Winning Space Race with Data Science

Simon Njoku

4th April, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through the API
  - Data Collection with Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis (EDA) with SQL
  - Exploratory Data Analysis (EDA) with Data Visualization
  - Use of Folium library for the Interactive Visual Analysis
  - Machine Learning (ML) Predictions
- Summary of all results
  - Exploratory Data Analysis using Python
  - Screenshot results of the Interactive analytics
  - Results of Predictive Analysis extracted from the Machine Learning Lab

Introduction

- Project background and context

  The age of commercial space is here and for that companies are making space travel affordable for everyone. SpaceX is the most successful of them all because they offer rocket launches specifically Falcon 9 with a cost as low as 62 million dollars while other providers offer rocket launches above 165 million dollars for each launch. This cost saving from SpaceX is as a result of the re-use of their first stage of launch and re-landing the rocket to be used on the next mission.

  The goal of this project as a Data Scientist is to train a Machine Learning model to predict the first stage landing outcome in the future and this will play a vital role in establishing the right cost for a rocket launch that will compete against SpaceX.

- Problems

  What are the factors affecting the landing outcomes

  What relationship exists amongst the variables that plays a part in the outcome

  What is the optimal condition for increasing a successful landing

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - SpaceX Launch Data is gathered from SapceX REST API and Web scraping related Wiki Pages. This API will give us data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
- Perform data wrangling
  - Data was processed using the one-hot encoding for categorical features
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

Data Collection

- Data collection involves processes of gathering and measuring useful information relating to the questions that need answered, establishing the variables and evaluating the results. Dataset used in this process were collected from REST API and Related Web scraping Wiki Pages.
- REST API: the URL (https://api.spacexdata.com/v4/launches/past) was used to target a specific endpoint of the API to get past launch data, then applied the“get request” library to obtain the data from API. The response will be in JSON which is converted and normalized into pandas dataframe with jason_normalize() method. The raw data was cleaned, checked for null values and transformed into a meaningful dataset for analysis.
- Web scraping: used BeautifulSoup to extract the launch records as HTML table, parse the table and convert it to a pandas dataframe for further analysis

Data Collection – SpaceX API

- Collected Rocket Launch Data from API using get request library
- Applied json_normalize method to convert json format to dataframe
- Performed data cleaning and filled null values

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

[13]: response = requests.get(spacex_url)
```

```
In [57]: # Lets take a subset of our dataframe keeping only the features we want and the flight number, and date_utc.
data = data[['rocket', 'payloads', 'launchpad', 'cores', 'flight_number', 'date_utc']]

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

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

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

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

Completed github code link below:

(https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/predict%20if%20the%20Falcon%209%20first%20stage%20will%20land%20successfully.ipynb

Data Collection - Scraping

- Web scraping: Requested Falcon 9 Related Wiki Pages from the url
- Created BeautifulSoup from the HTML response
- Extract all column/variable names from the HTML header
- Github code link below:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/web%20scraping%20to%20collect%20Falcon%209%20historical%20launch%20records%20from%20a%20Wikipedia%20page%20titled%20List%20of%20Falcon%209%20and%20Falcon%20Heavy%20launches.ipynb

In [9]:
```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
```

In [15]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

In [17]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

In [28]:
```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as number corresponding to launch a number
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
        else:
            flag=False
        #get table element
        row=rows.find_all('td')
        #if it is number save cells in a dictonary
        if flag:
            extracted_row += 1
            # Flight Number value
            # Append the flight_number into launch_dict with key `Flight No.`
            launch_dict["Flight No."].append(flight_number)
            #print(flight_number)
            datatimelist=date_time(row[0])
```

Data Wrangling

- Determined the number of launches on each site using value_counts()
- The Dataset were processed by converting outcomes of the landing into training labels by designating1 to be booster successful and 0 means unsuccessful

```
In [7]: # Apply value_counts() on column LaunchSite
        df.value_counts(['LaunchSite'])

Out[7]: LaunchSite
        CCAFS SLC 40    55
        KSC LC 39A      22
        VAFB SLC 4E     13
        dtype: int64
```

```
In [8]: # Apply value_counts on Orbit column
        df.value_counts(['Orbit'])

Out[8]: Orbit
        GTO      27
        ISS      21
        VLEO     14
        PO        9
        LEO       7
        SSO       5
        MEO       3
        ES-L1     1
        GEO       1
        HEO       1
        SO        1
        dtype: int64
```

```
In [13]: landing_class = []

         for i in df["Outcome"]:
             if i in bad_outcomes:
                 landing_class.append(0)
             else:
                 landing_class.append(1)

         print(landing_class)

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
```

- GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/Space%20X%20Falcon%209%20First%20Stage%20Landing%20Prediction_data%20wrangling.ipynb

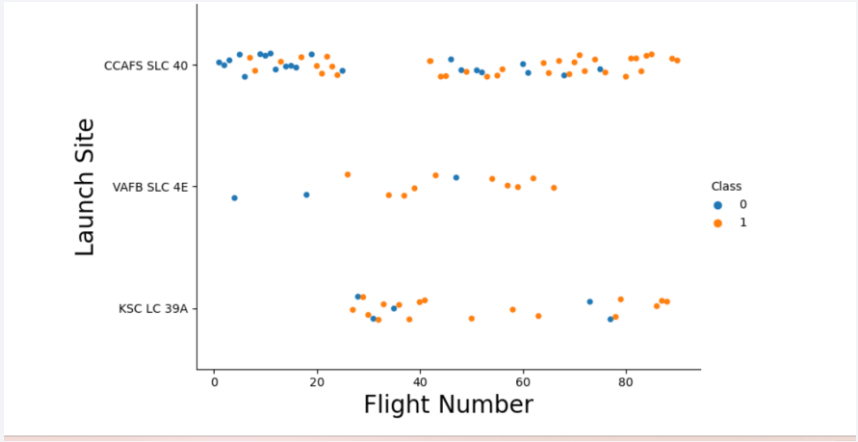
EDA with Data Visualization

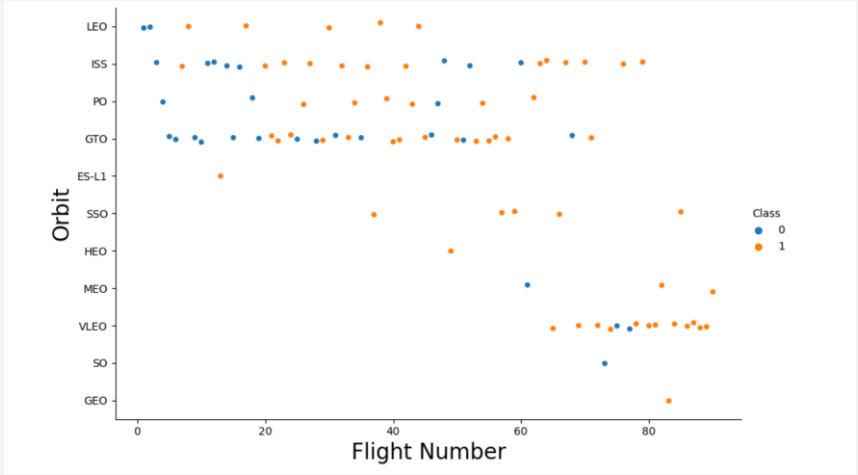
- The following charts were plotted to identify the relationships between them:
- Payload and Flight Number.
- Flight Number and Launch Site.
- Payload and Launch Site.
- Flight Number and Orbit Type.
- Payload and Orbit Type.

Scatter point chart displays a better relationship of attributes to each other. The Pattern from the graph is used to determine factors contributing to the success of the landing outcomes.

- GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/Data%20Viz_2%20launch%20site.ipynb





EDA with SQL

The following queries were performed to get better understanding of the dataset:

- Displaying the names of the unique launch sites in the space mission
- Displaying 5 records where launch sites begin with the string 'CCA'.
- Displaying the total payload mass carried by booster launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first successful landing outcome in ground pad was achieved.
- Listing the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster_versions which have carried the maximum payload mass.
- Listing the failed landing_outcomes in drone ship, their booster versions, and launch sites names for in year 2015.
- Rank the count of landing outcomes or success between the date 2010-06-04 and 2017-03-20, in descending order.
- GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/jupyter-labs-eda-sql-coursera_sqllite(1).ipynb

Build an Interactive Map with Folium

- Interactive map was created to visualize the launch data by considering the latitude and longitude coordinates at each launch site and added a circle marker around them labelling the name of each launch site.
- Assigned the launch_outcomes (failure,success) dataframe to classes 0 and 1 with red and green markers on the map using MarkerCluster().
- These objects were added to be able to easily identify launch sites that have relateively high success rate.
- It also helps to identify the proximity of these launch sites to railways, coastlines, highways, etc.
- GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/Launch%20Sites%20Location%20Aanalysis%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- Interactive dashboards using Plotly dash has been created with the data allowing the user to view different data at a given time
- These Plots and interaction were added to enable stakeholders filter out only the information they need.
- GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/spacex_dash_app_week%205.py

Predictive Analysis (Classification)

Model Development

- Using Python and loading dataset with NumPy and Pandas
- Data Transformation into test set and training set by splitting
- Machine Learning application
- set the parameters and algorithms to GridSearchCV and fit it to dataset.

Model Evaluation

- Model accuracy check
- retrieve tuned hyperparameters for each type of algorithms.
- plotting the confusion matrix.

Model Improvement

- The use of Engineering and Algorithm Tuning features

Discover the Best Model

- The model that gives the best accuracy score will be the best performing model.

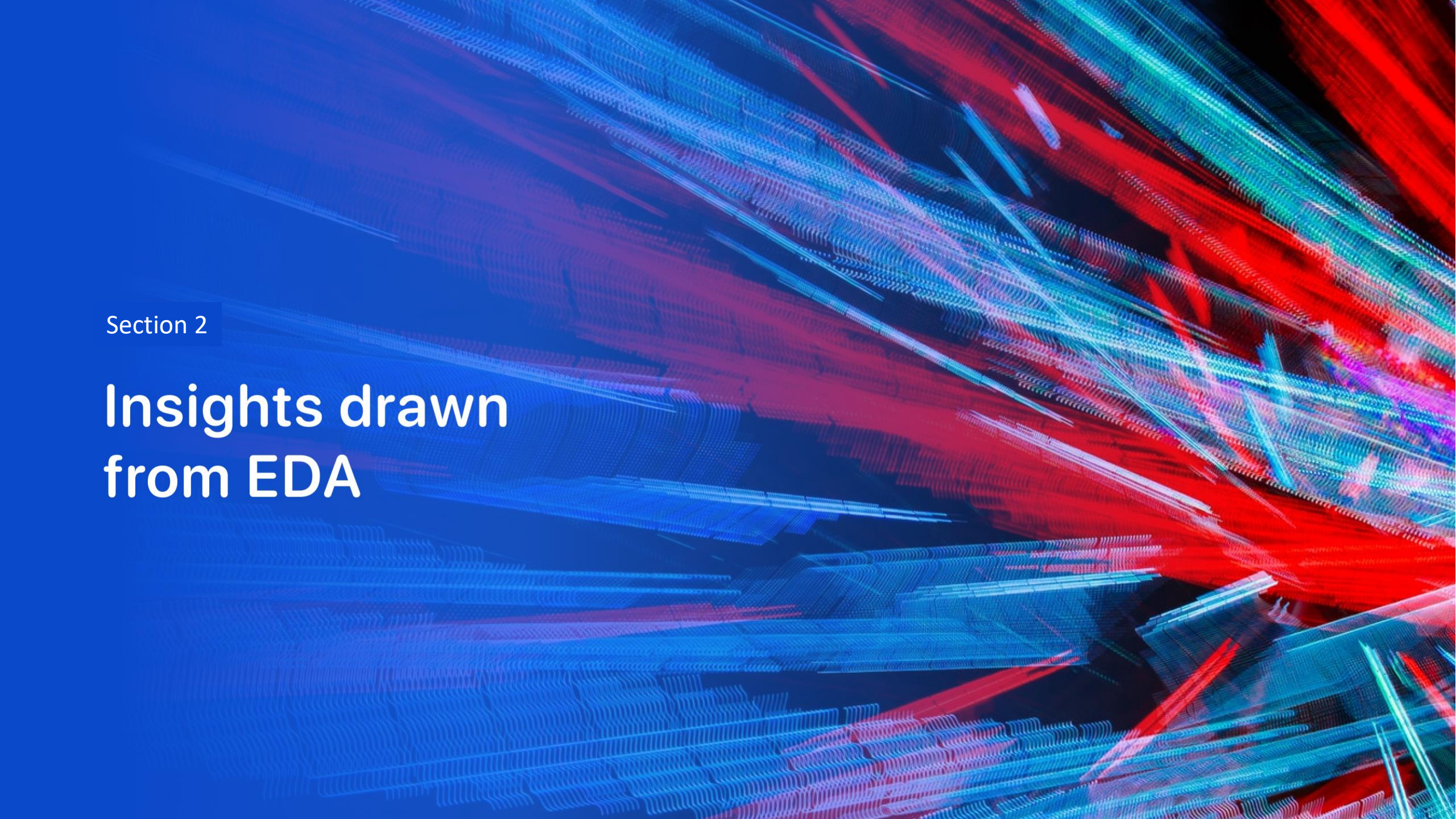
GitHub URL:

https://github.com/simonsteve/Applied-Data-Science-Capstone/blob/Master/spacex_dash_app_week%205.py

Results

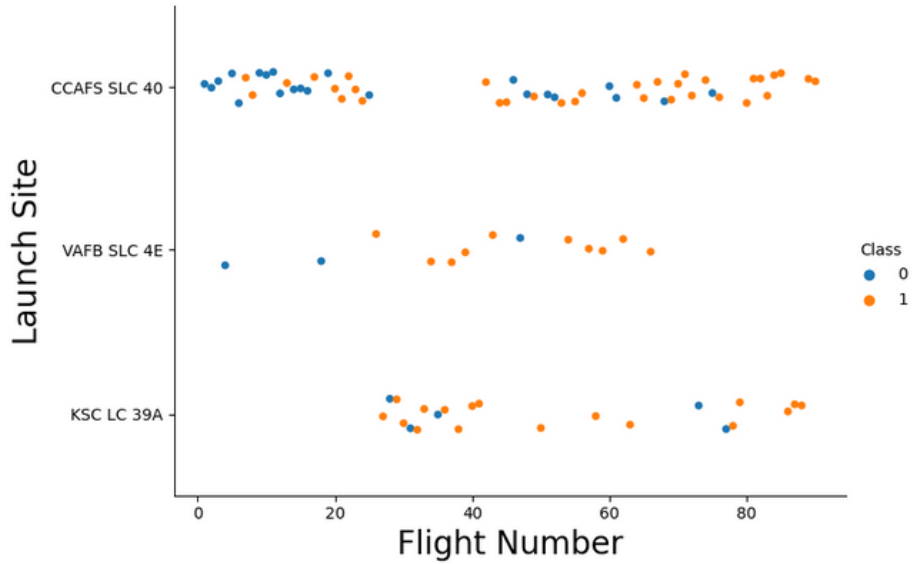
- This section is grouped into the following:
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

- Predictive Analysis showed that Support Vector Model, K-Nearest Neighbors and Logistic Regression Model are all best model to predict successful landings, having accuracy over 83% and accuracy for test data over 94%.


