Winning Space Race with Data Science

Nguyen Thi Ngoc Mai
September 2022

Outline

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

Executive Summary

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

Introduction

- **Project background and context**

  ✓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.

- **Problems you want to find answers**

  ✓ What are the main characteristics of a successful or failed landing?

  ✓ What are the effects of each relationship of the rocket variables on a successful or failed landing?

  ✓ What are the conditions which will allow SpaceX to achieve the best successful landing rate ?

Section 1

Methodology

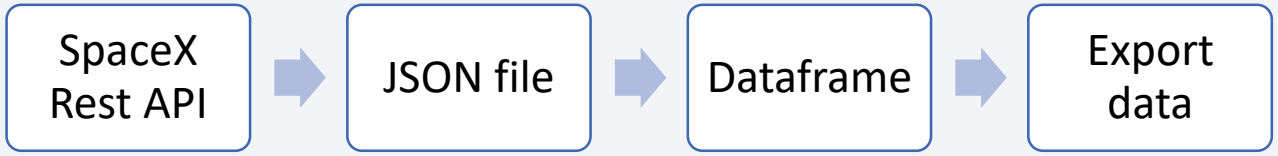
Methodology

Executive Summary

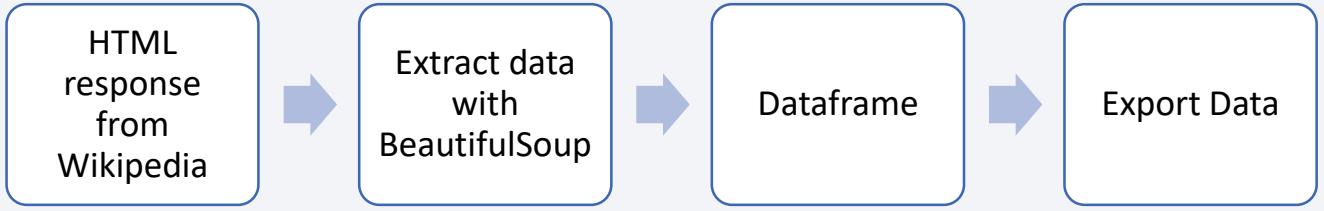
- Data collection methodology:
  - SpaceX REST API
  - Web Scrapping from Wikipedia
- Perform data wrangling
  - Delete unnecessary columns
  - One Hot Encoding for classification models
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- Datasets are collected from Rest SpaceX API and webscrapping Wikipedia
- The information obtained by the API are rocket, launches, payload information.
- The Space X REST API URL is api.spacexdata.com/v4/



- The information obtained by the webscrapping of Wikipedia are launches, landing, payload information.
- URL https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922



Data Collection – SpaceX API

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

1

```
In [6]: spacex_url="https://api.spacexdata.com/v4/launches/past"

In [7]: response = requests.get(spacex_url)
```

2

```
In [12]: # Use json_normalize meethod to convert the json result into a dataframe
         data = response.json()
         data = pd.json_normalize(data)
```

3

```
In [18]: BoosterVersion[0:5]
Out[18]: ['Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 1', 'Falcon 9']
```

we can apply the rest of the functions here:

```
In [19]: # Call getLaunchSite
         getLaunchSite(data)

In [20]: # Call getPayloadData
         getPayloadData(data)

In [21]: # Call getCoreData
         getCoreData(data)
```

4

```
In [22]: 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}
```

5

```
In [23]: # Create a data from launch_dict
         data = pd.DataFrame({key:pd.Series(value) for key, value in launch_dict.items()})
```

6

```
In [25]: # Hint data['BoosterVersion']!='Falcon 1'
         data_falcon9 = data[data['BoosterVersion']!='Falcon 1']
```

7

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

8

Data Collection - Scraping

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

1

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

2

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(response.text, "html5lib")
```

3

```
# 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')
```

4

```
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

for th in first_launch_table.find_all('th'):
    name = extract_column_from_header(th)
    if name is not None and len(name) > 0 :
        column_names.append(name)
```

5

```
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

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

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

Data Wrangling

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

In the dataset, there are several cases where the booster did not land successully.

- True Ocean, True RTLS, True ASDS means the mission has been successful.
- False Ocean, False RTLS, False ASDS means the mission was a failure

Transform string variables into categorical variables where 1 means the mission was successful and 0 means the mission was failed.

```
df['LaunchSite'].value_counts()
```

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

```
df['Orbit'].value_counts()
```

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

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

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

```
landing_class = []
for key,value in df["Outcome"].items():
    if value in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

