Space X Falcon 9 Landing Analysis

Sayali Deshpande
March 20,2024

Outline

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

Executive Summary

- Methodologies Summary:
  - Data Collection
  - Data Wrangling
  - Exploratory Data Analysis with Data Visualisation
  - Exploratory Data Analysis with SQL
  - Interactive Visual Analysis
  - Predictive Analysis (Classification)

- Results Summary:
  - Exploratory Data Analysis Results
  - Geospatial Analysis
  - Interactive Dashboard
  - Predictive Analysis with Classification model

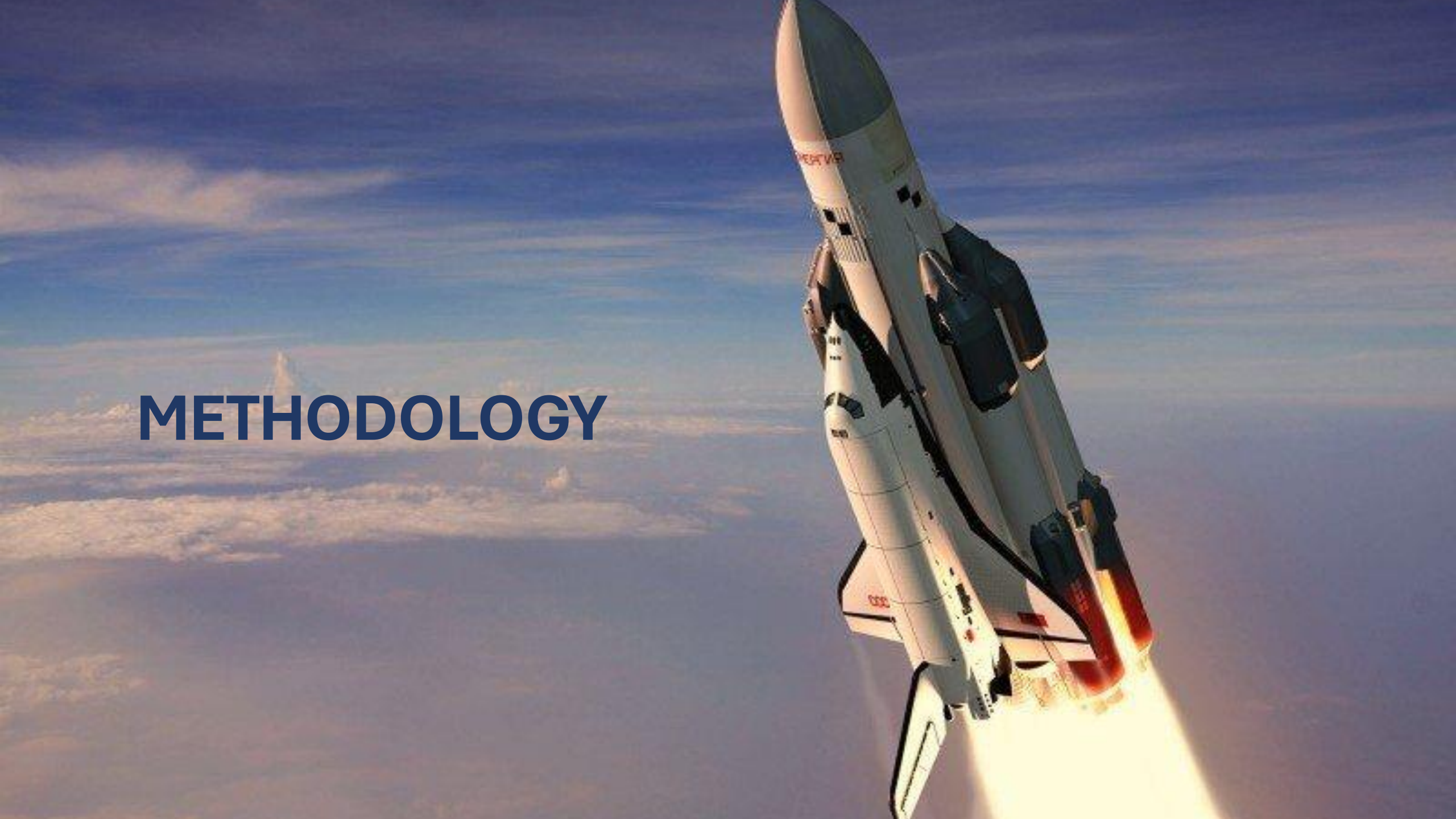
Introduction

- Falcon 9 rocket launch costs for upward of 165 million dollars offered by some providers in the market whereas Space X advertised launch of Falcon 9 at 62 million dollar price. This is considerably cheaper solution. The shift in price is because Space X can reuse the first stage.

- We have to draw predictions on successful landing of first stage to determine the cost of one launch. It will also help in case any other company would want to bid against Space X for the rocket launch.

- This exercise will predict Space X's Falcon 9 first stage successful landing.

METHODOLOGY



Methodology Summary:

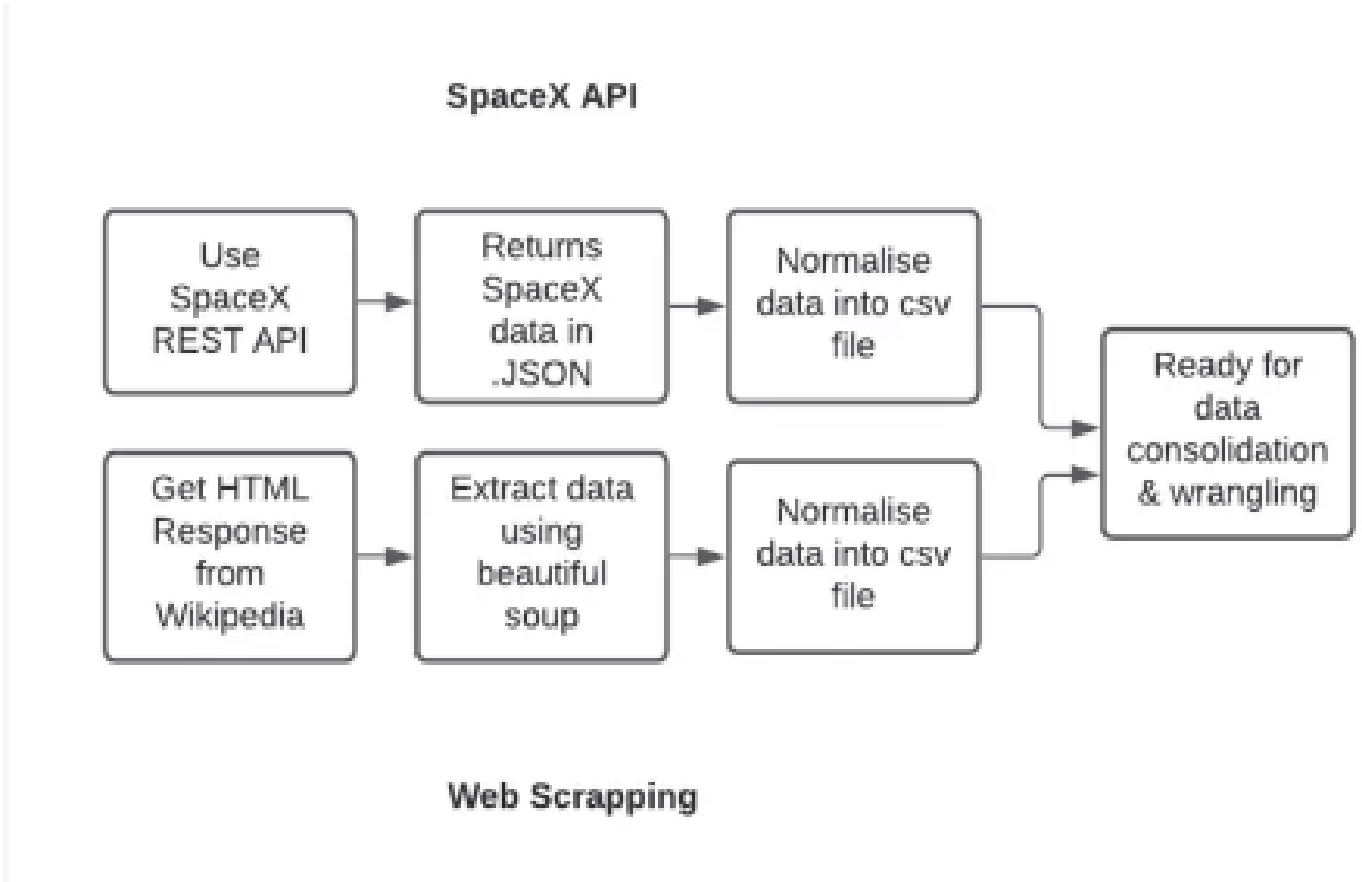
- Data Collection:
  - Using SpaceX REST API
  - Web scraping
- Data Wrangling:
  - *.fillna():* To remove Null/Undefined values
  - *.value_counts():* To determine the following
    I. Number of launches on each site
    II. Number and occurrence of each orbit
    III. Number and occurrence of mission outcome per orbit type
  - Adding a landing outcome label
- Exploratory Data Analysis:
  - Using *SQL* for data manipulation on SpaceX dataset
  - Visual representation to identify pattern and relationship between variables using *Pandas* and *Matplotlib*

Contd. Methodology Summary..

- Interactive Visual Analytics:
  - Geospatial analytics using Folium
  - Interactive dashboard with Plotly Dash
- Data Modelling and Evaluation:
  - Determine training labels, Hyperparameter for SVM, Classification Trees and Logistic Regression
  - Plot confusion matrix for each classification model
  - Assess accuracy of each classification model

Data Collection

- For this exercise, the data has been collected from two sources:
  1. SpaceX API
  2. Web Scraping



Data Collection- **SpaceX API**

SpaceX has provided a Rest API which contains all the data about launches, information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.

Github URL: https://github.com/capstone-spacex-data-collection-api.ipynb

Steps:

- Make a GET request to SpaceX API
- Store the data into a pandas dataframe object.
- Further filter the data to get only Falcon 9 entries, remove null values and replace missing data.

1. SpaceX request

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

2. Create a dataframe dictionary for Falcon 9 dataset

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

df = pd.DataFrame.from_dict(launch_dict)

data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

3. Clean the dataframe

```
data_falcon9.isnull().sum()
falcon9_mean = data_falcon9['PayloadMass'].mean()
data_falcon9 = data_falcon9.fillna(
    value={'PayloadMass': data_falcon9['PayloadMass'].mean()})
```

Data Collection- **Web Scraping**

For this, we used a Wikipedia page- "List of Falcon 9 and Falcon Heavy launches" to collect Falcon 9 historical launch records.

