IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Kharamani, Reza

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection and Data Wrangling
  - Exploratory Data Analysis (EDA) with data visualization and SQL
  - Analysis of the Launch Site with Map (Folium)
  - Data Visualisation Dashboard (Plotly and Dash)
  - Predictive Analysis using Classification model
- Summary of all results
  - Analysis result on various EDAs
  - Screenshots of various charts and visuals
  - Results on Predictive Analysis

Introduction

- Project background and context

  In this report, we will predict if the Falcon 9 first stage will land successfully. SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore, if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch. In this lab, you will collect and make sure the data is in the correct format from an API. The following is an example of a successful and launch.

- Problems you want to find answers
  - Determine the key variables that correlates the success rate of successful landing.
  - What are the key determinants to ensure the best success rate of successful landing.

Section 1

Methodology

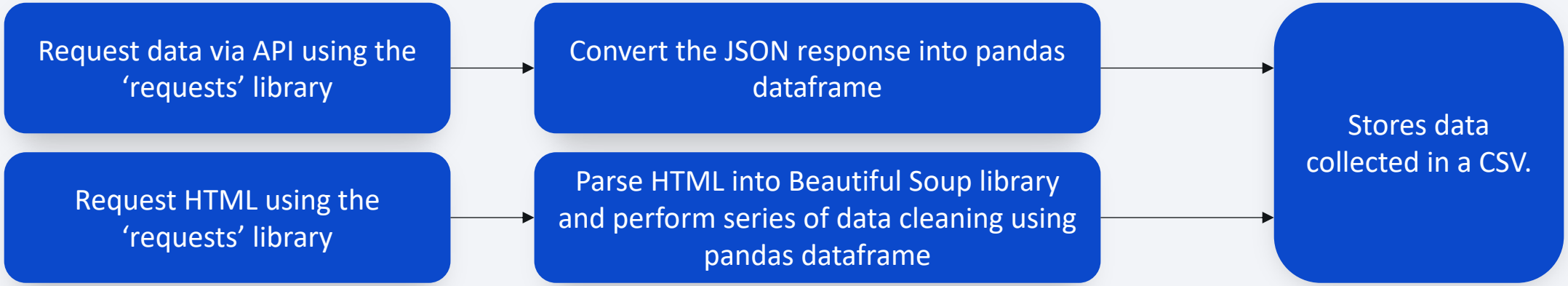
Methodology

Executive Summary

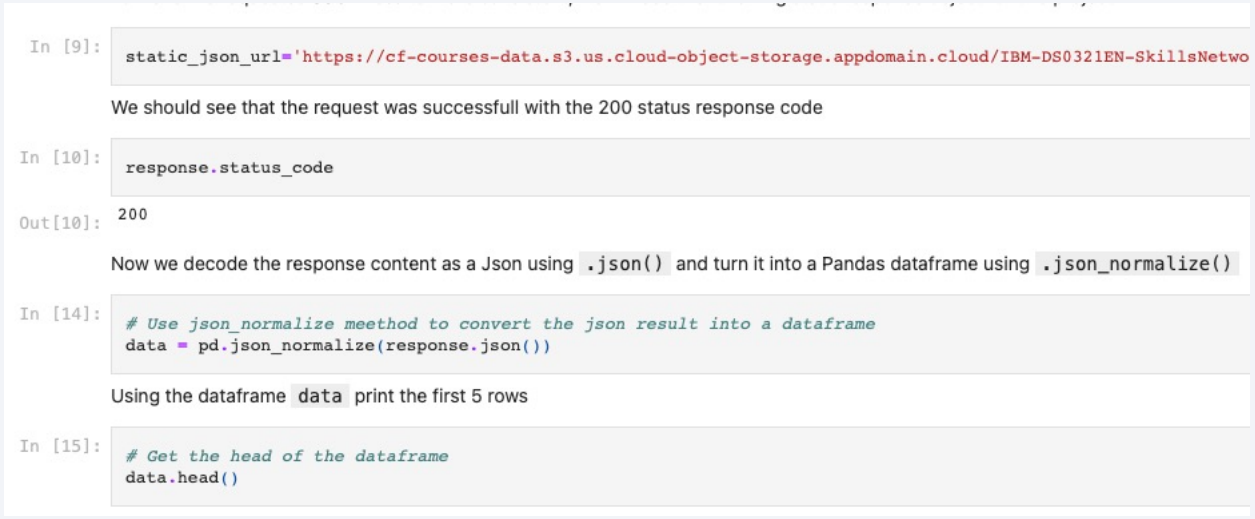
- Data collection methodology:
  - Data is collected via API method from SpaceX official APIs and web scraping from an article in Wikipedia.
- Perform data wrangling
  - Includes dropping irrelevant columns and converting certain attributes via one-hot-encoding method. The data is further refined by obtaining rocket information.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Model evaluation to determine the best possible accuracy.

Data Collection

- Two techniques are used for data gathering – API and Web Scraping.
  - Using the SpaceX Official APIs, we are able to obtain the relevant data such as information about the launches, the rocket which is used and landing outcomes.
  - Using web scraping library, Beautiful Soup, we are able to obtain relevant data such as information about the launches, the rocket which is used and landing outcomes.



Data Collection – SpaceX API



In [9]:
```
static_json_url='https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwo
```

