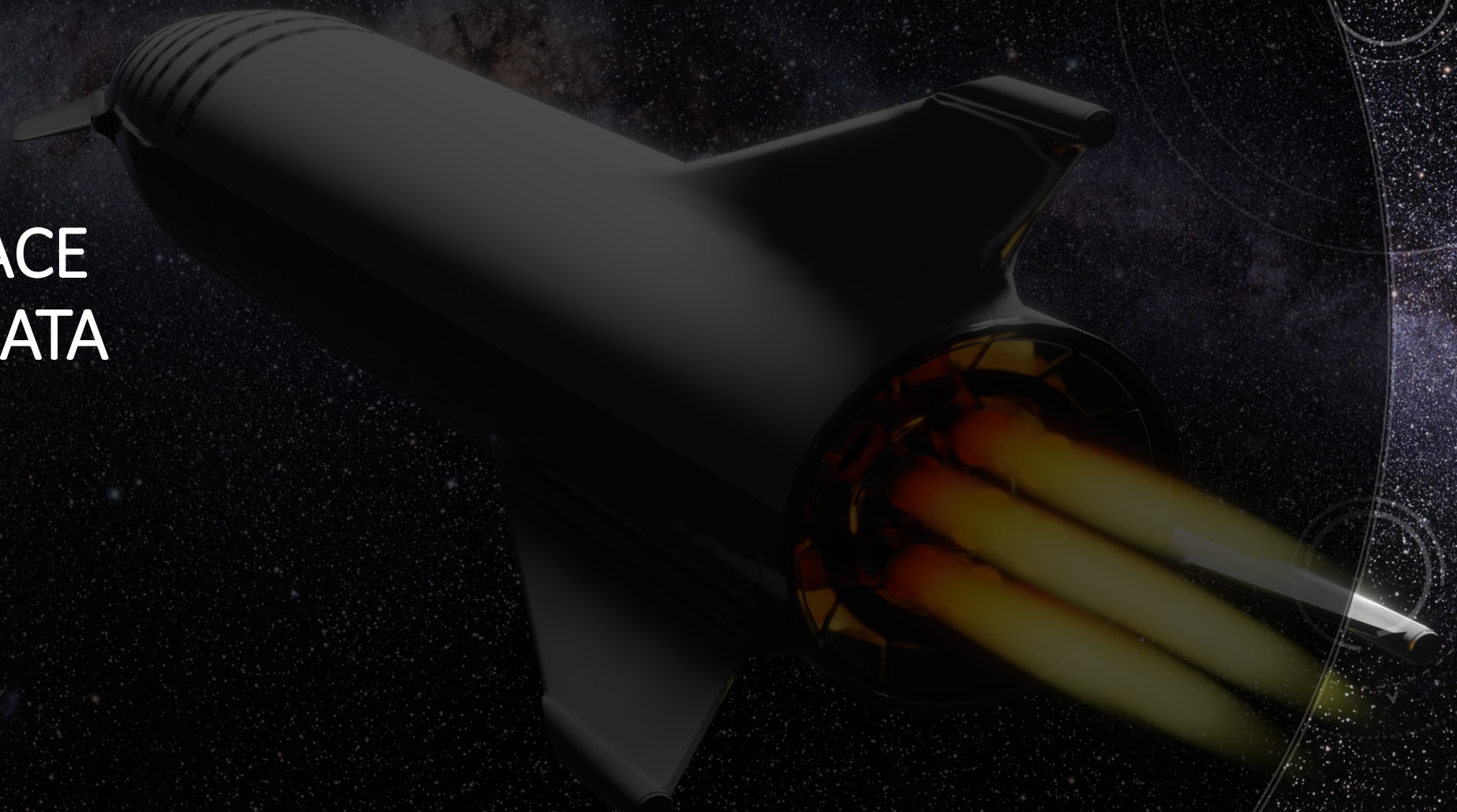
WINNING SPACE RACE WITH DATA SCIENCE



- Manav
- 19/03/2023

OUTLINE

Executive Summary

Introduction

Methodology

Results

Conclusion

Appendix

EXECUTIVE SUMMARY

Summary of methodologies

Results Summary

SUMMARY OF METHODOLOGIES

Steps taken are :-

Data Collection	Data was extracted from the SpaceX Wikipedia page and public SpaceX API.
Data Wrangling	The column 'Class' was created to categorize true successful landings. All categorical variables were converted to binary using one hot encoding, and all numeric columns were converted to float64 data type.
Exploratory Data Analysis	SQL, data visualization, folium maps, and plotly dashboards were used to explore the data. Important columns for use as features for predictive analysis were compiled.
Interactive Visual Analytics	Interactive visual analytics were used to further explore the data.
Predictive Analysis (Classification)	Standardized data and used GridSearchCV to find the best parameters for machine learning models. Displayed the accuracy rating for each model and confusion matrix for best performing models.

RESULTS SUMMARY

- This data science project involved exploratory data analysis (EDA), geospatial analytics, interactive dashboard creation, and predictive analysis using classification models. Four machine learning models were used, including Logistic Regression, Support Vector Machine, Decision Tree Classifier, and K Nearest Neighbors. All four models had similar results, with an accuracy rate of about 83.33%. However, all models overpredicted successful landings. However, more data is needed for better model determination and accuracy since the dataset used only had 90 records.
- Overall, this project provided valuable insights into the SpaceX dataset, and the visualizations and predictive models generated can be useful for future analysis and decision-making. However, the limitations of the data should be considered when interpreting the results.

Introduction

- The commercial space age has brought about A new era in space travel, with companies making it more affordable and accessible for everyone. Spacex has emerged as A leader in this industry, offering the most inexpensive launches at $62 million compared to the industry standard of $165 million. One of the key reasons for this cost advantage is the ability to recover and reuse the first stage of the rocket, known as stage one. This recovery process significantly reduces the cost of rocket launches.
- Space Y, A new player in the commercial space industry, aims to compete with spacex. One potential strategy for achieving this goal is to make accurate predictions about whether the first stage of the rocket will land successfully. By doing so, the cost of the launch can be determined, and this information can be used to assess whether an alternate company should bid against spacex for A rocket launch.

- •This project will ultimately predict if the space X falcon 9 first stage will land successfully.

ZLSADESIGN

METHODOLOGY

METHODOLOGY

- Data collection methodology:
  - GET requests to the SpaceX REST API.
  - Web Scraping Space X Wikipedia page.
- Perform data wrangling
  - Describe how data was processed
  - Find missing values and replace with mean.
  - Used .value_counts() to find frequencies in columns
  - Classified landings as successful (class=1) and unsuccessful (class=0)
- 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, evaluate classification models

Data Collection

1. Data Collection involved many Api requests from SpaceX Public Api.
   - Using the SpaceX API to retrieve data about launches, including information about the rocket used, payload delivered, launch specifications, landing specifications, and landing outcome.
2. Web Scraping using BeautifulSoup library from the SpaceX Wikipedia page.
   - Web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled List of Falcon 9 and Falcon Heavy launches.

Data Collection – SpaceX API

- We have maded many Api requests to the SpaceX Public Api.

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

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

Make a get response to the Space X Api

Convert the response to a .json file and then to a pandas Dataframe.

Use custom logic to clean the data and define lists to store data.

Call the custom functions and fill data in lists and use list as value in dictionary.

Make the DataFrame from the dictionary and filter the DataFrame to only Falcon 9 launches.

Reset the FlightNumber column and find the no. of missing values per column and replace with mean of column.

```
#Global variables
BoosterVersion = []
PayloadMass = []
Orbit = []
LaunchSite = []
Outcome = []
Flights = []
GridFins = []
Reused = []
Legs = []
LandingPad = []
Block = []
ReusedCount = []
Serial = []
Longitude = []
Latitude = []
```

```
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)
```

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = launch_df[launch_df['BoosterVersion'] == 'Falcon 9']
```

Github link

Data Collection - Scraping

- We've used BeautifulSoup library for Scraping the data from the SpaceX Wikipedia page.

Make a get request to the Space X Wikipedia page of Falcon 9 and store response in a object.

Create a BeautifulSoup object from the HTML response and Find all tables within the HTML page

Extract the columns names from the table and store in a list.

Make dictionary and use column names list as key and use the functions to values in the dictionary.

Convert the dictionary to DataFrame

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

```python
soup = BeautifulSoup(response.content, 'html.parser')
html_tables = soup.find_all('table')
```

```python
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)
```

```python
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']=[]
```

Github link

DATA WRANGLING

- In the Dataset of SpaceX there are several launch facilities and each launch is dedicated to orbit, different orbit are in Orbit column of df.
- Here we found count & occurrence of the several Launch Sites, Orbits & Landing Outcome.

calculate the percentage of the missing values in each attribute.

Calculated the number of launches on each site.

Calculated the number and occurence of each orbit and mission outcome per orbit type.

Found success rate by finding mean of class column.

Create a class column in df from Outcome column and use the set to classify the successful(1) and failed (0) outcomes.

Make a set of bad outcomes by taking keys and observing them and selcting bad ones.

```
df.isnull().sum()/df.count()*100
```

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

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

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

```
for i,outcome in enumerate(landing_outcomes.keys()):
    print(i,outcome)
bad_outcomes=set(landing_outcomes.keys()[[1,3,5,6,7]])
```

```
# landing_class = 0 if bad_outcome
# landing_class = 1 otherwise
landing_class = []

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

```
df["Class"].mean()
```

```
0.6666666666666666
```

Github link

ZLSADESIGN

EDA with Data Visualization

Cat Chart

A cat plot, also known as a categorical plot, is a type of data visualization that displays the relationship between a categorical variable and a numerical variable.

A Cat chart was produced to visualize the relationships between:

- `PayloadMass` and Flight Number

Scatter Chart

A scatter plot is a type of data visualization that displays the relationship between two numerical variables.

Here Scatter charts were produced to visualize the relationships between:

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

Bar Chart

Bar charts are useful for visualizing categorical data, where the data is divided into distinct categories, and for comparing the values of different categories or variables.

A bar chart was produced to visualize the relationship between:

- Success Rate and Orbit Type

Line Chart

Line charts contain numerical values on both axes, and are generally used to show the change of a variable over time.

A Line charts was produced to visualize the relationships between:

- •Success Rate and Year (i.e. the launch success yearly trend)

Github link

EDA with SQL

To gather some information about the dataset, some SQL queries were performed.

SQL queries performed are for :-

- 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 the average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome on a ground pad was achieved.
- List the names of the boosters which had success on a drone ship and a payload mass between 4000 and 6000 kg
- List the total number of successful and failed mission outcomes.
- List the names of the booster versions which have carried the maximum payload mass.
- List the failed landing outcomes on drone ships, their booster versions, and launch site names for 2015.
- 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.

Github link

INTERACTIVE MAP - FOLIUM

•Steps taken to visualize the launch data of Space X on a interactive map:

- To mark all launch sites on a map
  - First initialized map using Folium map object.
  - Add Folium circle and marker for each launch site
- To mark the success/failed launches for each site on the map
  - Initialized cluster object to cluster the launches as there are many launches in each site .
  - At first we have assigned marker color as successful (class = 1) as green, and failed (class = 0) as red.
  - Added the launches to cluster and added marker to each launch.
- To calculate the distances between a launch site to its proximities
  - Added a marker to show distance from the site to the point.
  - Then added a line from site to point using Polyline by Folium.

