IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Ravi Kiran Upadrasta>
<21 – July - 2023>

Outline

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

Executive Summary

- Summary of methodologies
  - Data Collection through API and webscraping
  - Data Wrangling
  - Exploratory Data Analysis with SQL and Data Visualizations
  - Interactive Visual Analytics with Folium and Plotly Dash
  - Machine Learning Prediction using various algorithms
- Summary of all results
  - Identified critical factors for success rate of launch incl payload, launch site, orbit
  - Found the best prediction algorithm for the data

Introduction

- Project background and context
  - Falcon 9, rocket designed by SpaceX claims to reduce the cost of rocket launch as it can use the first phase. Depending on various factors like Launch Site, Payload, customer, orbit, SpaceX decides whether to reuse the first stage or not. SpaceY wants to compete with SpaceX on developing a similar rocket. Goal is to create a machine learning pipeline to predict if the first stage will land successfully.
- Problems you want to find answers
  - Factors determining the successful landing of the rocket – like Launch Site, Orbit etc
  - Relation between these factors, if any
  - Determine the price of each launch

Section 1

Methodology

Methodology

Executive Summary

- Data collection methodology:
  - Data Sets were collected using the Space X APIs and web scraping from wikipedia
- Perform data wrangling
  - 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
  - Build, tune, evaluate classification models

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

Data Collection – SpaceX API

1) Used responses.get to receive data from the SpaceXAPI
2) Used json_normalize() to convert json response to pandas dataframe
3) Used API to get specific information for the columns - rocket, payloads, launchpad and cores (functions were pre-defined to get this information). This data is stored in lists
4) A dictionary is created using the data obtained in step 3 which is then converted to dataframe using pd.Dataframe
5) Dataframe is filtered to retain only data pertaining to Falcon 9 rocket
6) Empty Values of Payload Mass are replaced with the mean payload mass.
7) This data is then exported to a csv file using .to_csv()

#Code snippets are provided in the annexure

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

Data Collection - Scraping

1) Used requests.get() [HTTP GET method] to request Falcon 9 Wikipedia html page.
2) A Beautifulsoup object is created from the above response
3) Used find_all() to get all the tables from the soup object
4) Iterated through the table headers to extract column names. An empty dictionary is created with all the column names
5) Parsed the tables to populate the dictionary with values from respective columns
6) Dictionary converted into dataframe using .DataFrame()
7) File is exported to csv using .to_csv()

#codesnippets provided in annexures

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-webscraping.ipynb

Data Wrangling

1) Used .value_counts() to determine # of rockets from various launch sites, various orbits and the mission outcomes
2) Segregated the landing outcomes and created a new column “Class” as a landing outcome label
3) Found the overall success rate by measuring the mean of the Class column created above.

#codesnippets provided in annexures

https://github.com/istupe/DS_Capstone/blob/main/labs-jupyter-spacex-Data%20wrangling.ipynb

EDA with Data Visualization

Used Data Visualization to identify patterns or trends among the variables and with the dependent variable i.e. launch outcome.

Following Data Visualizations were explored:

1) Scatter Plot – to understand relation between landing outcome, flight number and payload mass
2) Scatter Plot – to understand relation between landing outcome, launch site and Flight number
3) Scatter Plot – to understand relation between payload mass, launch site and landing outcome
4) Bar Chart – to understand relation between landing outcome and orbit
5) Scatter Plot – to understand relation between landing outcome, Flight number and orbit
6) Scatter Plot – to understand relation between landing outcome, payload mass and orbit
7) Line graph – to understand yearly trend(s) in landing outcome success

#Graphs provided in Section 2

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-eda-dataviz.ipynb

EDA with SQL

Used SQL queries to identify the following:

1) Unique Launch sites in the mission
2) 5 records with Launch Site beginning with the string CCA
3) Total payload mass carried by boosters launched by NASA (CRS)
4) Average payload mass carried by booster version F9 v1.1
5) Date when the first succesful landing outcome in ground pad was acheived
6) Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
7) Total number of successful and failure mission outcomes
8) Names of the booster_versions which have carried the maximum payload mass
9) Records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.
10) Rank the count of landing outcomes

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

- Marked and added all launch sites as circles, success or failure of launches as markers and distance from railway line, airport, sea and city as lines. Added these objects to answer few questions like
  - Are the launch sites in proximity to equator (similar latitudes) or a coast – **YES. The launch sites are either on west coast or east coast of the United States**
  - Color coding the markets (0 or failure as red and 1 or success as green) helped in identifying performance of a launch site.
  - Distances to understand if launch sites are closer or farther from railways, highways, coastline or nearest city – **Launch Sites seem closer to railways, highways and coastline but seem far from cities**

https://github.com/istupe/DS_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

To view along with maps use the nbviewer link as github isn't rendering maps
https://nbviewer.org/github/istupe/DS_Capstone/blob/main/lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

- Built a dashboard with option to input a) Launch Site as dropdown and b) Payload Mass as a slider
- When the Launch Site is selected, It shows a pie chart of the total launches from the selected site with distribution between successful and unsuccessful launches.
- Then we created a scatter plot with inputs as Launch Site (From dropdown) and Payload mass (from slider) and output indicating correlation between payload and success.

https://github.com/istupe/DS_Capstone/blob/main/spacex_dash_app.py

Predictive Analysis (Classification)

1) Created a numpy array from dataframe using to_numpy(). Standardized the data using transform and split the data into training and testing with test size of 20%

2) Built different machine learning models (Logistic Regression, SVM, Decision tree Classifer, and K Nearest Neighbours) with hyperparameters using GridSearchCV.

3) Calculated the accuracy of the models using score and plotted the confusion matrix to identify the best method

https://github.com/istupe/DS_Capstone/blob/main/SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite.ipynb

Results

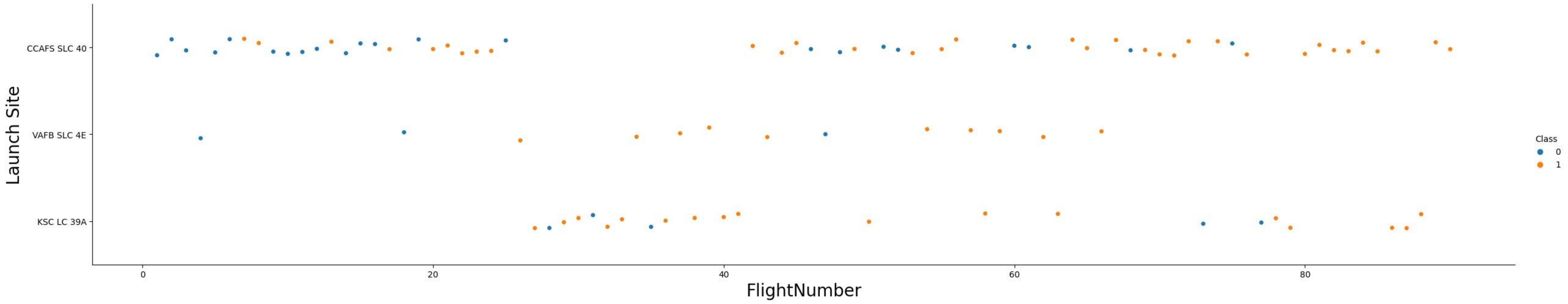
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