We should see that the request was successfull with the 200 status response code

In [10]:
```
response.status_code
```

Out[10]:
```
200
```

Now we decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()`

In [14]:
```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

Using the dataframe `data` print the first 5 rows

In [15]:
```
# Get the head of the dataframe
data.head()
```

Request data via API using the 'requests' library

↓

Convert the JSON response into pandas dataframe

↓

Stores data collected in a CSV.

Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%201%20Collecting%20Data.ipynb

Data Collection - Scraping

In [11]:
```
# use requests.get() method with the provided static_url
# assign the response to a object
response = requests.get(static_url)
```

Create a `BeautifulSoup` object from the HTML `response`

In [12]:
```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.content, 'html.parser')
```

Print the page title to verify if the `BeautifulSoup` object was created properly

In [13]:
```
# Use soup.title attribute
print(soup.title)
```

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Request HTML using the 'requests' library

↓

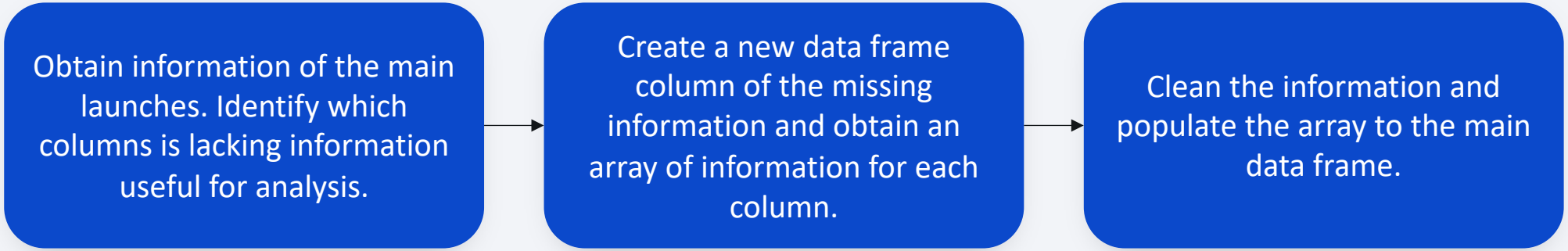
Parse HTML into Beautiful Soup library and perform series of data cleaning using pandas dataframe

↓

Stores data collected in a CSV.

Data Wrangling

- Joining other information to obtain more information
  - We noticed that a lot of the data were presented as IDs, such as rocket, payloads, launchpad and cores information. To obtain those information, we required to call another API and obtain more information via the API and join these information.



Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%201%20Collecting%20Data.ipynb

EDA with Data Visualization

- Scatter Plots
  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit Type
  - Payload vs Orbit Type
- Bar Chart
  - Success Rate of each Orbit Type
- Line Chart
  - Trend of Success Rate over Time (by Year)

Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%202%20%20the%20EDA%20with%20Visualization.ipynb

EDA with SQL

- Understanding data from SQL queries
  - Names of unique launch sites.
  - First 5 records of launch sites which named starting with 'CCA'.
  - Total payload mass carried by boosters lunached by NASA (CRS).
  - Average payload mass carried by booster version F9 v1.1.
  - Date of first successful landing outcome on ground pad.
  - Names of boosters which have success in drone ship landing with payload mass greater than 4000 but less than 6000.
  - Total number of successful and failure mission outcomes.
  - List of names of Booster Versions which have carried the maximum payload mass.
  - List of months which failure landing outcomes and year is 2015.
  - Number of successful landing outcomes between date 04/06/2010 and 20/03/2017.

Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%202%20EDA%20with%20SQL.ipynb

