IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Sanil Saju
14 June 2023

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection via API, Web Scraping
  - Exploratory Data Analysis(EDA) with Data Visualization
  - EDA with SQL
  - Interactive Map with Folium
  - Dashboards with Plotly Dash
  - Predictive Analysis
- Summary of all results
  - Exploratory Data Analysis results
  - Interactive maps and dashboard
  - Predictive results

Introduction

- Project background and context

The objective of this project is to forecast the successful landing of the Falcon9 first stage, which according to SpaceX's website costs $62 million for launch, whereas other providers' costs exceed $165 million each. This discrepancy in prices is justified by the fact that SpaceX can recycle the initial stage. Evaluating the landing of the stage will enable us to compute the cost of the launch, a crucial piece of information for a firm seeking to rival SpaceX in rocket launches.

Problems you want to find answers

- What are the primary attributes that determine whether a landing is successful or unsuccessful?
- How do the various relationships between the rocket variables impact the outcome of a landing, whether it succeeds or fails?
- What are the prerequisites that SpaceX needs to meet to attain the highest possible success rate for its landings?

Section 1

Methodology

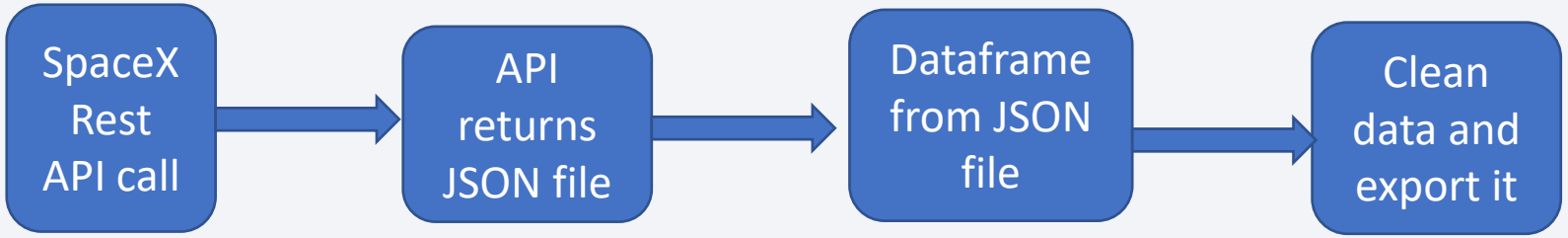
Methodology

Executive Summary

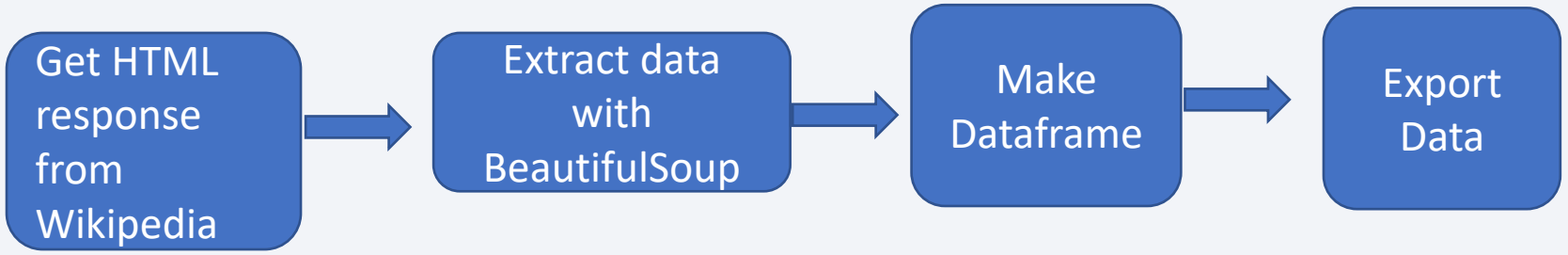
- Data collection methodology:
  - SpaceX Rest API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Dropping unnecessary columns
  - One hot encoding for classification models. (Dummy variables)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Datasets are collected using Rest SpaceX API and web scraping from Wikipedia.
  - The API provides data on rockets, launches, payload, and cores.
    - The SpaceX REST API URL is https://api.spacexdata.com/v4/



  - Scrapping Wikipedia yields information on launches, landings, and payload.
    - URL - https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

1. Getting response from API

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

2. Convert Response to JSON File

```
response = requests.get(static_json_url).json()
data=pd.json_normalize(response)
```

3. Transform Data

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

the list has now been update

```
BoosterVersion[0:5]
```

```
['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

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

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

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

4. Dictionary with data

```
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}
```

[Python code link]

5. Create dataframe

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

6. Filter dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9=data[data['BoosterVersion']!='Falcon 1']
data_falcon9.head()
```

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

7. Export to

Data Collection - Scraping

1. Getting response from HTML

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

2. Create BeautifulSoup object

```
# Use BeautifulSoup() to create a BeautifulSoup object fr
soup = BeautifulSoup(data, "html.parser")
```

3. Find all tables

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

4. Get Columns names

```
column_names = []
first_launch_table = soup.find_all('th')
for x in range(len(first_launch_table)):
    try:
        name=extract_column_from_header(first_launch_table[x])
        if(name is not None and len(name) > 0):
            column_names.append(name)
    except:
        pass
```

5. Create dictionary

```
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'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

6. Add data to keys

```
extracted_row = 0
#Extract each table
for table_number,table in enumerate(soup.find_all('tab
   # get table row
    for rows in table.find_all("tr"):
        #check to see if first table heading is as num
        if rows.th:
            if rows.th.string:
                flight_number=rows.th.string.strip()
                flag=flight_number.isdigit()
```

7. Create dataframe

```
df = pd.DataFrame.from_dict(launch_dict)
df.head()
```

8. Export to file

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

Python code link

Data Wrangling

- The dataset includes multiple instances where the booster failed to land properly.
  - The terms True Ocean, True RTLS, and True ASDS indicate that the mission was successful, while False Ocean, False RTLS, and False ASDS indicate that it was a failure.
- It is necessary to convert the string variables into categorical variables.

1. **Calculate the number of launches on each site**

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

```
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

2. **ach orbit**

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

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

3. **Calculate the number and occurence of mission outcome per orbit type**

```
# landing_outcomes = values on Outcome column
landing_outcomes=df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS       41
None None       19
True RTLS       14
False ASDS       6
True Ocean       5
False Ocean      2
None ASDS        2
False RTLS       1
Name: Outcome, dtype: int64
```

