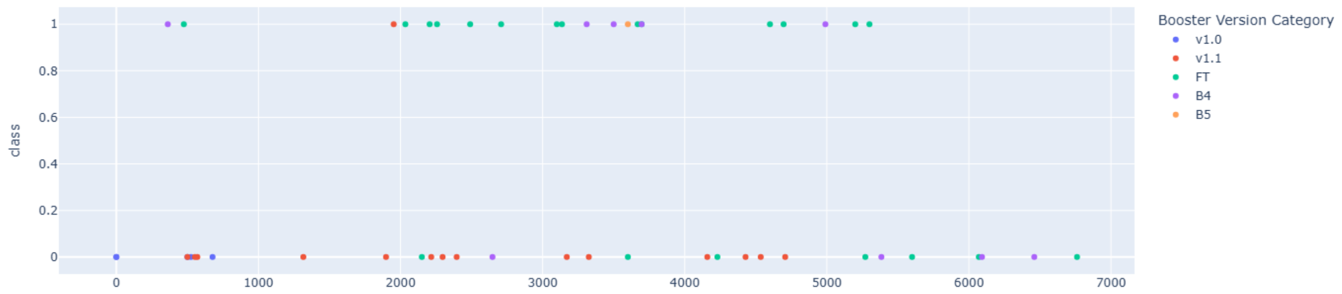


Varying payload ranges versus outcome success

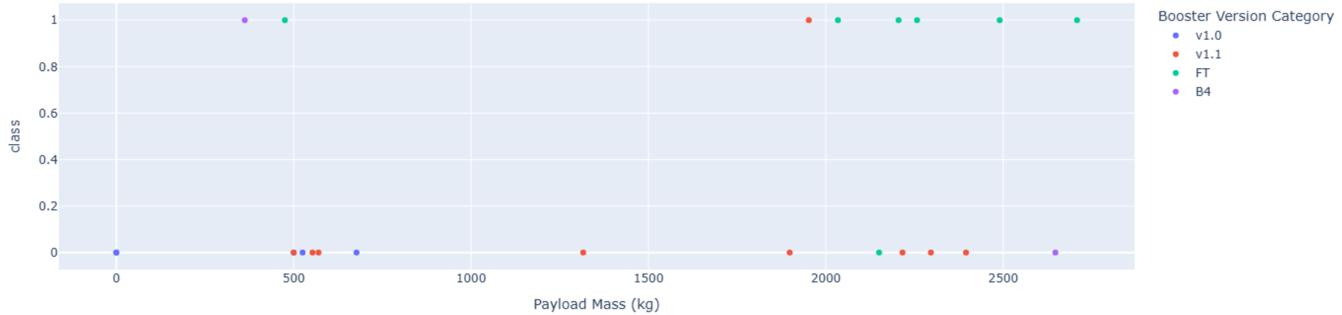
Success count on Payload mass for all sites



Success count on Payload mass for all sites



Success count on Payload mass for all sites



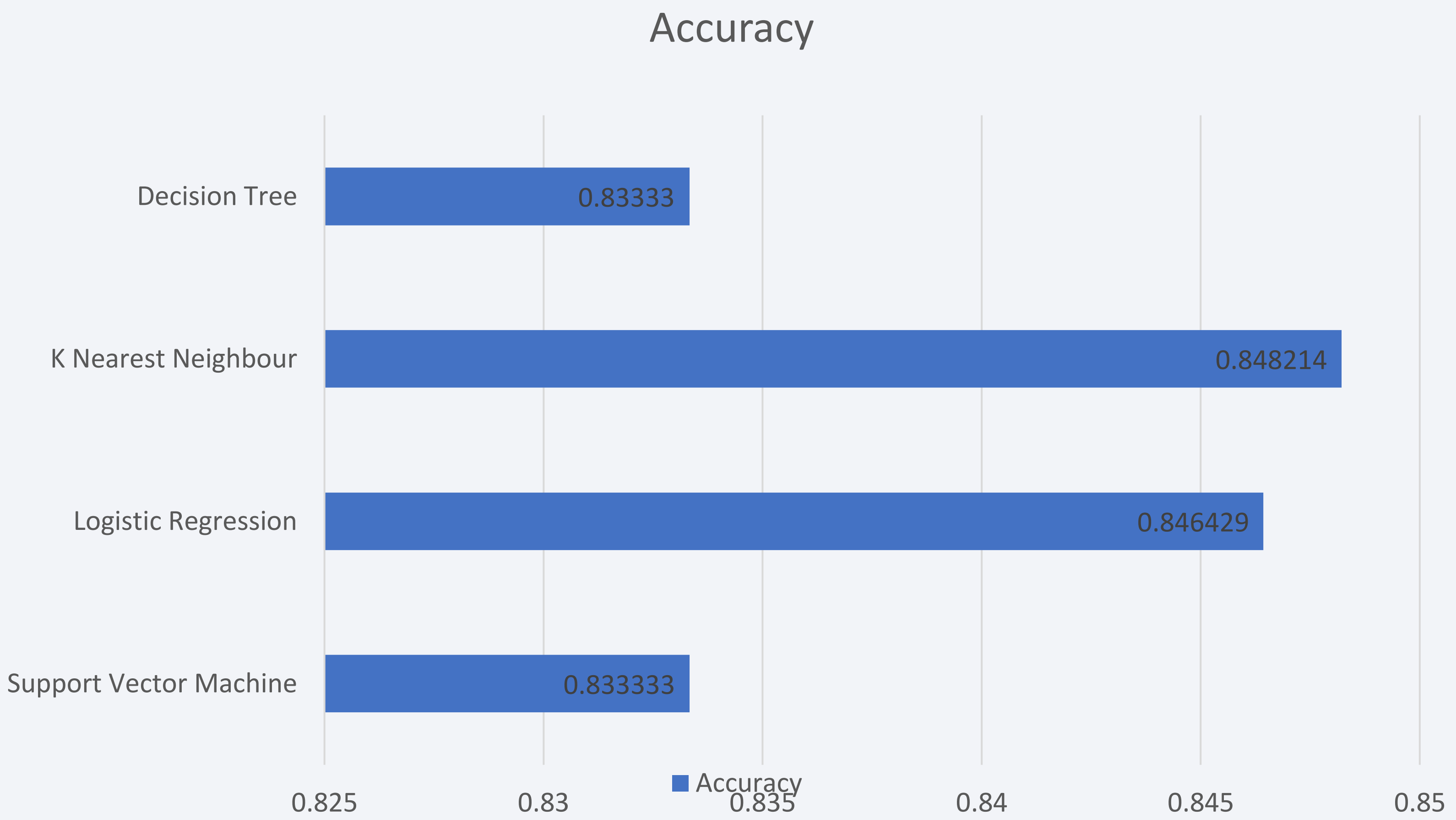
Payload range between 2k and 7k has the largest success rates with Booster Version FT being most successful