Build an Interactive Map with Folium

- To understand the launch data a bit better, we take the latitude and longitude of each launch sites and add circle markers with the name label of the launch site.
- Launch Outcomes are converted via one-hot-encoding and labeled its corresponding 0 and 1 value with red and green color (0 means fail).
- Haversine's formula is used to calculate the distances from launch sites to various landmarks.

Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%203%20Interactive%20Visual%20Analytics%20with%20Folium.ipynb

Build a Dashboard with Plotly Dash

- A pie chart shows the total launches by selecting a certain site or all sites.
  - This gives insight to the proportion of success vs failures.
- A scatter graph shows the relationship between the Outcome and Payload Mass for each boosters.
  - This gives insight to this relationship.

Github:
https://github.com/reza1372/FinalAssignment/blob/main/Week_5_SPACEX.py

Predictive Analysis (Classification)

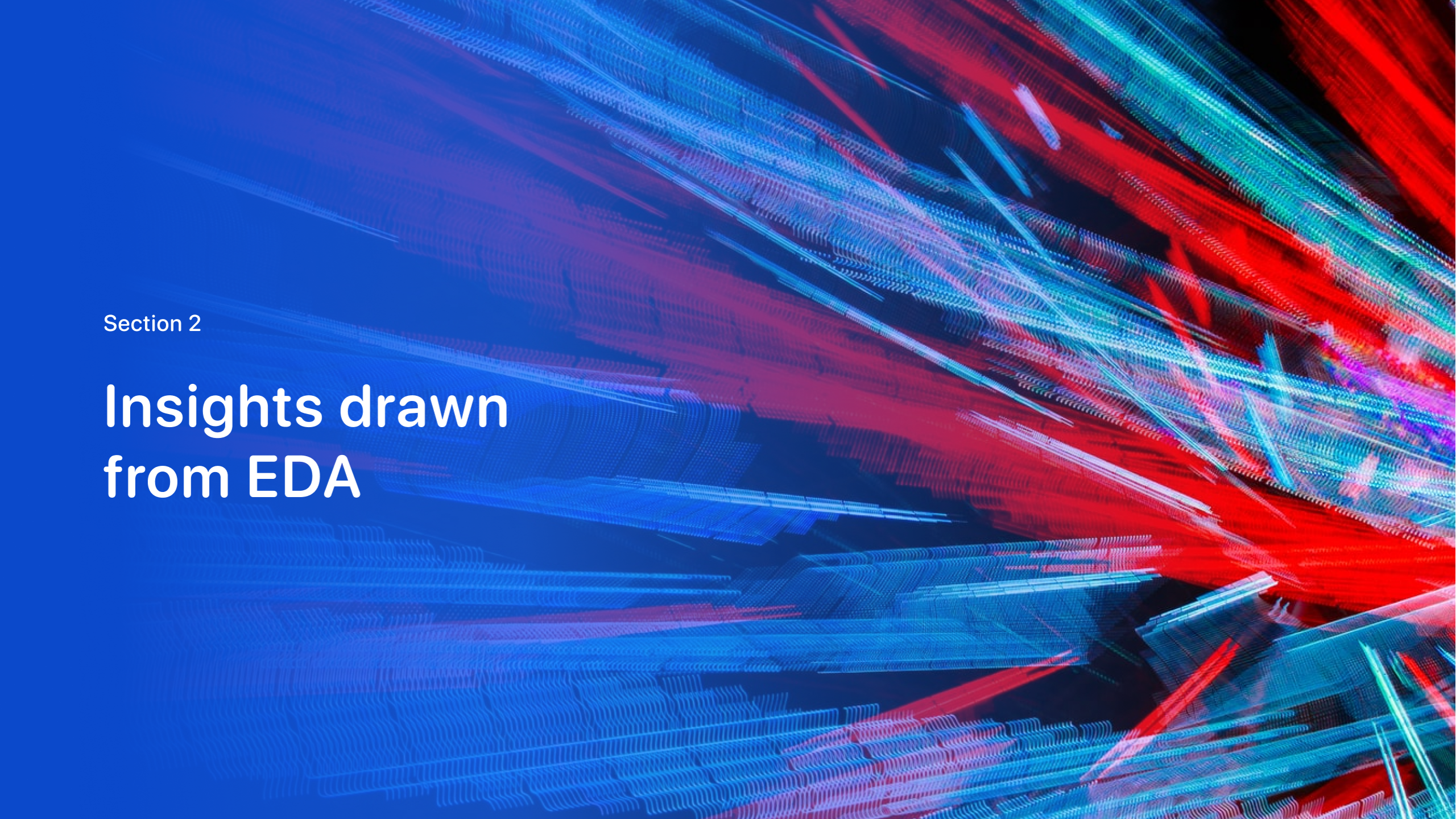
- Building the Model
  - Load dataset using Numpy and Pandas library
  - Transform data
  - Split data into training and test sets.
  - Experiment with different types of machine learning model
  - Fit dataset to GridSearchCV objects and train the dataset.
- Evaluating the Model
  - Determine the accuracy for each model
  - Plot the confusion matrix
- Improving the Model
  - Feature Engineering
  - Algorithm Tuning
- Decide the best performing model
  - Compare accuracy scores between each model and decide on the best performing model.

Github:

https://github.com/reza1372/FinalAssignment/blob/main/Week%204%20the%20Machine%20Learning%20Prediction%20lab.ipynb

Results

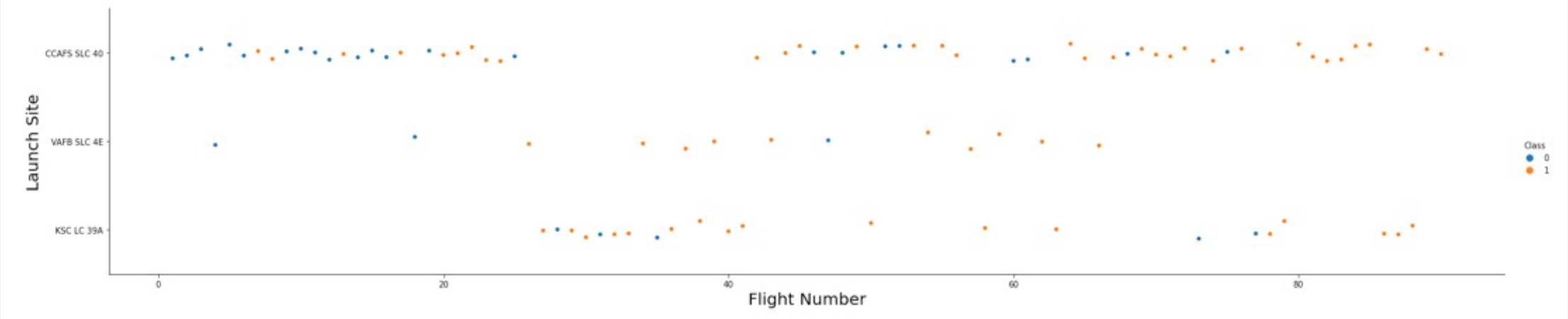
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results



Section 2

Insights drawn from EDA

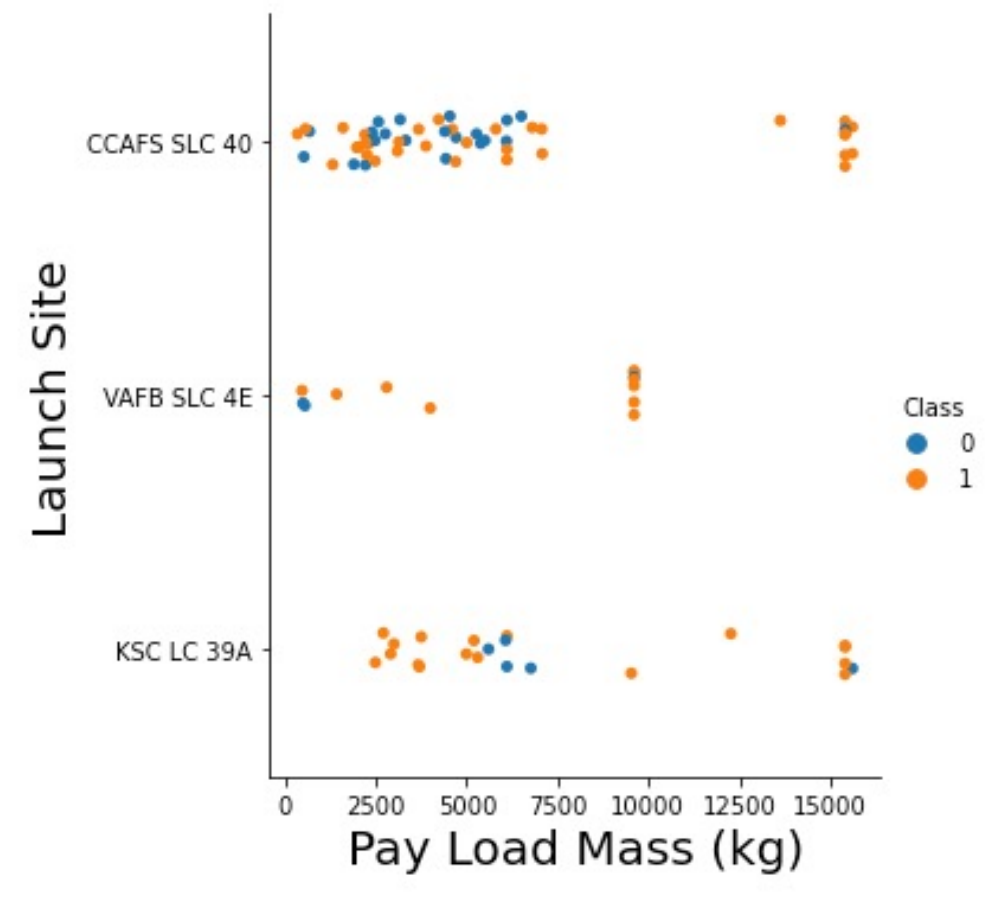
Flight Number vs. Launch Site



Launch site is found to have no concrete correlation against successful outcome.

It is however, suggests that there may be other variables impacting successful outcome as failed outcome becomes more unlikely over time.

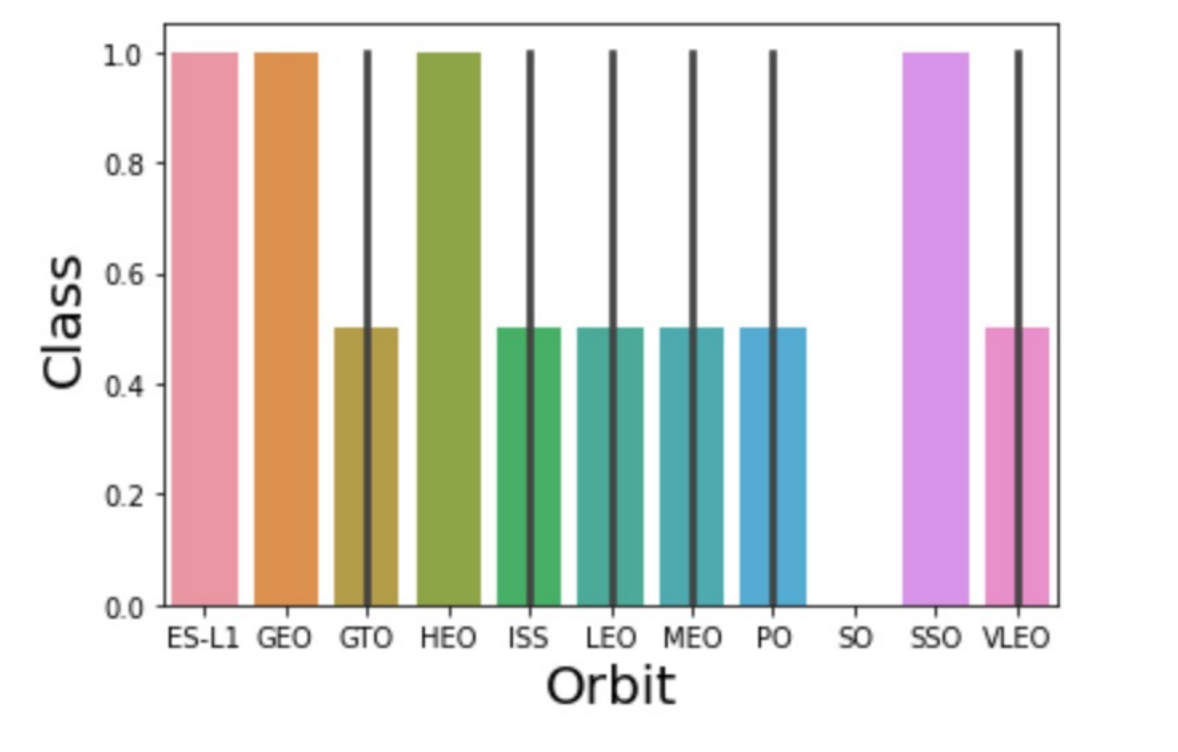
Payload vs. Launch Site



We found that VAFB-SLC launch site has no rocket launch for payload mass greater than 10000 KG.

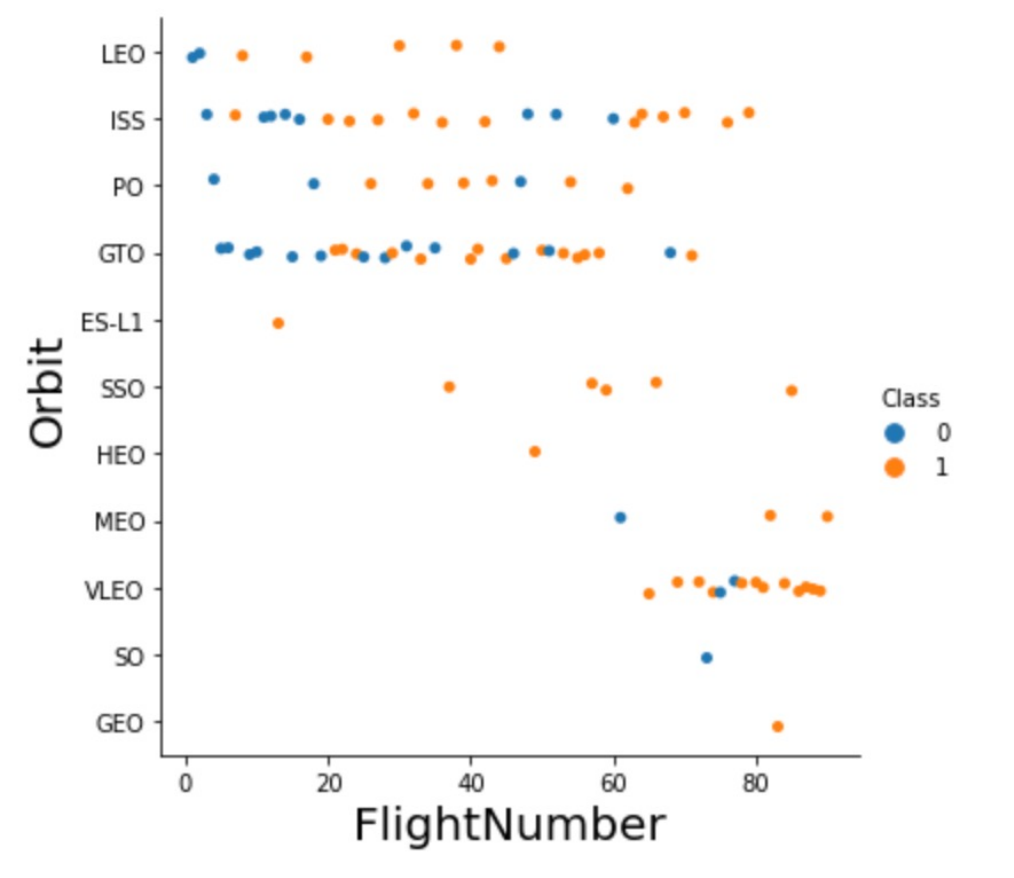
It also suggests that the higher the payload, the higher the success rates.

Success Rate vs. Orbit Type



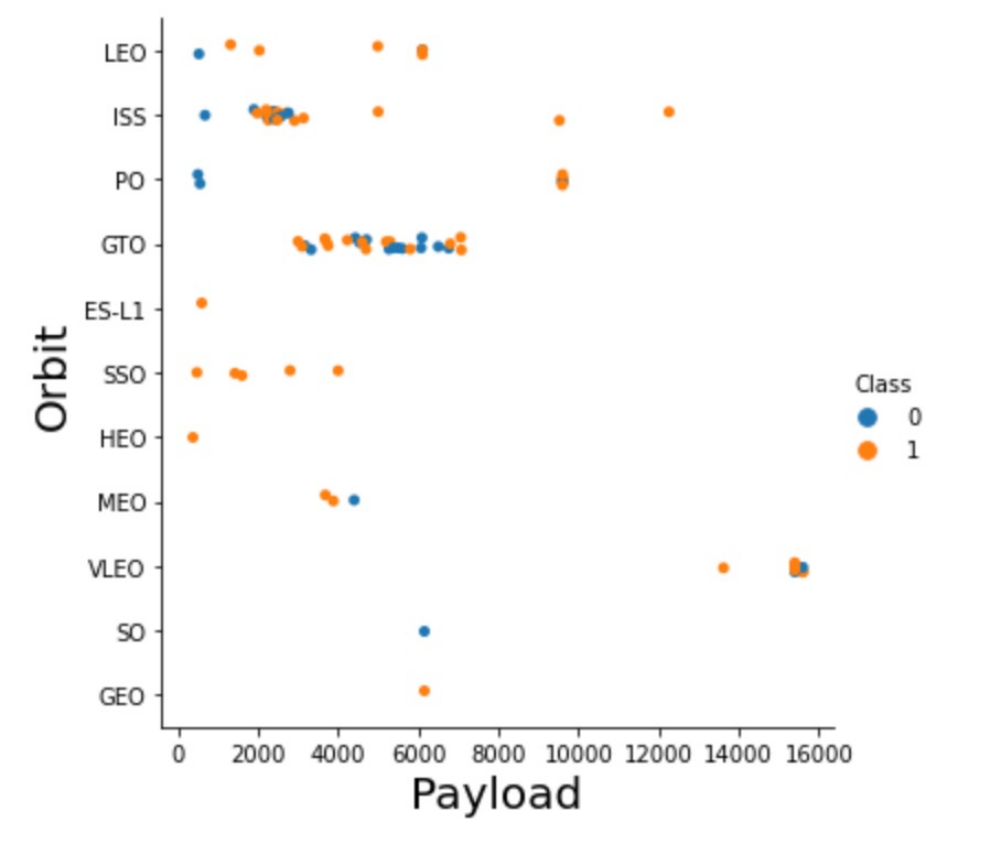
It suggests that Orbit ES-l1, GEO, HEO and SSO has 100% success rate.

Flight Number vs. Orbit Type



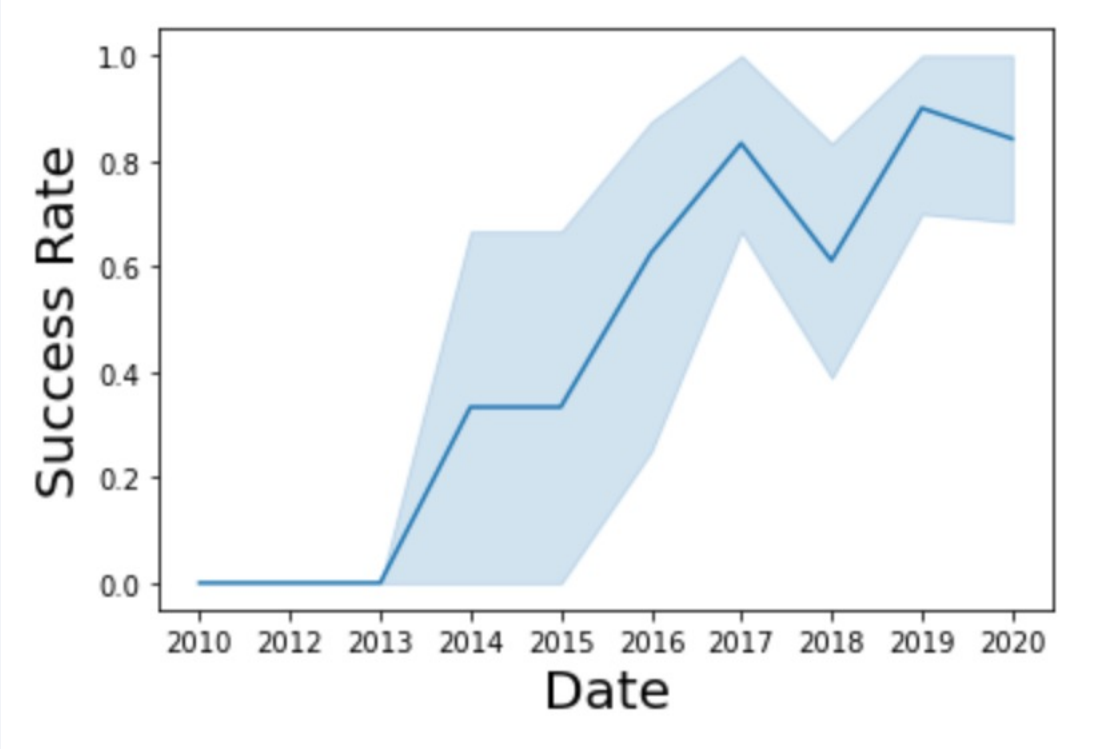
It suggests that the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

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



It is shown that the success rate since 2013 kept increasing till 2020

All Launch Site Names

- Names of the unique launch sites
  - CCAFS LC-40
  - VAFB SLC-4E
  - KSC LC-39A
  - CCAFS SLC-40