```
df.to_csv("dataset_part_2.csv", index=False)
```

EDA with Data Visualization

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

Scatter Graphs
- Flight Number vs. Payload Mass
- Flight Number vs. Launch Site
- Payload vs. Launch Site
- Orbit vs. Flight Number
- Payload vs. Orbit Type
- Orbit vs. Payload Mass

Bar Graph
- Success rate vs. Orbit

Line Graph
- Success rate vs. Year

EDA with SQL

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

Performed SQL to gather data from dataset:

- Displaying 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 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.
- List the records which will display the month names, failure landing outcomes in drone ship, booster versions, launch site for months in 2015.
- Rank the count of successful landing outcomes between the date 04-06-2010 and 20-03-2017 in descending order

Build an Interactive Map with Folium

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

Folium is a map centered on NASA Johnson Space Center at Houston, Texas

- Red circle at NASA Johnson Space Center's coordinate with label showing its name
- Red circles at each launch site coordinates with label showing launch site name
- The grouping of points in a cluster to display multiple and different information for the same coordinates
- Markers to show successful and unsuccessful landings: Green for successful landing and Red for unsuccessful landing
- Markers to show distance between launch site to key locations (railway, highway, coast way, city) and plot a line between them

These objects are created in order to understand the problem and the data. We can show easily all launch sites, their surroundings and the number of successful and unsuccessful landings.

Build a Dashboard with Plotly Dash

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

Dashboard has dropdown, pie chart, rangeslider and scatter plot components

- Dropdown allows a user to choose the launch site or all launch sites
- Pie chart shows the total success and the total failure for the launch site chosen with the dropdown component
- Rangeslider allows a user to select a payload mass in a fixed range
- Scatter chart shows the relationship between two variables, in particular Success vs Payload Mass

Predictive Analysis (Classification)

GitHub URL: https://github.com/NgocMai20594/Applied-Data-Science-Capstone/

Data preparation

- Load dataset
- Normalize data
- Split data into training and test sets.

Model preparation

- Selection of machine learning algorithms
- Set parameters for each algorithm to GridSearchCV
- Training GridSearchModel models with training dataset

Model evaluation

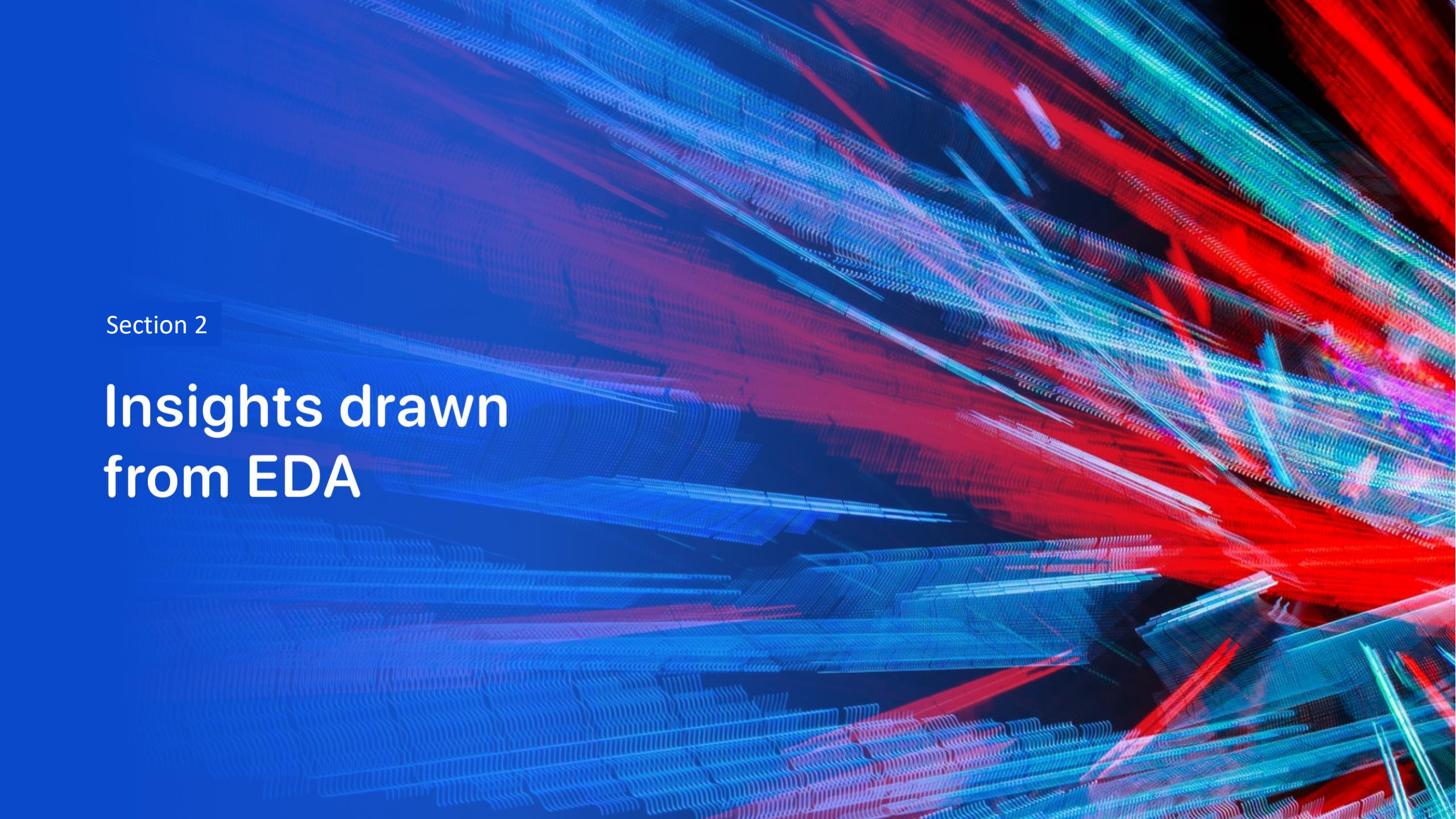
- Get best hyperparameters for each type of model
- Compute accuracy for each model with test dataset
- Plot Confusion Matrix

Model comparison

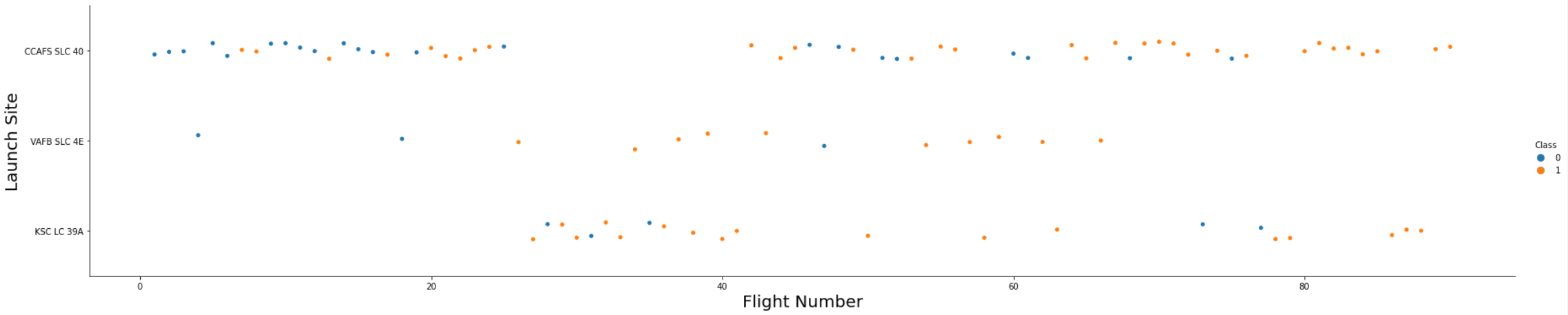
- Comparison of models according to their accuracy
- The model with the best accuracy will be chosen

Results

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

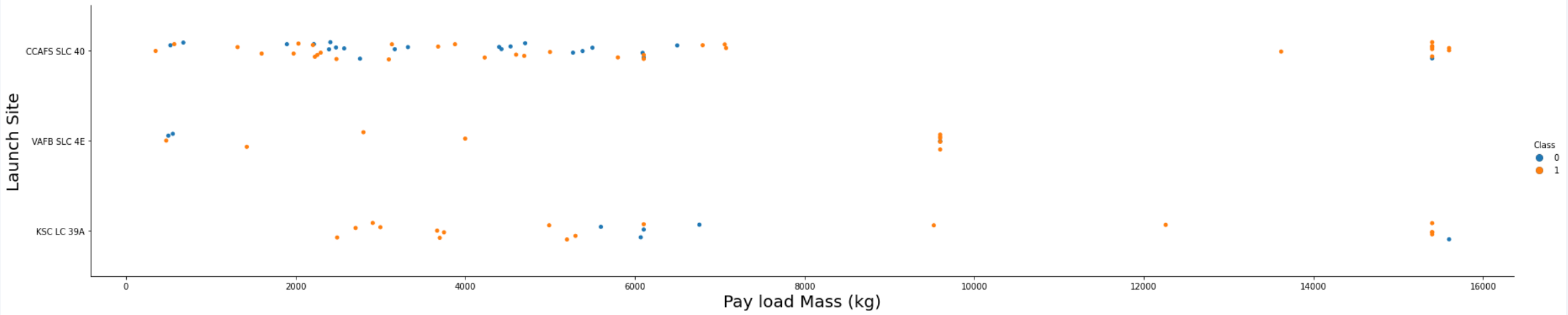

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site



success rate is increasing

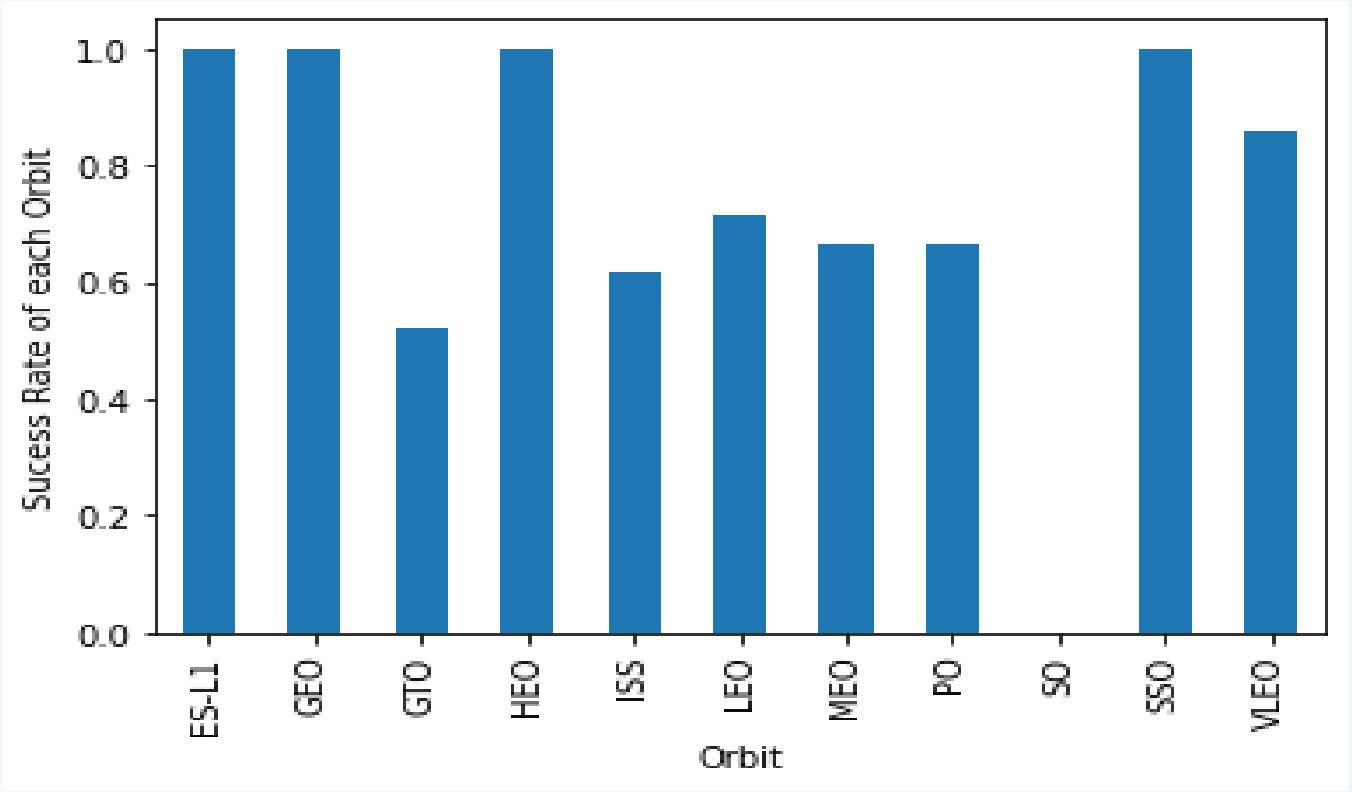
Payload vs. Launch Site