Github link

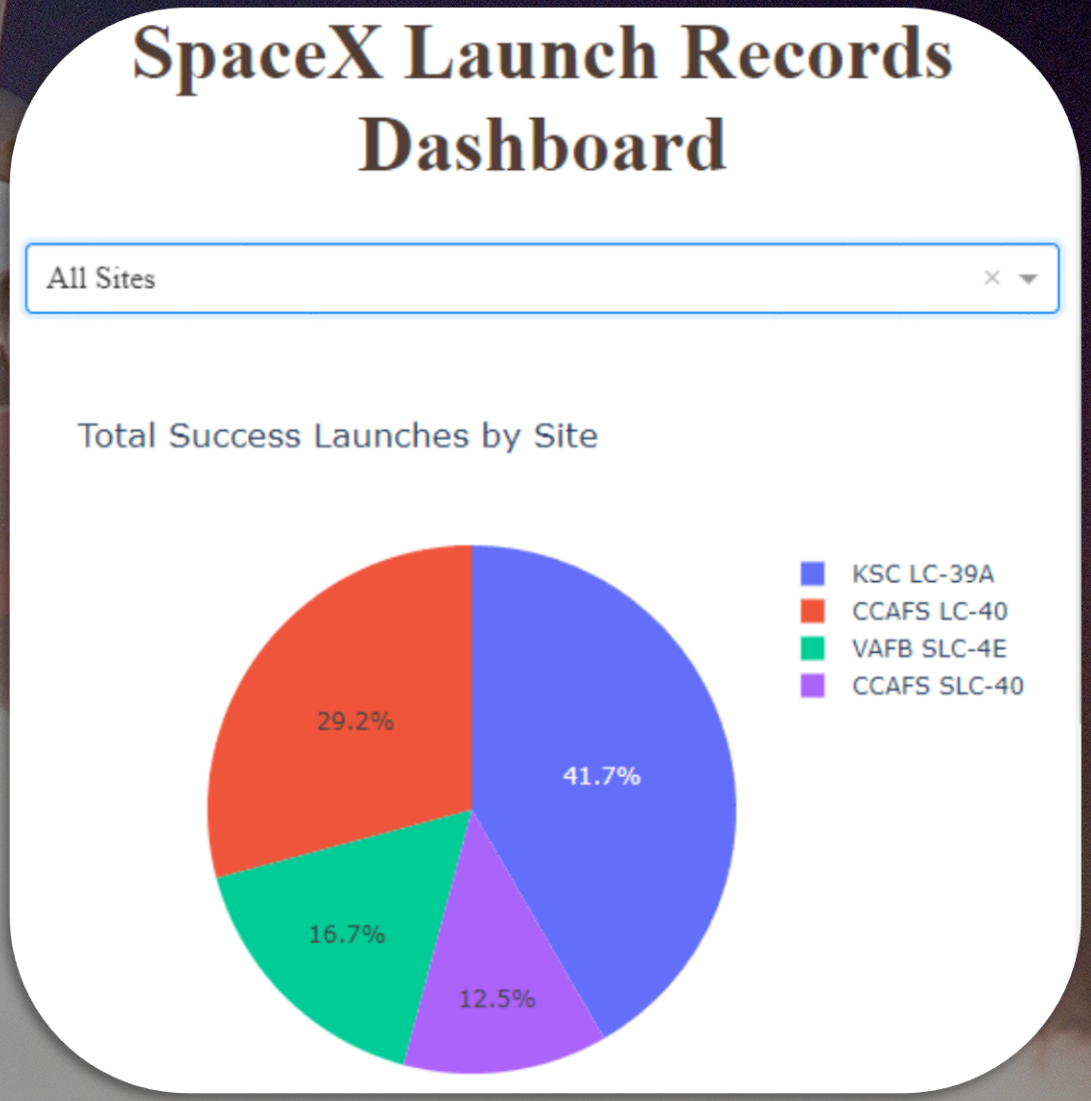
Dashboard with Plotly Dash

Plots used in Plotly Dash Dashboard :-

- Pie Chart
  - Showing the percentage of successful launches from each site.
  - We can select site from the dropdown menu and see percentage of successful launches.
- Scatter plot
  - To show correlation between outcome (success or not) and payload mass (kg)
  - We can select range of payload also from range slider.

Interaction used in Plotly Dash Dashboard :-

- Dropdown
  - Can select a site from several sites with the help of dropdown menu and see percentage of successful launches of that site.
- Range slider
  - We can select range of payload from range slider.



Github link

Predictive Analysis (Classification)

The following steps were taking to develop, evaluate, and find the best performing classification model:

Model Development

- Prepare the Dataset for development
  - Load the Dataset
  - Pre-process and standardize the dataset.
  - Split dataset for training and testing.
  - Decide the most appropriate models.
- For each model
  - Create a GridSearchCV object and a dictionary of parameters.
  - Fit the model.
  - Use training dataset to train the model.

Model Evaluation

- For each model
  - Use output GridSearchCV object calculate
    - Tuned hyperparamerts (best parameters)
    - Accuracy score of parameters.
  - Calculate the accuracy score of the model.
  - Plot the confusion matrix of the model.

Best Model

- Compare accuracy score of models
- Model with highest accuracy will be considered as best model.

Github link

RESULTS

Exploratory data analysis

Interactive analytics

Predictive analysis

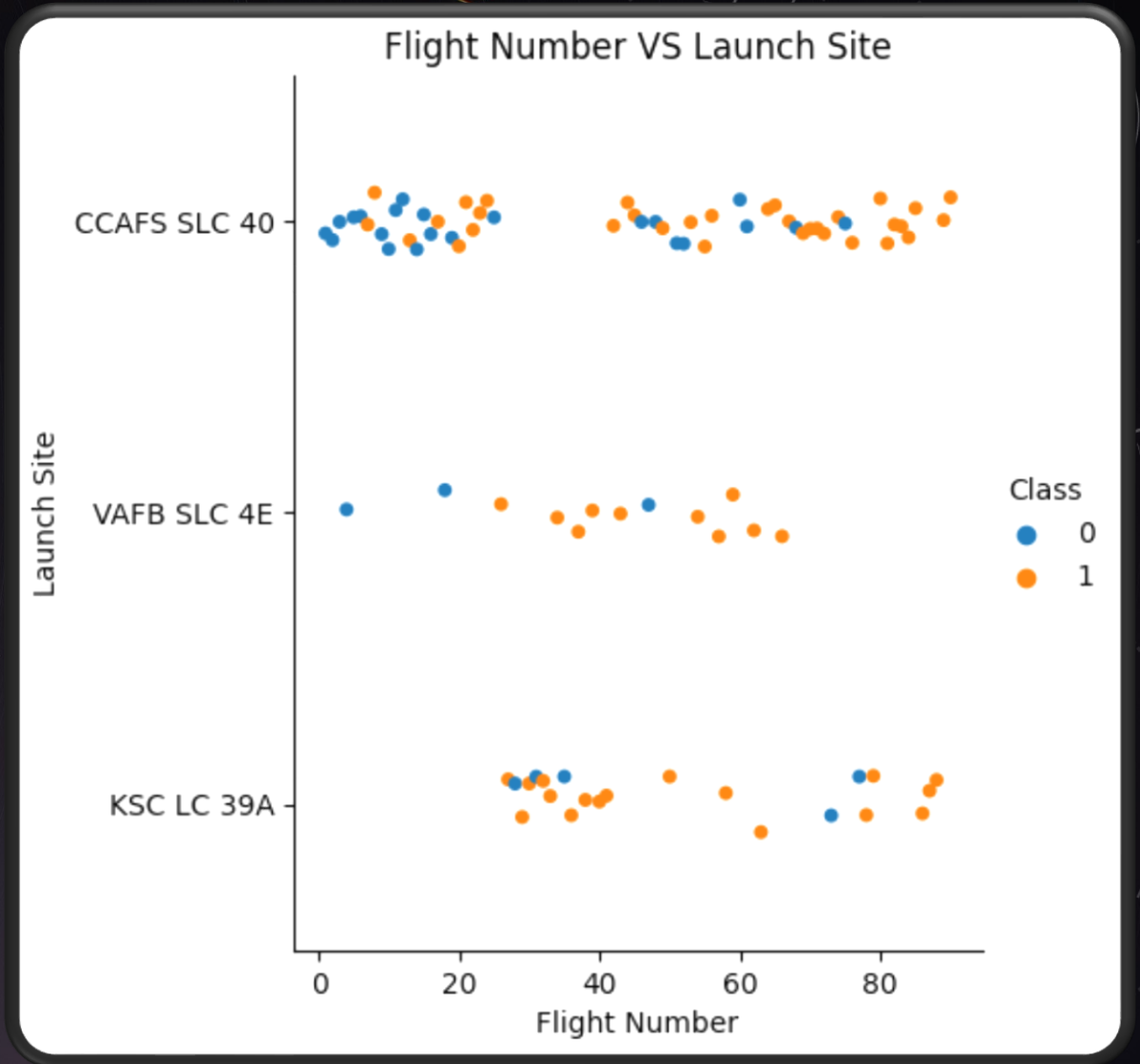
ZLSADESIGN

EDA - With Visualization

FLIGHT NUMBER VS. LAUNCH SITE

The Scatter Plot of Flight Number Vs Launch Site shows that :-

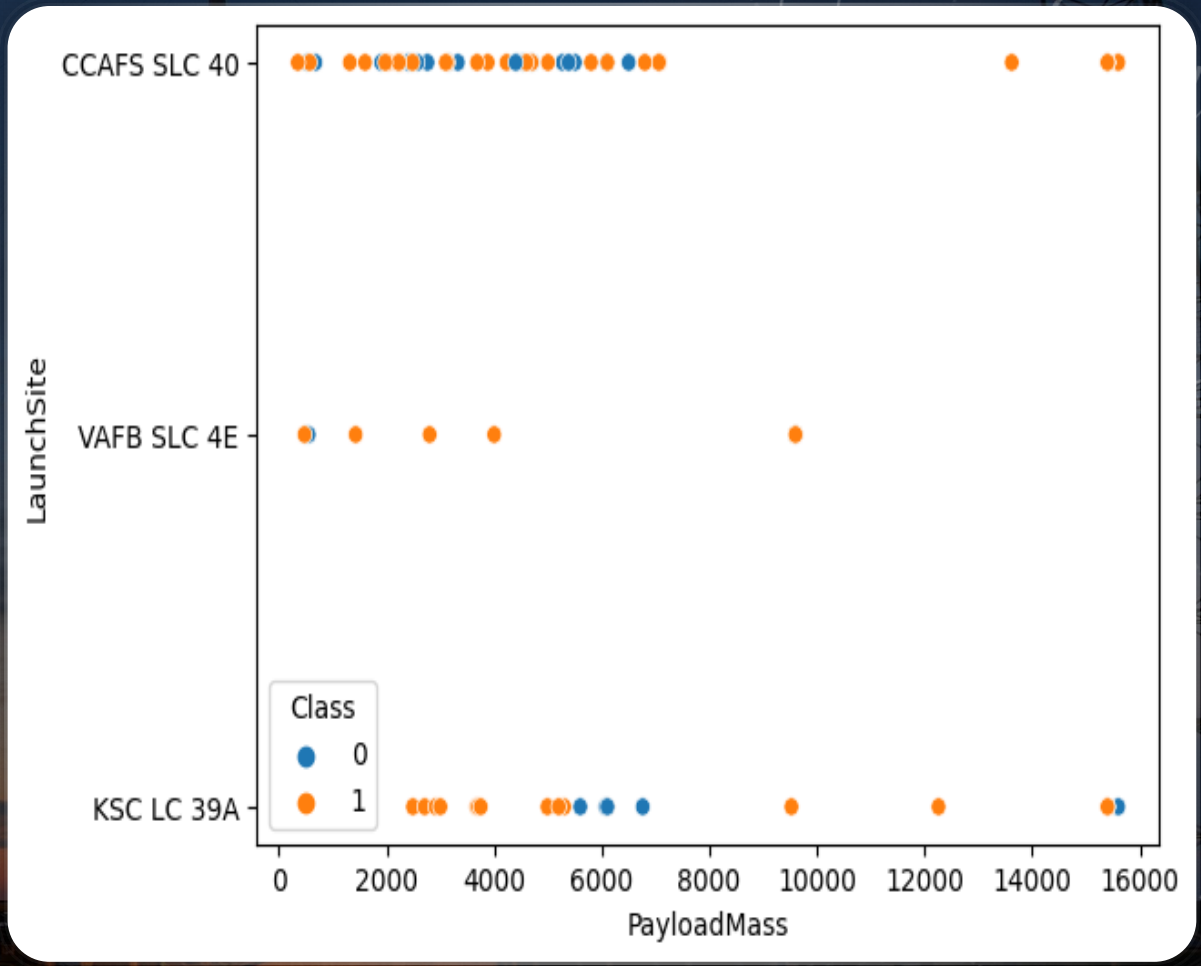
- Success rate increases as the flight number increases.
- We can see CCAFS SLC 40 launch site have more flights than any other launch site.
- All early flights (Flight number < 30) are launched in CCAFS SLC 40 site and are more unsuccessful.
- VAFB SLC 4E site has the lowest launches, but here success ratio is good.
- There is no early flights in KSC LC 39A site and Success ratio of this site is better than others.
- By this plot we can Analyse that early flights are more unsuccessful and as flight number increse success ratio increases.
- After Flight Number = 30 there are more successful flights



