IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Travis Thomas
June 22, 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection using SpaceX API
  - Data Collection using Web Scraping
  - Data Wrangling
  - Exploratory Data Analysis using SQL
  - Exploratory Data Analysis with Data Visualization
  - Interactive Visualization with Folium and Plotly Dash
  - Predictive Analysis using Machine Learning
- Summary of all results
  - Exploratory Data Analysis Results
  - Screenshots of Interactive Analysis
  - Results of Predictive Analysis

Introduction

- Project Background and Context
  - SpaceX advertises Falcon 9 rocket launches on its website at a cost of $62 million while other providers launches cost upwards of $165 million each. Much of SpaceX lower cost is due to the reusable nature of its rocket's first stage.
  - If we can determine whether a first stage will land, we may be able to determine the cost of a launch.
  - This information could be used to support competitive bids for launches against SpaceX.
  - We will create a machine learning pipeline to attempt to predict whether the first stage of a SpaceX Falcon 9 rocket will land successfully.
- Problems you want to find answers
  - What factors determine whether a rocket will land successfully?
  - How do the various determining factors interact between successful and unsuccessful launches?
  - What operating conditions need to exist to maximize the probability of a successful landing?

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data for analysis was collected using the SpaceX API and Wikipedia.
- Perform data wrangling
  - Categorical features were encoded using “One-Hot” encoding.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Construct, train, and evaluate various classification models to determine the optimal predictive model.

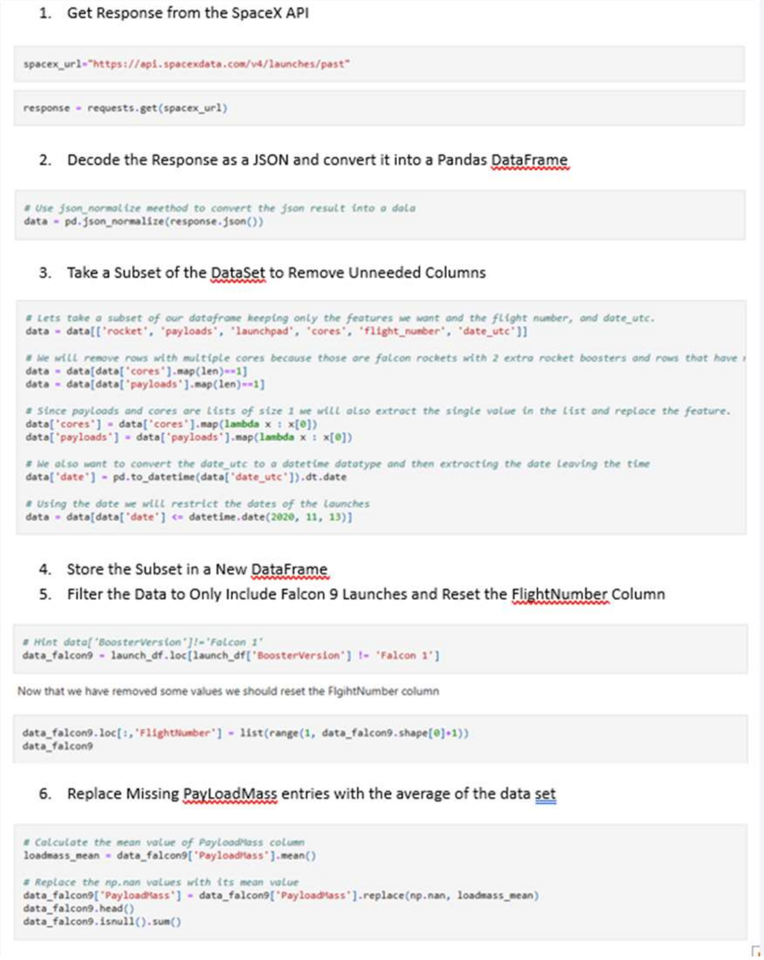
Data Collection

- Data was collected using a Get request to the SpaceX API.
- Next, the response content was decoded as a Json object using the .json() method, which was then converted into a Pandas dataframe using the .json_normalize() method.
- Data was then checked for missing values and missing values were replaced with the mean of the column, when necessary.
- Additional data was scraped from the Wikipedia page of Falcon 9 launches using a BeautifulSoup object.
- Data was then converted to a Pandas dataframe by parsing the HTML page for tabular data.

Data Collection – SpaceX API

- Data was collected using SpaceX's REST API.

- [IBM-Data-Science-Capstone/Lab 1.ipynb at main · travt2000/IBM-Data-Science-Capstone (github.com)]()



1. Get Response from the SpaceX API

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

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

2. Decode the Response as a JSON and convert it into a Pandas DataFrame

```
# Use json_normalize meethod to convert the json result into a dala
data = pd.json_normalize(response.json())
```

3. Take a Subset of the DataSet to Remove Unneeded Columns

```
# 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
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)]
```

4. Store the Subset in a New DataFrame
5. Filter the Data to Only Include Falcon 9 Launches and Reset the FlightNumber Column

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df.loc[launch_df['BoosterVersion'] != 'Falcon 1']
```

Now that we have removed some values we should reset the FlgihtNumber column

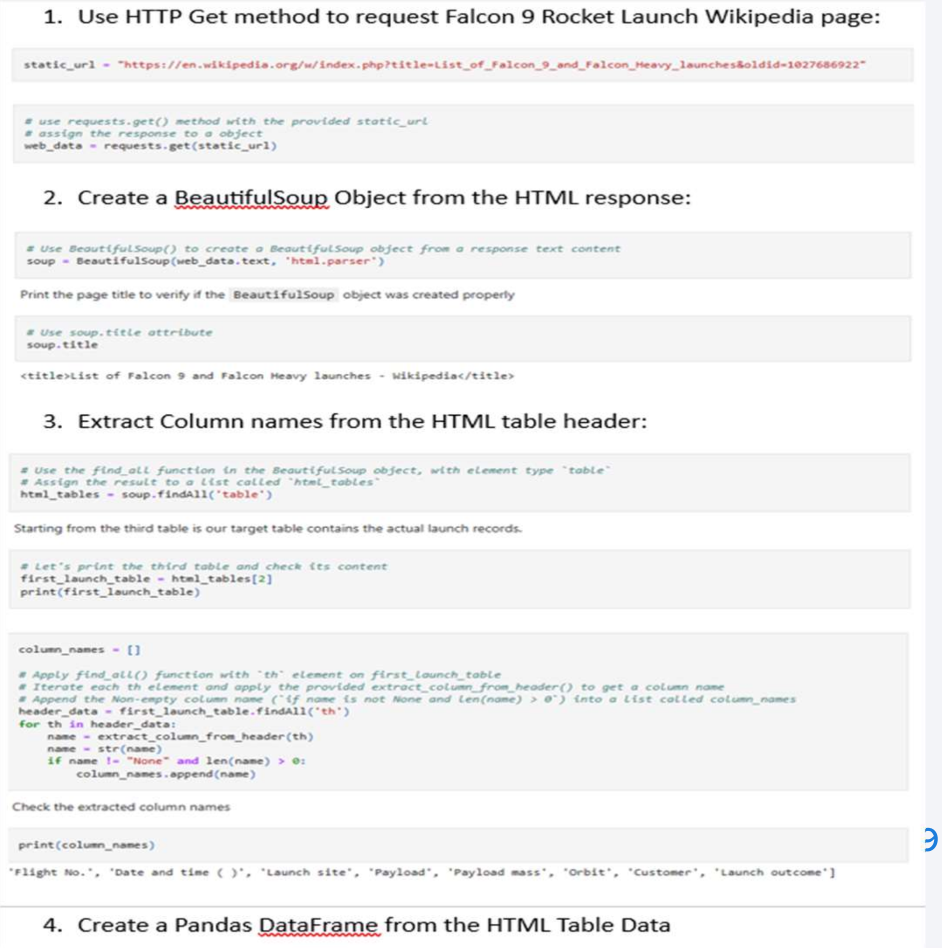
```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

6. Replace Missing PayLoadMass entries with the average of the data set

```
# Calculate the mean value of PayloadMass column
loadmass_mean = data_falcon9['PayloadMass'].mean()

# Replace the np.nan values with its mean value
data_falcon9['PayloadMass'] = data_falcon9['PayloadMass'].replace(np.nan, loadmass_mean)
data_falcon9.head()
data_falcon9.isnull().sum()
```

Data Collection - Scraping

- Web scraping was used to collect Falcon 9 launch records with BeautifulSoup and then converted to a Pandas Dataframe.
- IBM-Data-Science-Capstone/Data Science Captsone - Web Scraping.ipynb at main · travt2000/IBM-Data-Science-Capstone · GitHub



1. Use HTTP Get method to request Falcon 9 Rocket Launch Wikipedia page:

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

```
# use requests.get() method with the provided static_url
# assign the response to a object
web_data = requests.get(static_url)
```

2. Create a BeautifulSoup Object from the HTML response:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(web_data.text, 'html.parser')
```

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

```
# Use soup.title attribute
soup.title
```

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

3. Extract Column names from the HTML table header:

```
# Use the find_all function in the BeautifulSoup object, with element type `table`
# Assign the result to a list called `html_tables`
html_tables = soup.findAll('table')
```

Starting from the third table is our target table contains the actual launch records.