Section 2
Insights drawn from EDA

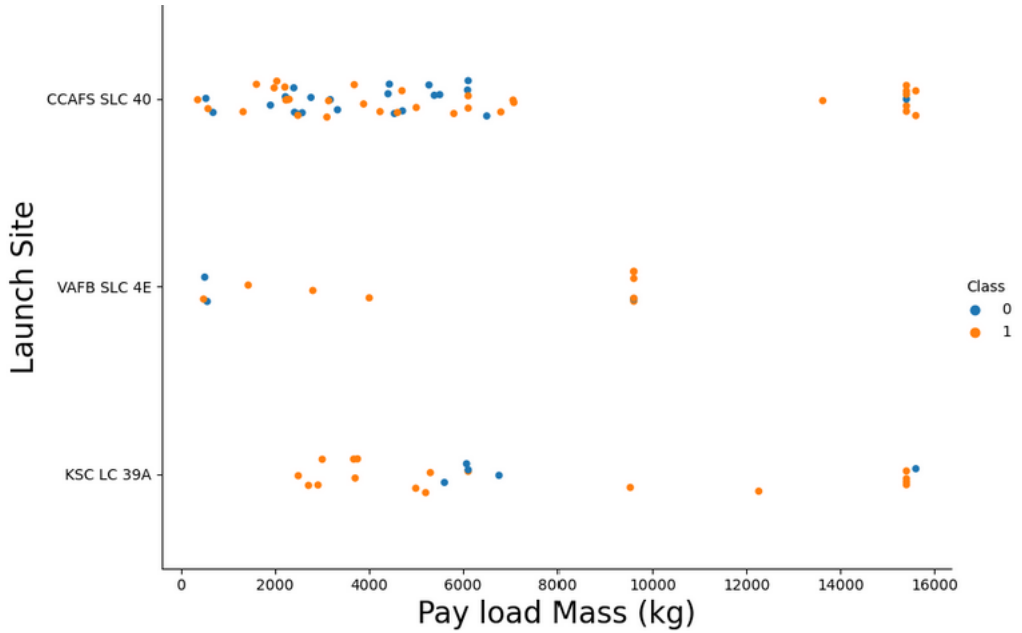
Flight Number vs. Launch Site

- This scatter plot shows the relationship between Flight Number on X-axis and Launch Site on Y-axis. We can see from the plot with class 1 as success rate and 0 as failure rate, that as the number of flights increase the better chances of the launch site success rate will be. But site CCAFS SLC40 shows the least pattern of this.



Payload vs. Launch Site

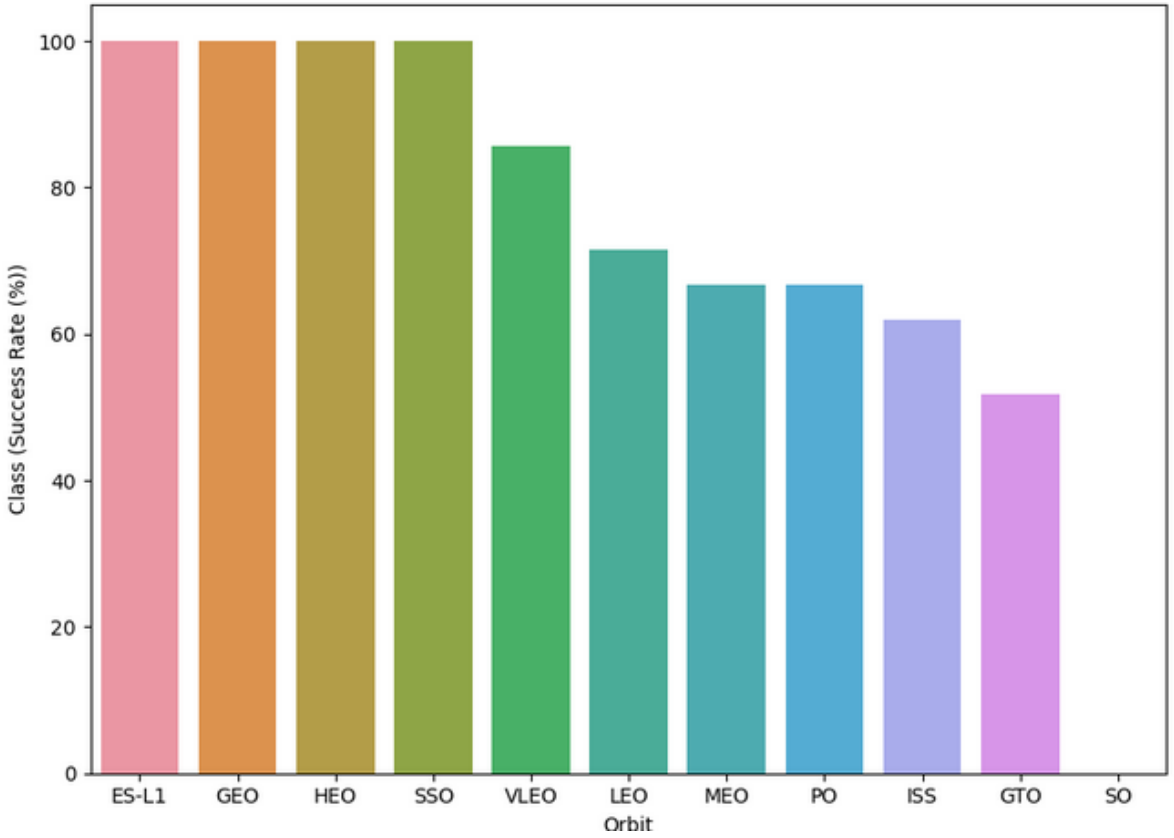
In this scatter point chart of Payload Vs Launch Site, you will observe that that there are no rockets launched for heavy-payload mass greater than 10,000kg



Success Rate vs. Orbit Type

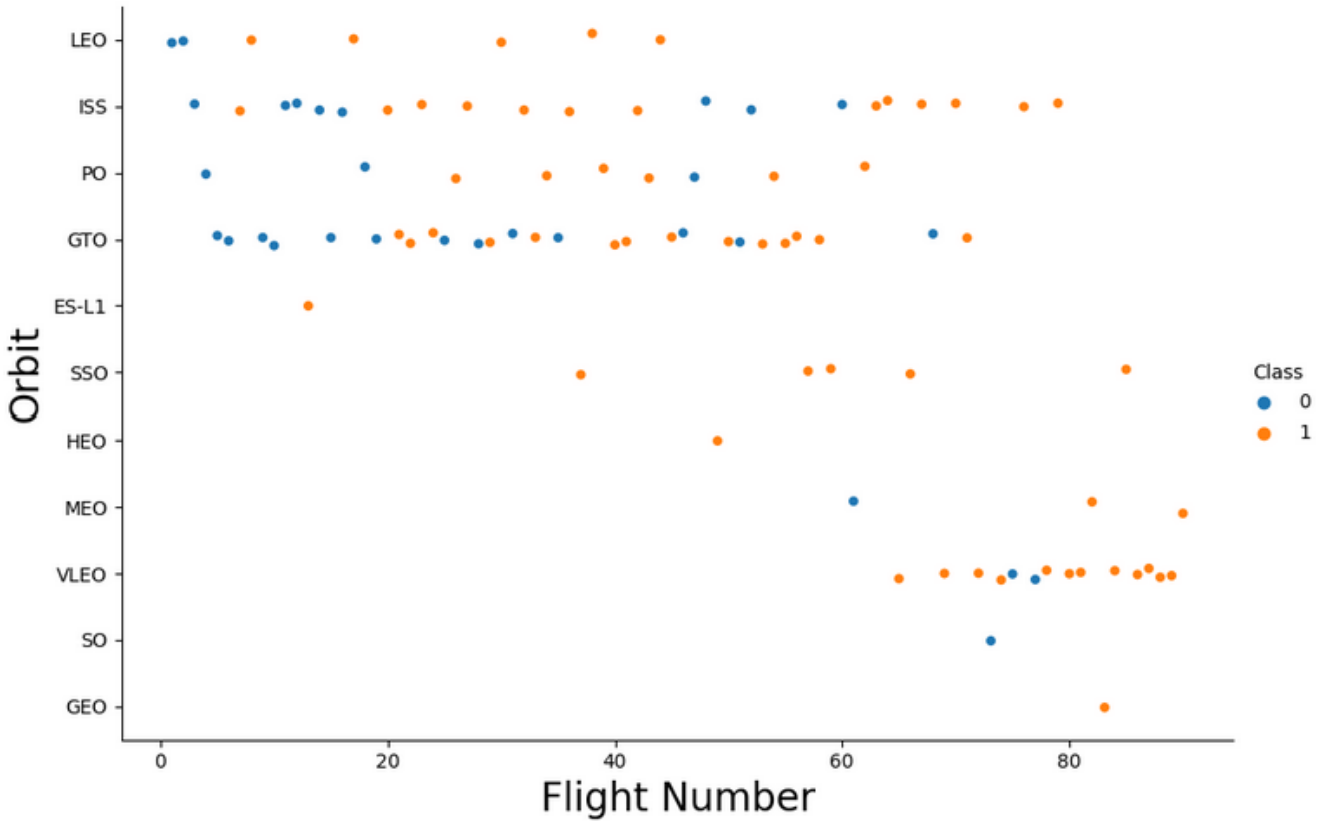
Orbits play an important role in the suc of landing outcomes, Orbits like ES-L1, GEO,HEO and SS0 have 100% success ra while SO orbit has zero percent.