Source: *https://en.wikipedia.org/wiki/List_of_Falcon_9_and_Falcon_Heavy_launches*

Github URL: https://github.com/capstone-webscraping.ipynb

Steps:

- Make a get request to Wikipedia URL
- Store the html page into a BeautifulSoup object
- Parse the BeautifulSoup html table with all the required columns.
- Convert the parsed BeautifulSoup object into a Pandas Dataframe.

1. Request to Wikipedia page

```
response = requests.get(
    "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1827686922"
)
data = response.text
```

2. Load data into BeautifulSoup object and filter Falcon 9 dataset

```
soup = BeautifulSoup(data)
html_tables = soup.find_all('table')
first_launch_table = html_tables[2]
```

3. Convert parsed data into Pandas dataframe object

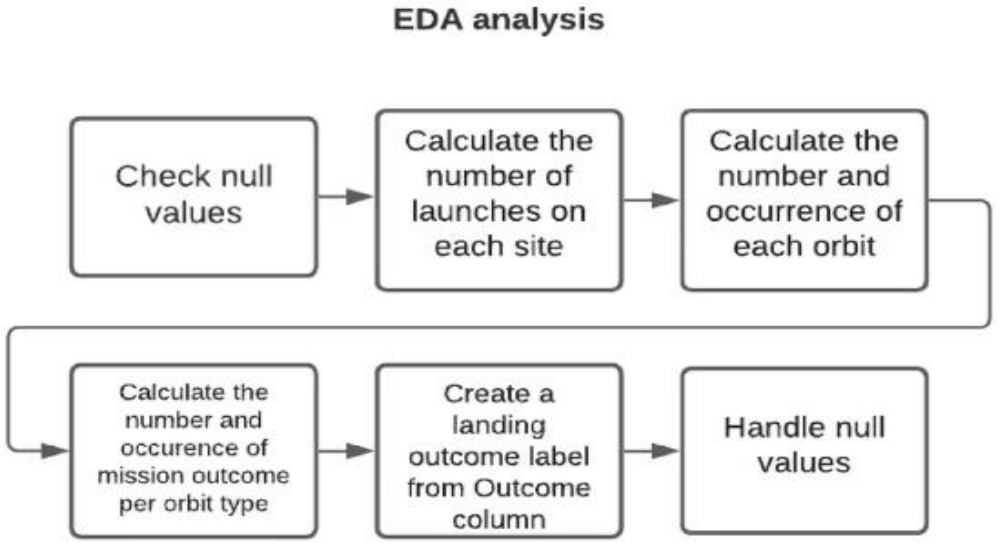
```
column_names = []
th_elements = first_launch_table.find_all('th')
for row in th_elements:
    name = extract_column_from_header(row)
    if(name != None and len(name)>0):
        column_names.append(name)
launch_dict= dict.fromkeys(column_names)
# Remove an irrelevant column
del launch_dict['Date and time ( )']
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

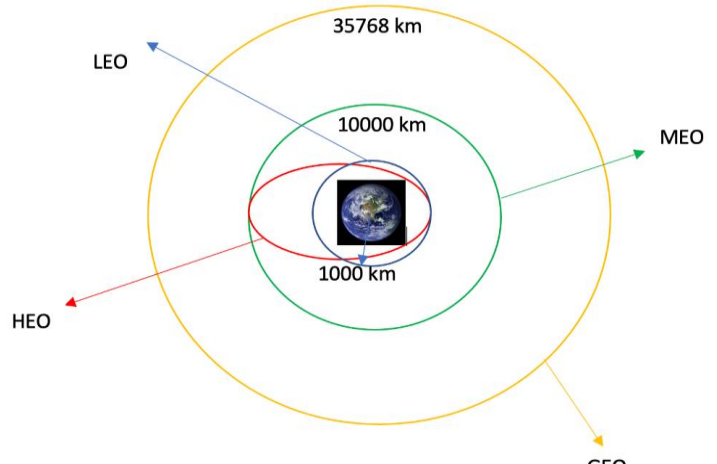
Data Wrangling

The SpaceX dataset contains several Space X launch facilities, and each location is in the "*LaunchSite*" column.
Each launch aims to a dedicated orbit, and some of the common orbit types are shown in the figure below. The orbit type is in the Orbit column.

Github URL: https://github.com/capstone-spacex-Data-wrangling.ipynb

1. Initial Data Exploration: Using the .value_counts() method to determine the following:
   - Number of launches on each site
   - Number and occurrence of each orbit
   - Number and occurrence of landing outcome per orbit type





Contd. Data Wrangling…

2. The landing outcome is shown in the Outcome column:
   - 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 to a drone ship
   - False ASDS – the mission outcome was unsuccessfully landed to a drone ship.
   - None ASDS and None None – these represent a failure to land.

3. To determine whether a booster will successfully land, it is best to have a binary column, i.e., where the value is 1 or 0, representing the success of the landing.

This is done by:
   - Defining a set of unsuccessful (bad) outcomes, bad_outcome
   - Creating a list, landing_class, where the element is 0 if the corresponding row in Outcome is in the set bad_outcome, otherwise, it's 1.
   - Create a Class column that contains the values from the list landing_class
   - Export the DataFrame as a .csv file.

Exploratory Data Analysis with Visualization

For the visual EDA, following charts are plotted.

SCATTER CHARTS

Scatter charts were produced to visualize the relationships between: Flight Number and Launch Site Payload and Launch Site Orbit Type and Flight Number Payload and Orbit Type

Scatter charts are useful to observe relationships, or correlations, between two numeric variables.

BAR CHART

A bar chart was produced to visualize the relationship between: Success Rate and Orbit Type

Bar charts are used to compare a numerical value to a categorical variable. Horizontal or vertical bar charts can be used, depending on the size of the data.

LINE CHARTS

Line charts were produced to visualize the relationships between: Success Rate and Year (i.e. the launch success yearly trend)

Line charts contain numerical values on both axes, and are generally used to show the change of a variable over time

Github URL: https://github.com/capstone-eda-dataviz.ipynb