<u>These are presented in Section 2 of the presentation</u>

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site



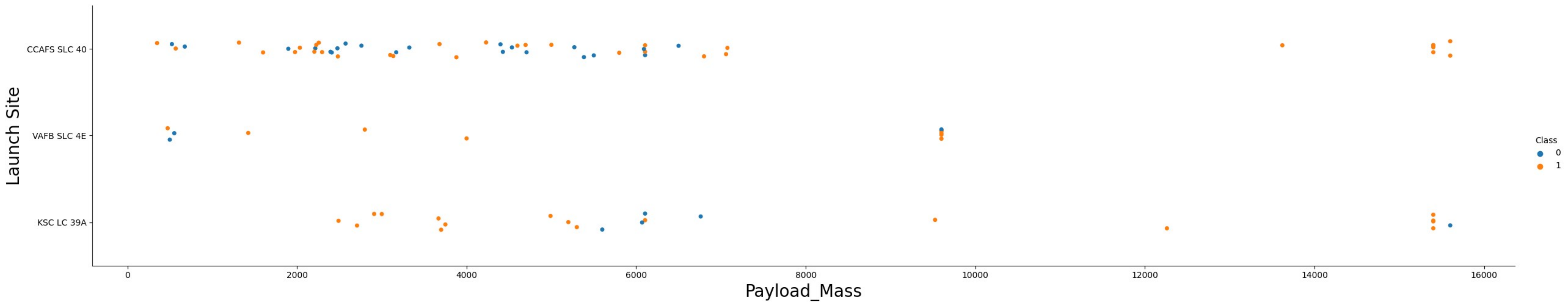
From the Visualization we can notice that:

1) Earlier flights lauch were from CCAFS-SLC-40 site ,Followed by KSC-LC-39A

2) Most Launches are Launched from CCAFS-SLC-40 and least Launches from VAFB SLC 4E

3) For a brief period during Launches 20 to 40, launches were shifted from CCAFS SLC 40 to KSC LC 39A

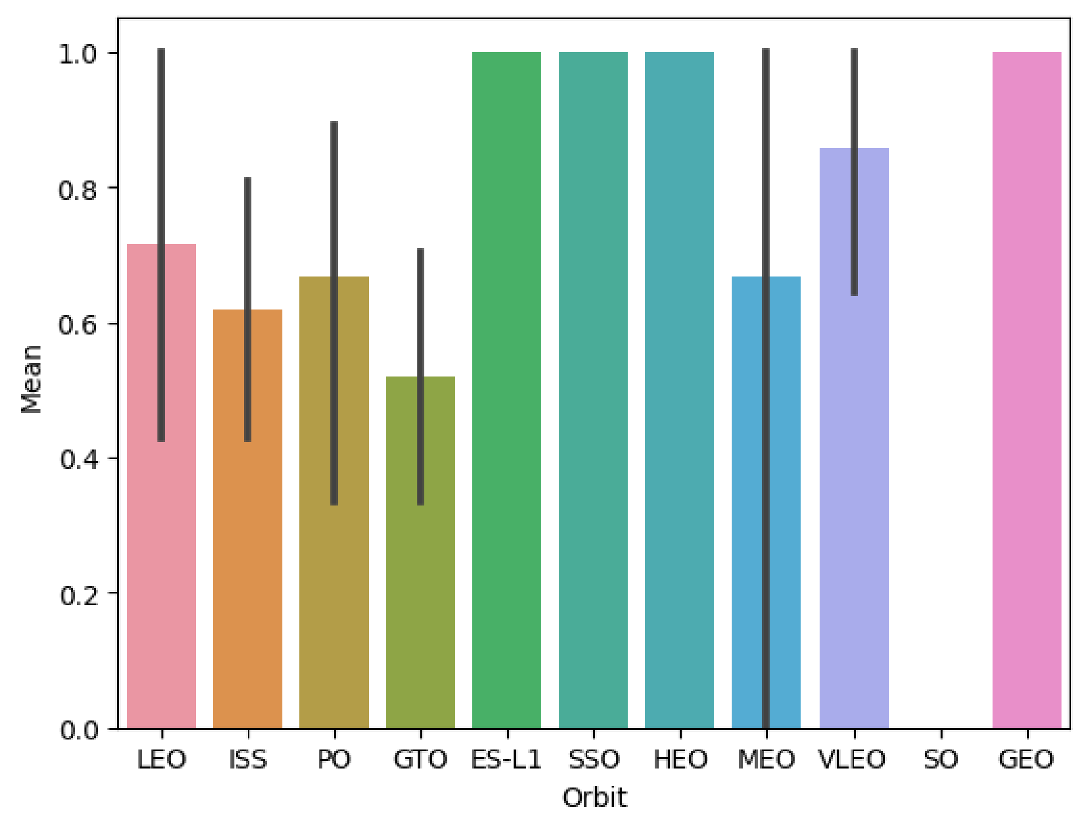
Payload vs. Launch Site



From the Visualization we can notice that:

1) Payloads over 10000 KG are not launched at VAFB SLC 4E
2) Higher payloads (above 14000 kg) seem to have higher success rate

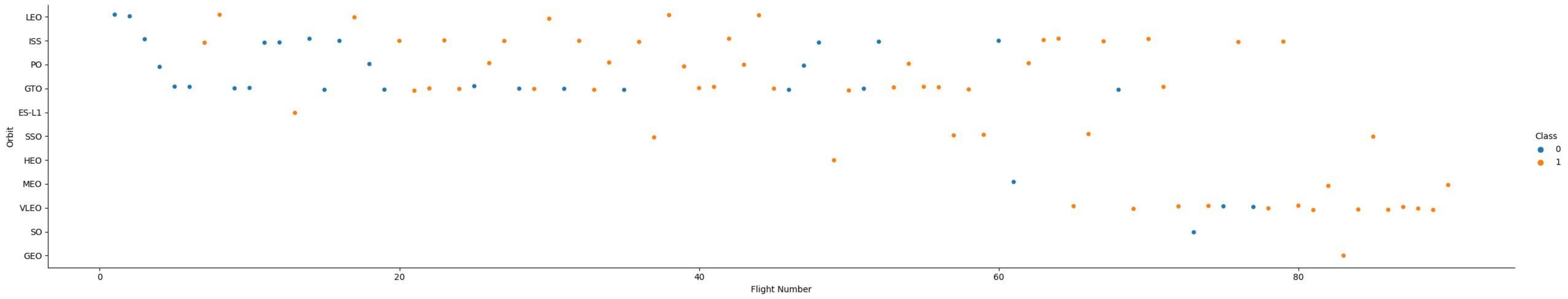
Success Rate vs. Orbit Type



From the Visualization we can notice that:

1) ES-L1, GEO, HEO, SSO, VLEO had the most success rate.

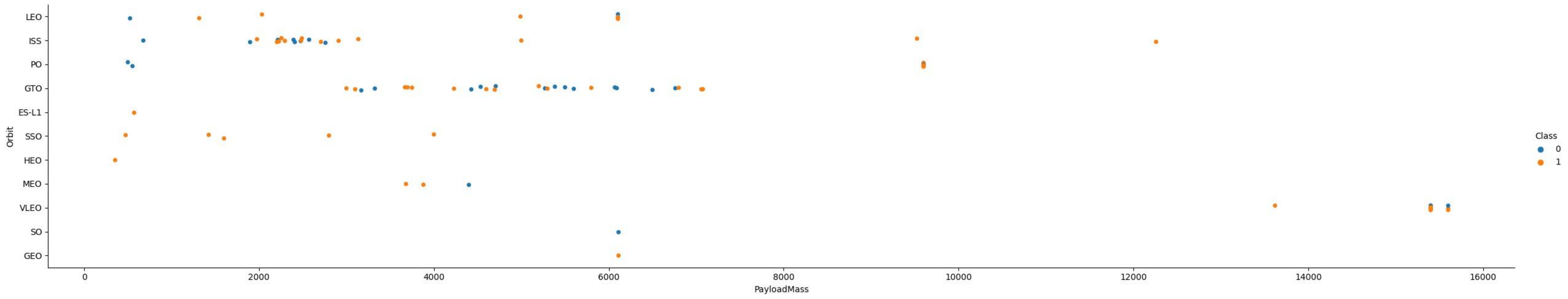
Flight Number vs. Orbit Type



From the Visualization we can notice that:

1) VLEO orbit flights came much later but have good success rate along with SSO orbit
2) Initially lot of failures in GTO and ISS orbit launches
3) GTO and ISS also seem to have the most launches followed by VLEO

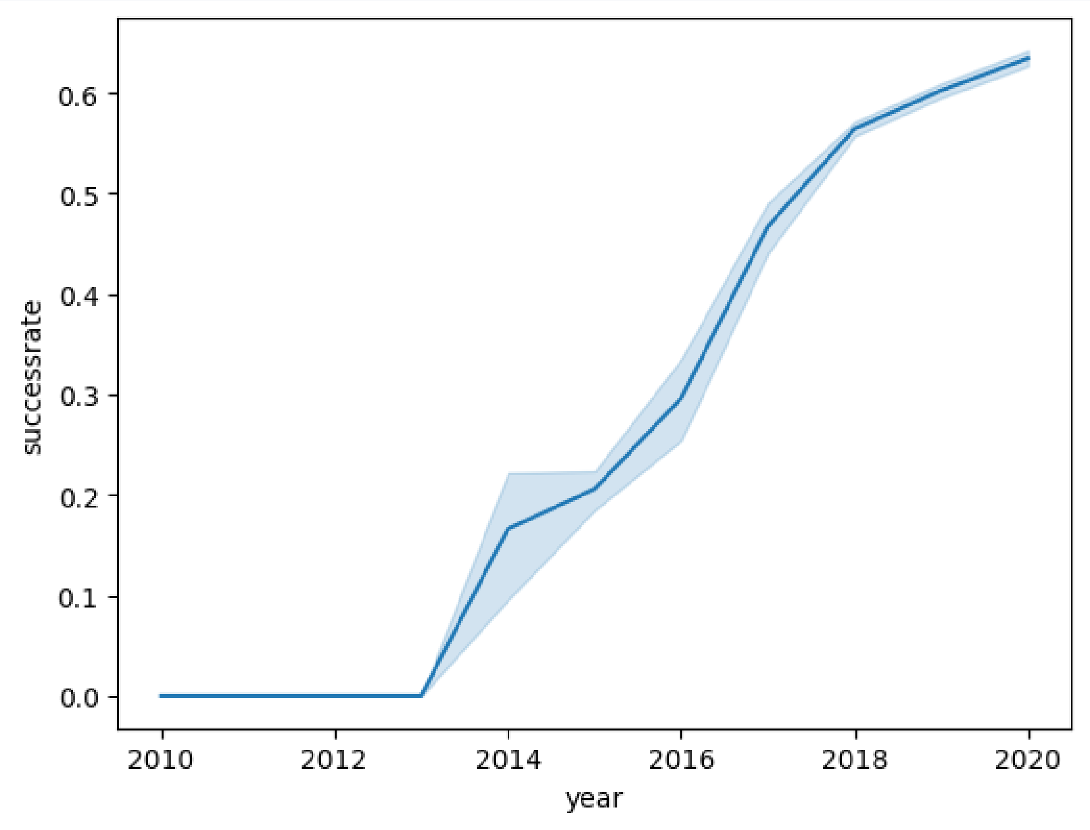
Payload vs. Orbit Type



From the Visualization we can notice that:

1) Most of the ISS payloads are between 2000 to 4000. Most of the GTO payloads are between 3000 and 7000. VLEO has only higher payloads. HEO, ES-L1 and have very small payloads
2) With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
3) For GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend



From the Visualization we can notice that:

1) Success rate kept increasing from 2013 onwards

All Launch Site Names

Display the names of the unique launch sites in the space mission

In [9]:
```
%sql select distinct(Launch_site) from SPACEXTBL
```

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

Out[9]:

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

To get the unique launch sites, we used distinct (launch_site) from the database. There are 4 unique launch sites

Launch Site Names Begin with 'CCA'

Display 5 records where launch sites begin with the string 'CCA'

In [14]:

```
%%sql
select * from SPACEXTBL where Launch_site like 'CCA%' LIMIT 5
```

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

Out[14]:

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 atte
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No atte
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No atte

We used the condition Launch_site like 'CCA%' to identify launch sites whose name begins with CCA and limited to 5 records using LIMIT keyword

Total Payload Mass

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [18]:
```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'
```

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

Out[18]:

sum(PAYLOAD_MASS__KG_)
45596.0

Used the Sum(Payload_mass__Kg_) with the condition that customer is NASA (CRS) to calculate total payload

Average Payload Mass by F9 v1.1

Display average payload mass carried by booster version F9 v1.1

In [23]:
```
%sql select avg(PAYLOAD_MASS__KG_) from SPACEXTBL where Booster_Version = 'F9 v1.1'
```

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

Out[23]:

avg(PAYLOAD_MASS__KG_)
2928.4

Used the Avg(Payload_mass__Kg_) with the condition that Booster_Version is F9 v1.1 to calculate average payload

First Successful Ground Landing Date

In [46]:
```
#to check various distinct landing outcomes
#%sql select DISTINCT(Landing_Outcome) from SPACEXTBL

%sql select Date as First_Successful_Landing, Landing_Outcome from SPACEXTBL where Landing_Outcome = 'Success (ground pad)'

#formula using min function
# %sql select min(date) as First_Successful_landing from SPACEXTBL where Landing_Outcome like 'Success (ground pad)'
```

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

Out[46]:

First_Successful_Landing	Landing_Outcome
22/12/2015	Success (ground pad)

We can get the required result by either query

%sql select Date as First_Successful_Landing, Landing_Outcome from SPACEXTBL where Landing_Outcome = 'Success (ground pad)' order by Date desc LIMIT 1

or

%sql select min(date) as First_Successful_landing from SPACEXTBL where Landing_Outcome like 'Success (ground pad)'

Successful Drone Ship Landing with Payload between 4000 and 6000

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [56]:

```
# selected payload mass for verification
%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL where Landing_Outcome like 'Success (drone ship)'  and PAYLOAD
```

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

Out[56]:

Booster_Version	PAYLOAD_MASS__KG_
F9 FT B1022	4696.0
F9 FT B1026	4600.0
F9 FT B1021.2	5300.0
F9 FT B1031.2	5200.0

%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ from SPACEXTBL
where Landing_Outcome like 'Success (drone ship)' and PAYLOAD_MASS__KG_ between 4000 and 6000

Total Number of Successful and Failure Mission Outcomes

In [58]:

```
%sql select Mission_Outcome, count(Mission_Outcome) as Total from SPACEXTBL group by Mission_Outcome
```

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

Out[58]:

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

Used count of mission_outcome to get the number of successful and failure mission outcomes