4. **Landing outcome label**

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class=[]
for x in df['Outcome']:
    if x in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
```

Python Code Link

EDA with Data Visualization

- Scatter plots

Scatter plots shows the relationship between variables

  - Flight Number vs Launch Site
  - Payload vs Launch Site
  - Flight Number vs Orbit Type
  - Payload vs Orbit Type

- Bar graph

Bar graph shows the relationship between numerical and categorical variables

  - Success rate vs Orbit

- Line Graph

Line graph shows data variables and their trends.

Python Code Link

  - Success rate vs Year

EDA with SQL

- We performed SQL queries to gather and understand data from dataset
  - Display the names of the unique launch sites in the space mission
  - Display 5 records where launch sites begin with the string 'CCA'
  - Display the total payload mass carried by boosters launched by NASA (CRS)
  - Display average payload mass carried by booster version F9 v1.1
  - List the date when the first successful landing outcome in ground pad was achieved.
  - List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
  - List the total number of successful and failure mission outcomes
  - List the names of the booster_versions which have carried the maximum payload mass. Use a subquery
  - List the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
  - Rank the count of successful landing_outcomes between the date 04-06-2010 and 20-03-2017 in descending order.

Python Code Link

Build an Interactive Map with Folium

- By utilizing the Folium map, we can generate a map that centers on the NASA Johnson Space Center located in Houston, Texas
  - Red circle at its coordinates that displays its name as a label.
  - The launch sites can be indicated on the map with red circles at their respective coordinates, and their names can be displayed as labels.
  - To present different information for the same coordinates, the points can be grouped into clusters.
  - Markers can be used to denote successful and unsuccessful landings, with green markers indicating successful landings and red markers indicating unsuccessful ones.
  - Markers can be used to demonstrate the distance between the launch site and crucial locations like the railway, highway, coast way, and city, and a line can be drawn between them.
- Creating these objects allows for a clearer comprehension of the data by displaying all launch sites, their surroundings, and the number of successful and unsuccessful landings in a more easily digestible format.

Python Code Link

Build a Dashboard with Plotly Dash

- A dashboard can be produced using Plotly Dash, which comprises various components such as a dropdown, pie chart, range slider, and scatter plot.