Exploratory Data Analysis with SQL

EDA was done with SQL as well. Below are the queries performed using 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 the average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome on a ground pad was achieved
- List the names of the boosters which had success on a drone ship and a payload mass between 4000 and 6000 kg
- List the total number of successful and failed mission outcomes
- List the names of the booster versions which have carried the maximum payload mass
- List the failed landing outcomes on drone ships, their booster versions, and launch site names for 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

Github URL: https://github.com/capstone-eda-sql.ipynb

Interactive Maps with Folium

The following steps were taken to visualize the launch data on an interactive map:

1. Mark all launch sites on a map
   - Initialise the map using a Folium Map object
   - Add a folium.Circle to add a highlighted circle area with a text label on a specific coordinate and folium.Marker for each launch site on the launch map

2. Mark the success/failed launches for each site on a map
   - As many launches have the same coordinates, it makes sense to cluster them together.
   - Before clustering them, assign a marker colour of successful (class = 1) as green, and failed (class = 0) as red.
   - To put the launches into clusters, for each launch, add a folium.Marker to the MarkerCluster() object.
   - Create an icon as a text label, assigning the icon_color as the marker_colour determined previously.

Contd. Interactive maps...

3. Calculate the distances between a launch site to its proximities
   - To explore the proximities of launch sites, calculations of distances between points can be made using the Lat and Long values.
   - After marking a point using the Lat and Long values, create a folium.Marker object to show the distance.
   - To display the distance line between two points, draw a folium.PolyLine and add this to the map.

Github URL: https://github.com/capstone-interactive-maps-folium.ipynb

Interactive Dashboard with Plotly Dash

The following plots were added to an interactive dashboard for visualisation of the data:

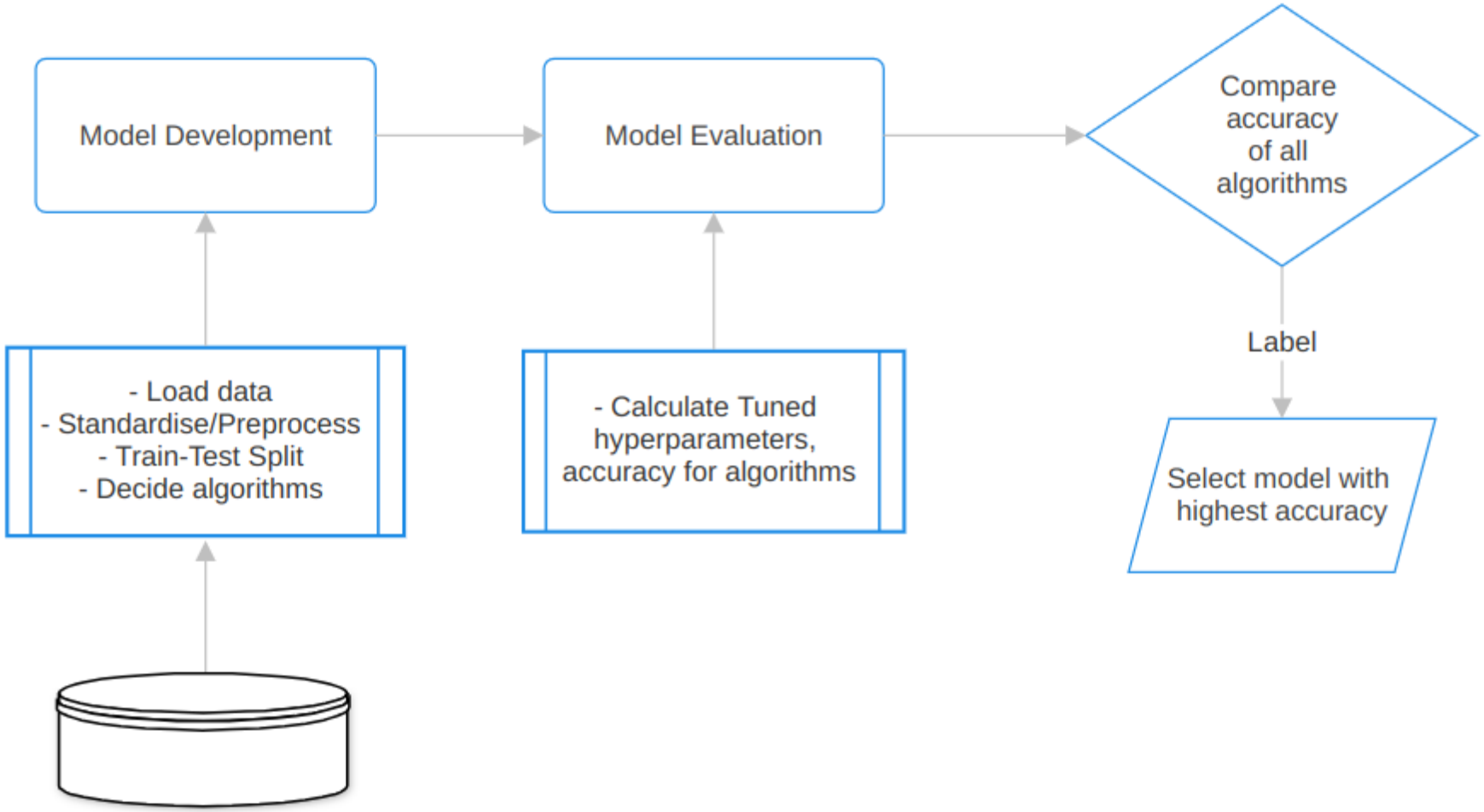
1. Pie chart: It shows the total successful launches per site.
   - This makes it clear to see which sites are most successful.
   - The chart could also be filtered (using a dcc.Dropdown() object) to see the success/failure ratio for an individual site

2. Scatter graph: It shows the correlation between outcome (success or not) and payload mass (kg)
   - This could be filtered (using a RangeSlider() object) by ranges of payload masses
   - It could also be filtered by booster version

Github URL: https://github.com/spacex_dash_app.py

Predictive Analysis with Classification Models

Following steps were followed to find the best suited classification model:



Contd. Interactive maps...

1. Model Development:
   a. Preprare dataset for model development
      - Load dataset
      - Standardise and preprocess data
      - Split data into training and testing sets using train_test_split()
      - Choose an algorithm to build model for: Logistic Regression, Decision tree, Support Vector Machine (SVM), K-Neatest Neighbour(KNN)

   b. For each chosen algorithm:
      - Create a GridSearchCV object with dictionary parameters.
      - Fit the object to get the best parameters from dictionary parameters.
      - Use training data set to train the model

Contd. Interactive maps...

2. Model Evaluation:
    a. For each chosen algorithm:
        - Check tuned hyperparameters (best_params_) for output GridSearchCV
        - Check accuracy of output GridSearchCV (using best_score_ and .score())
        - Plot and examine confusion matrix

3. Conclusion: Find the best classification model
        - Review the accuracy score drawn from all the algorithms.
        - The model with highest accuracy score will be considered as best suited classification model.

Github URL: https://github.com/capstone-prediction-analysis.ipynb

Results

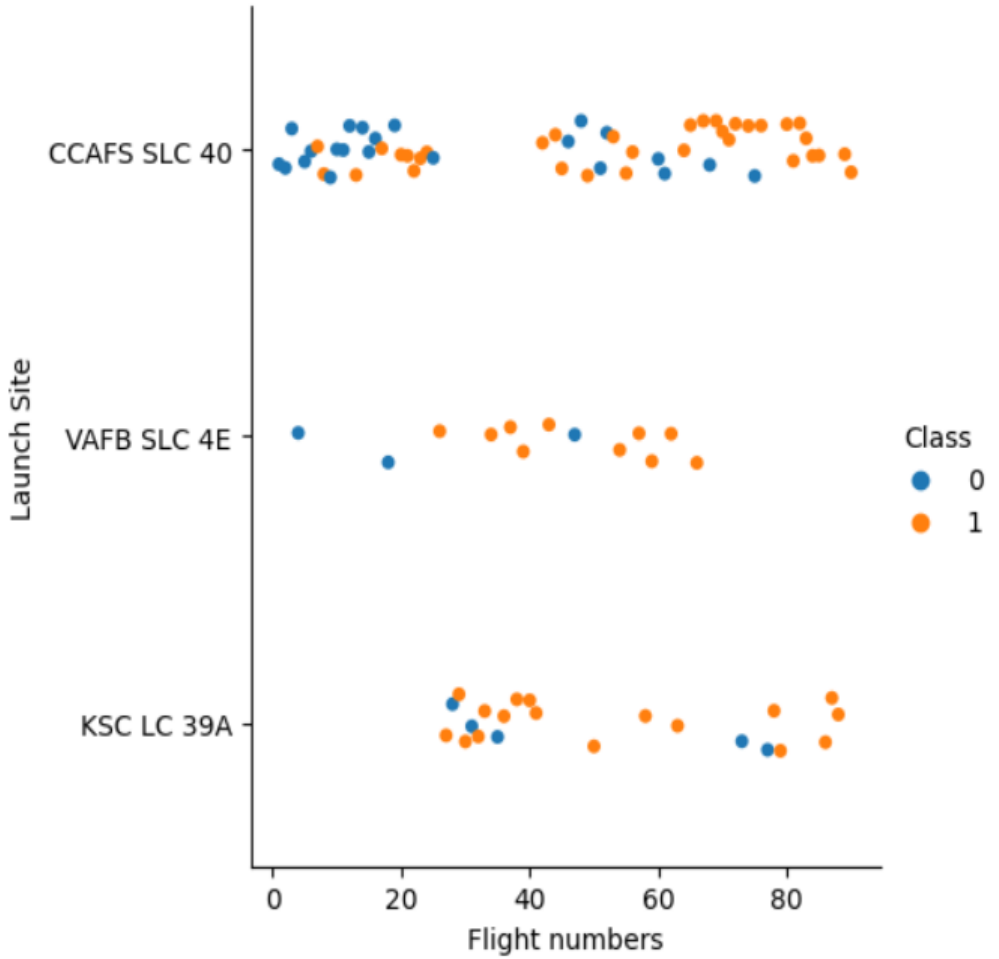
1. Exploratory Data Analysis
2. Interactive Analytics
3. Predictive Analysis

Exploratory Data Analysis (EDA) Insights

Flight Number vs. Launch Site

Following is the scatter plot of Launch Site vs. Flight Number.

- As the number of flights increased, launch site success rate also increased.
- Lot of CCAFS SLC 40 flights had unsuccessful landings for initial flights < 30
- No flights were launched initially for KSC LC 39A
- After 30 flights, there are significantly more successful landings for Class 1

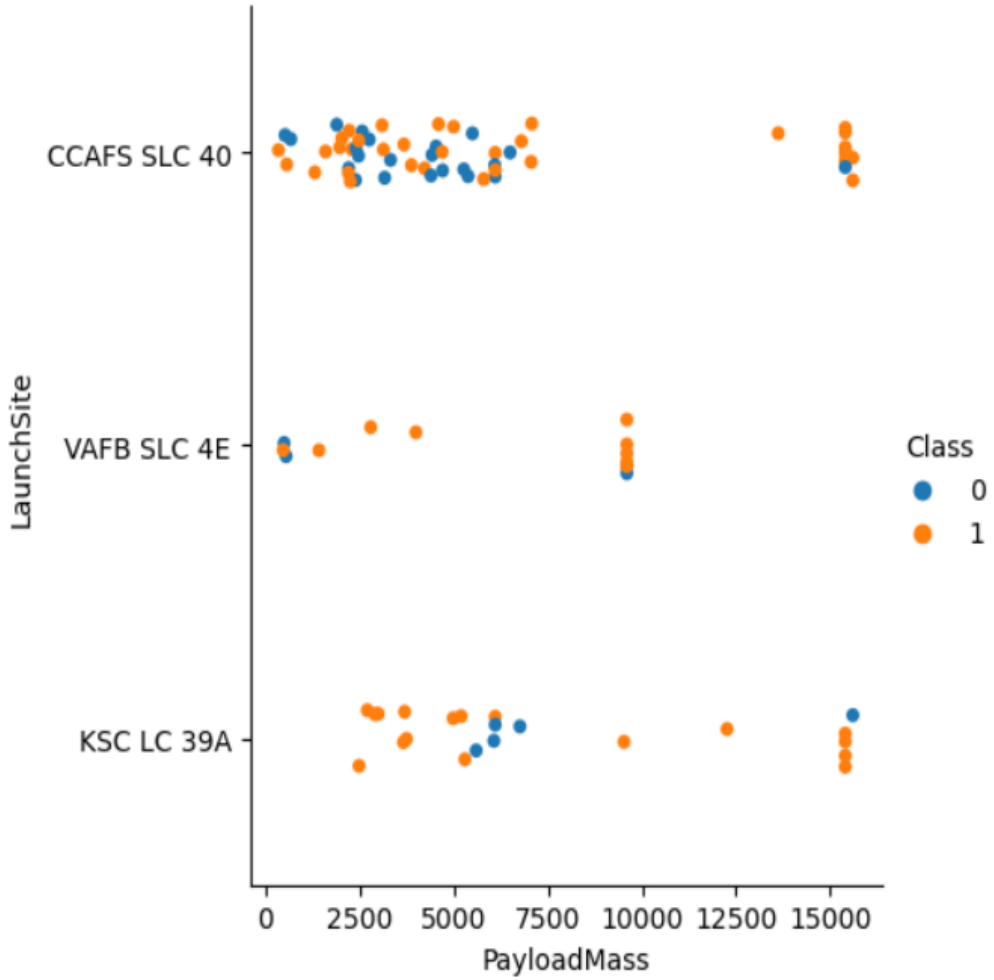


Payload vs. Launch Site

Following is the scatter plot of Launch Site vs. Payload

- Above a payload mass of around 7000 kg, there are very few unsuccessful landings, but there is also far less data for these heavier launches.
- There is no clear correlation between payload mass and success rate for a given launch site.

- All sites launched a variety of payload masses, with most of the launches from CCAFS SLC 40 being comparatively lighter payloads (with some outliers).



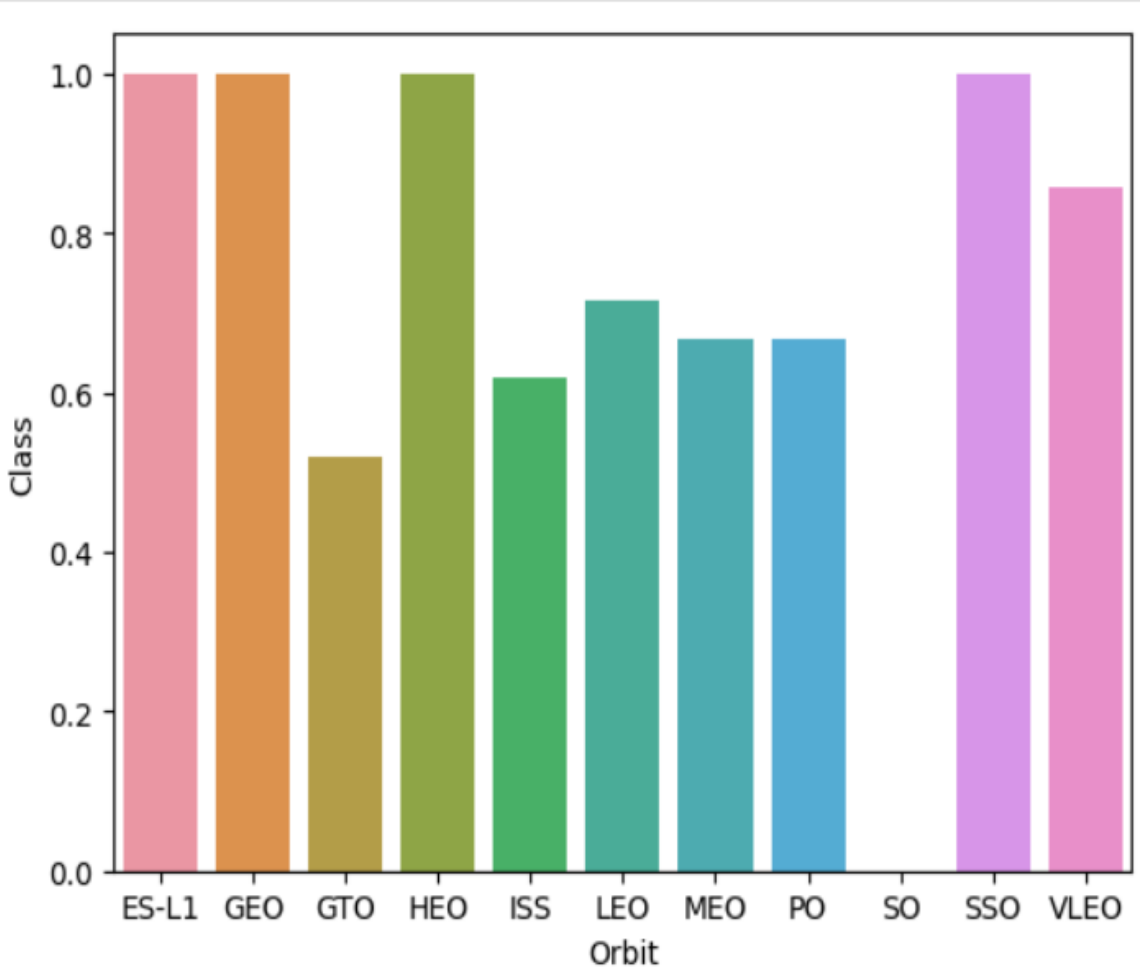
Success Rate vs. Orbit Type

The bar chart of Success Rate vs. Orbit Type shows that the following orbits have the highest (100%) success rate:

- ES-L1 (Earth-Sun First Lagrangian Point)
- GEO (Geostationary Orbit)
- HEO (High Earth Orbit)
- SSO (Sun-synchronous Orbit)

The orbit with the lowest (0%) success rate is:

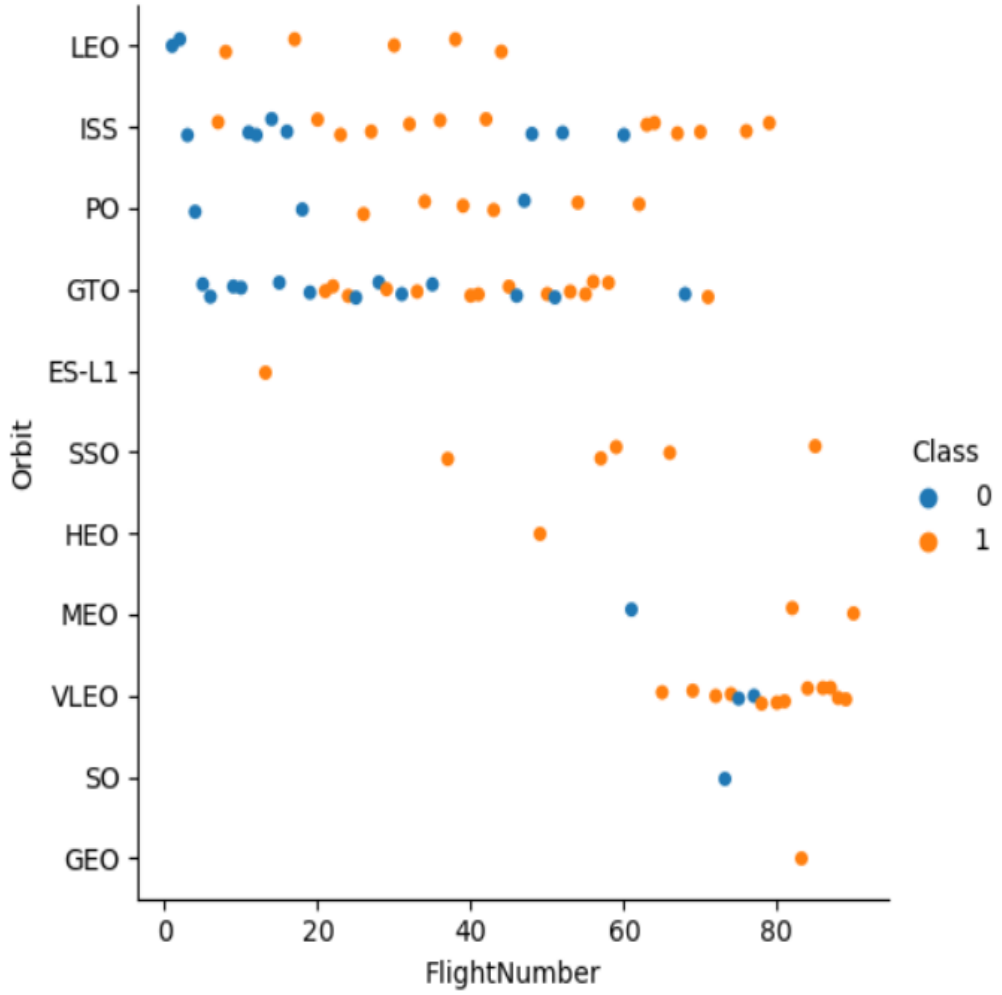
- SO (Heliocentric Orbit)



Flight Number vs. Orbit Type

This scatter plot of Orbit Type vs. Flight number shows a few useful things that the previous plots did not, such as:

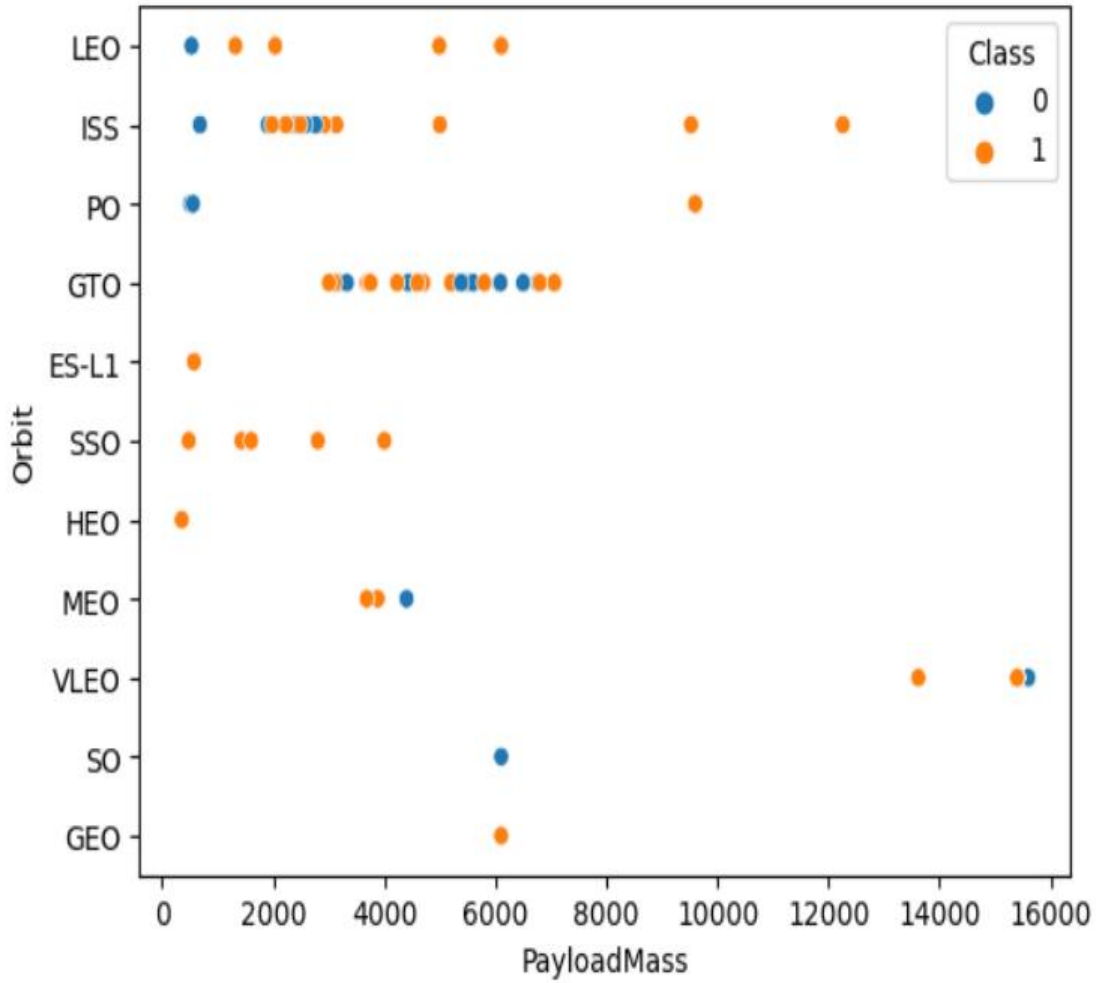
- The 100% success rate of GEO, HEO and ES-L1 orbits can be explained by only having 1 flight into the respective orbits.
- The 100% success rate in SSO is more impressive, with 5 successful flights.
- There is little relationship between Flight Number and Success Rate for GTO.
- Generally, as Flight Number increases, the success rate increases. This is most extreme for LEO, where unsuccessful landings only occurred for the low flight numbers (early launches).



Payload vs. Orbit Type

This scatter plot of Orbit Type vs. Payload Mass shows that:

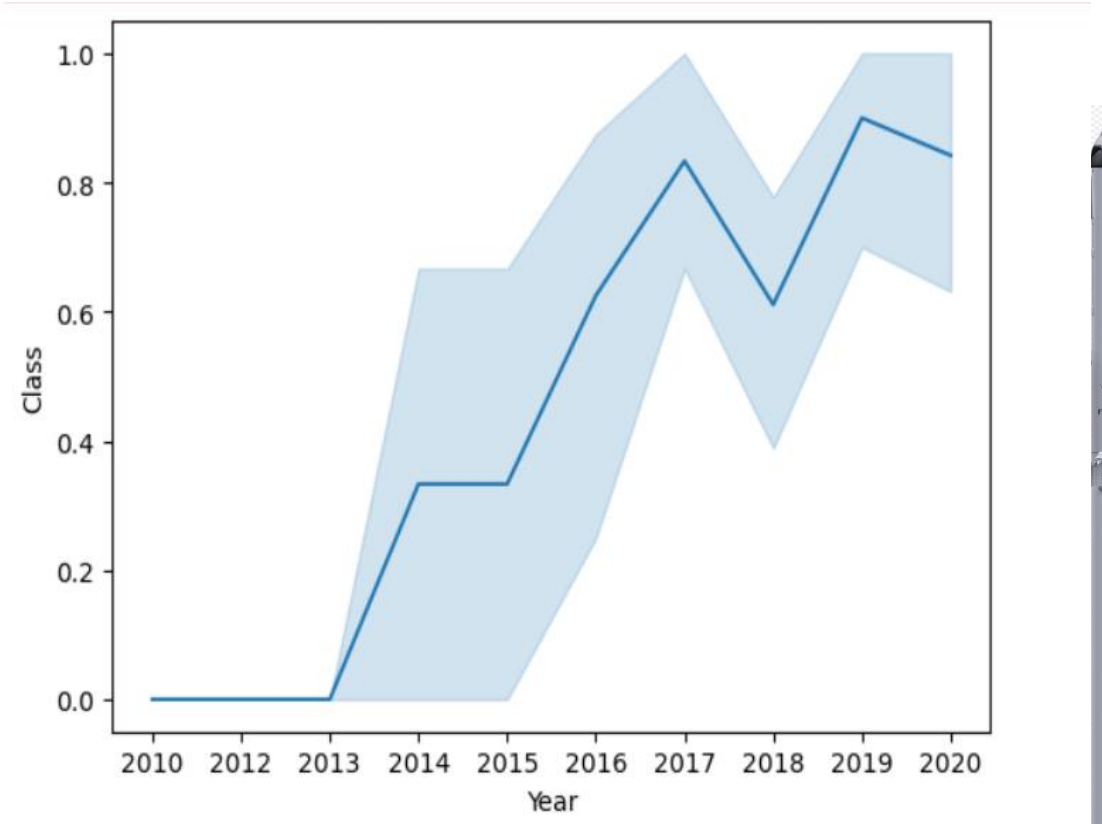
- Orbit types PO, ISS, LEO have more success with heavy payloads. Although the number of data points for PO is small.
- For GTO, the relationship between payload mass and success rate is unclear.
- VLEO (Very Low Earth Orbit) launches are associated with heavier payloads, which makes intuitive sense.



Launch Success Yearly Trend

The line chart of yearly average success rate shows that:

- Between 2010 and 2013, all landings were unsuccessful (as the success rate is 0).
- After 2013, the success rate generally increased, despite small dips in 2018 and 2020.
- After 2016, there was always a greater than 50% chance of success.



All Launch Sites

There are total 4 unique Launch Sites as follows:

1. CCAFS SLC 40
2. VAFB SLC 4E
3. KSC LC 39A
4. CCAFS LC-40