Boosters Carried Maximum Payload

```
In [61]: %sql SELECT Booster_Version, PAYLOAD_MASS__KG_ as Payload_Mass from SPACEXTBL where PAYLOAD_MASS__KG_ in (select max(PAYLOAD

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

Out[61]:

Booster_Version	Payload_Mass
F9 B5 B1048.4	15600.0
F9 B5 B1049.4	15600.0
F9 B5 B1051.3	15600.0
F9 B5 B1056.4	15600.0
F9 B5 B1048.5	15600.0
F9 B5 B1051.4	15600.0
F9 B5 B1049.5	15600.0
F9 B5 B1060.2	15600.0
F9 B5 B1058.3	15600.0
F9 B5 B1051.6	15600.0
F9 B5 B1060.3	15600.0
F9 B5 B1049.7	15600.0

%sql SELECT Booster_Version, PAYLOAD_MASS__KG_ as Payload_Mass from SPACEXTBL where PAYLOAD_MASS__KG_ in (select max(PAYLOAD_MASS__KG_) from SPACEXTBL)

Used a subquery to get the maximum payload and used it to match with payload to get the boosters carrying max payload

2015 Launch Records

```
In [80]:
%%sql

SELECT
  CASE
    WHEN substr(Date, 4,2) = '01' THEN 'January'
    WHEN substr(Date, 4,2) = '02' THEN 'February'
    WHEN substr(Date, 4,2) = '03' THEN 'March'
    WHEN substr(Date, 4,2) = '04' THEN 'April'
    WHEN substr(Date, 4,2) = '05' THEN 'May'
    WHEN substr(Date, 4,2) = '06' THEN 'June'
    WHEN substr(Date, 4,2) = '07' THEN 'July'
    WHEN substr(Date, 4,2) = '08' THEN 'August'
    WHEN substr(Date, 4,2) = '09' THEN 'September'
    WHEN substr(Date, 4,2) = '10' THEN 'October'
    WHEN substr(Date, 4,2) = '11' THEN 'November'
    WHEN substr(Date, 4,2) = '12' THEN 'December'
  END AS month_name,
  substr(Date, 4,2) as month, Landing_outcome, Booster_Version, Launch_Site from SPACEXTBL where substr(Date,7,4)='2015' and

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

Out[80]:

month_name	month	Landing_Outcome	Booster_Version	Launch_Site
February	02	No attempt	F9 v1.1 B1014	CCAFS LC-40
April	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40
April	04	No attempt	F9 v1.1 B1016	CCAFS LC-40
June	06	Precluded (drone ship)	F9 v1.1 B1018	CCAFS LC-40
October	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
November	11	Controlled (ocean)	F9 v1.1 B1013	CCAFS LC-40

substr(Date, 4,2) as month, Landing_outcome, Booster_Version, Launch_Site from SPACEXTBL where substr(Date,7,4)='2015' and Landing_Outcome not like '%Success%' order by month

To get the failure landing outcomes, used CASE to get the name of the month with the extracted month number,

Used where for identifying year and landing outcome doesn't contain the word success

Results ordered by month chronologically

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

descending order.

In [91]:

```
%%sql

select Landing_Outcome as OUTCOME,count(Landing_Outcome) as TOTAL
from SPACEXTBL
where date between '04-06-2010' and '20-03-2017'
Group by landing_outcome order by total desc
```

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

Out[91]:

OUTCOME	TOTAL
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

To get the landing outcomes by ranking, we used count to get the total landing outcomes

Then used group by which groups as per the landing outcome

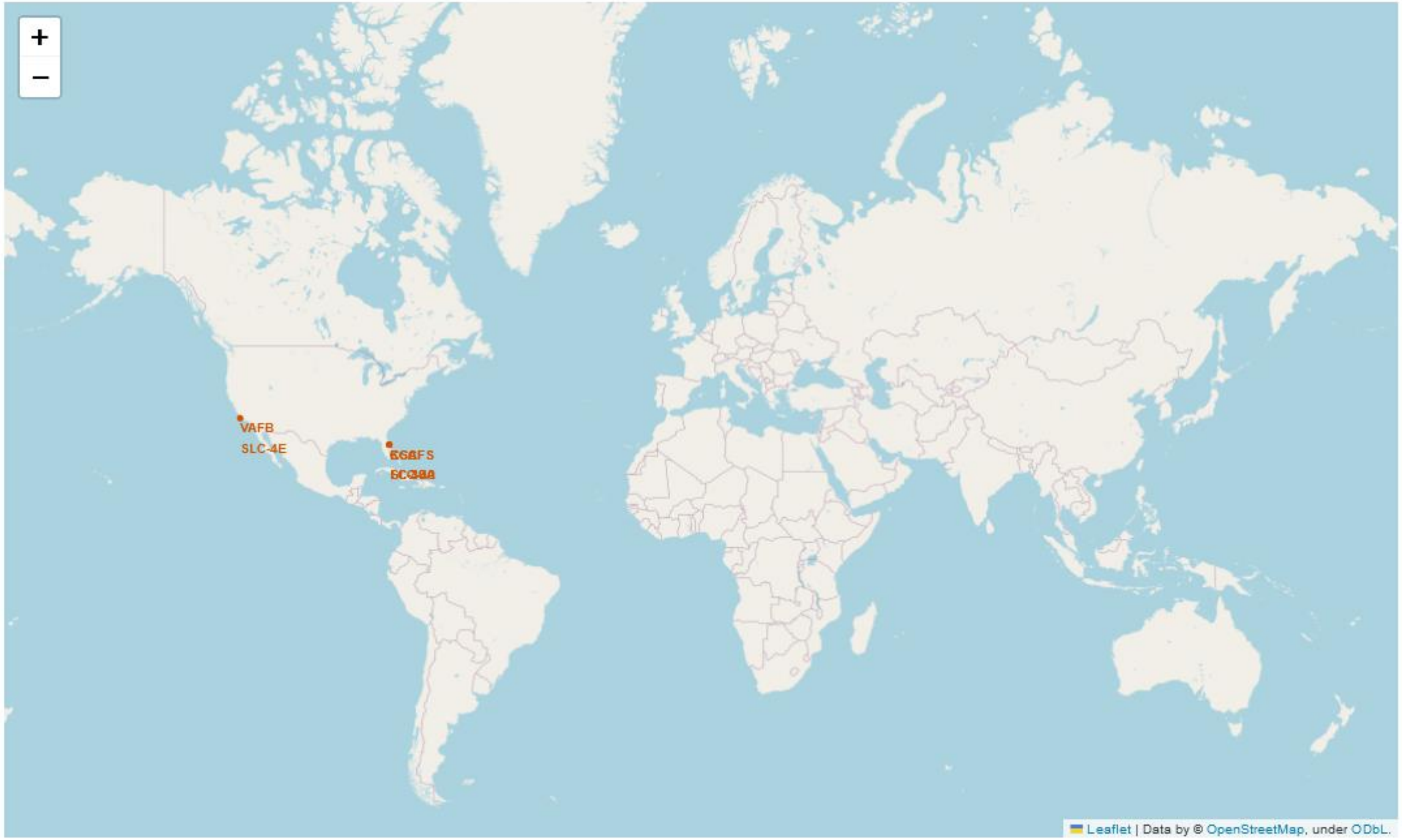
Used order by the count of landing outcomes and sorted using desc

Section 3

Launch Sites Proximities Analysis

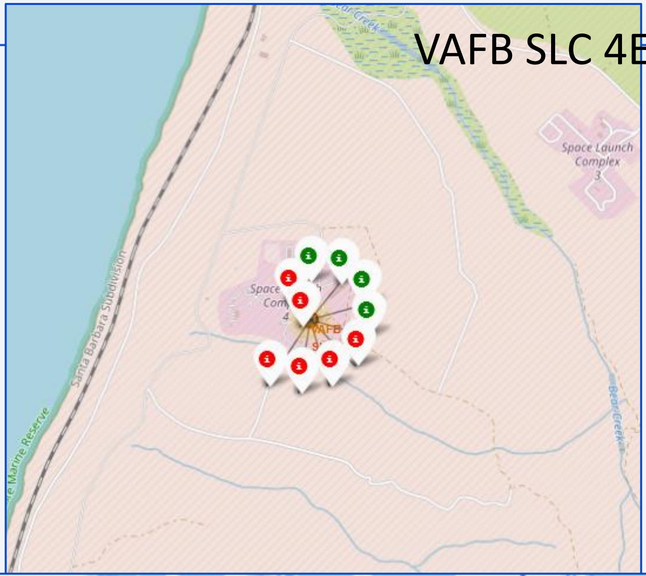
Global Launch Sites of SpaceX

Out[9]:

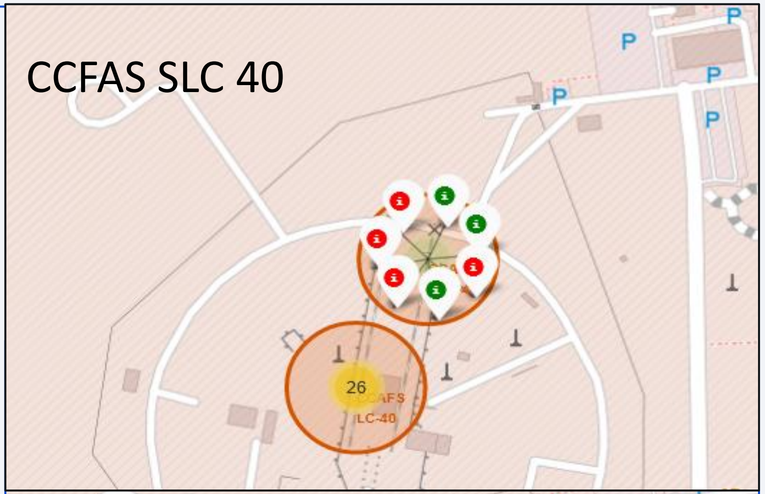


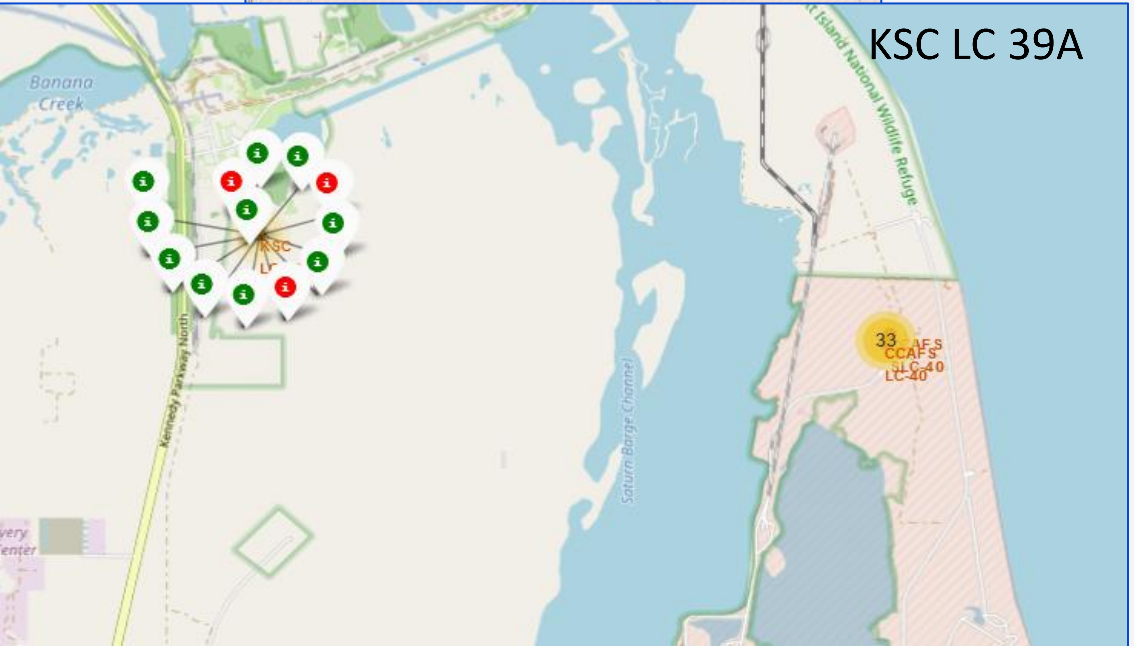
The launch sites of SpaceX are on the west and east coasts of the United States

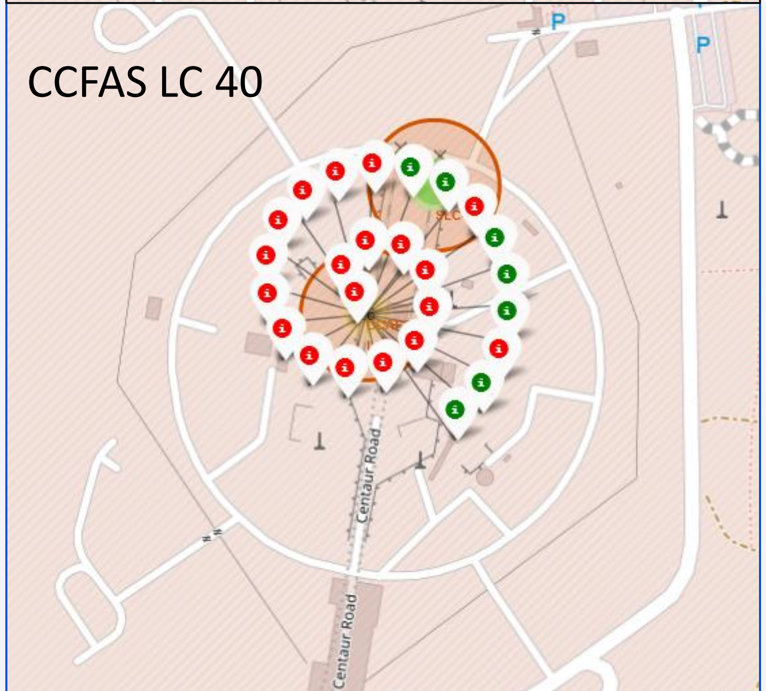
Launch Sites and Outcomes of Launches



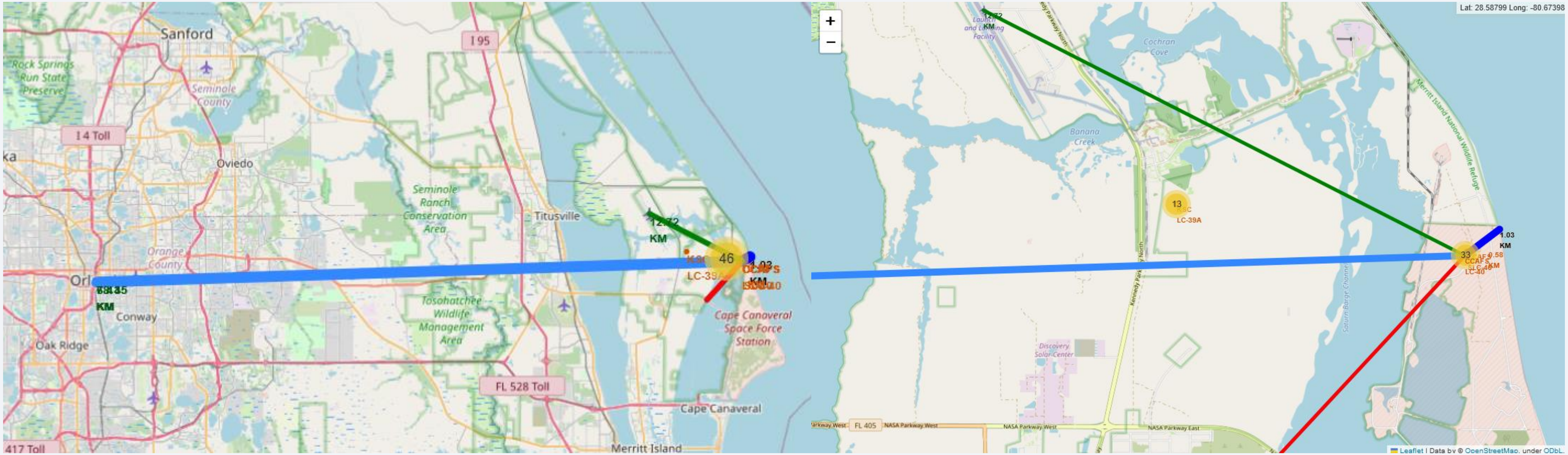
The four screenshots provide the locations of launch sites along with the outcomes of launches as markers. Red marker means a failure and green marker means success