PAYLOAD MASS VS. LAUNCH SITE

The Scatter Plot of Payload Mass Vs Launch Site shows that :-

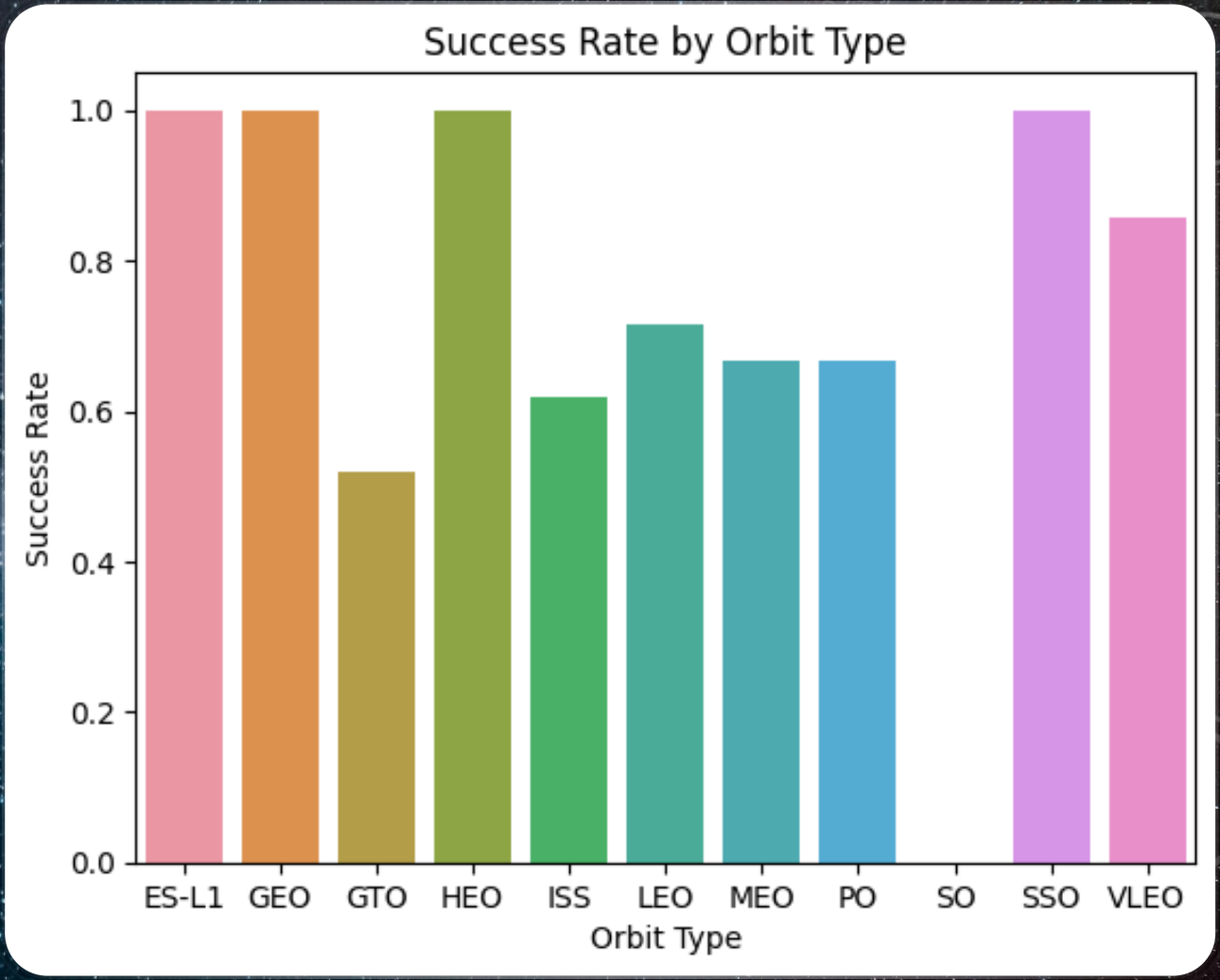
- As we can see above Payload Mass of around 7000 kg there are more successful launches but we have very less data of heavier Payload.
- CCAFS SLC 40 has more lighter launches where Payload Mass is less than 8000 Kg.
- By this plot it seems there is no correlation between the Payload Mass and Lauch site.
- VAFB SLC 4E has more successful launches but have very few launches.



SUCCESS RATE VS. ORBIT TYPE

The Bar chart of Success Rate Vs Orbit Type shows that :

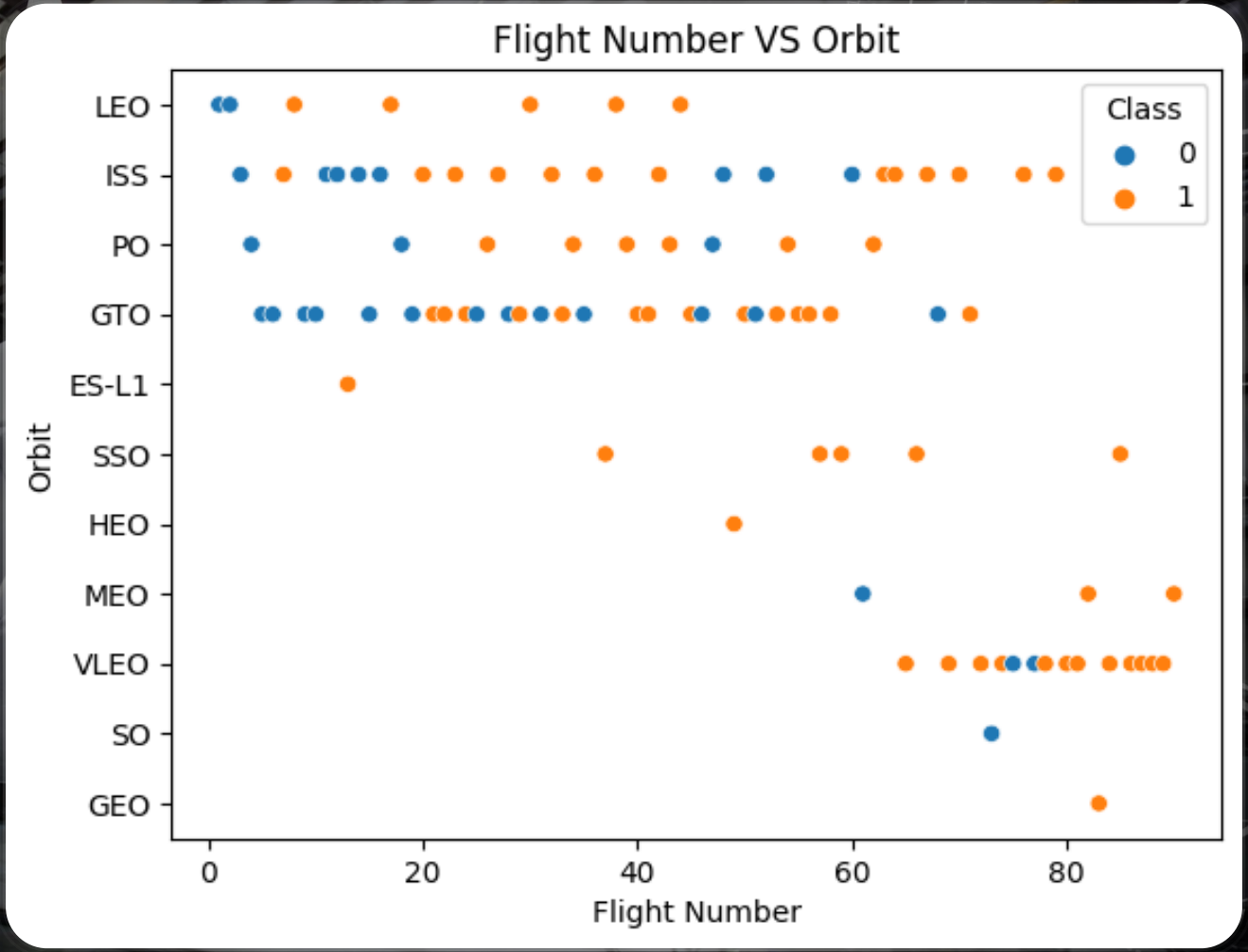
- The following orbits have the highest (100%) success rate:
  - ES-L1 (Earth-Sun First Lagrangian Point)
  - GEO (Geostationary Orbit)
  - HEO (High Earth Orbit)
  - SSO (Sun-synchronous Orbit)
- The orbit with the lowest (0%) success rate is:
  - SO (Heliocentric Orbit)



FLIGHT NUMBER VS. ORBIT TYPE

This scatter plot of Orbit Type vs. Flight number shows some more useful insights :