```
%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 'CCA'

Finding launch sites whose name begins with 'CCA'

```
%sql select * from SPACEXTABLE where "Launch_Site" like "CCA%" limit 5
```

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

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

- The 'Limit 5' keyword showcases only first 5 records
- The 'LIKE' keyword is used with wildcard symbol '%' to retrieve string values which starts with 'CCA'

Total Payload Mass

Calculate the total payload carried by boosters from NASA

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

```
%sql select Customer, sum("PAYLOAD_MASS__KG_") AS total_payload from SPACEXTABLE where Customer="NASA (CRS)"
```

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

Customer	total_payload
NASA (CRS)	45596

The 'SUM' is an aggregate function used to calculate summation of specified column on the data filtered with 'WHERE' condition.

Average Payload Mass carried by booster F9 V1.1

Calculate the average payload carried by boosters version F9 v1.1

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

```
%sql select Booster_Version, AVG("PAYLOAD_MASS__KG_") AS avg_payload from SPACEXTABLE where "Booster_Version" like "F9 v1.1%"
```

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

Booster_Version	avg_payload
F9 v1.1 B1003	2534.6666666666665

The 'AVG' is an aggregate function used to calculate average/mean of 'PAYLOAD_MASS__KG_' column on the data filtered with 'WHERE' condition to fetch 'Booster_Version' F9 v1.1 .

First Successful Ground Landing

Find the first successful ground landing date

```
%sql select MIN(Date) AS first_landing_date from SPACEXTABLE where "Landing_Outcome" = "Success (ground pad)"
```

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

first_landing_date
2015-12-22

The 'MIN' is an aggregate function used to fetch the smallest value of 'DATE' column on the data filtered with 'WHERE' condition for successful ground pad landing.

Successful Drone Ship Landing with Payload between 4000 and 6000

Listing names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000.

```
%%sql select Booster_Version, Launch_Site, PAYLOAD_MASS__KG_ from
SPACEXTABLE where "Landing_Outcome" = "Success (drone ship)"
and "PAYLOAD_MASS__KG_"> 4000 and "PAYLOAD_MASS__KG_" < 6000
```

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

Booster_Version	Launch_Site	PAYLOAD_MASS__KG_
F9 FT B1022	CCAFS LC-40	4696
F9 FT B1026	CCAFS LC-40	4600
F9 FT B1021.2	KSC LC-39A	5300
F9 FT B1031.2	KSC LC-39A	5200

Here, 'AND' clause is used to specify the range for 'PAYLOAD_MASS__KG_' to be between 4000 to 6000

Total Successful and Failed Mission Outcome

Get total number of failure and successful mission outcome.

```
%sql select Mission_Outcome, count(Mission_Outcome) as Count from SPACEXTABLE where Mission_Outcome like "Failure%"
```

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

Mission_Outcome	Count
Failure (in flight)	1

```
%sql select Mission_Outcome, count(Mission_Outcome) as Count from SPACEXTABLE where Mission_Outcome like "Success%"
```

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

Mission_Outcome	Count
Success	100

The COUNT keyword is used to calculate the total number of mission outcomes, and the GROUPBY keyword is also used to group these results by the type of mission outcome.

Boosters carrying maximum Payload

List the names of the booster which have carried the maximum payload mass.

```
%%sql select Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTABLE where PAYLOAD_MASS__KG_ =
(select max(PAYLOAD_MASS__KG_) from SPACEXTABLE)
```

```
 * 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

A subquery is used here. The SELECT statement within the brackets finds the maximum payload, and this value is used in the WHERE condition. The DISTINCT keyword is then used to retrieve only distinct /unique booster versions.

2015 Launch Records

Listing the failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql select Booster_Version, Launch_Site, Date, Landing_Outcome from SPACEXTABLE where
Landing_Outcome = "Failure (drone ship)" and substr(Date,0,5) ='2015'
```

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

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

The SUBSTR function is used to extract only year from the 'DATE'

Ranking 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.

```
%%sql select Landing_Outcome, count(Landing_Outcome) as No_of_outcome from SPACEXTABLE
Group by "Landing_Outcome"
Having Date > "2010-06-04" and Date < "2017-03-20"
Order by No_of_outcome DESC
```

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

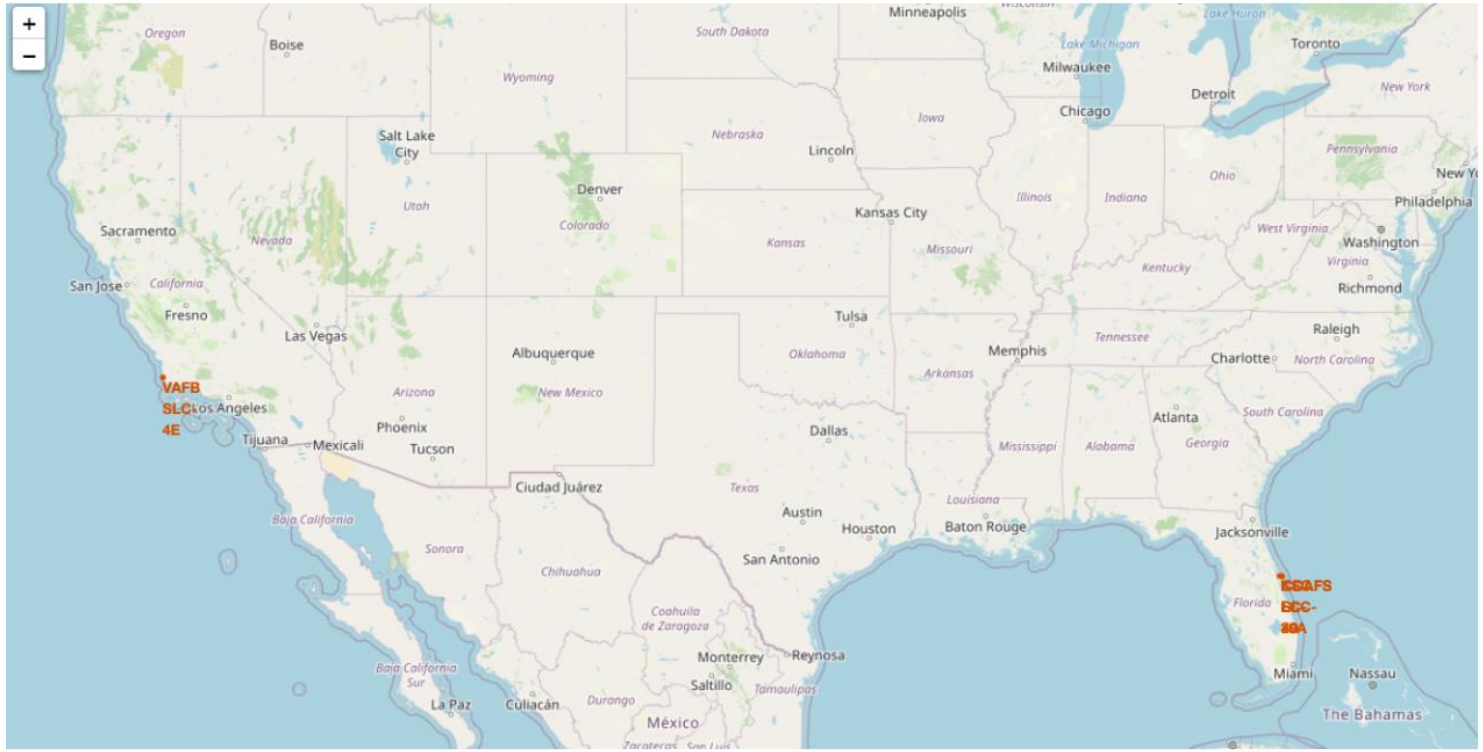
Landing_Outcome	No_of_outcome
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Uncontrolled (ocean)	2
Precluded (drone ship)	1

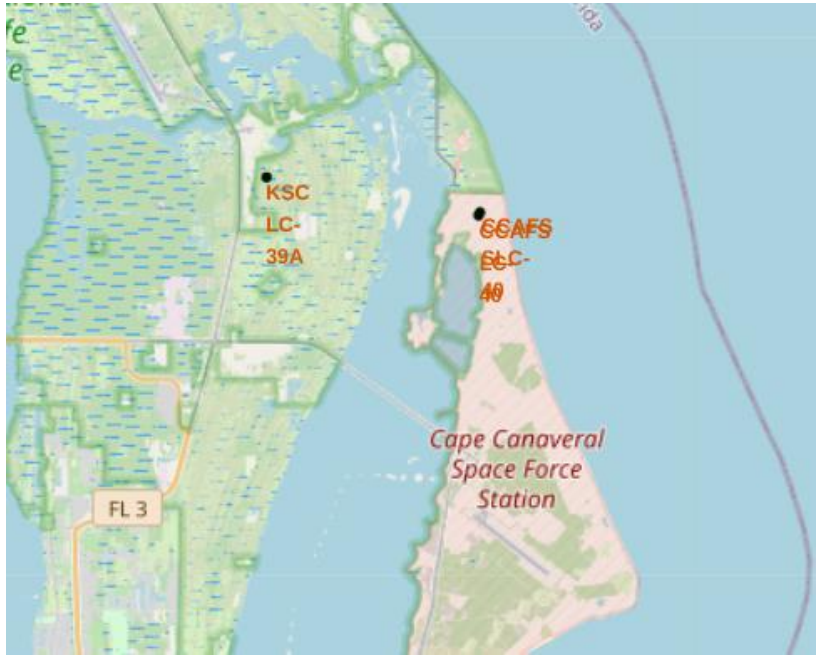
The WHERE keyword is used with the AND keyword to filter the results to dates only within those specified. The results are then grouped and ordered, using the keywords GROUP BY and ORDER BY, respectively, where DESC is used to specify the descending order.