Launch Site proximities and connectivity



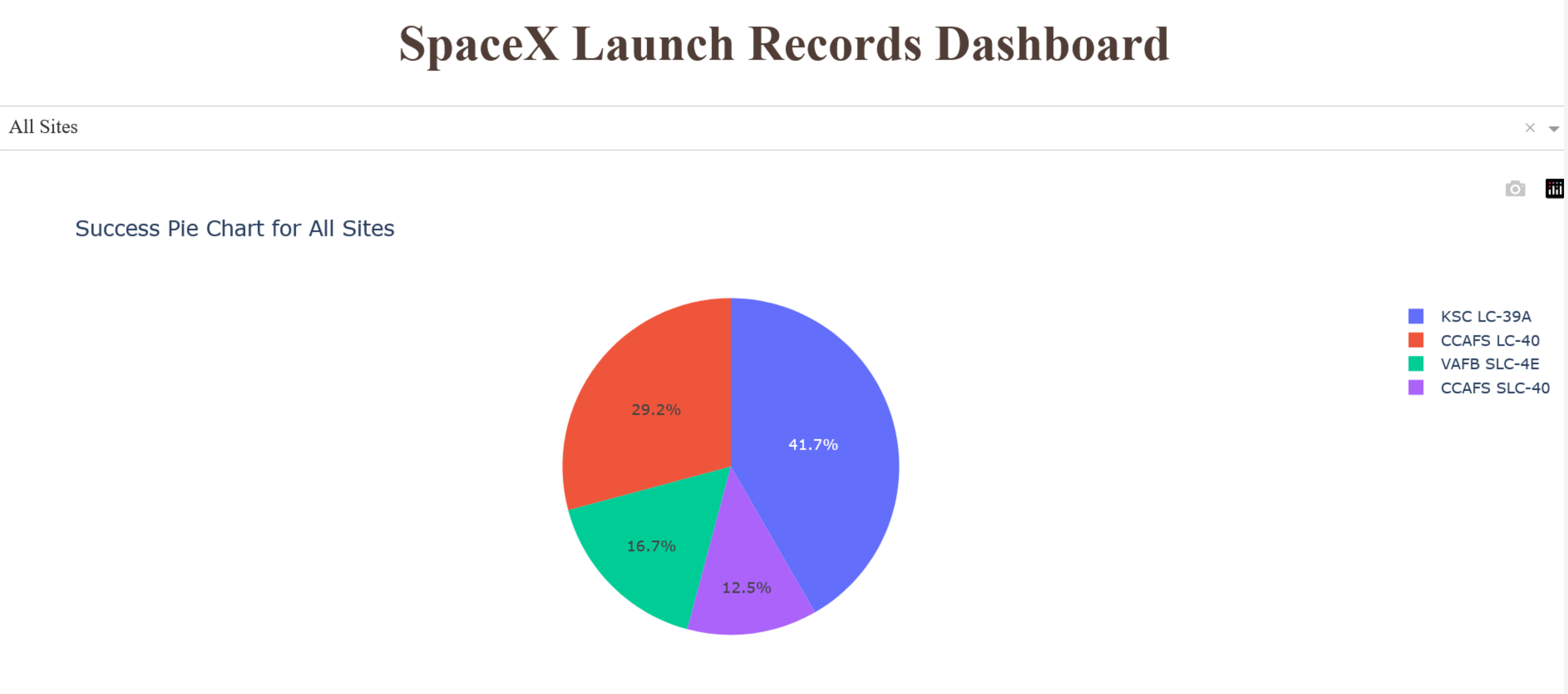
We have mapped the distance from the launch site to coast line, Nasa Parkway east highway, Launch and Training Facility airport and Orlando city. Distances and lines are drawn on the map.