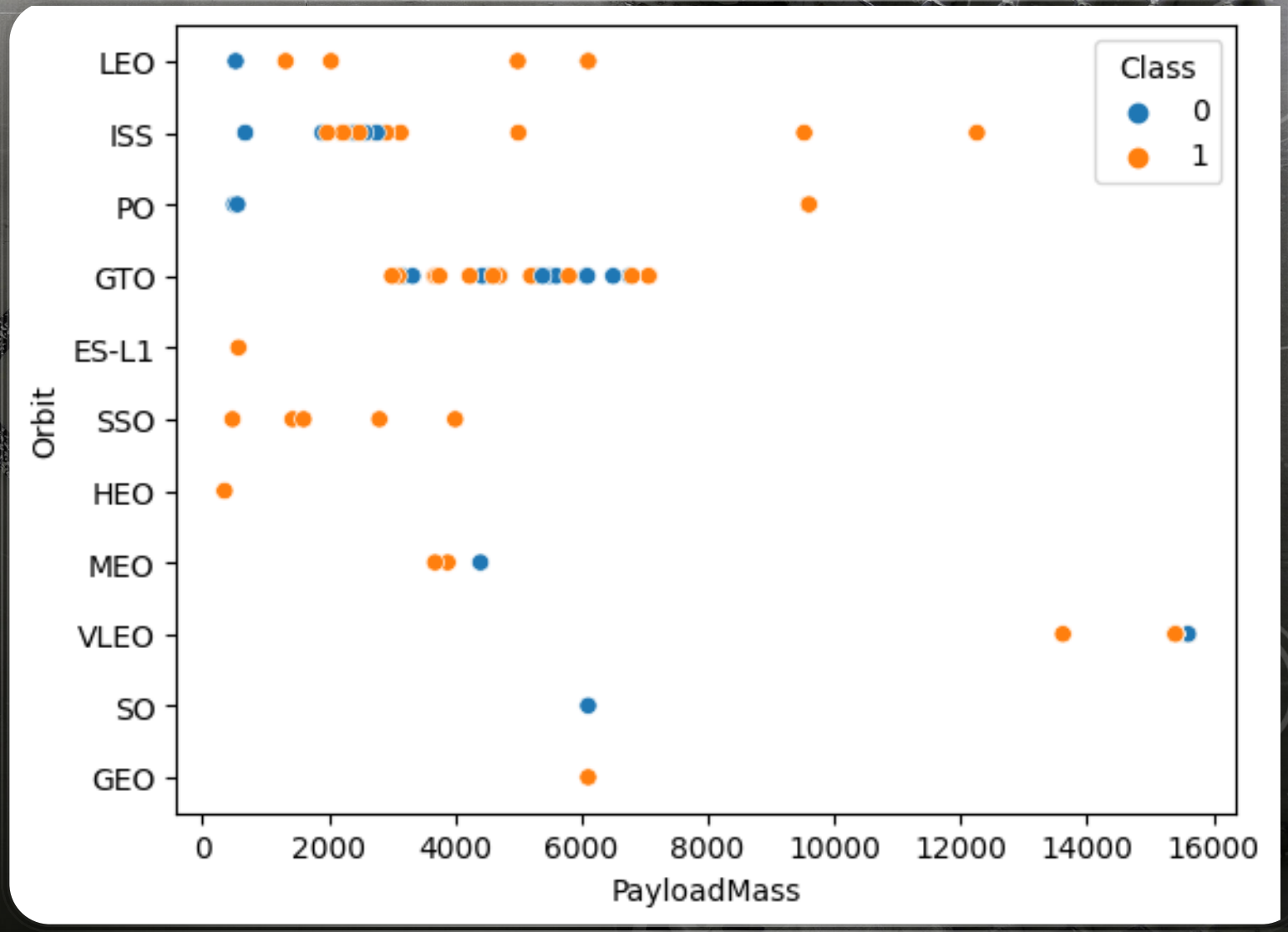
- As the Flight Number increases the number of orbit also increased i.e. launches are done for other orbits also.
- And also success rate increases with flight number.
- GEO, HEO & ES-L1 have only one launch so this explains the 100% success rate of them.
- SSO orbit type is only one which have 100% success.
- GTO orbit type have little relation to the Flight Number.



PAYLOAD VS. ORBIT TYPE

This scatter plot of Orbit Type vs. Payload Mass shows that:

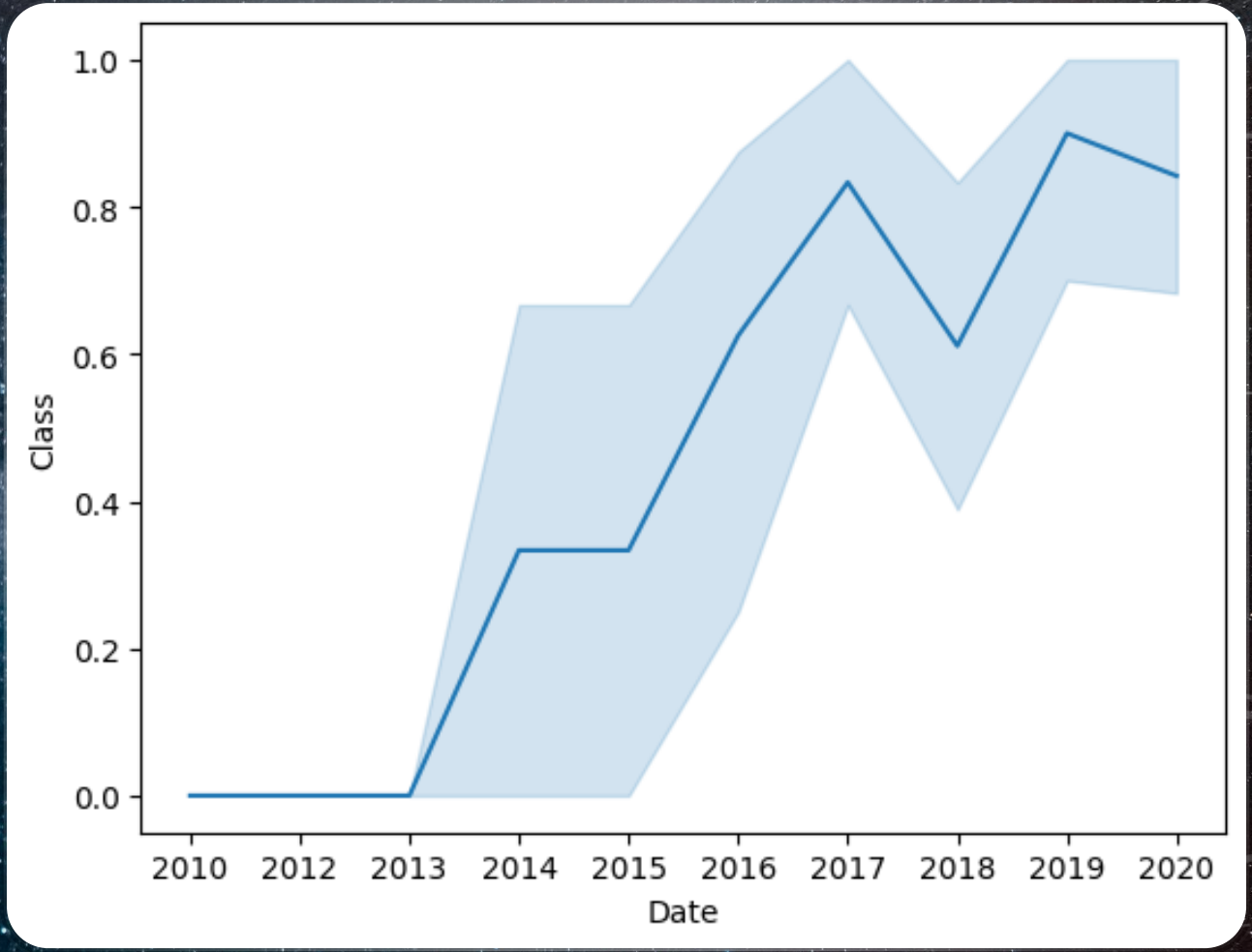
- Here we get to know that SSO orbit have success with lighter Payload Mass than 6000 Kg.
- Orbit types with heavier Payload Mass and high success ratio are :-
  - LEO
  - ISS
  - PO
  - VLEO
- Here also GTO orbit don't have relationship with Payload Mass



LAUNCH SUCCESS YEARLY TREND

The line chart of yearly average success rate shows that:

- Year by Year we have improved and more successful landings are achived.
- Before 2013 all the landings were unsuccessful.
- After 2013 the success rate increased.
- After 2015 we have seen success ratio to increase more.
- At 2018 we can see there is a dip in successful landings but then again we recovered from that.



EDA - With SQL

All Launch Site Names

The word 'distinct' will retrieve unique values from the column 'Launch_Site' from the table 'SPACEXTBL'.

```
stmt = "Select distinct Launch_Site from SPACEXTBL"
cur.execute(stmt)
rows = cur.fetchall()
for row in rows:
    print(row)
```

```
('CCAFS LC-40',)
('VAFB SLC-4E',)
('KSC LC-39A',)
('CCAFS SLC-40',)
```

Launch Site Names Begin with 'CCA'

'LIKE' is used for wildcard to get values starting with 'CCA%' and then 'LIMIT 5' used to get only 5 records.

```
stmt = "
select Launch_Site from SPACEXtbl where Launch_Site like 'CCA%' limit 5"
cur.execute(stmt)
rows = cur.fetchall()
for row in rows:
    print(row)
```

```
('CCAFS LC-40',)
('CCAFS LC-40',)
('CCAFS LC-40',)
('CCAFS LC-40',)
('CCAFS LC-40',)
```

Total Payload Mass

Sum function is used to find total Payload Mass on the filtered data where Customer is NASA (CRS)

```
stmt = "
select SUM(PAYLOAD_MASS__KG_) from SPACEXTBL where Customer = 'NASA (CRS)'"
cur.execute(stmt)
cur.fetchone()
```

```
(45596,)
```

Average Payload Mass by F9 v1.1

Filtered the table SPACEXTBL with Booster version F9 v1.1 and calculated the average of the PAYLOAD_MASS__KG

```
stmt = "
SELECT AVG(PAYLOAD_MASS__KG_) from spacextbl where
 Booster_Version = 'F9 v1.1'
"
cur.execute(stmt)
cur.fetchone()
```

```
(2928.4,)
```

First Successful Ground Landing Date

The strftime function in SQLite expects the date to be in the ISO 8601 format, which is YYYY-MM-DD. Therefore, you need to first convert your date strings to this format using the substr function before using strftime to find the minimum date. Then we have search for the Landing outcome equal to Success (ground pad).

```
stmt = "SELECT min(strftime('%d
-%m-%Y', substr(Date,7,4) || '-' || substr(Date,4,2)
|| '-' || substr(Date,1,2))) FROM spacextbl where
  [Landing _Outcome] = 'Success (ground pad)'
"
cur.execute(stmt)
cur.fetchall()
```

```
[('01-05-2017',)]
```

Successful Drone Ship Landing with Payload between 4000 and 6000

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

```
stmt = "
select Booster_Version from spacextbl where [Landing _O
utcome] = 'Success (drone ship)' and PAYLOAD_MASS__KG_
 between 4000 AND 6000
"
cur.execute(stmt)
cur.fetchall()
```

```
[('F9 FT B1022',), ('F9 FT B1026',), ('F9 FT  B1021.2',), ('F9 FT  B1031.2',)]
```

Total Number of Successful and Failure Mission Outcomes

Here we have listed the total number of successful and failure mission outcomes using count function where
in Mission_Outcome we used Like to use wildcard to get results starting with 'Failure%' and then we grouped them all by Mission_Outcome.

```
stmt = "
select Mission_Outcome, count(*) from spacextbl where Mission_Outcome LIKE
'Success%' OR Mission_Outcome LIKE 'Failure%' group by Mission_Outcome
"
cur.execute(stmt)
cur.fetchall()
```

```
[('Failure (in flight)', 1),
 ('Success', 98),
 ('Success ', 1),
 ('Success (payload status unclear)', 1)]
```

Boosters Carried Maximum Payload

Here we have listed the names of the booster_versions which have carried the maximum payload mass. Using a subquery.

```
stmt = "
select Booster_Version from spacextbl where PAYLOAD_MASS__KG_ =
 (select MAX(PAYLOAD_MASS__KG_) from spacextbl)
"
cur.execute(stmt)
cur.fetchone()
```

```
('F9 B5 B1048.4',)
```

2015 Launch Records

Here we have listed the records which will display the month names, failure landing_outcomes in drone ship ,booster versions, launch_site for the months in year 2015.

```
stmt = "
select substr(Date, 4, 2), Booster_Version, Launch_Site from
spacextbl where [Landing _Outcome] = 'Failure (drone ship)'
 and substr(Date, 7,4) = '2015'
"
cur.execute(stmt)
cur.fetchall()
```

```
[('01', 'F9 v1.1 B1012', 'CCAFS LC-40'),
 ('04', 'F9 v1.1 B1015', 'CCAFS LC-40')]
```