```
# Let's print the third table and check its content
first_launch_table = html_tables[2]
print(first_launch_table)
```

```
column_names = []

# Apply find_all() function with `th` element on first_launch_table
# Iterate each th element and apply the provided extract_column_from_header() to get a column name
# Append the Non-empty column name (`if name is not None and len(name) > 0`) into a list called column_names
header_data = first_launch_table.findAll('th')
for th in header_data:
    name = extract_column_from_header(th)
    name = str(name)
    if name != "None" and len(name) > 0:
        column_names.append(name)
```

Check the extracted column names

```
print(column_names)
```

```
'Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

4. Create a Pandas DataFrame from the HTML Table Data

Data Wrangling

- Data wrangling was performed to determine training labels and perform initial exploratory data analysis.
- The percentage of missing values was determined for each attribute.
- The number of launches per launch site, the number of launches by orbital type, and the number of mission outcomes by orbital type were calculated.
- Launch outcomes were analyzed and a binary landing Class column was created.
- IBM-Data-Science-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_1_L3_labs-jupyter-spacex-data_wrangling_jupyterlite.jupyterlite.ipynb at main · travt2000/IBM-Data-Science-Capstone · GitHub

EDA with Data Visualization

- Scatter plot of Flight Number vs Launch Site. Used to identify any trend in the usage of launch sites and the successes of launches over time.
- Scatter plot of Payload vs Launch Site. Used to analyze payload sizes launched at different launch sites.
- Bar Chart of Success Rate by Orbit Type. Used to analyze which orbits result in the most successful launches.
- Scatter plot of Flight Number vs Orbit Type. Used to changes in orbit type over time.
- Scatter plot of Payload and Orbit Type. Used to analyze the size of payloads used in different orbital missions.
- Line chart of Launch Success Rate vs Year. Used to see changes in mission success over time.
- IBM-Data-Science-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_2_jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb at main · travt2000/IBM-Data-Science-Capstone · GitHub

EDA with SQL

- The SpaceX data set was loaded into an SQLLite database within the Jupyter notebook.
- Various SQL queries were used to gain insight into the data. Queries were performed for the following questions:
  - The names of unique launch sites
  - The total payload mass carried by rockets launched for NASA (CRS)
  - The average payload mass by rocket booster version F9 v1.1
  - The total number of successful and failure mission outcomes
  - The failed landing missions in drone ship including their booster version and launch site names.
- For detailed results, please see the included notebook:
  - IBM-Data-Science-Capstone/jupyter-labs-eda-sql-coursera_sqllite.ipynb at main · travt2000/IBM-Data-Science-Capstone · GitHub

Build an Interactive Map with Folium

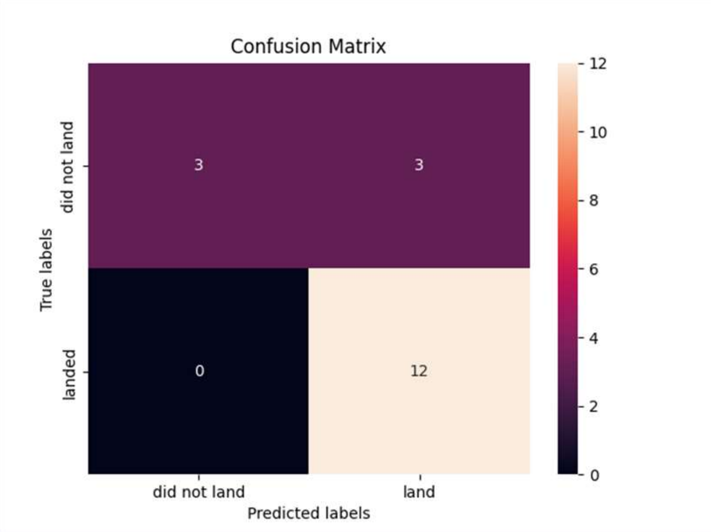
- Individual launch sites were marked and labelled and clusters of icons were added to note the number of successful and failed missions. In addition, locations such as coastlines and nearby cities were marked along with lines showing the distance from the launch site.
- Color coded clusters were used to note the number of launches at each site and the number of successful and failed missions.
- Nearby coastlines and cities were marked and labelled along with lines to denote the distance to the launch site.
  - It was found that all launch sites are near coastlines.
  - Launch sites are located well away from nearby cities.
- IBM-Data-Science-Capstone/Folium Captsone Project.ipynb at main · travt2000/IBM-Data-Science-Capstone · GitHub

Build a Dashboard with Plotly Dash

- Plotly Dash was used to create an interactive dashboard to analyze launch sites and the relationship between successful/failed missions and payload mass by booster version.
- Pie charts were used to show total launches by site and successful and failed missions at each site.
- Scatter plots were used to show the relationship between mission outcome and payload mass for each booster version.
- Source code for the Plotly Dash dashboard can be found at: IBM-Data-Science-Capstone/main.py at main · travt2000/IBM-Data-Science-Capstone · GitHub

Predictive Analysis (Classification)

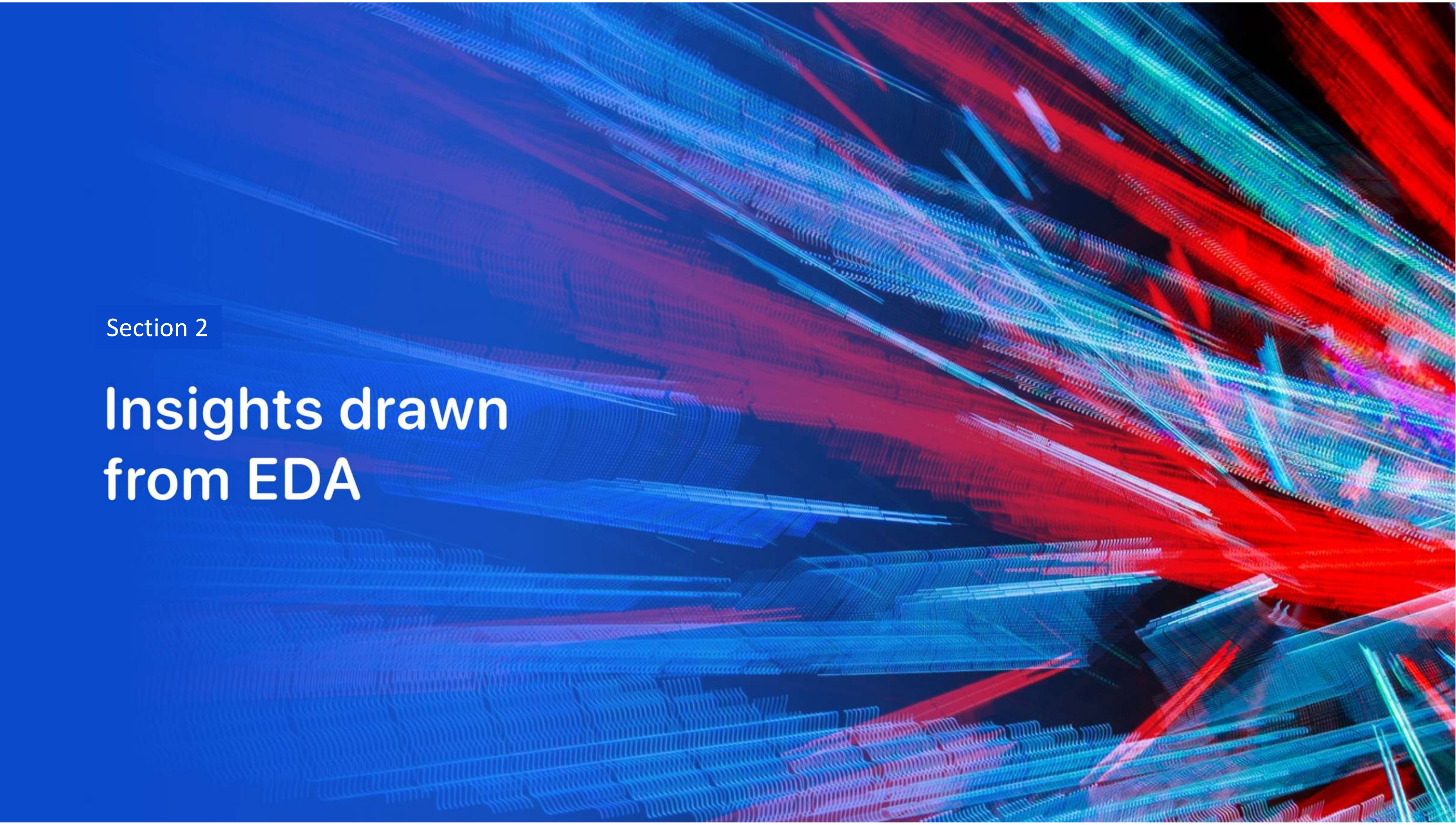
- Logistic Regression, Support Vector Machines, Decision Trees, and K-Nearest Neighbors Algorithms were used for predictive analysis using the Falcon 9 launch data set.
- The data set was broken into a Training and Test set sets with the Test set including 20% of the total data set.
- All 4 models produced an F1 score of approximately 0.83333.
- Each model correctly predicted all successful landings but correctly predicted only 50% of failed landings.



IBM-Data-Science-Capstone/IBM-DS0321EN-SkillsNetwork_labs_module_4_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite (1).ipynb at main · travt2000/IBM-Data-Science-Capstone (github.com)

Results

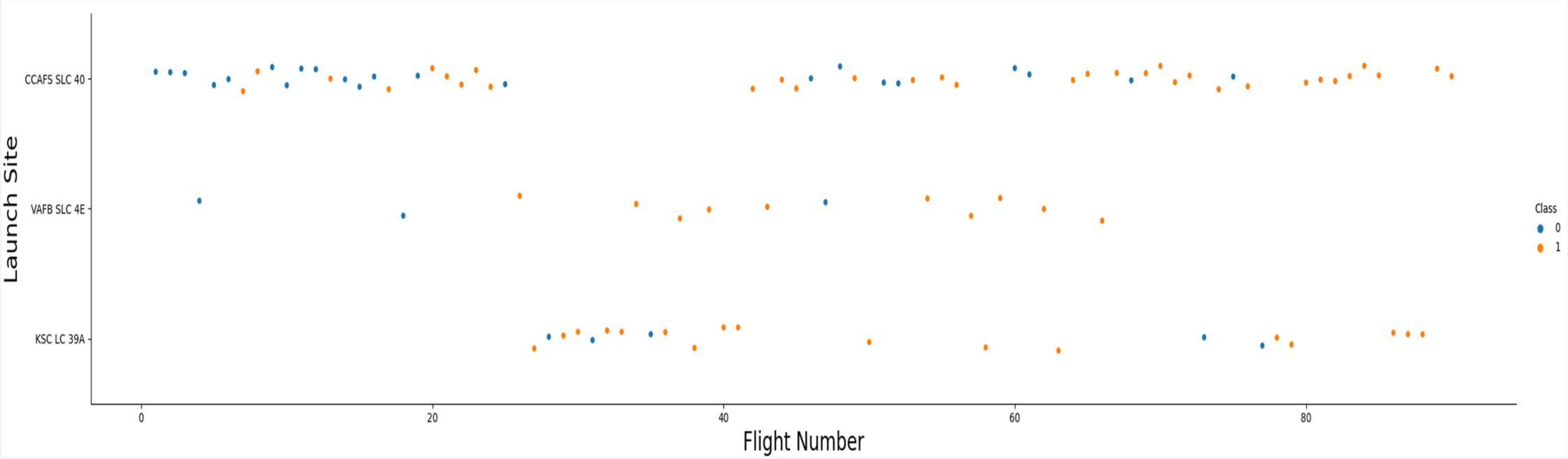
- A Decision Tree model produced the best overall accuracy, but all 4 models used produced similar results in terms of accuracy and precision.
- The success rate for SpaceX launches is directly proportional to the date of the launch, indicating improving performance as the launch process and technology continues to be refined.