```
In [67]: %sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
 * sqlite:///my_data1.db
Done.
Out[67]:
```

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

The DISTINCT keyword allow us to obtain unique values.

Launch Site Names Begin with 'CCA'

In [16]:
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5
```

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

Out[16]:

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

Using keyword LIMIT 5 limits shows only 5 records.

Total Payload Mass

```
In [19]: %sql SELECT SUM(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Customer='NASA (CRS)'
 * sqlite:///my_data1.db
Done.
Out[19]: SUM(PAYLOAD_MASS__KG_)
                          45596
```

Using keyword SUM() outputs the sum of the selected column.

Average Payload Mass by F9 v1.1

```
In [20]: %sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version='F9 v1.1'
 * sqlite:///my_data1.db
Done.
Out[20]: AVG(PAYLOAD_MASS__KG_)
         2928.4
```

Using keyword AVG() outputs the average of the selected column.

First Successful Ground Landing Date

```
In [39]: %sql SELECT MIN(Date) FROM SPACEXTBL WHERE Landing_Outcome='Success (ground pad)'
 * sqlite:///my_data1.db
Done.
Out[39]: MIN(Date)
         01-05-2017
```

Using keyword MIN() outputs the minimum value, in this case it outputs the first date.

Successful Drone Ship Landing with Payload between 4000 and 6000

In [35]:
```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_ > 4000 AND PAYLOAD_MASS__KG_ < 6000
```

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

Out[35]:

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 FT B1032.2

Using keyword AND in WHERE sets multiple conditions, in this case the payload mass to be between 4000 and 6000.

Total Number of Successful and Failure Mission Outcomes

In [38]:
```
%sql SELECT Mission_Outcome, COUNT(Mission_Outcome) AS Total FROM SPACEXTBL GROUP BY Mission_Outcome
```

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

Out[38]:

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

Using keyboard GROUP BY allows the counts to be grouped by the mission outcome, producing the tally for each mission outcome values.