Section 5

Predictive Analysis (Classification)

Classification Accuracy

K Nearest Neighbour with cv=10 performed the best, but only marginally.



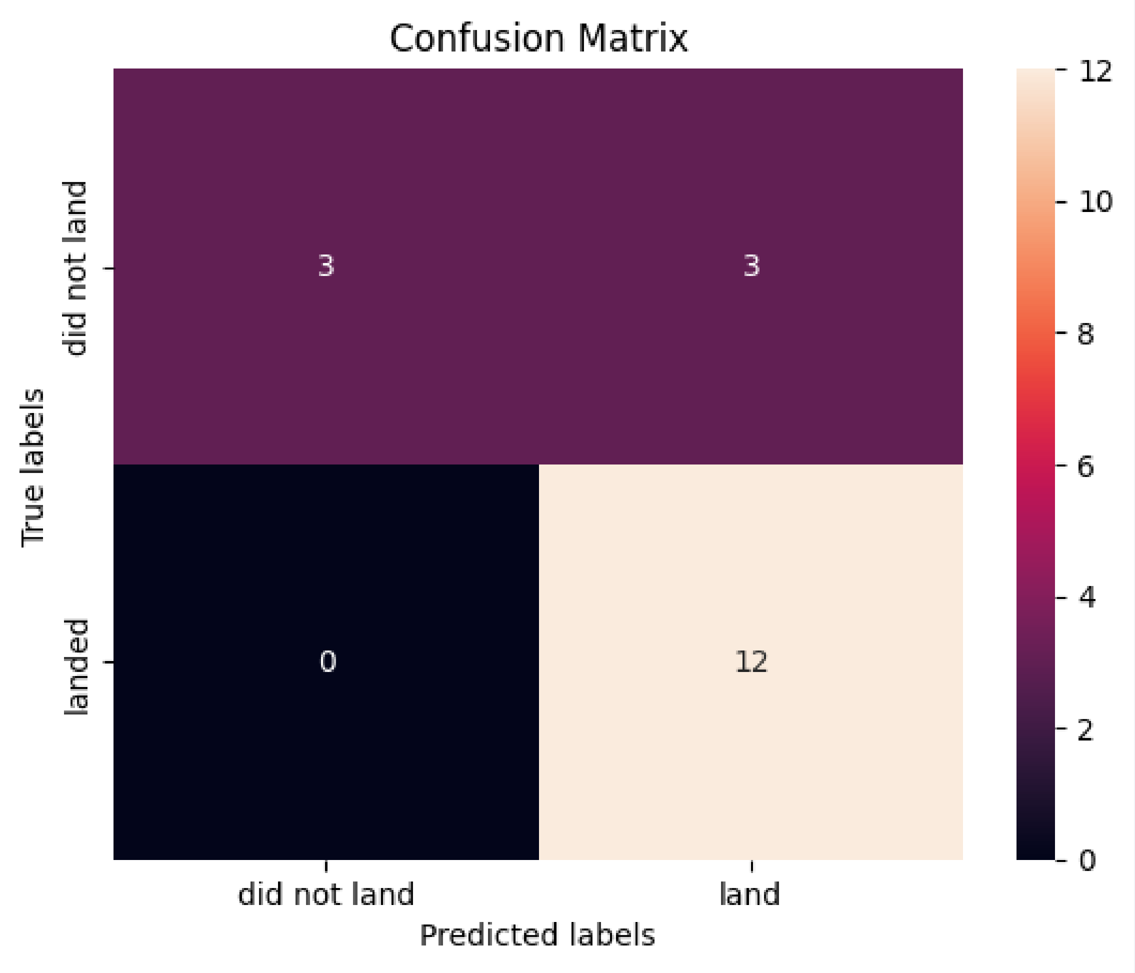
Confusion Matrix

All models produced identical confusion matrixes.

From the matrix we can see that there is a problem with false positives.

True Positive - 12 (True label is landed, Predicted label is also landed)

False Positive - 3 (True label is not landed, Predicted label is landed)



Conclusions

- Logistic regression, Support vector machine, Decision trees and K nearest neighbour was used to find the best method using a GridSearchCV.
- K nearest Neighbours has the highest accuracy using out train test split and a cv=10.
- All Confusion matrixes were identical and had the tendency to over predict false positives.

Appendix

- In order to draw accurate conclusions based on launch site location, additional calculations had to be made using alternative launch sites, and not only the most successful.
- Testing the dash skeleton led to a frozen dashboard, I recommend not running the dashboard until all code has been entered.

Thank you!