Further analysis shows that some of the orbits like the GEO, SO, HEO and ES-L1 h just only one occurrence which means there are incomplete data collected and need more data to draw conclusion.



Flight Number vs. Orbit Type

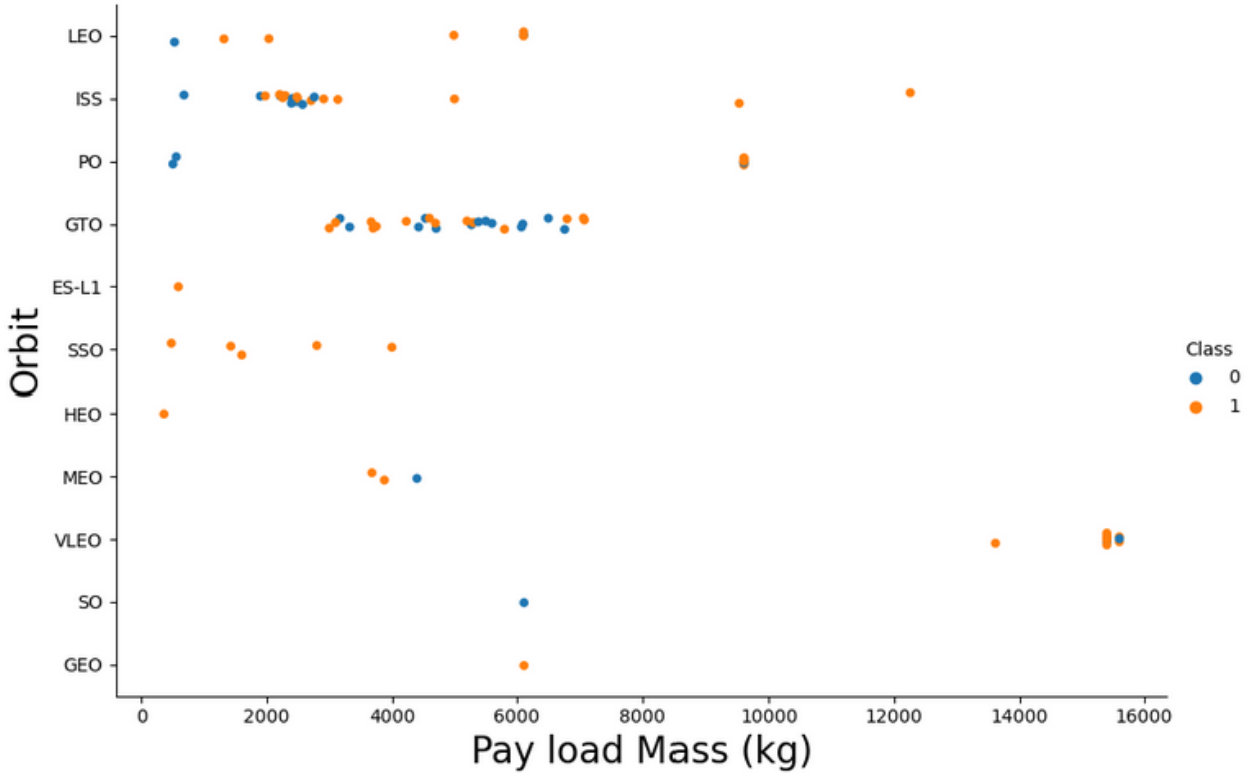
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.



Payload vs. Orbit Type

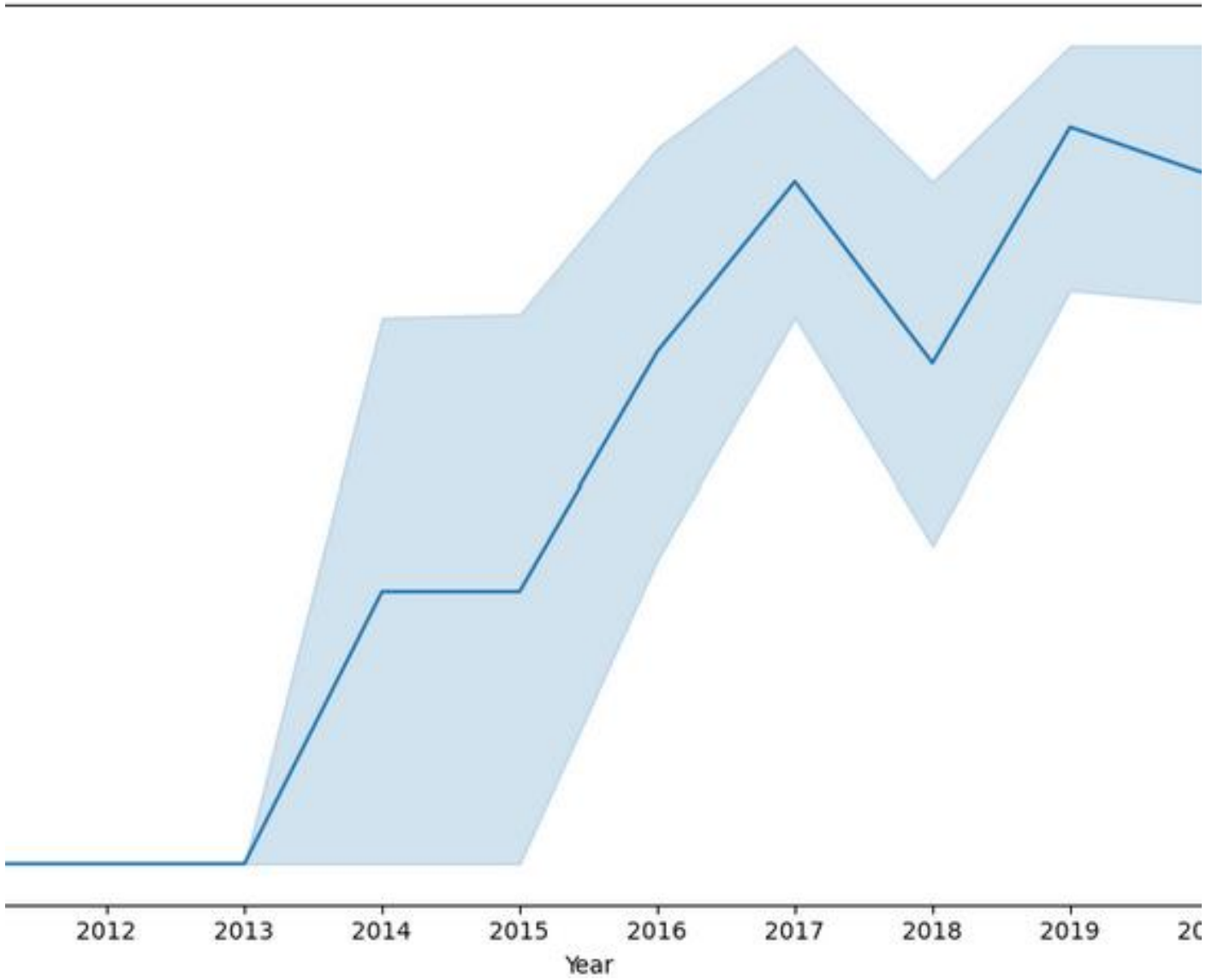
With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.

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

You can observe that the success rate since 2013 kept increasing till 2020



All Launch Site Names

```
In [8]: %sql SELECT DISTINCT LAUNCH_SITE as "Launch_Sites" FROM SPACEXTBL;
 * sqlite:///my_data1.db
Done.
```

Out[8]:

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

- Using the DISTINCT function of SQL in python to generate all the names unique Launch Site.
- All the names of Launch Sites have been shown in the plot.

Launch Site Names Begin with 'CCA'

In [9]:
```
%sql SELECT * FROM 'SPACEXTBL' WHERE Launch_Site LIKE 'CCA%' LIMIT 5;
```

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

Out[9]:

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing _Outcome
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
08-12-2010	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)
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

5 records where launch sites begin with `CCA` are as shown in the plot.