  - The dropdown component permits users to pick a launch site or all launch sites, and it can be created using dash_core_components.Dropdown
  - The pie chart illustrates the total number of successes and failures for the launch site selected using the dropdown component, and it can be generated using plotly.express.pie
  - The range slider component allows users to choose a payload mass within a predetermined range, and it can be created using dash_core_components.RangeSlider
  - The scatter plot component displays the correlation between two variables and can be created using plotly.express.scatter

  Python Code Link

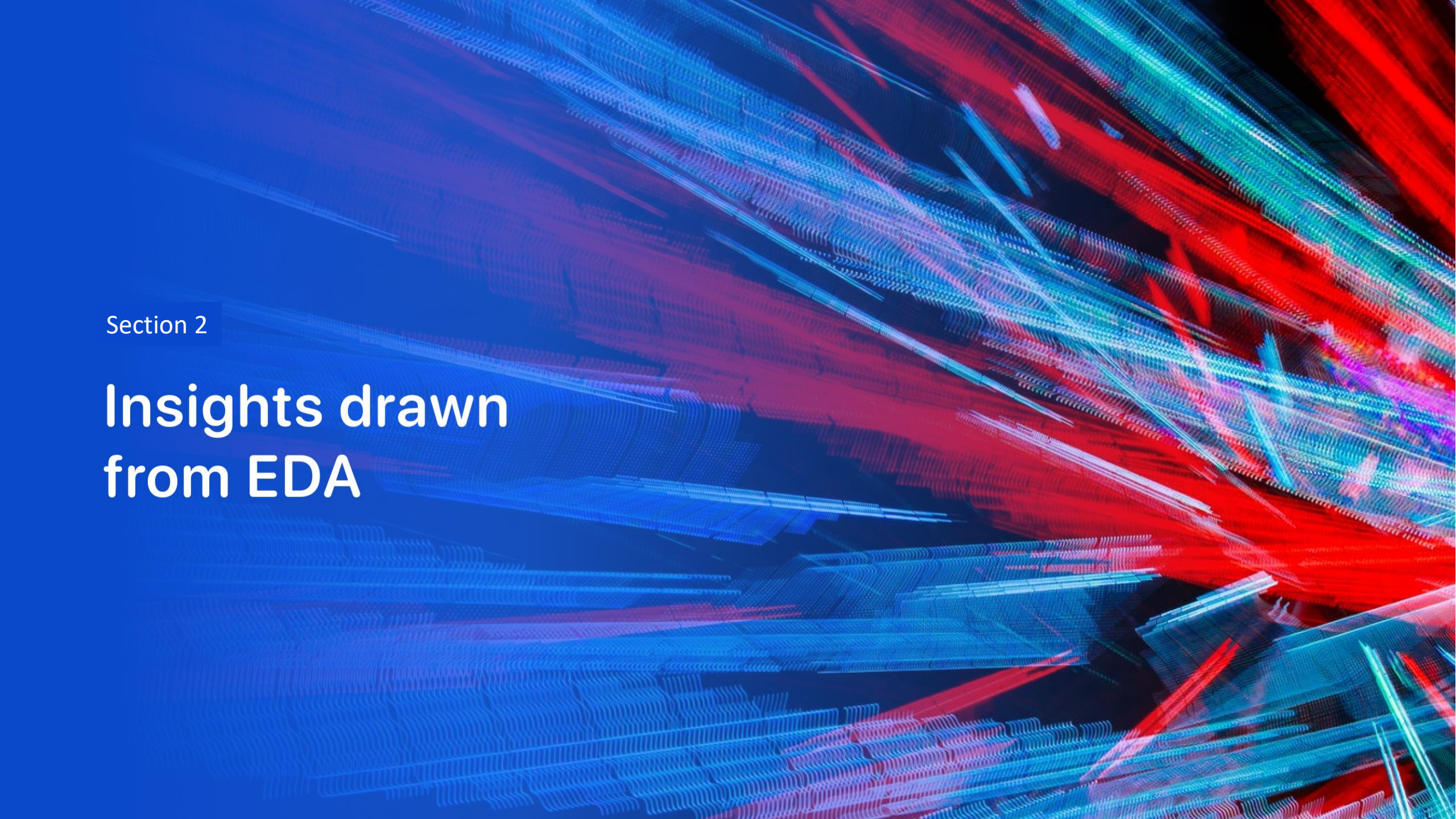
Predictive Analysis (Classification)

- Data Preparation
  - Load the Dataset
  - Normalize the Data
  - Divide the data into training and testing sets
- Model Preparation
  - Choose different machine learning algorithms
  - Define the parameters for each algorithm to GridSearchCV
  - Train the GridSearchModel with training set
- Model Evaluation
  - Obtain the best hyperparameter for each model type
  - Determine the accuracy of each model using the dataset
  - Create Confusion Matrix plot
- Model Comparison
  - Compare the accuracy of different models.
  - Select the model with the highest accuracy.

Python Code Link

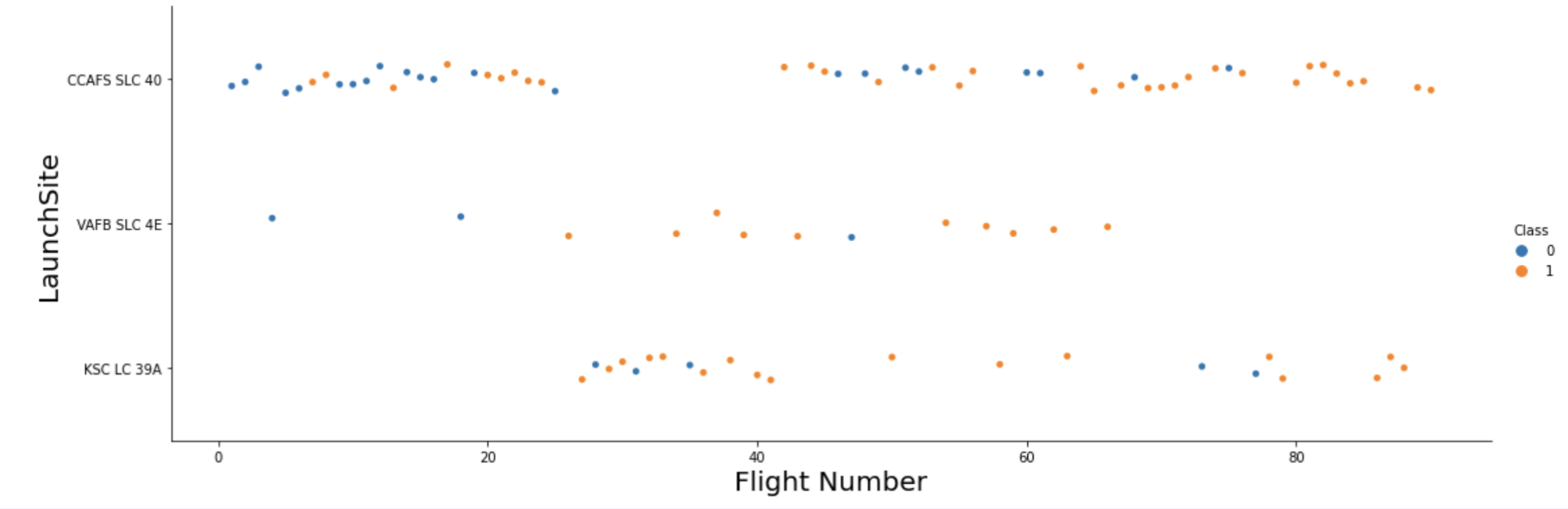
Results

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


Section 2
Insights drawn from EDA

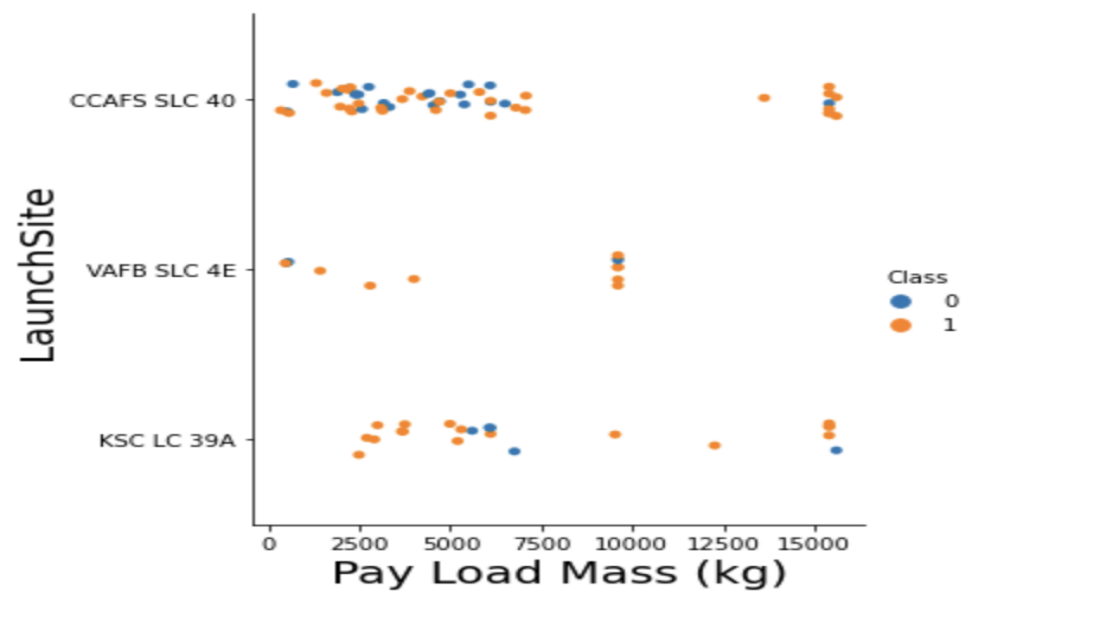
Flight Number vs. Launch Site

- Success rate is increasing for each site



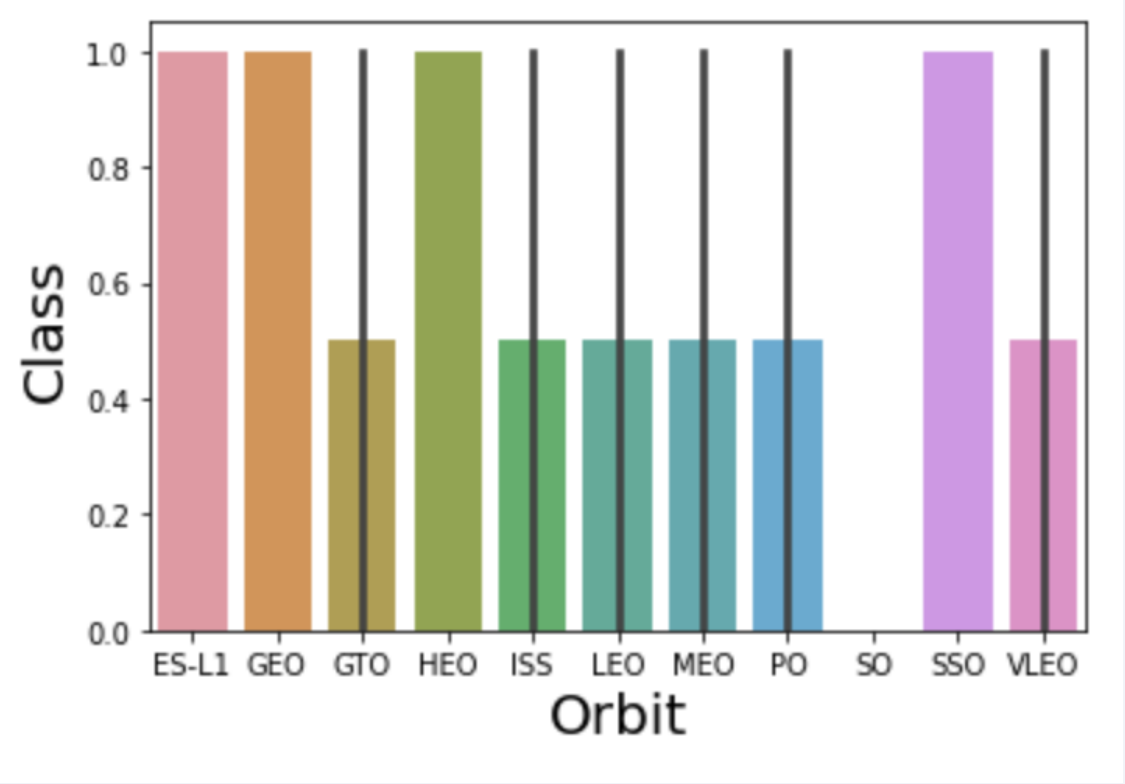
Payload vs. Launch Site

- As we can infer from the chart, a heavier payload may be considered for a successful landing depending on each launching site. Too heavy payload can lead to landing failure also.



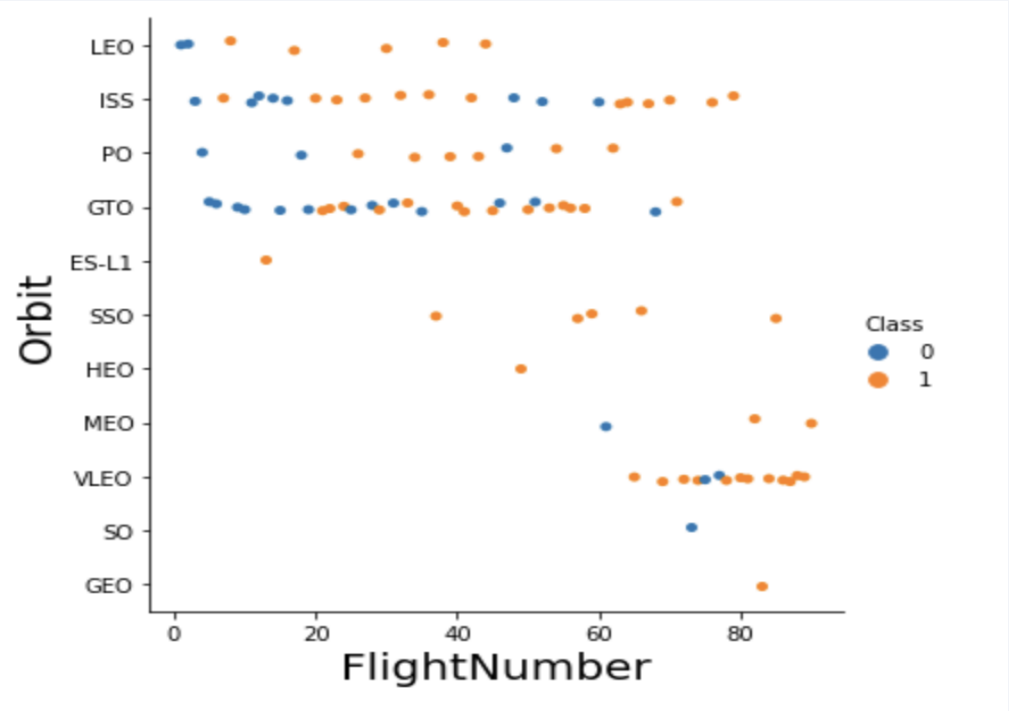
Success Rate vs. Orbit Type

- Below bar chart will show the success rate of different orbit types. ES-L1,GEO,HEO,SSO orbit types have the best success rate.



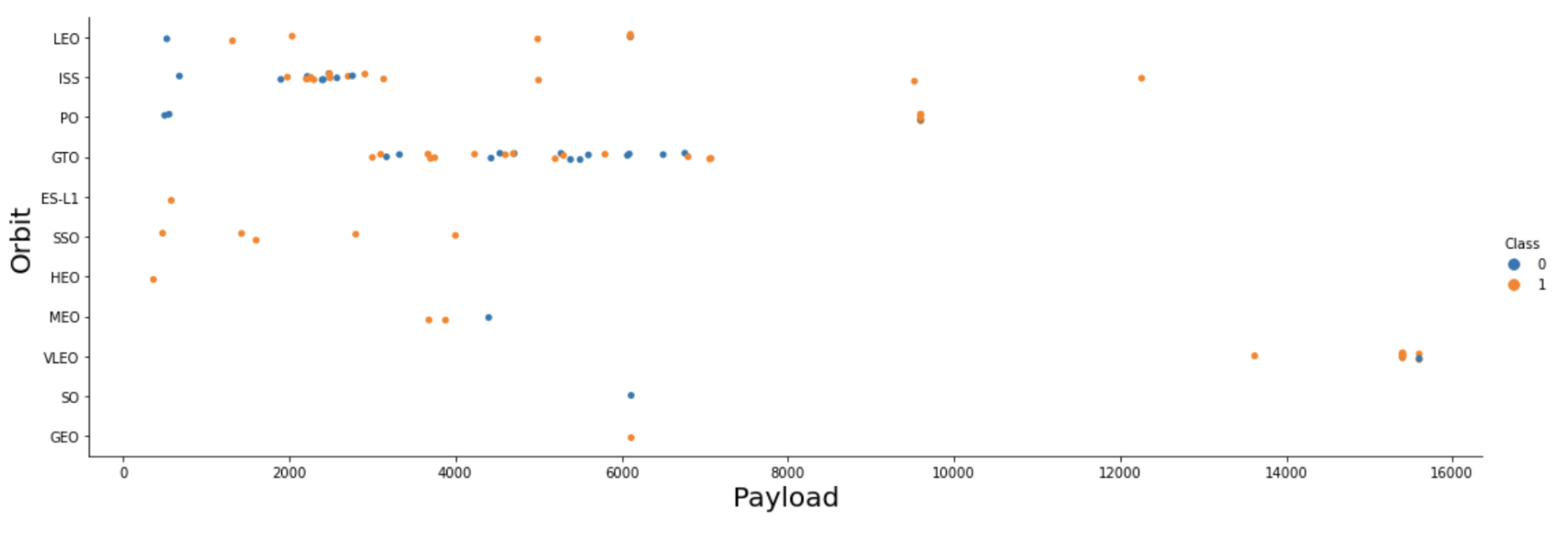
Flight Number vs. Orbit Type

- For 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. We can suppose that the high success rate of some orbits like SSO or HEO is due to the knowledge learned during former launches for other orbits.



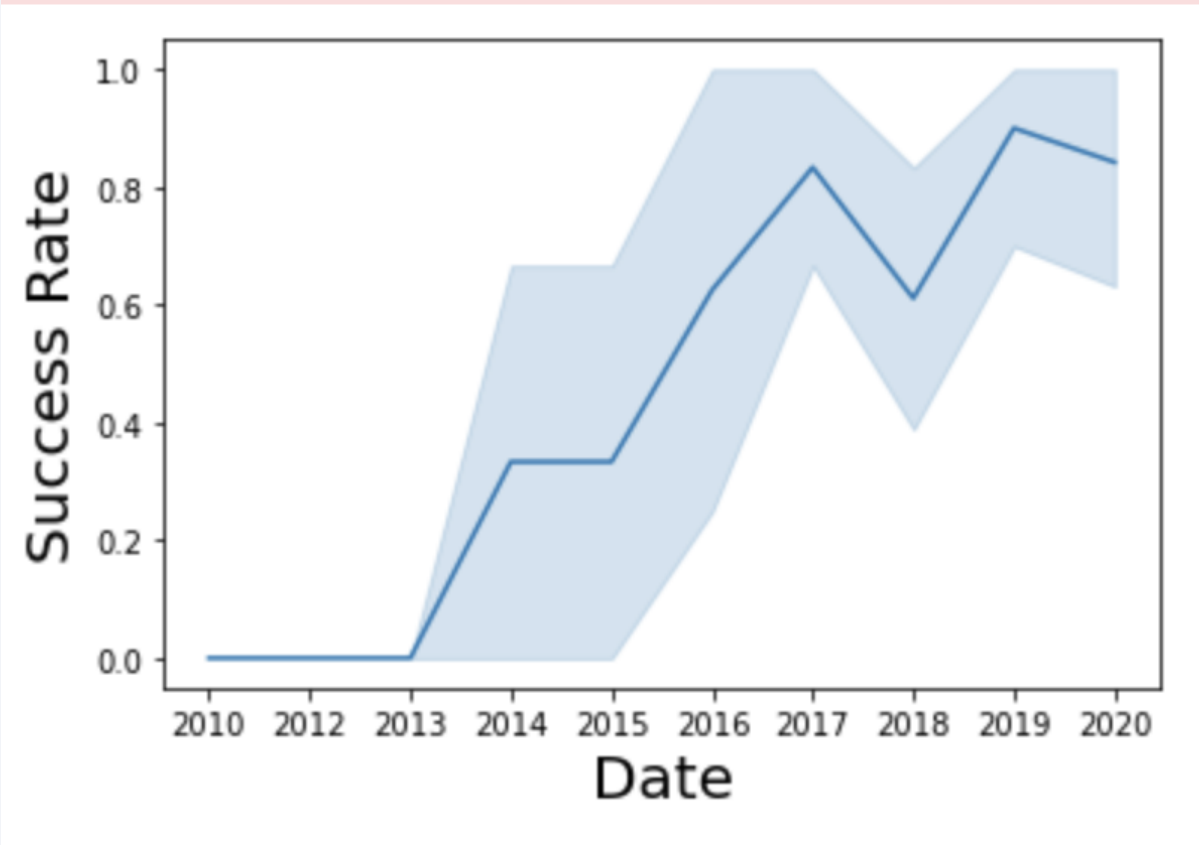
Payload vs. Orbit Type

- The weight of the payloads can have a great influence on the success rate of the launches in certain orbits. Heavier payloads increase the success rate of LEO orbit. On the other hand, by reducing the payload of GTO orbit results in better success rate.



Launch Success Yearly Trend

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



All Launch Site Names

SQL query