Section 2

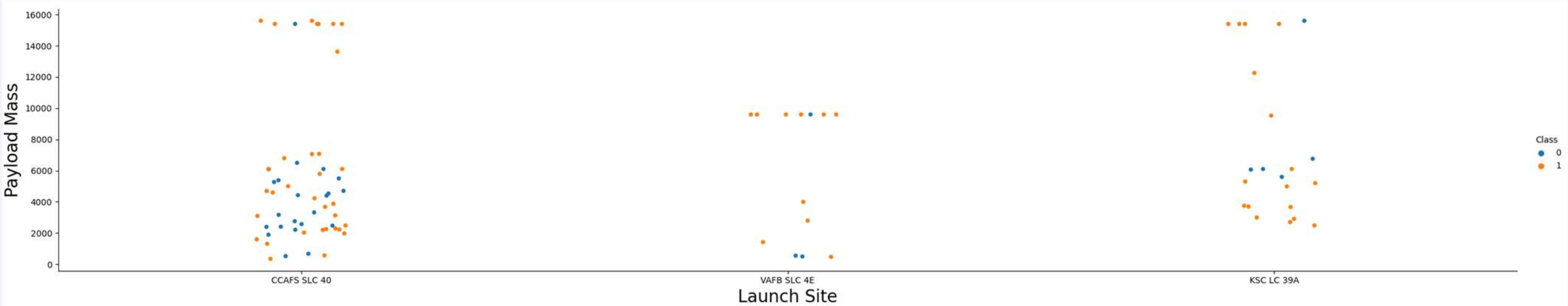
Insights drawn from EDA

Flight Number vs. Launch Site



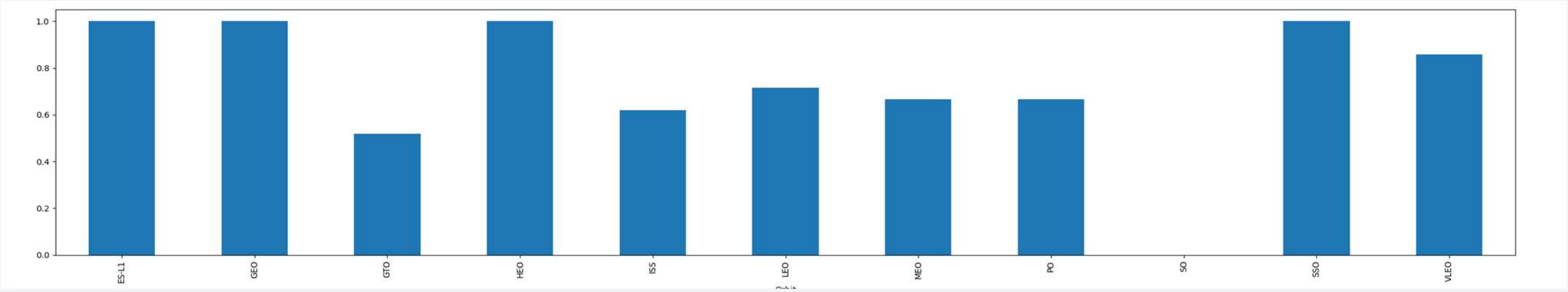
- There have been significantly more launches from site CCAFS SLC 40 than any other launch site.

Payload vs. Launch Site



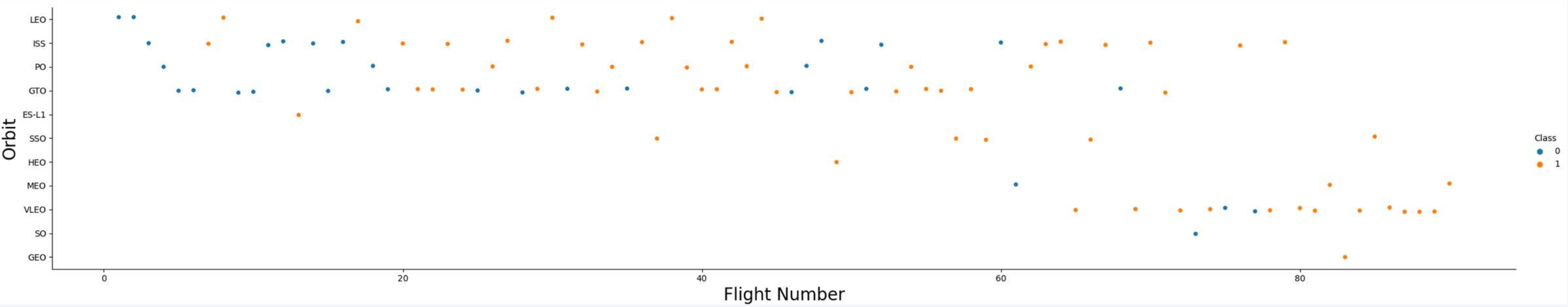
- Launch site VAFB SLC 4E has had no launches with payloads of 10,000kg.
- Launch site CCAFS SLC-40 has had a higher volume of launches with most launches being of lower payload mass.

Success Rate vs. Orbit Type



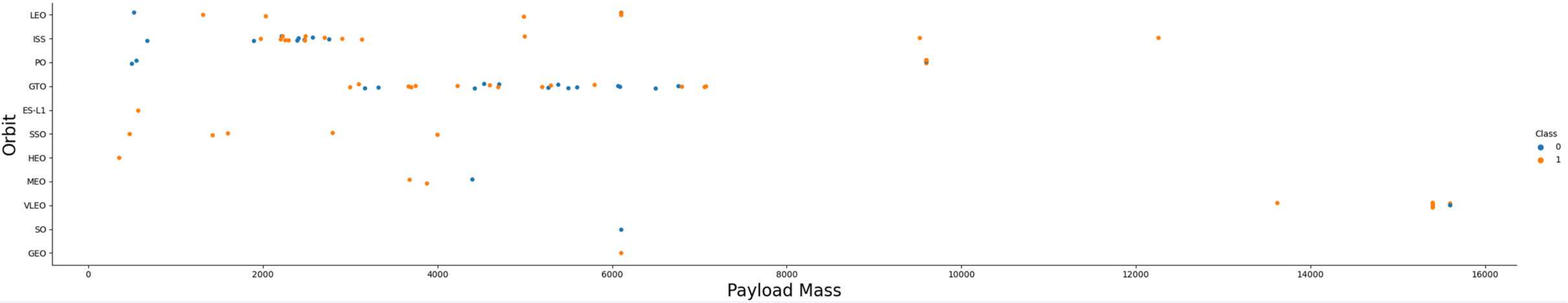
- Orbit types ES-L1, GEO, HEO, and SSO have high success rates at or near 100%
- Orbit type SO has had no successful missions
- Other orbit types have had varying success rates.

Flight Number vs. Orbit Type



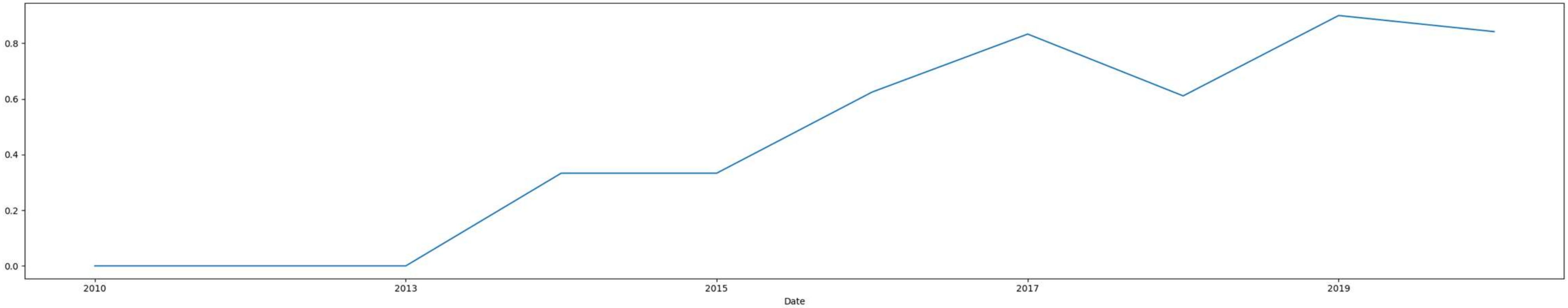
- Later launches have been primarily VLEO launches.

Payload vs. Orbit Type



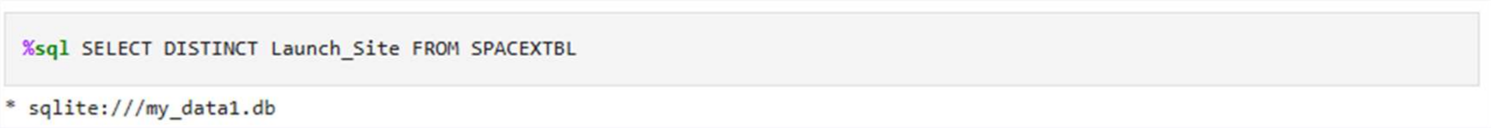
- There is a relationship between orbit classes and payload masses.
- Higher orbits typically have larger payload masses.
- Lower orbits typically have lower total payload masses.

Launch Success Yearly Trend



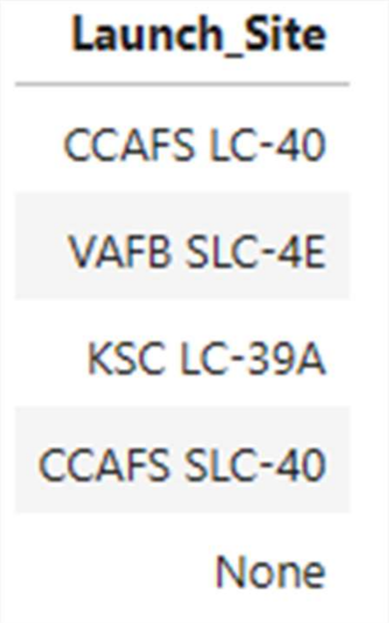
- Launch success rates have improved over time with a significant improvement after 2013.

