IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Marine Bentivoglio
23 August 2024

Outline

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

Executive Summary

- **Summary of methodologies:**

Data from the SpaceX REST API and the Falcon 9 Wikipedia page was collected, wrangled and analyzed to predict if the Falcon 9 rocket first stage will land successfully, and ultimately to determine the cost of a launch.

Analysis included EDA with visualization and SQL, interactive visual analytics with Folium maps and Plotly Dash, and predictive analysis with classification models such as Logistic Regression, SVM, Decision Tree and KNN.

- **Summary of all results:**

The launch outcome was examined at 4 distinct launch sites based on multiple variables including the launch sites locations, the rocket payload mass, the boosters version and the launch orbits.

Overall, the launch success rate has increased since 2013. Looking closely, the KSC LC-39A site, newer orbits, FT boosters and payloads in the [2490-5300 kg] range showed higher success rates.

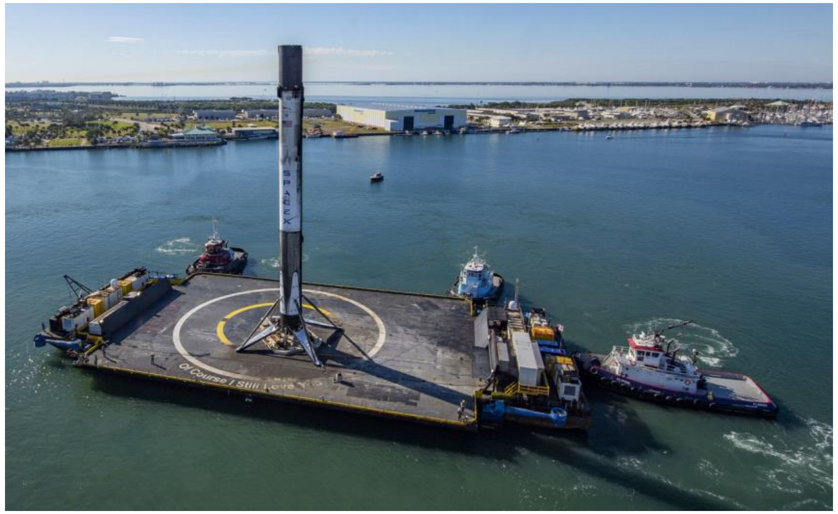
Finally, out of the 4 classification models that were built, the Decision Tree classifier performed the best with the given dataset.

Introduction

- **Project background:**

  SpaceX advertises Falcon 9 rocket launches with a cost of $62 million USD; other providers cost upward of $165 million USD each. Much of the savings is because SpaceX can reuse the first stage.

  Hence, predicting if the first stage will land makes it possible to determine the cost of a launch, and potentially bid against SpaceX.



- **Project goal:**

  This project aims at predicting if the Falcon 9 first stage will land successfully.

Section 1

Methodology

Methodology

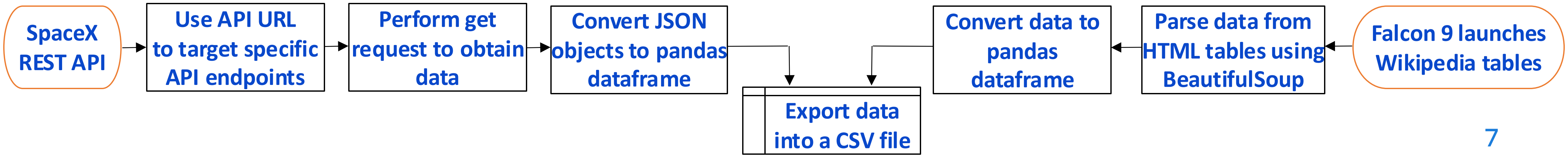
Executive Summary

- **Data collection methodology:**
    - SpaceX REST API – *get request*
    - Web Scraping from Wikipedia page – *Python BeautifulSoup*
- **Perform data wrangling:**
    - Exploring, transforming and validating raw dataset into high-quality data.
- **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 and evaluate classification models – Logistic Regression, SVM, Decision Tree and KNN.

Data Collection

Data sets were collected from 2 sources:

SpaceX REST API	Wikipedia HTML tables
• **Data:** rockets launches data including variables such as booster version, payload mass, orbit, launch site, landing pads, outcome, etc. • **Access:** main SpaceX API url https://api.spacexdata.com/v4 and different endpoints (launches, cores, launchpads, rockets, payloads, etc.). • **Method:** JSON objects are obtained from a get request, normalized into a pandas dataframe and exported into a CSV file.	• **Data:** Falcon 9 launches data including variables such as booster version, launch site, launch and booster landing outcome, orbit, payload mass, etc. • **Access:** Wikipedia page 'List of Falcon 9 and Falcon Heavy launches'. • **Method:** data is parsed from Wikipedia HTML tables using Python BeautifulSoup, then converted into a pandas dataframe and exported into a CSV file.



Data Collection – SpaceX API

GitHub Link

Data collection from SpaceX REST API calls:

1. **Requesting rocket launch data from SpaceX API:**

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

response = requests.get(spacex_url)
```

2. **Decoding the response content as a JSON and converting it into a Pandas dataframe 'data':**

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

3. **Applying some filters to the dataset to select only valuable data:**

```
# Lets take a subset of our dataframe keeping only the features we want and the flight
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
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
data['cores'] = data['cores'].map(lambda x : x[0])
data['payloads'] = data['payloads'].map(lambda x : x[0])
```

4. **Setting up functions to extract information from different API endpoints using identification numbers in the dataset:**

```
# Takes the dataset and uses the rocket column to call the API and append the data to the
def getBoosterVersion(data):
    for x in data['rocket']:
      if x:
        response = requests.get("https://api.spacexdata.com/v4/rockets/"+str(x)).json()
        BoosterVersion.append(response['name'])
```

5. **Calling created functions to get specific data from the different API endpoints:**

```
# Call getBoosterVersion
getBoosterVersion(data)
```

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

6. **Collecting the new data into a dictionary 'launch_dict' and converting it into a pandas dataframe 'launch_df':**

```
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}
```

```
# Create a data from launch_dict
launch_df = pd.DataFrame(launch_dict)
```

7. **Filtering the dataframe to include only Falcon 9 launches and exporting it to a CSV file:**

```
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

```
data_falcon9.to_csv('dataset_part_1.csv',index=False)
```

Data Collection – Scraping

GitHub [Link]

Data collection from Wikipedia page calls:

1. **Requesting the Falcon9 Launch Wiki page from its URL using BeautifulSoup:**

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&
response = requests.get(static_url).text
soup = BeautifulSoup(response,'html5lib')
```

2. **Extracting all column/variable names from the HTML table header:**

2.1 Finding all tables on the wikipage and selecting the third table:

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

2.2 Extracting the columns names one by one:

```
table_headers = first_launch_table.find_all('th')

column_names = []

for row in table_headers:
    column_name = extract_column_from_header(row)
    if (column_name is not None) and (len(column_name) > 0):
        column_names.append(column_name)
```

3. **Creating a dataframe by parsing the launch HTML tables:**

3.1 Creating an empty dictionary 'launch_dict' with keys from the extracted column names:

```
launch_dict = dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each value to be an empty list
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
```

3.2 Filling up the dictionary with the launch records data extracted from table rows:

```
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
            # TODO: 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])
            # Date value
            # TODO: Append the date into launch_dict with key `Date`
            date = datatimelist[0].strip(',')
            launch_dict['Date'].append(date)
            print(date)
```

3.3 Creating a pandas dataframe 'df' from the 'launch_dict' dictionary and exporting it to a CSV file:

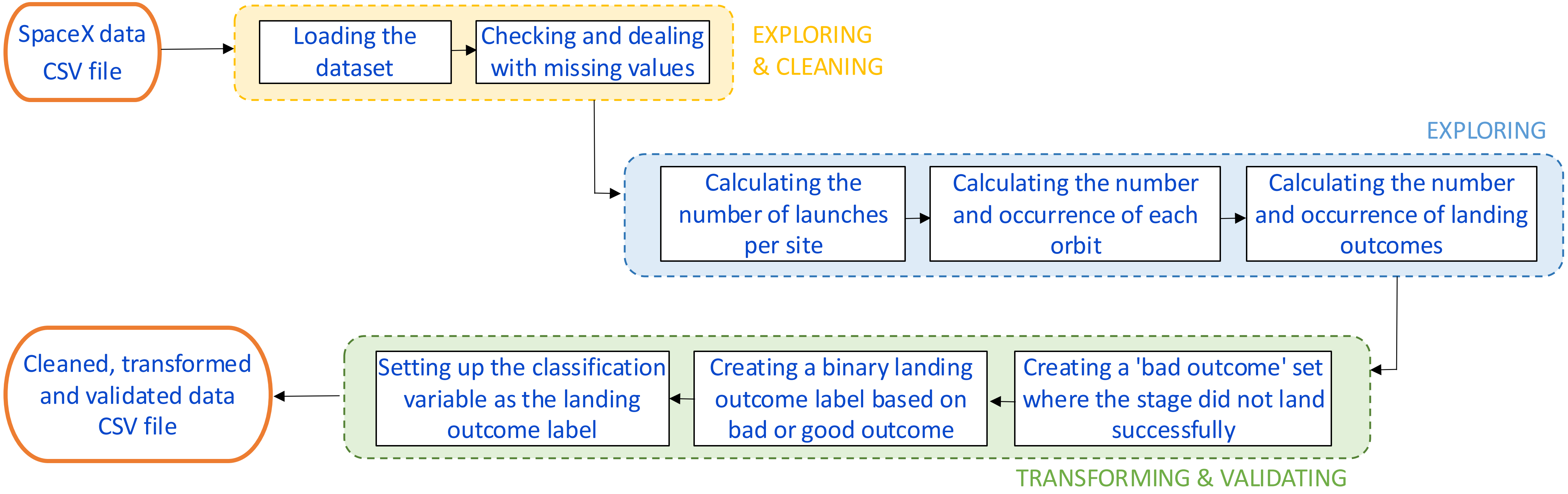
```
df = pd.DataFrame({key: pd.Series(value) for key, value in launch_dict.items()})
```

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

Data Wrangling

GitHub [Link]

Data wrangling process: exploring, transforming, and validating raw datasets into high-quality data.



EDA with Data Visualization

GitHub Link

Charts that were plotted and why:

- **Scatter plot (catplot)**: to visualize the relationship between Flight Number and Launch Site.
- **Scatter plot (catplot)**: to visualize the relationship between Payload Mass and Launch Site.
- **Bar chart**: to visualize the relationship between success rate of each orbit type.
- **Scatter plot (catplot)**: to visualize the relationship between FlightNumber and Orbit type.
- **Scatter plot (catplot)**: to visualize the relationship between Payload Mass and Orbit type.
- **Line chart**: to visualize the launch success yearly trend.

EDA with SQL

GitHub Link

SQL queries that were performed:

- 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 boosters launched by NASA (CRS).
- Displaying the average payload mass carried by booster version F9 v1.1.
- Listing the date when the first succesful landing outcome in ground pad was achieved.
- Listing the names of the successful boosters in drone ship and have payload mass between 4000 and 6000 kg.
- Listing the total number of successful and failure mission outcomes.
- Listing the names of the booster versions which have carried the maximum payload mass, using a subquery.
- Listing the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for the year 2015.
- Ranking 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.

Build an interactive map with Folium

GitHub Link

Map objects such as markers, circles, lines, etc. that were created and added to a folium map:

- **Folium.Circle** and **folium.Marker** for each launch site on the site map to locate the launch sites.
- Added a **folium.Marker** with its coordinate to a **Marker_cluster** for each launch result in the dataframe and customized the **Marker's icon property** to indicate if this launch was successful or failed.