```
%sql select distinct(LAUNCH_SITE) from SPACEXTBL
```

Result

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

The use of DISTINCT in the query allows to remove duplicate LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

SQL Query

```
%sql select * from SPACEXTBL where LAUNCH_SITE like 'CCA%' limit 5
```

Result

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

By using the WHERE clause with the LIKE clause, we can filter launch sites that have the substring CCA.
The LIMIT 5 clause restricts the output to only five records that meet the filtering criteria.

Total Payload Mass

SQL query

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'
```

Results

1
45596

The output of this query is the total of all payload masses in cases where the customer is identified as NASA (CRS).

Average Payload Mass by F9 v1.1

SQL query

```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where BOOSTER_VERSION = 'F9 v1.1'
```

Result

1
2928.400000

The output of this query is the mean of all payload masses in cases where the booster version includes the substring F9 v1.1.

First Successful Ground Landing Date

SQL query

```
%sql select min(DATE) from SPACEXTBL where Landing__Outcome = 'Success (ground pad)'
```

Result

1
2015-12-22

This query is used to retrieve the record of the earliest successful landing. The WHERE clause filters the dataset to include only records where the landing was successful, and then the MIN function is used to select the record with the oldest date.

Successful Drone Ship Landing with Payload between 4000 and 6000

SQL query

```
%sql select BOOSTER_VERSION from SPACEXTBL where Landing__Outcome = 'Success (drone ship)' and PAYLOAD_MASS__KG_ > 4000 and PAYLOAD_MASS_
```

Result

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

The output of this query is the booster version for cases where the landing was successful and the payload mass falls between the range of 4000 and 6000 kg. The WHERE and AND clauses are used to filter the dataset based on these criteria.

Total Number of Successful and Failure Mission Outcomes

SQL query

```
%sql select count(MISSION_OUTCOME) from SPACEXTBL where MISSION_OUTCOME = 'Success' or MISSION_OUTCOME = 'Failure (in flight)'
```

Result

1
100

The COUNT function counts records filtered by using the WHERE clause.