Total Payload Mass

```
In [10]: %sql SELECT SUM(PAYLOAD_MASS__KG_) as "Total Payload Mass(Kgs)", Customer FROM 'SPACEXTBL' WHERE Customer = 'NASA (CRS)';
 * sqlite:///my_data1.db
Done.
```

Out[10]:

Total Payload Mass(Kgs)	Customer
45596	NASA (CRS)

- Calculate the total payload carried by boosters from NASA
- The total payload mass is 45596kg

Average Payload Mass by F9 v1.1

In [11]:
```
%sql SELECT AVG(PAYLOAD_MASS__KG_) as "Payload Mass Kgs", Customer, Booster_Version FROM 'SPACEXTBL' WHERE Booster_Version LIKE 'F9 v1.1%';
```

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

Out[11]:

Payload Mass Kgs	Customer	Booster_Version
2534.6666666666665	MDA	F9 v1.1 B1003

- Calculate the average payload mass carried by booster version F9 v1.1
- The average payload mass by F9 v1.1 is 2534.67kg

First Successful Ground Landing Date

```
In [12]: %sql SELECT MIN(DATE) FROM 'SPACEXTBL' WHERE "Landing _Outcome" = "Success (ground pad)";
 * sqlite:///my_data1.db
Done.
Out[12]: MIN(DATE)
         01-05-2017
```

- Find the dates of the first successful landing outcome on ground pad
- The date of first successful ground landing is 01-04-2017

Successful Drone Ship Landing with Payload between 4000 and 6000

In [13]:
```
%sql SELECT DISTINCT Booster_Version, Payload FROM SPACEXTBL WHERE "Landing _Outcome" = "Success (drone ship)" AND PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ BETWEEN 4
```

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

Out[13]:

Booster_Version	Payload
F9 FT B1022	JCSAT-14
F9 FT B1026	JCSAT-16
F9 FT B1021.2	SES-10
F9 FT B1031.2	SES-11 / EchoStar 105

- Names of booster which executed successful drone ship landing with payload mass greater than 4000kg and less than 6000kg are as shown in the plot above.

Total Number of Successful and Failure Mission Outcomes

```
In [14]: %sql SELECT "Mission_Outcome", COUNT("Mission_Outcome") as Total FROM SPACEXTBL GROUP BY "Mission_Outcome";
 * sqlite:///my_data1.db
Done.
```

Out[14]:

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

- Calculate the total number of successful and failure mission outcomes
- You can see that there are about 98% in total success mission outcomes and about 1 % failure outcome

Boosters Carried Maximum Payload

In [16]:
```
%sql SELECT "Booster_Version",Payload, "PAYLOAD_MASS__KG_" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL);
```

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

Out[16]:

Booster_Version	Payload	PAYLOAD_MASS__KG_
F9 B5 B1048.4	Starlink 1 v1.0, SpaceX CRS-19	15600
F9 B5 B1049.4	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600
F9 B5 B1051.3	Starlink 3 v1.0, Starlink 4 v1.0	15600
F9 B5 B1056.4	Starlink 4 v1.0, SpaceX CRS-20	15600
F9 B5 B1048.5	Starlink 5 v1.0, Starlink 6 v1.0	15600
F9 B5 B1051.4	Starlink 6 v1.0, Crew Dragon Demo-2	15600
F9 B5 B1049.5	Starlink 7 v1.0, Starlink 8 v1.0	15600
F9 B5 B1060.2	Starlink 11 v1.0, Starlink 12 v1.0	15600
F9 B5 B1058.3	Starlink 12 v1.0, Starlink 13 v1.0	15600
F9 B5 B1051.6	Starlink 13 v1.0, Starlink 14 v1.0	15600
F9 B5 B1060.3	Starlink 14 v1.0, GPS III-04	15600
F9 B5 B1049.7	Starlink 15 v1.0, SpaceX CRS-21	15600

- List the names of the booster which have carried the maximum payload mass
- Names of boosters with the maximum payload have been shown in the plot.

2015 Launch Records

In [39]:
```
%sql SELECT substr(Date,7,4), substr(Date, 4, 2),"Booster_Version", "Launch_Site", Payload, "PAYLOAD_MASS__KG_", "Mission_Outcome", "Landing _Outcome" FROM SPACEXTBL where
```

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

Out[39]:

substr(Date,7,4)	substr(Date, 4, 2)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Mission_Outcome	Landing _Outcome
2015	01	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	Success	Failure (drone ship)
2015	02	F9 v1.1 B1013	CCAFS LC-40	DSCOVR	570	Success	Controlled (ocean)
2015	03	F9 v1.1 B1014	CCAFS LC-40	ABS-3A Eutelsat 115 West B	4159	Success	No attempt
2015	04	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	Success	Failure (drone ship)
2015	04	F9 v1.1 B1016	CCAFS LC-40	Turkmen 52 / MonacoSAT	4707	Success	No attempt
2015	06	F9 v1.1 B1018	CCAFS LC-40	SpaceX CRS-7	1952	Failure (in flight)	Precluded (drone ship)
2015	12	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	Success	Success (ground pad)

You can see that in 2015, Launch Site CCAFS LC-40 with Booster_Versions of F9 v1.1 B1012 & B1015 had failure(drone ship) landing outcomes

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

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

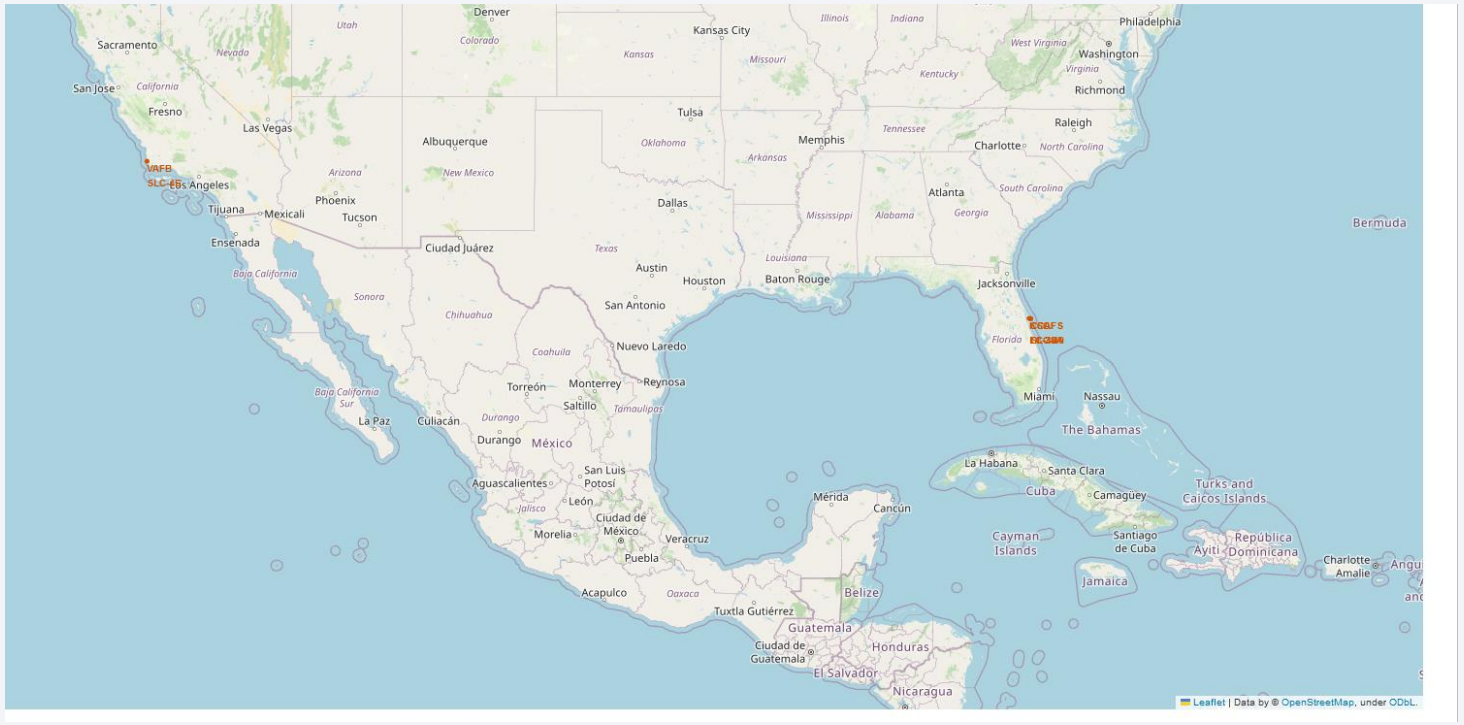
- To completely quantify the outcome, “No attempt” data must be considered as well.