Launch Sites Proximities Analysis

Mark All Launch Sites on Map

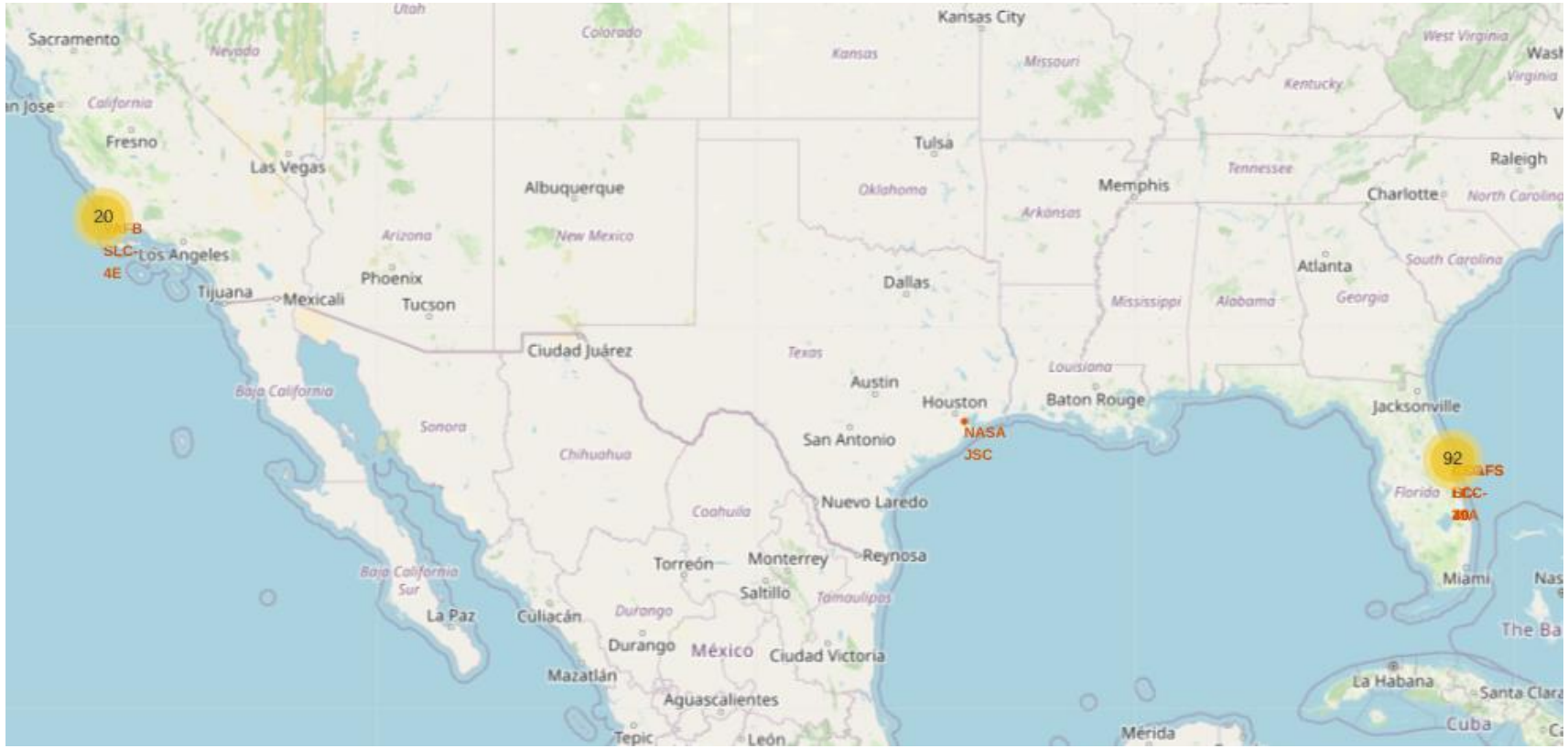
All SpaceX launch sites are on coasts of the United States of America, specifically Florida and California.

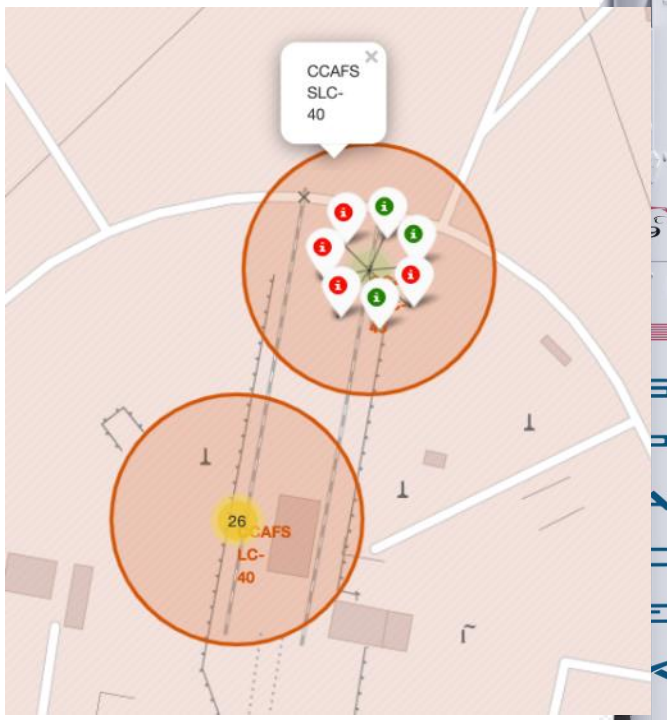




Mark Success and Failed Launch Sites on Map

Launches have been grouped into clusters, and annotated with green icons for successful launches, and red icons for failed launches.





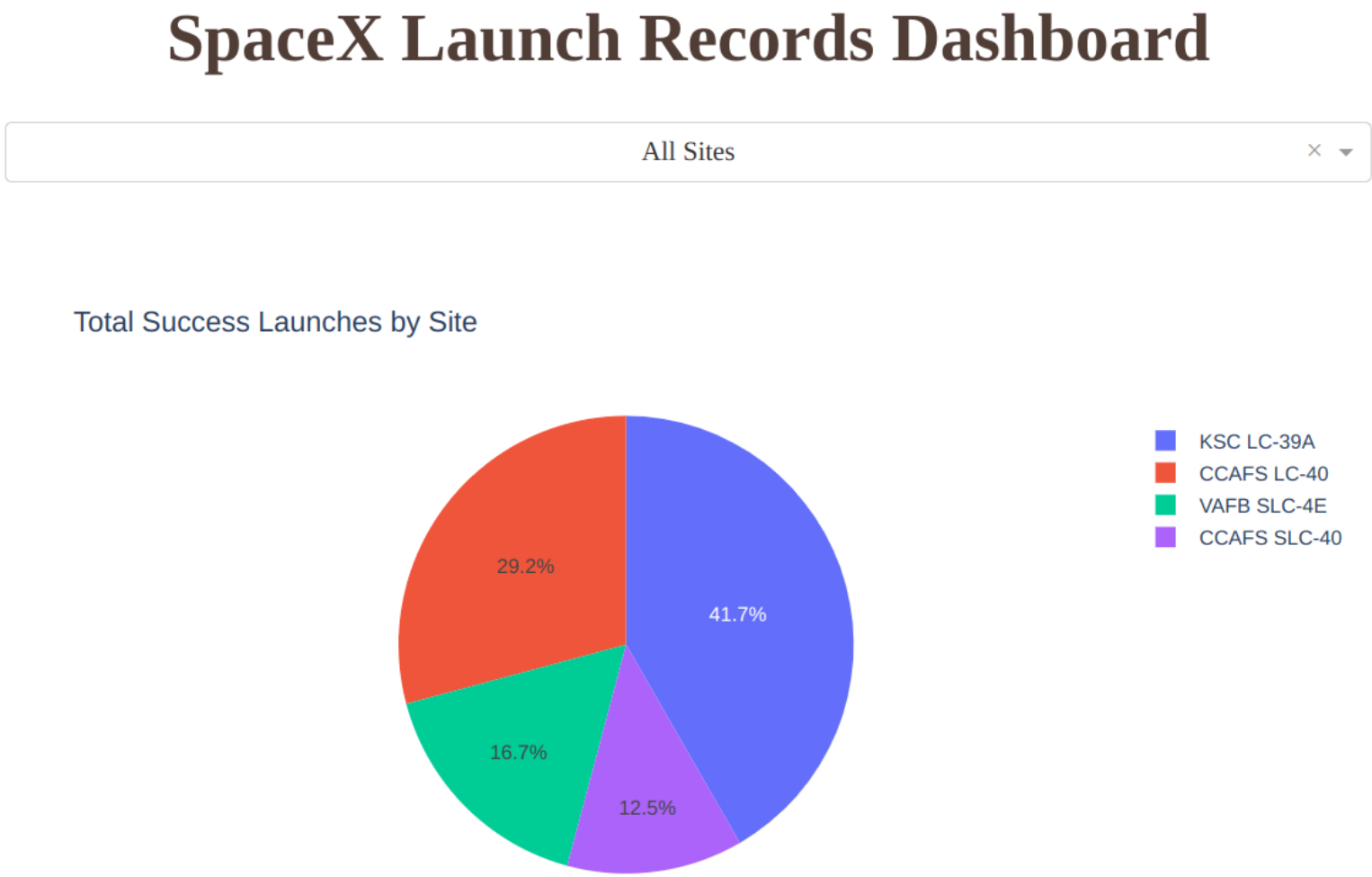
Distances between a Launch Site to Proximities

Using the CCAFS SLC-40 launch site as an example site, we can understand more about the placement of launch sites.