Boosters Carried Maximum Payload

SQL query

```
%sql select BOOSTER_VERSION from SPACEXTBL where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)
```

Result

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

We used the subquery to find the maximum of payload and then using the WHERE clause to filter out the results

2015 Launch Records

- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

- Present your query result with a short explanation here

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

SQL query

```
%sql select * from SPACEXTBL where Landing__Outcome like 'Success%' and (DATE between '2010-06-04' and '2017-03-20') order by date desc
```

Result

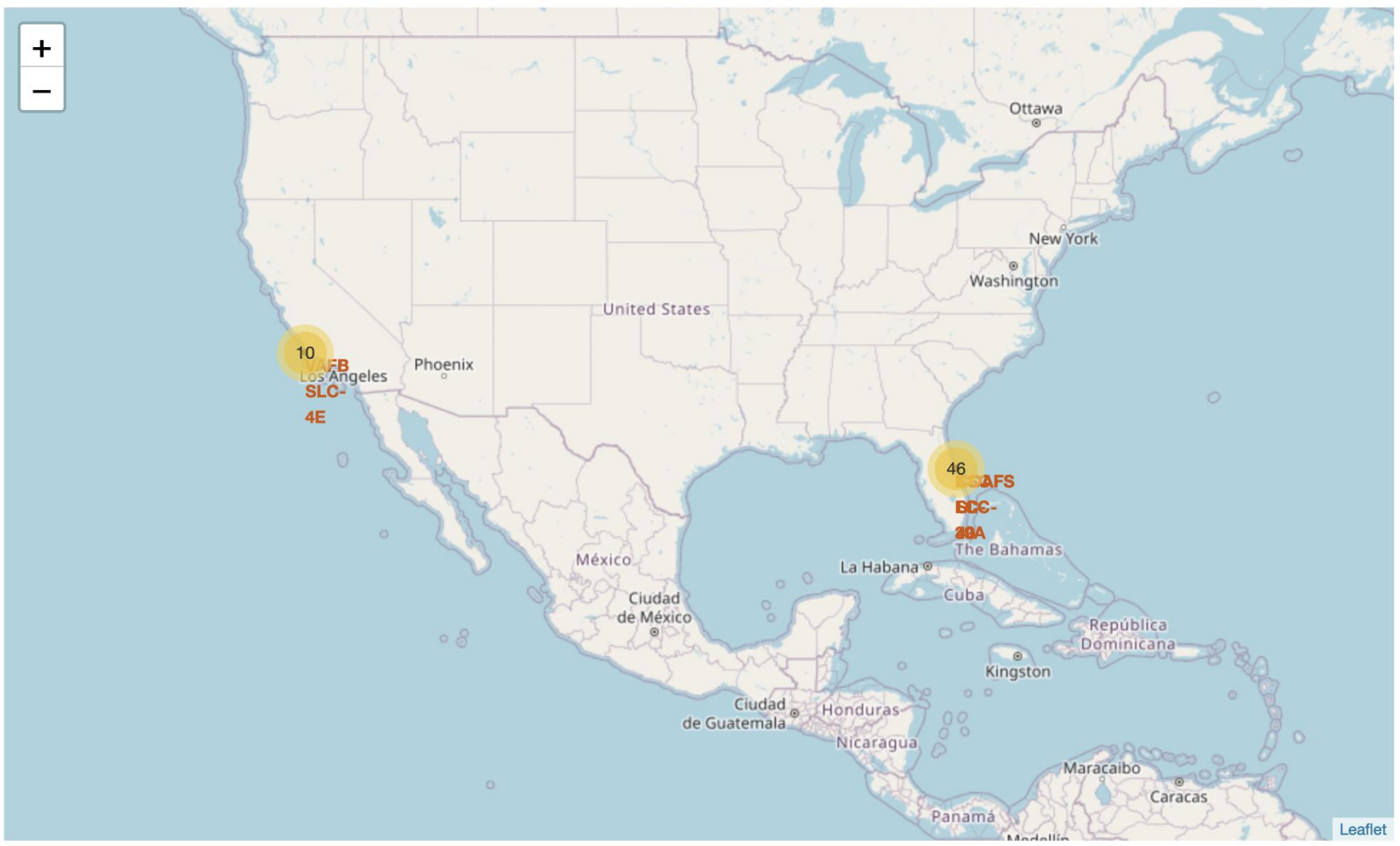
DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-01-14	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600	Polar LEO	Iridium Communications	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-07-18	04:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2016-05-27	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100	GTO	Thaicom	Success	Success (drone ship)
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-04-08	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136	LEO (ISS)	NASA (CRS)	Success	Success (drone ship)
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

This query returns the landing outcomes between 2010-06-04 and 2017-03-20 in the descending order

Section 3

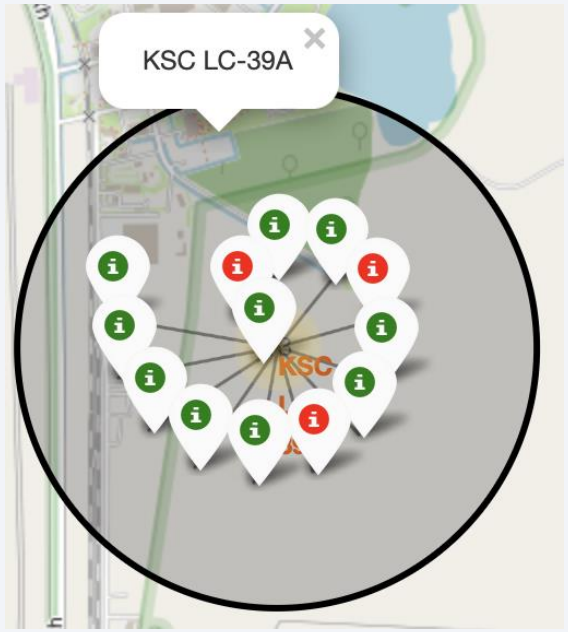
Launch Sites Proximities Analysis

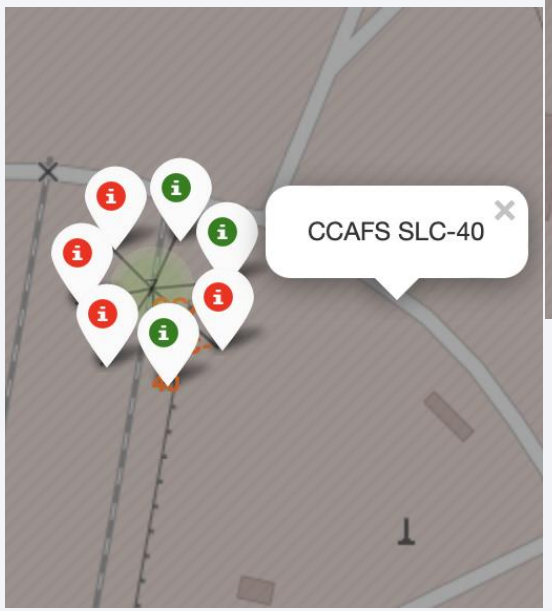
Folium Map – Launch Sites

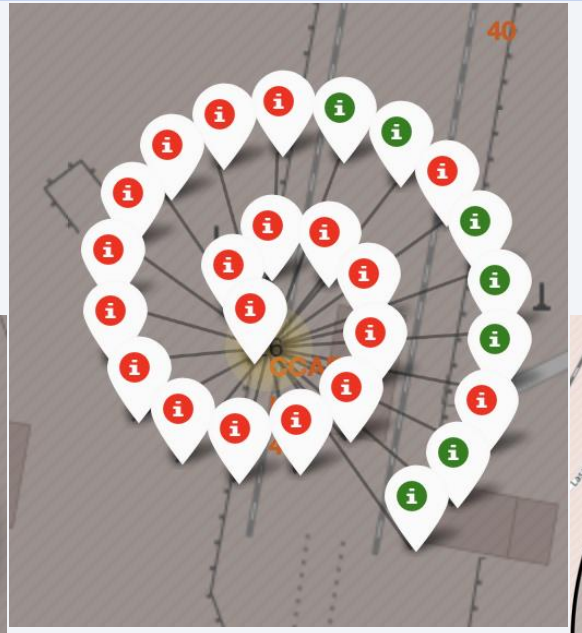


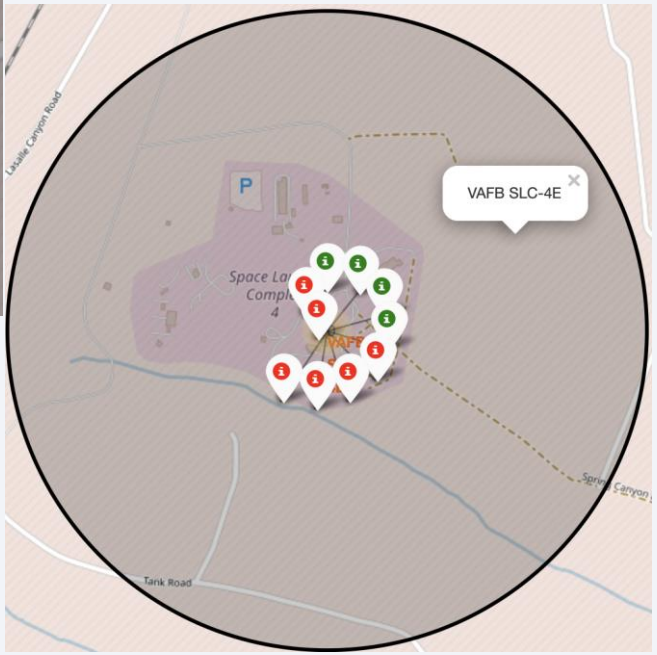
We can see that, all the launch sites are situated on the coast of the United States