All Launch Site Names



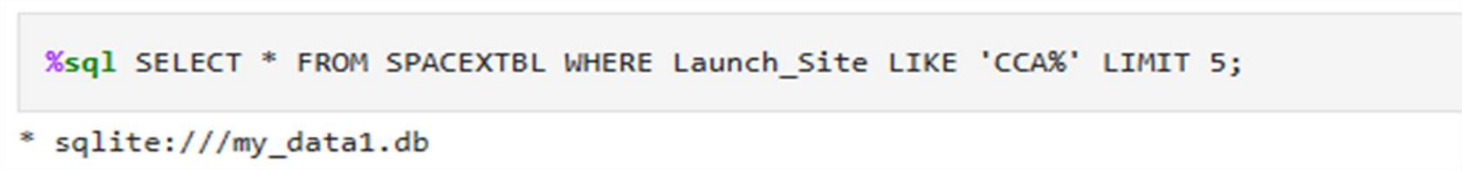
```
%sql SELECT DISTINCT Launch_Site FROM SPACEXTBL
* sqlite:///my_data1.db
```

- The query selects distinct launch sites from the SpaceX table.



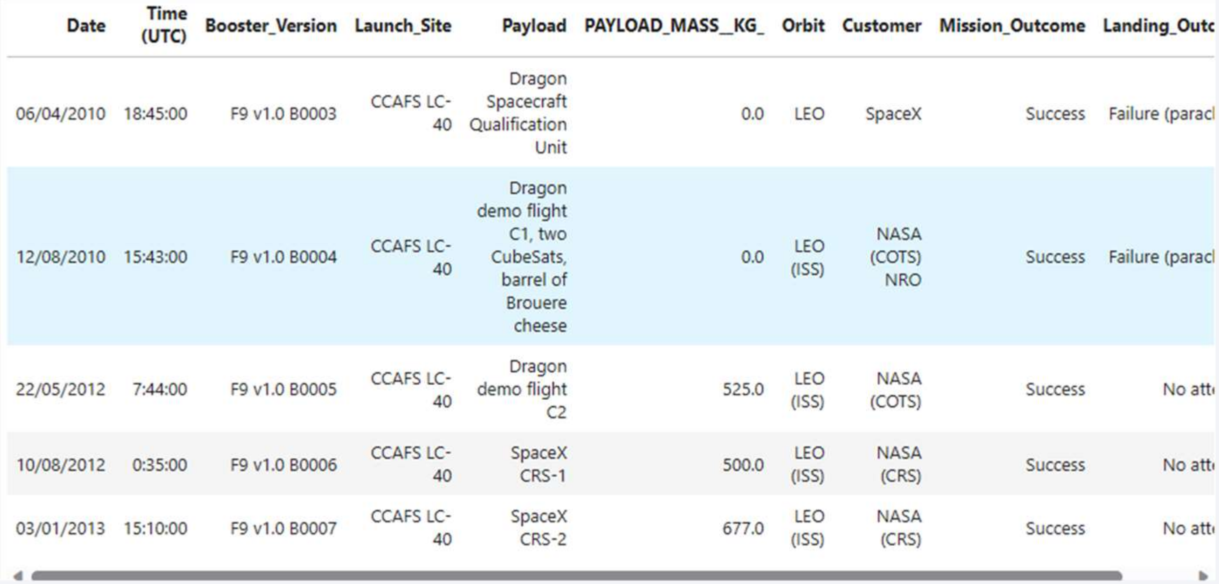
Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Launch Site Names Begin with 'CCA'



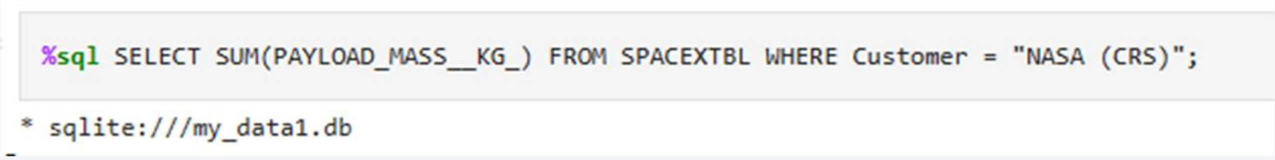
```
%sql SELECT * FROM SPACEXTBL WHERE Launch_Site LIKE 'CCA%' LIMIT 5;

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

- The query returns a total of 5 records where the launch site name begins with CCA.



Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

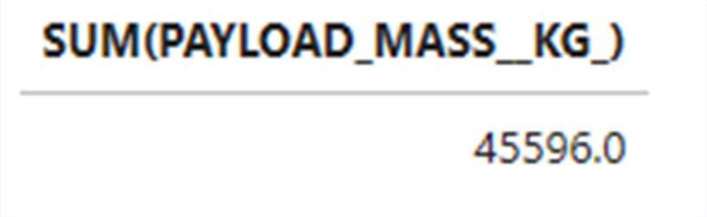
Total Payload Mass



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

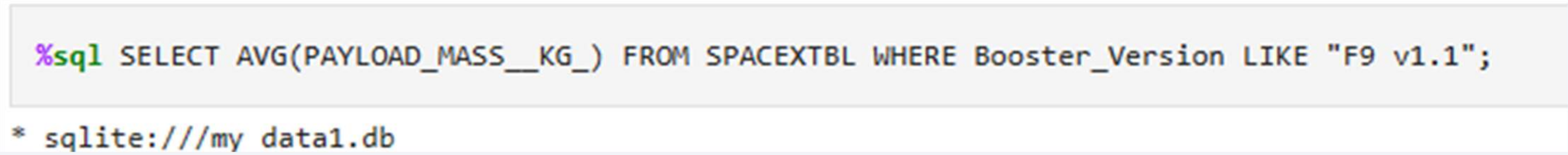
```
* sqlite:///my_data1.db
```

- The query returns the total (sum) mass of all launches by NASA.



SUM(PAYLOAD_MASS__KG_)
45596.0

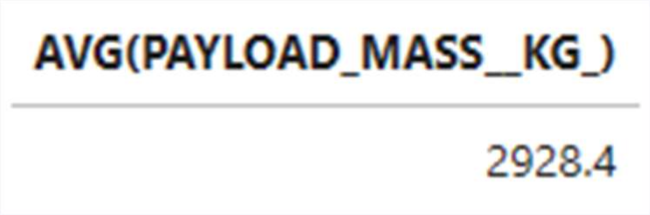
Average Payload Mass by F9 v1.1



```
%sql SELECT AVG(PAYLOAD_MASS__KG_) FROM SPACEXTBL WHERE Booster_Version LIKE "F9 v1.1";
```

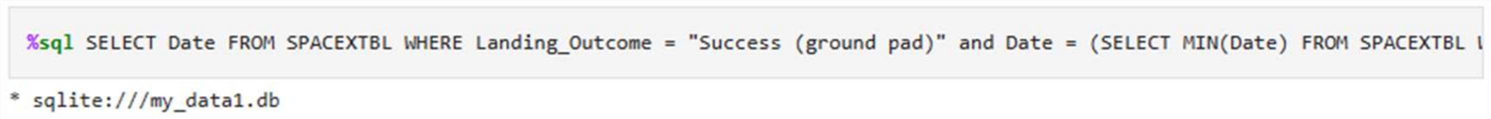
```
* sqlite:///my_data1.db
```

- The query returns the average payload mass carried by booster version F9 1.1.



AVG(PAYLOAD_MASS__KG_)
2928.4

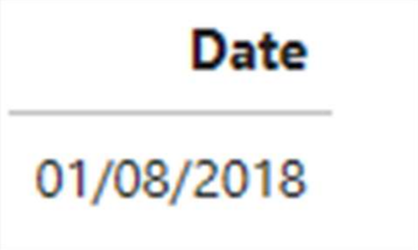
First Successful Ground Landing Date



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

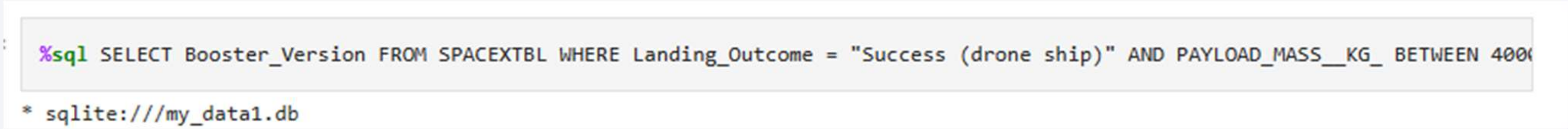
```
* sqlite:///my_data1.db
```

- The query returns the earliest date of a successful ground pad launch.



Date
01/08/2018

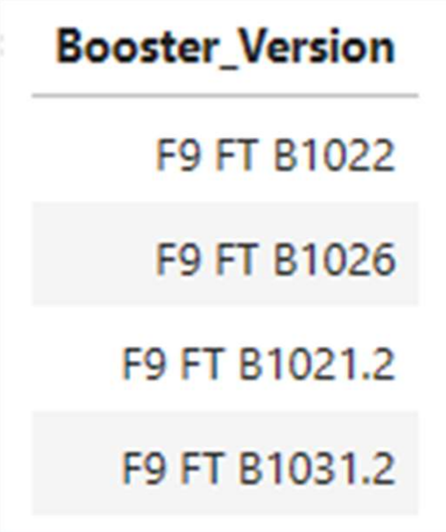
Successful Drone Ship Landing with Payload between 4000 and 6000