1. Are launch sites in close proximity to railways?
   YES. The coastline is only 0.87 km due East.
2. Are launch sites in close proximity to highways?
   YES. The nearest highway is only 0.59km away.
3. Are launch sites in close proximity to railways?
   YES. The nearest railway is only 1.29 km away.
4. Do launch sites keep certain distance away from cities?
   YES. The nearest city is 51.74 km away.

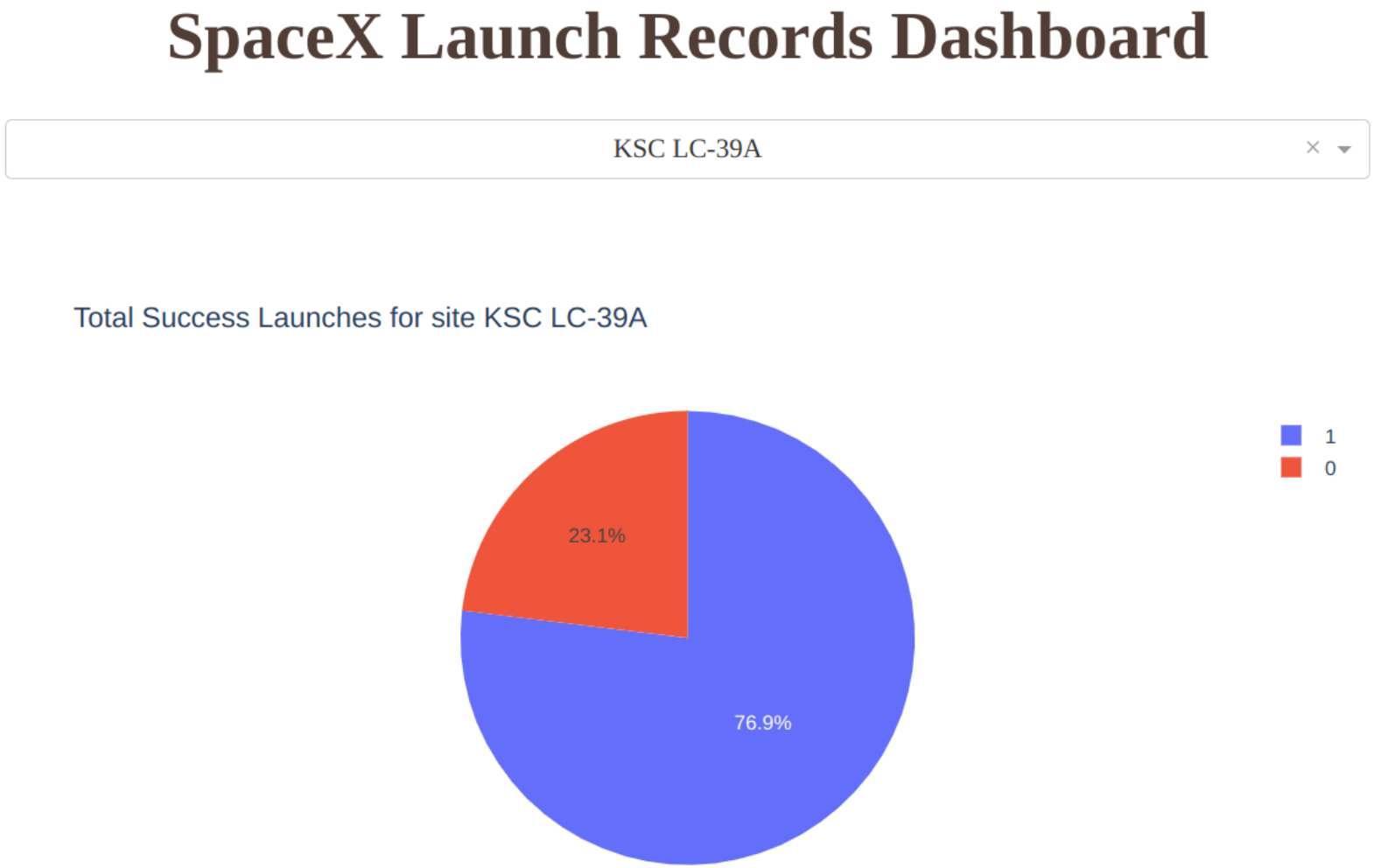
Interactive Dashboard with Plotly Dash

Launch Success for All Sites



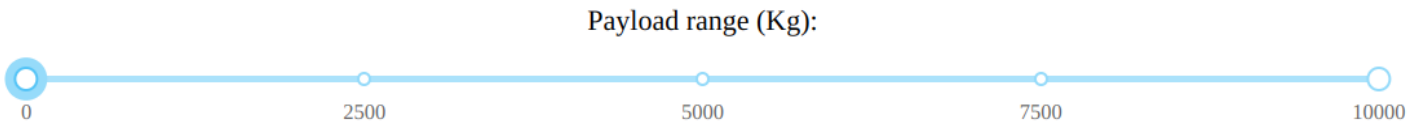
KSC LC 39A had most successful launches among all the sites with 41.7% and CCAFS SLC-40 with least successful launches of 12.5%

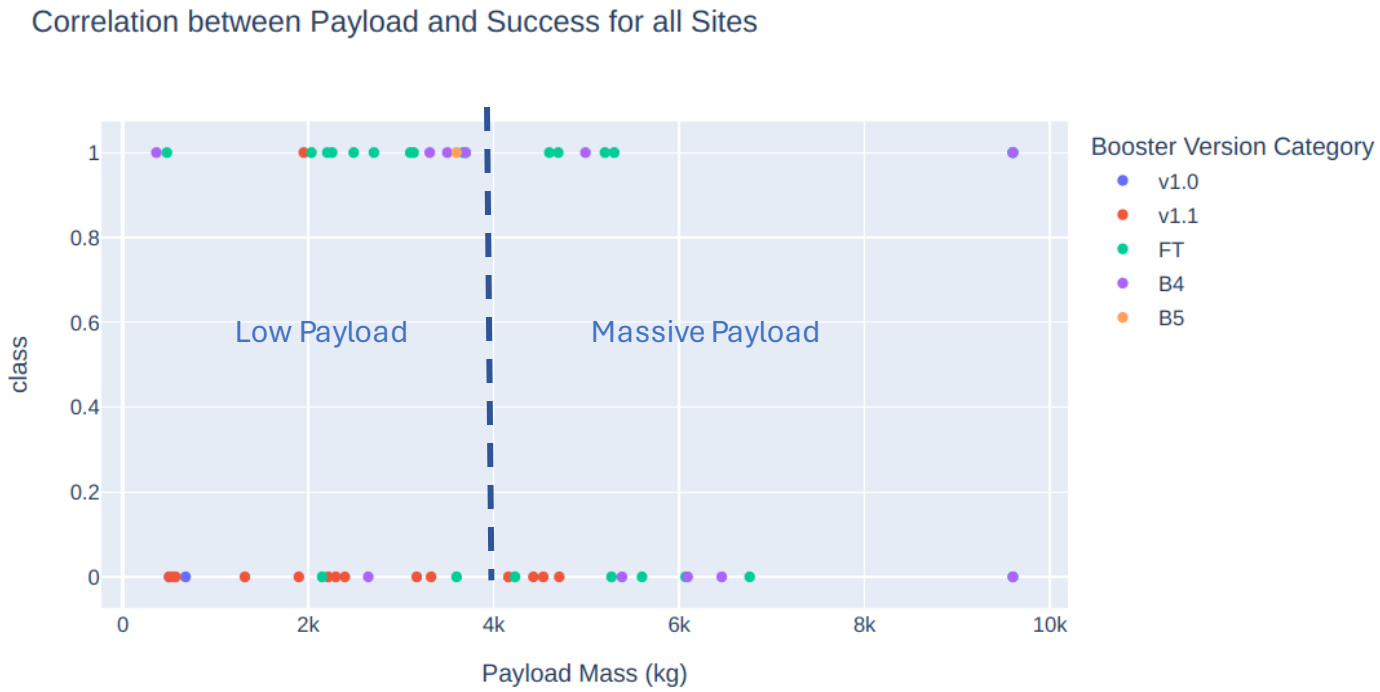
Launch Success for KSC LC 39A



KSC LC 39A had successful launches with a success rate of 76.9% It also high among all other site's success rate.

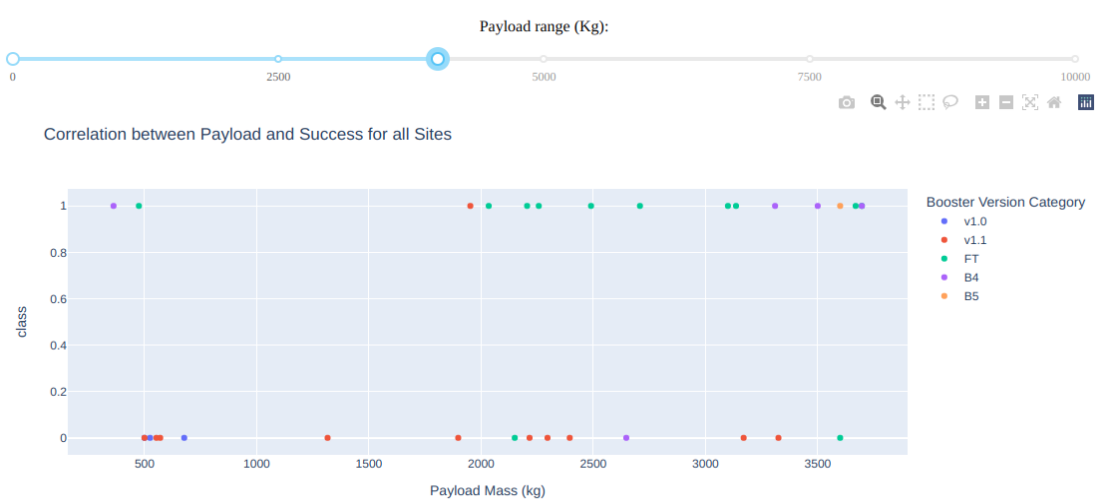
Launch Outcome VS. Payload scatter plot for all sites