Boosters Carried Maximum Payload

```
In [40]: %sql SELECT Booster_Version FROM SPACEXTBL WHERE PAYLOAD_MASS__KG_=(SELECT MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL)
 * sqlite:///my_data1.db
Done.
```

Out[40]:

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

Using subquery grabs the max payload mass from the table where it can be used to set the WHERE condition to obtain the list of booster versions with the maximum payload.

2015 Launch Records

In [47]:
```
%sql SELECT SUBSTR(Date,4,2) AS Month, Landing_Outcome, Booster_Version, Launch_Site FROM SPACEXTBL WHERE SUBSTR(Date,7,4)='2015'
```

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

Out[47]:

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
02	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40
03	No attempt	F9 v1.1 B1014	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
04	No attempt	F9 v1.1 B1016	CCAFS LC-40
06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40

Using SUBSTR, we able to slice the string from the Date string to display and sets condition. Note: SUBSTR is used because sqllite does not support MONTHNAMES() and YEAR().

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

In [55]:
```
%sql SELECT Date, COUNT(Landing_Outcome) AS Total FROM SPACEXTBL WHERE Date > ('04-06-2010') AND Date < ('20-03-2017') GROUP BY Landing_Out
```

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

Out[55]:

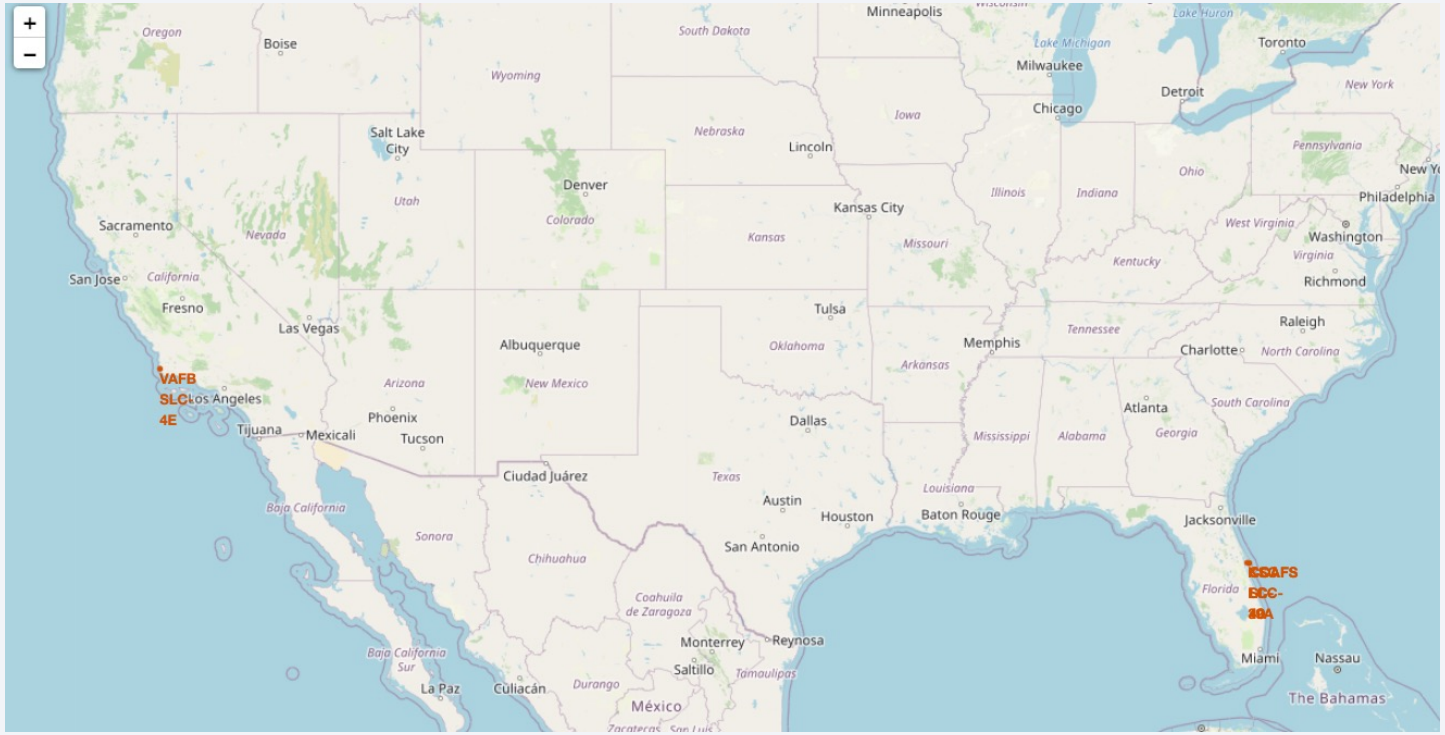
Date	Total
08-12-2010	1
06-08-2019	1
18-04-2014	3
05-12-2018	3
10-01-2015	4
18-07-2016	6
08-04-2016	8
08-10-2012	10
07-08-2018	20

Using combination of GROUP BY and WHERE conditions, we are able to obtain the outcomes between dates.