Section 3

Launch Sites Proximities Analysis

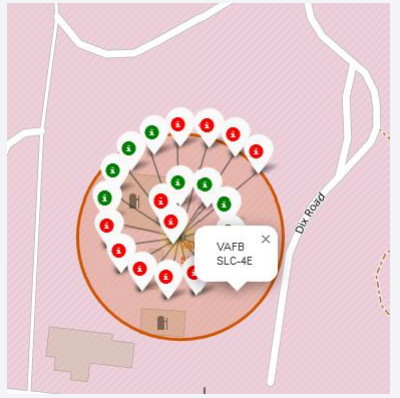
Location of all the Launch Sites

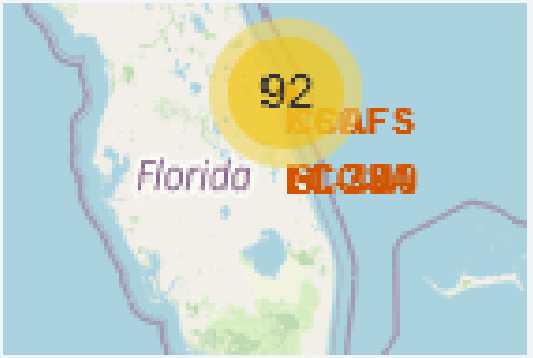
- Folium map showing all the Launch Sites.
- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map
- We can see that the Launch Sites are in the United States.

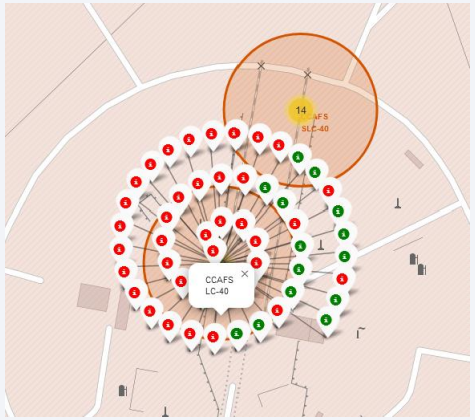


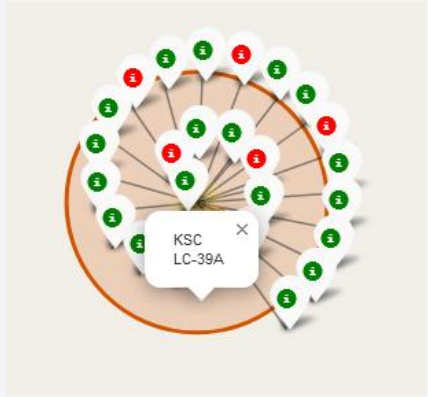
Marker showing Launch Sites with Color

- Folium map showing Launch Sites with Color
- There are one Launch site in California and 3 Launch Site in Florida. Each launch site in green marker signifies a successful launch and red marker shows a failed launch.