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

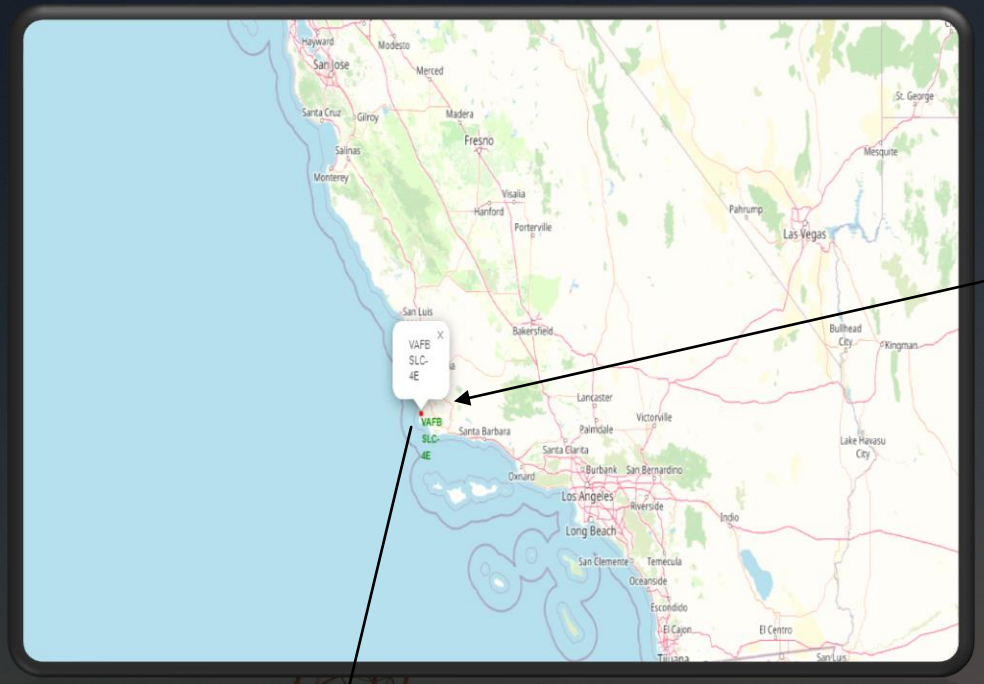
The strftime function in SQLite expects the date to be in the ISO 8601 format, which is YYYY-MM-DD. Therefore, you need to first convert your date strings to this format using the substr function before using strftime to find the date in the range 2010-06-04 and 2017-03-20. We have selected columns Landing_Outcome, Date & used count to find the count of the n_landings and then we have grouped by Landing_Outcome and ordered them by n_landing(Number of landings).

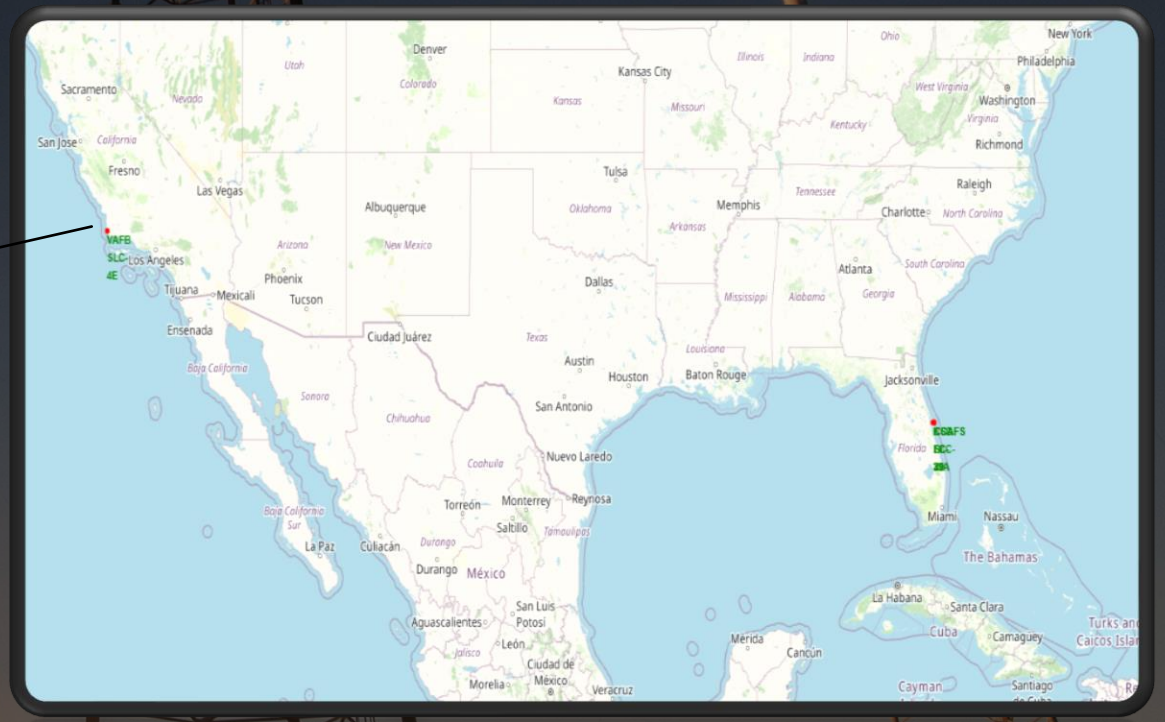
```
stmt = "
select [Landing _Outcome], Date, count(*) as n_landings from
spacextbl where [Landing _Outcome] like 'Success%' and DATE
(substr(Date,7,4) || '-' || substr(Date,4,2) || '-' || subst
r(Date,1,2)) BETWEEN '2010-06-04' AND '2017-03-20' group by
 [Landing _Outcome] order by n_landings desc
"
cur.execute(stmt)
cur.fetchall()
```

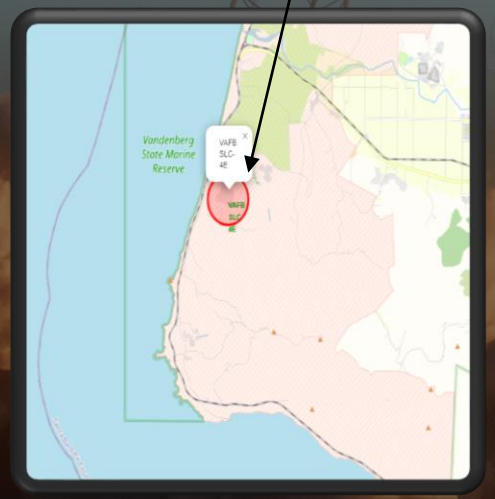
```
[('Success (drone ship)', '08-04-2016', 5),
 ('Success (ground pad)', '22-12-2015', 3)]
```