Heavier payload may be consider for a successful landing, too heavy payload can make a failed landing

Success Rate vs. Orbit Type

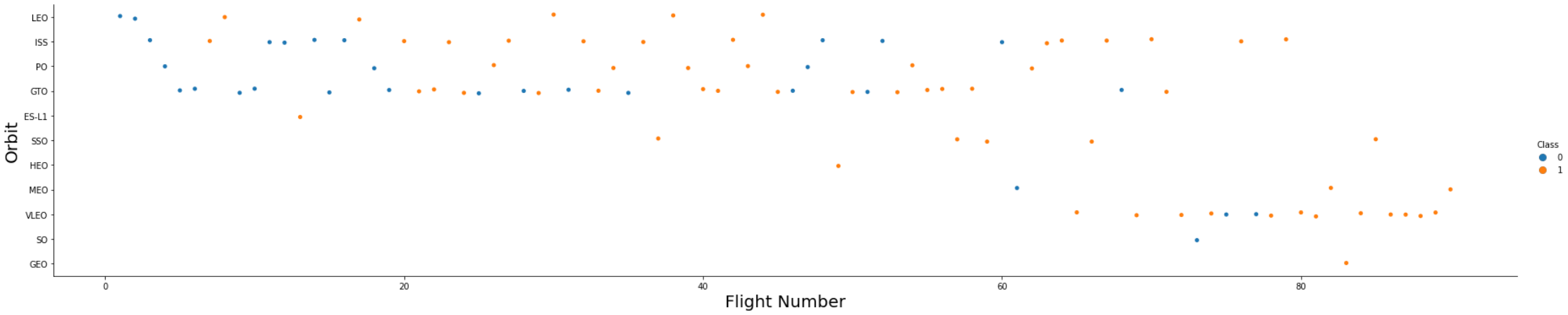
Bar chart for the success rate of each orbit type



ES-L1, GEO, HEO, SSO have the highest success rate

Flight Number vs. Orbit Type

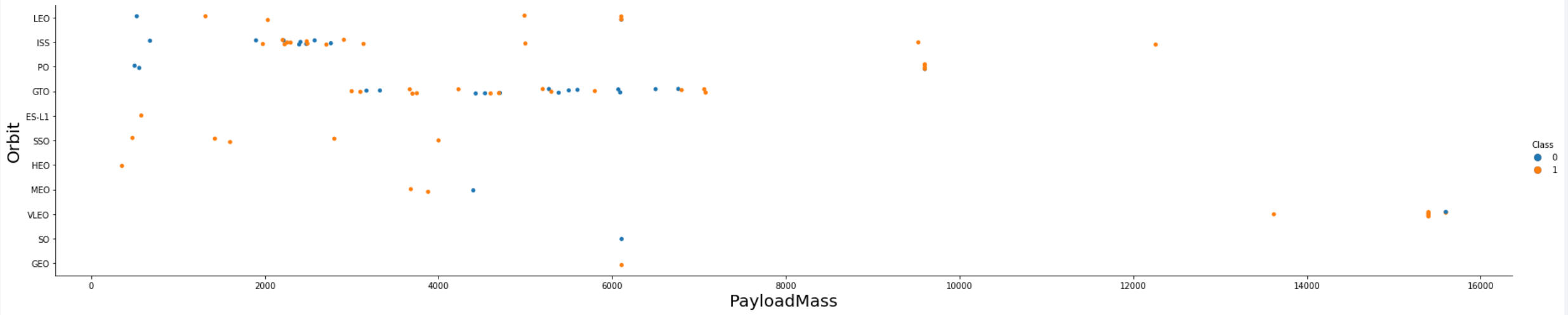
Flight number vs. Orbit type



success rate increases with the number of flights for the LEO orbit

Payload vs. Orbit Type

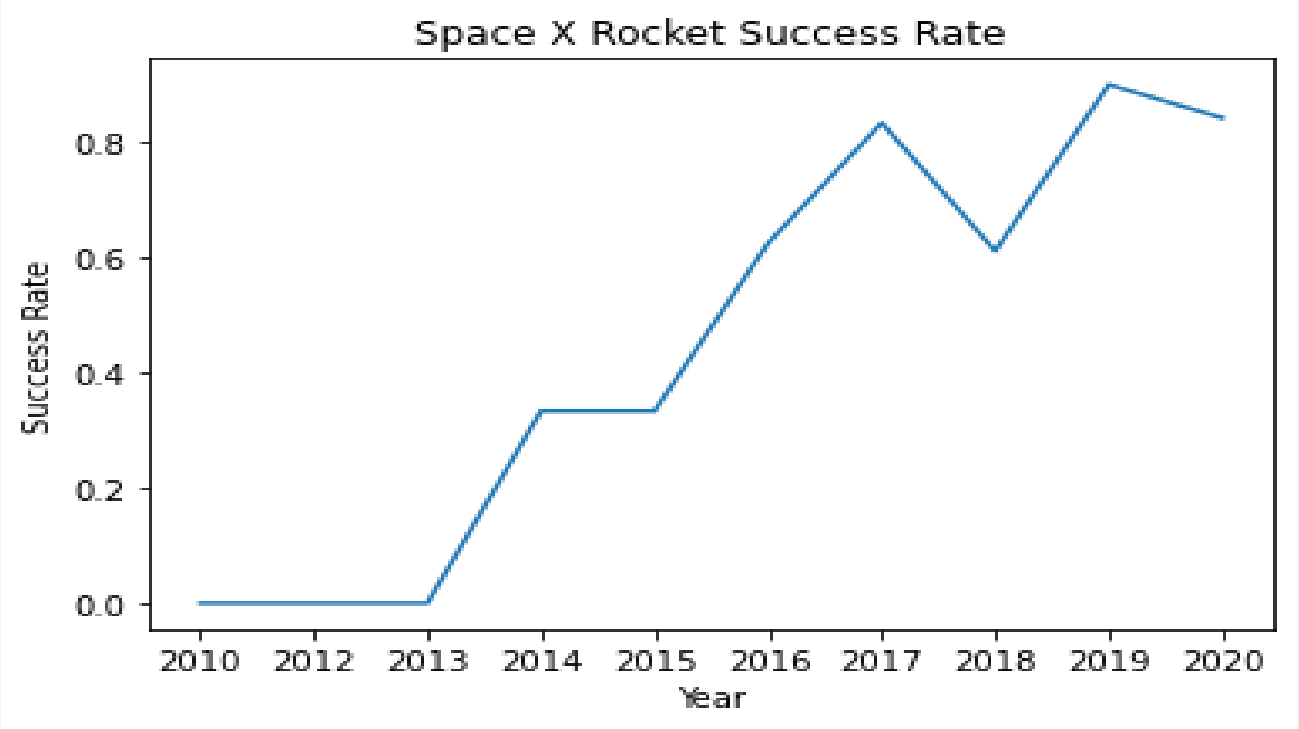
Payload vs. orbit type



Weight of the payloads have high influence on the success rate

Launch Success Yearly Trend

- Line chart of yearly average success rate



An increase in the Space X Rocket success rate since 2013

All Launch Site Names

- Find the names of the unique launch sites

SELECT DISTINCT “LAUNCH_SITE” FROM SPACEXTBL

- Present your query result with a short explanation here

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

Using DISTINCT query to remove duplicate values of LAUNCH_SITE.

Launch Site Names Begin with 'CCA'

- **Find 5 records where launch sites begin with `CCA`**

SELECT * FROM SPACEXTBL WHERE "LAUNCH_SITE" LIKE '%CCA%' LIMIT 5

- **Present your query result with a short explanation here**

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer
04-06-2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX
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
22-05-2012	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)
08-10-2012	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)
01-03-2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)

The condition WHERE and LIKE will filters launch sites contain the CCA.
LIMIT 5 only show 5 records

Total Payload Mass

- **Calculate the total payload carried by boosters from NASA**

SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "CUSTOMER" = 'NASA (CRS)'

- **Present your query result with a short explanation here**

SUM("PAYLOAD_MASS__KG_")
45596

Returns the sum of all payload where the customer is NASA (CRS)

Average Payload Mass by F9 v1.1

- **Calculate the average payload mass carried by booster version F9 v1.1**

SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "BOOSTER_VERSION" LIKE '%F9 v1.1%'

- **Present your query result with a short explanation here**

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Returns the average of all payload where the booster version contains the substring F9 v1.1

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad

SELECT MIN("DATE") FROM SPACEXTBL WHERE "Landing _Outcome" LIKE '%Success%’

- Present your query result with a short explanation here

MIN("DATE")
01-05-2017

Select the oldest successful landing

Successful Drone Ship Landing with Payload between 4000 and 6000

- **List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000**

SELECT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Success (drone ship)' AND "PAYLOAD_MASS__KG_" > 4000 AND "PAYLOAD_MASS__KG_" < 6000

- **Present your query result with a short explanation here**

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

Returns the booster version where landing was successful and payload mass is between 4000 and 6000 kg

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes

```
SELECT (SELECT COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE
"MISSION_OUTCOME" LIKE '%Success%') AS SUCCESS, (SELECT
COUNT("MISSION_OUTCOME") FROM SPACEXTBL WHERE
"MISSION_OUTCOME" LIKE '%Failure%') AS FAILURE
```

- Present your query result with a short explanation here

SUCCESS	FAILURE
100	1

First SELECT shows the 2 subqueries that return results:
- The first subquery counts the successful mission.
- The second subquery counts the unsuccessful mission

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass

SELECT DISTINCT "BOOSTER_VERSION" FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" = (SELECT max("PAYLOAD_MASS__KG_") FROM SPACEXTBL)

- Present your query result with a short explanation here

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

Subquery to filter data by returning the heaviest payload mass with MAX function. The main query uses subquery results and returns unique booster version (SELECT DISTINCT) with the heaviest payload mass

2015 Launch Records

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

```
SELECT substr("DATE", 4, 2) AS MONTH, "BOOSTER_VERSION", "LAUNCH_SITE"
FROM SPACEXTBL WHERE "LANDING _OUTCOME" = 'Failure (drone ship)' and
substr("DATE",7,4) = '2015'
```

- Present your query result with a short explanation here

MONTH	Booster_Version	Launch_Site
01	F9 v1.1 B1012	CCAFS LC-40
04	F9 v1.1 B1015	CCAFS LC-40

Returns month, booster version, launch site where landing was unsuccessful and landing date took place in 2015

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

- Rank 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