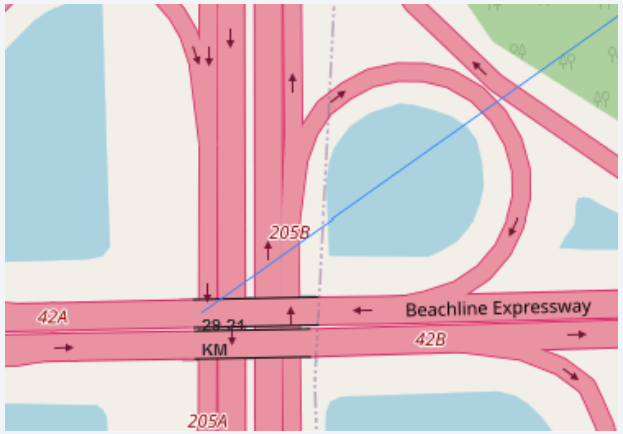


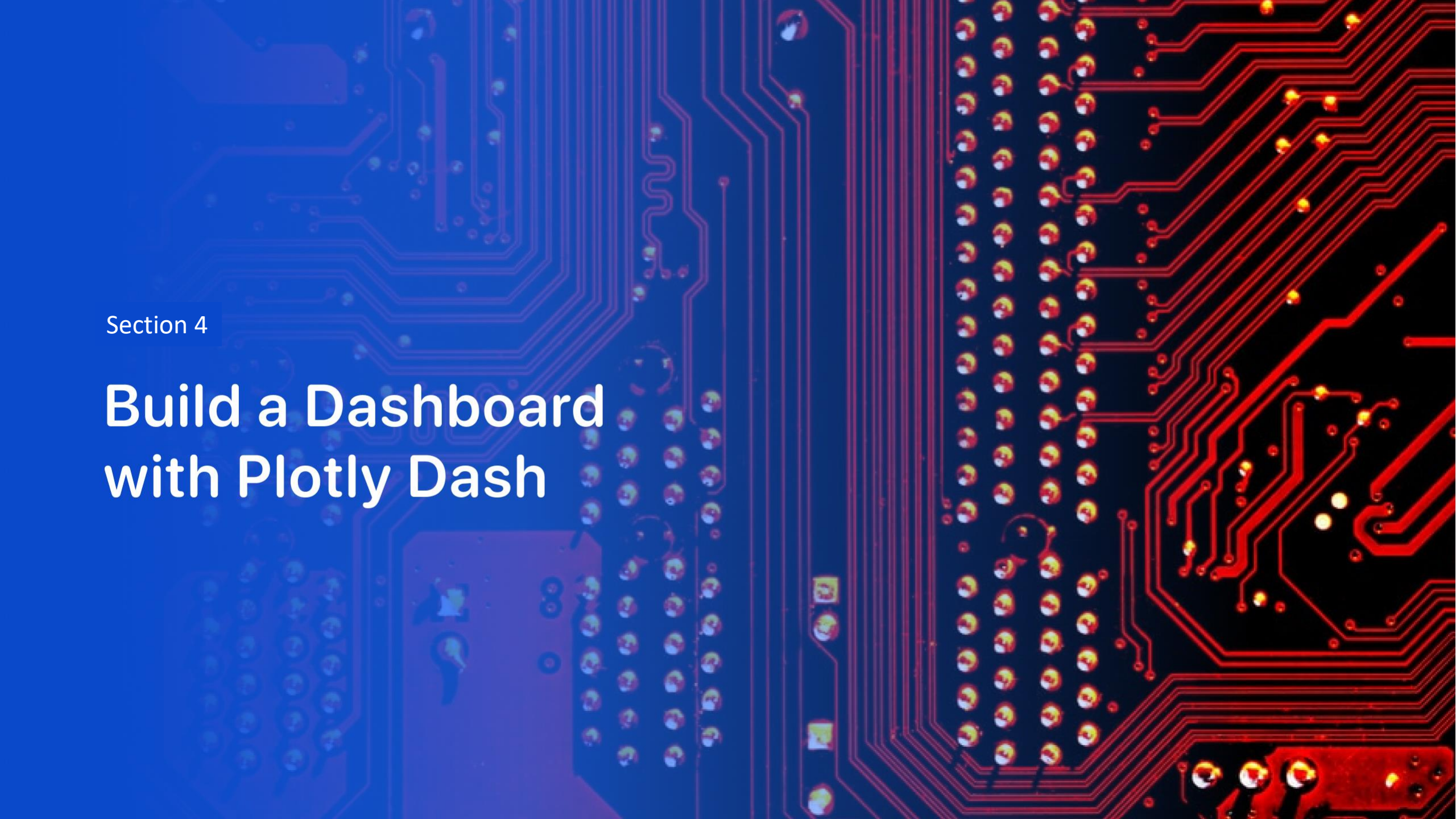




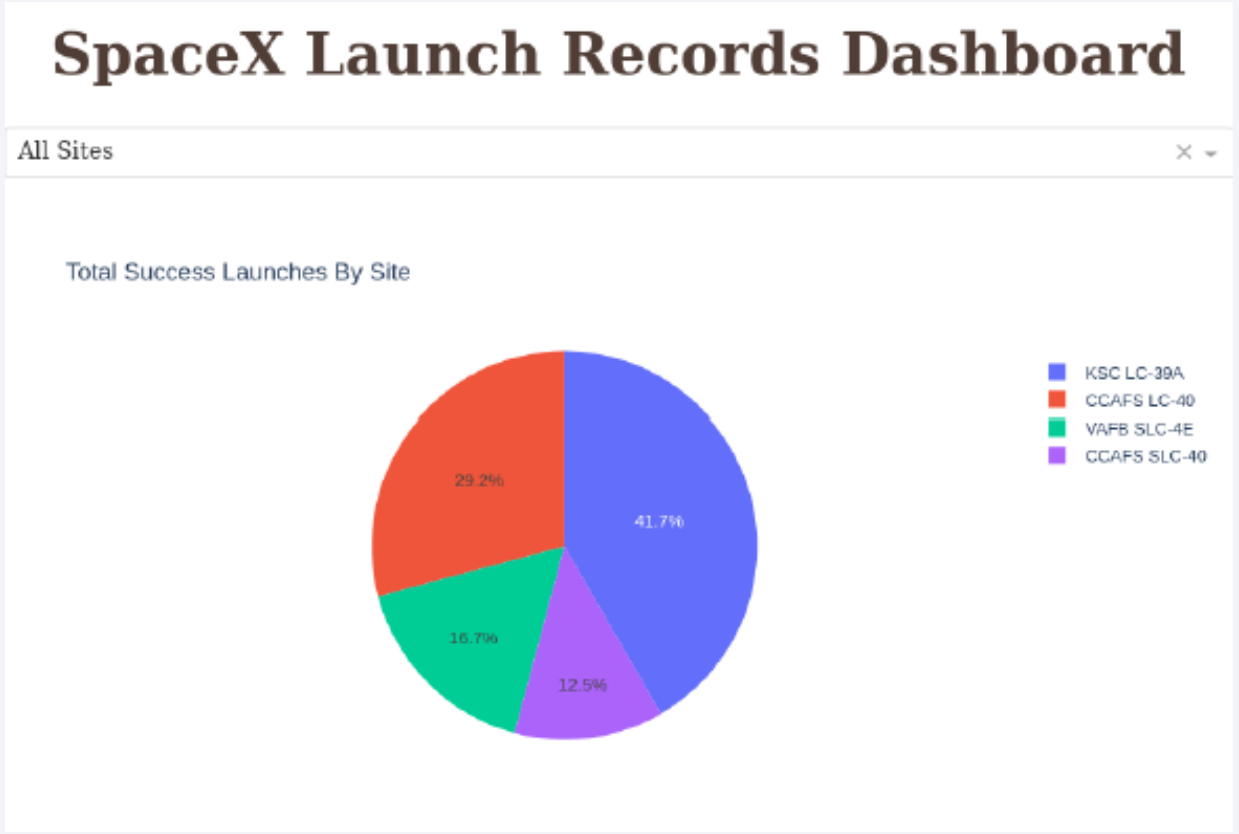
Launch Sites Distance to Landmarks

- The proximities of these Launch sites show no close proximity to railways. However, there are about 29km proximity from Launch site CCAFS SLC-40 to Highway and about 79km to Florida City. The Coastline proximity is about 5km from CCAFS-LC-40