Section 4

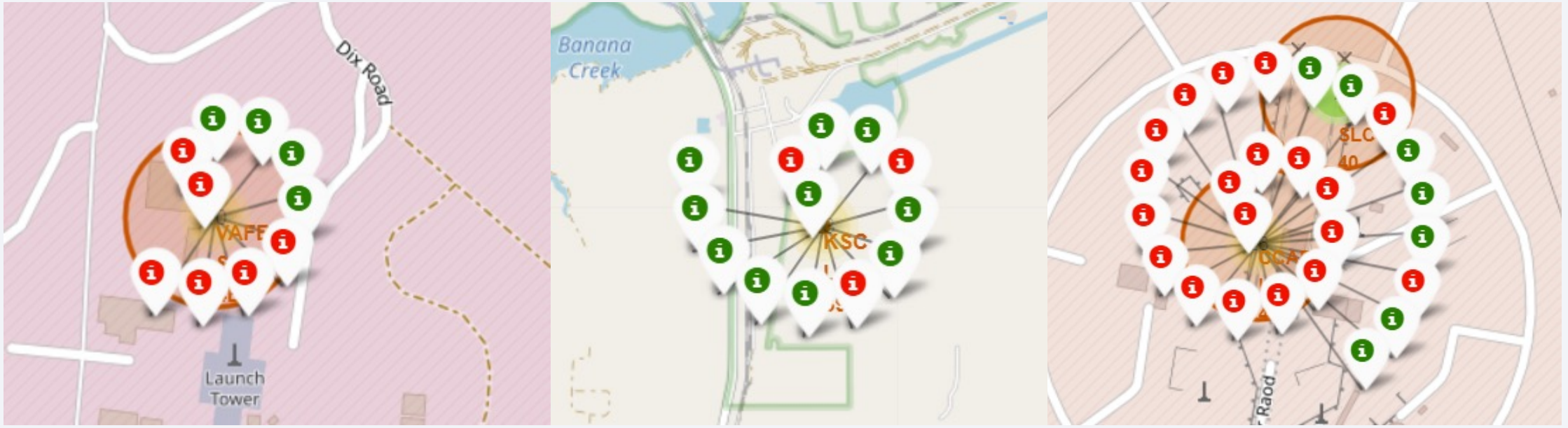
Launch Sites Proximities Analysis

<Folium Map Screenshot 1>



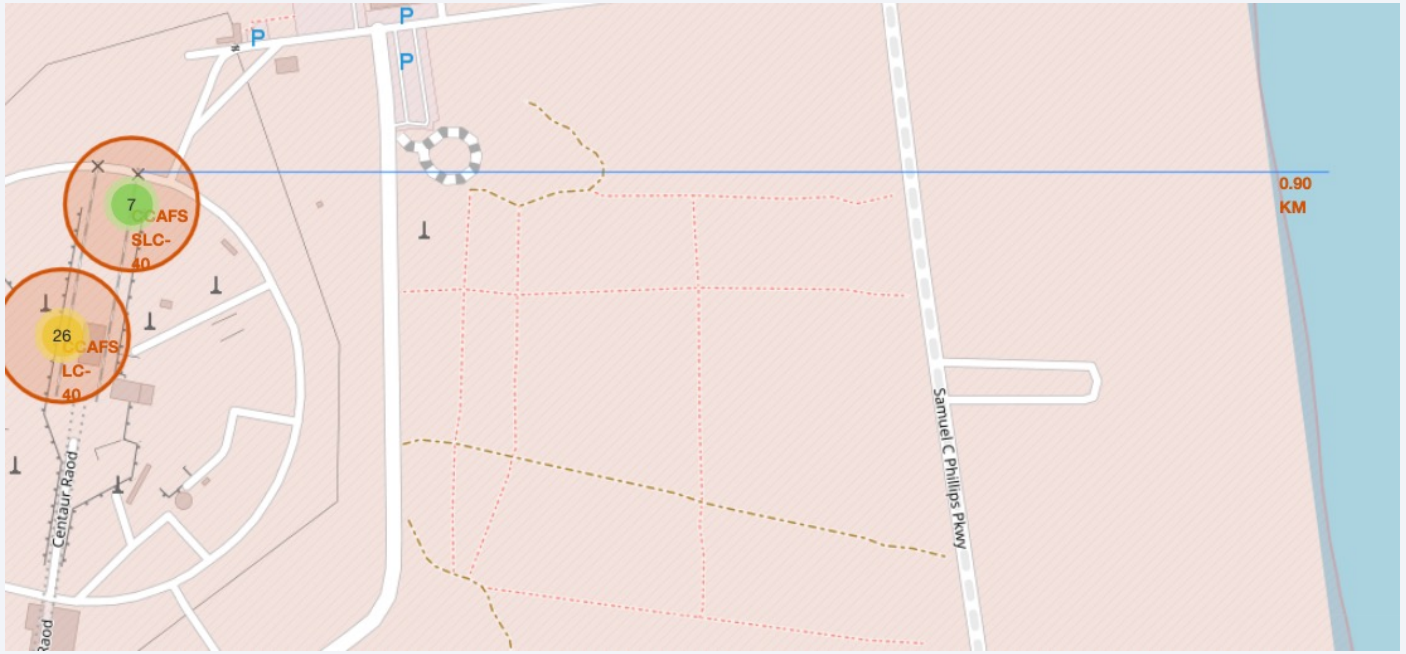
We can see that all launch sites are situated within USA, on the left and right coastal lines.

<Folium Map Screenshot 2>

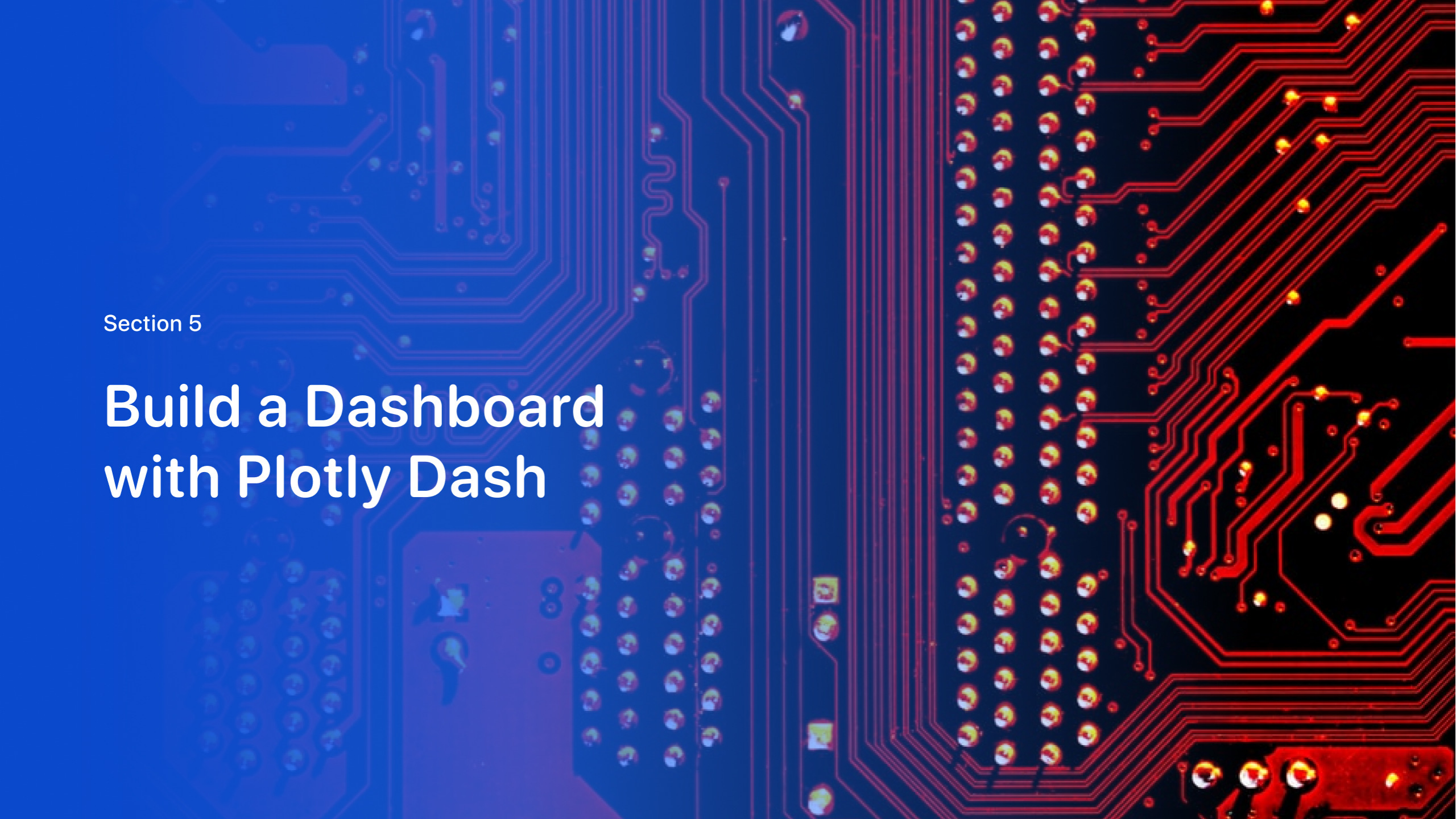


The green markers indicate successful launches, while red markers indicate failed launches.