```
SELECT "LANDING _OUTCOME", COUNT("LANDING _OUTCOME") FROM SPACEXTBL WHERE "DATE" >= '04-06-2010' and "DATE" <= '20-03-2017' and "LANDING _OUTCOME" LIKE '%Success%‘ GROUP BY "LANDING _OUTCOME" ORDER BY COUNT("LANDING _OUTCOME") DESC ;
```

- Present your query result with a short explanation here

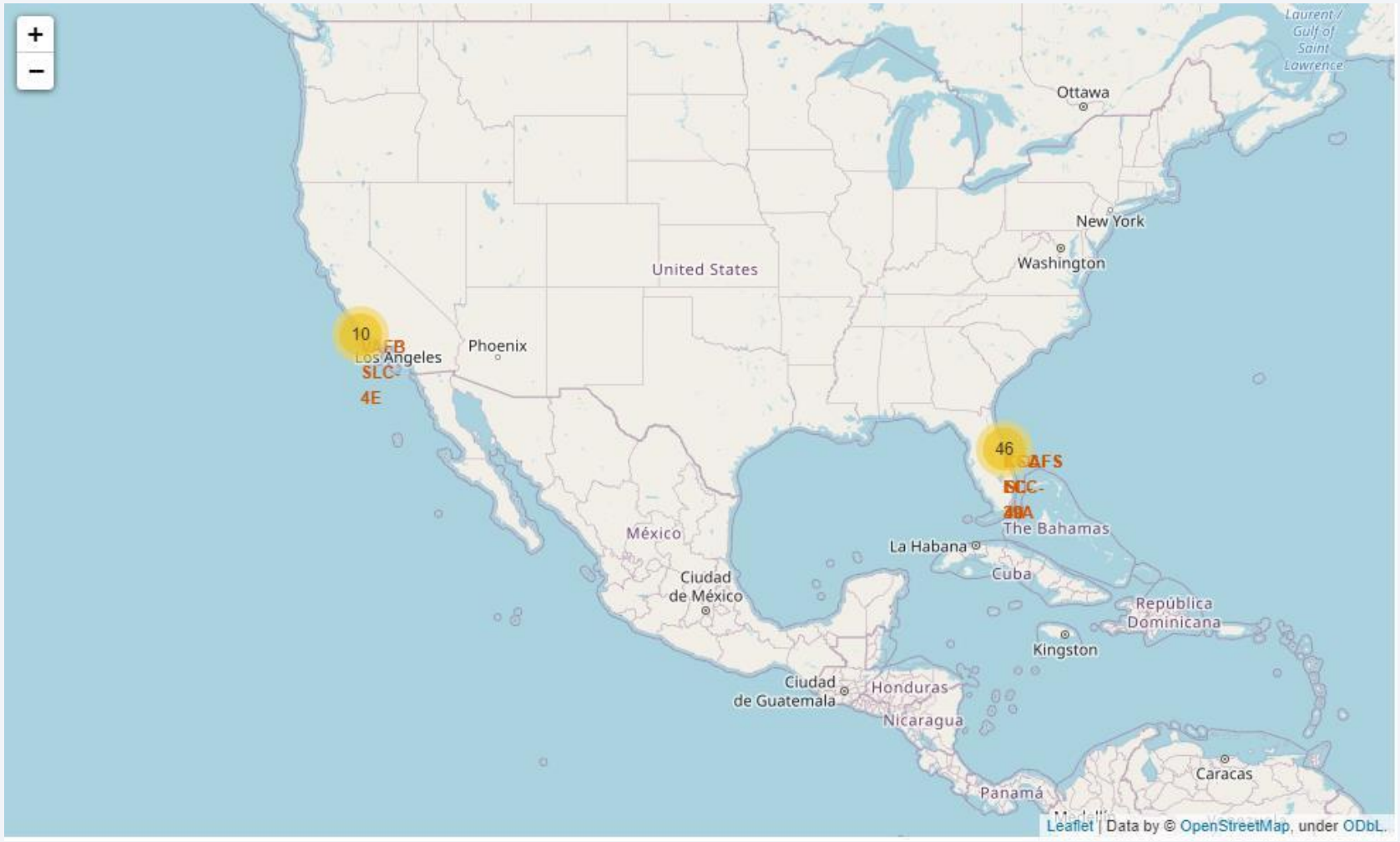
Landing _Outcome	COUNT("LANDING _OUTCOME")
Success	20
Success (drone ship)	8
Success (ground pad)	6

Returns landing outcomes and their count where mission was successful and date is between 04/06/2010 and 20/03/2017

Section 3

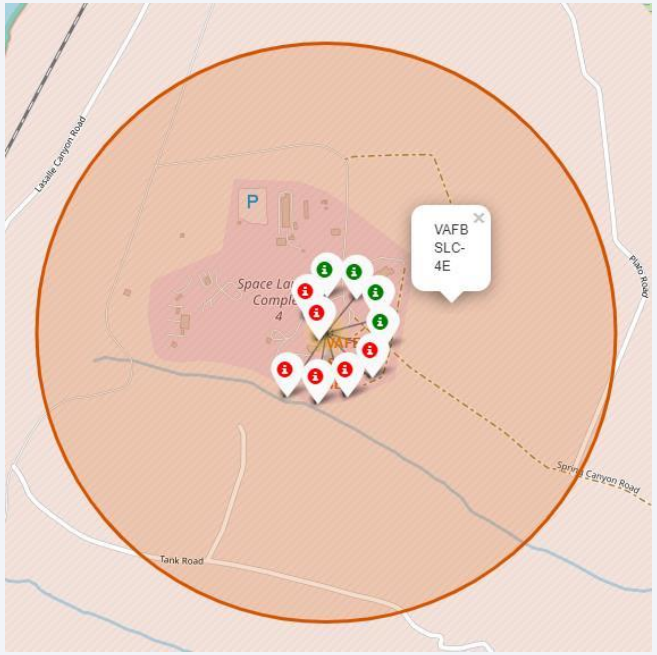
Launch Sites Proximities Analysis

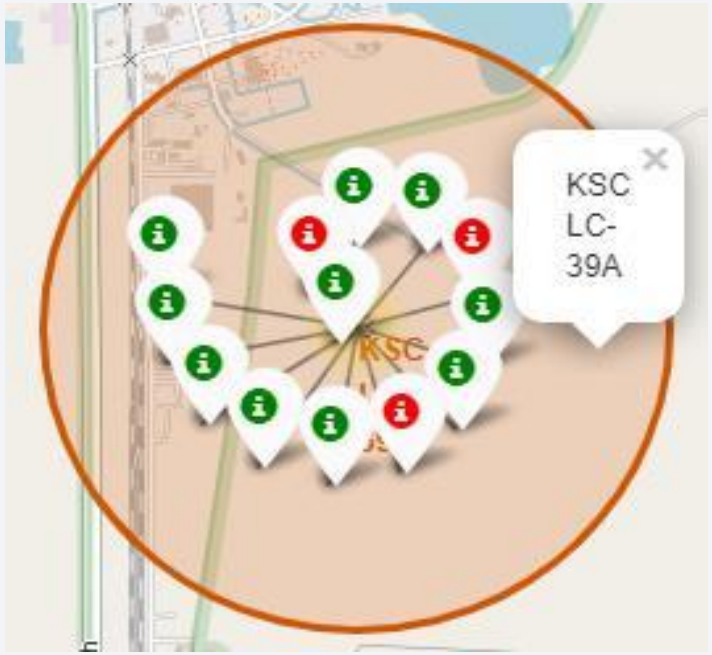