```
%sql SELECT Booster_Version FROM SPACEXTBL WHERE Landing_Outcome = "Success (drone ship)" AND PAYLOAD_MASS__KG_ BETWEEN 400
```

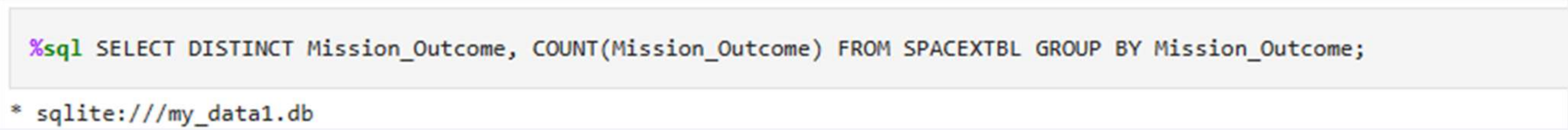
```
* sqlite:///my_data1.db
```

- The query returns the list of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000



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

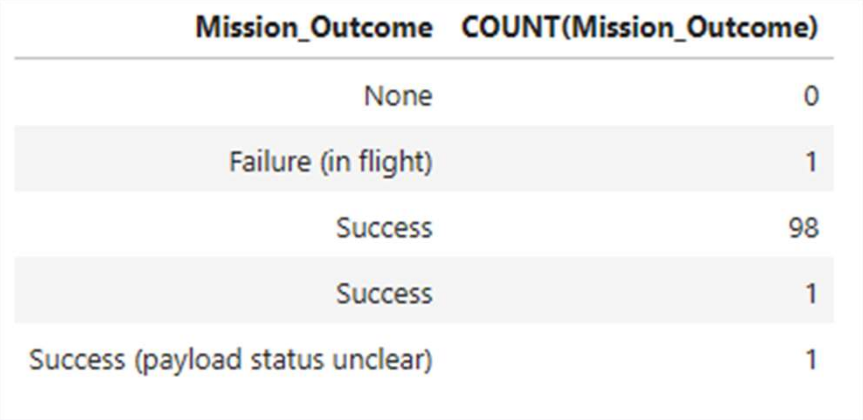
Total Number of Successful and Failure Mission Outcomes



```
%sql SELECT DISTINCT Mission_Outcome, COUNT(Mission_Outcome) FROM SPACEXTBL GROUP BY Mission_Outcome;
```

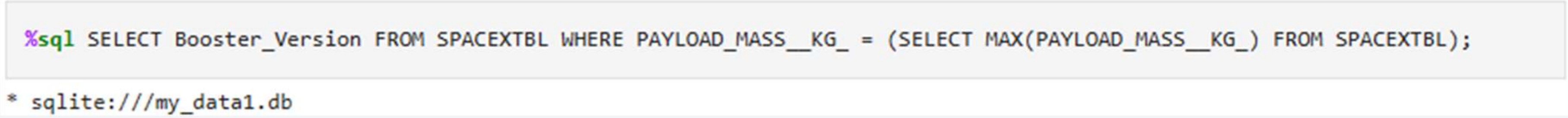
```
* sqlite:///my_data1.db
```

- The query returns the total number of successful and failed missions.



Mission_Outcome	COUNT(Mission_Outcome)
None	0
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

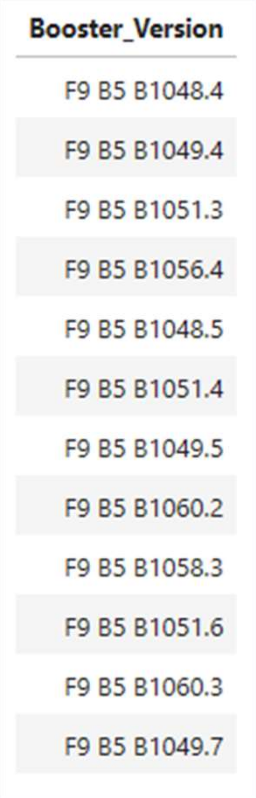
Boosters Carried Maximum Payload



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

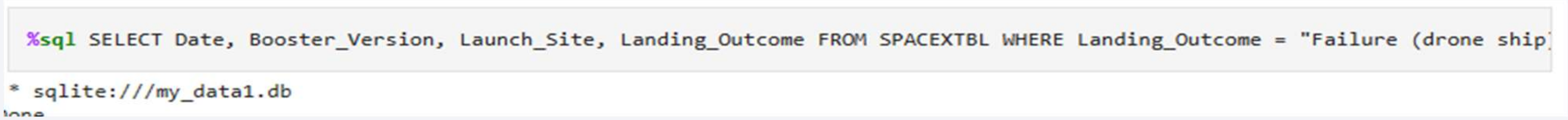
```
* sqlite:///my_data1.db
```

- The query returns the names of booster versions that have carried the maximum payload mass.



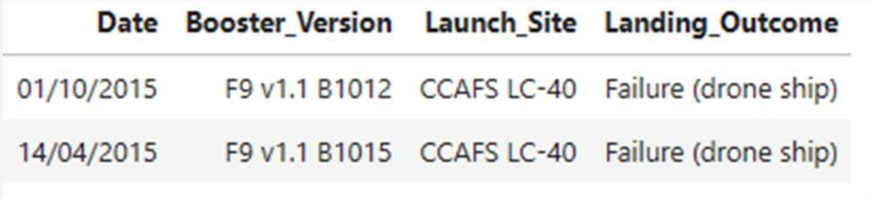
Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

2015 Launch Records



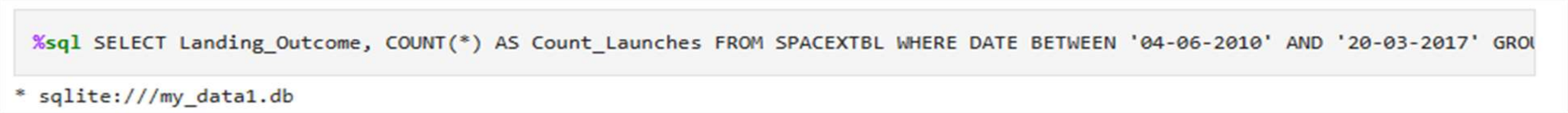
```
%sql SELECT Date, Booster_Version, Launch_Site, Landing_Outcome FROM SPACEXTBL WHERE Landing_Outcome = "Failure (drone ship]

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

- The query returns a list of failed landings in drone ship along with their booster version and launch sites.



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

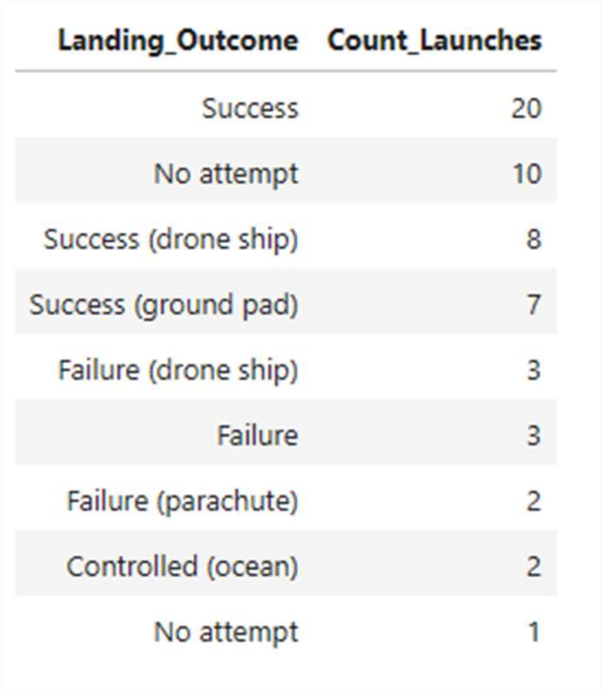
Rank Landing Outcomes Between 2010-06-04 and 2017-03-20



```
%sql SELECT Landing_Outcome, COUNT(*) AS Count_Launches FROM SPACEXTBL WHERE DATE BETWEEN '04-06-2010' AND '20-03-2017' GROU
```

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

- The query returns the count of landing outcomes from 2010-06-04 and 2017-03-20 ranked by outcome, in descending order.

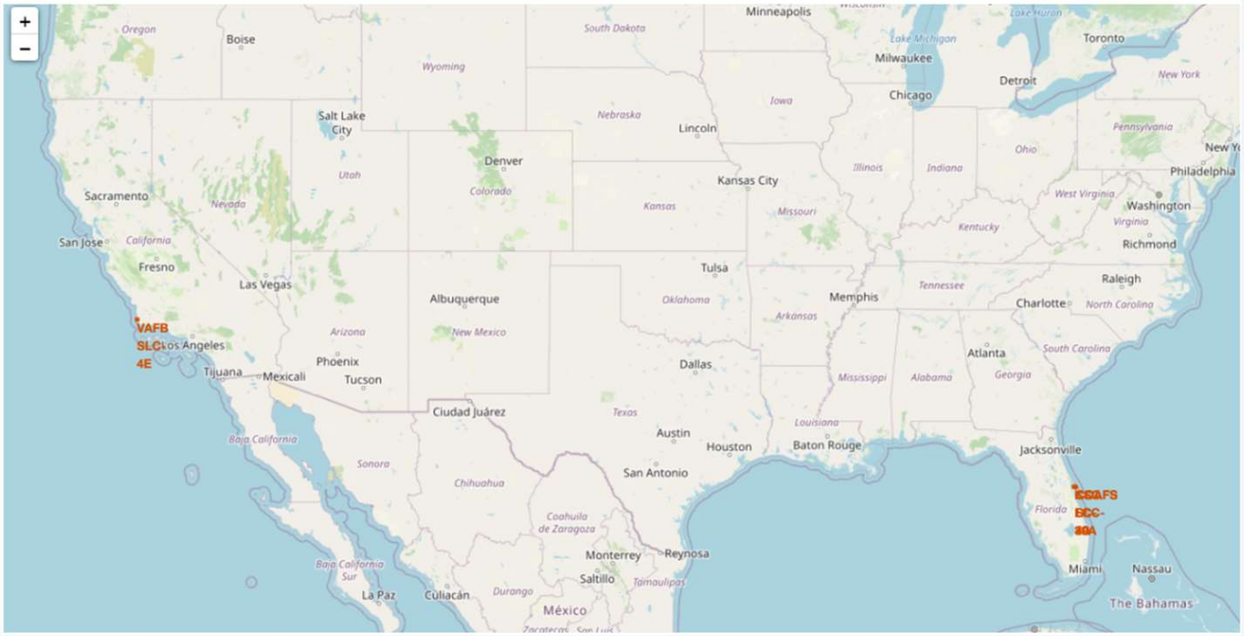


Landing_Outcome	Count_Launches
Success	20
No attempt	10
Success (drone ship)	8
Success (ground pad)	7
Failure (drone ship)	3
Failure	3
Failure (parachute)	2
Controlled (ocean)	2
No attempt	1