<Folium Map Screenshot 3>



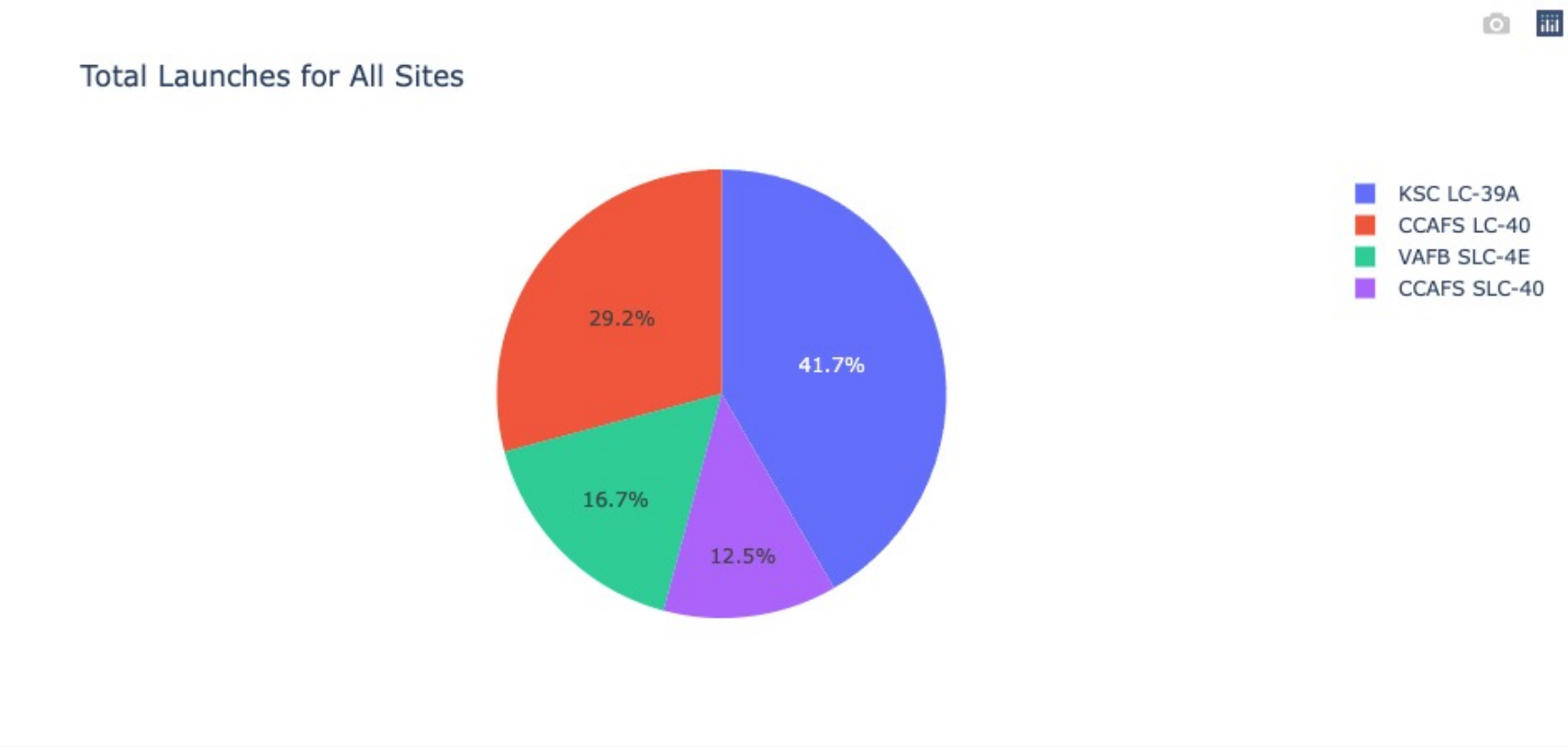
Most launch sites are situated very close proximity to coastline.



Section 5

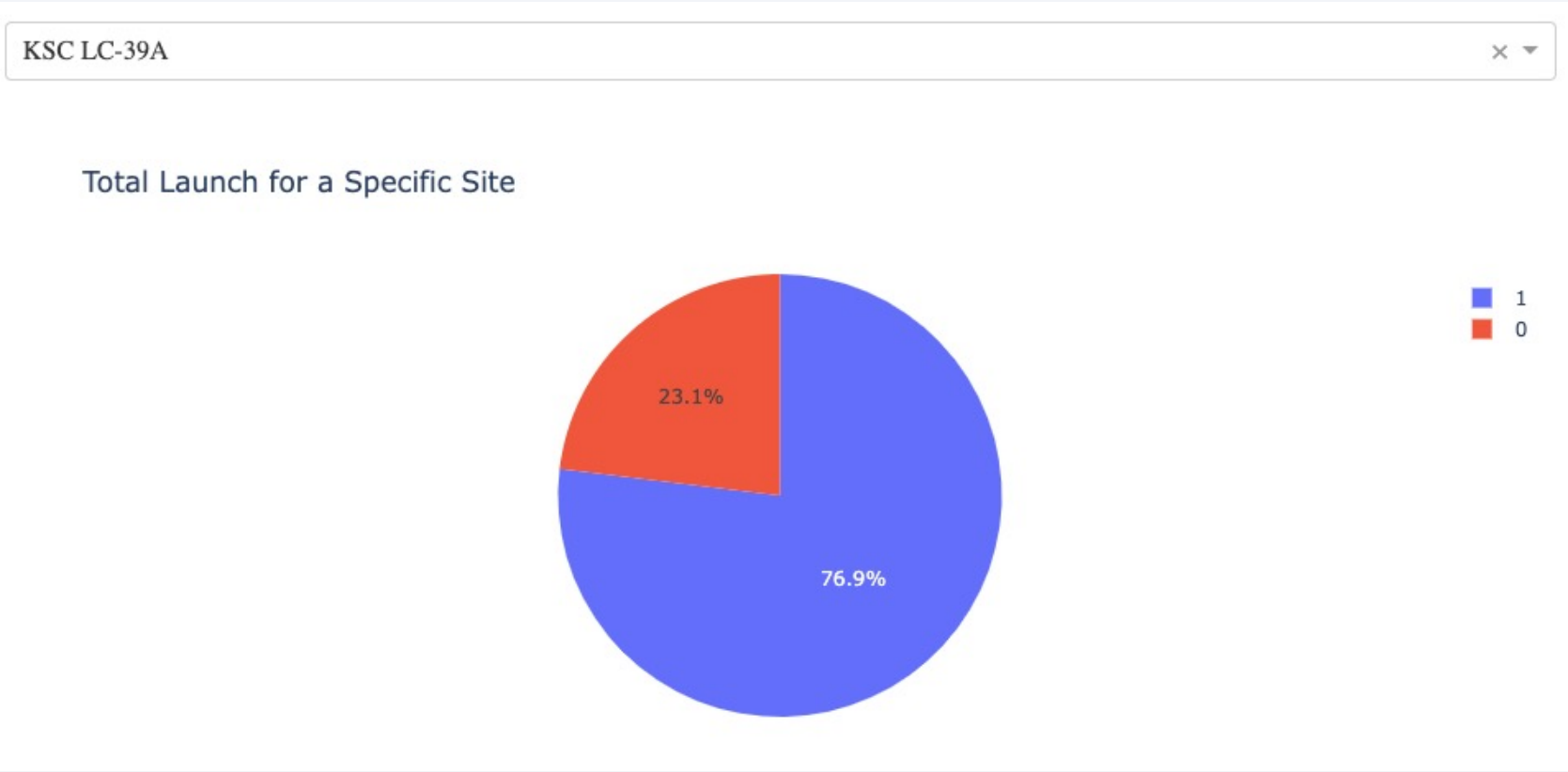
Build a Dashboard with Plotly Dash

Total Launches for All Sites



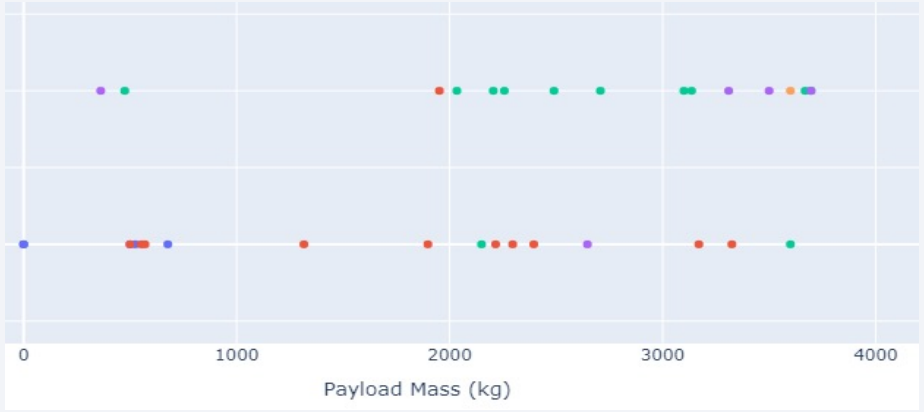
We can see that KSC LC-39A launch site has a large proportion of launches among other sites.

Launch Site with Highest Launch Success Ratio

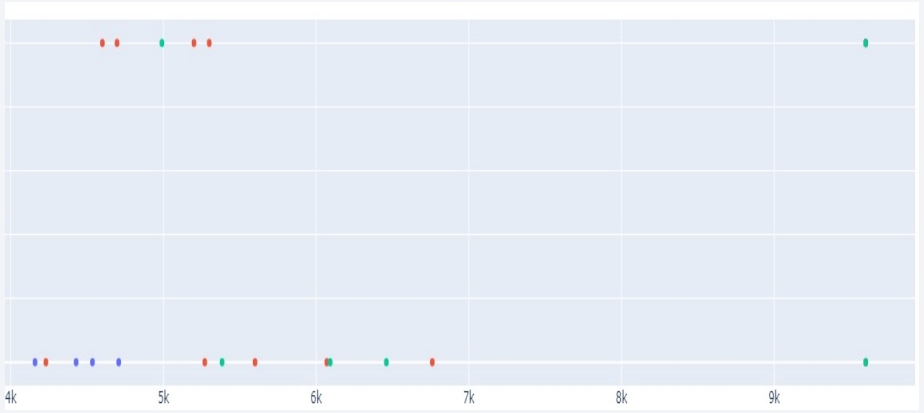


We find that KSC LC-39A launch site has the highest launch success ratio with 76.9% success rate.