LAUNCH SITES PROXIMITY ANALYSIS – FOLIUM INTERACTIVE MAP

Launch Site – VAFB SLC 4E

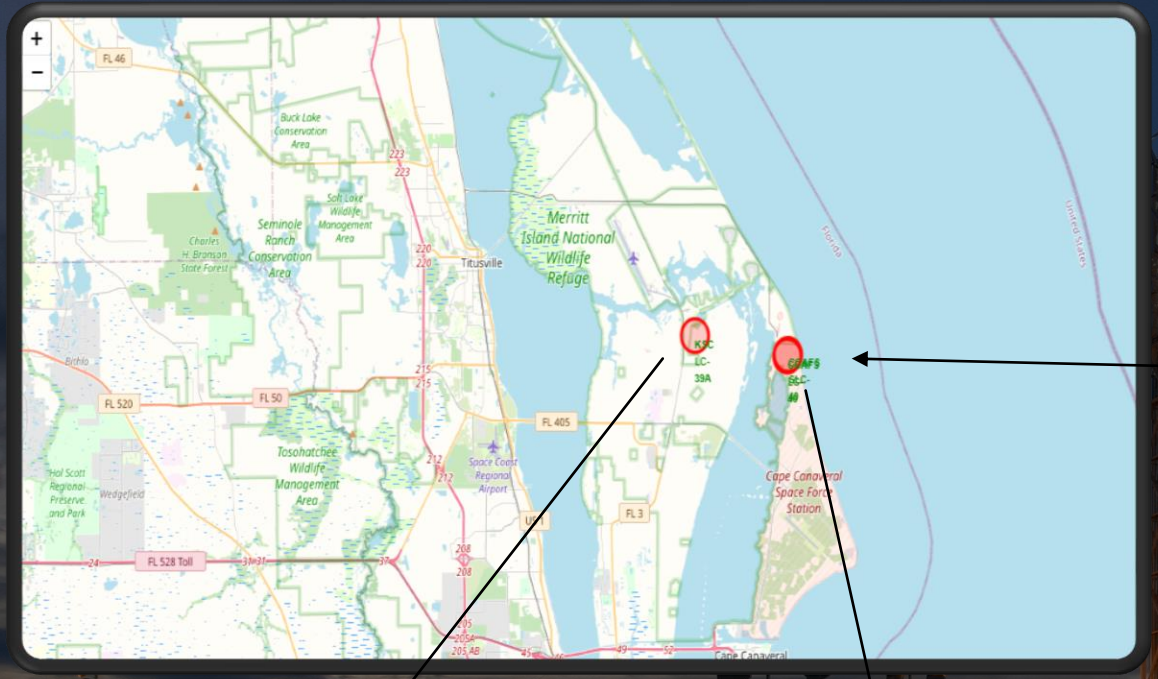


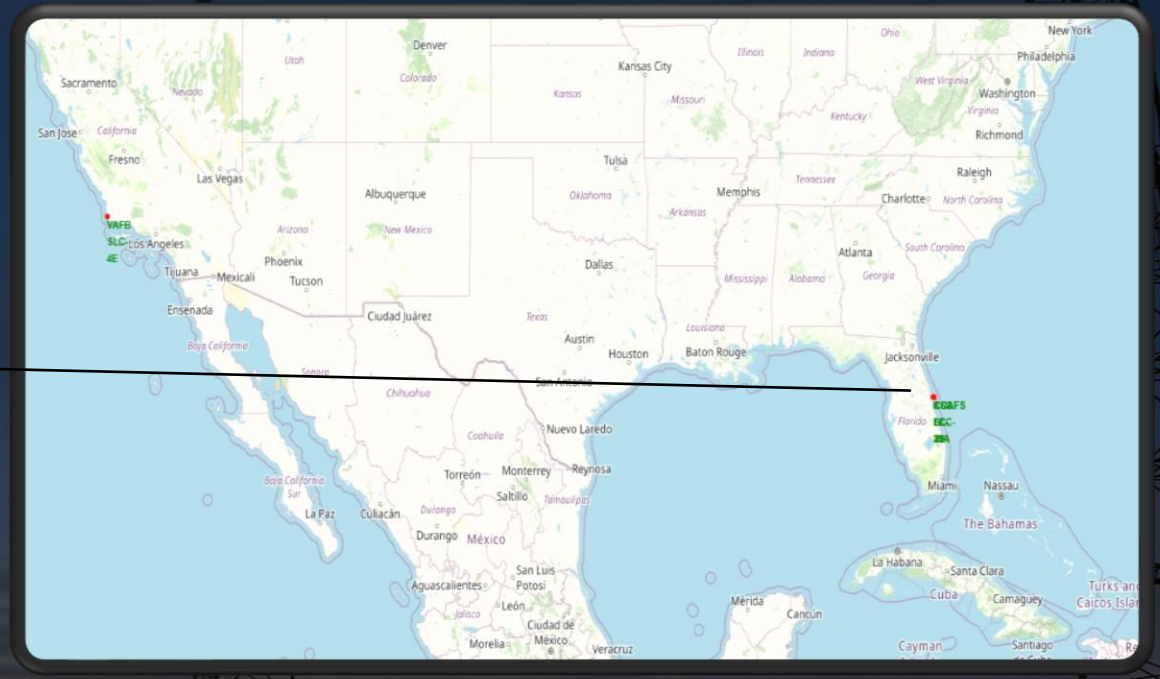


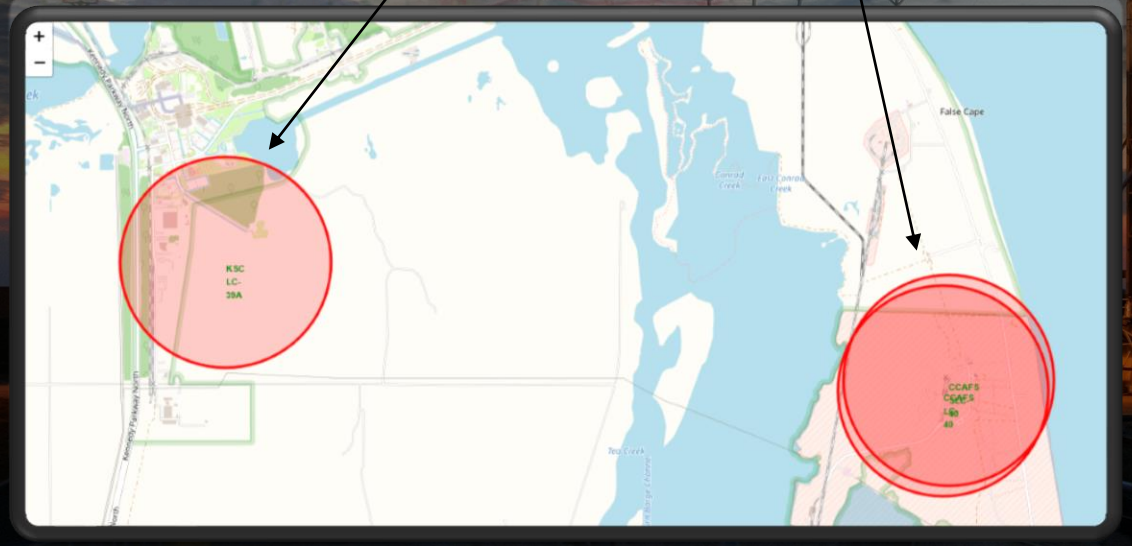


By observing launch sites on map we find that all launch sites are located near coast line and in southern part of the United States.
Ps. Zoom slides to view map in more detail.

Launch Sites – KSC LC-39A, CCAFS SLC 40 & CCAFS LC 40

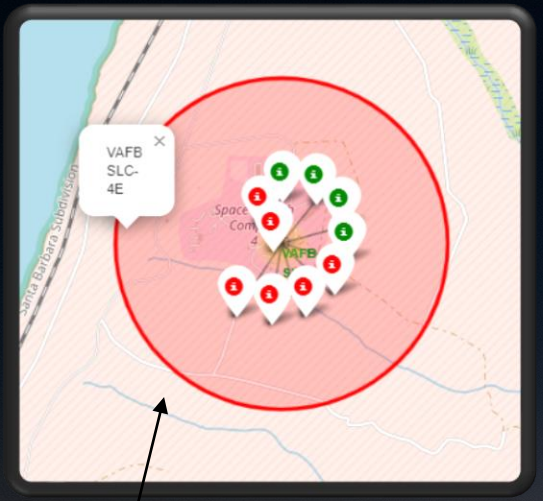






By observing launch sites on map we find that all launch sites are located near coast line and in southern part of the United States and are located in a island apart from residential areas.
Ps. Zoom slides to view map in more detail.

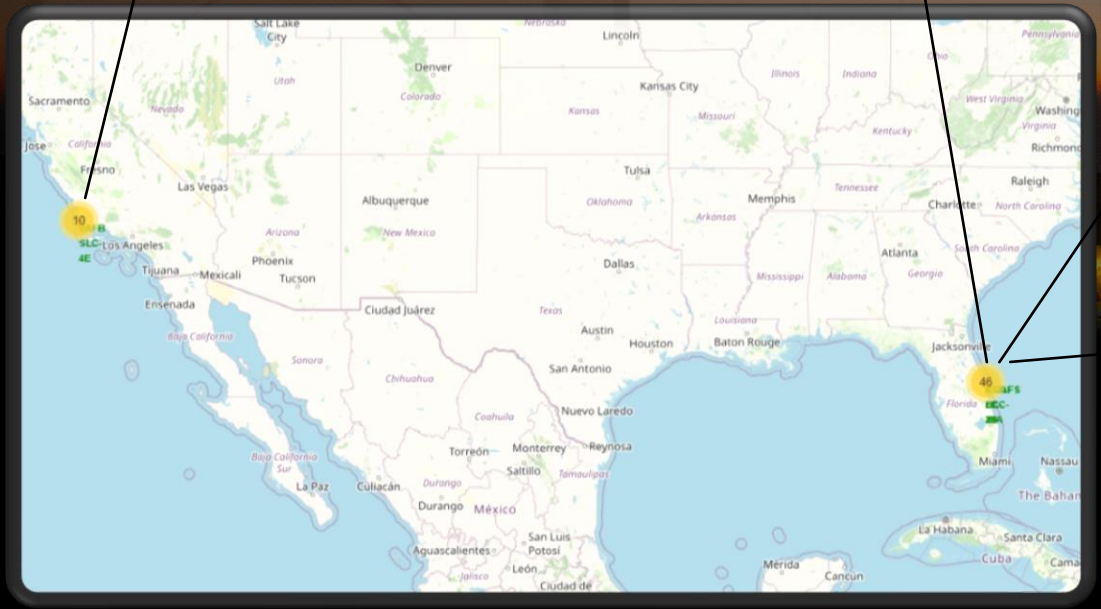
Outcome (Success/Failure) of Launch Sites

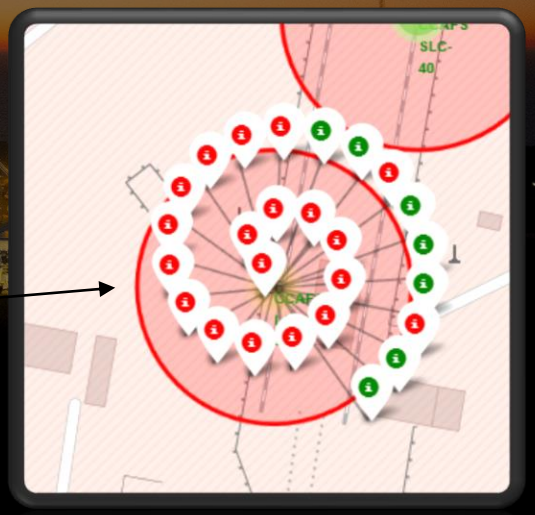


VAFB SLC 4E

KSC LC-39A

CCAFS LC 40





CCAFS SLC 40

- Here we have marked the Successful landing outcome with Green icon and failed with Red icon.
- Site KSC LC-39 have more successful lading outcome.
- Whereas, site CCAFS SLC 40 have more failed landing outcome.

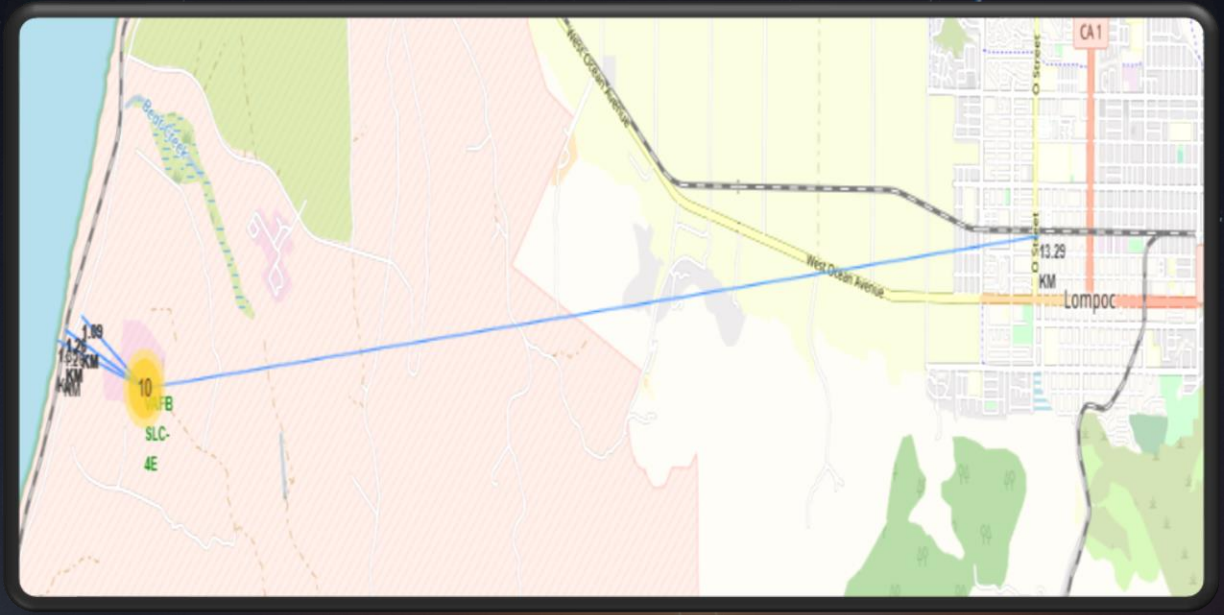
Ps. Zoom slides to view map in more detail.

PROXIMITY OF LAUNCH SITES TO OTHER POINTS OF INTEREST

Taking site VAFB SLC 4E as example here we can observe the proximity to the resources and get insights.

After see this we can answer these :

- Are launch sites in close proximity to railways?
  - Yes, The nearest Railway is 1.26 Km away.
- Are launch sites in close proximity to highways?
  - Yes, the nearest highway is 1.1 Km away.
- Are launch sites in close proximity to coastline?
  - Yes, the nearest coastline is 1.35 Km away
- Do launch sites keep certain distance away from cities?