Section 3

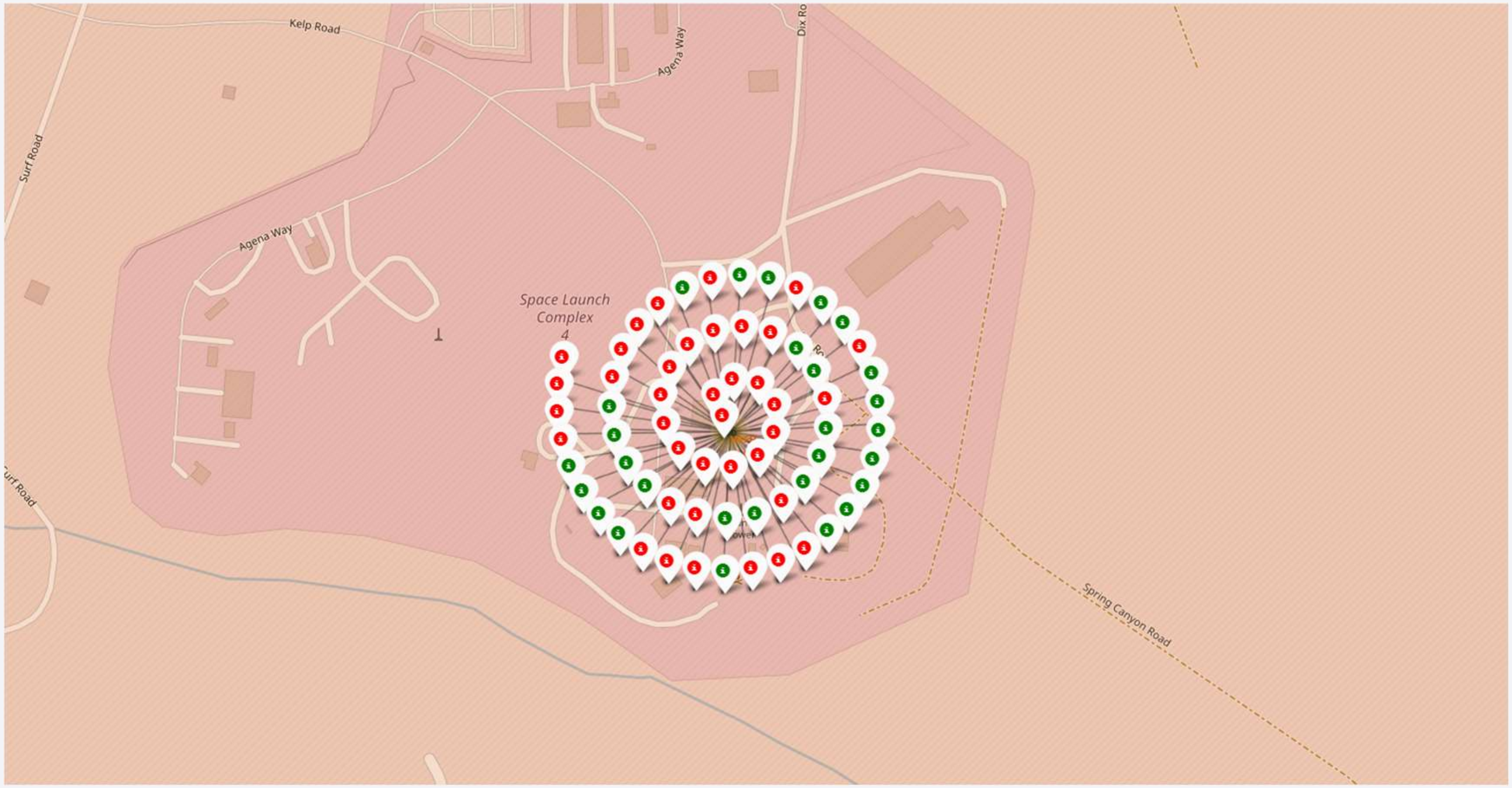
Launch Sites Proximities Analysis

Falcon 9 Launch Sites



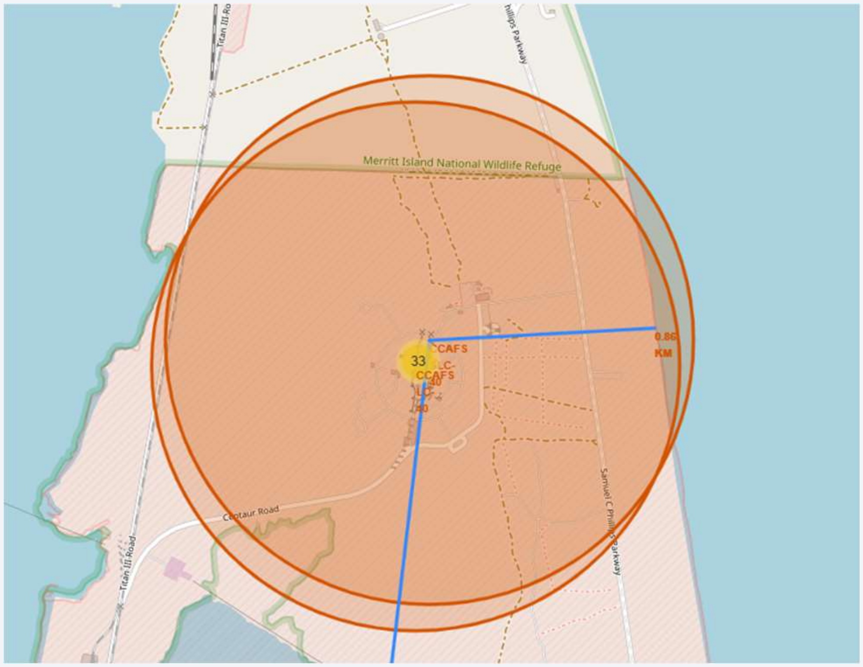
- All Launch Sites are Located in Coastal Areas, Away from Large Cities

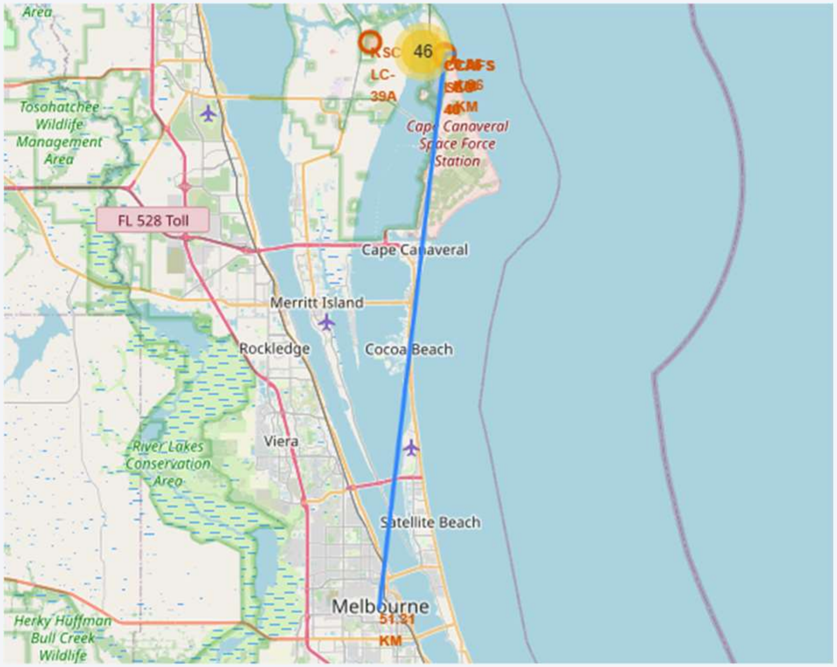
Successful and Failed Launches Plot



- Launch site VAFB SLC-4E had 66 launches. Successful launches are labeled in green and failed launches are labeled in red.

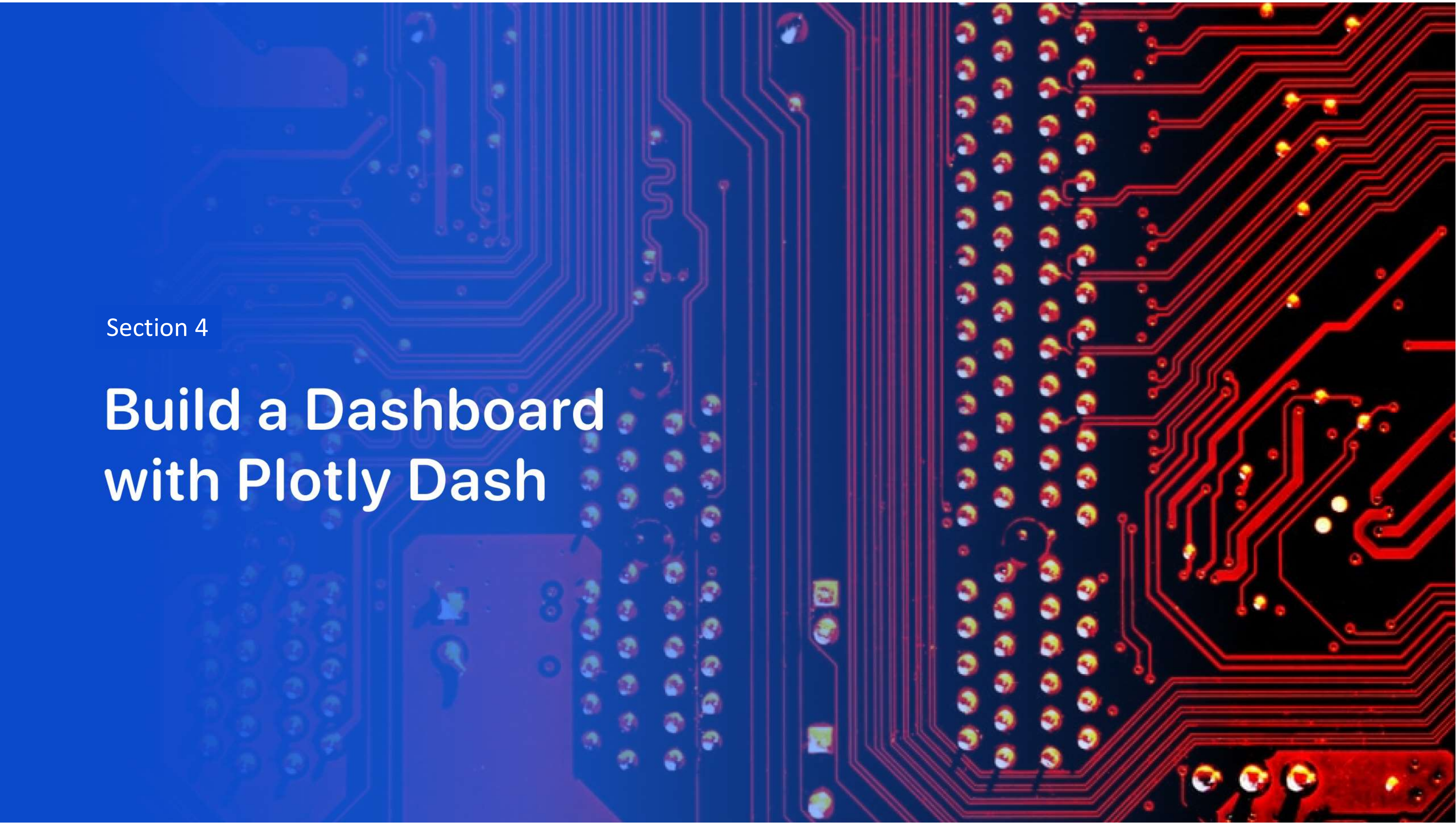
Launch Site Geographic Features


Merritt Island National Wildlife Refuge
33
CAFS
LC-
CCAFS
0.86
KM
Centaur Road
Titan III Road
Samuel C Phillips Parkway