Payload vs Launch Outcome



0kg – 4000kg



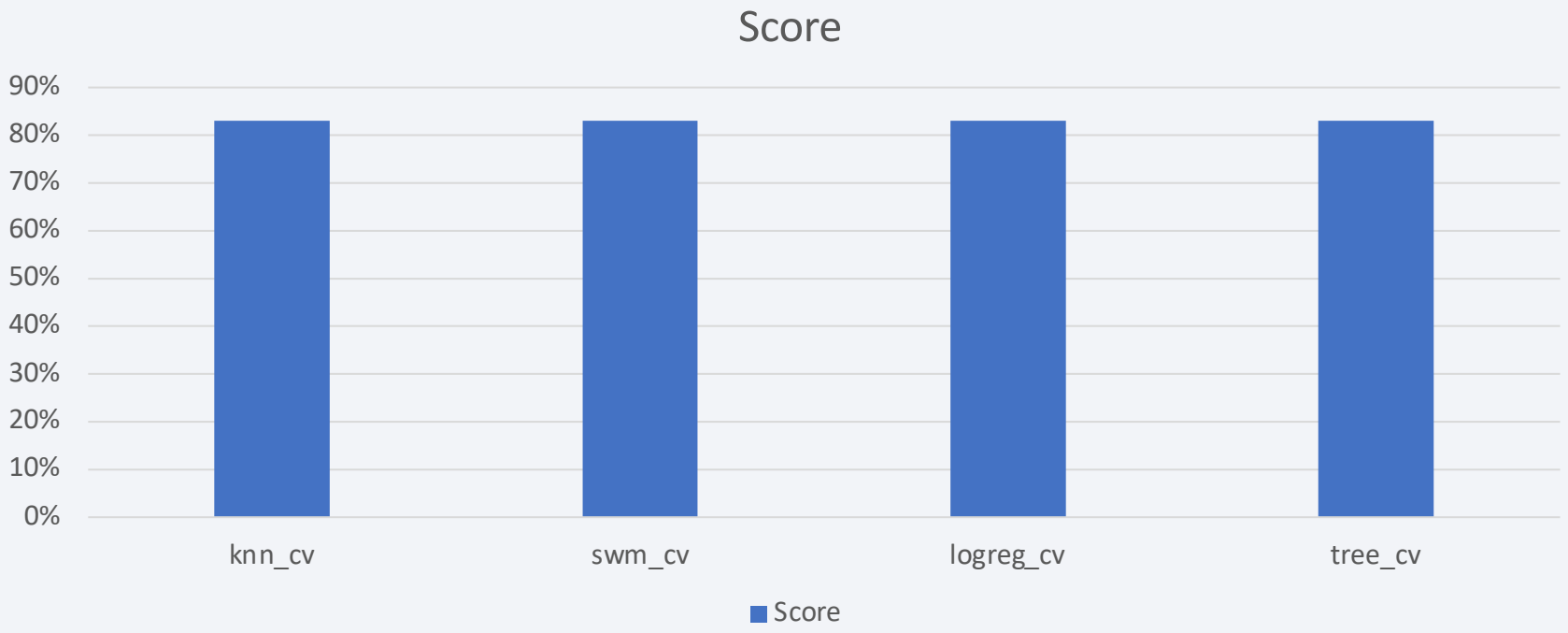
4000kg to 10000kg

It is suggested that low payload weight has a higher success rate compared to high payload weight.

Section 6

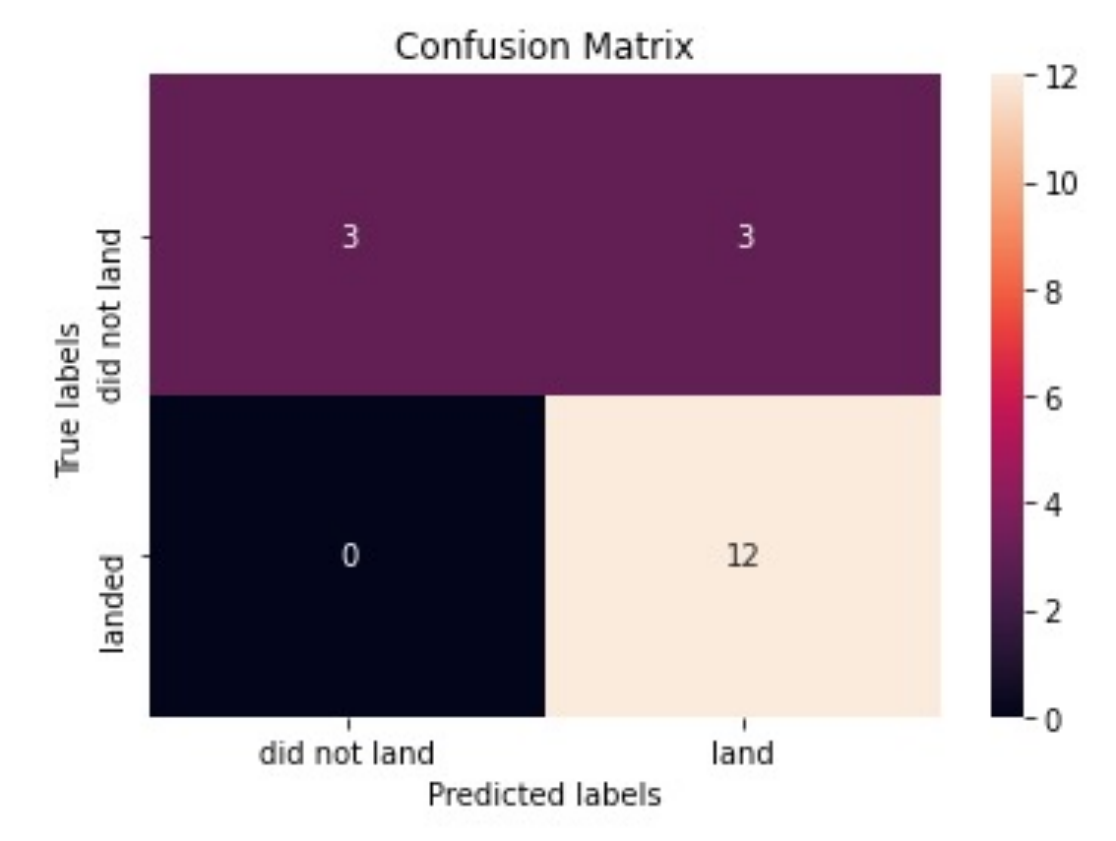
Predictive Analysis (Classification)

Classification Accuracy



We find that either algorithm is the best algorithm with accuracy of 83.3%.

Confusion Matrix



We find that has 3 false positives in our tests across all algorithms, however all algorithm also did not produce any false negatives.

Conclusions

Throughout the data analysis, we observed that

- Success rate has been increasing in trend, year to year. This correlation has little to do with the independent variables that we have observed in this data analysis project. This suggests that there may be other factors that were not observed in this data analysis, i.e. technology advancement in their rocket ships.
- Orbits ES-l1, GEO, HEO and SSO has a better success rate compared to other orbits.
- KSC LC-39A has better success-fail ratio compared to other launch sites.
- A lower payload has a better success rates than higher payload.
- A larger dataset might be required to perform machine learning algorithms.

Appendix

- Tools Used:
  - IBM Watson Studio
  - Microsoft Power Point
  - VS Code
  - Jupyter Notebook
- Programming Language/Library Used:
  - Python
  - Numpy, Pandas, Scikit-Learn, Folium, Plotly

Thank you!