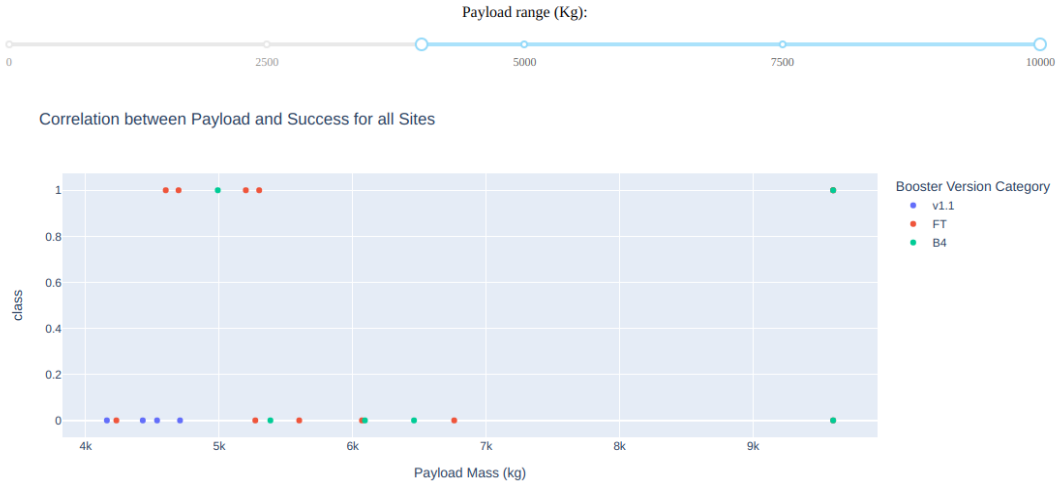


- Plotting the launch outcome vs. payload for all sites shows a gap around 4000 kg, so it makes sense to split the data into 2 ranges:
  - 0 – 4000 kg (low payloads)
  - 4000 – 10000 kg (massive payloads)

Contd. Launch Outcome VS. Payload scatter plot for all sites



Low Payload

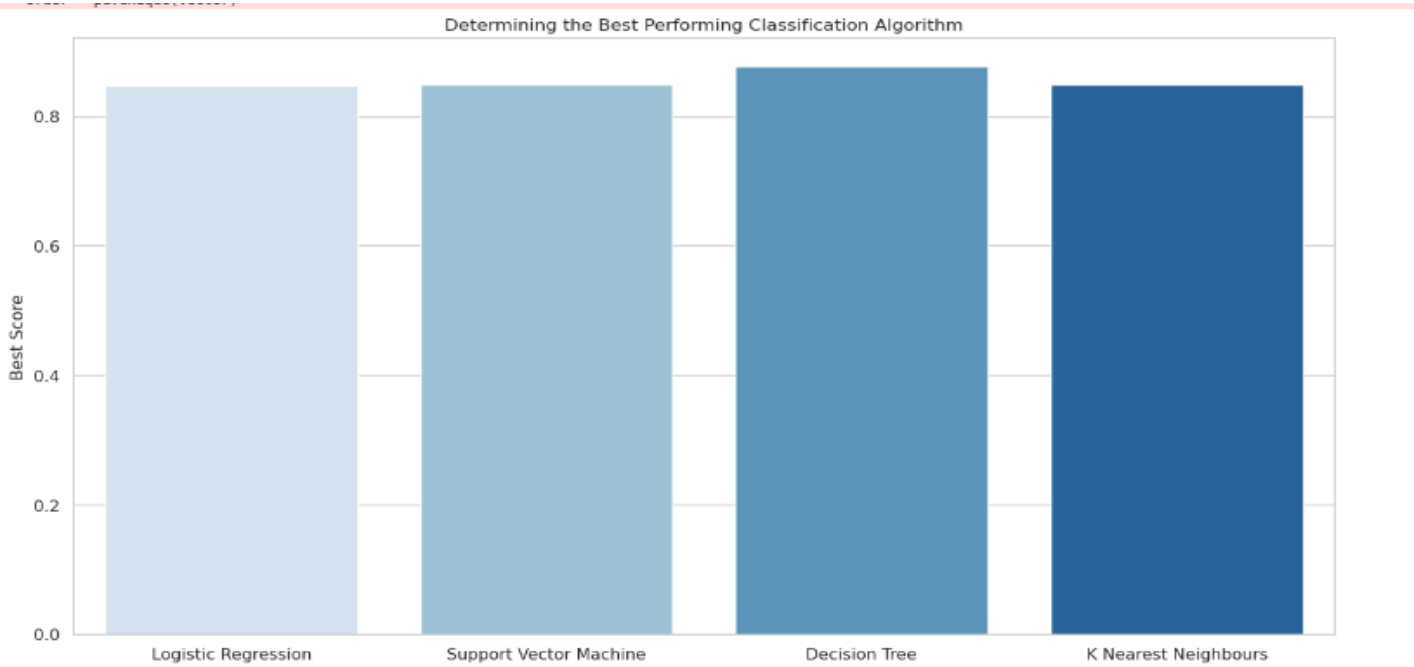


Massive Payload

- From these 2 plots, it can be shown that the success for massive payloads is lower than that for low payloads.
- It is also worth noting that some booster types (v1.0 and B5) have not been launched with massive payloads.

Predictive Analysis (Classification)

Classification Accuracy

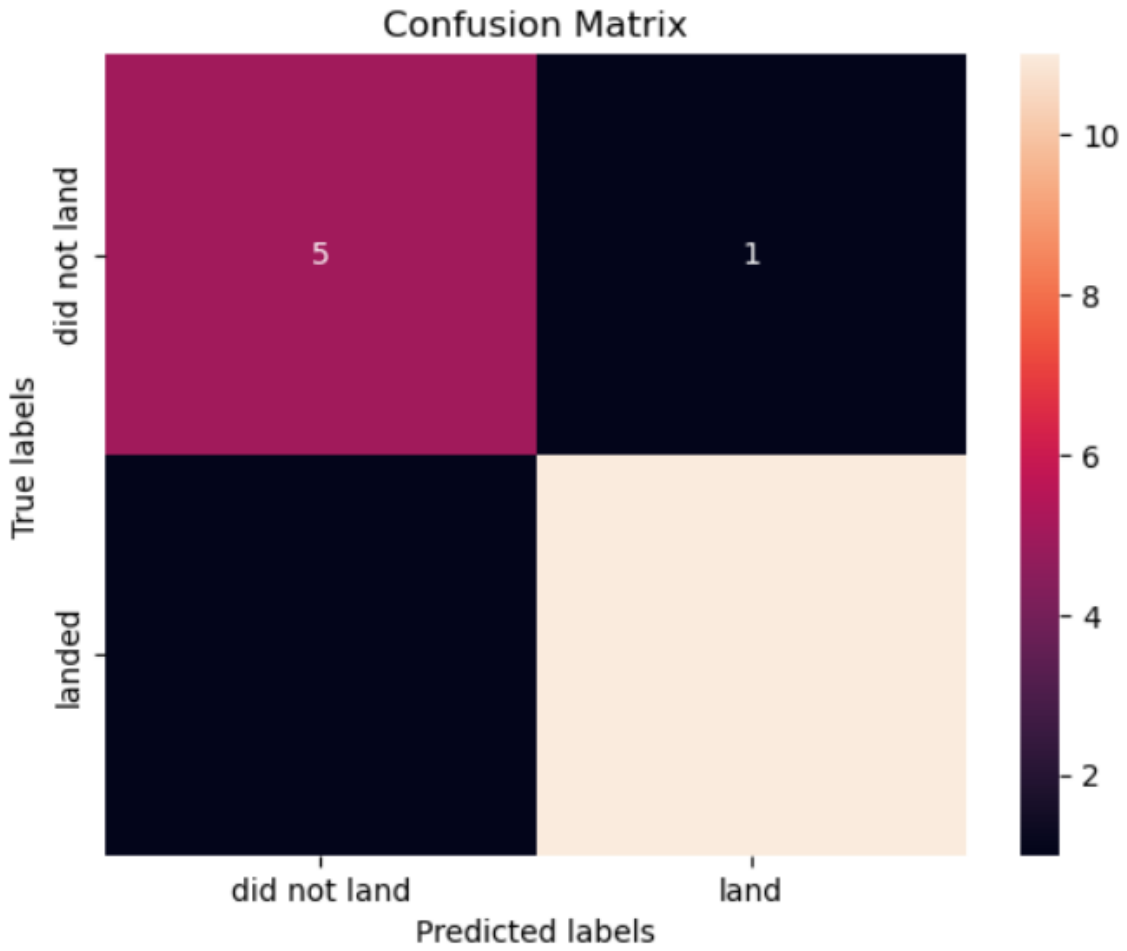


Algorithm	Accuracy Score	Best Score
Logistic Regression	0.833333	0.846429
Support Vector Machine	0.833333	0.848214
Decision Tree	0.888889	0.876786
K Nearest Neighbours	0.833333	0.848214

Plotting the Accuracy Score and Best Score for each classification algorithm produces the following result:

- The Decision Tree model has the highest classification accuracy
- The Accuracy Score is 94.44%
- The Best Score is 90.36%

Confusion Matrix



- As shown previously, best performing classification model is the Decision Tree model, with an accuracy of 94.44%.
- This is explained by the confusion matrix, which shows only 1 out of 18 total results classified incorrectly (a false positive, shown in the top-right corner).
- The other 17 results are correctly classified (5 did not land, 12 did land).

Conclusion

Conclusion

- As the number of flights increases, the rate of success at a launch site increases, with most early flights being unsuccessful. I.e. with more experience, the success rate increases.
  - Between 2010 and 2013, all landings were unsuccessful (success rate is 0).
  - After 2013, the success rate generally increased, despite small dips in 2018 and 2020.
  - After 2016, the chance of success has been more than 50%.
- Orbit types ES-L1, GEO, HEO, and SSO have the highest (100%) success rate.
  - The 100% success rate of GEO, HEO, and ES-L1 orbits can be explained by only having 1 flight into the respective orbits.
  - The 100% success rate in SSO is more impressive, with 5 successful flights.
  - The orbit types PO, ISS, and LEO, have more success with heavy payloads: VLEO (Very Low Earth Orbit) launches are associated with heavier payloads, which makes intuitive sense.
- The launch site KSC LC-39 A had the most successful launches, with 41.7% of the total successful launches, and also the highest rate of successful launches, with a 76.9% success rate.
- The success for massive payloads (over 4000kg) is lower than that for low payloads.
- The best performing classification model is the Decision Tree model, with an accuracy of 94.44%.

Appendix

- Data Collection:
  - Custom functions to retrieve required information
  - Custom logic to clan the data

1. SpaceX Rest API:

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

2. Web Scrapping: Read Wikipedia page from its URL

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

- Data Wrangling:
  - Exploratory Data Analysis
  - Determine Training Labels

If Outcome is in state {'False ASDS', 'False Ocean',
'False RTLS', 'None ASDS', 'None None'}
then landing_class = 0
Else landing_class = 1

```
landing_class=[]

for outcome in df['Outcome']:
    if outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Cont. Appendix

- Data Visualization:
  - Plotting different types of plots to identify relationship between each entity.
  - Types of plots used: Scatter plot, Bar chart and Line chart
  - Libraries used: Seaborn and Matplotlib
  - Features Engineering: Identifying features that can be used in success prediction.

```
features = df[['FlightNumber', 'PayloadMass', 'Orbit', 'LaunchSite', 'Flights', 'GridFins', 'Reused', 'Legs', '
features
```

Creating dummy categorical columns

```
features_one_hot = pd.get_dummies(features)
```

- SQL:
  - Connecting to database

```
con = sqlite3.connect("my_data1.db")
cur = con.cursor()
```

  - Read data into a table

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

Thank You!