- **MousePosition** on the map to get coordinate for a mouse over a point on the map.
- **Folium.Marker** on a selected closest coastline point to a launch site on the map to evaluate its distance from the launch site.
- **Folium.PolyLine** between a launch site and the selected coastline point to help visualize the distance.
- **Folium.Marker** at the selected closest city, railroad and highway for each launch site to indicate the distance between each launch site and its proximities, and drew **lines** between them to visualize the distances.

Build a Dashboard with Plotly Dash

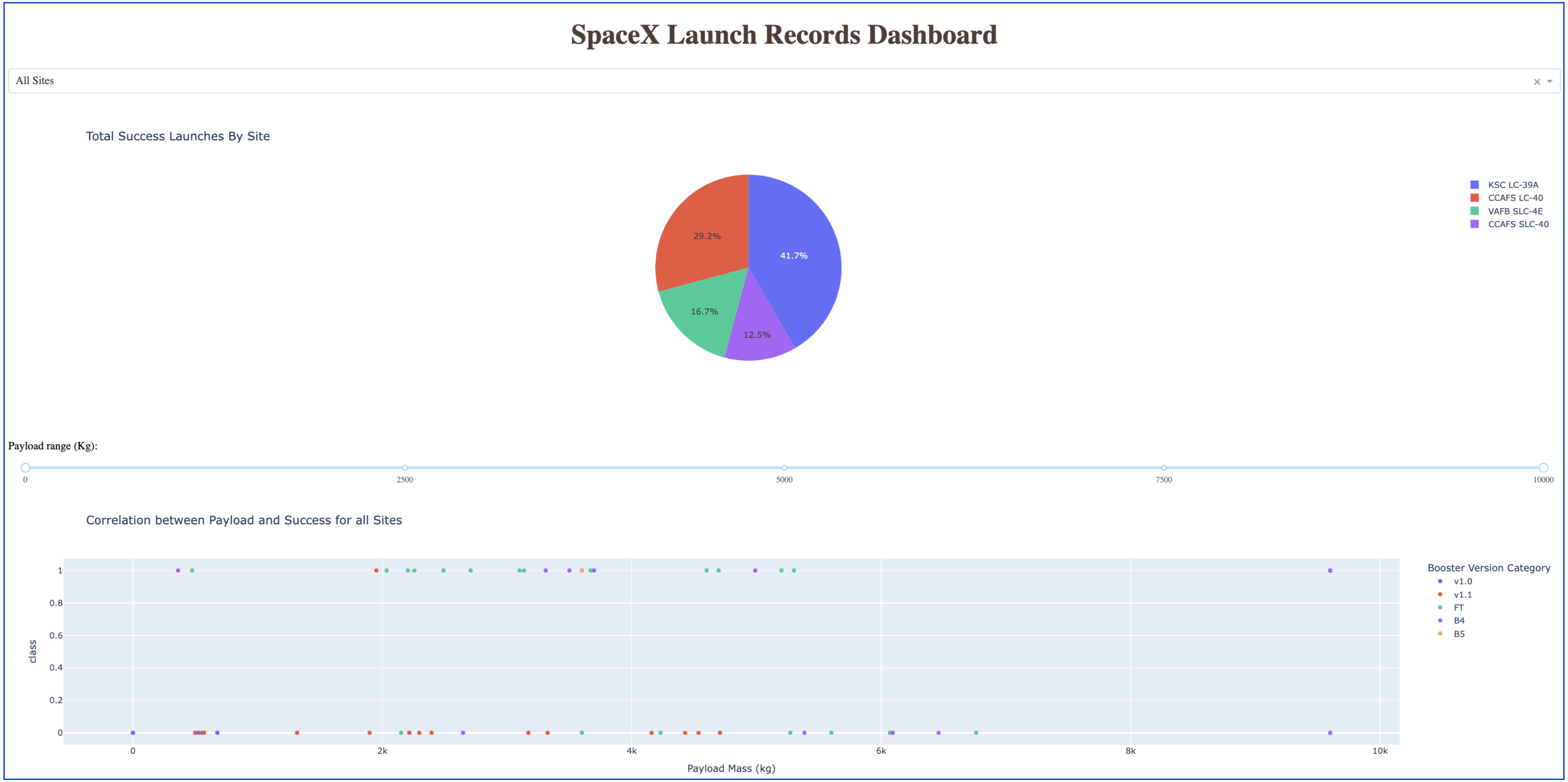
GitHub [Link]

Plots/graphs and interactions that were added to the dashboard:

- **Launch Site Drop-down Input Component**: to select one specific site and check its detailed success rate.
- **Callback function**: to get the selected launch site from the site drop-down and render a **pie chart** visualizing launch success counts.
- **Range Slider**: to easily select different payload ranges and see if we can identify some visual patterns.
- **Callback function**: to render the success-payload-scatter-chart scatter plot to visually observe how payload may be correlated with mission outcomes for selected site(s).
  - **Color-labeled Booster version** on each scatter point to observe mission outcomes with different boosters.

Build a Dashboard with Plotly Dash

GitHub Link

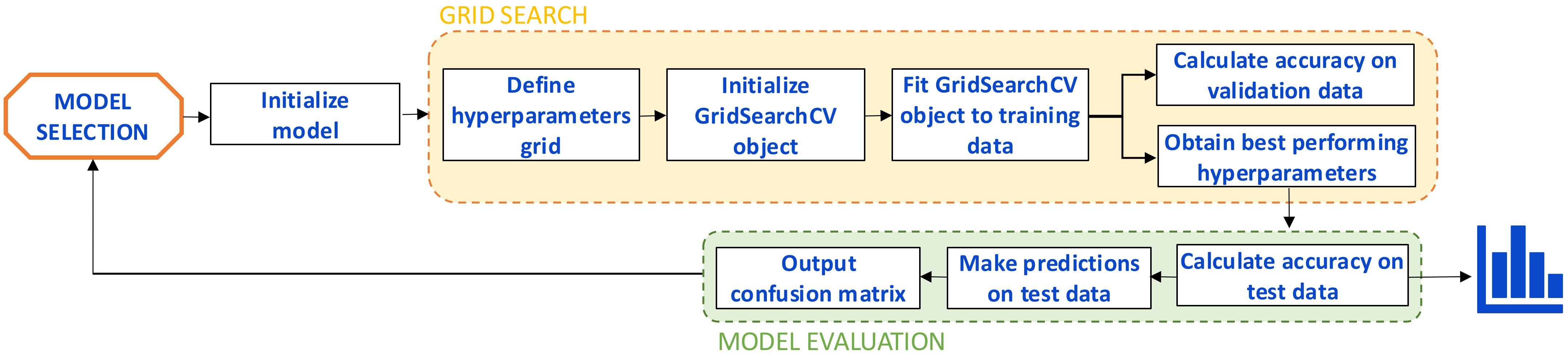


Predictive Analysis (Classification)

GitHub Link

Model development process:

1. Loading the dataset and creating a column for the class.
2. Preprocessing data: to standardize the data.
3. Splitting the data into training and testing data using train_test_split function (test_size=0.2): to evaluate how well the model performs on the data it was not trained on.
4. Training the model (Logistic Regression, Support Vector Machine, Decision Tree Classifier, K-nearest Neighbors):



5. Making a bar chart of the accuracy on the test data of each model: to compare and select the best performing model.

Results

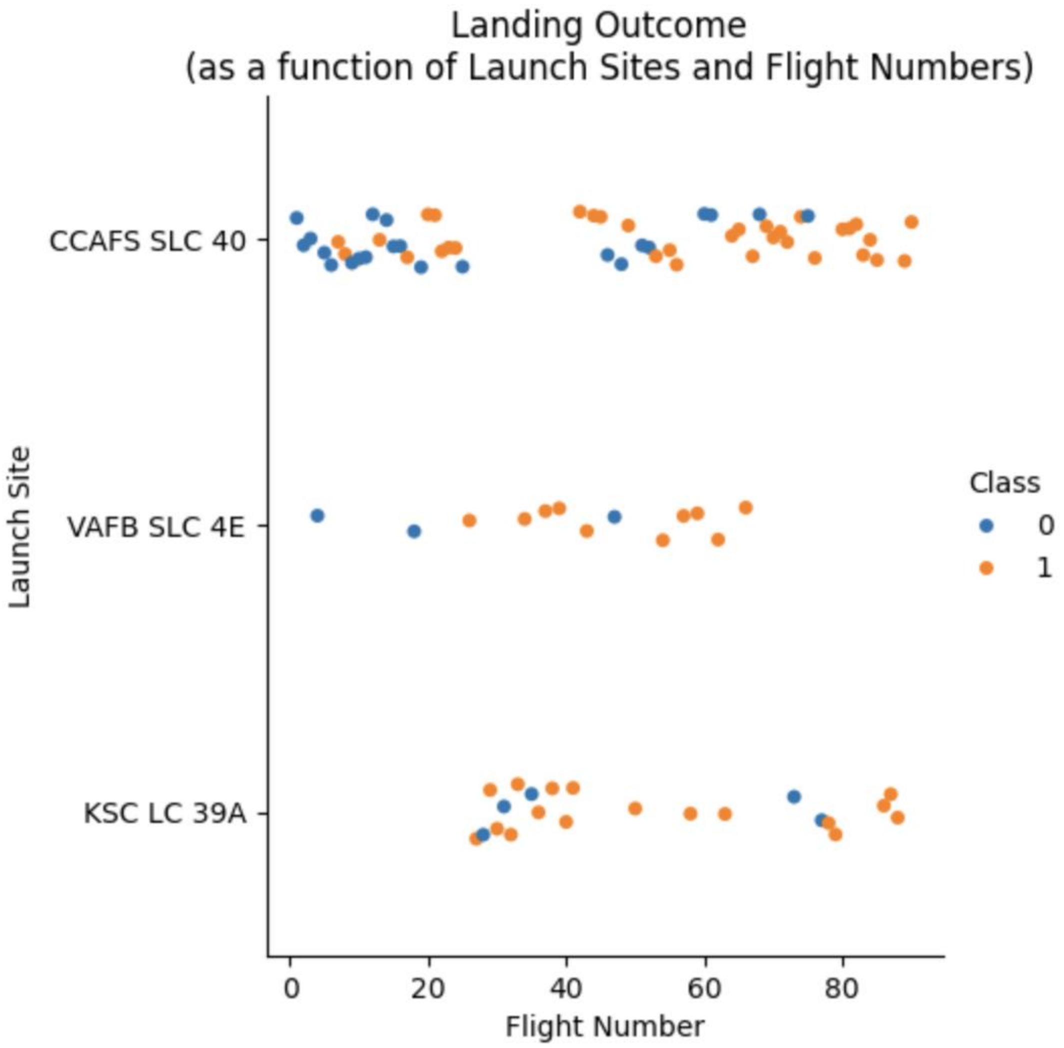
Executive Summary

- **Exploratory data analysis results**
- **Interactive analytics demo (in screenshots):**
    - Launch sites proximities analysis with Folium maps
    - Launch outcomes analysis with Plotly Dashboard
- **Predictive analysis (classification) results**

Section 2

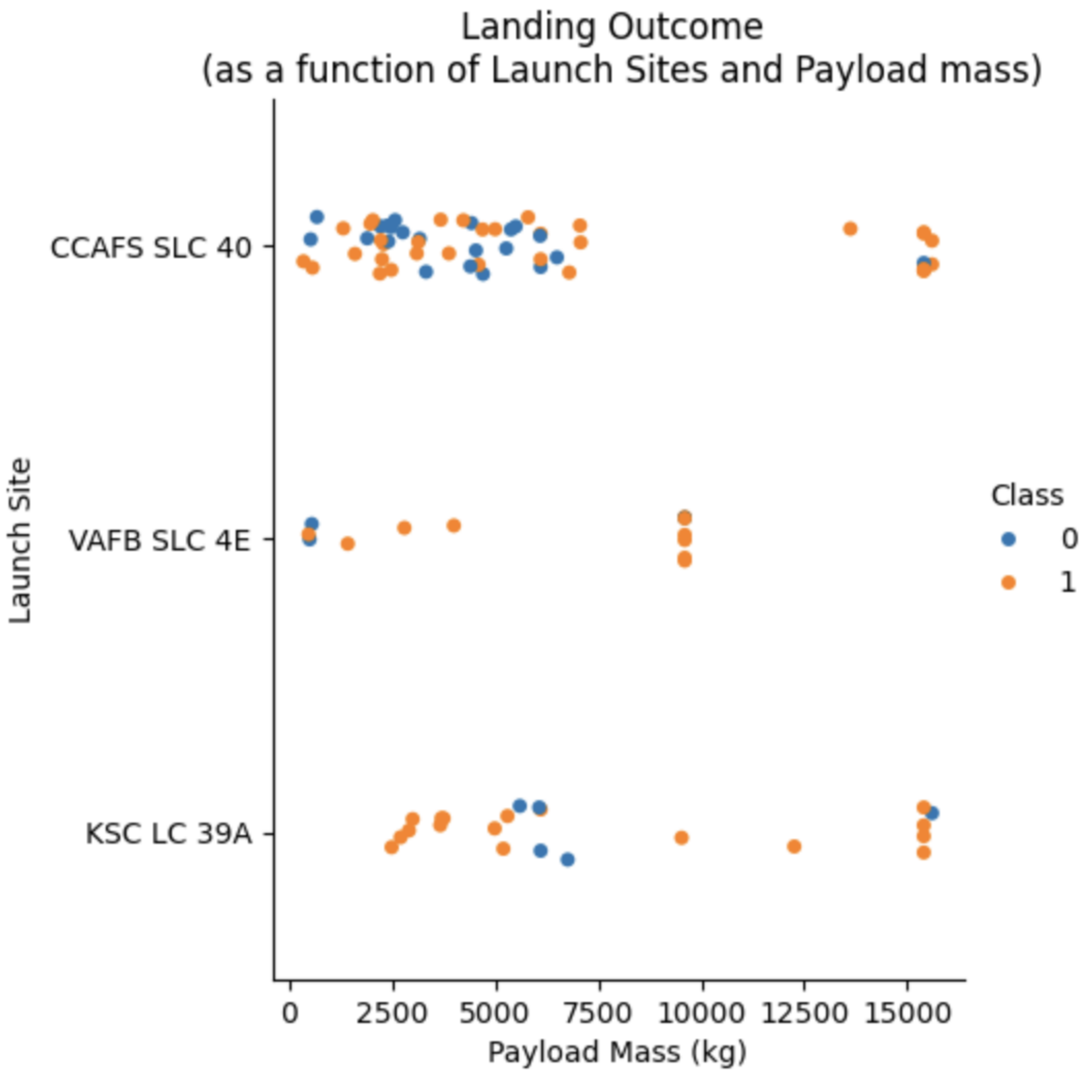
Insights drawn from EDA

Flight Number vs. Launch Site