Folium Map - Colour Labeled Markers



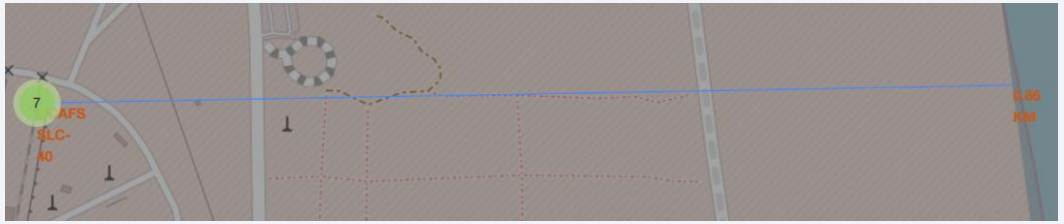


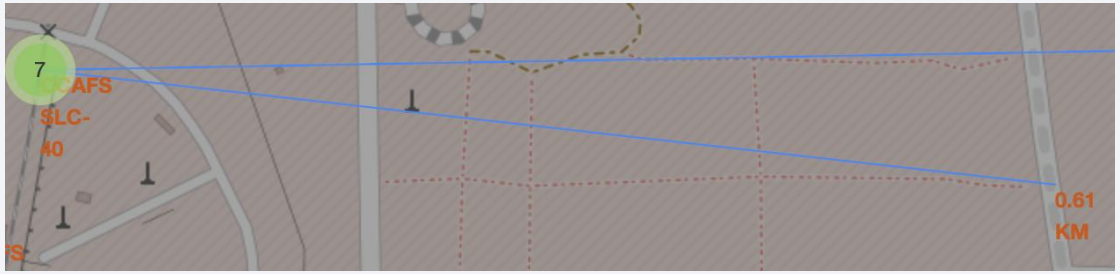


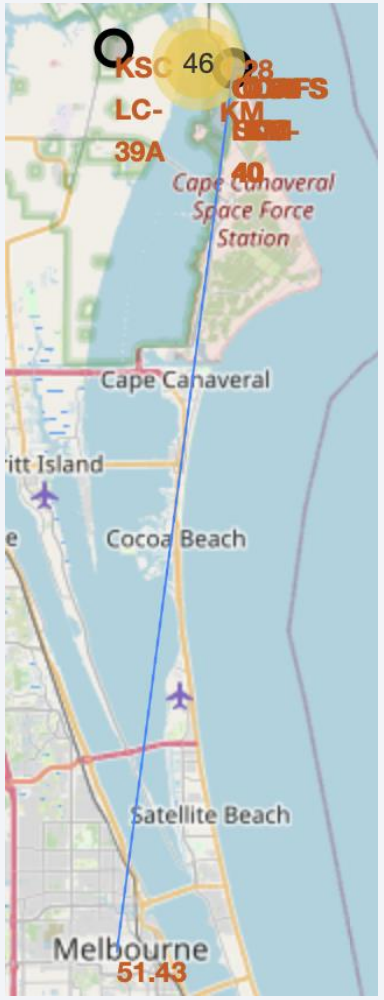


The color of the markers on the map indicates the outcome of the launch: green represents successful launches, while red represents unsuccessful launches. It can be observed that KSC LC-39A has a higher success rate for launches.

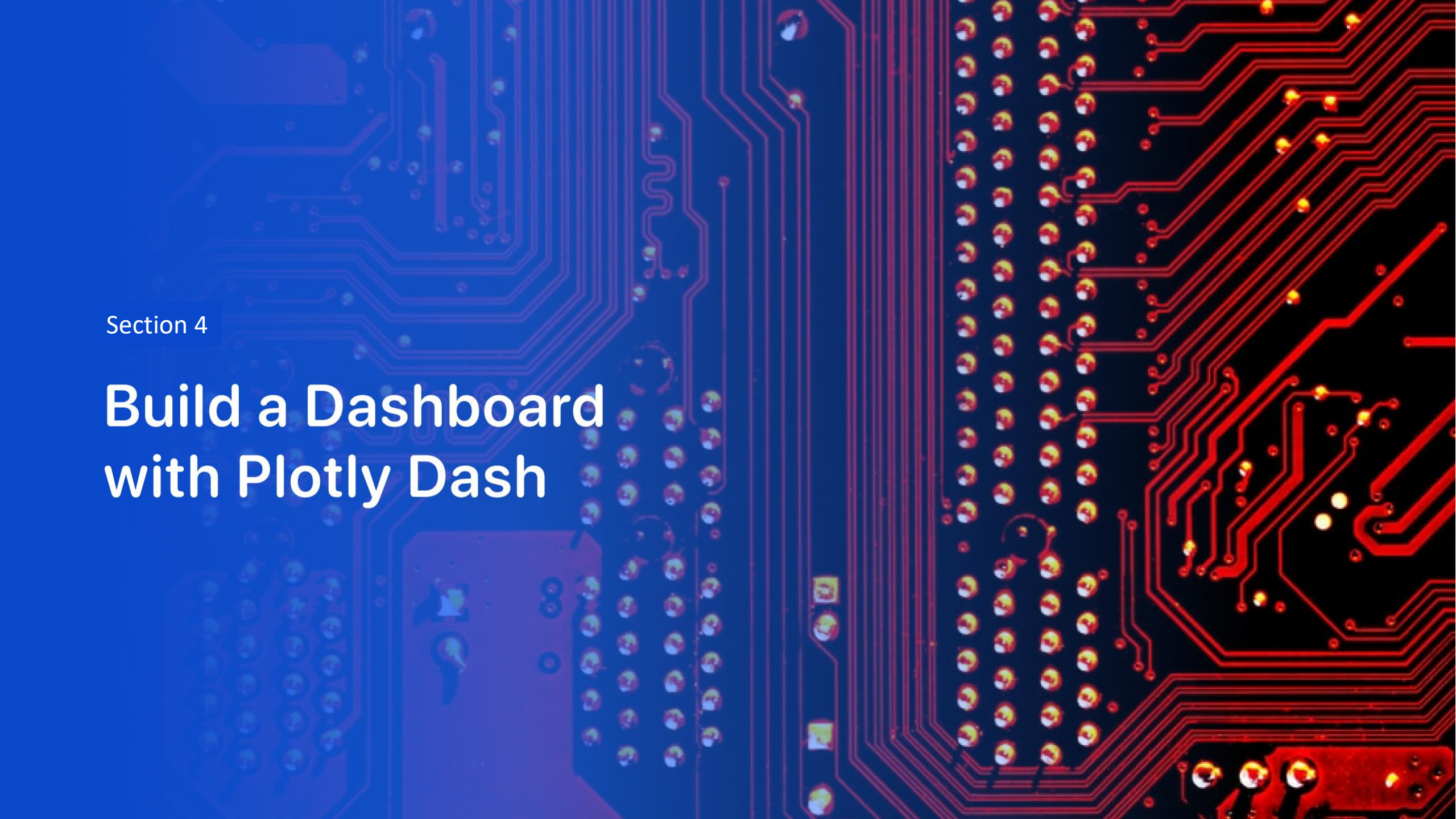
Folium Map – Distances between CCAFS SLC-40 and its proximities