Launch site is close to the coast, airport and highway but far from the city

Section 4

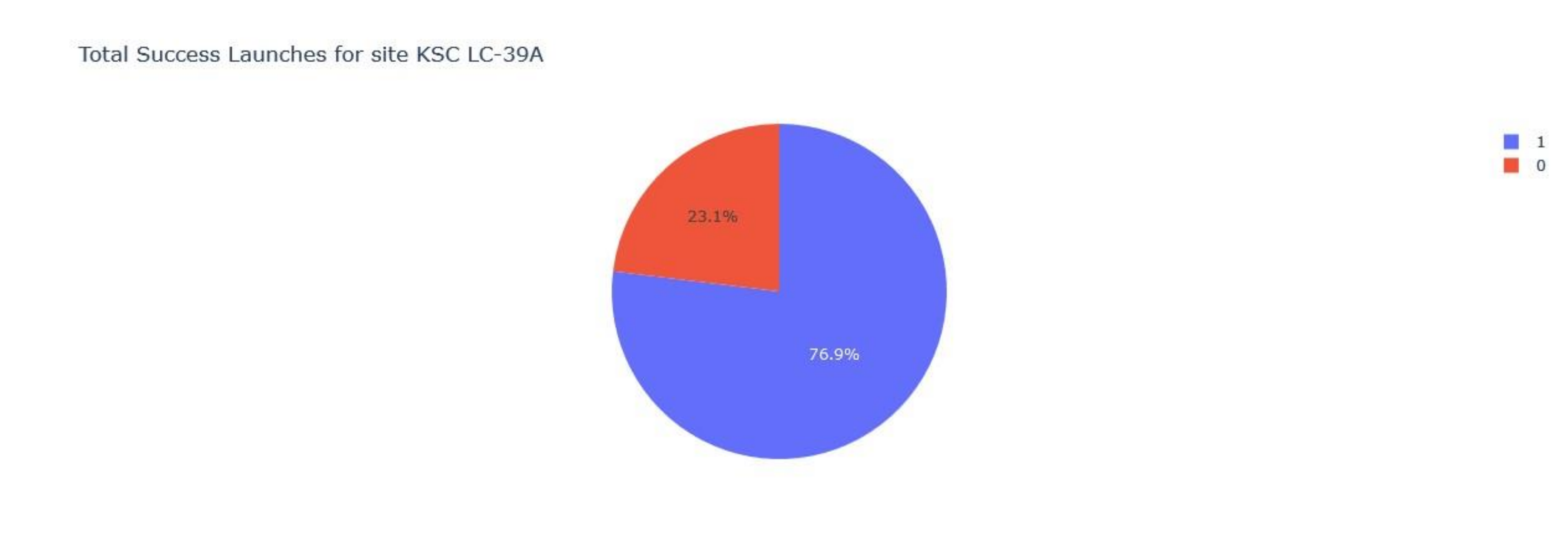
Build a Dashboard with Plotly Dash

Success Pie chart for all sites



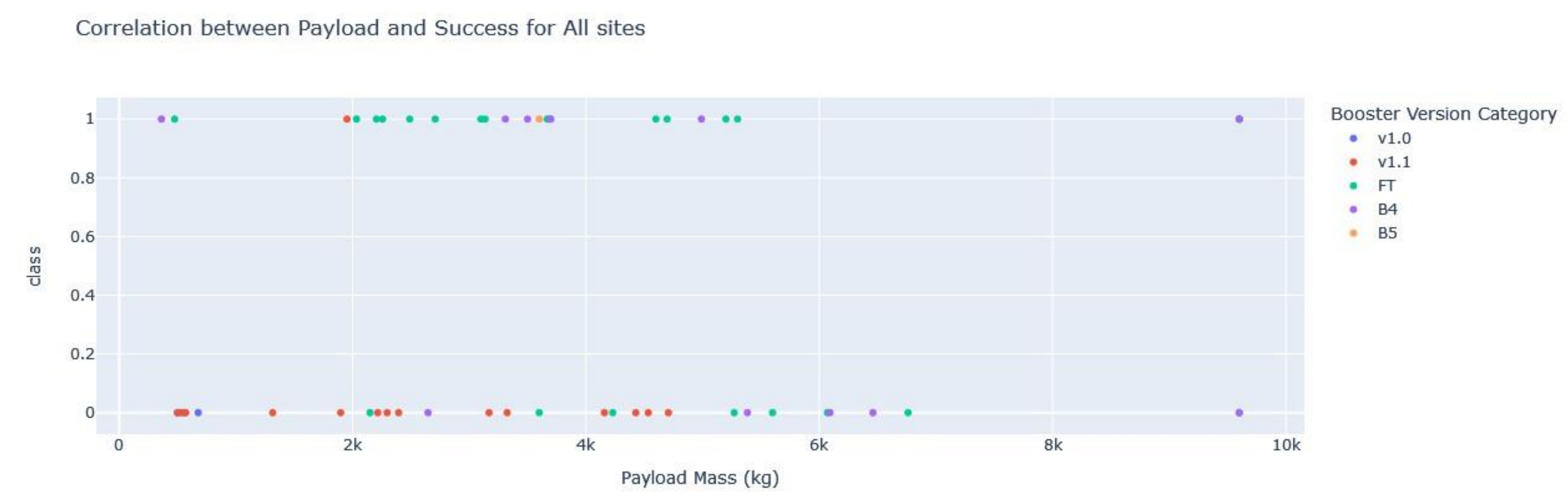
KSC-LC-39A had contributed to the most successful launches followed by CCAFS LC 40

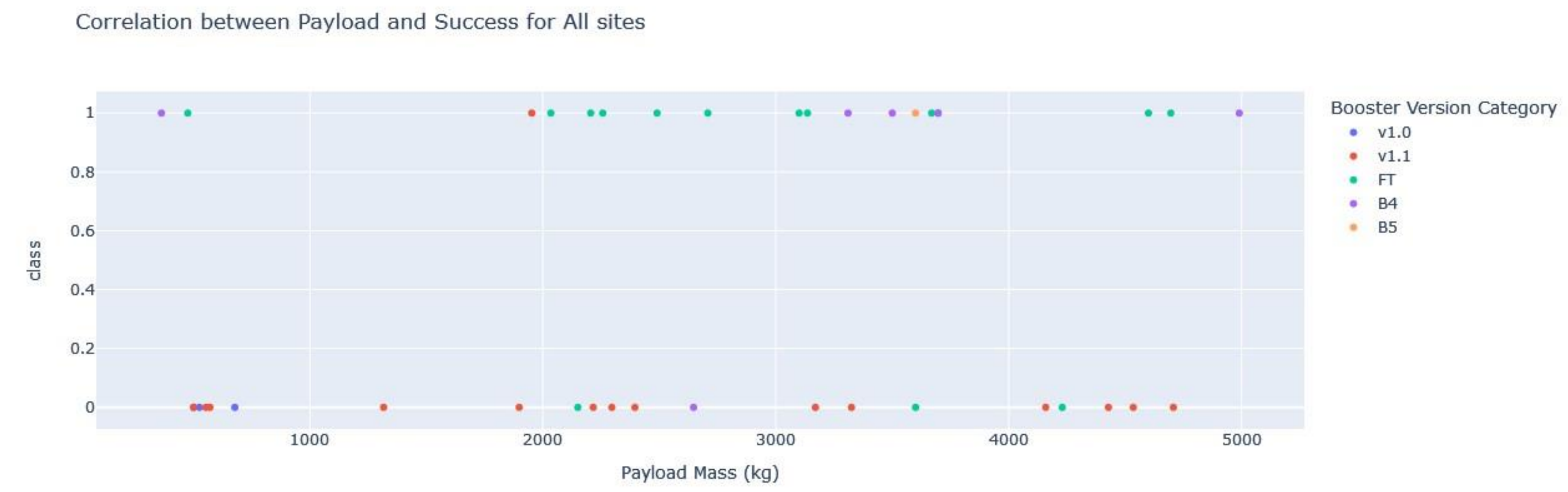
Launch Site with highest success ratio



KSC-LC-39A – 76.9% of the launches at the site were successful

Scatter Plot – Payload vs Launch Outcome – All Sites





Two graphs are presented plotting Payload Mass vs Class for all sites with each dot color-coded as per booster version.

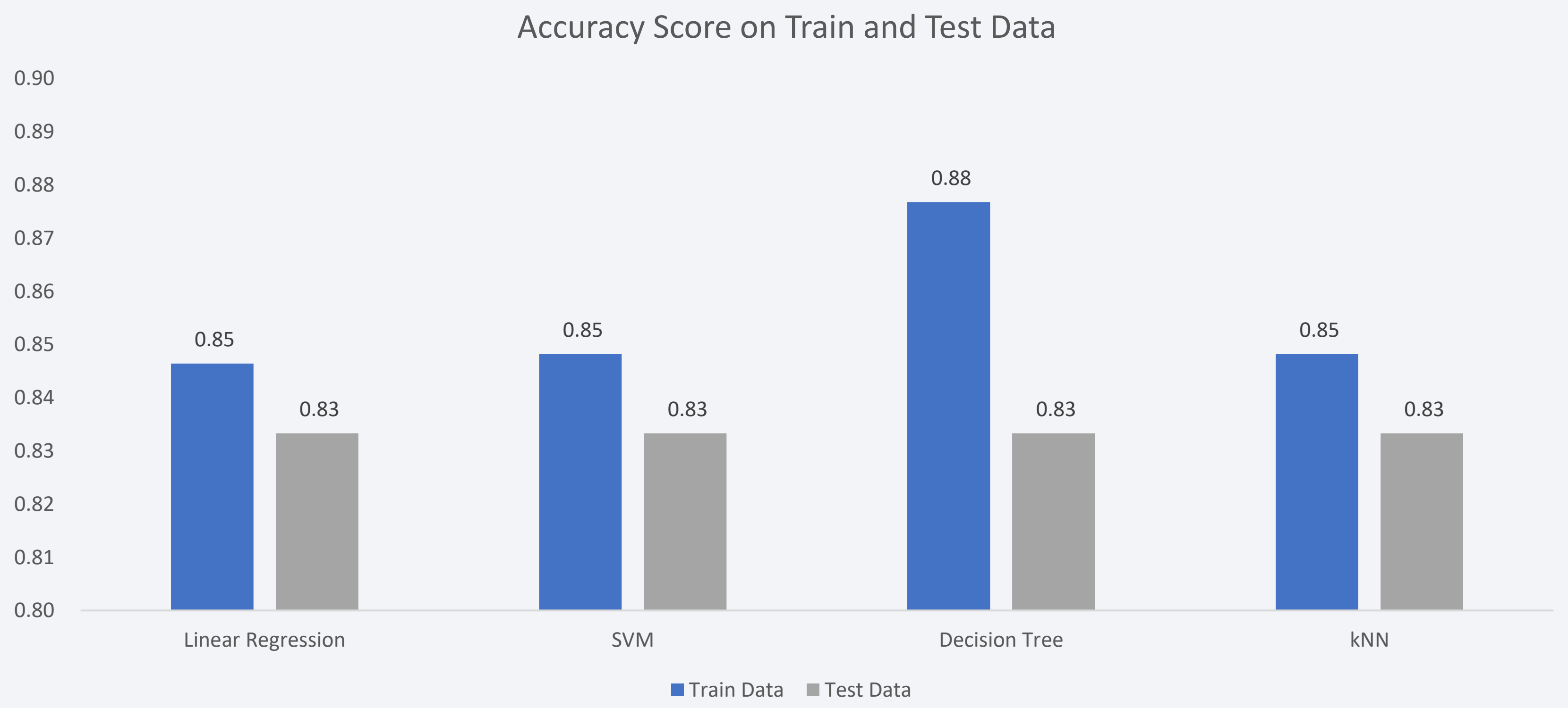
Above graph is for all payloads while below is for payloads till 5000 kg