<Folium Map Screenshot 1>

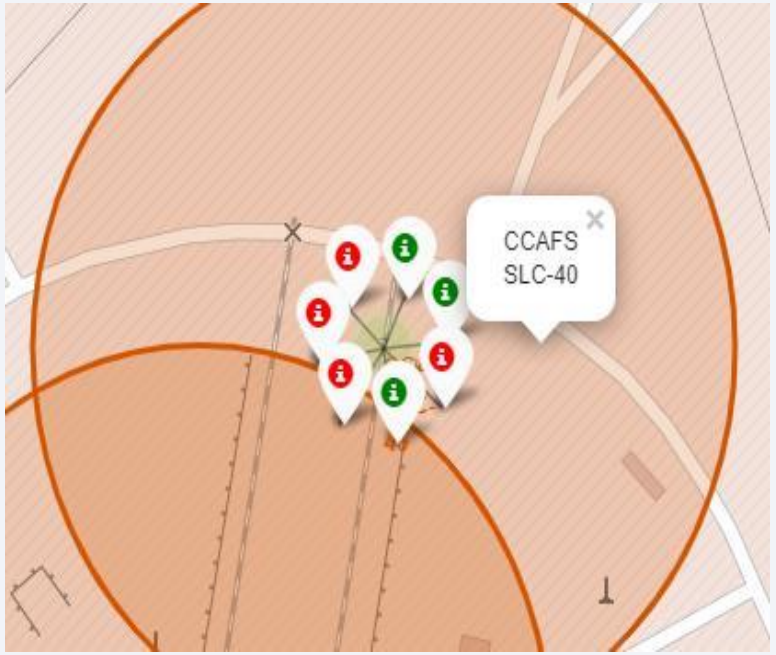


SpaceX launch sites are located on the coast of the United States

<Folium Map Screenshot 2>

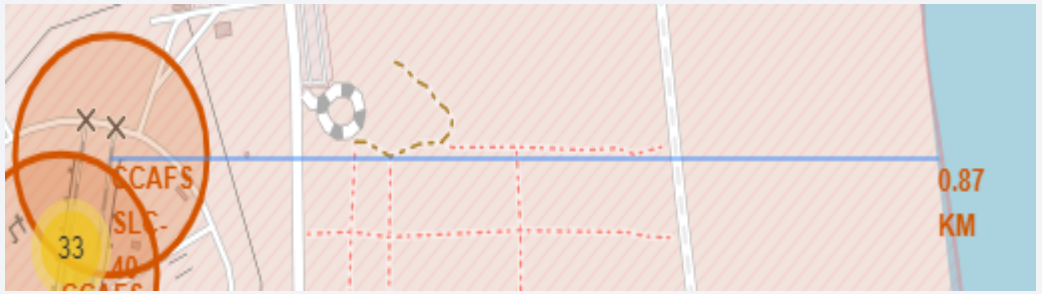


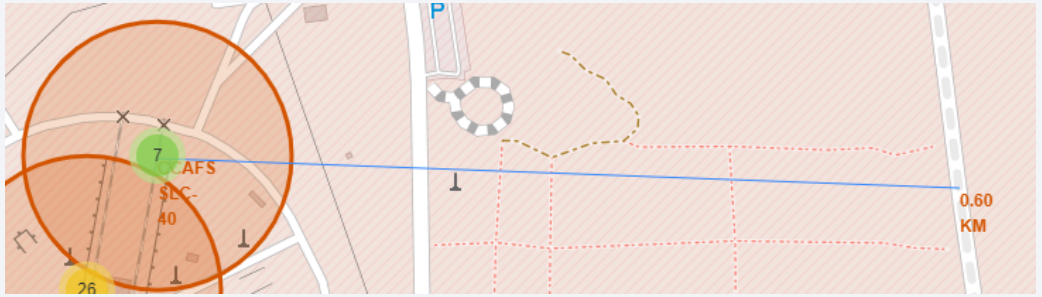


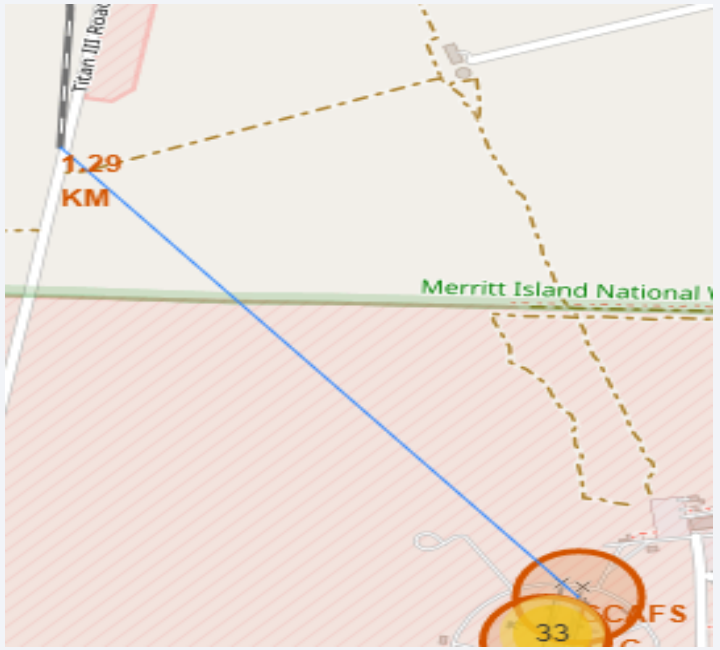


Green show successful launches.
Red show unsuccessful launches.

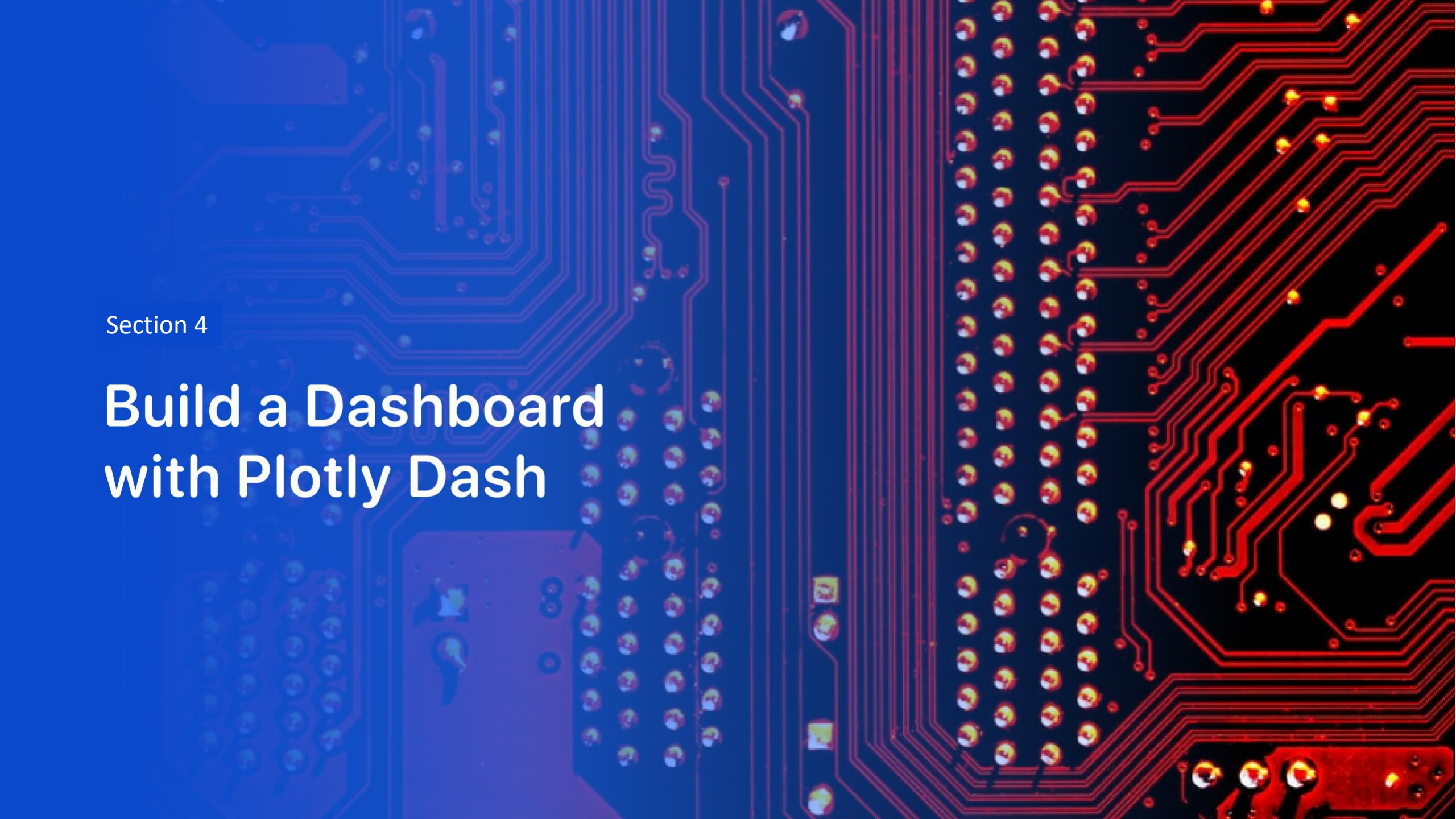
<Folium Map Screenshot 3>



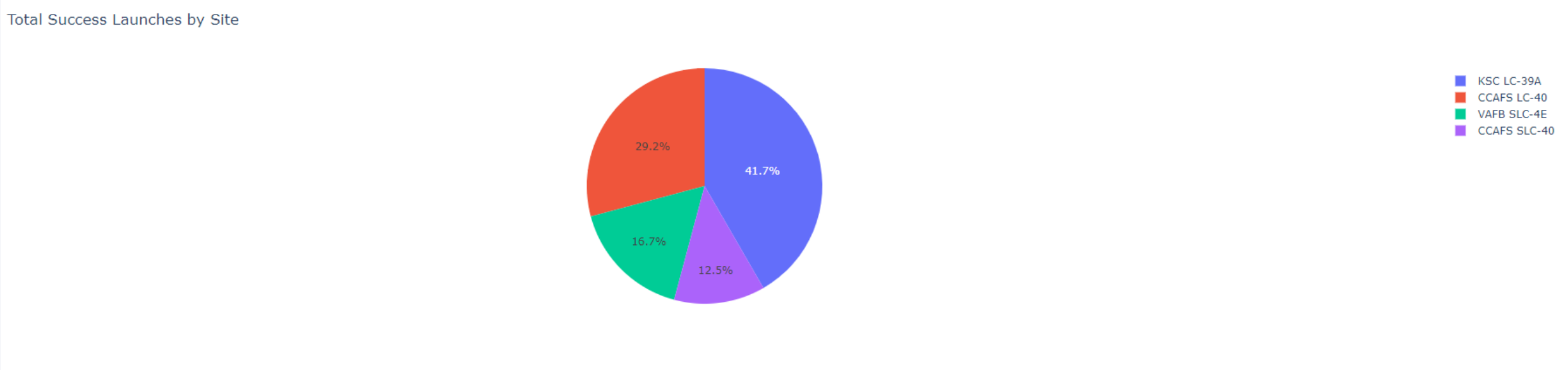




ICCAFS SLC-40 in close proximity to railways? Yes
CCAFS SLC-40 in close proximity to highways? Yes
CCAFS SLC-40 in close proximity to coastline? Yes
CCAFS SLC-40 keeps certain distance away from cities? No