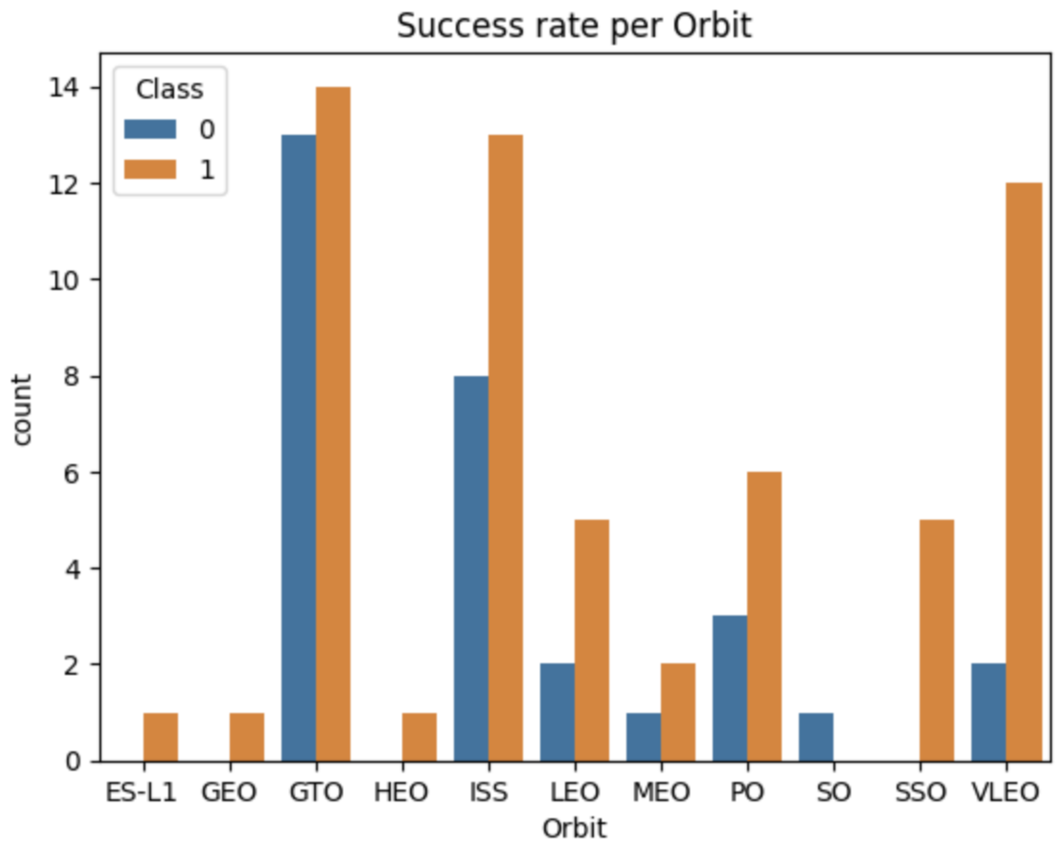
- Most of the flights departed from the CCAFS SLC 40 launch site.
- The last 30 flights were launched from CCAFS SLC 40 and KSC LC 39A launch sites.
- Most of the unsuccessful landings occured at the CCAFS SLC 40 launch site, although it seems like the most recent flights at this location had better outcomes.
- VAFB SLC 4E and KSC LC 39A have a better success rate.

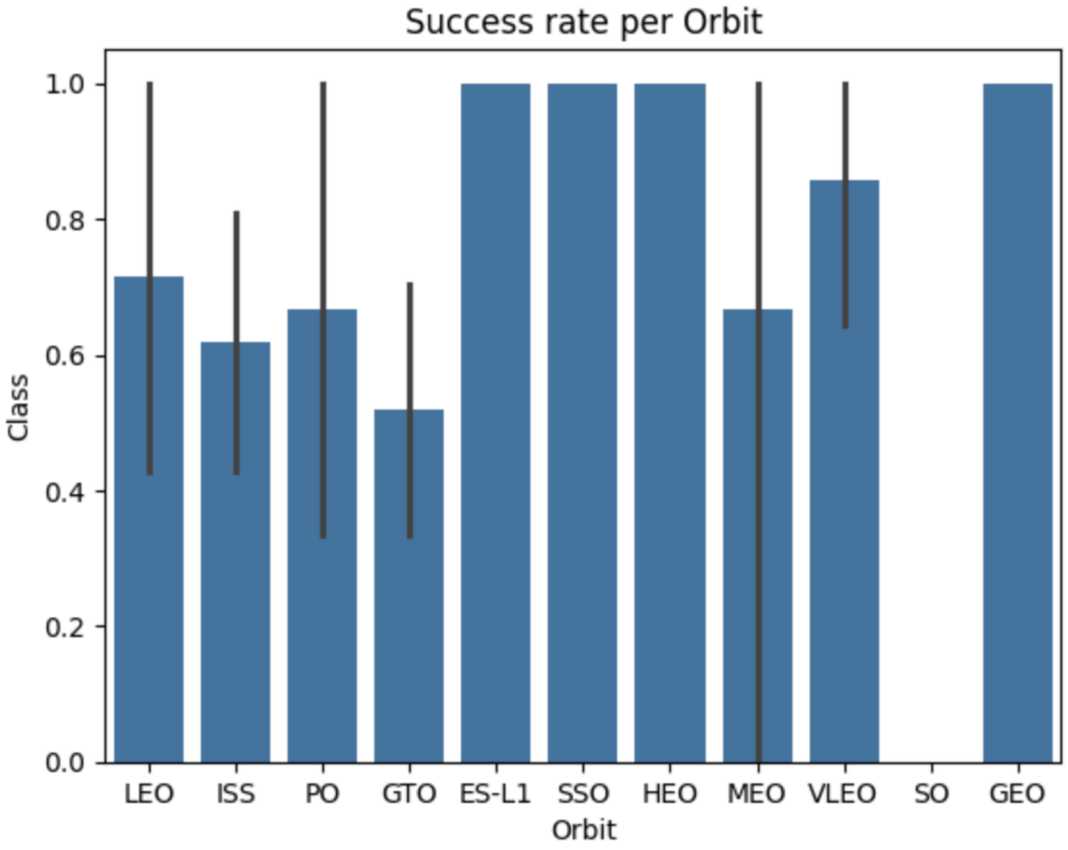
Payload vs. Launch Site



- No heavy payload mass rockets (greater than 10000 kg) were launched from the VAFB SLC 4E launch site.
- No payload mass rockets between 7500 and 12500 kg were launched from the CCAFS SLC 40 launch site.
- CCAFS SLC 40 and KSC LC 39A launch sites seem to have a better outcome with heavier payload mass rockets than with lighter ones.

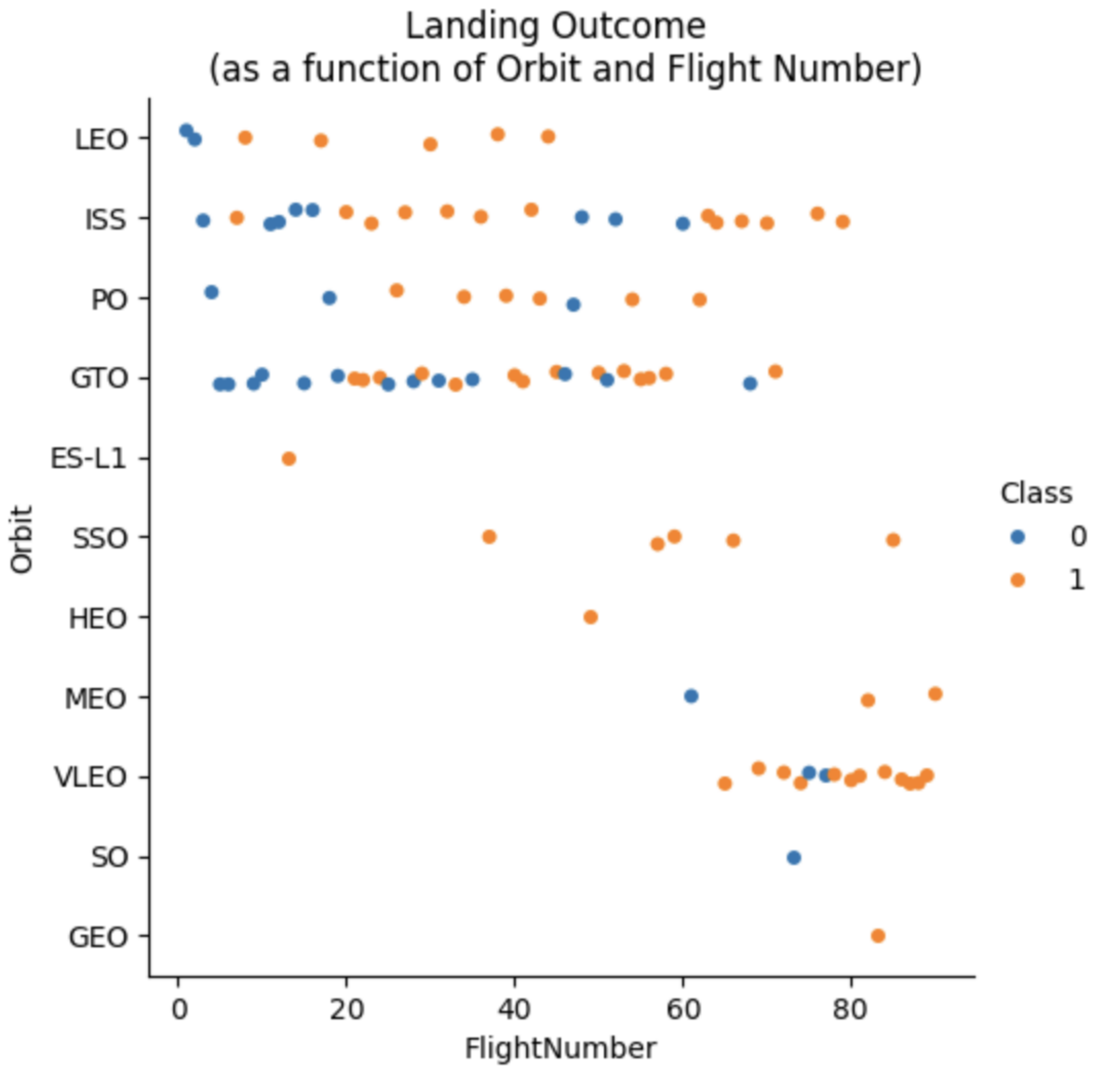
Success Rate vs. Orbit Type





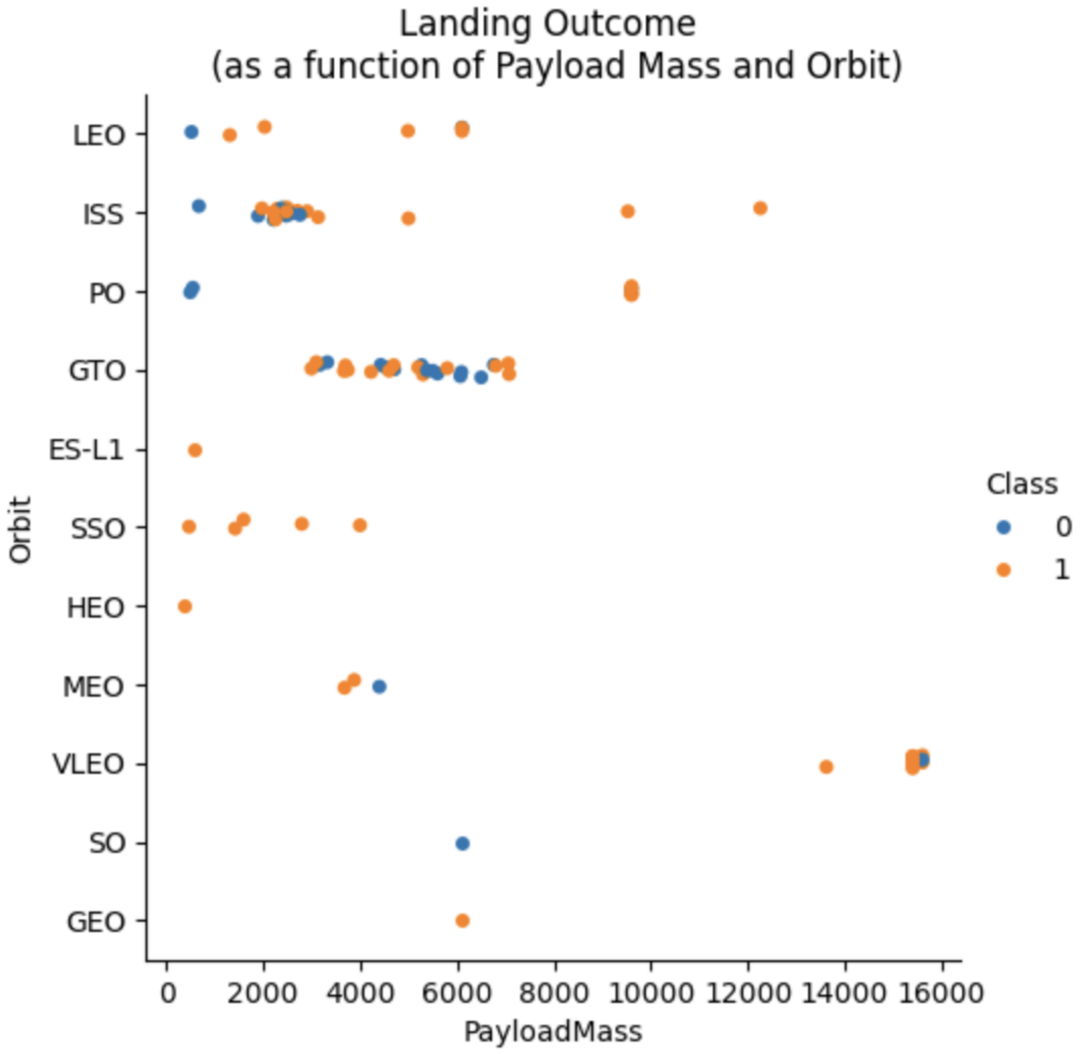
- ES-L1, GEO, HEO and SSO have the highest overall success rate (100%), while SO has the lowest (0%).
- VLEO has a high success rate (> 80%). LEO, PO, MEO and ISS have a descent success rate (between 60-70%), and GTO has a success rate close to 50%.
- Considering the launches count, VLEO and SSO appear to have the most reliable success rate.

Flight Number vs. Orbit Type



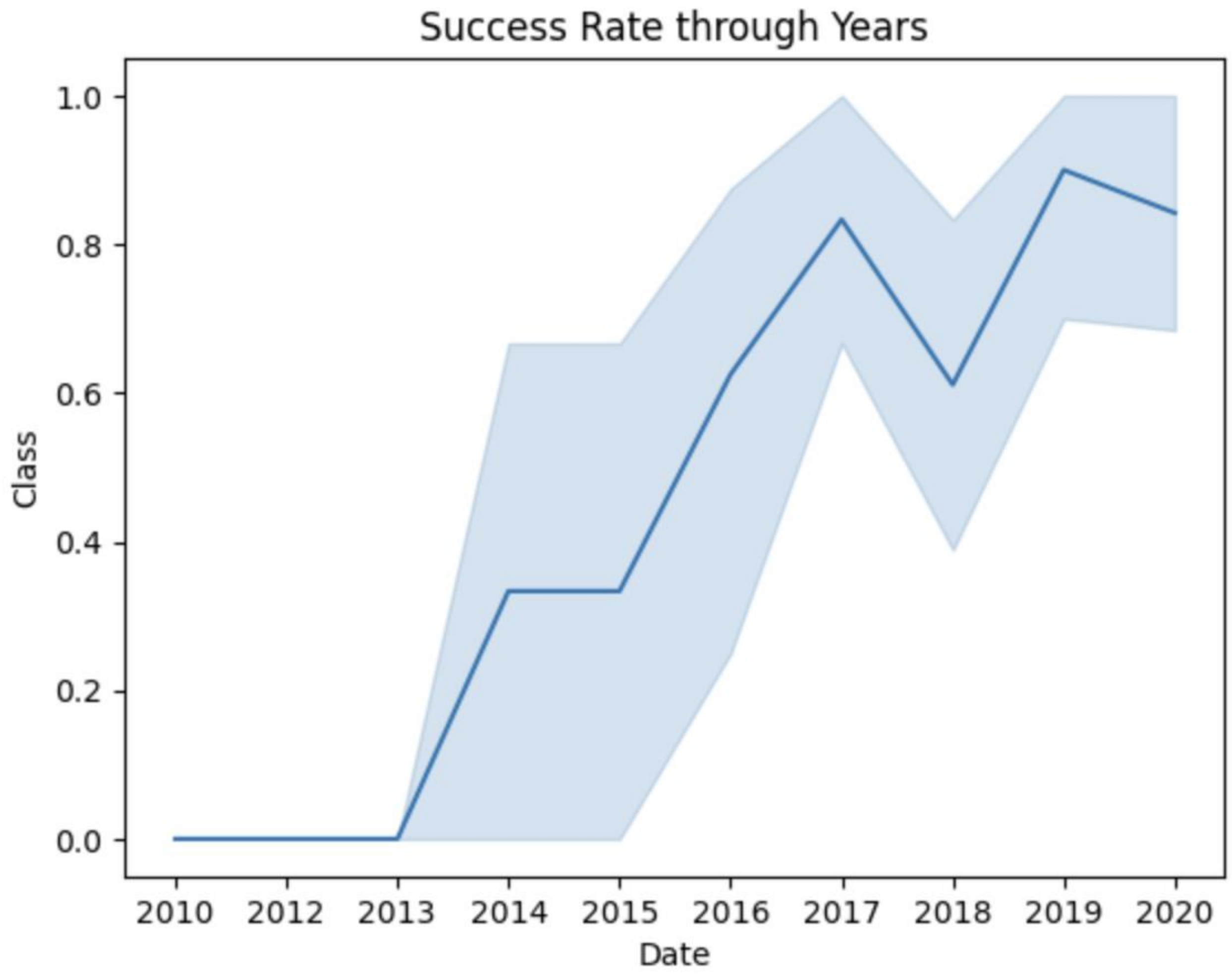
- Earlier flights were sent to LEO, ISS, PO, GTO and ES-L1 orbits.
  Flights to SSO, HEO, MEO, VLEO, SO and GEO orbits are more recent.
- Outcome has improved with the number of flights in LEO orbit: success seems to be related to the number of flights; which is not the case in GTO orbit.
  Flights to ISS and PO orbits seem to have a similar alterning outcome evolution.
- The group of more recent orbit flights (SSO, HEO, MEO, VLEO, GEO) appear to have a better overall success rate.

Payload vs. Orbit Type



- No heavy payload mass rockets (> 8000 kg) are launched to LEO, GTO, ES-L1, SSO, HEO, MEO, SO and GEO orbits.
- Only very heavy payload mass rockets (greater than 13000 kg) are launched to VLEO orbits.
- ISS and PO orbits have a broader range of payload mass rockets.
- LEO, ISS and PO orbits have a better outcome with heavier payload mass rockets than with lighter ones.
- GTO orbit has a mixed outcome throughout its payload mass range.

Launch Success Yearly Trend



The success rate has been increasing since 2013 (with a slight decrease around 2017-2018), probably as a result of the gained experience and technological improvements.

All Launch Site Names

- Query:

```
%sql select distinct "Launch_Site" from SPACEXTABLE
```

- Names of the unique launch sites:

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

Launch Site Names Beginning with 'CCA'

- Query:

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

- 5 records where launch sites begin with 'CCA':

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass from NASA (CRS)

- Query:

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

- Total payload mass carried by boosters from NASA (CRS):

sum("PAYLOAD_MASS__KG_")
45596

Average Payload Mass by F9 v1.1

- Query:

```
%sql select avg("PAYLOAD_MASS__KG_") from SPACEXTABLE where "Booster_Version"="F9 v1.1"
```

- Average payload mass carried by booster version F9 v1.1

avg("PAYLOAD_MASS__KG_")
2928.4

First Successful Ground Landing Date

- Query:

```
%sql select Date from SPACEXTABLE where "Landing_Outcome"="Success (ground pad)" limit 1
```

- Date of the first successful landing outcome on ground pad:

Date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000 kg

- Query:

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

- Names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000 kg:

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

Total Number of Successful and Failure Mission Outcomes

- Query:

```
%sql select "Mission_Outcome", count("Mission_Outcome") as Count from SPACEXTABLE GROUP BY "Mission_Outcome"
```

- Total number of successful and failure mission outcomes:

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

Boosters Carried Maximum Payload

- Query:

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

- Names of the boosters which have carried the maximum payload mass:

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

2015 Launch Records – Failed Drone Ship

- Query:

```
%%sql select case cast(substr(Date,6,2) as integer)
    when 1 then 'January'
    when 2 then 'February'
    when 3 then 'March'
    when 4 then 'April'
    when 5 then 'May'
    when 6 then 'June'
    when 7 then 'July'
    when 8 then 'August'
    when 9 then 'September'
    when 10 then 'October'
    when 11 then 'November'
    else 'December' end as Month,
    "Landing_Outcome","Booster_Version","Launch_Site" from SPACEXTABLE where substr(Date,0,5)='2015' and "Landing_Outcome"="Failure (drone ship)"
```

- List of the failed landing outcomes in drone ship, their booster versions, their launch site names and the corresponding month names for the year 2015:

Month	Landing_Outcome	Booster_Version	Launch_Site
January	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
April	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Ranking Landing Outcomes between 2010-06-04 and 2017-03-20

- Query:

```
%%sql select "Landing_Outcome",count("Landing_Outcome") as Count from SPACEXTABLE
    where "Date" <= '3/20/2017' GROUP BY "Landing_Outcome" ORDER BY count("Landing_Outcome") DESC
```

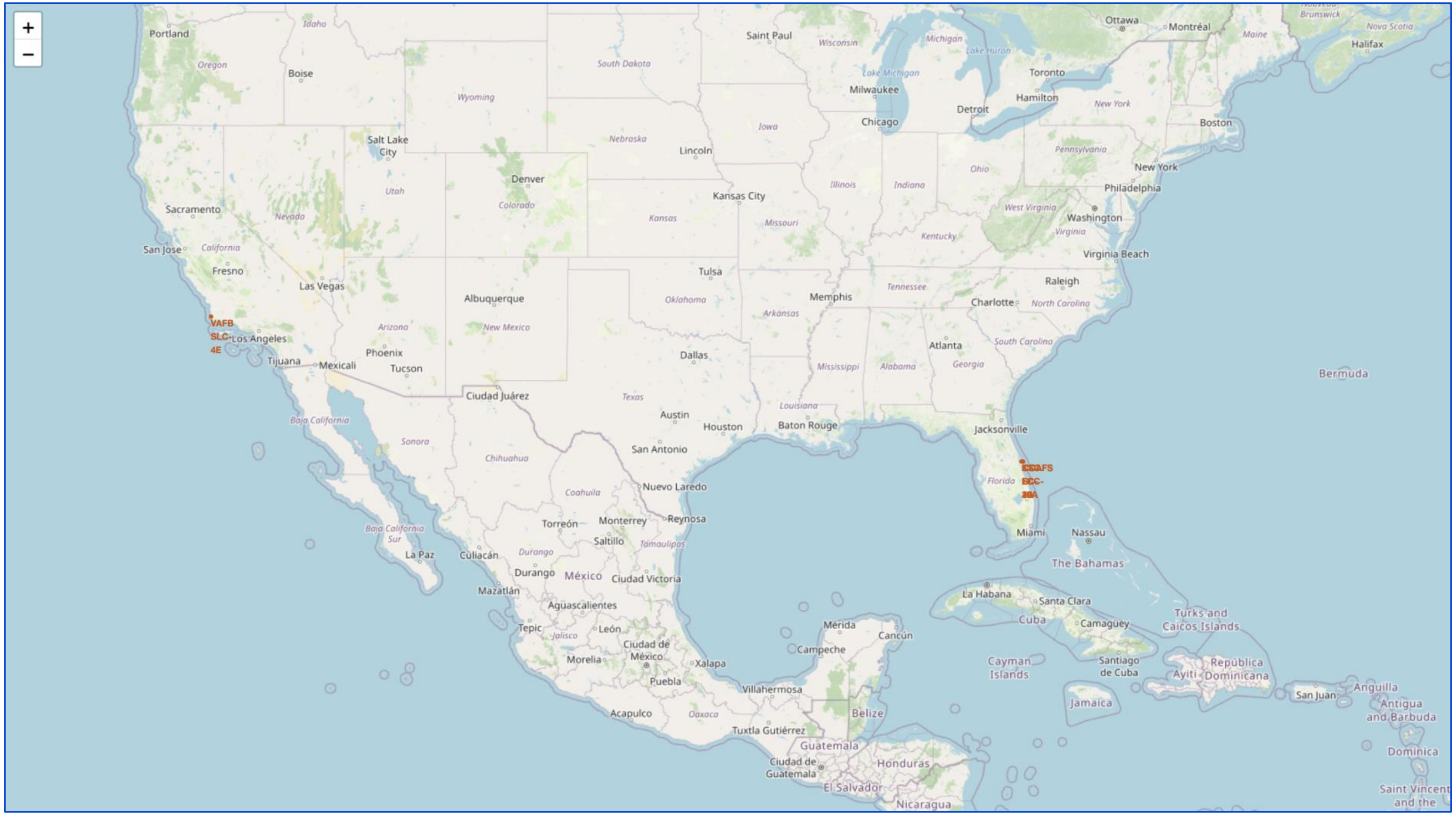
- Ranking of the landing outcomes count between the date 2010-06-04 and 2017-03-20, in descending order:

Landing_Outcome	Count
Success	38
No attempt	21
Success (drone ship)	14
Success (ground pad)	9
Failure (drone ship)	5
Controlled (ocean)	5
Failure	3
Uncontrolled (ocean)	2
Failure (parachute)	2
Precluded (drone ship)	1
No attempt	1

Section 3

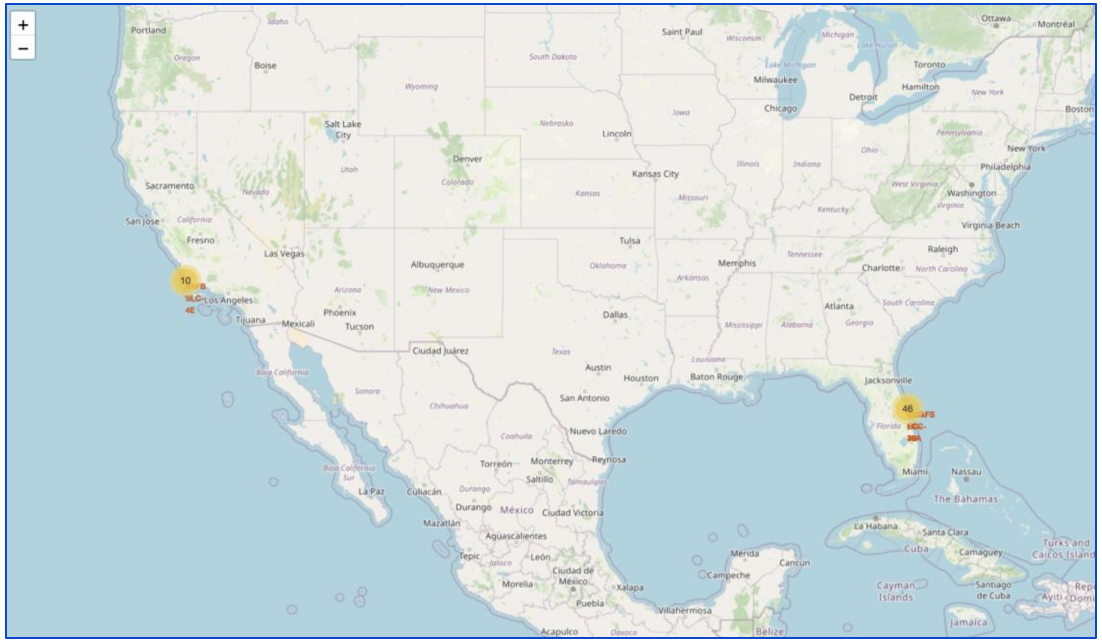
Launch Sites Proximities Analysis

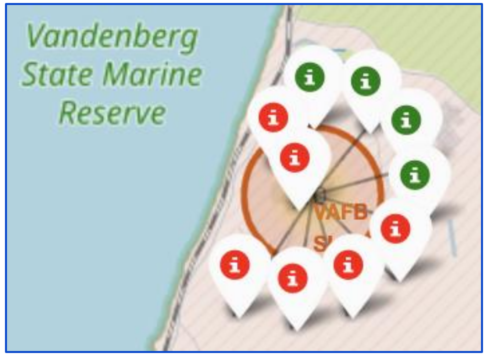
Folium Map 1: Launch Sites Location



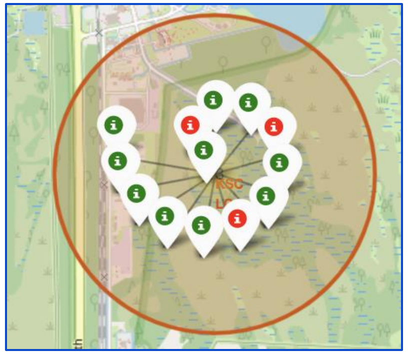
- **VAFB SLC-4E** is located on the southern coast of California.
- **KSC LC-39A**, **CCAFS SLC-40** and **CCAFS LC-40** are located on the eastern coast of Florida.
- The Florida launch sites are closer to the Equator line (28.5°N) than the California site (34.6°N).
- All sites are in **very close proximity to the coast**, with KSC LC-39A being a bit more inland, which could explain why KSC LC-39A has a better landing outcome (success rate = 0.77) since it might be less exposed to marine conditions.

Folium Map 2: Launch Outcomes Per Site



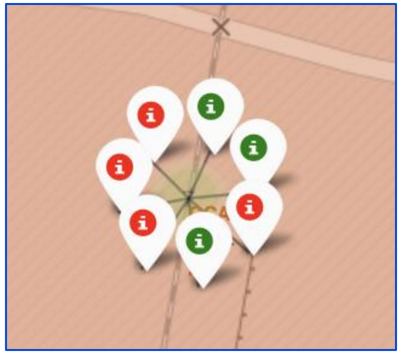


VAFB SLC-4E Site:
4 Success / 6 Failure
40%



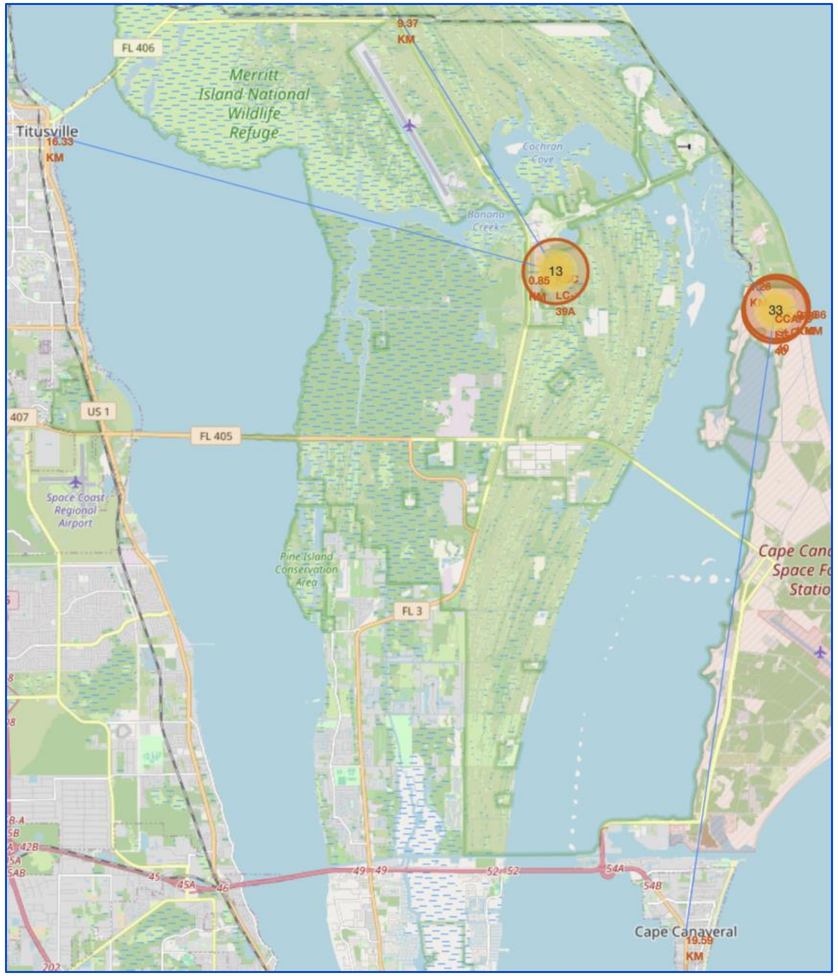
KSC LC-39A Site:
10 Success / 3 Failure
77%

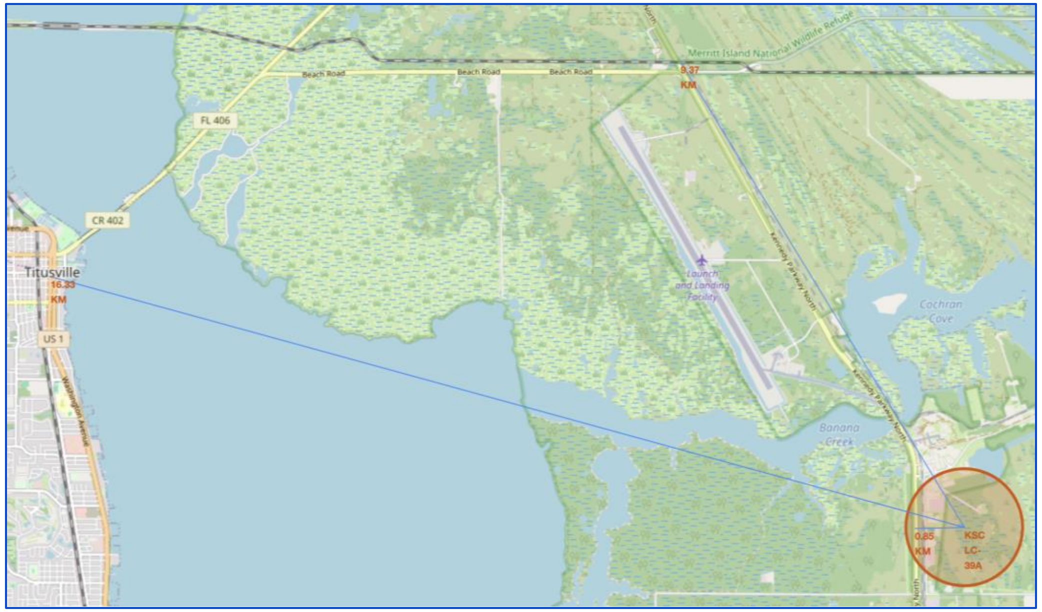
CCAFS LC-40 Site:
7 Success / 19 Failure
27%

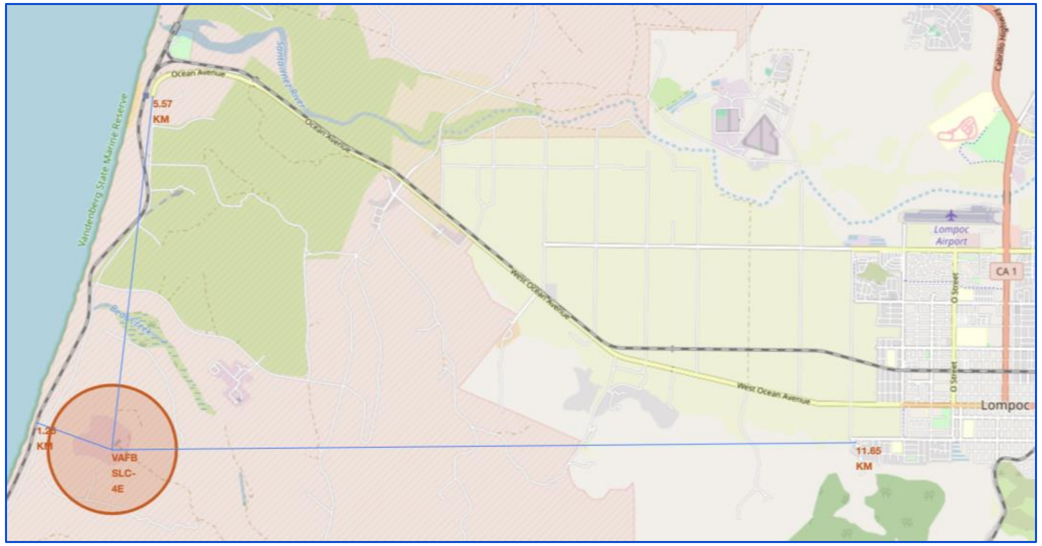


CCAFS SLC-40 Site:
3 Success / 4 Failure
43%

Folium Map 3: Launch Sites Proximities





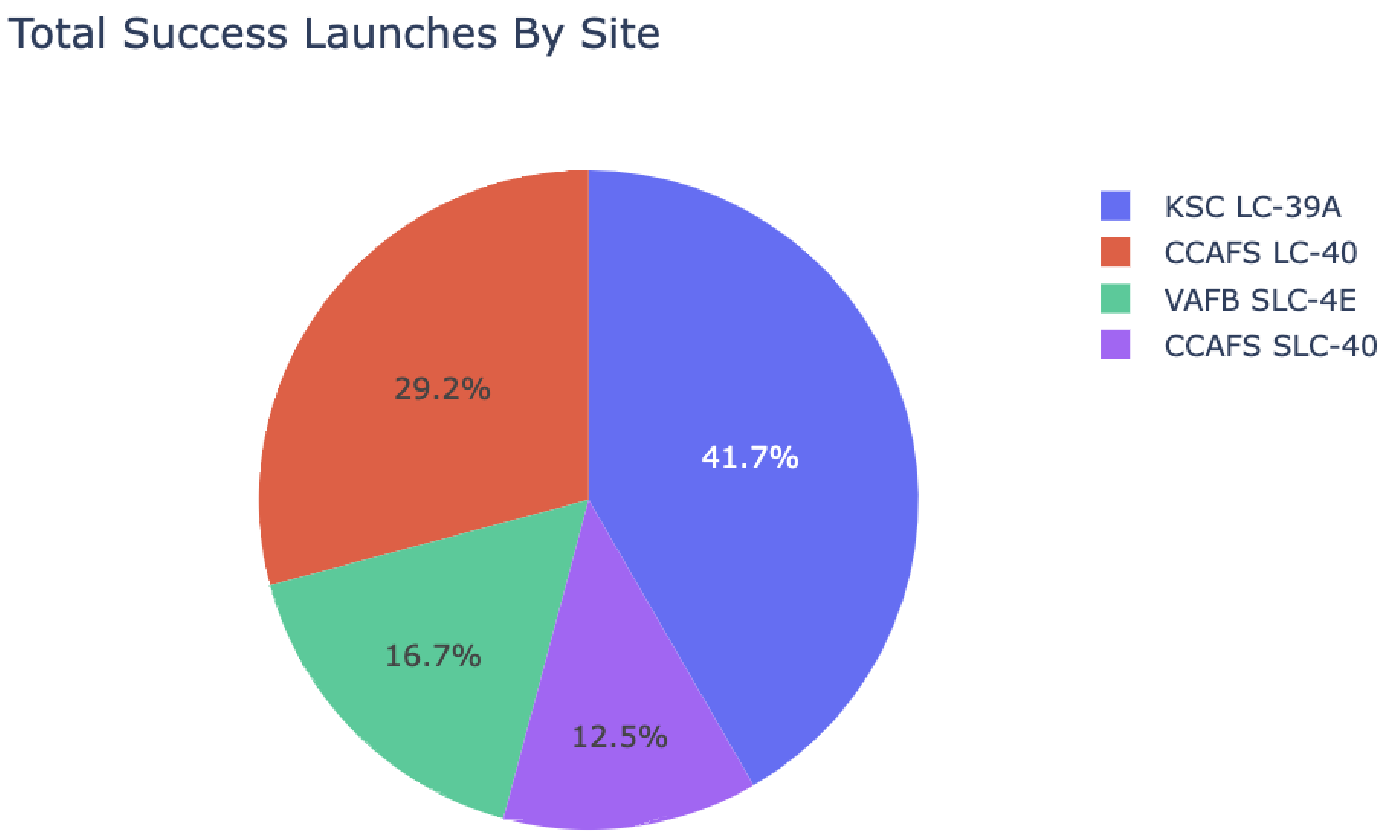


- **Railroad:** Sites are close to railroads (<1.3 km, except for KSC LC-39A <10 km) for convenience and ease of transportation.
- **Highway:** Sites are close to highways (<1 km, except for VAFB SLC-4E <6 km).
- **Coastline:** Sites are close to the coastline probably for ease of access, cooling and reduced exposure to land obstacles and population.
- **Cities:** Sites keep some distance from the cities (> 11 km) to lower the risk of incident.

Section 4