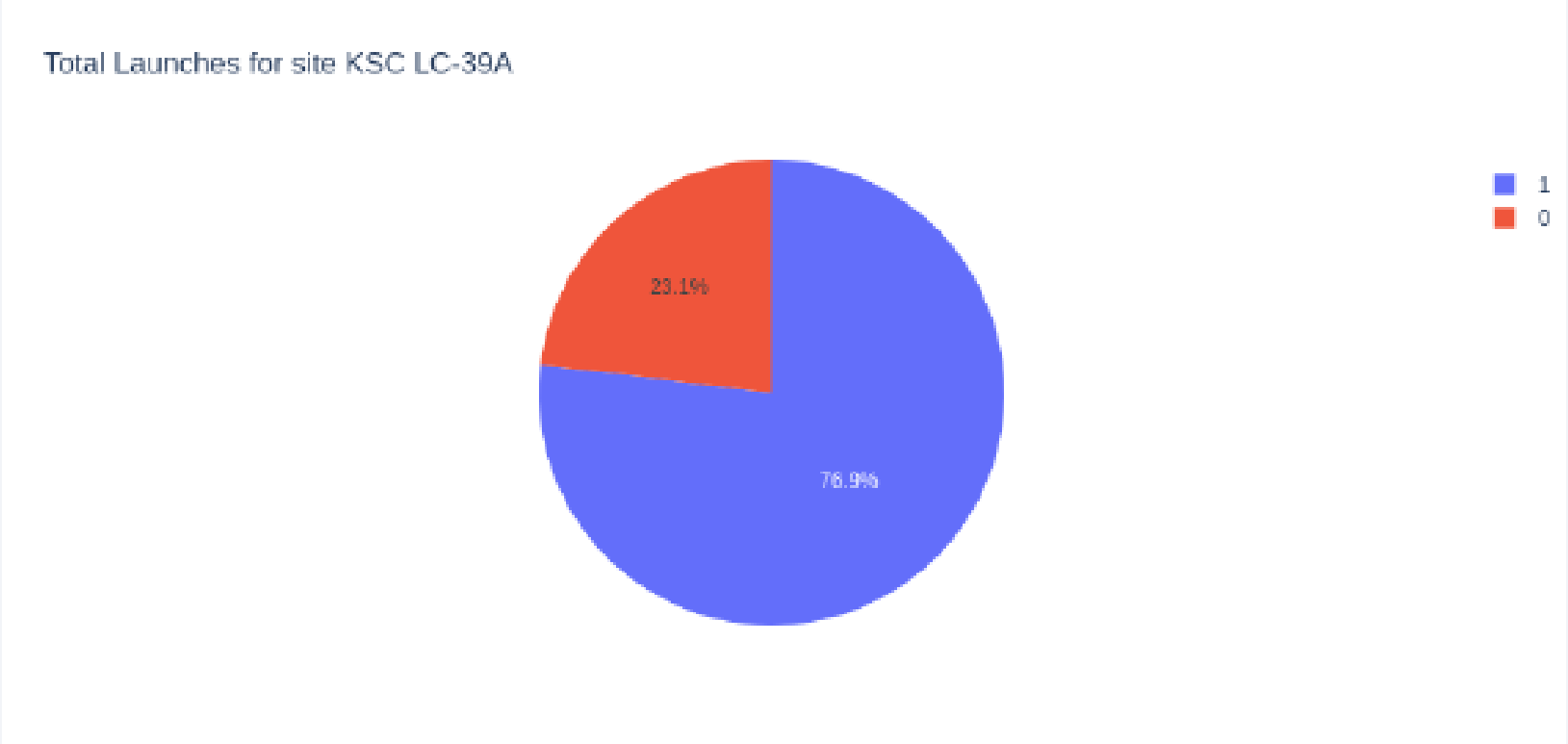

Section 4
Build a Dashboard with Plotly Dash

Successful Launches by Site



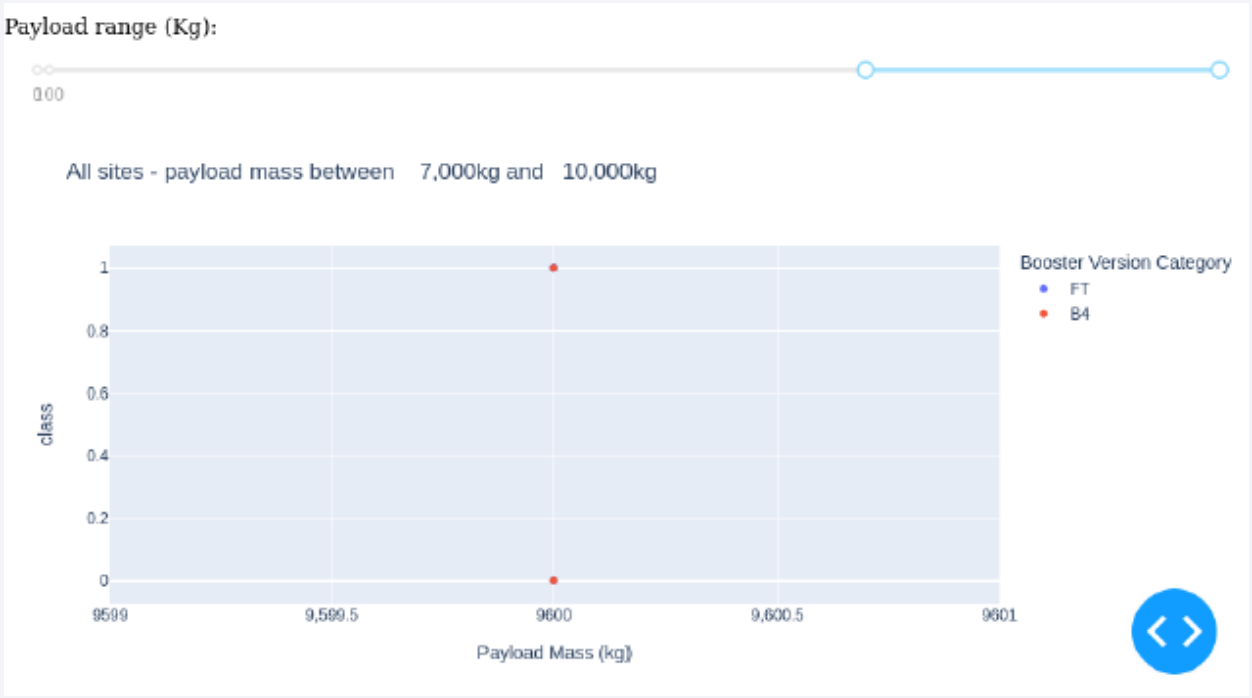
- Launch locations play a vital role in the success of a mission

The Highest Launch-Success Ratio: KSC LC-39A



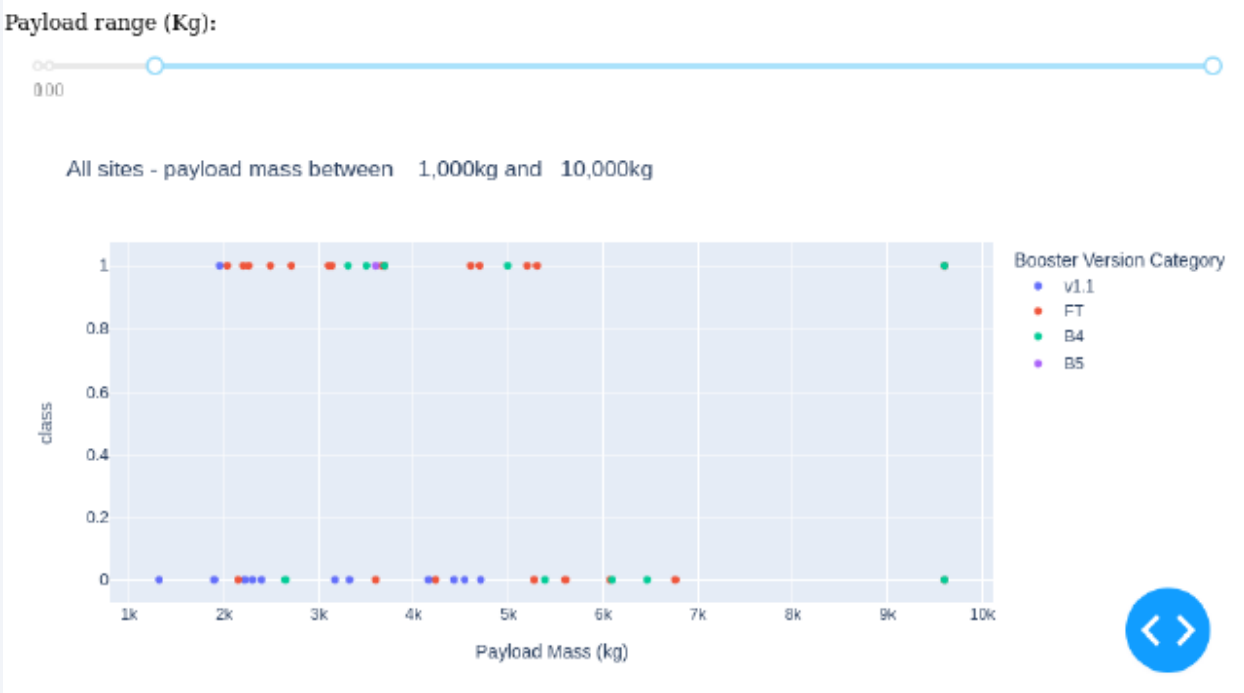
- 76.9% of Launches are successful in this site with KSC-LC-39A being the highest.

Payload VS Launch Outcome Scatter Plot



- There are limited data to estimate risk of launches over 7,000kg

Payload VS Launch Outcome Scatter Plot



- Payloads that are under 6000kg and FT boosters are the most successful combination

Section 5

Predictive Analysis (Classification)

Classification Accuracy

In [46]:

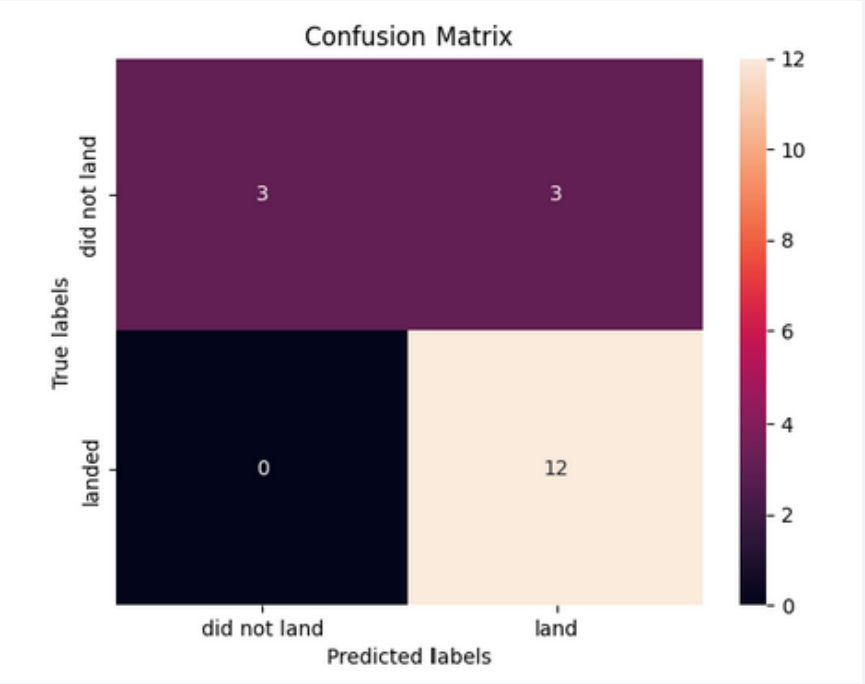
```
print('Accuracy for Logistics Regression method:', logreg_cv.score(X_test, Y_test))
print( 'Accuracy for Support Vector Machine method:', svm_cv.score(X_test, Y_test))
print('Accuracy for Decision tree method:', tree_cv.score(X_test, Y_test))
print('Accuracy for K nearsdt neighbors method:', knn_cv.score(X_test, Y_test))
```

```
Accuracy for Logistics Regression method: 0.8333333333333334
Accuracy for Support Vector Machine method: 0.8333333333333334
Accuracy for Decision tree method: 0.7222222222222222
Accuracy for K nearsdt neighbors method: 0.8333333333333334
```

- Looking at the results of the above charts, we could identify that the best algorithm to be Logistic Regression, Support Vector and K-Nearest Neighbors Model all have the same highest classification accuracy.

Confusion Matrix

Confusion matrix of Logistic Regression, Support Vector and K-Nearest Neighbors Model proves its accuracy by showing the big numbers of true positive and true negative compared to the false ones.



Conclusions

We can conclude that:

- The Logistic Regression, Support Vector and K-Nearest Neighbors Model are the best Machine Learning approach for this dataset.
- The low weighted payloads (which define as 4000kg and below) performed better than the heavy weighted payloads.
- Starting from the year 2013, the success rate for SpaceX launches is increased, directly proportional time in years to 2020, which it will eventually perfect the launches in the future.
- KSC LC-39A have the most successful launches of any sites; 76.9%
- SSO orbit have the most success rate; 100% and more than 1 occurrence.

Thank you!