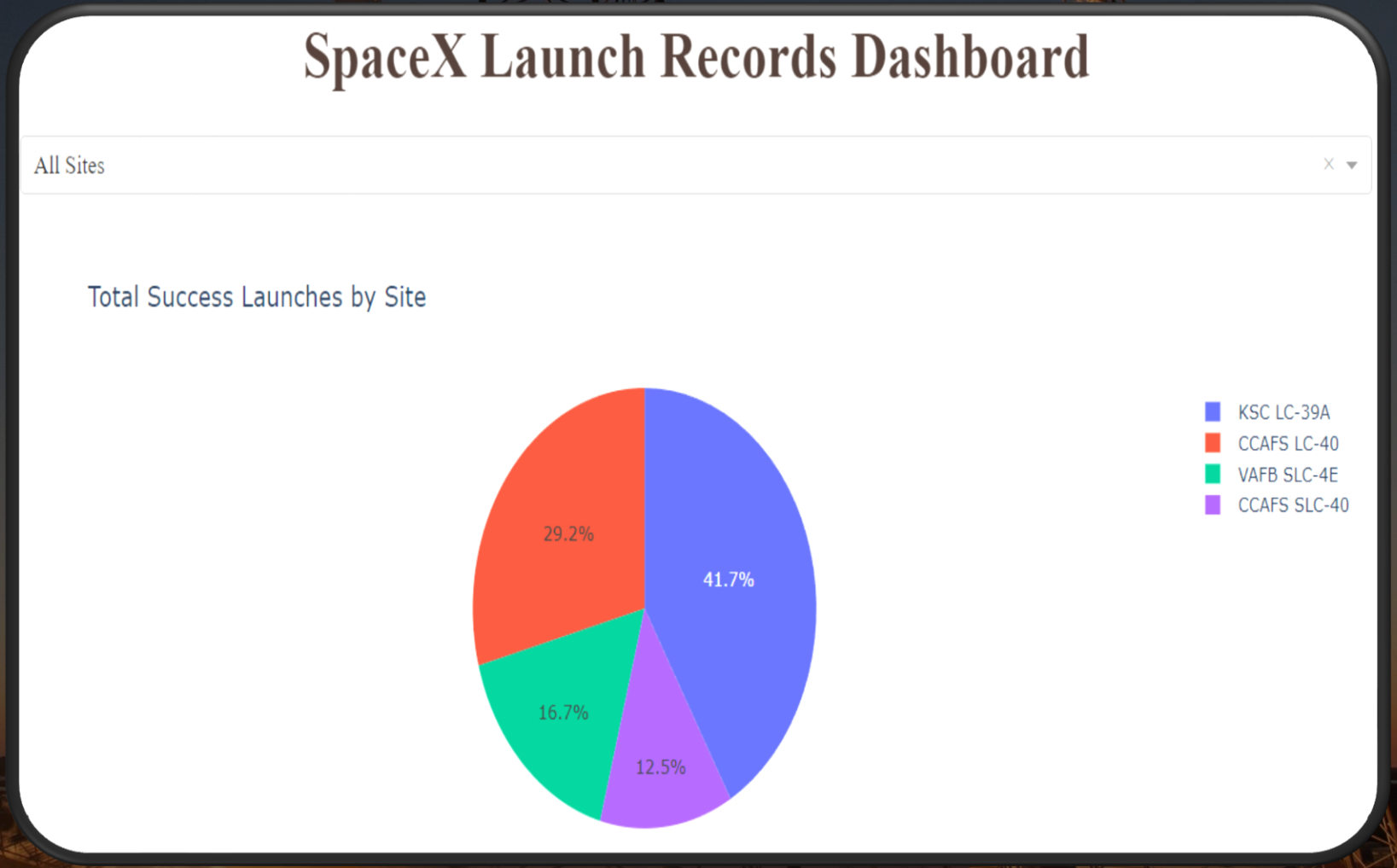


INTERACTIVE DASHBOARD WITH PLOTLY DASH

Launch Success Ratio of All Sites

The pie chart illustrates the following :

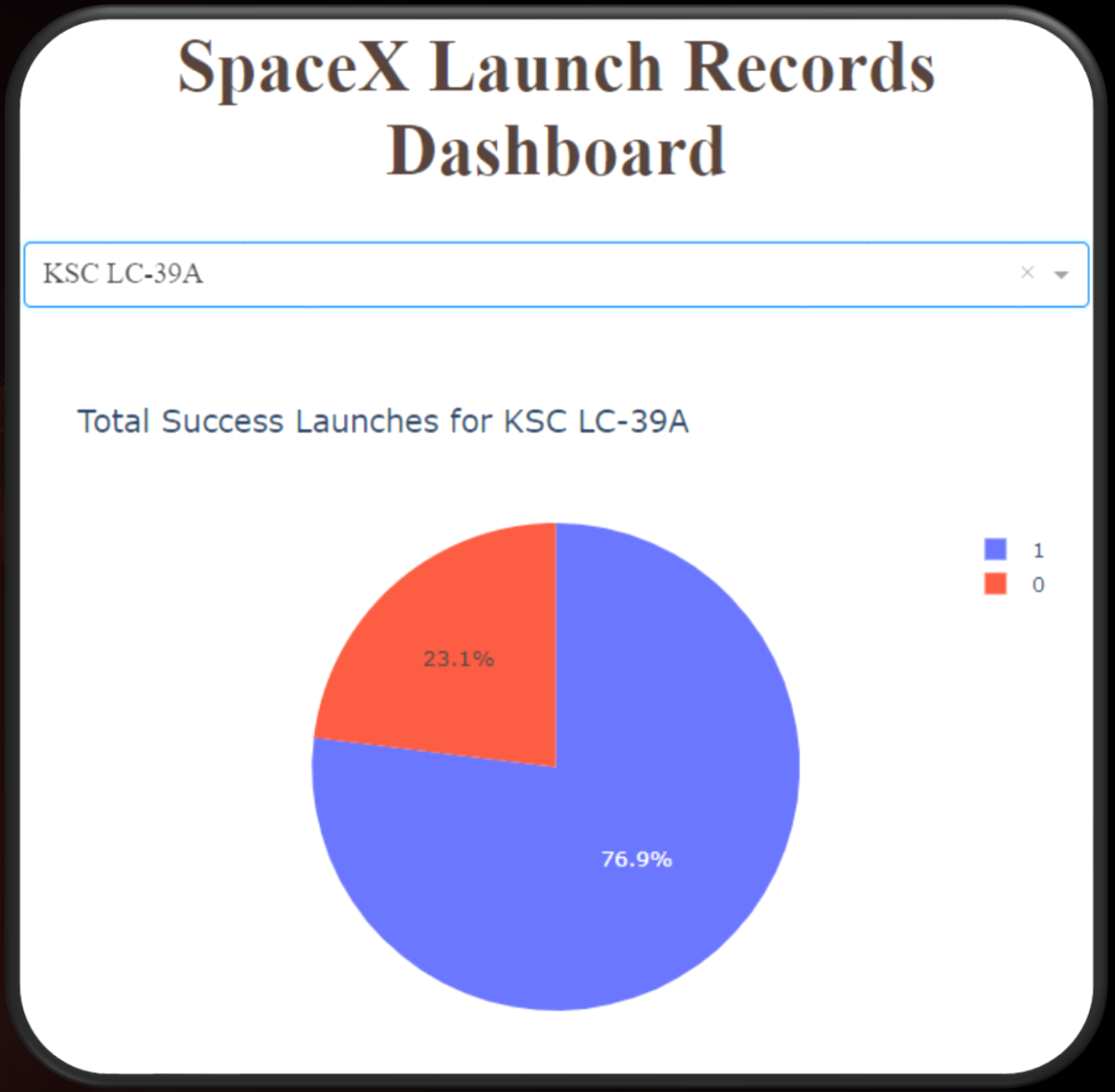
- It shows there are 4 launch sites.
  - KSC LC-39A
  - CCAFS LC – 40
  - CCAFS SLC 40
  - VAFB SLC 4E
- KSC LC – 39A has highest success ratio of 41.7%
- While, CCAFS SLC 40 have the lowest success ratio of 12.5%.



LAUNCH SITE WITH HIGHEST SUCCESS RATIO (KSC LC -39A)

The pie chart illustrates the following :

- KSC LC – 39A has highest success ratio of 76.9% of the total launches.
- While, having a failure rate of 23.1% of total launches.

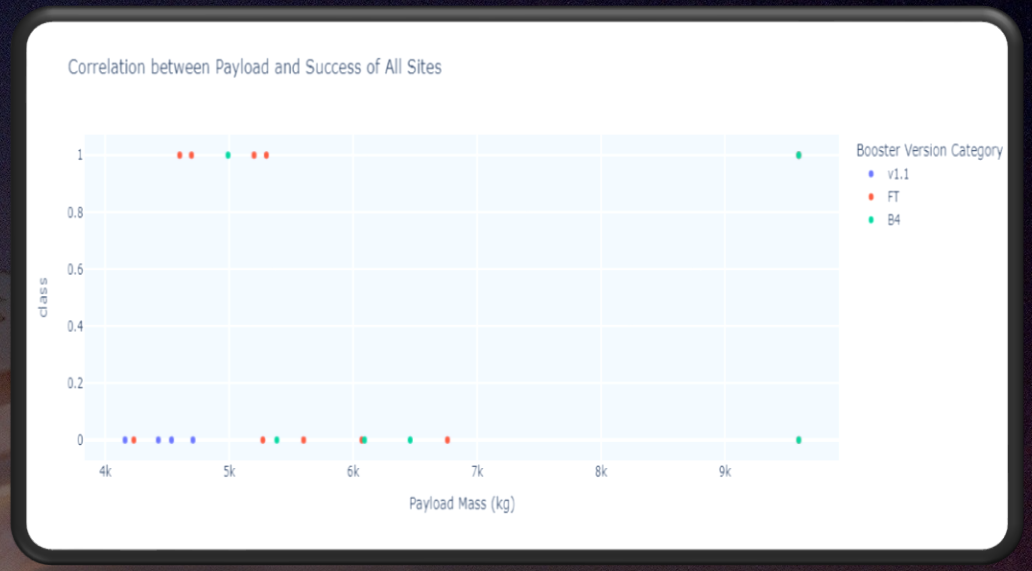


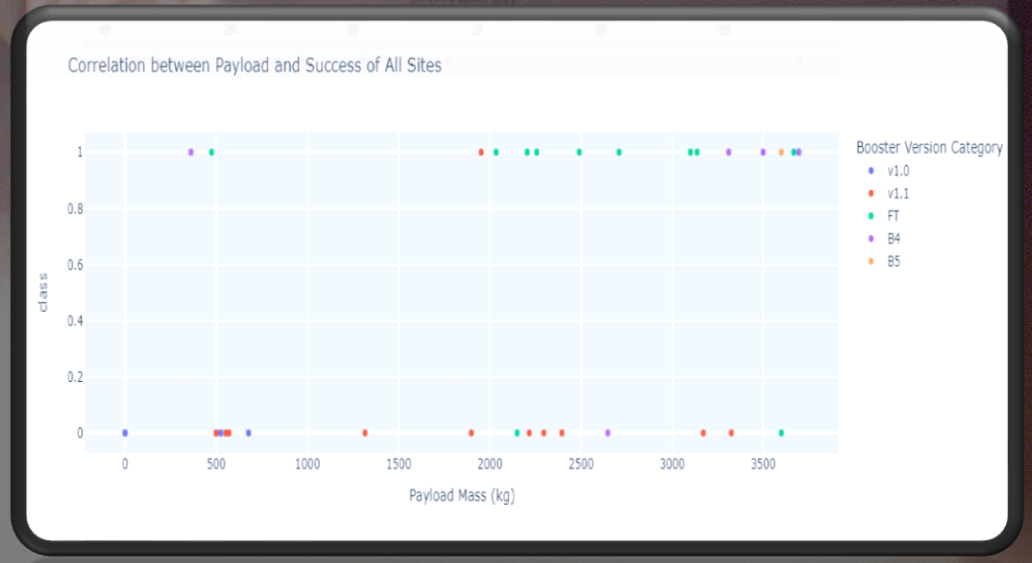
ZLSADESIGN

Launch Success Ratio of All Sites

The Scatter chart illustrates the following :

- It shows Booster version category of ‘FT’ has more success but only when Payload Mass is less than 6000 Kg.
- Whereas, v1.1 & v1.0 are most failed booster category.
- Also we can see in range of Payload Mass of 2000 to 4000 Kg there is more success.