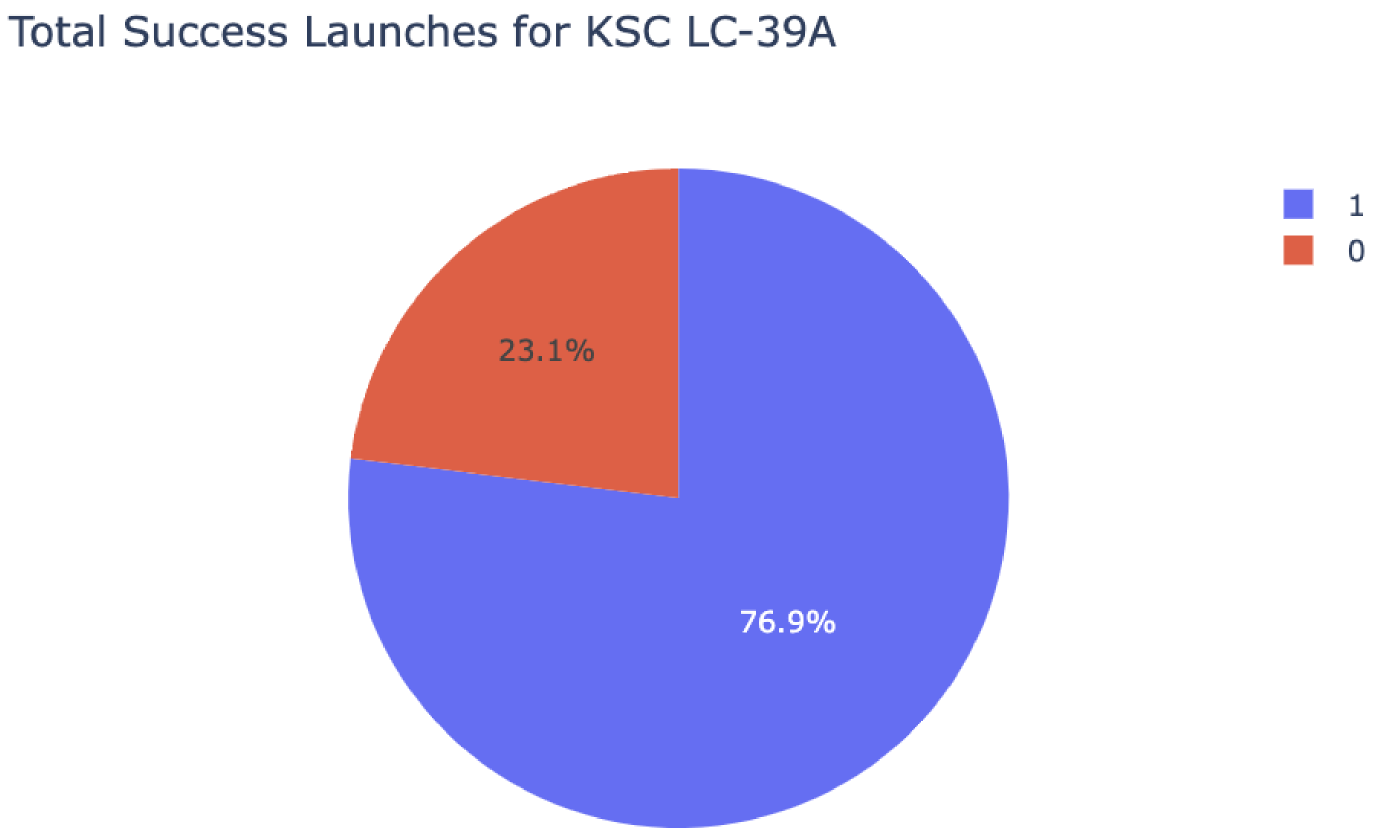
Build a Dashboard with Plotly Dash

Dashboard: All Sites Launch Success Count



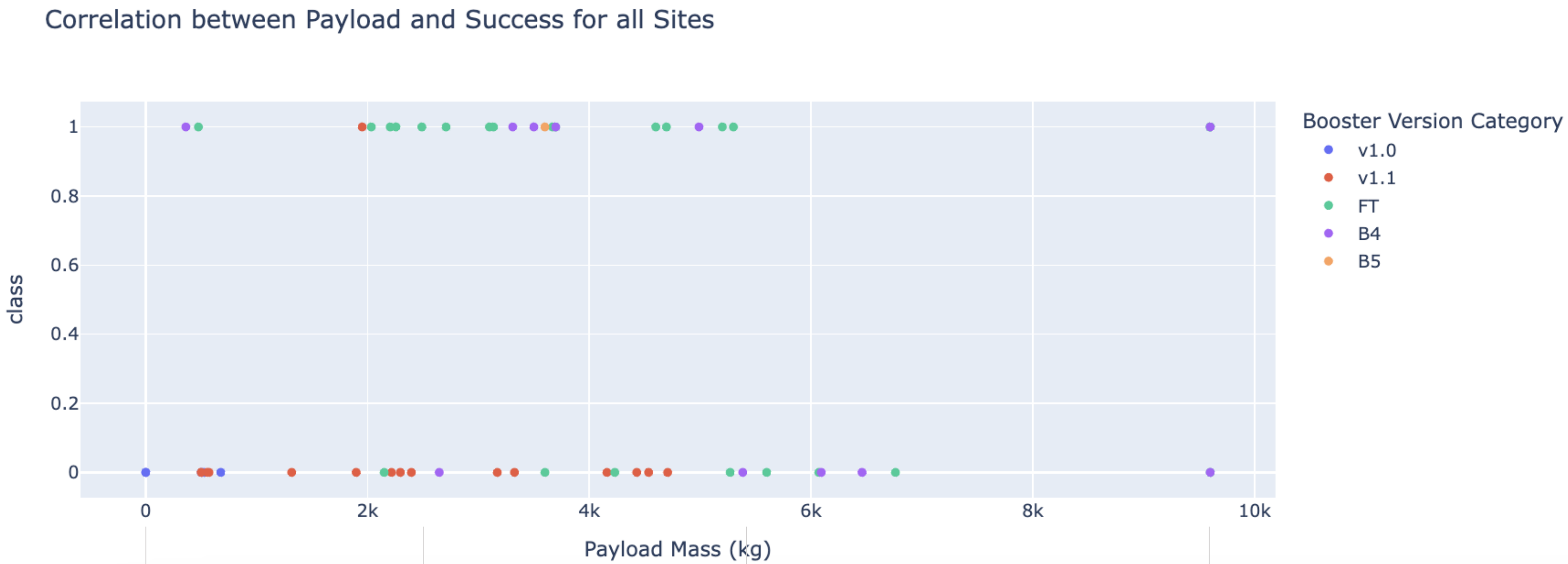
KSC LC-39A has the largest number of successful launches, representing 41.7% of all the successful launches.

Dashboard: Highest Launch Success Rate Site



KSC LC-39A has the highest launch success rate of all sites, with 76.9% of success.

Dashboard: Payload vs. Launch Outcome (1/3)



The **overall success rate** over the entire payload range for all sites is **42.6%**.

3 main payload ranges:

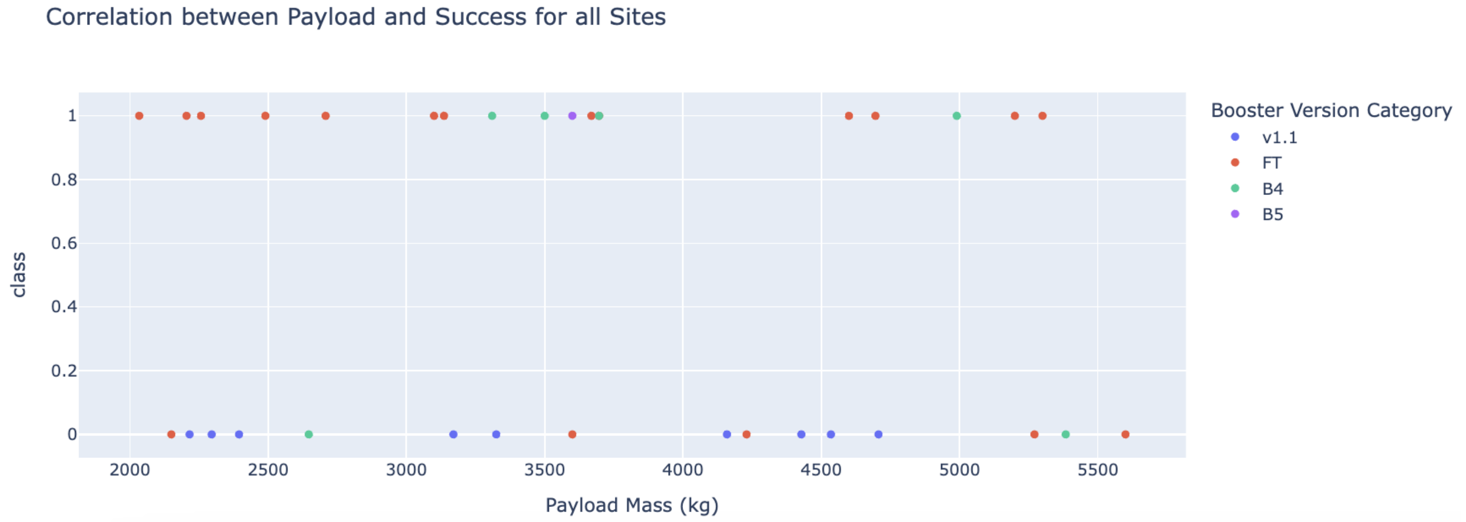
Light	**Medium**	**Heavy**
[0-2395] kg	[2490-5300] kg	[5384-9600] kg

- **Total launches:** most of the launches are in the **medium (46.3%)** and **light (35.2%)** payload ranges.
- **Total successful launches:** most of the successful launches are in the **medium payload range (65.2%)**.
- **Boosters outcome:**

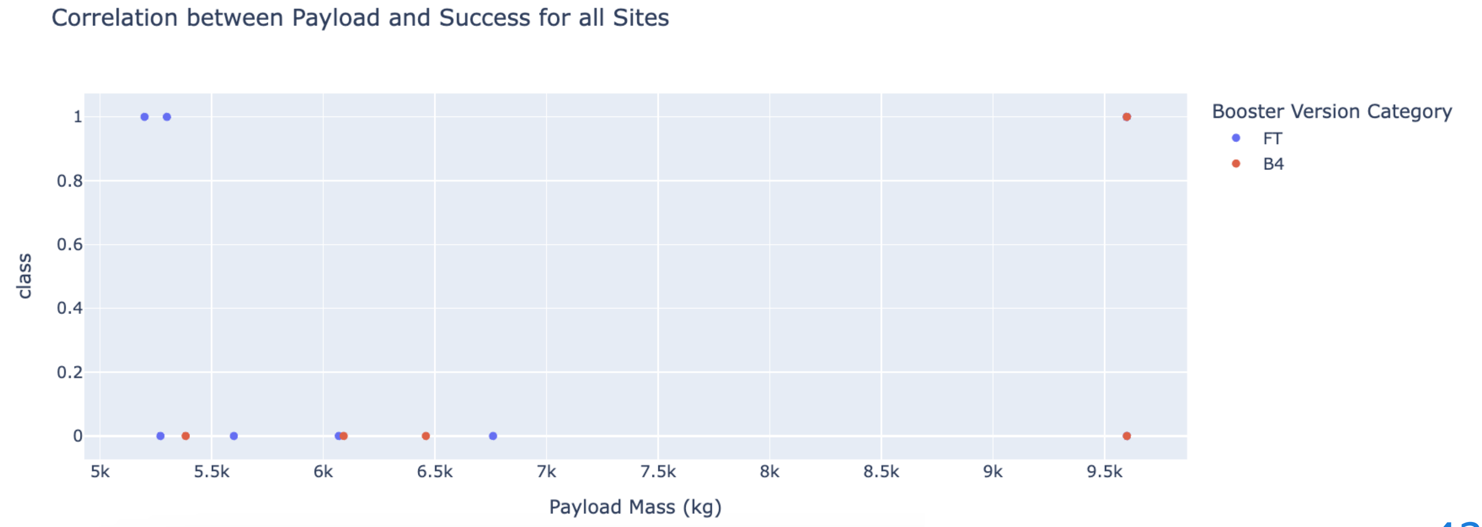
	Light range	**Medium range**	**Heavy range**
Most used	v1.1 (47.4%)	FT (52%)	FT (50%) and B4 (50%)
Best ratio	FT (80%) and B4 (100%)	B5 (100%) and FT (76.9%)	FT (20%) and B4 (20%)
Worst ratio	v1.0 (0%) and v1.1 (11.1%)	V1.1 (0%)	-
Highest # success	FT (66.7%)	FT (66.7%)	FT (50%) and B4 (50%)
Highest # failure	v1.1 (61.5%)	v1.1 (60%)	-

Dashboard: Payload vs. Launch Outcome (2/3)

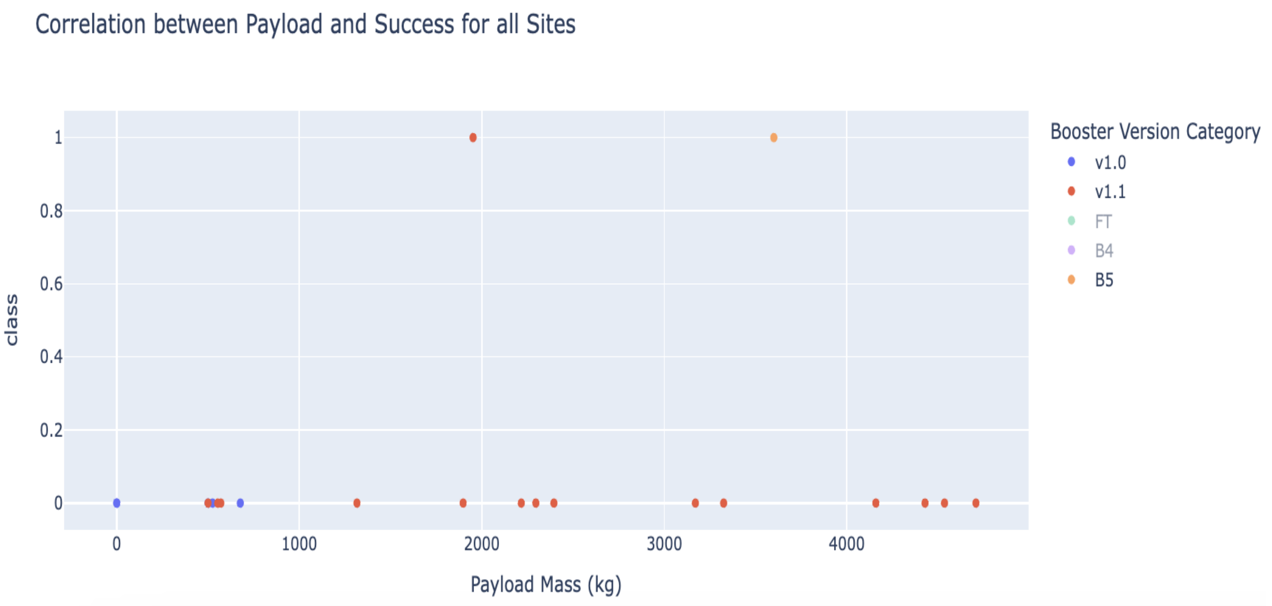
The payload range **[2490-5300 kg]** has the highest launch success rate: **60%.**



The payload ranges **[0-2395 kg]** and **[5384-9600kg]** have the lowest launch success rates: **31.6%** and **20%** respectively.



Dashboard: Payload vs. Launch Outcome (3/3)

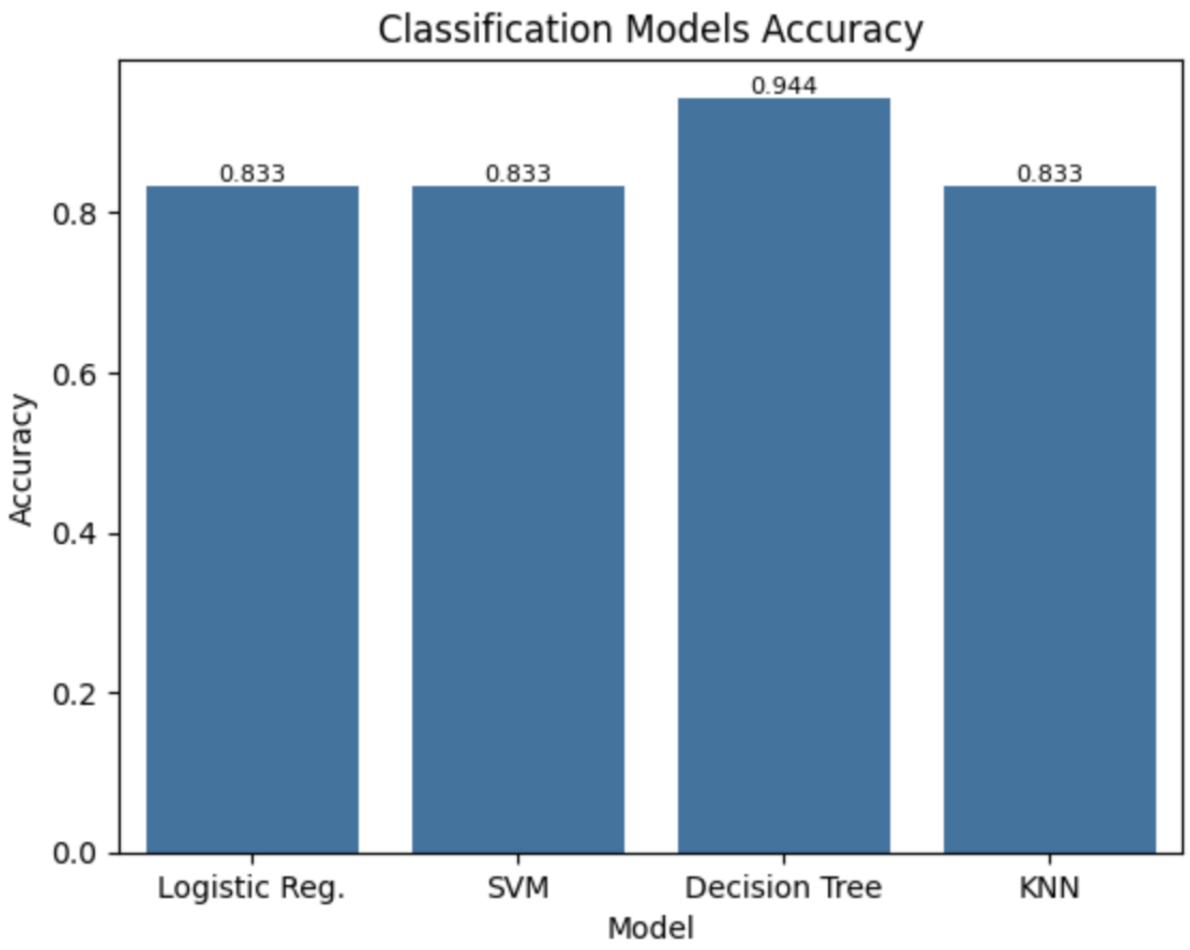


- **v1.0, v1.1 and B5** boosters are mostly used over light and medium payload ranges.
- **v1.0** is the **least successful** booster (0% of successful launches), and **v1.1** has the **highest number of failed** launches (45.2% of all failed launches).
- **FT and B4** boosters are used over all payload ranges, but are more successful over lighter payload ranges.
- **FT** is the most used (42.6% of all launches) and the **most successful** booster across all payload ranges (65.2% of all successful launches).

Section 5