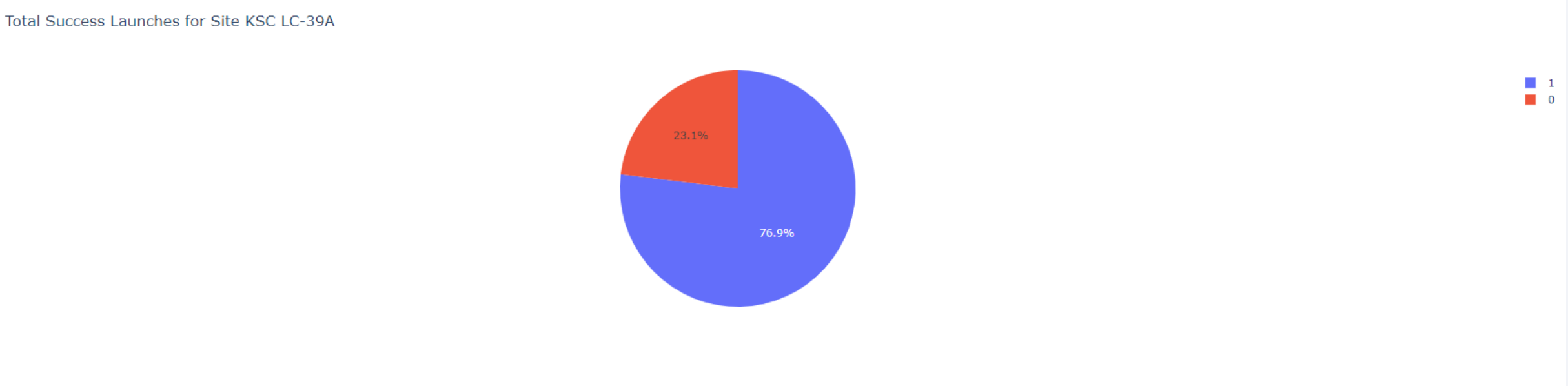

Section 4
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



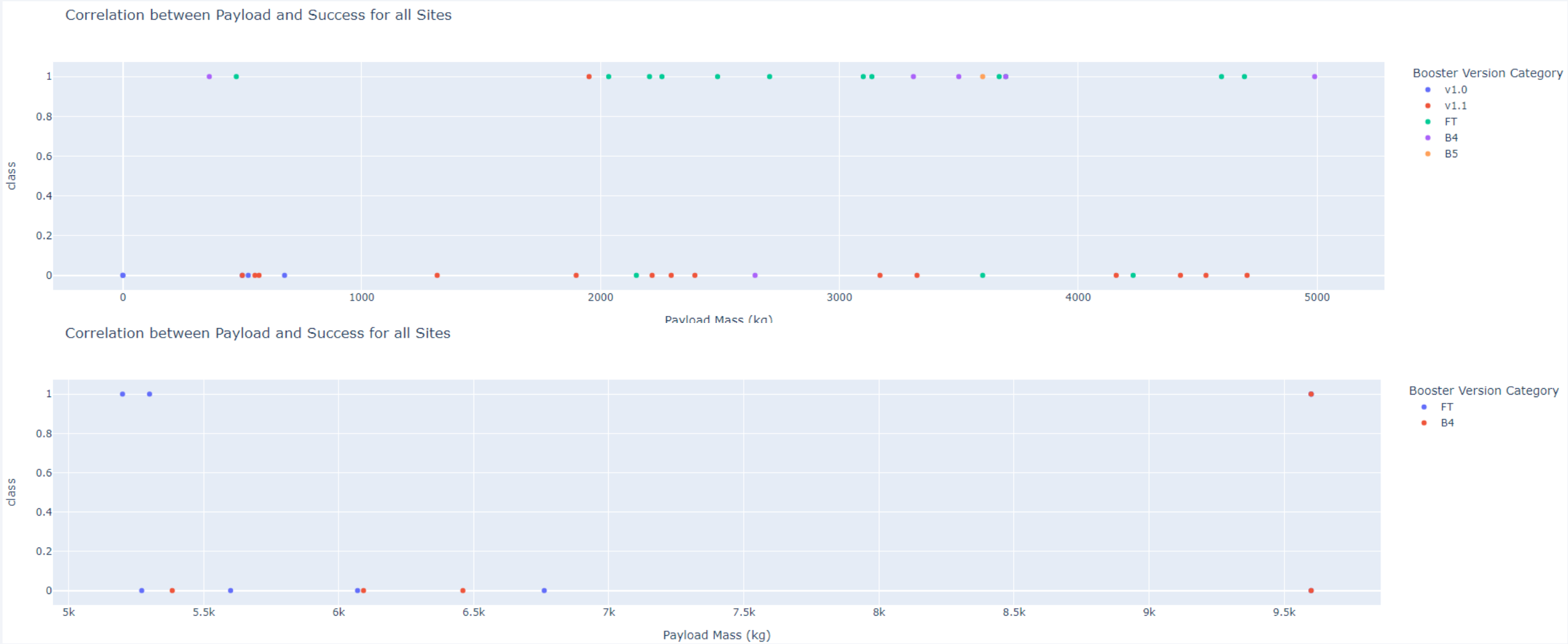
KSC LC-39A has the highest success rate of launches.

<Dashboard Screenshot 2>



KSC LC-39A has 76.9% success rate and 23.1% failed rate

<Dashboard Screenshot 3>

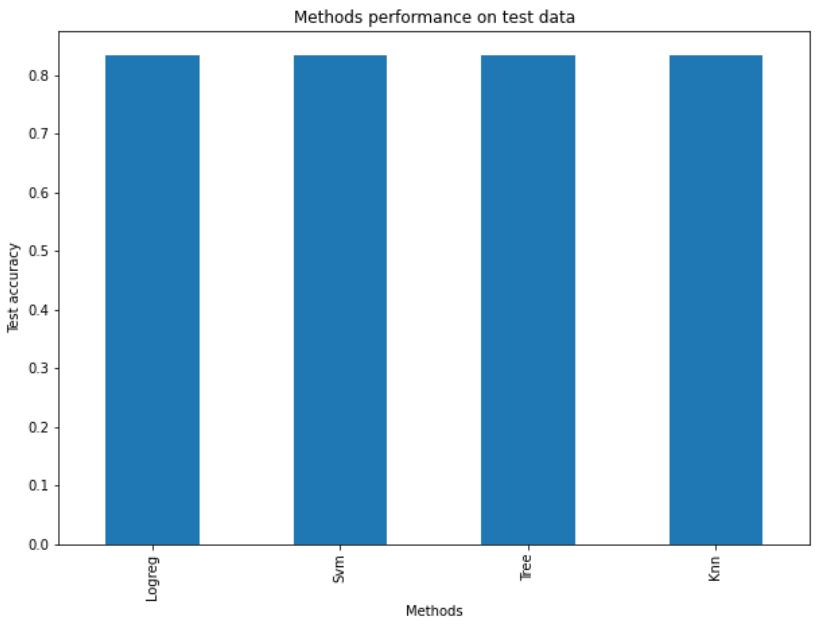


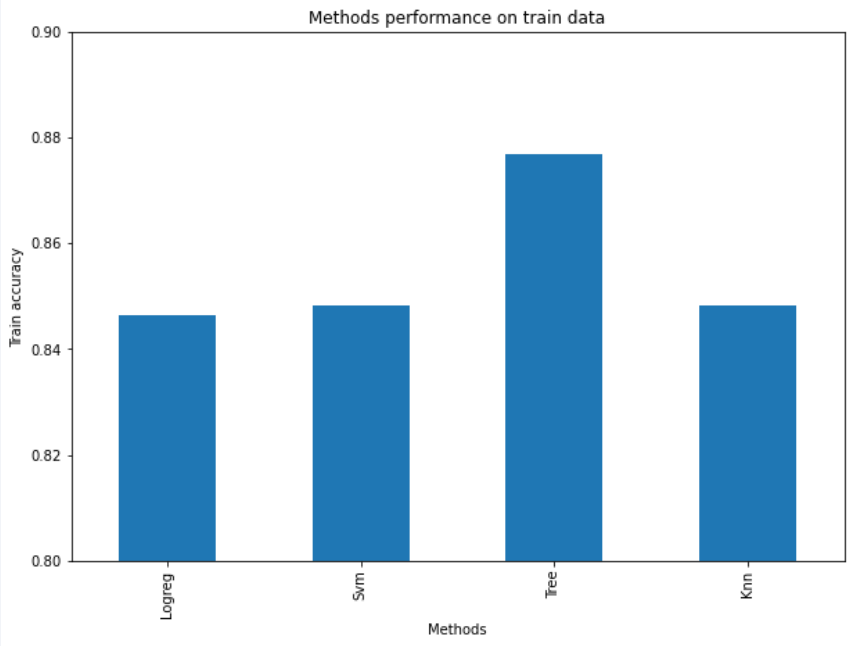
Low payloads have a better success rate than the heavy payloads

Section 5

Predictive Analysis (Classification)

Classification Accuracy





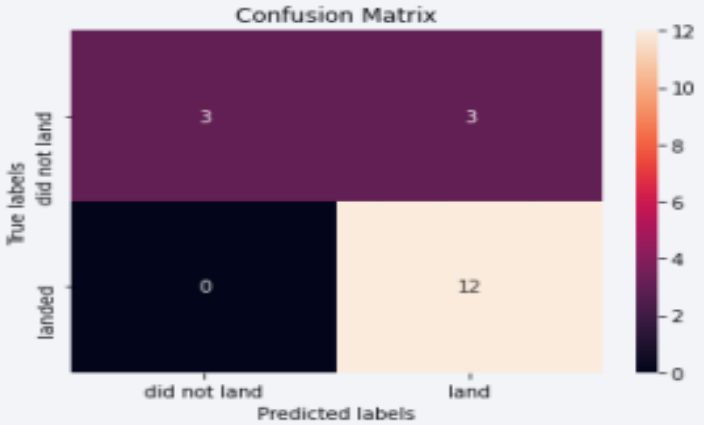
	Accuracy Train	Accuracy Test
Tree	0.876786	0.833333
Knn	0.848214	0.833333
Svm	0.848214	0.833333
Logreg	0.846429	0.833333