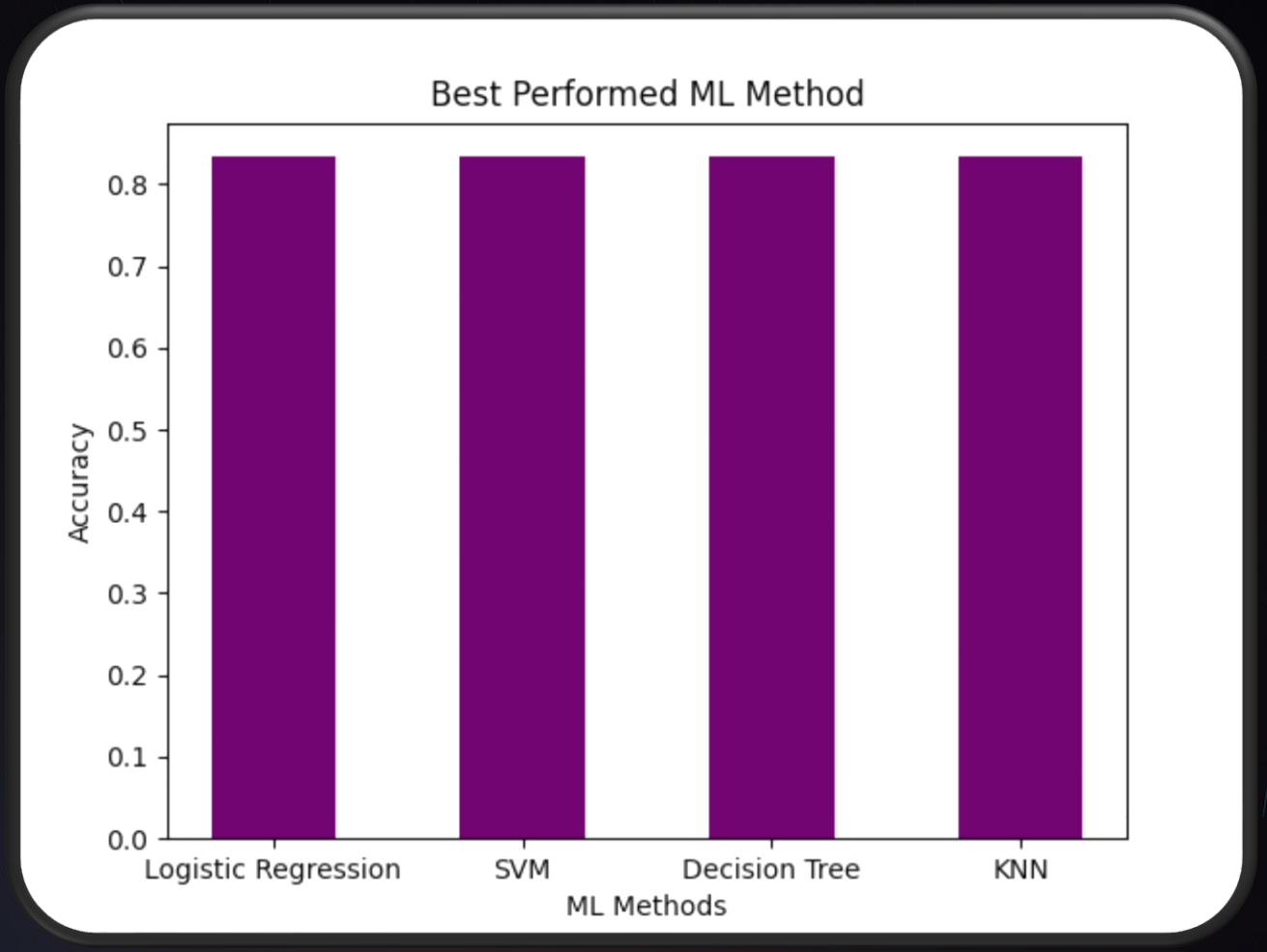


Predictive Analysis – Classification

Classification Accuracy

The Bar chart illustrates the following :

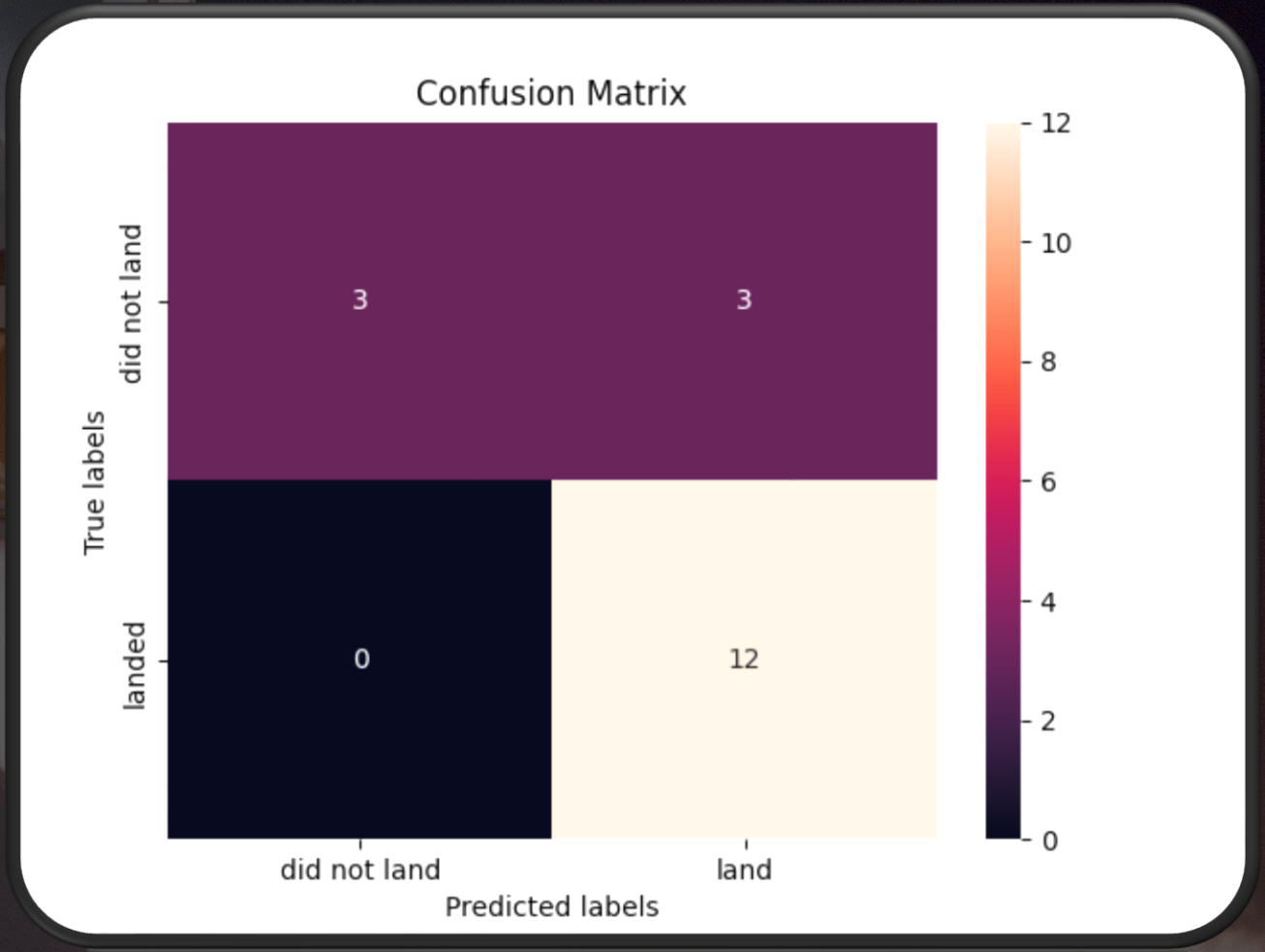
- The Models we used :
  - Logistic Regression
  - SVM
  - Decision Tree
  - KNN
- All the Model showed same accuracy of around 83%.



Confusion Matrix

As all the models have same accuracy so the matrix are same for all and matrix taken here is of KNN (K Nearest Neighbors) model.
The Confusion Matrix illustrates the following :

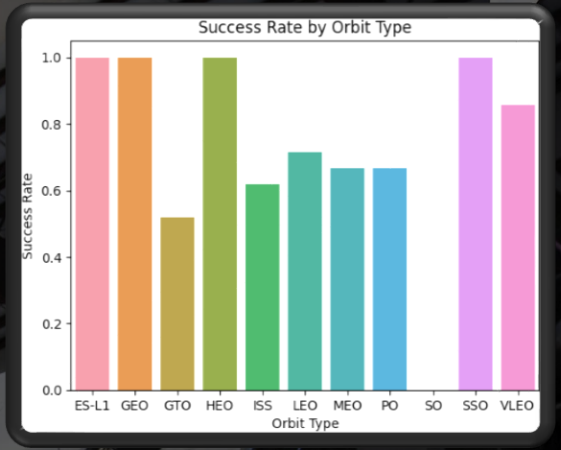
- True labels are actual values where we have :
  - Landed – 12
  - Did not land – 6
- Predicted labels are outcome we got from the model :
- Here we can see KNN model predicted more successful landings than actual and less failed landings.

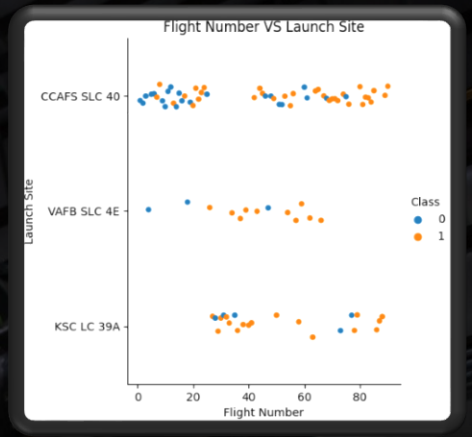


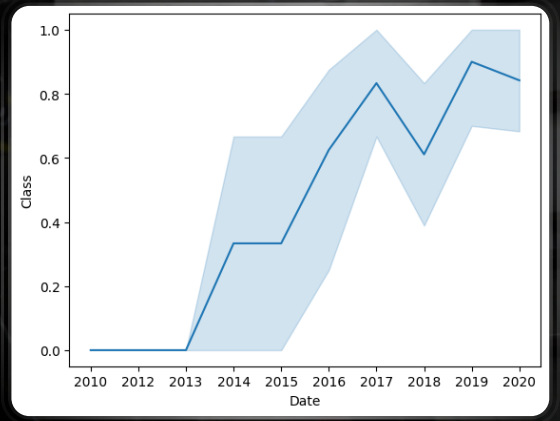
Conclusion

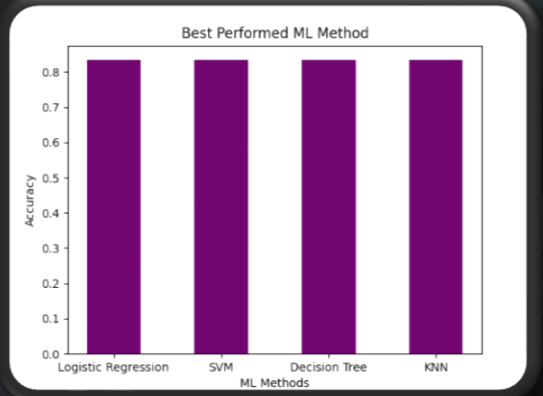
The Conclusion for this analysis are the following :

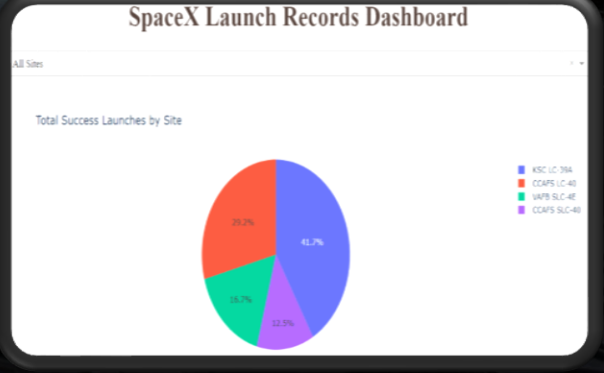
- Success rate increases as the flight number increases.
- There is no early flights in KSC LC 39A site and Success ratio of this site is better than others
- Orbit types ES-L1, GEO, HEO, and SSO, have the highest (100%) success rate.
- SSO orbit have success with lighter Payload Mass than 6000 Kg.
- After 2013 Success ratio increased and before 2013 all the landings were unsuccessful.
- All launch sites are located near coast line and in southern part of the United States.
- KSC LC – 39A has highest success ratio of 41.7% out of all launch sites launches and 76.9% success ratio of total launches at KSC LC – 39A.
- see in range of Payload Mass of 2000 to 4000 Kg there is more success.
- All the Model showed same accuracy of around 83%.











Thank you!