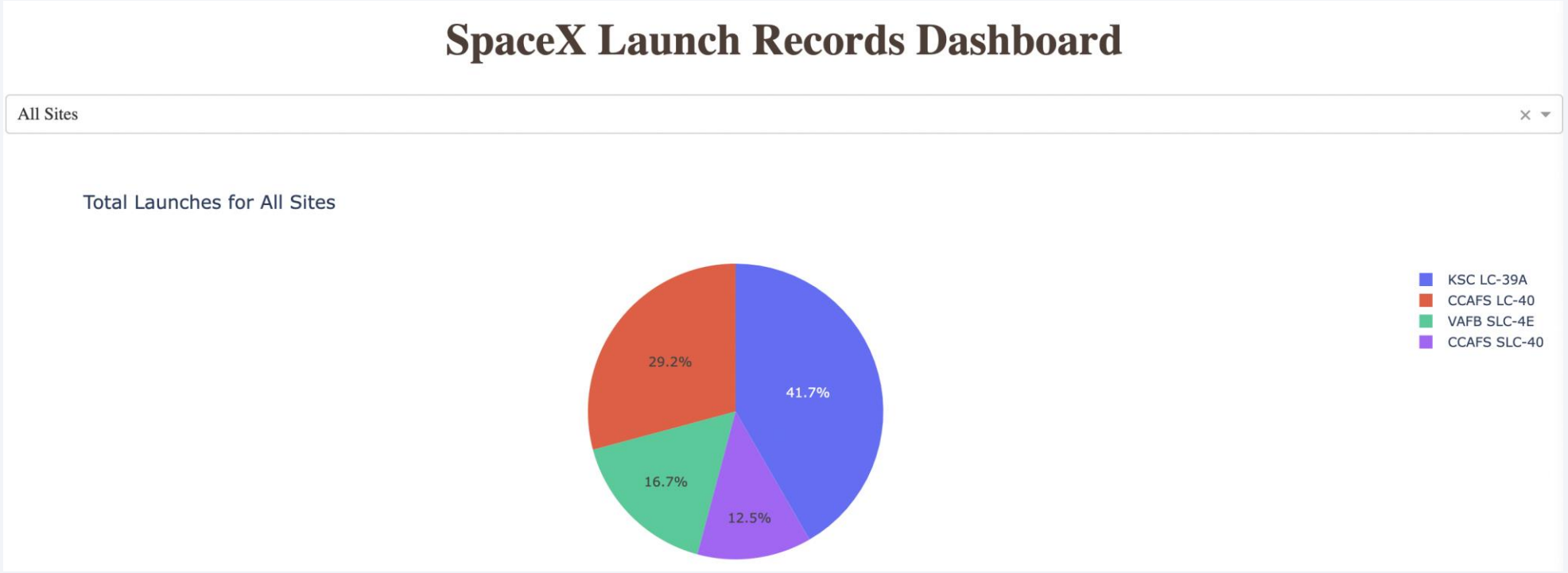




Is CCAFS SLC-40 in close proximity to railways ? Yes
Is CCAFS SLC-40 in close proximity to highways ? Yes
Is CCAFS SLC-40 in close proximity to coastline ? Yes
Do CCAFS SLC-40 keeps certain distance away from cities ? Yes cities are far away from the sites

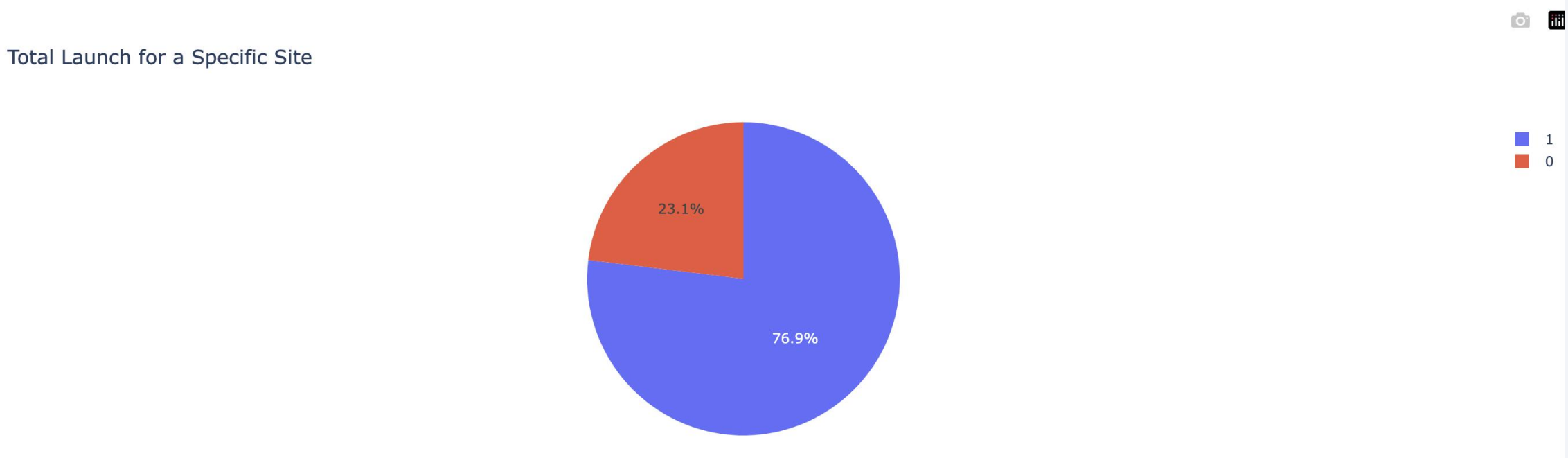

Section 4
Build a Dashboard with Plotly Dash

Dashboard – Launch Success Count



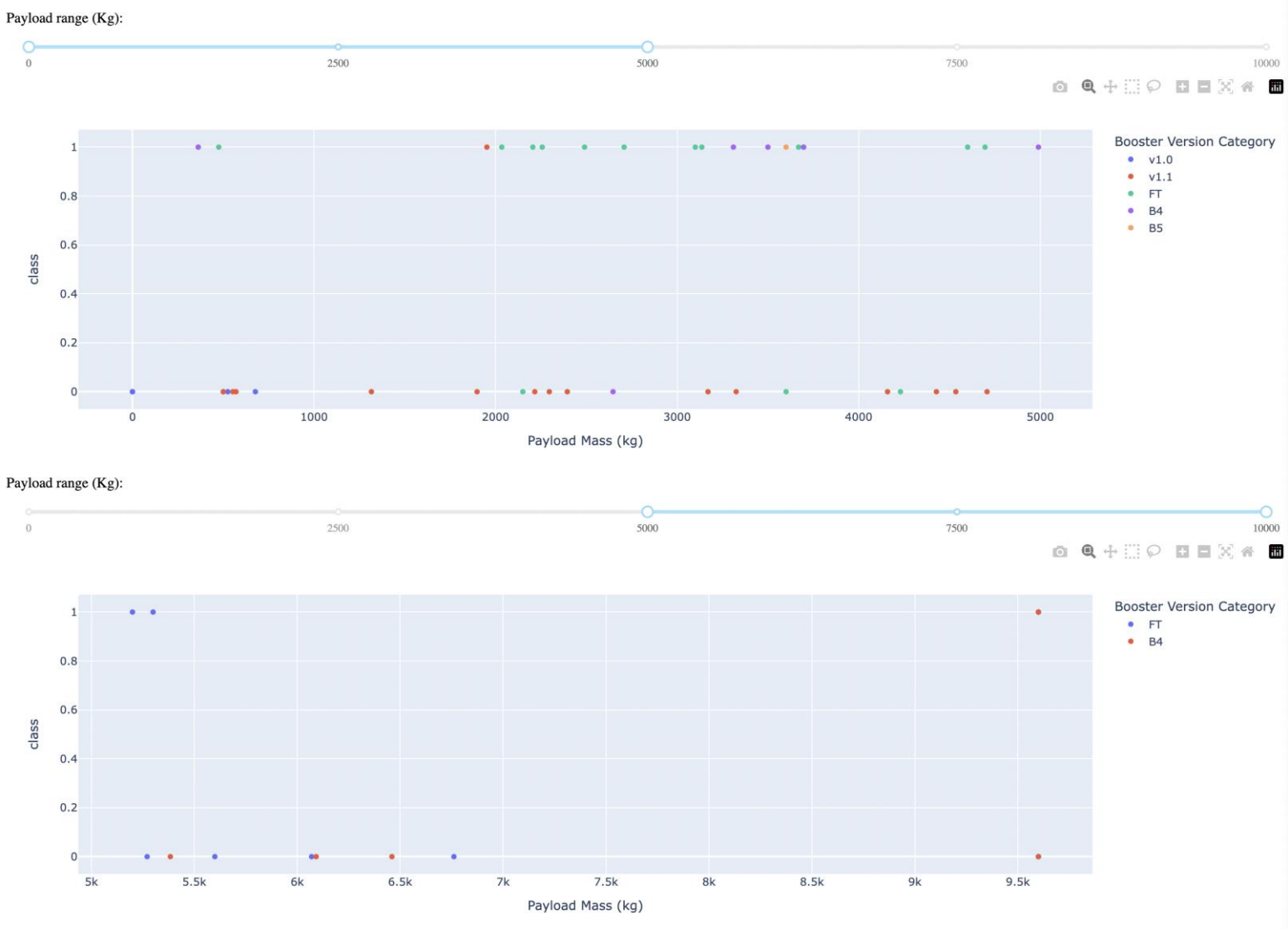
We can infer from the pie chart that KSC LC-39A has best success rate of launches

Dashboard – Total success launches for Site KSC LC-39A



We see that KSC LC-39A has achieved a 76.9% success rate while getting a 23.1% failure rate.