Launch Site CCAFS is located 0.86km from the coast

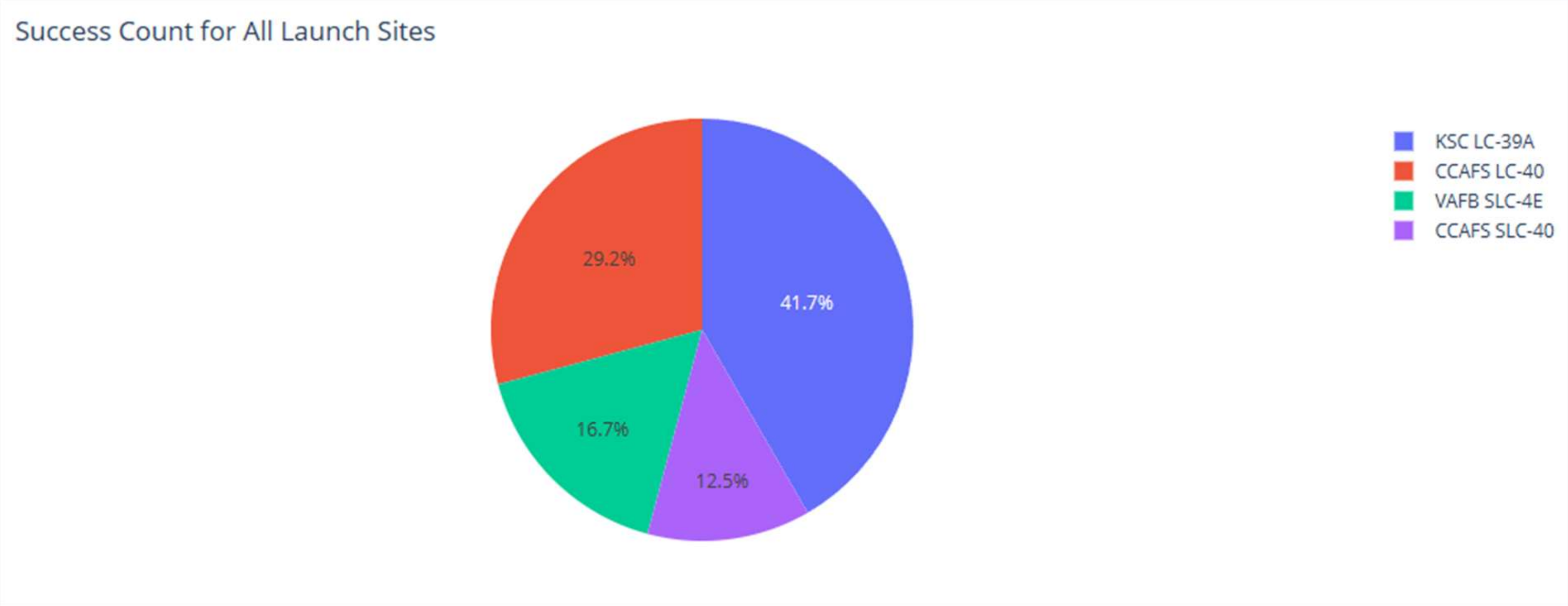
Launch site CCAFS is located 51.21km from Melbourne, FL.



Section 4

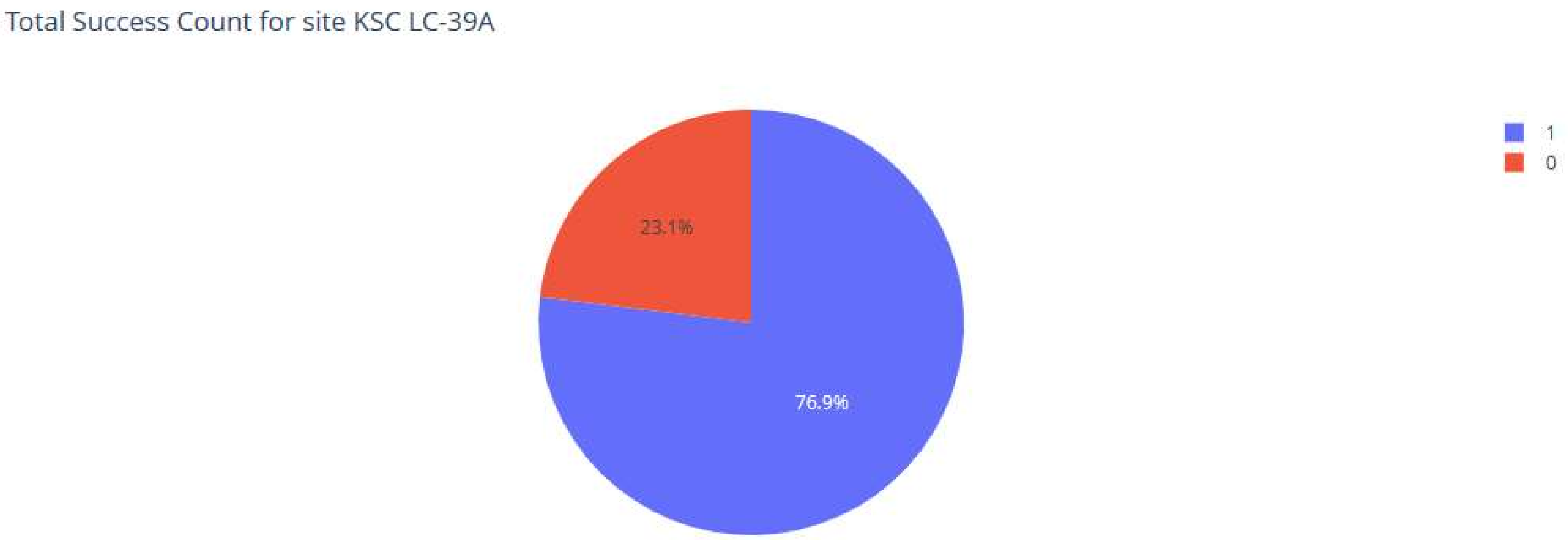
Build a Dashboard with Plotly Dash

Total Successful Missions by Launch Site



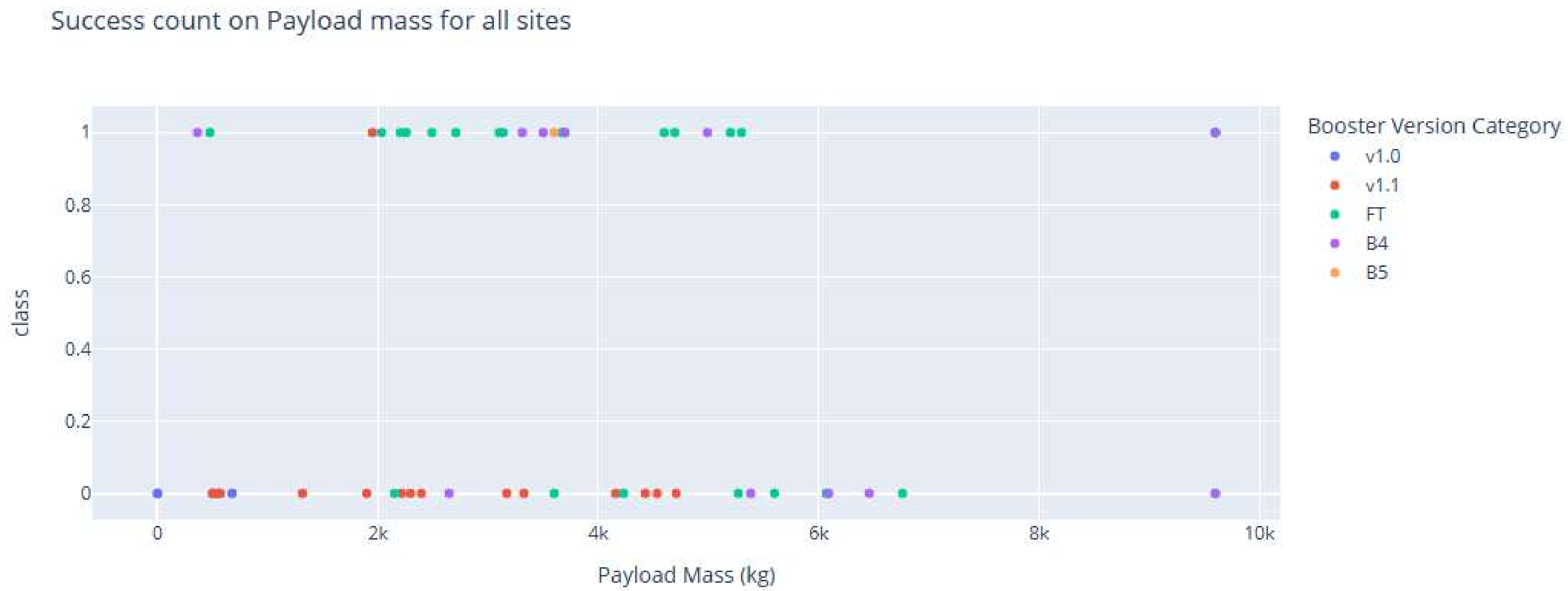
- Site KSC LC-39A the most successful launches among the launch sites.