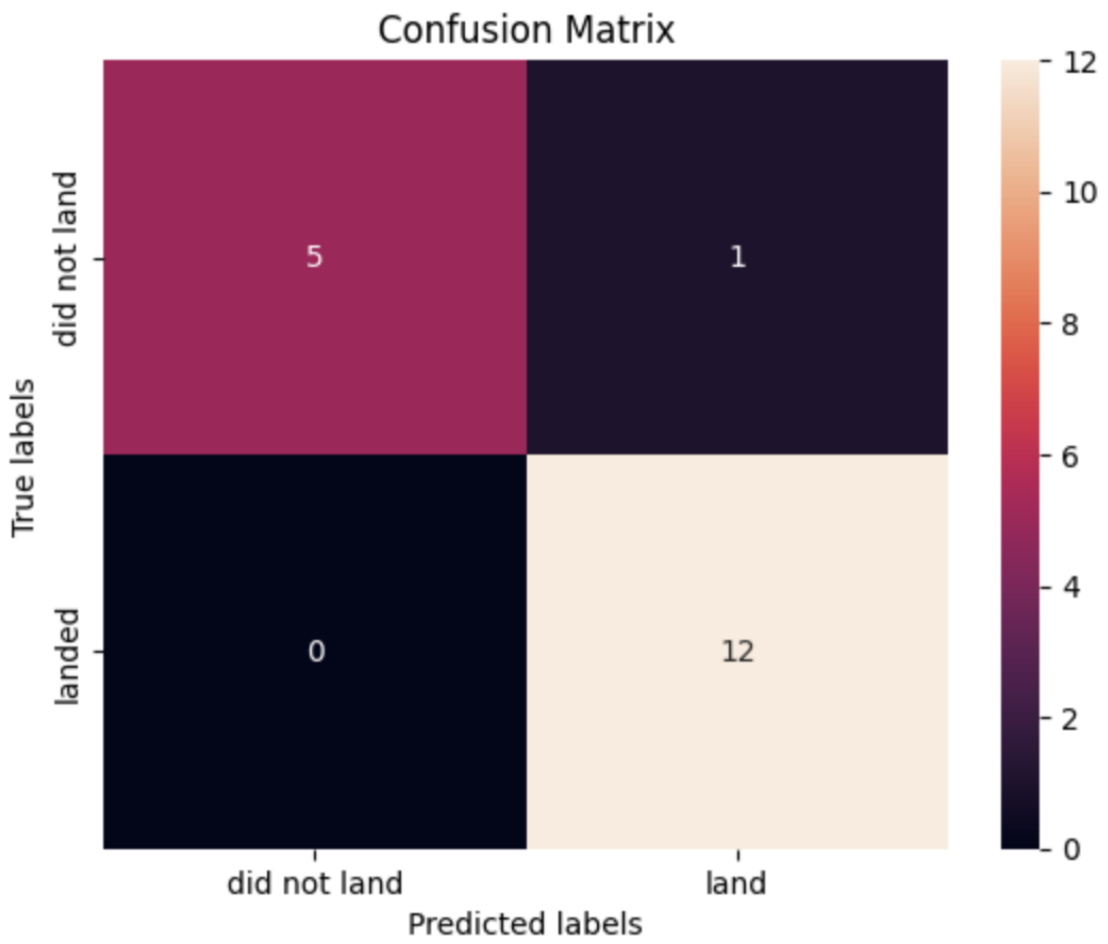
Predictive Analysis (Classification)

Classification Accuracy



The **Decision Tree Classifier** model has the highest classification accuracy: **94.4%**.

Confusion Matrix



The Decision Tree Classifier appears to correctly predict and separate the different classes:

- **True positive ('landed'): 12/12**.
  => the classifier correctly predicted 12 of the 12 successful landings.
- **True negative ('did not land'): 5/6**.
  => the classifier correctly predicted 5 of the 6 failed landings.

The classification report indicates a weighted average (F-1 score) of 0.94, which confirms that the classifier is **performing well**.

	precision	recall	f1-score	support
0	1.00	0.83	0.91	6
1	0.92	1.00	0.96	12
accuracy			0.94	18
macro avg	0.96	0.92	0.93	18
weighted avg	0.95	0.94	0.94	18

Conclusion

Overall, the success rate has increased since 2013, probably as a result of the gained experience and technological improvements.

- **Sites:**
  - KSC LC-39A has the highest launch success rate of all sites (76.9%).
  - CCAFS SLC 40 has the highest number of launched flights but the lowest success rate, although it is improving.
  - CCAFS SLC 40 and KSC LC 39A have a better outcome with heavier payload mass rockets than with lighter ones.
- **Orbits:**
  - Newer orbits (SSO, HEO, MEO, VLEO and GEO) have a better overall success rate.
  - GTO and SO have the lowest success rate.
  - LEO, ISS and PO have a better outcome with heavier payload mass rockets than with lighter ones.
- **Boosters:**
  - FT is the most used and the most successful booster across all payload ranges (65.2% of all successful launches).
  - v1.0 is the least successful booster (0%) and v1.1 has the highest number of failed launches (45.2% of all failed launches).
- **Payload:**
  - [2490-5300 kg] range has the highest launch success rate (60%).
  - [5384-9600kg] range has the lowest success rate (**20%**).
- **Model:** Decision Tree Classifier has the highest classification accuracy with the given dataset (94.4%).

Appendix

- Relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that were used and created during this project can be found on the following GitHub repository:

[GitHub Capstone Project]

Thank you!