Dashboard – Payload mass vs Outcome for all sites with different payload mass selected



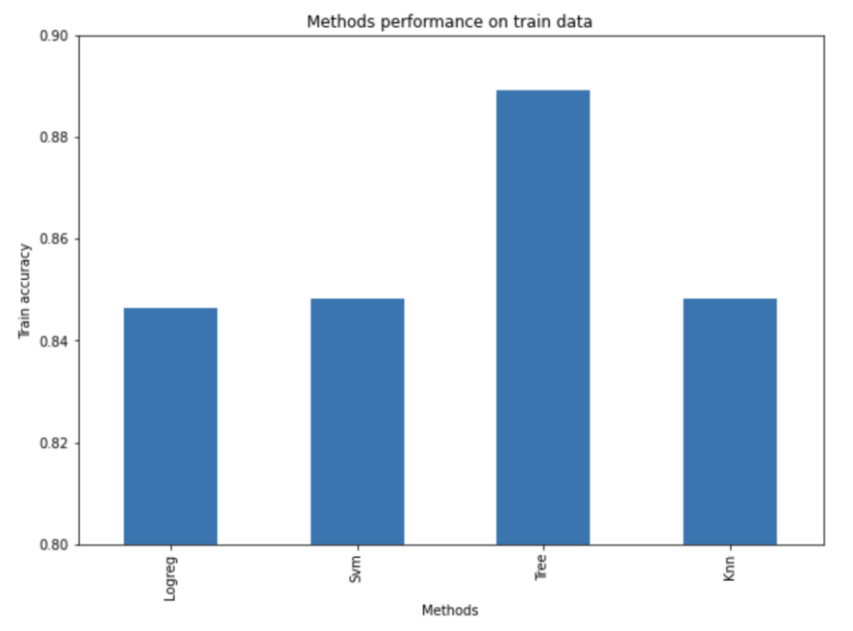
We can infer from the images that, Low weighted payloads have a better success rate than the heavy weighted payloads.

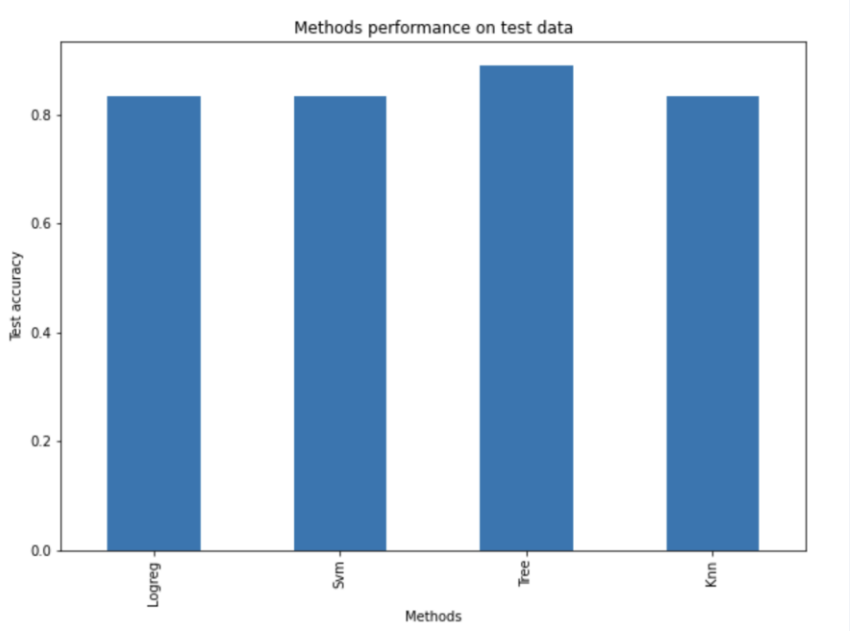
Section 5

Predictive Analysis (Classification)

Classification Accuracy

	Accuracy Train	Accuracy Test
Logreg	0.846429	0.833333
Svm	0.848214	0.833333
Tree	0.889286	0.888889
Knn	0.848214	0.833333

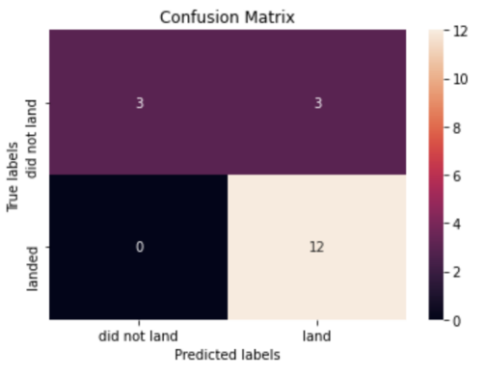




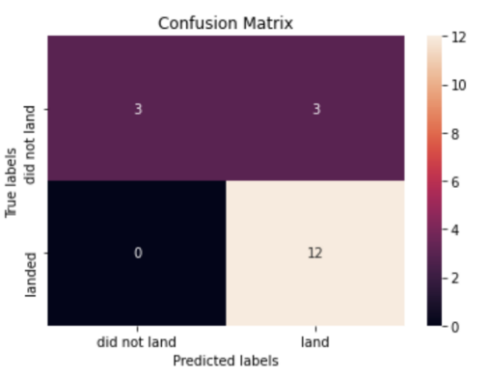
During the accuracy test, all methods produced comparable results. To make a definitive choice between them, we could obtain additional test data. However, if we had to select one method immediately, we would choose the decision tree.

Confusion Matrix

1. Logistic Regression

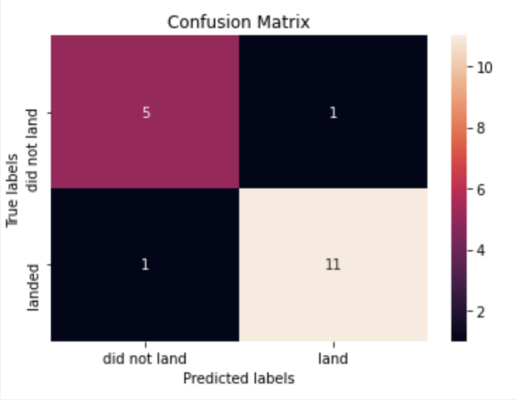


2. SVM

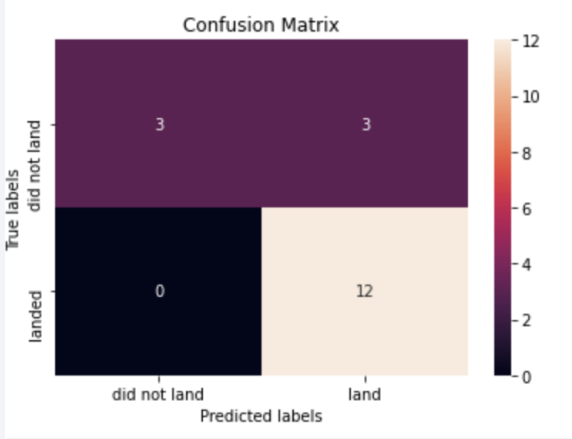


Since the test accuracies of all models are the same, the confusion matrices are also the same. However, false positives are the primary issue with these models.

3. Tree



4. KNN



Conclusions

- There are various factors that contribute to the success of a space mission, including the launch site, orbit, and the number of previous launches. Progress in knowledge gained from previous launches may have enabled the transition from launch failures to successful launches.
- The most successful orbits are GEO, HEO, SSO, and ES-L1.
- Payload mass is a crucial factor in the success of a mission, depending on the orbit. Some orbits require a light or heavy payload mass, but generally, lighter payloads perform better than heavier ones.
- Currently, it is unclear why some launch sites are more effective than others (KSCLC-39A being the most effective). Obtaining atmospheric or other relevant data could help answer this question.
- Although the test accuracy of various models used in the dataset is the same, the Decision Tree Algorithm is chosen as the best model because it has a higher train accuracy

Thank you!