- FT has more success rate below 5000kg payload
- Most v1.1 launches are failures irrespective of payload

Section 5

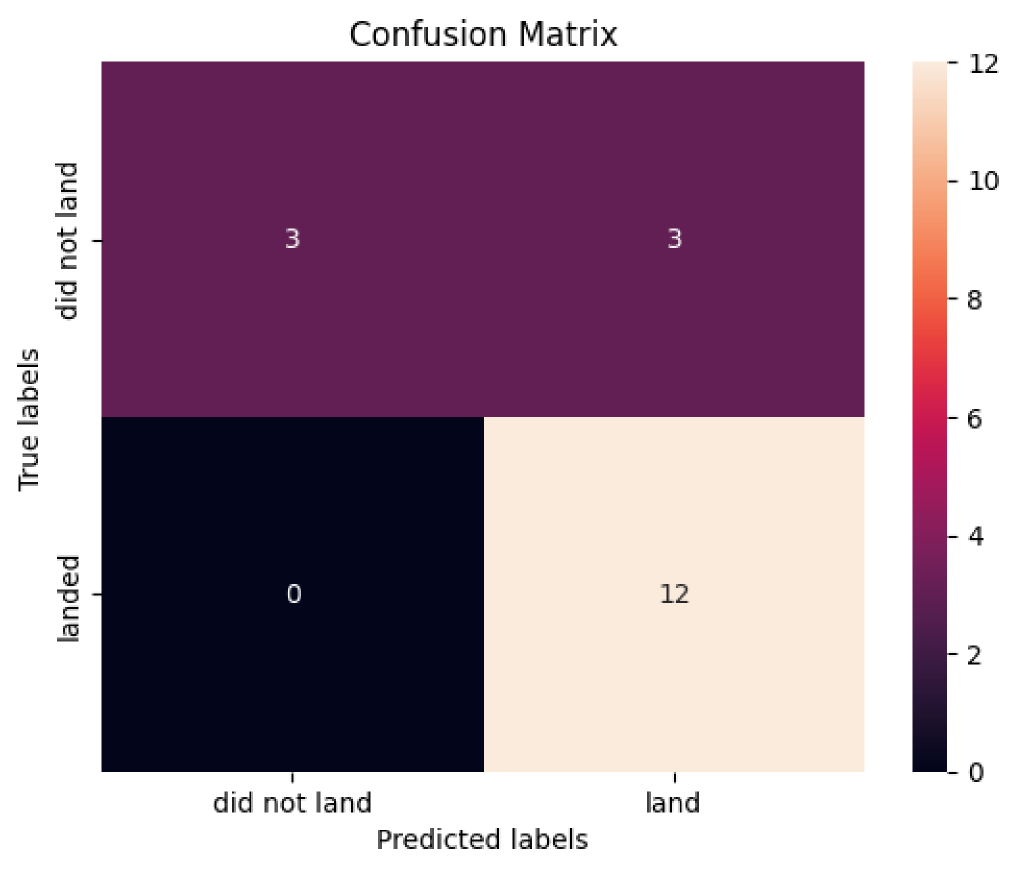
Predictive Analysis (Classification)

Classification Accuracy



Accuracy scores of the algorithms on train and test data are plotted above. It can be seen that Decision tree algorithm has the best classification accuracy

Confusion Matrix



The confusion matrix for the decision tree classifier shows that the classifier can distinguish between the different classes.

The model has identified 15 outcomes accurately.

There are no false negatives i.e., model did not predict unsuccessful landing when there is successful landing

The major problem is the false positives .i.e., unsuccessful landing marked as successful landing by the classifier.

Conclusions

- The larger the flight number, the greater the success rate.
- Launch success rate started to increase in 2013 till 2020.
- Orbits ES-L1, GEO, HEO, SSO, VLEO had the most success rate. Payloads over 10000 KG are not launched at VAFB SLC 4E
- KSC LC-39A had the most successful launches of any sites with 77% success rate.
- Launch Sites are on the coastal areas, close to transport like highways but far from cities.
- The Decision tree classifier is the best machine learning algorithm for this task.

Appendix – Code Snippets – Data Collection API

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

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

Check the content of the response

In [8]:
```python
print(response.content)
```

b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://image
s2.imgbox.com/94/f2/NN6Ph45r_o.png","large":"https://images2.imgbox.com/5b/02/QcxHUb5V_o.png"},"reddit":{"campaign":null,"laun

Used responses.get to receive data from the SpaceX API

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-spacex-data-collection-api.ipynb

In [11]:
```python
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
# data.head()
```

Out[11]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ship
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin	Engine failure at 33 seconds	[]	[

Used json_normalize to convert into datframe

In [32]:
```python
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion']!='Falcon 1']
data_falcon9['BoosterVersion'].value_counts()
# data_falcon9['FlightNumber'].value_counts()
```

Out[32]:
```
FlightNumber
6      1
83     1
```

Remove Falcon 1 Launches from the data

Appendix – Code Snippets – Data Collection Scraping

In [5]:

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

Used requests.get to get data from webpage

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

In [6]:

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

Created a beautiful soup object

In [8]:

```
# 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.find_all('table')
```

Used find_all to get the tables

https://github.com/istupe/DS_Capstone/blob/main/jupyter-labs-webscraping.ipynb

In [34]:

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

Out[34]:

```
Booster landing
Success       80
No attempt    18
Failure       10
```

Converted dictionary to dataframe

Appendix – Code Snippets – Data Wrangling

Identify and calculate the percentage of the missing values in each attribute

In [3]:
```
df.isnull().sum()/df.count()*100
```

Calculating missing values

In [5]:
```
# Apply value_counts() on column LaunchSite
df['LaunchSite'].value_counts()
```

Out[5]:
```
LaunchSite
CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
```

Launches against launch sites

In [6]:
```
# Apply value_counts on Orbit column
df['Orbit'].value_counts()
```

Out[6]:
```
Orbit
GTO     27
ISS     21
VLEO    14
PO       9
```

Launches in orbits

In [19]:
```
df['Class']=landing_class
df[['Class']].head(8)
```

Out[19]:

Class

Creating a column Class to identify whether launch is success or failure

Thank you!