```
print("tuned hyperparameters :(best parameters) ", tree_cv.best_params_)
print("accuracy :", tree_cv.best_score_)
```

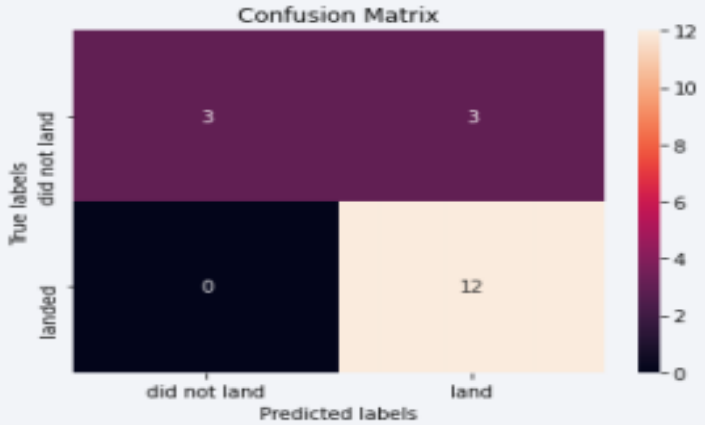
```
tuned hyperparameters :(best parameters)  {'criterion': 'entropy', 'max_depth': 12, 'max_features': 'sqrt', 'min_samples_leaf':
4, 'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.8767857142857143
```

Confusion Matrix

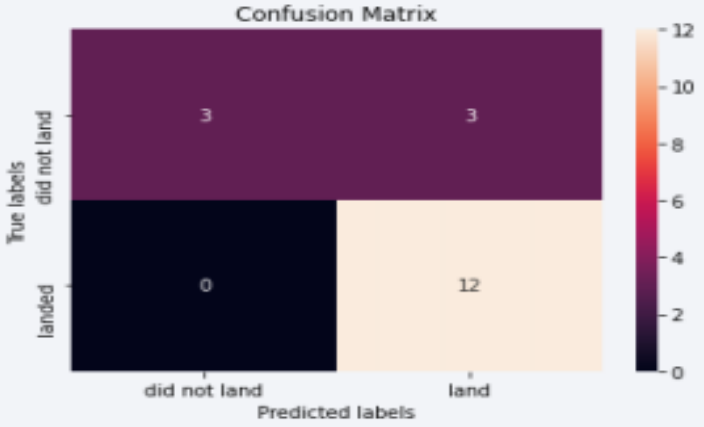
Logistic regression



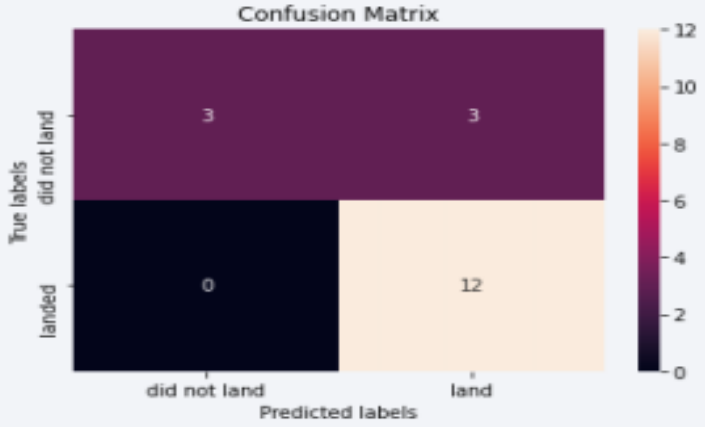
Decision Tree



kNN



SVM



Conclusions

- GEO, HEO, SSO, ES-L1 are the orbits with best success rates
- Payload is a characteristic for the success of a landing.
- KSC LC-39A is the best launch site
- Decision Tree Algorithm as the best model because it has a better accuracy

Thank you!