Success Rate by Site – KSC LC-39A



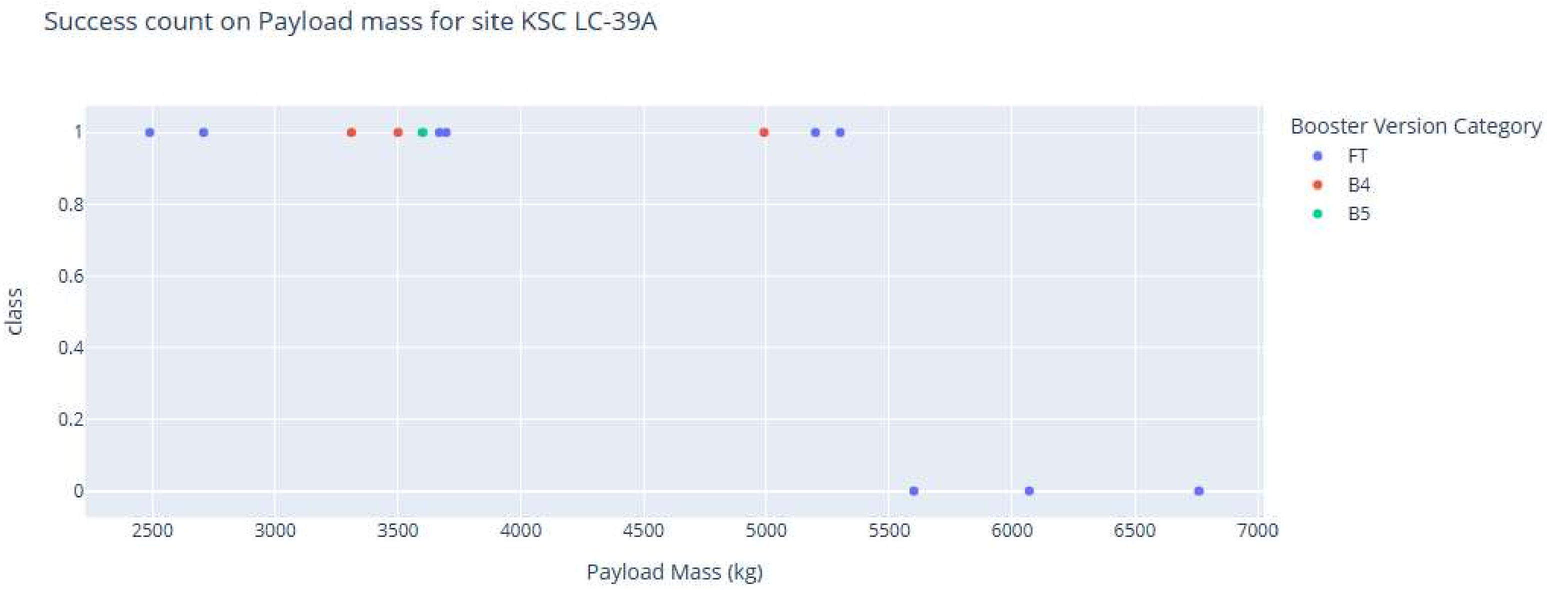
- 76.9% of site KSC LC-39A’s launches have been successful.

Success Rate by Payload Mass for All Sites



- Across booster versions, the success rate is higher with lower payload masses.

Success Rate by Payload Mass – Site KSC LC-39A

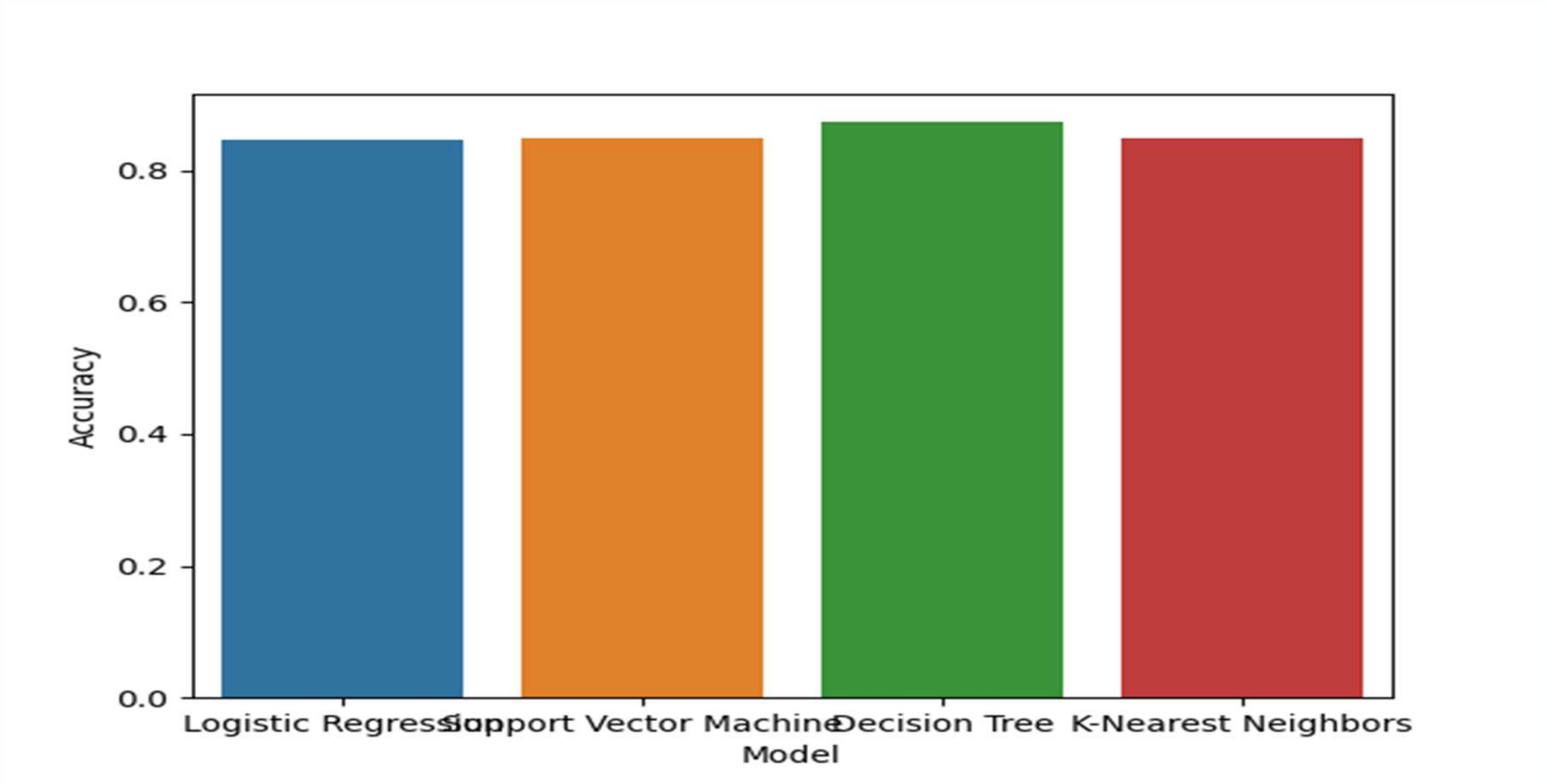


- Site KSC LC-39A launches are typically of a lower total payload mass. This correlates to the site's high success rate.

Section 5

Predictive Analysis (Classification)

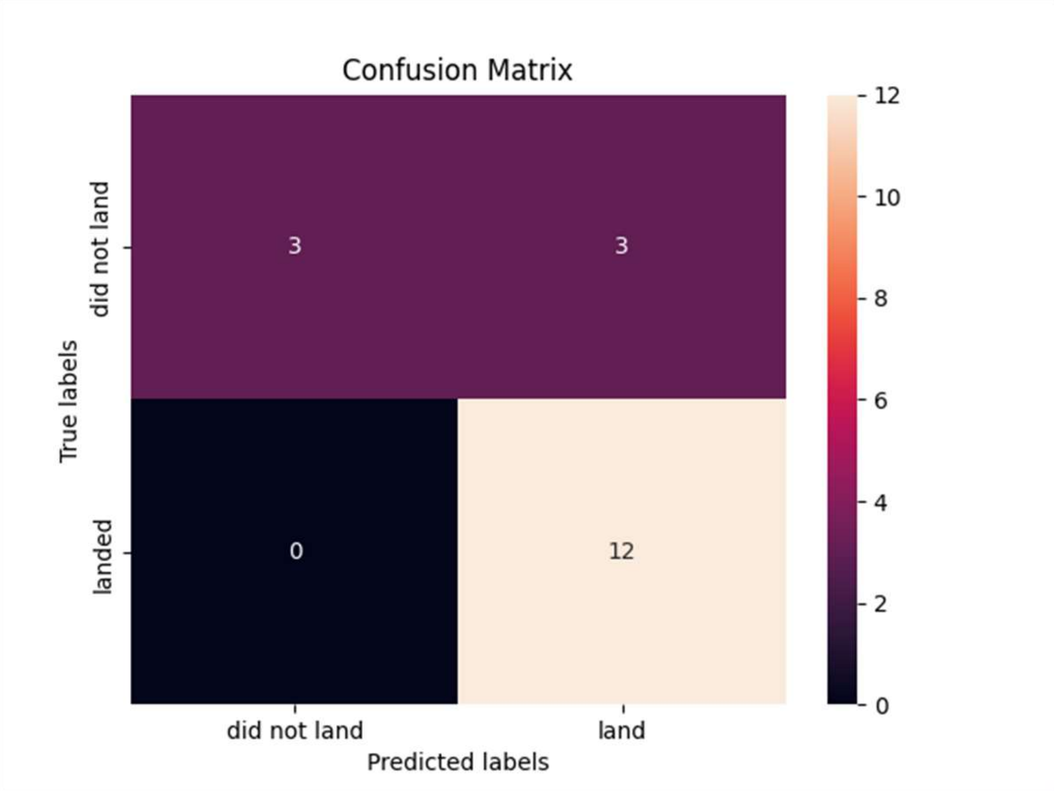
Classification Accuracy



- The Decision Tree showed the highest model accuracy.

Confusion Matrix

- The Decision Tree accurately predicted all successful launches.
- The Decision Tree correctly predicted 50% of unsuccessful launches.
- The best_score_ method showed that the Decision Tree was slightly more accurate than the other models used.



Conclusions

- The Decision Tree delivered the highest numerical accuracy.
- The Confusion Matrixes showed that all 4 models accurately predicted all successful launches and correctly predicted 50% of failed launches.
- F1 scores for all models (score() method) was 0.833333. This indicates that all 4 models had the same level of accuracy and precision.

Thank you!